ABERYSTWYTH UNIVERSITY

DOCTORAL THESIS

ORFs, StORFs and Pseudogenes: Uncovering Novel Genomic Knowledge in Prokaryotic and Viral Genomes.

Author:
Nicholas John DIMONACO

Supervisors:
Prof. Christopher J. CREEVEY
Dr Amanda CLARE
Dr Wayne AUBREY
Dr Kim KENOBI
Prof. Robert HOEHNDORF
Dr Arwyn EDWARDS

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Institute of Biological, Environmental and Rural Sciences

Tuesday 19th April, 2022

Declaration of Authorship

I, Nicholas John DIMONACO, declare that this thesis titled, "ORFs, StORFs and Pseudogenes: Uncovering Novel Genomic Knowledge in Prokaryotic and Viral Genomes." and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: N J Dimonaco

Date: 19/04/2022

"I'm not interested in preserving the status quo; I want to overthrow it"

Niccolò Machiavelli

ABERYSTWYTH UNIVERSITY

Abstract

Institute of Biological, Environmental and Rural Sciences

Doctor of Philosophy

ORFs, StORFs and Pseudogenes: Uncovering Novel Genomic Knowledge in Prokaryotic and Viral Genomes.

by Nicholas John DIMONACO

Often viewed as simplistic when compared to eukaryotic genomics, the multi-faceted processes behind prokaryotic genome annotation have been re-evaluated in this thesis. In order to undertake this re-evaluation, both contemporary and novel methods of characterising prokaryotic genomic data were thoroughly investigated and developed to further our understanding of these organisms.

In Chapter 2, historic and contemporary prokaryotic genome annotation techniques are evaluated via the development of a novel genome annotation comparison and improvement platform, ORForise (https://github.com/NickJD/ORForise). The results of ORForise outlined that these techniques are effective at identifying genes that are similar to those in existing genomic databases. However, there are two key findings in Chapter 2 which point to notable inadequacies. Firstly, no single annotation tool performed best for all genomes studied, with the type of gene and organism being annotated being the most important criteria when choosing a genome annotation tool. Secondly, taking into account many of the limitations consistent among the annotation tools considered in this study, there were an unexpected number of large regions of each genome which were consistently labelled as'intergenic' or without annotation.

In Chapter 3, a thorough investigation of many of the specific weaknesses identified in the annotation tools from Chapter 2 was performed. This resulted in the identification of a set of full-length CDS gene sequences in these 'intergenic' regions which formed part of known and novel core and soft-core gene families in the *E. coli* pangenome. Additionally, a large number of highly conserved gene families were found in 'intergenic regions' across multiple genera. This adds evidence to the contention that regions of DNA labelled as 'intergenic' by existing annotation tools contain real genes and, as such, these regions were renamed 'unannotated regions'. This discovery and the redefinition of 'intergenic' regions was possible via the development and modification of two novel techniques and software platforms, ORForise and StORF-Reporter (https://github.com/NickJD/StORF-Reporter). These

allowed for the extraction of additional and novel genomic information from existing genomic databases. Additionally, as StORF-Reporter found a number of putative CDS gene fragments in these unannotated regions, Chapter 4, focuses on the absence of pseudogenes in canonical genome annotations. This uncovered thousands of potential pseudogenised and functional genes that were missed by annotation tools due to either terminating in-frame stop codons or alternative use of stop codons to code for amino acids. The results from Chapters 3 and 4 have led to the redefinition of not only the gene collection of the *E. coli* pangenome and many of the studied genera, but also may impact our understanding of their phylogeny.

To enable the discoveries and analysis in Chapters 2, 3 and 4, a number of passive computational approaches (those which can only operate alongside a rigid set of predefined rules) were used and developed. The majority of rules or parameters in these approaches were tuned through a thorough investigation of genomic features identified manually. However, biology as a domain has too many exceptions and too many rules for a passive computational approach to be universally tractable. The scale of this problem is only matched by the genomic data that we now have available. To overcome this, machine learning methodologies were investigated during a research visit to King Abdullah University of Science and Technology (KAUST) in Saudi Arabia. Specifically, the growing affinity between machine learning and biology was investigated in Chapter 5 with a novel neural network algorithm named FrameRate (`https://github.com/NickJD/FrameRate`). FrameRate was developed to offer insight into the coding potential of unassembled DNA sequences without the need for sequence homology or assembly.

Lastly, at the beginning of the current SARS-CoV-2 pandemic, an opportunity was presented to apply the skills and knowledge gained throughout the development of this thesis to the novel SARS-CoV-2 genome. Chapter 6, describes how a novel hybrid genome annotation approach which combined *ab initio* gene prediction and sequence alignment techniques was developed and used to annotate coronavirus genomes found in human, bat and pangolin hosts. Additionally, unlike other contemporary gene prediction tools, StORF-Reporter was able to identify the enigmatic ORF10 Open Reading Frame in SARS-CoV-2 without sequence alignment or RNA-Seq analysis.

Acknowledgements

- Supervisors:
  - Prof Christopher J. Creevey
  - Dr Amanda Clare
  - Dr Wayne Aubrey
  - Dr Kim Kenobi
  - Prof Robert Hoehndorf
  - Dr Arwyn Edwards
- Academic Support:
  - Dr Mazdak Salavati
  - Wang Liu-Wei
  - Dr Jessica Friedersdorff
  - Dr Francesco Rubino
  - Dr Keiron Teilo O'Shea
  - Dr Nigel Rodenhurst
- Holistic Support:
  - The Dimonaco Family
  - Heather Phillips
- Technical Support:
  - Dr Colin Sauze: Institute of Biological, Environmental and Rural Sciences (IBERS Aberystwyth) High Performance Computing Facilities
  - Supercomputing Wales
  - King Abdullah University of Science and Technology: KAUST Supercomputing Lab (KSL)
- Funding and Awards:
  - Institute of Biological, Environmental and Rural Sciences: PhD Fellowship
  - BCS, The Chartered Institute for IT: Travel Grant
  - Worshipful Livery Company of Wales: Travel Scholarship
  - Aberystwyth Old Students' Association: Scholarship

Contents

List of Figures

List of Tables

Open Source Research Statement

One of the most underdiscussed but important aspects of modern science is the availability, re-usability and quality of software and data. Throughout this thesis, I have made use of a number of open source software which I have endeavored to list and describe as best as possible. I also have committed to present all software I have developed openly and have it available on my Github page (https://github.com/NickJD). While some chapters have their own specific repository listed, the 'Bioinformatic-Tools' repository which has a number of tools that were used at some point by all of the chapters. Additionally, the databases versions I have used across the different chapters are also listed - see the table below.

Title	Type	Source	Versions	Chapters
Bioinformatic-Tools	Code Repository	Github	N/A	All
ORForise	Code Repository	Github	v1.0.0	2, 3, 4
StORF-Reporter	Code Repository	Github	v.9.0	3, 4
FrameRate	Code Repository	Github	v0.9.0	5
CoronaHack	Code Repository	Github	N/A	6
Ensembl Bacteria	Genome Database	Ensembl Bacteria	Release 46	All
Swiss-Prot	Protein Database	UniProt	2020_01	All

TABLE 1: List of software developed and databases used for this thesis. Chapter identifier 'All' is used to mark the version of the resource used across the entire thesis, irrespective of its use in every chapter.

List of Abbreviations

ORF	**O**pen **R**eading **F**rame
StORF	**S**top-**O**pen **R**eading **F**rame
Con-StORF	**C**onsecutive - **S**top **O**pen **R**eading **F**rame
CDS	**C**o**D**ing **S**equence
PCG	**P**rotein **C**oding **G**ene
HGT	**H**orizontal **G**ene **T**ransfer
PGAP	**P**rokaryotic **G**enome **A**nnotation **P**ipeline
StORF	**S**top **O**pen **R**eading **F**rame
CF	**C**oding **F**rame
NCF	**N**on **C**oding **F**rame
FR	**F**rame **R**ate
NGS	**N**ext **G**eneration **S**equencing
RT	**R**ead **T**hrough
LGT	**L**ateral **G**ene **T**ransfer
NCBI	**N**ational **C**enter for **B**iotechnology **I**nformation
GFF	**G**eneral **F**eature **F**ormat
GTF	**G**eneral **T**ransfer **F**ormat
MO	**M**odel **O**rganism
BLAST	**B**asic **L**ocal **A**lingment **S**earch **T**ool
CEA	**C**urrent **E**nsembl **A**nnotation
IR	**I**ntergenic **R**egions
UR	**U**nannotated **R**egions
COG	**C**lusters of **O**rthologous **G**enes
SNP	**S**ingle **N**ucleotide **P**olymorphism
OSC	**O**ut of frame **S**top **C**odon
RBS	**R**ibosomal **B**inding **S**ite
CNN	**C**onvolutional **N**eural **N**etwork
ACE2	**A**ngiotensin **C**onverting **E**nzyme
CoV	**C**orona**V**irus
DB	**D**ata**B**iology
E	**E**nvelope
M	**M**embrane
MERS-CoV	**M**iddle **E**ast **R**espiratory **S**yndrome **C**oronavirus
N	**N**ucleocapsid
PCA	**P**rinciple **C**omponent **A**nalysis
RaTG13	SARSr-**R**a-BatCoV-Ra**TG13**

Chapter 1

Background

1.1 Prokaryotes, their Importance and Study

The domains of bacteria and archaea, collectively referred to as prokaryotes, have fundamental roles in almost every life cycle on earth, and have such long been and continues to be at the forefront of the scientific communities attention (Trüper, 1992; Ollivier et al., 2018). This importance and study of prokaryotic life-forms has only increased due to their impact on contemporary world-wide problems such as greenhouse gas production, antibiotic resistance and prevention, and their utility in industrial processes. These studies, while diverse and contrasting in aim and scope, have led to the expansion of ecological and evolutionary understanding of this diverse division of life and have shown the contrasting affects they have on both humans and the planet as a whole.

The common position that prokaryotes are simple organisms, reflected by their presumed genome simplicity and their lack of many of the complexities known to eukaryota study, have led to an abundance of tools and methodologies to study them, especially in the form of genome annotation. With this in mind, the scientific community's efforts have been focused on discovering new genomes from evermore niche environments, and comparatively little work has been done to evaluate and refine either the genomic knowledge already accumulated or and methods devised to obtain such knowledge already accumulated over the past four decades.

1.2 Sequencing and Assembly

The practice of genome sequencing has seen dramatic changes over the past few decades, in part due to the increased affordability and throughput of new technologies (Land et al., 2015; Goodwin, McPherson, and McCombie, 2016). Once the realm of large corporations and government-backed research groups, it is now commonplace for entire prokaryotic genomes and environmental metagenomes to be sequenced, assembled, and annotated by small university labs. However, as previously reported, while the direct cost of the ever growing rate and capacity of genomic sequencing continues to decrease, that of storage and analysis is not decreasing as rapidly (Sboner et al., 2011). Although these advances in DNA sequencing

and assembly have made it easier and faster to sequence and assemble prokaryote genomes specifically at an accelerated rate, this does not guarantee their quality (Denton et al., 2014). Furthermore, the advent of Nanopore and the cost and logistics it requires, has enabled the era of classroom genetics, continuing the availability of DNA sequence data to ever increasing numbers of studies (Zaaijer et al., 2016). This democratisation of genomic sequencing, whilst enabling the independent study of both niche and canonical specimens, has facilitated the export and proliferation of errors and bad practice. Such examples of this have been found at least as far back as the late 1990s (Médigue et al., 1999) and while the overall rate of genomic assembly has increased in the subsequent decade, so has the number of low quality and fragmented genomes (Klassen and Currie, 2012). Although specific gene prediction tools such as FragGeneScan (Rho, Tang, and Ye, 2010) have had some success in contending with these fragmentary genomes, they still are often reported with substantially numbers of genes (Denton et al., 2014).

As with all fields of study, human choice is a core factor of what gets sequenced. This, along with the inherent difficulty in isolating, culturing and sequencing the majority of the earth's microbiome, has led to only a fraction of life being sequenced (Lewin et al., 2018). Projects such as the Earth Microbiome Project (Thompson et al., 2017) and the Tara Oceans project (Sunagawa et al., 2020) have made great strides in sampling and sequencing yet to be studied parts of the Earth's microbiome. However, there is still no agreement on what proportion of specimens, and therefore genomic diversity from any environmental sample, are actually culturable (Martiny, 2019). While metagenomic sequencing has contributed to pushing the frontiers of microbial study through the enabling of genome assembly without the need for specimen isolation, high quality assemblies are still difficult to produce and often exhibit high levels of error and cross-contamination. The true number of arguably 'complete' prokaryotic genome assemblies is only a fraction of what is presented in the databases, and furthermore, long term biases exist towards specific species which have been more tractable and of scientific or industrial interest, thus targeting them for culturing studies (Amann, Ludwig, and Schleifer, 1995). The resulting impact on the utility and quality of downstream analyses such as genome annotation is still being felt today:

> *"The composition of the reference databases is not representative of the species composition of the natural world, but rather reflects a focus on human pathogens, other species of interest to humans, and the challenges of isolating and sequencing DNA from various species" (Lu and Salzberg, 2018).*

As with DNA sequencing, several approaches have been developed for genome assembly, many in tandem with NGS technologies (Miller, Koren, and Sutton, 2010). Some are designed specifically for eukaryote, plant or prokaryote genomes while

others are sequencing technology specific. While prokaryote genomes are structurally simpler than those of eukaryota, in regards to genome assembly, most often lacking separate chromosomes and being of much reduced size, their variability due in part to their increased mutation rate, still produces significant difficulties in assembly. However, "Regardless of the adopted method, obtaining a genome draft with few errors depends on the quality of data generated in the sequencing." (Carneiro et al., 2012). Quality control and read trimming are commonplace and effective at reducing suspect error in the input data going into assembly platforms. However, this still constitutes one of the most overlooked stages of genomic study and is often only applied to the raw data itself and not subsequent assembly or alignment (Guo et al., 2014). As noted by (Salzberg, 2019), "Paradoxically, the incredibly rapid improvements in genome sequencing technology have made genome annotation less, not more, accurate.". An example of this can be found in one study on the interchangeability of next generation sequencing (NGS) and genomics approaches for HIV surveillance, which has suggested that "The low inter-laboratory reproducibility of NGS sequences may also be at least partly related to input amplifiable copy number.... and differences in the bioinformatics pipelines used." (Parkin et al., 2020). While differences between studies due to bioinformatic approaches may be somewhat mitigated by further steps of comparative study and error checking, those observed between studies utilising undersampled data such as viral genomes (Sutton et al., 2019) and metagenomes (Walt et al., 2017) are substantially more difficult to resolve.

1.3 Prokaryote Genome Annotation

The value of NGS technologies and contemporary assembly methods, irrespective of resulting genome assembly quality, can only be fully realised if genome annotation methods are also improved.

The process of annotating genomes, that of eukaryotes, prokaryotes or viruses while having been long considered as non-trivial, has not changed in aim or scope for over two decades. However, it is broad in subject, covering the detection of genes (protein coding, pseudogene, rRNA, tRNA etc.), promoters, operons, structural elements, and other more cryptic genomic elements. Therefore, unsurprisingly, there are now many approaches for characterising genomes in increasingly automated fashions. Nonetheless, some such approaches have either changed little or are still routinely used decades on, even when more recent methods have become available (Olsen et al., 2020; Shariat, Timme, and Walters, 2021). The lack of progress in these approaches may be indicative of the fact that advances in other fields of computing and genomics over the same period have not yet had a significant impact on genome annotation (Salzberg, 2019). The sampling of cryptic species and metagenomeic environments, which involves the broader and deeper sequencing and assembly of lesser-known and novel genomes, often without close relatives for reference, has

also presented a problem for gene and function annotation (Carr and Borenstein, 2014).

The work in this thesis focuses on the specific challenge of CoDing Sequence (CDS) gene (otherwise known as protein coding gene) detection in prokaryote genome annotation.

1.3.1 CoDing Sequence Gene Prediction

Before the enormous stimulus provided to the field of DNA sequencing, instigated by the Human Genome Project which saw the rapid development of biological tools and techniques for genome sequencing (Lander et al., 2001; Hood and Rowen, 2013), there were only a small number of species with a substantial region of their genome sequenced, let alone assembled and annotated. However, even these relatively small regions of DNA required an annotation. In its earliest form, back in the early 1980's, CDS prediction was a specalised but simple process, basic enough to be undertaken without a computer (Fickett, 1982). Statistical nucleotide order and patterns, among other features, which were laboriously identified, often by hand, were used to distinguish what was common in coding DNA but not in non-coding DNA (Staden, 1984; Gribskov, Devereux, and Burgess, 1984; Fickett, 1982). This was then used to infer whether a particular region of DNA was likely coding or non-coding and was seen as a "decision procedure, which when presented with a DNA sequence, would classify it as either coding or non-coding" (Fickett, 1982). These approaches could be considered as the origin of gene prediction. However, the laborious nature of these approaches and the need for computation, an uncommon resource at the time, limited its use and often did not allow for the identification of the origin or terminus of transcription.

Spurred by the rapid increase of computing potential and availability during the late 1990's and the early 2000's, computational biology, and specifically automated CDS prediction, not only became more feasible but also provided the opportunity to annotate entire genomes. However, some of the first computational gene predictors were still developed explicitly for select model organisms (Fickett, 1982; Krogh, Mian, and Haussler, 1994; Salamov and Solovyev, 1997). Forming the bedrock of contemporary computational genome annotation, these methodologies struggled to identify genes in genomes which diverged from the original model organism they were built for. As the databases grew in size, researchers were then able to use sequence alignment methods such as BLAST (Altschul et al., 1990) to identify potential CDSs (Robison, Gilbert, and Church, 1994). However, due to the vast diversity of CDS gene sequences, many gene types were undetected and this has likely led to a contemporary problem where these omitted genes continue to be absent or underrepresented in public databases (Warren et al., 2010; Huvet and Stumpf, 2014). For example, even a model organism as well studied as *Escherichia coli (E. coli)* still contains a large proportion (35%) of hypothetical functionally uncharacterised genes

in their genomic annotations, more than twenty years after these methods were first used (Ghatak et al., 2019).

The release of the computationally efficient *ab initio* algorithm of GeneMarkS (2001) was groundbreaking. The historic use of GeneMarkS (2001) by numerous genomic repositories and studies (Tatusova et al., 2016) has likely led to many of the more accurate and recently developed tools being influenced by the predictions, assumptions, and genomic knowledge sequestered by the tool and the other pioneering gene predictors of the 1990's and early 2000's. Most of the well established genome annotation methods and pipelines rely on experimentally annotated genes to functionally characterise their predictions via these methods. However, the scope and variability of the genomic data currently held across databases captures only a fraction of what is in nature. Furthermore, the creators of the original GeneMark tool, which was designed for specific model organism study, stated in 1993 that:

> *"To achieve good results the sequence to be analysed should be taken from the same statistical population as the training set is. So, one cannot expect that the algorithm trained on E. coli sequence set will be successfully applied to the sequence taken from the genome of the other species." (Borodovsky and McIninch, 1993)*

Furthermore, since the rapid decline in cost and increasing access to genome sequencing in recent years, there has been the facilitation of a large number of both complete and draft prokaryote genomes into public databases. The genomes of most of these species have been sequenced and annotated by relatively large sequencing centers. However, many smaller centers and even individual laboratories, such as clinics, have contributed and continue to do so at a growing rate (Shendure and Ji, 2008; Fricke and Rasko, 2014; Kwong et al., 2015). Many of these do not have substantial in-house bioinformatics expertise and rely on different approaches which are often 'black box', meaning the reasoning behind the decisions made by the tools are unknown to the researchers. As such, the comparison between the different tools is challenging and are not routinely compared. This can lead to a lack of understanding of the consequences of using each tool and how they affect the resultant annotations in unforeseen ways. The annotation process can vary greatly from one center to the next, and even within a center it varies from year to year, with different programs used for gene finding, alignment, and assigning gene names. The continuing reduction in the cost of sequencing suggests that the trend towards sequencing by small laboratories will increase substantially in the future. This combination of factors has resulted in a mixture of biases which have recently been accepted in historic and contemporary genomic database entries and bioinformatic methods (Stoeger et al., 2018). However, studies investigating the level of bias are nuanced to each of their domain and do not yet offer any definitive solutions (Schnoes et al., 2013; Ross et al., 2013; Troudet et al., 2017; Haynes, Tomczak, and Khatri, 2018).

1.3.1.1 Open Reading Frames and CoDing Sequence Genes

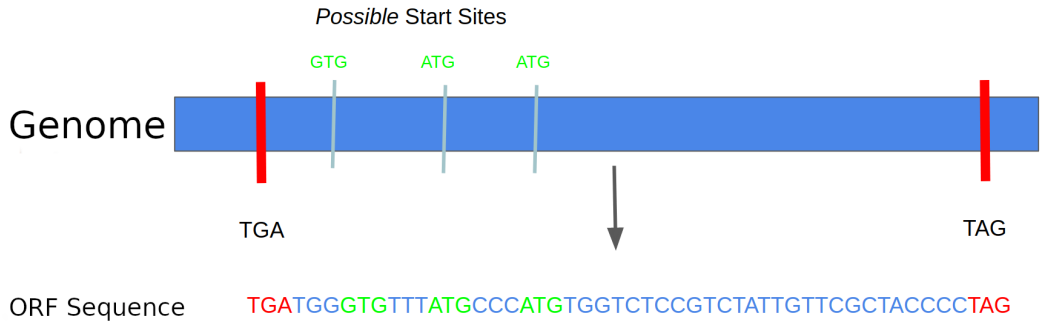


FIGURE 1.1: Diagram presenting the Sequence Ontology definition of an Open Reading Frame (ORF) bounded by two inframe stop codons. Three potential start codons are displayed.

Despite the characterisation of an increasing number of non-coding genes in prokaryote genomes, often the first step in any genome annotation is identification of all possible Open Reading Frames (ORFs). ORFs have long been interpreted differently, according to their use case and users. Typically known to represent the genomic locus of a CDS gene from a contiguous sequence of DNA (Furuno et al., 2003), the nomenclature has often been debated and has recently been revisited once again, with the conclusion that an ORF is bounded by two stop codons (Sieber, Platzer, and Schuster, 2018) (see Figure 1.1). The Sequence Ontology (Eilbeck et al., 2005) (used to define elements in genome annotation formats such as GFF and GTF) also describes an ORF as "The in-frame interval between the stop codons of a reading frame which, when read as sequential triplets, has the potential to encode a sequential string of amino acids". However, it is still the norm for ORFs to be reported by genome annotation tools and in canonical genome annotations, as regions of DNA bounded by an in-frame start and stop codon (as a start codon is expected to indicate the start of DNA transcription (Brent, 2005)).

Searching for an ORF, whilst a technically simple process, is not all that is needed to detect a CDS gene. As reported over two decades ago (Fickett, 1995), the presence of an ORF (regardless of the exact definition) does not indicate the presence of a gene, let alone one with protein coding potential. Without some form of filtering, a nominal length of DNA can harbor a high number of ORFs (including those which overlap and are nested within others) but only a fraction would be protein coding. For this matter, it is important to consider that without *in vivo* experimentation, it is extremely difficult to determine whether any specific predicted ORF, without homology to a previously validated gene, is in fact protein coding, let alone its function (Eisenhaber, 2006; Lee, Redfern, and Orengo, 2007) .

Due to the overlap between the definitions of an ORF and a CDS, in this thesis I refer to the ORFs yet to undergo any form of selection such as sequence alignment,

homology or codon usage analysis as ORFs and those which have as CDSs. This definition is in line with other studies which have been conducted (Andrews and Rothnagel, 2014; Sieber, Platzer, and Schuster, 2018).

1.3.1.2 GC Content, Alternative Genetic Codes and Codon Usage

The differences in GC content observed between species, strains and core/non-core genes are likely to have downstream implications for comparative, metagenomic and pangenomic studies in which certain ORFs or start and stop codons may be incorrectly prioritised. Unsurprisingly then, genome-wide GC content, alternative genetic codes and codon usage variability across prokaryota has been investigated and used by a number of studies for its impact and importance in gene prediction (Borodovsky and McIninch, 1993; Delcher et al., 2007; Hyatt et al., 2010).

CDS gene prediction involves a number of steps, often the first of which - start site identification, has been a problem for prokaryote annotation for decades (Hannenhalli et al., 1999). Certain genetic codes in prokaryotes, such as the codons which initiate translation of a protein coding gene, were often believed to be universal since the beginning of their identification in the 1960's (Adams and Capecchi, 1966; Nirenberg and Leder, 1964). However, from the first complete *E. coli* genome sequence in the 1990s, the concept of the three canonical start codons was at odds with the detection of at least one ATT and possibly a CTG codon being used for translation initiation (Blattner et al., 1997). Later, translation from non-canonical start codons was investigated with a study of the remaining 61 codons that quantified the translation initiation of the green fluorescent protein and nanoluciferase in *E. coli* (Hecht et al., 2017). Initiation of protein synthesis above measurement background was detected for 47 codons, albeit in lab conditions. While translation from non-canonical start codons was at much lower levels, ranging from 0.007% to 3% relative to translation from the AUG (ATG), it did show that translation initiation and alternative genetic codes and codon usages are complex processes we still do not fully understand. Significantly, translation from 17 non-AUG codons exceeded the highest reported rates of UUG (TTG) (another prokaryote canonical start codon) recognition, indicating that under different environmental pressures, alternative peptides may be expressed from a single ORF. Also of note is that recent research has shown that it is possible for a gene to utilise multiple different start sites to encode for "genes-within-genes... [which] may carry important functions" (Meydan, Vazquez-Laslop, and Mankin, 2018). Although most bacterial genomes are still reported in repositories to use the start codon 'ATG' for around 80% of their CDS genes, there are more and more species and even gene families which have been shown to use very different start codon profiles (Villegas and Kropinski, 2008). However, while specific codon tables have been developed, such as 'The Mold, Protozoan, and Coelenterate Mitochondrial Code and the *Mycoplasma/Spiroplasma* Code' (Pritchard et al., 1990) or

genetic code 4, the apparent rarity of species and genes which do not use the universal codon table has made specific development for tools to handle such (especially without user-direction), not a priority.

Although the archaea ribosome is closer to that of the eukaryotic type than that of bacteria, it has shown to have similar start codon profiles to bacteria (Schmitt et al., 2020). While the process of start codon selection has many different pressures (environmental constraints, taxonomic and gene family preference, translation initiation factors, mutation rate, tRNA availability, and horizontal gene transfer (HGT) (Bentele et al., 2013; Belinky, Rogozin, and Koonin, 2017; Villegas and Kropinski, 2008; Panicker, Browning, and Markham, 2015)), recent studies have continued to confirm that GC content has one of the most visible influences on the selection of amino acids an organism can use and therefore is likely to be a driving factor in prokaryote evolutionary processes (Du et al., 2018). Interestingly, in *E. coli*, it has been posited that rare, low-GC codons are selected specifically to initiate transcription due to their inherent ability to reduce over genome GC composition (Bentele et al., 2013). However, a GC content between 30-60% does not seem to have any significant impact on codon usage, while at extreme low and high GC ($< 30\%$ & $> 80\%$) respectively, ATG and GTG are often more prominent (Villegas and Kropinski, 2008). Furthermore, GC variability is not only observed between coding or non-coding regions, but also core genes in particular often exhibit lower GC variability compared to non-core (Bohlin et al., 2017). Lastly, alternative start codon selection has been shown to have a number of genomic impacts such as the requirement for mutational compensation in the Shine-Dalgarno sequence towards a stronger translation initiation signal and possibly an effect on the rate of mutation for such genes with alternative start codons (Belinky, Rogozin, and Koonin, 2017).

Studies into alternative genetic codes which specifically focus on start codon usage are less common and studied than alternative stop codon usage. The identification of stop codons that have been reassigned to code for amino acids is an inherently different problem. Instead of simply designating an amino acid coding codon to a translation initiation site (which still effectively codes for methionine), stop codon recoding utilises a codon canonically thought to be without amino acid assignment (Dybvig and Voelker, 1996). As with most codons, it could be assumed that the availability of the canonical stop codons are GC-content dependent, with it previously being shown that the use of TAA decreases and TGA increases in alignment with GC content. However, the frequency of TAG seems to be independent of GC content. This low frequency, along with its low usage in most lineages, suggests that TAG is universally suboptimal in bacteria (Povolotskaya et al., 2012). In addition to this, while both TAG and TGA contain the same nucleotides and therefore should be equally dependent on genome GC content, another study found variation in their use. "The frequency of use of TGA in the gene sequences generally increased with the GC content of the chromosome, while the frequency of use of TAG, like that

of TAA, was inversely proportional to the GC content" (Wong et al., 2008). In another study, the underuse of TAG codons has also been linked to the greater propensity for TAG to be misread *in vivo*. A number of codons in a gene were replaced with TAG and showed significantly higher activity compared to wild-type. As this would only be possible if the protein synthesis did not stop at the TAG codon, TAG has the potential to be a hotspot of "mistranslation" (Kramer and Farabaugh, 2007). Interestingly, however, there are conflicting examples such as the *Mycoplasma* genus which, while often exhibiting a low GC content between 20-40%, recodes TGA as tryptophan, which complicates this process even further (see Figure 1.2). Although genetic codes are often assumed to be species specific, in a two-decade old study, *Mycoplasma* genes with internal TGA codons were expressed as full-length protein products (with relatively low efficiency) in recombinant wild-type *Bacillus* genomes (Kannan and Baseman, 2000). Lastly, the frequency of TAG and TGA codons has been shown to correlate well with the amount of available mRNA and protein for the release factors RF1 and RF2 during exponential growth (Korkmaz et al., 2014).

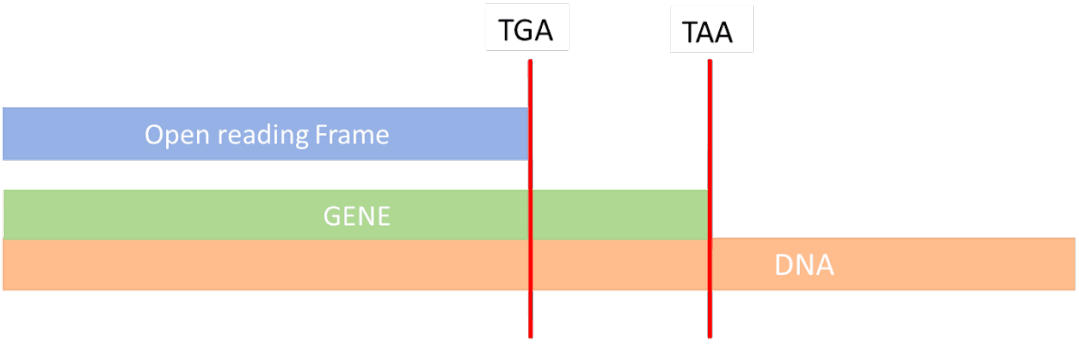


FIGURE 1.2: Presented here is a case of an organism which uses the 'non-standard' genetic code 4 that reassigns a canonical stop codon, unknown to the genome annotation method. In this instance, the canonical stop codon 'TGA' is reassigned to code for an amino acid and so the true CDS gene continues to the correct 'TAA' stop codon, while the predicted ORF is prematurely truncated.

1.3.1.3 Translational Readthrough

The process of protein synthesis termination is imperfect. Recognition of translation termination sites are known to be suppressed by a number of natural mechanisms, including ribosomal frameshifting, suppressor tRNAs (aminoacylated tRNAs with anticodons complementary to stop codons in mRNA), programmed stop codon readthrough (RT) and even the presence of certain upstream codons (Dabrowski, Bukowy-Bieryllo, and Zietkiewicz, 2015). Translational readthrough, the process that enables the ribosome to pass through a conventional termination codon in mRNA and continue translation to the next stop codon in the same reading frame, is just another example of how the theory of 'one gene equals one protein' no longer fits with contemporary understanding. While it has been long known that viruses, and in

particular RNA viruses seem to utilise translational readthrough to express different proteins to their advantage, it is often an underrepresented field in prokaryote studies (Yoshinaka et al., 1985; Dabrowski, Bukowy-Bieryllo, and Zietkiewicz, 2015). Although as far back as 1983, this phenomenon was known to be a crucial process, "It should be noted that readthrough is not merely a translation error; several biologically important proteins are synthesized as a result of read-through." (Ryoji, Hsia, and Kaji, 1983).

We do not understand many of the mechanics of translational readthrough. Some studies have investigated the processes behind it but have not yet come to any conclusive or complete theories (Belinky et al., 2021). Chemical-induced translational readthrough has been studied since at least the 1960s for both its antibiotic and genetic disease mitigation (Davies, Gilbert, and Gorini, 1964; Du et al., 2009). In one recent study, it was discovered that excess carbon in growth media can substantially increase readthrough levels of all three canonical stop codons (Zhang et al., 2020a), although the mechanics behind such processes are unclear. Furthermore, "the level of readthrough fluctuates extensively among single cells that are genetically identical and grown under the same stress condition. Cells with different levels of readthrough vary in phenotypes, and individual cells with high readthrough recover better from the acid stress." (Zhang et al., 2020a). The efficiency (or occurrence thereof) of translational readthrough depends on a variety of factors, including selection of the termination codon (as discussed in the previous section), the context surrounding the mRNA sequence, and the presence of stimulating compounds. Additionally, there are 2 classes of release factors (I and II) which facilitate termination. "In bacteria, the class I RFs, RF1 and RF2, recognize the UAG/UAA and UGA/UAA stop codons, respectively, by the recognition loop...", however "The translation termination mechanism in archaea is considered to be similar to that in eukaryotes." (Kobayashi et al., 2012). In both bacteria and eukaryotes, it has been found that the base that exerts the strongest influence on RT efficiency is that that immediately follows the end of the stop codon. The hypothesis to which this had led proposes that the actual translation termination signal consists of a tetranucleotide sequence and not only the stop codon itself (Dabrowski, Bukowy-Bieryllo, and Zietkiewicz, 2015). Thus, termination of translation is one of the most complex stages in protein biosynthesis and continues to require further study.

1.3.1.4 Post-Transcriptional and Post-Translational Modification

Alternative expression of CDS genes in eukaryote genomes is a well-known procedure which involves the complex process of post-transcriptional splicing of the different introns and exons which make up a gene. Not only has it been widely recognised as a source of proteome diversity in eukaryotic species (Griffith and Marra, 2007), but also it has often also been used as a distinguishing factor between eukaryotes and prokaryotes (Darnell, 1978). While prokaryotic genes are rarely assumed

to undergo post-transcriptional modification, its consequences are now being studied. These studies have investigated the influence and impact of post-transcriptional modification in bacteria on important elements such as virulence and adaptation to fluctuations in nutrient availability via metabolic changes (Pisithkul, Patel, and Amador-Noguez, 2015; Macek et al., 2019; Macek et al., 2019).

As with post-transcription, post-translational modification of peptide sequences is used by both prokaryotes and eukaryotes to perform often distinct processes. For example, many pathogenic bacteria target the post-translation of eukaryotic genes and have been shown to be an important tool during infection (Ribet and Cossart, 2010). While the level of enzyme modification differs greatly among bacterial species, the extent of the modified proteome has strongly been linked to environmental conditions (Macek et al., 2019). Additionally, "protein PTMs [post-translational-modifications] were shown to be widespread in bacteria and involved in virtually every major physiological process in the bacterial cell" (Macek et al., 2019). However, as the modification machinery differs greatly among bacterial species, it is therefore difficult to attribute specific general functions to every type of modification. Even further complications arise from the fact that many post-translational modifications are observed to be reversible.

As noted previously, the notion that one ORF equates to one protein sequence is clearly no longer complete. The competitive and ever-changing natural environments inhabited by prokaryotes require them to be efficient at adapting their metabolism to inherent fluctuations in nutrient availability and other environmental variables. In addition to necessitating other mechanisms, responsive post-transcriptional and post-translational regulatory measures seem to be key to their ability to sustain replication in such environments (Michard and Doublet, 2015; Macek et al., 2019).

As with the other stages of protein synthesis discussed in this section, the impact of these complexities, some of which we still do not yet fully understand, are still causes of genome annotation difficulty and inaccuracy, often requiring multiple annotations for a single gene to be complete (Meydan et al., 2019). Therefore, it is unlikely that we can overcome this with computational approaches alone.

1.3.1.5 Overlapping and Short Genes

The presence and importance of overlapping ORFs, first found in bacteriophages (Barrell et al., 1978) in the late 1970's, but now found across most forms of life (Huvet and Stumpf, 2014; Kumar, 2009), are indisputable. However, overlapping genes are often mispredicted and also completely missed in annotation. Unfortunately, studies into the annotation quality of overlapping genes often use homology to orthologs to decipher the validity of a predicted overlapping gene (Pallejà, Harrington, and Bork, 2008). However, this relies on the previously annotated genes to be correct

themselves and also ignores the possibility of slight length variations between orthologous genes. Interestingly, most overlaps are observed to be between two genes on the same strand. In addition to this, the phases or frames of the strand have been shown to be biased for certain types of genes and frequencies of initiation and termination codons in the two phases (Sabath, Graur, and Landan, 2008). Unsurprisingly, many sets of genes that overlap with each other are also part of the same operons, supporting the hypothesis of mutual evolution and functional dependence through co-expression (Huvet and Stumpf, 2014). Previous studies have observed that the number of genes that overlap in bacterial genomes is positively correlated with the number of genes, implying that gene overlap may be mainly the result of accidental or random "trespassing" of one gene into another (Fukuda, Nakayama, and Tomita, 2003). Another study concluded that "... overlapping genes are a consistent feature (approximately one-third of all genes) across all microbial genomes sequenced to date, have homologs in more microbes than do non-overlapping genes, and are therefore likely more conserved" (Johnson and Chisholm, 2004).

The belief that one ORF encodes one protein at a minimum length is at odds with the serendipitous discoveries of translated Short-ORFs (Orr et al., 2020) (note that, while defined as putative CDSs, the literature names these as Short-ORFs rather than Short-CDSs). The initial lack of research into Short-ORFs can be in part attributed to the fact that while we are learning more about their importance and presence within prokaryotic proteomes. The long standing perception surrounding minimum CDS gene length and the inherent bias these assumptions perpetuate, prevent many techniques from accurately identifying them. Unfortunately, another problem that plagues the Short-ORF study is the lack of agreement on the length below which to define an ORF as short (often between 100-400 nt) (Goli and Nair, 2012; Su et al., 2013; Andrews and Rothnagel, 2014; Storz, Wolf, and Ramamurthi, 2014; Duval and Cossart, 2017; Baek et al., 2017; Orr et al., 2020). Many studies have interpreted it differently, not only between eukaryotes and prokaryotes, but also within each group. Here I take the minimal definition often found in the literature: a CDS gene no less than 100 nt. The traditional assumptions regarding protein-coding genes are becoming an obstacle for future discoveries of Short-ORFs and biological knowledge. Recently, a number of studies have observed compelling evidence of translated Short-ORFs that have begun to expand our understanding of their true impact and importance in a number of crucial biological processes (Baek et al., 2017; Andrews and Rothnagel, 2014). However, these discoveries often require high-quality RNA expression evidence of the short peptides, which makes their identification with computational-only methods challenging, if not impossible. While it is accepted that Short-ORFs are not only more common than previously thought in prokaryotic genomes, but also have important roles, many genomic annotations are still undertaken with software which contain hard-coded limitations to minimum ORF length and therefore Short-ORFs are often left out of analysis entirely. Additionally, Short-ORFs encoded upstream and downstream of annotated

(normal length) CDS genes, from alternative start sites nested within these CDS genes, and from RNAs previously considered non-coding have been found (Goli and Nair, 2012). It is becoming clear that the initial assumptions surrounding minimal CDS lengths are incorrect or at least not comprehensive.

1.3.1.6 The Unknowns of CoDing Sequences and the use of RNA-Seq Data in CDS Prediction

Two of the clearest characteristics that distinguish the genomes of prokaryotic (and viral) organisms from that of eukaryotic organisms are the compactness of their genomes and their lack of introns (outside of a small number of niche genes (Edgell, Belfort, and Shub, 2000)). Therefore, it could be inferred that the detection of prokaryotic CDS's should be more straightforward and not require additional *in vivo* experimental data such as the difficult-to-use RNA-Seq data (Wang, Gerstein, and Snyder, 2009) This is reflected in the core toolsets used to identify prokaryotic CDS genes. For example, prediction tools that rely solely on DNA data are by far the most common for the annotation of prokaryotic genomes. Even state-of-the-art prediction tools used by the genomic community such as PROKKA (Seemann, 2014) and the NCBI Prokaryotic Genome Annotation Pipeline (PGAP) (Angiuoli et al., 2008) only use DNA data for annotation. This is mostly because prokaryotic genome annotation is an inherently different problem from eukaryotic genome annotation. As discussed above, since the majority of prokaryotic DNA is protein coding, the challenge is not the determination of whether a certain predicted ORF (or segment of ORF with respect to eukaryotic introns) is coding or not, but instead it is the exact identification of the start and stop codons.

Prokaryote genome annotation is not as straight forward as it may seem at first and there are a number of complex and poorly understood processes that are yet to be fully accounted for in the current collection of annotation techniques. For example, the bacterium *Haemophilus influenzae* has been shown to have the potential to utilise a different set of start codons positioned within the 'same' CDS gene (Dixon et al., 2007). RNA-Seq data could therefore, enable the identification of these alternative transcription start sites. However, there is a high level of difficulty and resource requirements for RNA-seq data which most often hinders its use for prokaryotic genome annotation (Salzberg, 2019). RNA-Seq data is dependent on a number of complex and often difficult to control factors which include: The expression of some mRNA is only carried out during specific conditions such as those from quorum sensing genes (Miller and Bassler, 2001; Wang, Gerstein, and Snyder, 2009), the sheer amount of ribosomal and structural (sometimes up to 80-95% of a sample) RNA in a prokaryotic sample effectively obscures the detection of CDS transcripts (Chu and Corey, 2012), Poly(A) Tails which are often used in eukrayotic mRNA studies are seldom present in prokaryotes (Sarkar, 1997; Régnier and Marujo, 2003), RNA-Seq data does not inherently match the start-stop location of a coding gene but instead

often contains regulatory regions such as the 5′ and 3′ UTRs (Bischler et al., 2015), and lastly, as discussed in Subsection 1.3.1.3 and above, the potential use of multiple alternative start codons and stop codon readthrough (Belinky et al., 2021) by a single gene, adds to the likelihood that RNA-Seq data may not report the 'true' or at least 'all' possible start and stop positions of a coding gene (see Figure 1.3 for an example). The alternative use of different start and stop codons is particularly difficult to identify in prokaryotes, as there is no mature mRNA processing, as in the case in eukaryotic cells. Mature RNA transcripts that have been spliced and processed, can be used to identify the exact start and stop codons, including the positions of introns and exons. As reported in Figure 1.3, in prokaryotes, the mRNA sequence includes the entire sequence starting from the promoter region which itself is not always in-frame with the CDS gene. Further to this, prokaryotic mRNA ends at a transcription termination signal, of which there are at least three: an intrinsic termination by the formation of a RNA hairpin, Rho dependent termination, and Mfd-dependent termination, all of which makes accurately identifying the correct stop location challenging (Roberts, 2019). Therefore, for prokaryotes, RNA-Seq data could result in the introduction of a significant amount of noise into the annotation process.

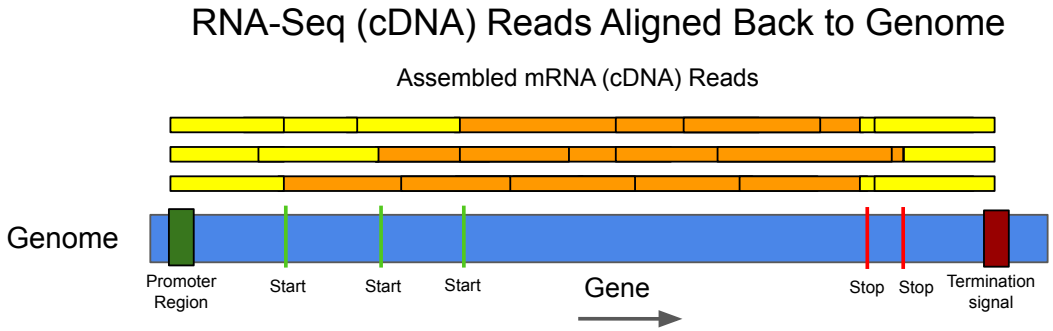


FIGURE 1.3: Diagram presenting the complexity of aligning RNA-Seq (often complementary DNA - cDNA) reads to a prokaryotic genome. The three examples of assembled RNA reads (orange boxes representing assembled transcript reads and yellow reported non-translated RNA segments of mRNA transcript between the promoter region and the transcription termination signal) report three different CDSs from the same gene by aligning to the three different start codons and two stop codons through processes such as alternative start codon usage and stop codon readthrough by a single gene.

1.4 Pangenomes and Gene Sharing Dynamics

The role of pangenomes and how they form vital evolutionary avenues for species survival and evolutionary opportunities is still being explored (McInerney, McNally, and O'Connell, 2017). Formed from sets of differentially conserved genes throughout a species range, pangenomes offer greater insight into a species than any single genome can. Core genes, those found in "most" samples, are more likely to be present in databases and have experimental evidence of function because they are inherently representative of a species and therefore exhibit less divergence (Medini et al., 2005). However, those genes on the peripherals of a pangenome, often known as "accessory" or "dispensable" genes, are often less likely to be experimentally validated and therefore more likely to be misannotated and missing from gene collections (Segerman, 2012). Different strains within the same species have been observed to differ greatly in the size of their pangenomes. This is interesting when considering that an extended or 'global' bacterial pangenome has been known for some time and suggests a core of shared essential genes across the majority of bacterial genomes studied (Lapierre and Gogarten, 2009). There are, however, still many unanswered questions surrounding the mechanisms behind the formation of both species and global pangenomes. The fluidity of gene content within prokaryote genomes has often been thought of as a reserved matter for non-essential genes. Intriguingly, while it may be assumed that the core genes of a pangenome are the set of essential genes of a particular organism, several studies have shown that it is, in fact, the accessory genes that are often essential (McInerney, McNally, and O'Connell, 2017). In addition to this, their role in the CRISPR-Cas system of bacteria and archaea is now being studied (Shah et al., 2019).

The impact of certain mechanisms of pangenome growth and shrinkage has long been studied, with one study suggesting " ... the transferability of genes [through HGT] seems to depend heavily on their functions" (Nakamura et al., 2004). Additionally, it was noted in a study over twenty years ago that the more numerous and complex the interactions of a protein are, the less likely it is to be successfully horizontally transferred (Jain, Rivera, and Lake, 1999). However, as previously discussed, even model organisms have large proportions of uncharacterised genes, both in their core and non-core pangenome (Ghatak et al., 2019). Therefore, any determination of what functions may influence certain genes to be more or less likely to be shared within and between species is not possible for a large proportion of genes. As a result, the role of gene sharing processes such as HGT and Lateral Gene Transfer (LGT) in pangenomes and how they form vital evolutionary avenues for species survival and evolutionary opportunities are still being explored (McInerney, McNally, and O'Connell, 2017; Nagies et al., 2020). With the increase in the number and prominence of metagenomic and pangenomic studies, the importance of complete and high-quality genome annotation will only increase further.

1.5 Genomic Annotation Databases

As with all fields of study, much of what is possible and carried out today, is not only based on foundations built on previous work, but is also susceptible to the limitations and weaknesses of those foundations. In genomics, the genomic knowledge deposited across the vast number of online databases and repositories is clearly representative of this.

1.5.1 The Completeness of Genomic Databases and Annotations

In 2007, there was a consensus that contemporary archival repositories such as GenBank and Ensembl had been and were still serving the needs of the scientific community for a vast open access of genomic annotations (Salzberg, 2007). However, it was acknowledged that there was no process for tracking re-annotations or easily accessible or transparent database of the impacts these improvements or updates had on their biological importance and replicability. As of 2022, databases such as NCBI's RefSeq (Pruitt, Tatusova, and Maglott, 2007) do have ongoing initiatives to reannotate their collection of genomes. However, they are currently using newer versions of the same tools that were originally used to annotate many of the historically deposited genomes (Tatusova et al., 2016). This may make it difficult to identify the shortcomings between the old and new approaches without any external evaluation. Tens of thousands of GenBank and RefSeq prokaryotic genomes are being reannotated every year with 'new' (or 'improved') annotation tools and there is currently no method for research groups to account for the intricate differences between the resulting annotations. A study in 2014 of 32,000 GenBank genomes showed that while the computed 'quality' scores were high, more than 80% of the genomes were in draft status. Additionally, Ensembl Genomes (Howe et al., 2020) has a sub-database specifically for bacteria which uses GenBank records, amongst other databases, as the primary source of sequence and annotation (Kersey et al., 2010). This is just one example of a potential error or bias being passed from one database to another with little chance of identifying the origin of the said error (see Figure 1.4).

```
Firstly, it is important to remember that Ensembl does not make annotation by
itself. We are dependant on the submissions that different annotators provide
to ENA. The best course of action at this point is to contact the original
submitter and suggest a revision of their annotation.
```

FIGURE 1.4: Correspondence from Ensembl in response to a request asking for clarification on an error found in one of the GFF annotation files for a number of genomes in Release 46 of Ensembl Bacteria. Even large consortia such as Ensembl Bacteria struggle to keep track of where their data comes from and how to interpret it. Different annotation methods can be used, but they are not clearly reported in the final data.

Completeness as a concept can be defined as something which contains all necessary parts or is lacking in nothing. However, the definition of 'complete' in regard to genome annotation in the literature is itself lacking. Studies such as Doxey *et al* (Lobb et al., 2020) use the CDS predictions from PROKKA (which uses Prodigal (Hyatt et al., 2010) for CDS prediction) in their analysis of annotation 'completeness' across the bacterial tree of life. This method assumes that the annotation from PROKKA, notably a state-of-the-art and competent tool, is itself 'complete'. Several studies have attempted to identify the level of completeness and quality of genomic annotations by investigating the similarity of 'intergenic regions' compared to known genes in large genomic databases (Wood et al., 2012). Representative gene families in these databases have also been harnessed to quantify and further complete genome annotations (Dunne and Kelly, 2017). However, many types and families of genes continue to be absent or underrepresented in public databases (Warren et al., 2010; Huvet and Stumpf, 2014), especially those with unknown function and Short/Small-ORFs (Short ORFs) (Storz, Wolf, and Ramamurthi, 2014; Duval and Cossart, 2017; Su et al., 2013). Therefore, unfortunately, this approach can only account for the missed or pseudogenes which share similarity with sequences already identified and deposited by other groups. Nevertheless, the size and accessibility of contemporary sequence databases such as those held in UniProt (Bateman et al., 2020) and NCBI (Haft et al., 2018) may have inadvertently led to the often used assumption that if a large portion of the CDS predictions from a studied genome align to previously reported sequences, the annotation can be defined as 'complete' (Stothard and Wishart, 2006). This use of contemporary databases to confer completeness of annotation continues to be used in more recent studies (Richardson and Watson, 2013).

It is likely unfair to blame the inadequacies of genome annotation on prediction methods alone. The numerous complexities of genomics, many of which we continue to discover (Belinky et al., 2021), have made it almost surprising how competent genome annotation has been over the last three decades (Dimonaco et al., 2021). Nevertheless, there has been little discernible progress in novel genome annotation techniques as most still rely on the same type of algorithmic design (although improved upon greatly) that has been used since the start of genome annotation (Tatusova et al., 2016). Either through lack of understanding of the problem or the complexity of the technical and time constraints, the use of a single tool without external validation and not consensus, augmentation, or supplementation of predictions, is too often the avenue selected for genome annotation. There is, therefore, a need to reevaluate both how genome annotation is undertaken and how a genome annotation is defined as 'complete'.

1.5.2 Bias and Error in Genomic Databases

Research bias, taxonomic distribution, genome size and tractability are just some of the factors that have been shown to influence annotation completeness (Lobb et al., 2020). However, studies into determining the level of bias held in the genomic knowledge contained within the public databases are incomplete and are yet to offer any realistic solutions in the current scientific climate.

With the accuracy of genome annotation methods still contested, it is important that the implications of their potential failings are not only understood, but are also mitigated against. Additionally, bias has been attributed to the number of publications on specific organisms and the completeness of their genome annotations (Lobb et al., 2020). However, many studies still rely on such databases (and therefore inherent biases) to confirm the accuracy of their methods (see Figure 1.5. Genome annotation methods that use information from existing sequence databases to build models for genome annotation are in turn ill equipped to identify genes belonging to gene families underrepresented in the original databases. Sydney Brenner noted in 1999 that incomplete and incorrect annotations in existing databases would lead to a continuation of future incomplete annotation, as many modern prediction methods rely heavily on previous annotations to create models from which to predict new ORFs.

> "The procedure need cycle only a few times without corrections before the resources that made computational function determination possible 'the annotation databases' are so polluted as to be almost useless." (Brenner, 1999).

The overreliance on automated genome annotation tools has resulted in many gene families being under-represented in genomic databases (Warren et al., 2010) along with annotation error or omission (Schnoes et al., 2009; Wood et al., 2012), which inevitably leads to further error and bias in annotation pipelines reliant on inference from known genes. Furthermore, studies have shown that many existing genome annotation tools systemically miss CDS genes that are fewer than 33 amino acids, causing many gene families to be absent or underrepresented in public databases (Warren et al., 2010). Shortcomings in existing model organisms' genome annotations are inherited by the prediction tools which use them as ground truth. This problem is compounded further by the fact that many genome annotation tools use features of genes extracted from model organisms' databases to define their rules for gene prediction, creating a problem which perpetuates previous annotation bias. During the last three decades, specific groups of tools such as the GeneMark family (Besemer and Borodovsky, 2005) have developed a number of different approaches for predicting genes in prokaryotes, eukaryotes, viruses and metagenomes, whose annotations have become the bedrock of much of the genomic knowledge held in

public repositories such as RefSeq and GenBank (Tatusova et al., 2016; Haft et al., 2018). As noted by S. Salzberg in 2019,

> *"...genome annotation still uses very nearly the same technology that we have used for the past two decades... but errors in annotation are just as prevalent as they were in the past, if not more so." (Salzberg, 2019)*

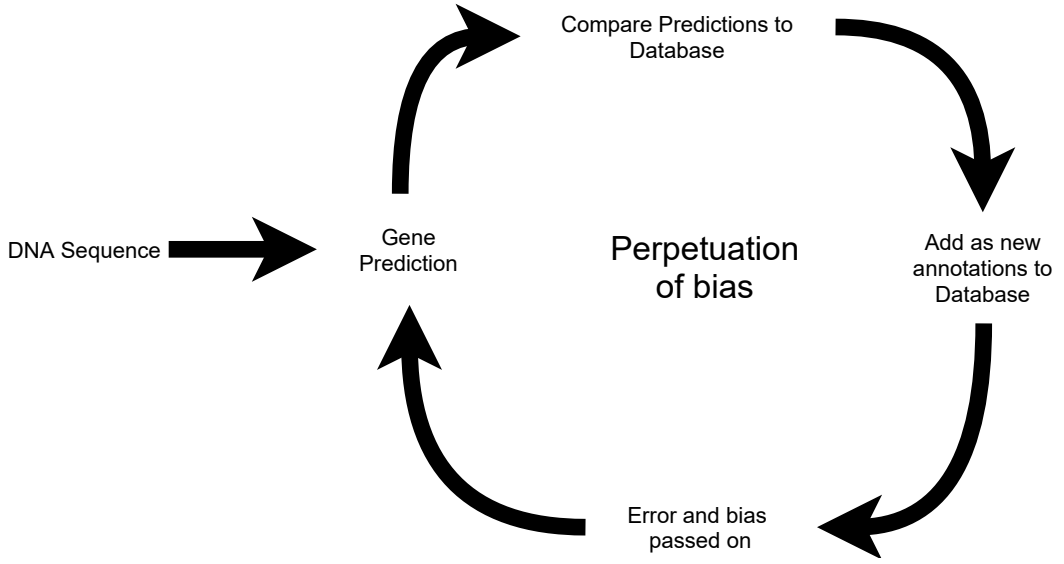


FIGURE 1.5: Circular annotation: The cycle needs only continue for a limited number of revolutions before the initial bias and error limits the scope of future discoveries.

Genes identified in fragmented genomes are also much more likely to be incorrectly annotated with Pfam, KEGG and homology information. A previous study in 2012 on the quality of genomes deposited in Genbank revealed that "... fragmented Open Reading Frames (ORFs) comprised 80% of the predicted ORFs in some genomes and that increased fragmentation correlated with decreased genome assembly quality" (Klassen and Currie, 2012). Fragmentation of genomes deposited in databases continues today with the NCBI genome repository database, which is aggregated from a number of separate databases such as GenBank and RefSeq (Haft et al., 2017), reports only 16,638 genomes as 'complete' out of a total number of 223,590 bacterial genomes, as of January 2020.

More genomes are being sequenced than ever before. The vast majority of genes predicted in these genomes will never be experimentally characterised. It is therefore of paramount importance that we understand the limitations of gene predictors as our reliance on them is likely to increase. Novel ORF finding tools are often evaluated using genomic data that are more than 20 years old and likely have an incomplete annotation, generated by tools that have since been superseded. The original NCBI prokaryotic genome annotation pipeline (PGAP) (Angiuoli et al., 2008),

made use of GeneMarkS (Besemer, Lomsadze, and Borodovsky, 2001) and GLIMMER(I) (Delcher et al., 1999) (both developed 20 years ago) and has now been superseded by GeneMarkS-2 ("GeneMarkS-2: Raising Standards of Accuracy in Gene Recognition") and GLIMMER3 (Delcher et al., 2007) respectively. It is unclear what proportion of historic annotations held in existing databases have been updated by the newer methods. However, it is important to recognise that many tools developed since 2001 (the publication date of GeneMarkS) are likely to have been trained on the data produced by the GeneMark-family and thus possibly were developed with the same biases that were introduced in 2001. This self-fulfilling prophecy can be seen in the results of the PGAP-3.1 publication compared to a number of GenBank (Benson et al., 2012) annotations. The new version of PGAP-3.1 (Tatusova et al., 2016) was evaluated against existing GenBank annotations. They found that prediction mismatches at the 5′ end of ORFs were considerably higher than for the 3′ end. One reason given was that "Sources of differences could be related to errors either in automatic or in GenBank annotation" (Tatusova et al., 2016). The precision of PGAP-3.1 (which is not clearly defined) when compared to the corresponding GenBank annotations is 89.9%. The above study explained this in the following way: "arguably, a difference of more than 2% may indicate some issues with the GenBank record (such as absence of continuous curation; e.g. the last updates of the GenBank annotation records of *N. meningitidis MC58* and *B. subtilis* were made in 2005 and 2009, respectively)." (Tatusova et al., 2016). This explanation does not report the complete picture and instead raises more questions than it answers.

> *"Many of the challenges of de novo gene prediction that have been observed over the years remain challenges today. Even the best prediction programs tend to split and fuse genes, and they have difficulty accurately predicting stop codons and especially start codons. They only predict a single isoform at each locus..." (Brent, 2005).*

Many of the rules such as standard CDS length, genetic codes and codon usage are inferred from previously predicted CDSs which have been deposited into a number of database resources such as UniProt (UniProt Consortium, 2019) and RefSeq (Haft et al., 2017). Unfortunately, many of these reference sequences have likely been identified with the same set of rigid rules. The potential for annotation errors to be deposited in varying databases is well established, and this is unlikely to be resolved in the near future (Bork and Bairoch, 1996; Karp, 1998) without significant coordination between repositories (Klimke et al., 2011). This has led to the likely bias in gene predictors, especially in those which rely on previously identified gene sequences to build classification models. An important issue yet to be addressed is whether important tools such as NCBI's PGAP are biased towards predicting what we already know and not what we want to know. How such error and bias can affect genomic knowledge held in public databases is still largely unknown (Devos and Valencia, 2001; Furnham, Beer, and Thornton, 2012).

There is no easy solution to correct for biases and incomplete assumptions in our genomic knowledge. One may assume that human curation is a possible solution. However, it is not feasible, especially with the advent of non-model and metagenomic studies. Human curation has often been focused on specific model organisms (Braun et al., 2005) or genetypes of direct human importance, such as cancer genes (Tate et al., 2019). Additionally, there is still much debate on not only how manual curation should be undertaken, but also how effective it can be (Odell et al., 2017; Salzberg, 2019; Ritter et al., 2019) These problems are long-term and the limitations they impose must be dealt with, but most importantly, they must be understood and accepted.

1.5.3 How we Define Genomic Features

An often unaddressed problem in biology, and specifically computational biology and genomics, is that the definitions and how we record some of the most fundamental genomic elements, such as the previously discussed Open Reading Frames and CDS genes, are continually changing. Often described as one of the "Deadly Sins of Bioinformatics", the development of ever-increasing numbers of file formats is still a major problem in bioinformatics. The uncertainty that this perpetuates is often comically derided; however, it can have some serious consequences (see Figure 1.6).

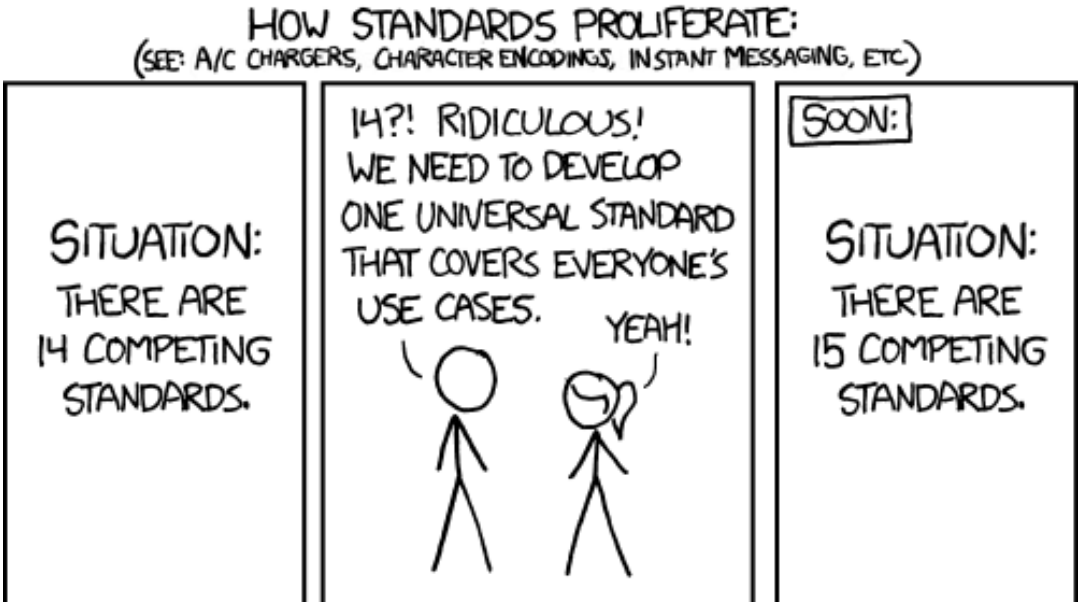


FIGURE 1.6: This xkdc webcomic (xkdc.com) depicts a very common problem in informatics: the continuously changing and growing number of standards.

There have been a number of coordinated efforts to bring unity to some of the most used formats in bioinformatics. One of the most important and most used of these is The Generic Feature Format, commonly named GFF3 and is currently in version 3. The GFF3 specification, described at https://github.com/The-Sequence-Ontology/

Specifications/blob/master/gff3.md, has been devised by The Sequence Ontology (Eilbeck et al., 2005). There is a clear logic to this, as the inherent constraints of an ontology (restricted and directional vocabulary) can be used to keep the definitions within GFF3 universal. However, at its current version of v1.26, GFF3 and its creators allude to some additional problems they were faced with.

> *"Although there are many richer ways of representing genomic features via XML and in relational database schemas, the stubborn persistence of a variety of ad-hoc tab-delimited flat file formats declares the bioinformatics community's need for a simple format that can be modified with a text editor and processed with shell tools like grep. The GFF format, although widely used, has fragmented into multiple incompatible dialects." - The Sequence Ontology (Eilbeck et al., 2005).*

While the GFF format is by no means the only genomic feature format out there, it is not only the most widely used, but it also has relatively good interoperability with other formats such as GTF (https://mblab.wustl.edu/GTF22.html) and GenBank (https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html). Additionally, GFF3 addresses many of the commonly utilised extensions which have historically been made to GFF, while preserving backward compatibility with previous formats. However, even with all the constraints and improvements, there are still occasions when the scope of the specification falls short. Biology is neither linear nor binary, despite previous and ongoing attempts to make it so. For example, intrinsic aspects of many prokaryotes, such as the circular nature of their genomes, are very difficult if not impossible to plot along a linear configuration, such as those held in the abovementioned formats. As seen in Figure 1.7, the AAK43339 gene of *Saccharolobus solfataricus* p2 begins at the 5' end of the linearly displayed genome and ends at its beginning. This is clearly due to the unfortunate coincidence that the gene spans the points at which the circular genome is cut in its linear presentation.

Annotating non-linear elements on a linear structure is inherently difficult. Therefore, it is possibly unsurprising that there are several ways this phenomenon can be presented, as can be seen in Figure 1.8, which alone shows three alternative but canonically agreed methods to annotate genome-wrapping genes from Ensembl. Although unrelated, an additional complication identified here is the name change of *Sulfolobus solfataricus* to *Saccharolobus solfataricus* (Sakai and Kurosawa, 2018), however, this is currently not reflected in the Ensembl bacteria database (release 46) (Howe et al., 2020). Although these changes and disagreements are typical of the constant flux that characterises the fields of genomics and bioinformatics, there seems to be little agreement on what the solution is.

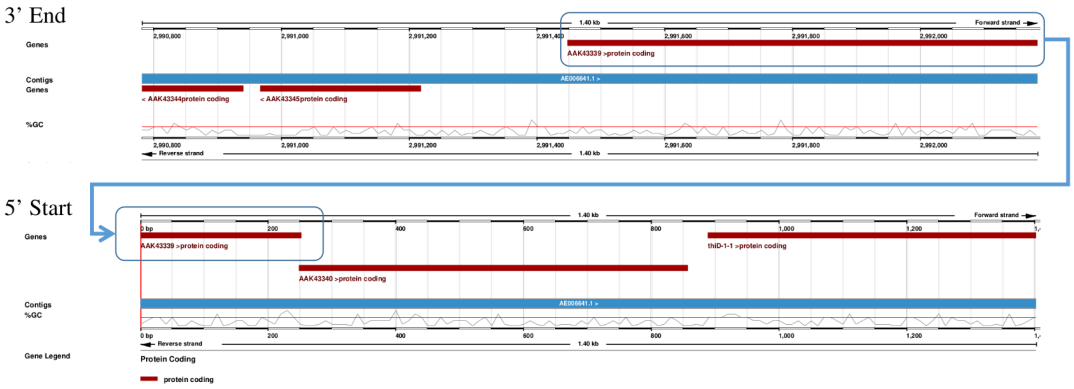


FIGURE 1.7: This figure presents the start of the AAK43339 gene, which is located at the end of the *Sulfolobus solfataricus* p2 genome as reported in Release 46 of Ensembl Bacteria. As the genome is represented linearly in the GFF format, the gene effectively 'falls-off' the end of the genome and is therefore difficult to interpret on a linear display.

```
In the CDS FASTA files, the coordinates are listed as the start and the
end with respect to the coordinate system. i.e the start position is
2991448 and the end position is 252.

However in the GFF3 file, the coordinates are listed with the end equal to
the 'end of the coordinate system + 252 = 2,992,245 + 252 = 2,992,497

Furthermore, in the GTF file, the coordinates are listed with the start
equal to the 'start of the coordinate system - 797 (the length of the
transcript before the 0 coordinate):
```

FIGURE 1.8: An email response by Ensembl Bacteria in reply to my query regarding the gene AAK43339 of the *Sulfolobus solfataricus* p2 genome. The 'genome-wrapping' gene coordinates are reported in a different way for each of the 3 available formats: CDS, GFF3 and GTF.

1.6 The Future of Prokaryote Genome Annotation

A transition in bioinformatics is currently well underway. Advancements in the field of computing and especially machine learning have led to a great many advancements such as the protein structure predictor Alpha Fold (Jumper et al., 2021) and have been highlighted a number of times as the future of bioinformatics (Tang et al., 2019; Li et al., 2020a). However, back in the field of genome annotation, while a number of contemporary techniques are now harnessing similar recent advances, limitations inherent to machine learning are still present.

From the very beginning, various genome annotation methods began by constructing models formulated from organism specific parameters such as genetic code use, GC content and average gene length of model organisms (Besemer and Borodovsky, 1999; Stanke and Morgenstern, 2005). With opinions shifting on the use and importance of model organisms and the increased prospecting of non-model species through methods such as metagenomics, these models became less and less useful

(Levy and Currie, 2015; Russell et al., 2017; Hunter, 2008b). This provided clear justification of the need for species agnostic prediction methodologies which do not rely on exact examples of previously identified sequences. Furthermore, the volume of putative prokaryotic CDSs has led to the supposition that machine learning approaches such as neural networks can be applied. One such example, Balrog (Sommer and Salzberg, 2021) predicts gene CDSs by training from an array of non-hypothetical protein coding sequences from thousands of bacterial prokaryote genomes and aims to provide gene prediction across diverse species. However, this approach is heuristic and "Experimentally-validated start sites are not available for the vast majority of bacterial genes"(Sommer and Salzberg, 2021). Machine learning models are known to be ineffective at making predictions for classes (e.g. genes) whose training data exhibit high levels of bias, are underrepresented for specific groups (e.g., gene families) and groups for which they have not been trained (Schafer and Graham, 2002). In addition to this, prokaryotic gene families are chronically undersampled (Warren et al., 2010).

As more and more genomes are sequenced, we will inevitably discover aspects of life that we currently do not account for in our computational methods and parameters. Examples of this are varied, but in one study of 61 *E. coli* strains, it was found that there was a maximum genome size difference of around one million nucleotides, or approximately 20%, and that the largest *E. coli* genome had 1,158 more genes than the smallest (Lukjancenko, Wassenaar, and Ussery, 2010). Therefore, methods which use parameters derived from one individual from a species would therefore struggle with the vastness of divergence between strains, potentially making species-specific parameters not representative of the whole species-range. Although this should not be unexpected as "the transfer of genetic variation from one population to another (gene flow) can cause rapid and large-scale additions and rearrangements of genomic regions" (Lukjancenko, Wassenaar, and Ussery, 2010), it is still often not taken into consideration. New understanding surrounding the complexity of prokaryotic genomic processes such as the aforementioned fluidity of alternative gene expression, genetic transfer between diverse species, and the concept that one ORF may not always code for exactly one protein further emphasizes how current methodologies are no longer adequate. Throughout the last three decades, a number of studies have been conducted to investigate the strategies employed to computationally annotate prokaryotic genomes. Although there has been progress in identifying a number of limitations and apparent biases (see Chapter 2), there has been little progress in addressing them in historic annotation or through novel method development (Roberts, 2004; Frishman, 2007; Salzberg, 2019). Furthermore, the current Coronavirus pandemic has reaffirmed the paramount importance of fast and high-quality genome annotation. The incredible speed with which the SARS-2 genome was sequenced, assembled, annotated, and most importantly presented to the world, showcased just how far we have come in the last two decades alone (Lu et al., 2020). A juxposition of this can be found when comparing the first SARS

outbreak in 2003 to that of SARS-CoV-2 in 2019/2020. It took nearly 3 months for its identification and subsequent initial investigation after first being misdiagnosed as a chlamydia outbreak (Zhong and Zeng, 2006). Therefore, it is crucial that we understand the limitations of gene predictors, as our reliance on automated genome annotation is only likely to increase (Brenner, 1999).

In computational genome annotation, genes are often seen as independent and isolated elements, however, their expression is often controlled by a number of factors, many of which are still unknown. There are a number of examples in bacteria and archaea. One of these is the use of small regulatory noncoding RNAs (ncRNAs) to control the expression of specific genetic processes. These have been found within both the intergenic and protein-coding regions (Dar and Sorek, 2018). The presence or absence of these expression elements could be key in helping us determine not only where crypitc and other undetected genes are, but whether predicted ORFs are likely to be expressed. However, methods such as these rely heavily on coordination between biologists and bioinformaticians.

> *As outlined by Michael R Brent in 2005 (Brent, 2005). "It is abundantly clear that the methods we have been using to identify ORFs for most of the last 10 years are inadequate for finishing the job. [...] I argue that we cannot rely on any of the following to get us through the home stretch of ORF identification: obtaining EST or mRNA sequences from randomly selected cDNA clones, aligning expressed sequences to loci other than those from which they were transcribed, e.g., to the loci of gene family members or orthologs in other species, sequencing more genomes, annotating manually by using human curators. All of these things are valuable, but none of them is likely to get us to a new, higher plateau in the quest for a complete ORF at each protein coding locus."*

Big data, once a buzzword for the future of informatics, almost seems to have been designed almost exclusively for genomics. In the previous decade, there were a number of papers and reports on the future of genomics which postulated heavily on the impact that big data and therefore 'big curation' would have (Howe et al., 2008; Pennisi, 2008; Lathe et al., 2008). However, curation increasingly lags behind data generation in all aspects, including development, funding, and most importantly, recognition (Howe et al., 2008). Unfortunately, there is resistance from almost all stakeholders as all proposed solutions are currently untenable or unpalatable to those involved (Pennisi, 2008).

> *"Lastly, but importantly, the growing number of genome databases, analysis tools, and other resources available on the web has made it daunting for researchers to use these resources effectively" (Lathe et al., 2008).*

1.7 Aims of Thesis

The review of the literature presented in this chapter has identified several avenues that require further study. Fundamentally, it has become clear that first a systematic review of not only contemporary genome annotation techniques in use today, but also those which have been used in the past, is required. This will be presented in Chapter 2 (Dimonaco et al., 2021).

A review alone, even one that identifies the weaknesses and shortcomings of genome annotation, will not take us to the next stage. Once the limitations which exist in annotation strategies have been identified, they must then begin to be addressed. Therefore, in Chapters 3 and 4, I will endeavor to target specific limitations in genome annotation techniques and evaluate not only whether annotation can be supplemented and improved, but also the impact that the additional annotations may have on future studies.

The final characterisation of an organism can only be as good as the completeness and quality of its genome assembly. Therefore, in Chapter 5, to overcome many of the limitations inherent to genome assembly, I propose an assembly-free approach to functionally profile a metagenome with machine learning.

Lastly, due to the current COVID-19 pandemic, Chapter 6 presents the application of several annotation techniques, influenced by the work of the previous chapters, to identify potential mutational hotspots across SARS-like genomes (Dimonaco, Salavati, and Shih, 2021).

A breakdown of the results chapters is listed below:

- Chapter 2: A systematic review of prokaryotic genome annotation techniques presented with suggestions and mechanisms for improvement.
- Chapter 3: Development of a novel supplementary annotation method which targets specific limitations in genome annotation identified in the previous chapter.
- Chapter 4: An extension to the work of the previous chapter to identify potential pseudogenised and alternative genetic code using genes across diverse prokaryotic species.
- Chapter 5: A proposed novel machine learning method for assembly-free coding frame profiling of [meta]genomic reads.
- Chapter 6: The knowledge and experience gained throughout this thesis was used to perform a hybrid annotation of SARS-like genomes to identify possible mutation hotspots in different sets of host-associated coronavirus genomes.

Chapter 2

Prokaryotic gene prediction tool annotations are highly dependent on the organism of study

2.1 Chapter Summary

The biases in CoDing Sequence (CDS) prediction tools, which have been based on historic genomic annotations from model organisms, impact our understanding of novel genomes and metagenomes. This hinders the discovery of new genomic information as it results in predictions being biased towards existing knowledge. To date, users have lacked a systematic and replicable approach to identify the strengths and weaknesses of any ORF prediction tool and allow them to choose the right tool for their analysis.

This chapter presents an evaluation framework (ORForise) based on a comprehensive set of 12 primary and 60 secondary metrics that facilitate the assessment of the performance of ORF prediction tools. This makes it possible to identify which performs better for specific use-cases. We use this to assess 15 *ab initio* and model-based tools representing those most widely used (historically and currently) to generate the knowledge in genomic databases. It is found that the performance of any tool is dependent on the genome being analysed, and no individual tool ranked as the most accurate across all genomes or metrics analysed. Even the top-ranked tools produced conflicting gene collections which could not be resolved by aggregation. The ORForise evaluation framework provides users with a replicable, data-led approach to make informed tool choices for novel genome annotations and for refining historical annotations.

This work is now published in Bioinformatics:
https://doi.org/10.1093/bioinformatics/btab827

Software Availability: https://github.com/NickJD/ORForise

2.2 Introduction

The last two decades has borne witness to great advancements in genomics, due in great part to the rapid and continued development of sequencing and and assembly methods. However, as discussed in Background Section Sequencing and Assembly, this has not all been positive or without obstacle.

The prediction of protein-coding genes, specifically their corresponding CoDing Sequence (CDS) in prokaryote genomes has often been seen as an established routine. This is in part due to a number of assumptions and features such as the high density (protein-coding genes contribute ∼80-90% of prokaryote DNA) and the lack of introns (Lobb et al., 2020; Salzberg, 2019). However, this process involves the complex identification of a number of specific elements such as: promoter regions (Browning and Busby, 2004), the Shine–Dalgarno (Dalgarno and Shine, 1973) ribosomal binding site, and operons (Dandekar et al., 1998), which all contribute to identifying gene position and order. Additionally, the role of horizontal gene transfer (HGT) (Jain, Rivera, and Lake, 1999) and pangenomes further complicates an already difficult process and likely contributes to errors and a lack of data held in public databases (Devos and Valencia, 2001; Furnham, Beer, and Thornton, 2012). Research bias, taxonomic distribution, genome size and experimental tractability are among the factors which have been shown to influence annotation completeness (Lobb et al., 2020). However, studies into determining the level of bias in the genomic knowledge contained within the public databases are nuanced and do not yet offer any definitive solutions. Finally, our ability to validate and characterise the functions of regions of DNA (which has been generally reserved for model organisms and core genes (Russell et al., 2017)) is being outstripped by the rate of genomic and metagenomic sequence data generation from non-model organisms and non-core gene DNA sequences.

Before the turn of the century, it was understood that a great deal of work was still needed to address these issues. Studies had shown that many existing CDS prediction tools systematically failed to identify or accurately report genes whose features lay outside a rigid set of rules, such as non-standard genetic codes, those which overlap other genes or those below a specified length (Guigo, 1997; Burge and Karlinb, 1998). Since then, a systematic overview of 1,474 prokaryotic genome annotations in GenBank concluded "the cause of the high rates of missed genes is less clear, largely due to a lack of information about the annotation methods used."(Wood et al., 2012). Interestingly, while the majority of missed genes reported were under 300 nt, the annotation tools which performed the incomplete annotations were developed to report CDSs at a minimum length of 110 nt. While there has been much work to address the problem of incomplete annotation, many gene types continue to be absent or underrepresented in public databases (Warren et al., 2010; Huvet

and Stumpf, 2014), such as Short/Small-ORFs (Short ORFs) (Storz, Wolf, and Ramamurthi, 2014; Duval and Cossart, 2017; Su et al., 2013). This means that CDS prediction methodologies that use information from existing sequences are in turn ill-equipped to identify genes belonging to these underrepresented/missing gene types. It is therefore of paramount importance that we understand the limits of current CDS predictors as our reliance on automated genome annotation of novel genomes continues to increase (Brenner, 1999). Measures to compare both novel and contemporary CDS prediction tools are not well established or universally employed and novel tool descriptions tend to focus on algorithmic improvements rather than carrying out a systematic assessment of where the strengths or weaknesses in their approaches lie. This prevents researchers from gaining meaningful insight into the specific features of genes which led to them being missed or partially detected, resulting in a lost opportunity to improve our understanding of prokaryote genome content.

Genome annotation is challenging and is not a single step process. CDS prediction, often the first step, is fast, with little user input, but may require augmentation by different methods to supplement the initial predictions. One example is a tool such as smORFer (Bartholomäus et al., 2021) that specialises in finding short ORFs through the use of RNA-seq which can detect transcription events under certain environmental conditions. Further examples use sequence conservation scores and homology searches that can use existing database knowledge (Dunne and Kelly, 2017; ÓhÉigeartaigh et al., 2014; Badger and Olsen, 1999). Furthermore, pipelines are constructed (such as PROKKA (Seemann, 2014) and NCBI's PGAP (Tatusova et al., 2016)) to automate these further rounds of annotation. However, the underlying CDS prediction tools are still core components of these pipelines and are still widely used as standalone tools.

Previous studies which have evaluated prokaryotic CDS predictors generally only compared a small number of tools, focusing on algorithm design, and did not go into depth when reporting prediction accuracy with few other informative metrics used (Al-Turaiki et al., 2011; Mathé et al., 2002). Two more recent studies, BEACON (Kalkatawi, Alam, and Bajic, 2015) and AssessORF (Korandla et al., 2020), considered a small range of metrics including genes "denoted as identical, similar, unique with overlap or unique without overlap" to either a reference annotation or from the output of 3 pipelines (PGAP, AAMG, RAST). Unfortunately, the types of genes missed were also not investigated further, leading to a lack of understanding of not only why and how they were missed, but also the impact on our biological understanding of the genome as a whole.

Many prediction methods used today are iterations of original concepts and thus are as in flux as the genomic databases themselves. Future development of CDS prediction techniques are now harnessing the recent advances in machine learning

and other computational methods. While previous methods involve the construction of models built from organism specific parameters such as codon usage, GC content, complex motifs and average CDS length (Besemer and Borodovsky, 1999; Stanke and Morgenstern, 2005), opinions are shifting on the use and importance of model organisms (Levy and Currie, 2015; Russell et al., 2017; Hunter, 2008b). The increased prospecting of non-model species through methods such as metagenomics, provides clear justification of the need for species agnostic prediction methodologies which do not rely on previous genome or gene examples. The volume of prokaryotic protein-coding gene sequences have enabled advanced machine learning approaches such as neural networks to predict CDSs that share common characteristics with a selection of previously annotated genes. One such example, Balrog (Sommer and Salzberg, 2021) predicts protein-coding genes by training from an array of non-hypothetical protein-coding sequences from thousands of bacterial prokaryote genomes and aims to provide gene prediction across diverse species. This approach is heuristic and "Experimentally-validated start sites are not available for the vast majority of bacterial genes, so we made the assumption that the annotated start sites of known genes would usually, but not always, be correct." (Sommer and Salzberg, 2021). Machine learning models can be poor at making predictions for classes (e.g. genes) whose training data exhibit high levels of bias, error, are under-represented for specific groups (e.g. gene families) and groups for which they have not been trained (Schafer and Graham, 2002). In addition to this, prokaryotic gene families are chronically under-sampled (Warren et al., 2010). It is becoming clear, that even with these advances in computational approaches, it is unlikely that we will ever be capable of identifying the complete picture of CDS gene diversity without exhaustive experimental work.

The majority of existing studies which aimed to evaluate gene prediction tools have focused on those developed for eukaryotes (Wang, Chen, and Li, 2004; Mathé et al., 2002), such as plants (Pavy et al., 1999) and vertebrates (Wang et al., 2003), and are now over at least a decade old. Acknowledgement that many of the methods used for annotation are still unchanged from their early conception is in the literature as recently as 2019, however, no solutions are provided (Salzberg, 2019). By comparison, little work has been published which systematically evaluates the performance of such tools, let alone those applied to prokaryotic genomes. Those studies which do aim to evaluate prokaryotic ORF prediction generally only investigate a small number of tools, focus on algorithm design and do not detail the complexities and nuances involved in reporting prediction accuracy with few other informative metrics used (Al-Turaiki et al., 2011; Mathé et al., 2002). Measures to compare both novel and contemporary tools are not well established or universally employed. Prediction accuracy is reported, with few other metrics used and the complexities and nuances omitted (Al-Turaiki et al., 2011; Mathé et al., 2002). The focus of such publications are often on algorithm design and not on a standardised system to compare

and analyse errors without human intervention (Devos and Valencia, 2001). Furthermore, such work to investigate genome annotation methods often omits important information such as the degree to which a predicted gene's loci is inaccurate at the 5′ or 3′ end or require extensive manual inspection, "all previous work on the estimation of error rates is heavily based on the intervention of human experts" (Devos and Valencia, 2001). This prevents researchers from gaining meaningful insight into the specific features of genes which are routinely missed, leading to a missed opportunity to improve our understanding of prokaryote genome content and whether certain types of genes are routinely missed, the types of inaccuracies attributed to a given method or the causality of such mispredictions.

> *"It is paradoxical that the sophisticated computerized methods used during the sequencing and annotation of whole genomes have not been followed by a systematic evaluation of the associated errors of these predictions." (Devos and Valencia, 2001)*

To address these concerns, in this chapter, I extensively evaluate a collection of 15 widely used CDS prediction tools that form the basis of most of the annotations deposited in public databases and therefore have largely been used to build the genomic knowledge used by the scientific community. Provided is a comparison platform developed to allow researchers to compare 12 primary and a further 60 secondary metrics to systematically compare the predictions from these tools and study the effect on the resulting genome annotations for their species of interest. This platform allows for in-depth and reproducible analysis of aspects of gene prediction which are often not investigated and allows researchers to understand the impact of tool choice on the resulting prokaryotic gene collection. Further analysis in this study was conducted using Prodigal (Hyatt et al., 2010) as it was not only shown to be the top-performing tool, but also has had wide contemporary use in metagenomic and pangenomic studies as part of the PROKKA (Seemann, 2014) and Roary (Page et al., 2015) platforms. Additionally, as Prodigal is not part of the GeneMark family, which has been responsible for over two decades of genomic annotations, we incorporated it into this study in order to review a leading tool with a level of independence from previous methodologies.

2.3 Materials and Methods

2.3.1 Current Ensembl Genome Annotations

Six bacterial model organisms (MO) and their canonical annotations were downloaded from Ensembl Bacteria (Howe et al., 2020)[1]. *Bacillus subtilis (B. subtilis)* BEST7003 strain (assembly ASM52304v1), *Caulobacter crescentus (C. crescentus)* CB15 strain (assembly ASM690v1), *Escherichia coli (E. coli)* K-12 ER3413 strain (assembly ASM80076v1), *Mycoplasma genitalium (M. genitalium)* G37 strain (assembly ASM2732v1), *Pseudomonas fluorescens (P. fluorescens)* UK4 strain (assembly ASM73042v1), *Staphylococcus aureus (S. aureus)* 502A strain (assembly ASM59796v1), were chosen for their scientific importance, range of genome size, GC content, assumed near complete and high quality genome assembly and annotation provided by Ensembl Bacteria. They are presented in Table 2.1 and further information regarding these model organisms can be found in Appendix Section A.1.

Model Organism **[Assembly]**	Genome Size (Mbp)	Genes **[CDSs]**	Genome Density **[CDSs]**	GC Content
Bacillus subtilis BEST7003 **[ASM52304v1]**	4.04	4,133 **[4,011]**	88.91% **[87.60%]**	43.89%
Caulobacter crescentus CB15 **[ASM690v1]**	4.02	3,875 **[3,737]**	90.60% **[90.23%]**	67.21%
Escherichia coli ER3413 **[ASM80076v1]**	4.56	4,257 **[4,052]**	86.28% **[84.35%]**	50.80%
Mycoplasma genitalium G37 **[ASM2732v1]**	0.58	559 **[476]**	92.03% **[90.62%]**	31.69%
Pseudomonas fluorescens UK4 **[ASM73042v1]**	6.06	5,266 **[5,178]**	84.75% **[84.20%]**	60.13%
Staphylococcus aureus 502A **[ASM59796v1]**	2.76	2,556 **[2,478]**	83.93% **[82.76%]**	32.92%

TABLE 2.1: An overview of genome composition for the 6 model organisms selected to evaluate CoDing Sequence (CDS) prediction tools compiled from data held by Ensembl Bacteria. Data is presented for all genes and CDS genes in bold square brackets. Note the relatively broad differences in genome size, gene density (percentage covered with annotation) and GC content.

[1] Available at https://github.com/NickJD/ORForise/tree/master/Genomes

For each of the chosen MOs, two data files were downloaded from Ensembl Bacteria; the complete DNA sequence (∗*_dna.toplevel.fa*) and the GFF (General Feature Format) file (∗*.gff3*) containing the position of each gene. The current collection of CDS genes presented in the MO annotations from Ensembl (Current Ensembl Annotation - CEA) were taken as the reference annotations for this study. Prokaryotic genomes exhibit high levels of gene density, often with little extraneous DNA, which is "commonly perceived as evidence of adaptive genome streamlining" (Sela, Wolf, and Koonin, 2016). Unannotated DNA represents between ∼10%-20% of the six MO genomes selected and while an additional 0.38% - 2.22% is attributed to non-coding annotations, there is still a measurable portion of each genome without any annotation. This study focuses specifically on the identification of CDSs, which constitute the significant majority of annotated genomic regions in the 6 genomes studied (82.76% - 90.62%, see Table 2.1).

The CDSs from each of the 6 genomes exhibit a range of differences which are known to impact the ability of prediction tools to identify them. These include, but are not limited to, GC content, codon usage and gene length. The GC content varies from 31.69% - 67.21% for these genomes, and even within a single genome, the CDS GC content varies widely (see Figure 2.1 for distributions). Furthermore, the canonical ATG start codon is used between 68.58% - 90.67% of the genes for the six genomes (see Table 2.2)

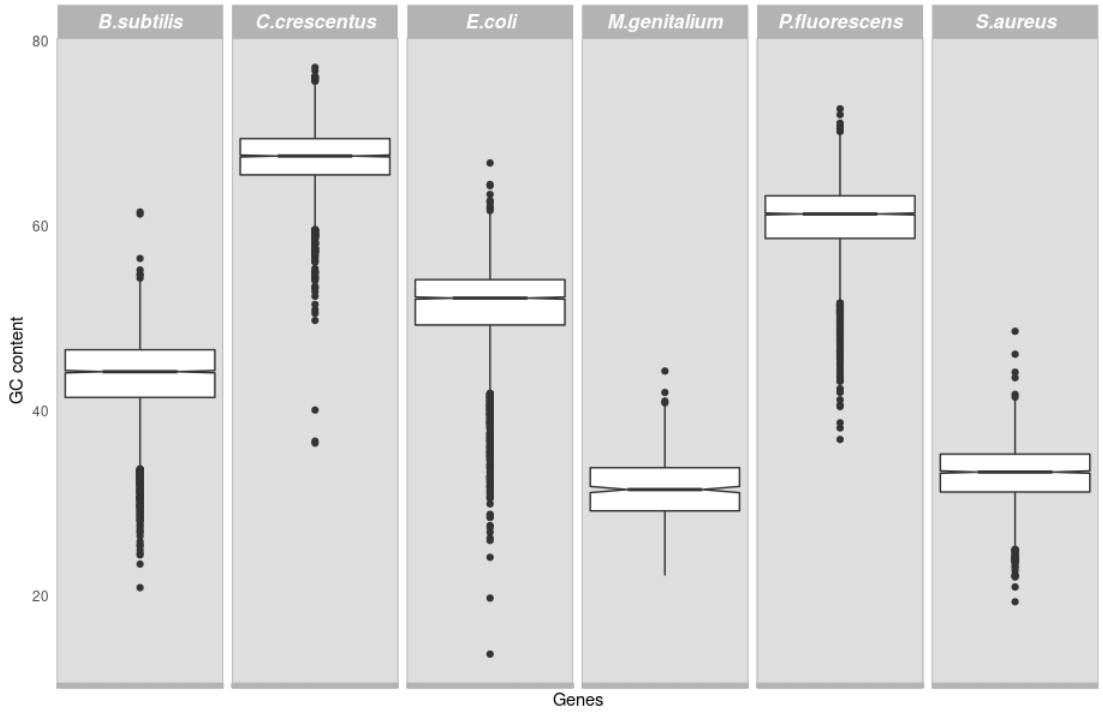


FIGURE 2.1: GC content of the six model organisms and their Ensembl annotated CoDing Sequences (CDSs). Note the high levels of variance within and between each genome.

Model Organism	ATG	GTG	TTG	ATT	CTG	Other
B. subtilis	76.81%	10.10%	13.09%	0.00%	0.00%	0.00%
C. crescentus	68.58%	17.69%	13.73%	0.00%	0.00%	0.00%
E. coli	90.67%	7.50%	1.70%	0.05%	0.05%	0.02%
M. genitalium	88.45%	7.56%	3.99%	0.00%	0.00%	0.00%
P. fluorescens	88.55%	7.55%	2.92%	0.21%	0.48%	0.29%
S. aureus	86.80%	6.62%	6.58%	0.00%	0.00%	0.00%

TABLE 2.2: Start codon usage for Current Ensembl Annotation CoDing Sequence (CDS) genes for the six model organisms. Note the variation in usage of canonical start codon ATG and the alternative GTG and TTG codons.

Additionally, *M. genitalium* uses the genetic code 4 (Pritchard et al., 1990) codon translation table, meaning one of the three universal stop codons (TGA/UGA) is instead used to code for tryptophan (Dybvig and Voelker, 1996), whereas the other 5 model organisms use the universal translation table 11 (see Tables 2.2 and 2.3 for more detail). While a similar median CDS length is shared across the six genomes, *B. subtilis* and *P. fluorescens* have a number of long genes ($>$ 8,000 nt, see Figure 2.2) and *S. aureus* contains the 31,421 nt "giant protein Ebh" (Cheng, Missiakas, and Schneewind, 2014) which is more than twice the length of the next largest CDS in this study. The diversity across the rest of prokaryotes is likely to be as great as, or greater than, reported here for these six.

Model Organism	TAG	TAA	TGA	Other
B. subtilis	13.96%	62.93%	23.11%	0.00%
C. crescentus	32.78%	19.86%	47.36%	0.00%
E. coli	6.89%	64.68%	28.41%	0.02%
M. genitalium	27.10%	72.90%	0.00%	0.00%
P. fluorescens	14.18%	30.42%	55.41%	0.00%
S. aureus	15.01%	74.17%	10.82%	0.00%

TABLE 2.3: Stop codon usage for Current Ensembl Annotation CoDing Sequence (CDS) genes for the six model organisms. *M. genitalium* recodes TGA for Tryptophan and *E. coli* uses CTT for one gene.

As discussed in Background section 1.3.1.1, there are competing definitions for an ORF or what is named a CDS gene here. The Sequence Ontology (Eilbeck et al., 2005) describes an ORF as "The in-frame interval between the stop codons of a reading frame which when read as sequential triplets, has the potential of encoding a sequential string of amino acids". However, it is conventional for ORFs to be reported as regions of DNA encompassed by a start and stop codon as a start codon is expected to indicate the start of DNA transcription (Brent, 2005). We acknowledge the difference in ontological definition and during this study, we refer to the region of DNA between an in-frame start and stop codon that is predicted to encode for an amino acid (protein) sequence, as a predicted CDS.

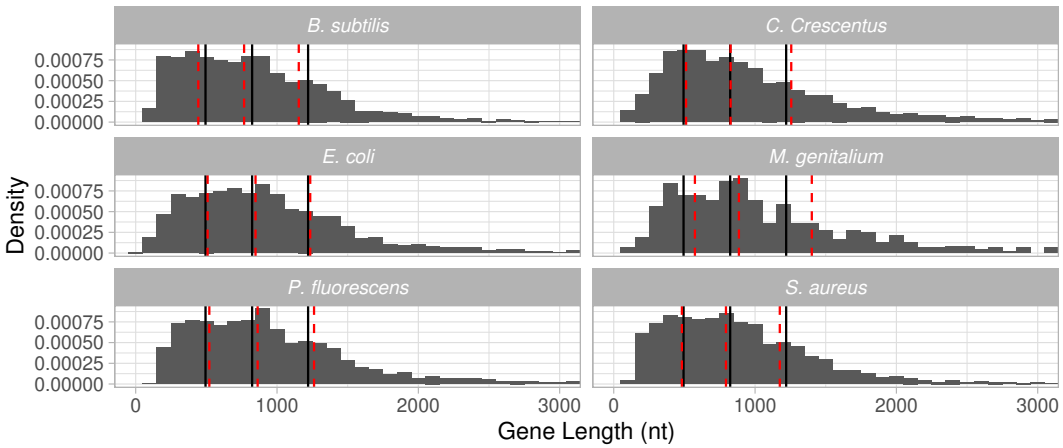


FIGURE 2.2: CoDing Sequence (CDS) lengths plotted for each model organism. The black, solid vertical lines are at the overall first quartile (494), median (824) and third quartile (1220) for all six model model organisms. The red dotted lines show the first quartile, the median and the third quartile for each organism individually. The x-axis is truncated at 3000 nt. The proportion of CDS lengths at or below this value are 0.964 for *M. genitalium*, 0.984 for *P. fluorescens*, 0.987 for *E. coli*, *S. aureus* and *C. crescentus*, and 0.990 for *B. subtilis*. A total of 23 CDSs were longer than 5000 nt. The distributions of CDS lengths for *E. coli*, *S. aureus*, *C. crescentus* and *P. fluorescence* are comparable to the overall distribution. The lengths for *B. subtilis* are somewhat smaller than expected overall, while the lengths for *M. genitalium* are longer than expected.

2.3.2 Prediction Tools

No.	Tool Name	Version	Reference
1	Augustus	3.3.3	Keller et al., 2011
2	EasyGene	1.2	Nielsen and Krogh, 2005
3	GeneMark.hmm	3.2.5	Lukashin and Borodovsky, 1998
4	GeneMark	2.5	Borodovsky and McIninch, 1993
5	FGENESB	'2020'	Salamov and Solovyevand, 2011
6	Prodigal	2.6.3	Hyatt et al., 2010
7	GeneMarkS	4.25	Besemer, Lomsadze, and Borodovsky, 2001
8	GeneMarkS 2	'2020'	Lomsadze et al., 2018
9	GLIMMER 3	3.02	Delcher et al., 2007
10	GeneMark (H.A)	3.25	Besemer and Borodovsky, 1999
11	TransDecoder	5.5.0	Haas et al., 2013
12	FragGeneScan	1.3.0	Rho, Tang, and Ye, 2010
13	MetaGene	2.24.0	Noguchi, Park, and Takagi, 2006
14	MetaGeneMark	'2020'	Zhu, Lomsadze, and Borodovsky, 2010
15	MetaGeneAnnotator	2008/8/19	Noguchi, Taniguchi, and Itoh, 2008

TABLE 2.4: Version number and reference for all tools used in this study. Tools 1-5 inclusive are model based tools. Tools 6-15 inclusive are *ab initio* based tools. Where no version number is available, the year when the tool was used is listed.

This study specifically investigates CDS predictors, tools which apply complex filtering after the identification of ORFs across a region of DNA. This is different to ORF finders, which return unfiltered ORFs (Stothard, 2000) that meet a set of predefined rules such as length and in-frame start and stop codons. This filtering is unique to each tool and dependent on properties such as codon usage, GC content, CDS length, overlap and similarity to known genes, and other more sophisticated parameters modelled on analysis of previously studied genes and genomes. Without such filtering methods, CDS predictors would typically report many false positives such as nested or heavily overlapping CDSs. An example of filtering can be found in in the GeneMark (Borodovsky and McIninch, 1993) algorithm which reports multiple variations of the same CDS with confidence scores. For this study, we chose the longest for each CDS after assessing the results.

We selected 15 different CDS prediction tools, some of which required a model (a rigid set of parameters adjusted to a particular organism), and the others which predicted *ab initio* from sequence. The tools which required a model were: GeneMark.hmm with *E. coli* and *S. aureus* models (Lukashin and Borodovsky, 1998); FGENESB with *E. coli* and *S. aureus* models (Salamov and Solovyevand, 2011); Augustus with *E. coli*, *S. aureus* and *H. sapiens* models (Keller et al., 2011); EasyGene with *E. coli* and *S. aureus* models (Nielsen and Krogh, 2005); GeneMark with *E. coli* and *S. aureus* models (Borodovsky and McIninch, 1993). Those which did not require a model were: GeneMarkS (Besemer, Lomsadze, and Borodovsky, 2001); Prodigal (Hyatt et al., 2010); MetaGeneAnnotator (Noguchi, Taniguchi, and Itoh, 2008); GeneMarkS-2 (Lomsadze et al., 2018); MetaGeneMark (Zhu, Lomsadze, and Borodovsky, 2010);

GeneMarkHA (Besemer and Borodovsky, 1999); FragGeneScan (Rho, Tang, and Ye, 2010); GLIMMER-3 (Delcher et al., 2007); MetaGene (Noguchi, Park, and Takagi, 2006); TransDecoder (Haas et al., 2013). The two groups are referred to as 'model-based' and '*ab initio*' henceforth and can be seen in table 2.4. Notably, TransDecoder was developed to predict coding regions within transcript sequences, often in eukaryotes.

2.3.2.1 Prediction Tool Types

2.3.2.1.1 Model-based tools , those designed for specific genomes require a model, or a pre-configured rigid set of parameters to perform predictions. These were common at the start of automated genome annotation and are still used widely for eukaryote annotation. The construction of these models rely heavily on having an accurate and complete set of genes and their gene families for a particular organism (among other information). Any inaccuracies or biases in the data that the models are produced from, are therefore likely to be present in the final models. Model-based gene predictors trained on a particular species are expected to preform well on strains with comparable gene and genome structure. However, there can be large differences in gene number, gene length and genome size within strains of the same species. Overfitting can occur, where only similar genes to those found in the databases are detected at a high sensitivity. Model-based prediction for certain model organisms where specific strains are often used for scientific and industrial purposes can still be effective as there may be little genetic difference between two isolates of the same strain.

2.3.2.1.2 *Ab initio* tools or self-training tools such as Prodigal do not require any prior knowledge of the genome it is to annotate and either trains a model from statistics gathered directly from the genome itself or simply employ a set of pre-defined parameters for the prediction. The criteria considered while making predictions include but it not limited to, overlapping ORFs, GC content of an ORF, length, start and stop codons and distances between ORFs(Delcher et al., 1999) (Besemer, Lomsadze, and Borodovsky, 2001). Unfortunately, these criteria are still based on prior knowledge as deciding between candidate ORFs still requires a number of assumptions based on previously studied genes and genomes which the developer has programmed into the tools. Many of these assumptions are made from studying previously annotated genomic sequences and therefore if certain types of genes are under or over represented in previous studies, the biases are likely to impact the predictions.

2.3.2.1.3 Metagenomic gene prediction tools form a subset of self-training *ab initio* tools which primarily rely on the same methods but are designed to contend with a number of additional difficulties common to metagenomic annotation. The dynamics of metagenomic DNA sequences such as chimeric and fragmented contigs assembled from different organisms, cause a number of problems for even self-learning predictors. Parameters chosen would need to be recalculated for every metagenomic contig as each is likely to have different characteristics. A given contig can be positioned either at the start, middle or end of a gene. Therefore simply looking for start and stop codons, which may not be present, along with changes in GC content outside of predicted gene regions, will not help to distinguish between coding and non-coding regions. These errors are extremely difficult to account for and tools have been produced to tackle them directly (Rho, Tang, and Ye, 2010) This study includes four prediction tools developed to work on metagenomes in this study, MetaGeneMark, MetaGene, MetaGeneAnnotator and FragGeneScan.

2.3.2.1.4 Whole genome annotation 'pipelines' such as PROKKA (Seemann, 2014) and NCBI's PGAP (Tatusova et al., 2016) were not included, but the initial CDS prediction components embedded in these pipelines such as Prodigal and GeneMarkS-2 were included in the study. Multiple separate tools from the GeneMark family (Besemer and Borodovsky, 2005) were included (some superseded) due to their extensive use and impact on genomic knowledge over the last three decades. Additionally, number of tools and methods were excluded for reasons as; being superseded by newer versions as in the case of GLIMMER 1 (Salzberg et al., 1998) and GLIMMER 2 (Delcher et al., 1999) being replaced by GLIMMER 3 (Delcher et al., 2007). Tools were also excluded where the original source code/online framework is no longer available or in the case of Orphelia (Hoff et al., 2009), where the software requires versions of operating systems and other software which are no longer supported and or used today. Tools which require any type of external evidence such as a sequence similarity match (BLAST) (Altschul et al., 1990) were also not examined because they are able to leverage information outside of the genomic information provided by the DNA sequence of the genome. Furthermore, contemporary genome annotation methods can implement a number of different strategies which can make evaluations and comparisons difficult. While some are only designed to report predicted CDSs, others produce predictions for other functional elements such as RNA components and frame-shifted genes. Only those tools which specifically predict CDSs from only DNA sequences have been chosen.

2.3.2.2 Run-time Parameters

To emulate the annotation process of a novel or less studied genome or metagenome, each tool was run using its default parameters (see Appendix Subsection A.2 for more detail). All tools were given the same input data from Ensembl for each analysis: a DNA (FASTA) file containing the entire assembled genome. The tools which required training models were given models from two different organisms, *E. coli - K-12* and *S. aureus* - Mu50 (strains were chosen when possible). These were chosen as the organisms were both in the chosen set of six bacteria and were freely available as training models in all the tools which required them. It was also decided that for Augustus, as it was originally developed for Eukaryote ORF prediction, the *H. sapien* model would be included in the analysis. This model was developed to identify not only ORFs but also exons which Eukaryotic genomes exhibit withing their genes. More information regarding each group and tool, and the parameters used to run them, can be found in Appendix Section A.2.

While most of the tools were available as online resources, those which were required to be downloaded and run locally, were run on a 64-bit Linux based machine with a i7 2600k CPU with 32GB RAM. None of the tools took more than a few minutes to run or required more than 500 MB of RAM. The gene prediction tools which were used offline in this study were downloaded from the links in their publications.

GeneMark, MetaGeneMark, GeneMark Heuristic Approach, GeneMark Hidden Markov Model, GeneMark S and GeneMark S2 were all from the same suite of tools and have many similarities with each other but are designed for different purposes and produce different results.

It was decided that no specific rules were to be enforced on the tools. Each tool was run on its default parameters and this was to get a baseline for their accuracy with the least amount of human support. Many hard-coded assumptions were consistent across the tools, such as minimum ORF length and the codons allowed to identify the start and end of an ORF. Some of the tools allowed the minimum ORF length to be altered, but the majority fixed the threshold to around 100 nucleotides. The three stop codons TAG, TAA and TGA were used by all tools to signal the end of an ORF. Alternative codon tables were supported by many of the tools which for example could have been selected to allow for TGA to code for Tryptophan or Glycine and not a stop codon as found in *M genitalium*. The default codon tables were selected for all tools. The availability and preference of start codons also differs among the tools, ATG is the most ubiquitous and most likely due to its having the highest translation efficiency (Hecht et al., 2017).All tools studied here use ATG to indicate the start of an ORF, whereas some allowed a wider choice of start codon. Some of the tools such as GLIMMER 3 also give increased weighting to ORFs which start with ATG. The tools which allow a set of different start codons however only allowed between 2 to 5 different start codons.

2.3.3 Comparison Method

A systematic software platform ORForise (ORF Authorise) was built to perform a fair, comparative, and informative analysis of the different tools examining different aspects of their predictions (see Appendix Subsection A.3.1 for more detail). Version 1.0 of the platform, written in Python3 (Van Rossum and Drake, 2009), was used and is freely available at `https://github.com/NickJD/ORForise`. It has been designed to process the standardised GFF3 format as well as the individual output formats produced by each tool listed in this study.

In this platform we endeavoured to choose a wide range of metrics that clearly and representatively capture the many intricacies of the predictions. A number of metrics used in previous studies, such as the number of CDSs predicted, accurate identification of start positions or the number of genes correctly detected, can give some indication of the 'accuracy' of each tool. However, it was found during our analysis that there were many complexities in the prediction results which would not be represented by these high-level metrics. For example, predicted CDS regions may overlap with one or more known CEA genes but be inaccurately extended or truncated on either the 5′ or 3′ end. It is also common for smaller CEA genes to be mistakenly encompassed by larger predicted CDSs and while the nucleotide regions of these genes are technically within the predicted regions, even if in-frame, they do not represent the true protein-coding sequence. Furthermore, different types of inaccuracies may be more or less important, depending on the aim of any given study. Therefore, clear and specific measures of accuracy that describe the detection of the entire locus of a gene are needed. Figure 2.3 illustrates how we determine correct CEA gene detection, but also explains its nuances and complexities. An example of this is the definition of short ORFs, which in prokaryotes are often described as having lengths of 100-300 nt (Storz, Wolf, and Ramamurthi, 2014; Duval and Cossart, 2017; Su et al., 2013). However, due to hard-coded cutoffs in many of the tools, we chose the 'upper-bound' of 300 nt or 100 codons to define short ORFs. We iteratively developed 72 metrics to help provide the most accurate and informative representation of a tool's prediction quality (see Figure 2.4. Additionally, as part of the ORForise platform, we provide a number of Python3 post-analysis scripts developed to aid in the interrogation between the CEA gene annotations and the CDSs predicted by each of the tools studied. These scripts were used to extract characteristics that are useful in the investigation of why specific CEA-genes are detected, missed or incorrectly reported.

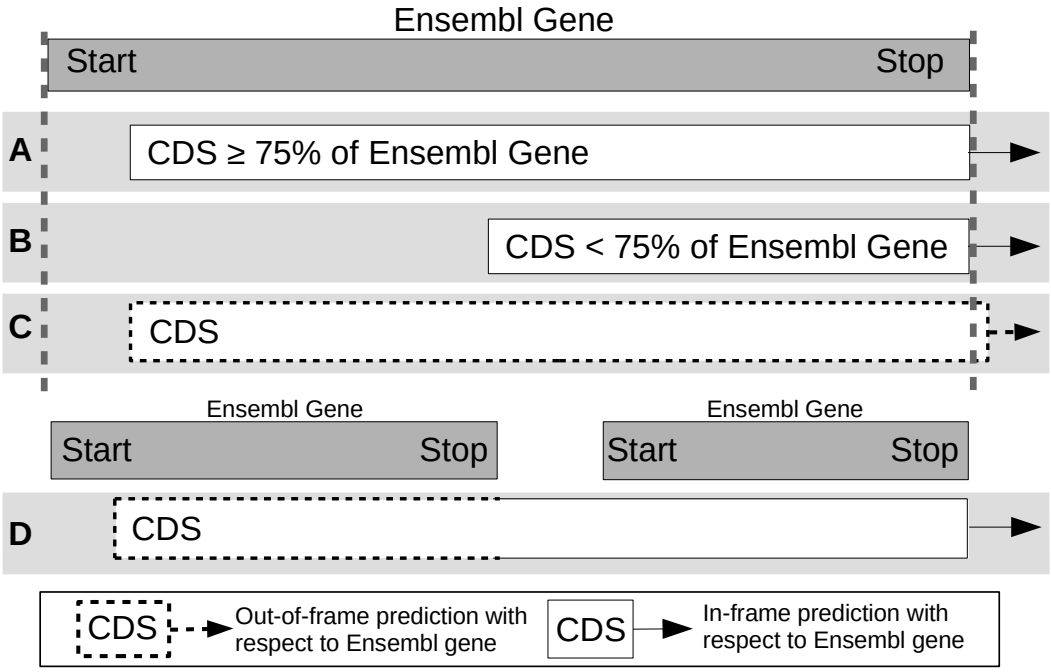


FIGURE 2.3: Illustration of how predicted CDSs are classified as having detected or not detected the CEA genes. Predicted CDSs are compared to the genes held in Ensembl. A - The predicted CDS covers at least 75% and is in-frame with Ensembl gene and therefore it is recorded as detected. B - The predicted CDS covers less than 75% of the Ensembl gene and therefore is recorded as not detected. C - The predicted CDS covers part of an Ensembl gene but is out of frame (dotted outline) and therefore is recorded as missed. D - The use of alternative stop codons causes the predicted CDS to be truncated or divided into two CDSs that span the Ensembl genes and therefore is recorded as missed.

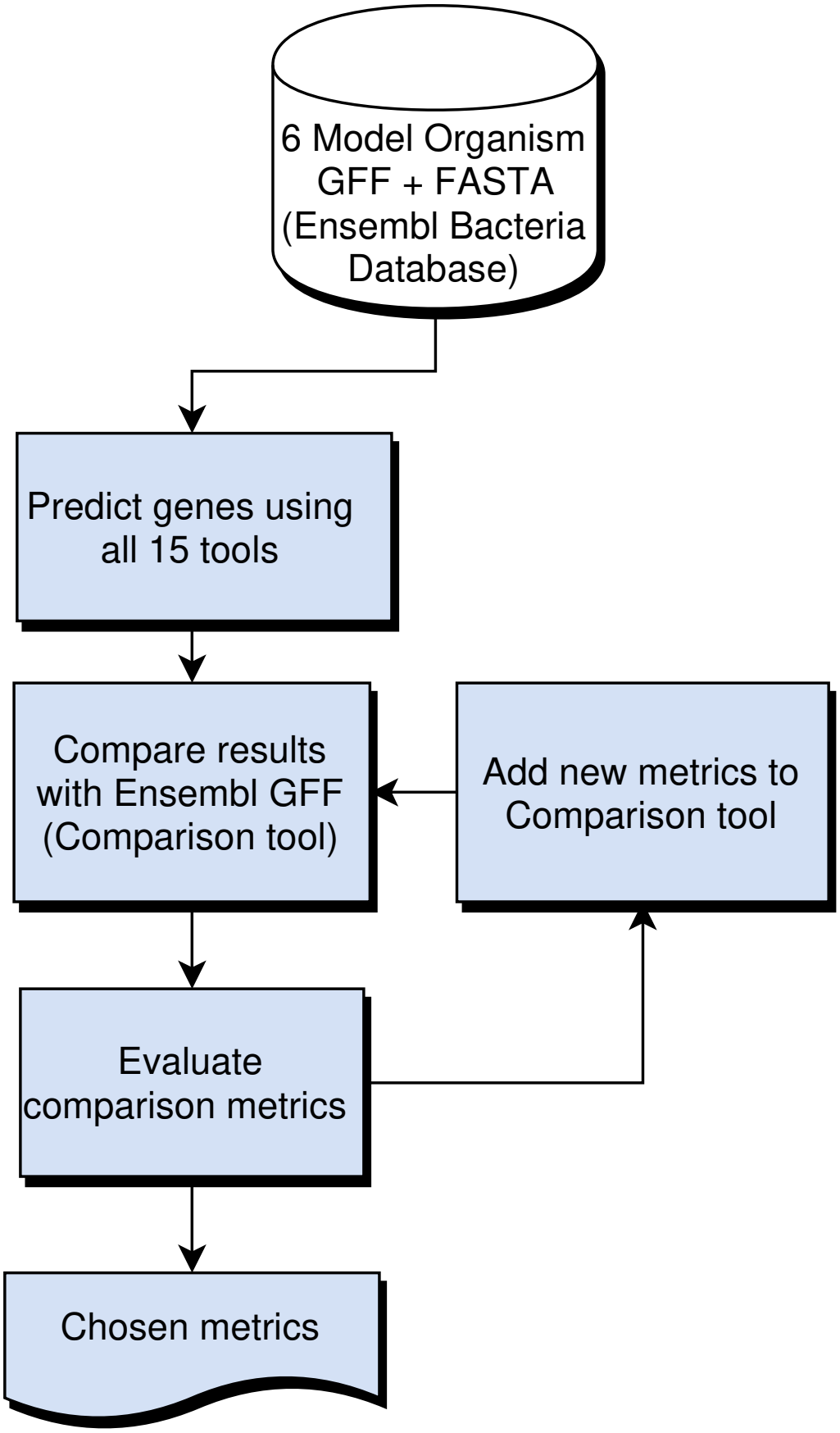


FIGURE 2.4: The 72 metrics used in this study to differentiate the predictions of the different tools were formed from a number of analysis cycles as shown.

2.3.4 Aggregated Tool Predictions

An extension to the ORForise comparison platform was built (Aggregate_Compare) to investigate whether an aggregation of predictions from a number of top-performing tools would perform better than individual tools (see Appendix Subsection A.3.2 for more detail). The CDS predictions from the selected tools are combined into a single data structure with duplicate CDSs filtered out, but alternative predictions for the same locus retained and ordered according to start position. The same comparison algorithm could then be employed on the set of unique CDS predictions identified by this union of the outputs of the selected tools (Prodigal, GeneMark-S-2, MetaGeneAnnotator, MetaGeneMark and GeneMark-S - chosen due to their individual performance) and as with the singular tool comparison, for every CEA gene, the CDS which deviated the least from the correct locus was selected as the closest match.

2.3.5 Discovering Additional ORFs

To enable the aggregation of different CDSs from contemporary and new annotations, we provide GFF_Intersector to create a single GFF representing the intersection of two existing annotations (see Appendix Subsections A.3.3 and A.3.4 for more detail). This also provides an option to allow the retention of CDSs that have a user-defined difference (default minimum 75% coverage and in-frame). Additionally, we also provide the GFF_Adder tool, which produces a new GFF containing CDSs from an existing annotation, plus the new CDSs, filtered to remove any that overlap existing CDSs by more than 50 nt (user definable).

2.4 Results

2.4.1 Metrics for Comparison of Tools

72 different metrics were chosen for this exhaustive evaluation in order to give the broadest possible scope to compare and contrast the performance of the tools. The full definitions for each of these metrics can be found in Appendix Section A.4 and are intended to be used as a resource for the community when deciding which tool to apply to both novel and contemporary genome annotation work. The following are 12 of the most informative metrics, selected for their ability to represent both a broad range and depth of different attributes which have been used to distinguish the prediction tools.

- **M1** Percentage of Genes Detected
- **M2** Percentage of Predicted CDSs that Detected a Gene
- **M3** Percentage Difference of Number of Predicted CDSs
- **M4** Percentage Difference of Median Predicted CDS Length
- **M5** Percentage of Perfect Matches
- **M6** Median Start Difference of Matched Predicted CDSs
- **M7** Median Stop Difference of Matched Predicted CDSs
- **M8** Percentage Difference of Matched Overlapping Predicted CDSs
- **M9** Percentage Difference of Matched Short Predicted CDSs
- **M10** Precision
- **M11** Recall
- **M12** False Discovery Rate

M1, Percentage of Genes Detected, is often used as the main indicator of tool performance in other comparisons but interpreted differently between studies. Here it is characterised as a predicted CDS which is in frame with a CEA gene and has captured at least 75% of its nucleotide sequence (Figure 2.3 A). In contrast to M1, which indicates when underprediction (or false negatives) occur, M2 suggests when overprediction (or false positives) have occurred.

For M3, M4, M8 and M9, *Percentage Difference* was used to identify differences between predicted and CEA metrics: 100∗*(Predicted CDS metric - Ensembl Gene Metric) / Ensembl Gene Metric*. The best score for a metric using the *Percentage Difference* calculation is 0, as 0 represents no deviation from the CEA annotations. The 'Matched CDSs' identifier used for M6, M7, M8 and M9 represent the CDSs which have correctly detected an CEA gene. M6 and M7 are calculated by taking the median codon position differences recorded for mispredicted start or stop codons. Metrics such as the Percentage of Perfect Matches (M5) can give a clearer overview of a tool's 'accuracy' or performance, as it is common for a tool to misidentify either the exact start or stop locus of a detected CEA gene, while metrics such as Median Start Difference of Matched Predicted CDSs (M6) can help establish the level of inaccuracy.

The tools were ordered by totalling the rankings for each of these 12 metrics, across the 6 model organisms. Supplementary Results 1 (https://github.com/NickJD/ORForise/tree/master/Supplementary_Data) contains the results used for the ranking. This ranking, based on a wide range of different performance measures, allows for a comparative overview of contemporary and future tools, and is presented in Figure 2.5. This figure also shows the Percentage of Genes Detected (M1) with an overlay of the Percentage of Perfect Matches (M5), demonstrating the inconsistency between the two metrics for each tool. Metrics such as Percentage of Genes Detected (M1) and Percentage of Predicted CDSs that Detected a Gene (M2) are informative and can be representative of a tool's prediction quality, however, they do not convey the complete picture when presented in isolation. This is of particular importance for those working with metagenomic or other fragmentary assemblies, as the likelihood of incomplete fragments and chimeric sequences is higher and can lead to varying mispredictions. Although the overall prediction quality of genes was high across most of the tools and genomes in this study, the additional metrics produced can be used to identify strengths and weaknesses inherent to them. For example, GeneMark.hmm (*S. aureus* model and genome), MetaGeneMark and MetaGeneAnnotator, GeneMarkS were all ranked highest for Percentage of Genes Detected (M1) for at least one model organism, while Prodigal and GeneMarkS were ranked highest twice (GeneMarkS and GeneMark.hmm were ranked joint highest for *S. aureus*). However, when inspecting the 12 metrics in Figure 2.6, it was clear that there were complex differences between the prediction results of not only the highest scoring tools, but also the lower ranked tools which were often ranked high for some metrics in some of the genomes.

While no tool or group of tools consistently ranked highest or equally across the 12 metrics or model organisms, MetaGeneAnnotator ranked best for *B. subtilis* and *M. genitalium*, GeneMarkS-2 ranked best for *C. crescentus* and Prodigal ranked best for *E. coli*, *P. fluorescens* and *S. aureus*.

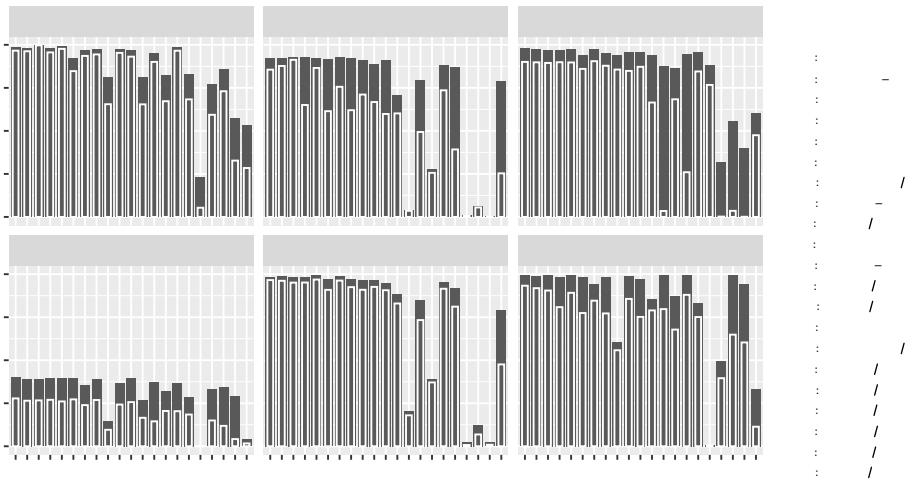


FIGURE 2.5: The result of all 15 gene prediction tools (21 with chosen models) on the 6 model organism genomes, ordered by the summed ranks across the 12 metrics. The *Y* axis represents the Percentage of Genes Detected (M1) by each tool in black and the Percentage of Perfect Matches (M5) in white. M5, which represents the ability for a tool to detect the correct start codon, has more variance between the tools than M1. Each column on the *X* axis represents a different tool (some model based tools were run multiple times). There is considerable variation in how well each tool performs across the different genomes, while all tools perform relatively poorly on the *M. genitalium* genome.

The combination of multiple metrics can be used to determine which tool should be used between two candidate tools with the same or similar Percentage of Genes Detected (M1). For *M. genitalium*, both GeneMarkS and MetaGeneMark obtained an M1 score of 39.50%, but MetaGeneMark reported a higher Percentage of Perfect Matches (M5) (65.96% compared to 61.17%) than GeneMarkS (see Figure 2.5) and is thus more accurate.

In addition, GeneMarkS is ranked first for Percentage of Genes Detected (M1) when applied to *P. fluorescens* with 99.29%, compared to Prodigal which is ranked 4th with 98.49%. However, Prodigal has the highest Percentage of Perfect Matches (M5), 92.86% vs 87.03% for GeneMarkS, which means that more of the CEA genes identified by Prodigal were exact matches. In this instance, choosing either Prodigal or GeneMarkS as the overall highest performing tool is not arbitrary.

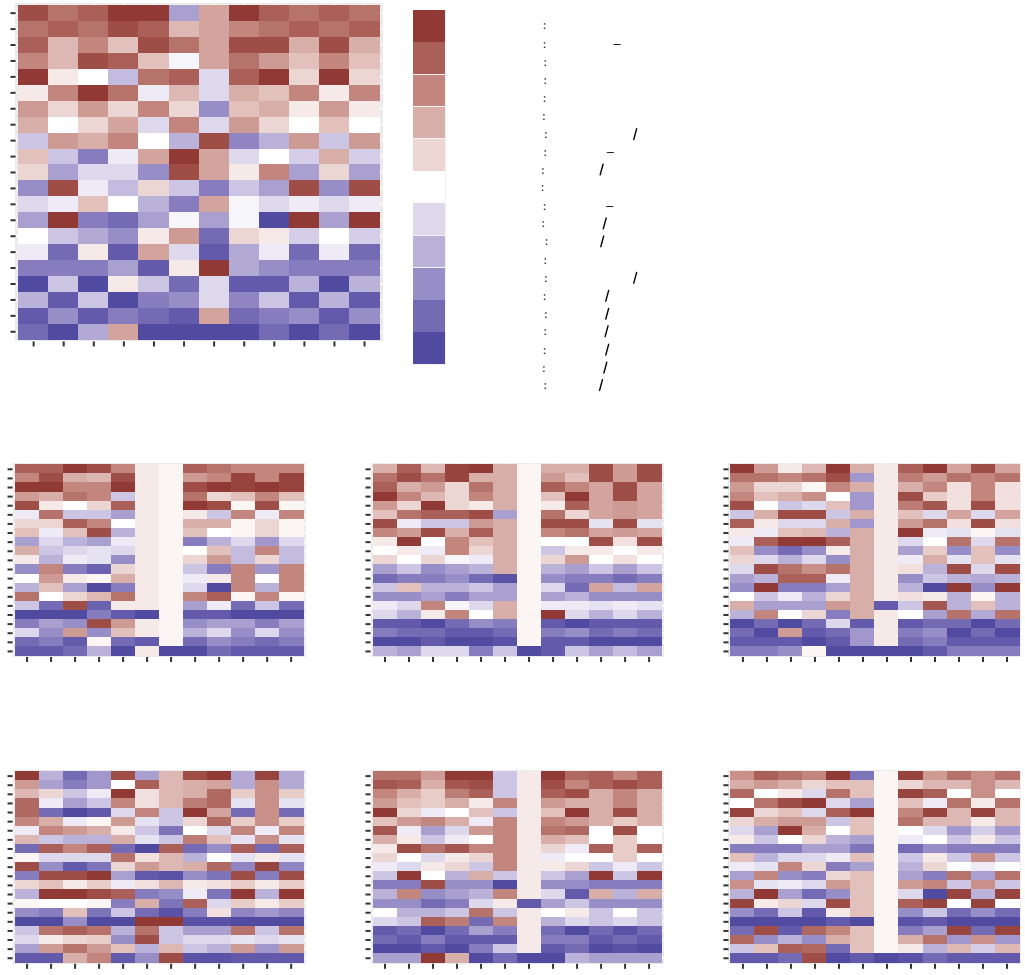


FIGURE 2.6: Heatmaps showing rankings of the tools by the 12 chosen metrics, overall and for each organism in turn. The tools are shown ordered by the summed ranks across the 12 metrics. While red is 'better' and blue is 'worse', it is clear that across the 6 model organisms, no tool stands out for these 12 metrics chosen as most representative. For example, for *C. crescentus*, GeneMark with *E. coli* model ranked 12th overall but reported the most accurate number of overlapping genes. For *P. fluorescens*, Prodigal was the overall highest ranked tool even though GeneMarkS detected the highest number of Ensembl genes. *M. genitalium* on the other hand, which uses an alternative stop codon, has some very interesting results showing the difficulty of identifying its genes by all tools. The pale coloured bands represent tools ranking the same for a particular metric.

2.4.2 Model-Based vs *ab initio* Tools

It was evident that the performance of model-based tools was less consistent across the 6 model organisms than the *ab initio* tools. They could perform as well as or better than a number of *ab initio* tools when the model selected was the same as the genome annotated. However, if genome and model were not the same, they often produced predictions of extremely low quality. For example, GeneMark with the *E. coli* model only predicted 71 CDSs for *S. aureus*'s 2,478 CEA genes, of which only 18 CDSs detected a CEA gene. However, while it could be expected that mixing different models and genomes could cause poor quality predictions from model-based tools, there were instances in which both model and genome were the same and the prediction performance was also poor. In particular, in the case of EasyGene using the *S. aureus* model, only 49.31% of *S. aureus* CEA genes were detected, a contrast from the ∼99% detected by the majority of *ab initio* tools.

Intriguingly, Augustus (a model-based tool) when employed with the *E. coli* model, was able to detect 96.64% of *P. fluorescens* genes. Both genomes are *Gammaproteobacteria*, and thus Augustus may be identifying common features of their genes. While this shows that model-based tools can perform well even when their model and target genomes are different, when Augustus was applied to *S. aureus* using the *S. aureus* model, it was only able to detect 20.53%, but unexpectedly detected 78.91% when using an *H. sapiens* model. This is indicative of the inconsistency of model-based prediction tools and the genome models they employ. In contrast, through the ranking approach we employed, the model-based tool GeneMark.hmm with the *E. coli* model ranked higher (7/21) than a number of *ab initio* tools in both the overall ranking and for individual metrics. Furthermore, GeneMark.hmm with the *S. aureus* model was joint top in detecting the highest number of *S. aureus* CEA genes with GeneMarkS. Additionally, for each of the model-based tools, the *E. coli* model performed better across the 6 model organisms than the *S. aureus* model.

2.4.3 GC Content

No significant variation was observed between the CEA gene median GC content and that of the predicted CDSs from each tool, even for those with poor predictions (see Supplementary Results 2 - `https://github.com/NickJD/ORForise/tree/master/Supplementary_Data`). As can be seen in Figure 2.1, each of the six genomes exhibit CEA genes with a wide range of GC content profiles, irrespective of their genome's median value. We note that the GC content of genes missed by Prodigal is lower for all six MOs, but within the 25-75th percentile range for all CEA genes (Figure 2.1 and Table 2.5). Notably, *E. coli* and *P. fluorescens* genes which were missed by Prodigal are nearly 10% lower in GC content than both detected and partial genes.

Model Organism	Ensembl GC	Detected GC	Partial GC	Missed GC
B. subtilis	44.19%	44.25%	43.99%	39.13%
C. crescentus	67.52%	67.71%	67.69%	65.65%
E. coli	52.15%	52.21%	52.14%	43.14%
M. genitalium	31.44%	32.90%	32.75%	30.76%
P. fluorescens	61.25%	61.36%	60.25%	53.36%
S. aureus	33.33%	33.33%	30.13%	32.62%

TABLE 2.5: GC content differences for Prodigal annotations. Shown here as median values are: GC content of Current Ensembl Annotation CoDing Sequence (CDS) genes, the genes detected by Prodigal, those Prodigal obtained a partial match and those it missed.

2.4.4 Overlapping CDSs

The overall number of CDSs predicted to have an overlap with another CDS varied across each of the tools and model organisms, with cases of both positive and negative percentage differences when compared to the CEA annotations (see Supplementary Results 2 - https://github.com/NickJD/ORForise/tree/master/Supplementary_Data). Proportionally, the number of overlapping CDSs reported by *ab initio* tools are closer to the number of overlapping CEA genes than those reported by the model-based group.

Most model-based tools underpredict the proportion of overlapping CDSs with the exception of GeneMark *E. coli* for *P. fluorescens*, which predicted 2,073 overlapping CDSs compared to the 1,251 reported by Ensembl (see Tables 2.6 & 2.7).

Model Organism	Ensembl	*Ab initio*	Model-Based
B. subtilis	21.37%	21.44%	15.44%
C. crescentus	32.73%	25.51%	21.84%
E. coli	22.53%	22.68%	18.20%
M. genitalium	46.43%	16.47%	11.65%
P. fluorescens	24.16%	25.42%	18.08%
S. aureus	19.61%	19.98%	15.72%

TABLE 2.6: Percentages of the Current Ensembl Annotation CoDing Sequence (CDS) genes and Predicted CDSs identified as overlapping. We show averages for *ab initio* and model-based predicted CDSs.

Correct detection of CEA overlapping genes is also a problem. By totalling and averaging the Percentage Difference of Matched Overlapping Predicted CDSs (M8), we were able to observe a clear difference between the two tool groups with respect to their ability to detect correct overlapping CEA genes (see Tables 2.6 & 2.7). The inability of the tools to account for the unusual nature of the *M. genitalium* genome was shown again with an average M8 across all tools of -88.21%, compared to the average of -27.77% for the other 5 genomes.

Group	Average	Standard Deviation	Standard Error
Matched, *ab initio*	-23.62%	7.16%	2.27%
Matched, model	-52.89%	24.79%	7.47%
All, *ab initio*	-6.07%	11.55%	3.65%
All, model	-30.15%	29.41%	8.87%

TABLE 2.7: Percentage Difference of overlapping predicted CDSs as compared to the Current Ensembl Annotation CoDing Sequence (CDS) genes. *Ab initio* and model based tools are separated into 2 groups each. 'Matched' represents the Percentage Difference for those predicted CDSs which were able to detect an Current Ensembl Annotation CDS gene whereas 'All' represents the Percentage Difference of the number of overlapping predicted CDSs across all predicted CDSs.

Furthermore, when making predictions for *E. coli*, while model-based tools such as Augustus and EasyGene with the *E. coli* model can closely predict the proportion of overall overlapping CDSs (Percentage Difference of -1.42% and -2.30% respectively), due to the poorer performance of these tools for correctly detecting CEA genes, their M8 scores for matched overlapping CDSs were substantially lower than the average score of the *ab initio* tools (grouped average of -52.89% as opposed to -23.62% - see Table 2.7). Prodigal exemplifies this difference between the two tool groups. It was able to predict all overlapping CEA from *P. fluorescens* and *S. aureus*, whereas even when paired with the same model and genome, model-based tools continued to perform poorly.

2.4.5 Short ORFs

The lengths of detected, partially matched and missed CEA genes when predicted by Prodigal are summarised in Figure 2.7. It shows that the CEA genes which were missed by Prodigal for each genome were substantially shorter in length than the genes which were detected, except for *M. genitalium*. For the other 5 model organisms, whose combined median length of missed genes is 317, less than half the combined median length of 837.5 of those detected (see Figure 2.7 and Supplementary Results 2 (`https://github.com/NickJD/ORForise/tree/master/Supplementary_Data`), it is alternative start codon selection which influences whether a predicted CEA is shortened or elongated.

The proportion of short CEA genes in the six genomes below 300 nt ranged from 4.8% to 13.6% for each of the 6 model organisms. All tools predicted many short CDSs for *M. genitalium* because they were incorrectly truncated due to its alternative stop codon usage. On average, *ab initio* tools were shown to be more likely to correctly detect short CEA genes across the other 5 model organisms (see Tables 2.8 and 2.9). Interestingly, unlike overlapping genes, short ORFs were more often overpredicted, but few were actually accurate when compared to the CEA. However, *ab initio* tools were much better suited to reporting the correct proportion of short predicted CDSs for all 6 genomes, often reporting the same proportion (see Table 2.8). While *M. genitalium* does exhibit the highest divergence in proportional reporting of short predicted CDSs, *ab initio* tools were still less divergent (see Table 2.10).

Model Organism	Ensembl	*Ab initio*	Model-based
B. subtilis	13.66%	12.58%	13.24%
C. crescentus	7.60%	8.11%	15.33%
E. coli	10.24%	10.45%	13.04%
M. genitalium	4.83%	38.44%	36.99%
P. fluorescens	7.84%	9.06%	19.01%
S. aureus	10.05%	11.26%	15.59%

TABLE 2.8: Percentage of the Current Ensembl Annotation CoDing Sequence (CDS) genes and predicted CDSs categorised as Short CDSs ($\leq$100 amino acids). We show averages for *ab initio* and model-based predicted CDSs. Note the large increase in Short CDSs predicted for *M. genitalium*.

Group	Average	Standard Deviation	Standard Error
Matched, *ab initio*	-26.38%	25.68%	8.12%
Matched, model	-53.69%	21.71%	6.55%
All, *ab initio*	9.07%	39.87%	12.61%
All, model	39.10%	91.22%	27.50%

TABLE 2.9: Percentage Difference of short predicted CDSs (≤100 amino acids) as compared to the Current Ensembl Annotation CDS genes. *Ab initio* and model based tools are separated into 2 groups each. 'Matched' represents the Percentage Difference for those predicted CDSs which were able to detect a Current Ensembl Annotation CDS gene whereas 'All' represents the Percentage Difference of the number of Short CDSs across all predicted CDSs. The results from *M. genitalium* were not included in this table's calculations.

Group	Average	Standard Deviation	Standard Error
Matched, *ab initio*	-27.34%	25.15%	7.95%
Matched, model	-55.28%	20.62%	6.22%
All, *ab initio*	261.11%	139.28%	44.04%
All, model	148.00%	164.58%	49.62%

TABLE 2.10: *M. genitalium*-only Percentage Difference of short CDSs (≤100 amino acids) as compared to the Current Ensembl Annotation CoDing Sequence (CDS) genes. *Ab initio* and model based tools are separated into 2 groups each. 'Matched' represents the Percentage Difference for those CDSs which were able to detect a Current Ensembl Annotation CDS gene whereas 'All' represents the Percentage Difference of the number of Short CDSs across all predicted CDSs.

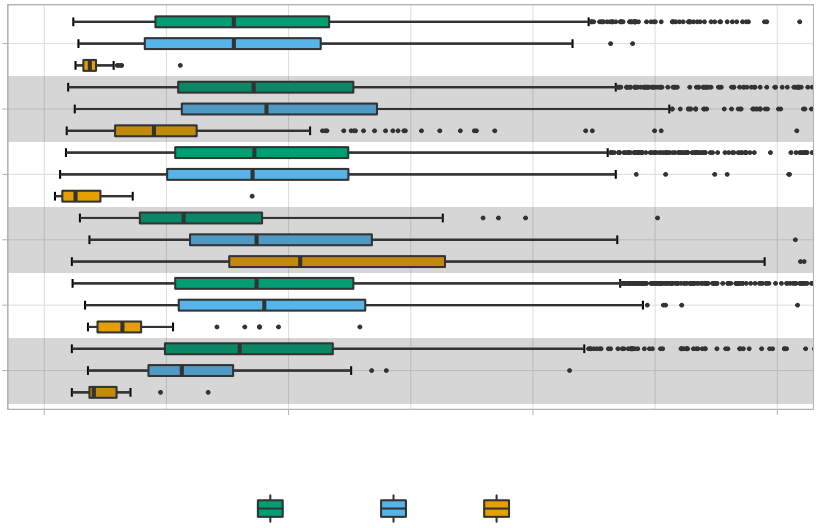


FIGURE 2.7: Lengths of Ensembl annotated genes, those which were partially matched by Prodigal and those which were missed, for each model organism. The x-axis is truncated at 3,000 nt. With the exception of *M. genitalium*, the distributions of lengths of the missed Ensembl genes are generally to the left of the distributions of the detected genes. Thus short genes are commonly overlooked by Prodigal and other tools.

2.4.6 Partial Matches

The number of missed CEA genes was low across the tools studied, with the exception of *M. genitalium* and some outliers from the model-based tools such as GeneMark, Augustus and EasyGene. However, we also found many genes that were incorrectly reported on the 5′ or 3′ end. These misannotations, which we have called 'partial matches' if in the correct frame and accounting for $>= 75\%$ of a CEA gene, constitute either an elongation or truncation of the protein product of the gene and therefore potentially have an unknown impact on the resultant sequence. A large number of genes were incorrectly reported on the 3′ end for *M. genitalium* by each tool. These 3′ prime truncated CDSs are explained by the alternative use of TGA as tryptophan in *M. genitalium* (tools incorrectly assume this encodes a stop codon). The stop codons predicted for *M. genitalium* by all the prediction tools were the same 'TGA,TAG,TAA' as for the CEA-genes of the other five model organisms. Interestingly, one CEA gene in *E. coli*, which used CTT as a stop codon, was missed by all tools except for FGENESB with its *E. coli* model. FGENESB incorrectly reported the very next codon, a TGA, as the stop position. This 78 nt CEA is the only example we found of a tool extending a CEA gene not from *M. genitalium*. Augustus with the Human model made a number of non-standard predictions due to its propensity to search for multiple CDSs for each gene but this is to be expected and is not reported in these results. Unlike 3′ prime misprediction, a large number of genes from all six genomes were predicted with alternative start codons (see Supplementary Results 2 https://github.com/NickJD/ORForise/tree/master/Supplementary_Data). This was true for all tools and especially a problem for *C. crescentus* with a relatively low 68.58% ATG start codon usage for all CEA genes. The CEA genes for which Prodigal was unable to obtain a 'Perfect Match' (M5), was just 37.40%. Prodigal used a much higher level of ATG (80.87%) for this set of partially matches genes. This misidentification of start codon usage was a consistent problem among all the tools and genomes studied. However, for *E. coli*, the level of misidentification was lower. As an example, the number of times the correct or incorrect start codons were selected by Prodigal, across all six model organisms, including the number of incorrectly chosen instances of the start codon (e.g. a different ATG further upstream of the real ATG) can be seen in Table 2.11.

	Correct codon				
	ATG	GTG	TTG	CTG	Other
Incorrect ATG	817	371	357	19	24
Incorrect GTG	106	76	49	4	0
Incorrect TTG	81	47	37	3	3
Incorrect CTG	0	0	0	0	1
Incorrect other	0	0	0	0	0
Correct codon	14933	1321	847	0	0

TABLE 2.11: Start codon substitution table for genes which were misreported on the 5′ prime end by Prodigal, combined for the six model organisms. Column headers represent Ensembl annotated start codons and row headers represent the incorrectly predicted start codons, having chosen an alternative further upstream or downstream of the true start codon. The last row, 'Correct codon', shows the numbers of Perfect Match CDSs by Prodigal with the specified start codons. Further start codons with low usage were combined into the category labelled 'other'.

Model Organism	CEA CDSs	Best Tool	Best Tool Detected **[PM]**	Agg' Detected **[PM]**	Best Tool ORFs	Agg' Extra ORFs **[PI]**
B. subtilis	4,011	MetaGeneAnnot'	99.85% **[1.40%]**	100% **[0.37%]**	4,058	1,692 **[41.09%]**
C. crescentus	3,737	MetaGeneMark	92.83% **[31.62%]**	93.66% **[23.17%]**	3,770	1,304 **[34.59%]**
E. coli	4,052	Prodigal	98.05% **[5.94%]**	98.82% **[1.57%]**	4,253	1,635 **[38.44%]**
M. genitalium	476	Prodigal	39.92% **[32.63%]**	40.13% **[30.89%]**	995	426 **[42.81%]**
P. fluorescens	5,178	GeneMarkS	99.29% **[12.97%]**	99.92% **[3.05%]**	5,513	1,891 **[34.03%]**
S. aureus	2,478	GeneMark.hmm (*S. aureus* model)	99.60% **[4.58%]**	99.84% **[0.28%]**	2,582	774 **[29.98%]**

TABLE 2.12: Aggregated tool predictions provide a small increase in Percentage of Genes Detected (M1) but over-predict a large number of additional CDSs. Here we compare the 'best tool' (tool with highest M1 score) predictions versus 'aggregated tools' (combination of predictions from top 5 ranked tools; Prodigal, GeneMark-S-2, MetaGeneAnnot[ator], MetaGeneMark and GeneMark-S) for the percentage of detected genes, partial matches (**[PM]**) and additional over-predictions (percentage increase **[PI]**) made by the aggregated tools which did not detect a Current Ensembl Annotated (CEA) gene. GeneMark.hmm results are reported for *S. aureus* as even though it performed joint best with GeneMarkS (M1), it reported a higher proportion of Perfect Matches (M5).

2.4.7 Aggregated Tool Predictions

Combined prediction approaches have previously been utilised to harness the prediction power from multiple tools to increase the number of detected CEA genes (Yok and Rosen, 2011; Tatusova et al., 2016). For each of the model organisms, taking the union of the top 5 tool predictions did provide a small increase in the number of Genes Detected (M1) (and a reduction of partial matches) compared to that of the 'best tool' (tool with highest percentage of Genes Detected (M1)) for any particular organism. However, even with this extreme case of using the union of all predicted CDSs, the increase in M1 was negligible (average increase of 0.47%) and came at the expense of predicting a large number of additional incorrect CDSs, as can be seen in Table 2.12. Even in the case of *M. genitalium*, the M1 was not improved more than 0.21% with the union prediction.

2.4.8 Improving Historic Annotations

Using the GFF_Adder tool, we investigated the the potential of Prodigal to add additional CDSs to the CEA annotations. There are more than 60 additional predicted CDSs that can be found for each of our model organisms, and more than 270 for *E. coli* and *P. fluorescens* (see Table 2.13).

Model Organism	Ensembl Genes	Additional Prodigal CDSs
B. subtilis	4,011	62
C. crescentus	3,737	64
E. coli	4,052	270
M. genitalium	476	70
P. fluorescens	5,178	293
S. aureus	2,478	74

TABLE 2.13: Numbers of additional CDSs predicted by Prodigal that can be added to Ensembl gene annotations. Additional CDSs are chosen if there are no fewer than 50 nucleotides overlapping with an Ensembl gene.

2.5 Discussion

2.5.1 *Ab initio* Tools Usually Perform Better than Model-Based

We found that *ab initio* tools usually perform better than model-based tools. While no one tool performed the best or worst across all metrics, the *ab initio* tools Prodigal, GeneMarkS-2, MetaGeneAnnotator, MetaGeneMark and GeneMarkS were ranked first to fifth respectively, across our 12 metric ranking (see Figures 2.6, Supplementary Results 2 and Rankings - https://github.com/NickJD/ORForise/tree/master/Supplementary_Data).

Strains of the same species can exhibit large intraspecies variation (Van Rossum et al., 2020). Additionally, genes resulting from horizontal transfer, which is more frequent within species (Van Rossum et al., 2020), are likely to contain features from the donor strains which the rigid model-based methods are unable to recognise. GeneMark, a model-based tool, published in 1993, even when both target genome and model were *E. coli*, was identified as one of the worst performing tools in this study, possibly driven by the well-known large open pangenome of this species (Lukjancenko, Wassenaar, and Ussery, 2010). The same was observed for *S. aureus*. While model-based tools can perform well even when their model and target genomes are different, in the case of Augustus, when applied to the *C. crescentus* genome using the *S. aureus* model, it was only able to detect 3.93% of CEA genes, but unexpectedly detected 78.75% when using the *H. sapiens* model. Unsurprisingly, model-based predictors have therefore fallen out of development and use over the last decade and *ab-initio* based tools such as Prodigal, GeneMarkS-2 and GLIMMER3 have become ubiquitous.

2.5.2 Codon Usage has a Large Influence on Accuracy

We found that codon usage has a large influence on accuracy due to its influence on start and stop codon choice, even in model organisms.

The recoding of a stop codon as an amino acid is rare and seems to be taxa specific (Dybvig and Voelker, 1996). While many of the tools offered the ability to change codon tables (often accounting for TGA specifically), the correct codon tables or codon preferences for each genome cannot be known in advance of annotation of a novel organism. Despite this, we would expect that they should be able to predict a significant proportion of genes, even in the absence of the knowledge of a different codon usage table. Some tools such as Prodigal will assess a genome using both the universal and *Mycoplasma* translation table, however remarkably this did not increase the accuracy of the tool when analysing *M. genitalium* genome (see Figure 2.5). Overall TGA was never predicted as tryptophan-coding in this genome by any tool (see Supplementary Results https://github.com/NickJD/ORForise/tree/master/Supplementary_Data).

While ATG is used for 80% of start codons in the canonical annotations for most prokaryote genomes, some species and even some species-spanning gene families have been shown to use very different start codon profiles (Villegas and Kropinski, 2008). The use of different start codons in prokaryote genomes has often been correlated to the genome-wide GC content: at extreme low and high GC ($< 30\%$ and $> 80\%$), ATG and GTG respectively are often more prominent. In our study the extreme example of this was *C. crescentus* which uses ATG as a start codon only 69% of the time. This is likely driven by its GC content of 67%. All of the tools performed poorly at predicting the correct start codon in this species (Figure 2.5). This has been reported in the literature, specifically in relation to the lack of translation initiation sequence motifs traditionally used by prediction tools to identify the frame and start locus of a gene (Schrader et al., 2014). This is not unique to *C. crescentus* and as shown in Table 2.11, for all 6 model organisms incorrect start codon selection resulted in either elongated or truncated coding sequences (see Supplementary Results 2 https://github.com/NickJD/ORForise/tree/master/Supplementary_Data). The analysis of *E. coli* exhibited the lowest divergence between CEA and predicted start codon selection, possibly as a result of its historic use as a model organism and having the largest use of the canonical ATG start codon in this study. Studies continue to investigate the possible fluidity of gene start codon selection and how some genes recorded in genomic databases may either have been annotated with the wrong start codon, or even require the annotation of multiple alternative start positions and therefore start codons (Villegas and Kropinski, 2008; Meydan et al., 2019; Baranov, Atkins, and Yordanova, 2015).

2.5.3 Metagenomic Annotation Approaches are Suitable for Whole Genome Sequences

Interestingly, tools made specifically for metagenomic and fragmented genome annotation performed better than most single genome tools (tools ranked 3rd, 4th and 6th were developed for metagenome annotation), possibly indicating that even 'complete' genomes may themselves still harbour elements of sequencing and assembly error which these types of algorithms have been designed to account for. Most genomes submitted to databases such as the NCBI Genome repository (Haft et al., 2018) are incomplete and can contain hundreds of fragments which can make gene prediction an even more difficult task. As S. Salzberg said in 2019 "Paradoxically, the incredibly rapid improvements in genome sequencing technology have made genome annotation less, not more, accurate." (Salzberg, 2019). This indicates that future annotation work performed on non-model and more diverse organisms may benefit from approaches implemented by metagenomic tools.

2.5.4 Short Genes and Overlapping Genes are Often Misreported

We found that short genes and overlapping genes are often misreported and that many tools still have hard-coded limitations and weightings against these types of genes, with model-based tools performing especially poorly.

It has been well-established in the literature that short genes are likely under-represented across genomic databases, and therefore, possibly even within the Ensembl data used in this study (Storz, Wolf, and Ramamurthi, 2014; Duval and Cossart, 2017; Su et al., 2013). The growing acceptance that short genes are not only common in prokaryotic genomes but also have important roles (Andrews and Rothnagel, 2014), is at odds with many tools still containing hard-coded limitations for minimum CDS length and algorithmic weights against short CDSs. As might be expected because of its re-coding of TAG, *M. genitalium* proved challenging for all tools to accurately identify CDSs, resulting in the early truncation of a large proportion of CEA genes and an increase in predicted short CDSs. This often led to the tools predicting additional spurious short CDSs in the missed regions (a result that can be seen in the low M10 Precision metric for this genome). However, for the other genomes, most tools also predicted too many short CDSs (9.07% and 39.10%, for *ab initio* and model-based tools respectively), but paradoxically still managed to miss a large proportion of Short CEA genes in the Ensembl annotations (missing 26.38% and 53.69% for *ab initio* and model-based tools respectively) (see Tables 2.8, 2.9 and 2.10.

For overlapping genes, while *ab initio* tools performed better than model-based tools (see Tables 2.6 2.7), in general they both under-predicted the number of overlapping CEA genes across the genomes (on average -6.07% and -30.15% for *ab initio* and model-based tools respectively). No tool was able to correctly detect more than 20% of *M. genitalium's* overlapping CEA genes. Overlapping and nested genes have now become an area of renewed interest for their potential impact on genomic organisation and evolution (Huvet and Stumpf, 2014; Krakauer, 2000). For example, mokC in *E. coli*, believed to be a regulatory peptide, completely overlaps hokC and enables hokC expression (Pedersen and Gerdes, 1999) and no tool was able to detect both genes correctly.

Overall, the tools struggled to handle overlapping gene loci, and often returned either only one or neither of the overlapping coding regions in their predictions. This may be due to the manner in which many tools filter multiple candidate ORFs for a single locus leading to sub-optimal predictions. For example, Prodigal reports a CDS in *C. crescentus* on the positive strand at 23,760-24,074 when the CEA CDS is 23,550-24,170 on the negative strand. The unallocated space (24,074-24,170) resulted in Prodigal reporting the next downstream CDS starting at 24,091, instead of 24,133 (as in the Ensembl annotation), erroneously including 5′ UTR in the predicted CDS.

There are now tools to identify putative short ORFs in both prokaryotes and eukaryotes using additional evidence such as RNA expression data (Bartholomäus et al., 2021; Ji, Cui, and Cui, 2020; Miravet-Verde et al., 2019). Our results suggest that the identification of short and overlapping CDSs can not be done independently without the potential for unforeseen consequences for annotation accuracy.

2.5.5 Species-Specific Misprediction

Throughout this work it has been clear that while specific types of genes, overlapping, short and those with alternative codon usage are still difficult to work with, for even the best *ab initio* tools, prokaryote CDS prediction is efficient and accurate. However, species-specific genomic features were shown to be a continued problems in this study. In particular, for *M. genitalium*, none of the tools studied were able to detect more than 39.92% of the genes from the annotations using the default parameters. Even with the aggregated tool prediction, it was only 40.13%. GeneMark using the *E. coli* model performed extremely poorly and reported only 3 ORFs, none of which correctly detected an CEA gene. These inaccuracies seem to have been caused by a number of factors, specifically relating to differences of the *M. genitalium* genome and the difficulties inherent in identifying *M. genitaium* genes has been reported two decades previously (Devos and Valencia, 2001; Brenner, 1999). *M. genitalium* has one of the smallest known genomes of a free-living bacterium at approximately 500-600KBs with 400-500 genes. It is also one of the few bacteria not to use TGA as a Stop codon but instead to employ it to encode for "tryptophan at a frequency 10 times greater than TGG, the usual codon for this amino acid" (Dybvig and Voelker, 1996). The Ensembl *M. genitalium* genome and annotation contained only 476 PCGs and exhibited a median gene length of 885.5 nt, which was the longest in this study. Although it had a 'normal' minimum gene length of 113 nt, its maximum gene length was by far the shortest at 5417 nt. The combined median gene length that the tools predicted for *M. genitalium* was 487.3 and only one tool came close with 732.79 nt (EasyGene with *E. coli* Model). This reduction of median gene length was due to the tryptophan coding TGA codon being mistaken as a stop codon by the prediction tools, thus truncating the CEA gene length. The perfect match percentage (M5) was also comparatively low at around 60% for the few genes which were detected by the tools. The complete reuse of a specific stop codon across an entire genome is rare and still seems to be taxa specific.

The detection of the correct start position of a gene is seemingly a much more complex process which was shown to be difficult across all organisms studied. As above mentioned, *C. crescentus* in particular only used ATG 68.58% of the time and was shown in the partial hit analysis to be a major cause of inaccurate start position identification. As with *M. genitalium*, this has been reported in the literature, specifically in relation to its lack of motifs traditionally used by prediction tools to identify the frame and start locus of a gene. "Surprisingly, 75.4% of protein-coding

genes lack a canonical translation initiation sequence motif (the Shine-Dalgarno site) which hybridizes to the 3' end of the ribosomal RNA, allowing translation initiation." (Schrader et al., 2014).

Perhaps interestlying, *C. crescentus* and *M. genitalium* represent model organisms which, while having been much studied themselves, exhibit genomic features that all tools studied struggled to account for. Even *ab initio* tools are unable to detect this recoding without user input. Our results have shown that the detection of the correct start codon and specific instance thereof, is a complex process and a problem across all tools and genomes studied, in particular for *C. crescentus*, which exhibited high divergence in start codon selection between tools and the CEA annotations.

2.5.6 Using An Eukaryotic Model for Prokaryote Genome Annotation

CDS gene prediction was undertaken with the Augustus *H. sapiens* model in an exploratory effort to decipher a number of different aspects of gene prediction which may be universal across both prokaryotic and eukaryotic genomes. As with other eukaryote predictors or those which employ eukaryote models, the recognition of eukaryote specific motifs such as introns or splice sites for each of the predicted genes, are integral steps to the algorithms employed. In the prediction results, each gene was reported with potentially multiple CDS' due to individual intron and exon prediction. We chose to treat each CDS region individually as a separate Predicted CDS. While Augustus using the *H. sapiens* model was the worst performing tool when ranked across all 6 model organisms, it performed noticeably better in some situations compared to when it used either the *E. coli* or *S. aureus* model and compared against some other tools in a number of instances. Percentage Difference of Median Predicted CDS Length (M4) is an important metric for which the Augustus *H. sapiens* model performed better than many other prokaryote-specific prediction tools. This may indicate that there are a number of underpinning rules shared by eukaryotic and prokaryotic gene-makeup which could be harnessed for future cross-species prediction work, such as in complex metagenomic studies.

2.5.7 Historic Bias Affects Gene Prediction Today

Overall we have observed an increase in accuracy in tools over time as can be seen with the different versions of GeneMark compared here: the overall rankings of model-based GeneMark (1993) (with *E. coli/S. aureus* models), *ab initio* GeneMarkS (2001) and *ab initio* GeneMarkS-2 (2018) are 20/17, 5 and 2 respectively. However, GeneMarkS (2001) performed better than its successor GeneMarkS-2 (2018) for 5 out of the 12 metrics in Figure 2.6 including Percentage of Genes Detected (M1) in *P. fluorescens*, *M. genitalium* and *B. subtilis* (see Supplementary Results 1 and Rankings `https://github.com/NickJD/ORForise/tree/master/Supplementary_Data`). The performance of GeneMarkS (2001) in M1 may reflect its use for an extended period of time in the NCBI Prokaryote Annotation Pipeline. Possibly as a result of this, many

of the CEA genes GeneMarkS (2001) detected, were originally identified by the tool itself. Similarly, all model-based tools performed at their best across the 12 metrics and 6 model organisms when using their *E. coli* model, hinting at the impact of historical research in this organism. Advances in the realms of machine learning and statistical modelling have the greatest potential to address these issues but are also likely to be the most prone to historical biases in the databases. Many of the rules, such as standard CDS length and codon usage, are inferred from previously identified CDSs. The existence of annotation errors and omissions in various sequence databases is well established and unlikely to be resolved in the near future without significant coordination between repositories (Klimke et al., 2011). Additionally, much of the sequence information derived from model organisms will become less relevant as greater numbers of novel organisms are sequenced (Hunter, 2008b).

These issues have been raised previously: In 2009, the "Best Practices in Genome Annotation" meeting report listed a number of areas of concern put forward by attendees (Madupu et al., 2010) including tool choice, strategy to update and correct previous annotations, tracking of changes in databases, prioritisation of certain genes for experimental evaluation, documenting processes and keeping up with technological advances. The work presented here addresses the issue of tool choice, but many of the recommendations are yet to be realised. The lack of any previous detailed systematic overview of method performance may also have played a part in these biases not being addressed to date. Our study has shown that tool selection needs to be fully informed by its intended purpose and the tool's weaknesses.

2.5.8 Current and Future Techniques are Needed to Continue Annotation Improvements

It is clear from both this and previous studies that combinatory approaches are fundamental in bridging the gap to the next stage of genome annotation. This has clearly already begun with pipelines such as PROKKA and PGAP which utilise a collection of techniques, most notably, advanced homology searching to complete annotations where traditional CDS predictions fail or produce competing predictions. However, this can also lead to conflicting annotations. As noted, homology searches are only as good as the database being used. The presence or absence of homology does not indicate whether an ORF is a true CDS gene, especially in the nuanced field of alternative ORFs Orr et al., 2020. Further complications involving alternative ORFs, many of which are overlapping, can be found with new evidence in *E. coli*, where "Ribosome profiling revealed out-of-frame internal minimal ORFs in 13 *E. coli* genes. Mutation of the start codon... in one gene, yecJ, resulted in an increase in translation of the main-ORF, suggesting that these minimal ORFs also can modulate translation of the main-ORF" (Meydan et al., 2019).

As users of computational genomic techniques, we must realise when we have reached the limit of what is possible with the contemporary data available. This,

together with other studies, proposes that the linchpin required for the next step in genome annotation, is not even more techniques reliant on current genomic knowledge, but instead more experimental work and species agnostic approaches. However, the near unlimited scope of growth conditions, environmental pressures et cetera, have made the prospect of experimentally validating all potential CDS regions unfeasible. Finally, while great strides have been made in experimentally validating difficult to characterise gene-types, one such study "... suggest[s] substantial numbers of small proteins remain undiscovered in *E. coli*, and existing bioinformatics techniques must continue to improve to facilitate identification." (VanOrsdel et al., 2018).

2.5.9 Conclusion

We have presented a comprehensive set of metrics which distinguish CDS prediction tools from each other and make it possible to identify which performs better for specific use-cases. The ORForise evaluation framework enables users to evaluate new and existing annotations and generate consensus and aggregate gene predictions. We have demonstrated that certain types of genes, such as short genes, overlapping genes and those with alternative codon usage, are still elusive, even to the most advanced *ab initio* techniques. Worryingly, the performance of any tool seems to depend on the genome that is being analysed. For instance, Prodigal which ranked best overall, was ranked first for *E. coli*, *S. aureus* and *P. fluorescens*, MetaGeneAnnotater was ranked first for *B. subtilis* and *M. genitalium* and GeneMarkS-2 was ranked first for *C. crescentus* (see Supplementary Rankings). Additionally, no individual tool ranked as the most accurate across all genomes for the Percentage of Genes Detected (M1) (the single metric historically used to assess tool performance) or any other individual metric. This is likely to have a measurable impact on downstream genomic and pangenomic studies. However, overall we found Prodigal to be one of the most well-rounded tools, not only detecting the highest number of CEA genes for two very diverse model organisms (*E. coli* and *M. genitalium*), but also performing overall best when ranked across the 12 metric rankings and 6 model organisms. It was also overall best for Perfect Matched genes (M5). However as outlined earlier, it was not always ranked first for all genomes, further suggesting that users should choose tools carefully, based on the organism and question they are studying. Finally, we advise against generating aggregated *ab initio* annotations from multiple tools where no existing annotation is available for the genome, as this results in poor overall performance. However, additional cycles of annotation with tools designed to identify putative CDSs in the unannotated regions, show promise for improving current prokaryotic genomic knowledge.

Chapter 3

StORF-Reporter: Finding Genes between Genes

3.1 Chapter Summary

To overcome some of the principal limitations and biases identified in the previous chapter, such as start codon selection and gene overlap, this chapter presents an additional step to be performed on the results of an existing genome annotation. I specifically investigated the intergenic regions (IRs) of prokaryote genomes. These IRs are often a misnomer, as this simply means that we have not yet found genes in these regions according to the canonical annotations. I henceforth call these regions Unannotated Regions (URs) to avoid this issue. The methodology I present here, StORF-Reporter, consists of two parts. The first part begins with the extraction of URs of an annotated genome. Next, Stop-ORFs (StORFs) are identified in these URs. StORFs are Open Reading Frames (ORFs) that are delimited by stop codons. Through this process, I have found that StORFs can return complete coding sequences (with/without homology to known genes), missed by canonical and novel genome annotations.

Throughout this chapter, I present my findings, which include that a selection of 6 model organisms all exhibit StORFs that capture canonical CDS genes missed by leading CDS prediction tools. Additionally, StORFs recovered a number of sequences in the URs of the Ensembl-annotated genomes which exhibited high levels of sequence identity in both the SwissProt protein database and the proteome of the individual Ensembl genomes. Inspecting the pangenome of *Escherichia coli (E. coli)* revealed that StORF-Reporter was able to extend the core, soft-core and accessory gene collections. There were also a number of StORF-Only gene clusters identified. I further applied this methodology to 6,223 prokaryotic genomes from the Ensembl Bacteria database. StORF-Reporter was able to identify a number of StORFs across these genomes to produce gene families which were extended into additional genera, not previously identified in the Ensembl annotations.

StORF-Reporter is available at https://github.com/NickJD/StORF-Reporter.

3.2 Introduction

Prokaryotic genomes are often observed with high levels of genome density, with the compact nature of their genes, both coding and non-coding, resulting in little extraneous DNA (Sela, Wolf, and Koonin, 2016). Much of this extraneous DNA is often referred to as intergenic regions (IRs), islands of DNA between functional elements such as genes, promoter regions or other structurally important elements. In eukaryota studies, relative IR sizes compared to a genome as a whole have been associated with the biological complexity of the organism (Taft, Pheasant, and Mattick, 2007). However, in prokaryotes, because of the much reduced size of their IRs, they are often assumed not to be functionally important and thus are little studied. Most studies into prokaryote IRs involve finding putative coding and non-coding genes (both complete and pseudogenised) (Sridhar et al., 2011). Therefore, in this work, the term unannotated region (UR) will be used to describe the regions of a genome that have yet to be characterised. Although these are still most often referred to as IRs, this work specifically investigates their potential to harbour yet-to-be annotated coding genes. Prokaryotic URs are still often assumed to not contain functional elements, even when shown to exhibit high levels of homology to important protein-coding and non-coding genes often missed by contemporary annotation methods (Hemm et al., 2008; Tsai et al., 2015). While there has been some analysis in particular of prokaryote URs and what they may contain, there is a distinct paucity of such work and it is rarely built upon or developed in future studies (Notari et al., 2014; Sridhar et al., 2011). Those studies which have investigated the conservation and selection of putative UR sites have shown that there is "consistent evidence of strong purifying selection on IGRs (URs)" (Thorpe et al., 2017). These regions should now be investigated to determine whether they may contain genes missed by contemporary prediction methods. Their potential to be reservoirs for historic genes, those pseudogenised due to mutation should also be investigated.

While studies have shown that URs warrant further investigation, the methods employed to study them still rely heavily on homology alignments of the entire UR sequence to known proteins deposited in databases. This practice is inefficient and more importantly inaccurate, as the part of UR which may exhibit homology to a known protein may be overshadowed by the surrounding UR DNA 'noise' and requires the homologous sequence to have been previously identified. Furthermore, homology searching to previously identified genes is unlikely to work if there is a systematic reason behind such genes being missed by contemporary genome annotation, and therefore they are unlikely to have representatives in such databases. Conventional protocol suggests that once a genome has been annotated by one of the many available tools, such annotation is complete. This observation has unfortunately led to the misunderstanding and false perception that URs are simply regions of DNA without function, are sparse, and do not require further investigation. While it is now becoming the norm that once cryptic genes are being annotated by

additional steps such as advanced homology searching and RNA expression analysis, the assumed streamlined nature of prokaryotic genomes is at odds with the amount of UR DNA.

In response to the weaknesses and biases in genome annotation, many of which have been outlined in Chapter 2, I have developed an approach that extracts Stop-ORFs (StORFs), ORFs (Open Reading Frames) that are delimited by stop codons from putative URs (see Figure 3.1). Examples of StORFs, extracted from a previously annotated genome, can be seen in Figure 3.2, potentially capturing multiple possible start codons.

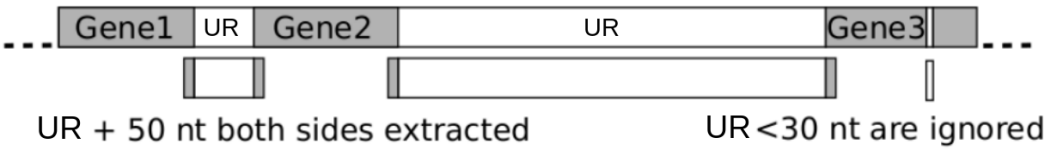


FIGURE 3.1: Visual representation of how Unannotated Regions (URs) are selected for extraction. URs that are less than 30 nt are not extracted. URs are extracted with additional 50 nt on their 5′ and 3′ ends to allow for overlapping genes.

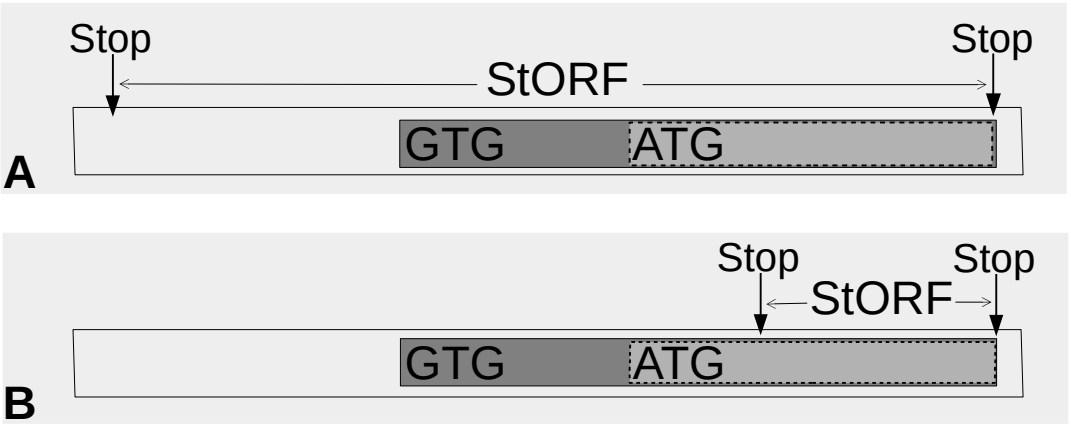


FIGURE 3.2: Visual representation of a StORF and how it can capture multiple potential start codons for a single gene in an unannotated region. Image A depicts a StORF capturing the two possible start positions/codons for a CDS gene and image B shows how a StORF can comprise of only a partial segment of a gene if that gene either recodes a canonical stop codon or has had an in-frame stop codon mutation.

This 'StORF-Reporter' approach consists of two parts, firstly the extraction of URs from an annotated genome, and secondly the identification and filtering of StORFs from those extracted URs.

The problem of multiple definitions of what an ORF is has been brought up recently (Sieber, Platzer, and Schuster, 2018) with the positives and negatives of the different definitions discussed. Interestingly, while most prokaryotic annotation tools will annotate from start codon to stop codon, the Sequence Ontology (Eilbeck

et al., 2005) defines an ORF as "The in-frame interval between the stop codons of a reading frame which when read as sequential triplets, has the potential of encoding a sequential string of amino acids.". StORFs therefore use the same interpretation as an ORF as the Sequence Ontology but for clarity and due to their specific use case, are named StORFs hereafter.

The reporting of stop codons, instead of predicting a specific start codon, removes a number of complexities, such as the Shine–Dalgarno motif (Dalgarno and Shine, 1973), promoter regions (Browning and Busby, 2004), ribosomal binding sites, and operons (Dandekar et al., 1998). Additionally, out of frame stop codons have been observed as a widespread phenomenon among prokaryotes and theorised to carry functional significance of being selected in the course of genome evolution to act against unintended frameshift (Tse et al., 2010). Thus, the existence of a gene-like length StORF is unlikely in the out of frame region of another, unreported gene. While the canonical 'ATG' codon is most-often used to initiate approximately 80% of prokaryotic gene CoDing Sequences (CDSs), many species and cross-species gene families have been shown to use very different start codon profiles (Villegas and Kropinski, 2008). Current prokaryotic annotation tools still struggle to determine the correct start codon, resulting in incomplete or missed gene detection (see Chapter 2 (Dimonaco et al., 2021)). StORFs therefore, can also negate the possibility that an incorrect or alternative start codon selection may truncate the start of a protein product, as the entire coding potential of the CDS is encapsulated by the StORF. Furthermore, there is growing evidence of the existence of "organisms that adjust their genetic code in response to changing environments", thus making traditional start codon identification ever more complex (Baranov, Atkins, and Yordanova, 2015). Although stop codon usage in prokaryote CDS genes has also been linked to a number of genomic features such as GC content, there is still much we do not yet know and many theories concerning their selection are still being posited by researchers (Belinky et al., 2018; Povolotskaya et al., 2012). However, unlike start codons, the three canonical ("TGA,TAG,TAA") stop codons are the terminus for almost all CDS genes across almost all species, meaning that identification is based solely on their presence.

This chapter proposes the StORF-Reporter methodology as an additional step to be performed on the output of contemporary prokaryotic genome annotation. Firstly, I look at six bacterial model organisms which exhibit many URs with median lengths long enough to contain interesting discoveries that can expand the narrative of their genome gene collection. Using StORF-Reporter, I was able to recover gene CDSs present in the model organism canonical genomic annotations provided by Ensembl Bacteria (Howe et al., 2020) but which were missed by Prodigal (Hyatt et al., 2010), which as the results of Chapter 2 concluded, is one of the best and well rounded CDS gene prediction tool. Additionally, as Prodigal is not part of the GeneMark family, which has been responsible for over two decades of genomic

annotations, another reason for its use in this study is to use a leading tool with a level of independence from previous annotations.

StORFs were also discovered in the URs of the Ensembl annotations, and many exhibited high levels of sequences identity to proteins in the Swiss-Prot database (Bateman et al., 2020) and the proteome of the genomes they were found in (possible duplicate genes), both of which support the theory that current annotations (including Ensembl) are incomplete. The combination of these StORFs with the canonical annotations from Ensembl can be referred to as 'GFF+' and represent a refining of current annotations as opposed to the traditional mantra of reannotation.

As discussed in the Background of this thesis, pangenomes, which consist of continuously expanding and contracting species-wide gene sets, dynamically in response to exogenous pressures, are studied for a number of important reasons. Therefore, I next extracted StORFs from a collection of *Escherichia coli (E. coli)* genomes and found StORFs which were not only part of core gene families, but also redefined a number of gene families through the inclusion of additional strains due to identification of genes which were missing from their Ensembl annotations. Lastly, the same methodology was applied to 6,223 prokaryotic genomes from Ensembl Bacteria and StORFs were found across different taxa in both StORF-Only gene families and as additional sequences in Ensembl annotated gene families.

3.3 Methods

3.3.1 Data Preparation

The canonical annotation and sequence data for 44,048 genomes of 8,244 prokaryotic species, which included 43,347 bacterial genomes and 493 archaeal genomes, were acquired from the 46th release of Ensembl Bacteria (Howe et al., 2020). For each genome, two data files were downloaded; the complete DNA sequence (∗*_dna.toplevel.fa*) and the GFF (Generic Feature Format) file (∗*.gff3*) containing the position information for each gene (coding and non-coding). As with the previous chapter, the genomic elements (including both coding and non-coding genes) presented in the Ensembl Bacteria GFF annotations were taken as Current Ensembl Annotation (CEA) and as reference annotations for this study. Fragmented genomes were removed by filtering out those with more than 5 contigs (not including plasmids). Next, genera which had less than 5 genomes were removed. This resulted in 6,223 genomes (see Table 3.1). Out of this 6,223, the six model organisms; *Bacillus subtilis (B. subtilis)* BEST7003 strain, *Caulobacter crescentus (C. crescentus)* CB15 strain, *Escherichia coli (E. coli)* K-12 ER3413 strain, *Mycoplasma genitalium (M. genitalium)* G37 strain, *Pseudomonas fluorescens (P. fluorescens)* UK4 strain, *Staphylococcus aureus (S. aureus)* 502A strain, shown in table 3.2, and used in the previous chapter, were again selected for this study.

Acetobacter	14	Clostridioides	5	Methanobacterium	6	Salmonella	235
Achromobacter	8	Clostridium	79	Methanobrevibacter	6	Selenomonas	7
Acidithiobacillus	5	Collimonas	6	Methanocaldococcus	6	Serratia	33
Acidovorax	6	Corynebacterium	110	Methanococcus	7	Shewanella	26
Acinetobacter	83	Coxiella	10	Methanosarcina	26	Shigella	20
Actinobacillus	8	Cronobacter	10	Methylobacterium	11	Sinorhizobium	10
Actinomyces	12	Cupriavidus	6	Microbacterium	13	Sphingobium	9
Aeromicrobium	7	Dehalococcoides	13	Micromonospora	5	Sphingomonas	9
Aeromonas	26	Deinococcus	11	Moraxella	8	Sphingopyxis	9
Aggregatibacter	8	Desulfitobacterium	5	Mycobacterium	733	Spiroplasma	17
Agrobacterium	8	Desulfotomaculum	6	Mycoplasma	95	Staphylococcus	329
Alcanivorax	5	Desulfovibrio	19	Myroides	7	Stenotrophomonas	14
Altererythrobacter	6	Dickeya	6	Neisseria	36	Streptococcus	343
Alteromonas	19	Edwardsiella	10	Nitrosomonas	6	Streptomyces	82
Amycolatopsis	9	Ehrlichia	15	Nocardia	5	Sulfolobus	21
Anaplasma	19	Enterobacter	82	Nostoc	7	Synechococcus	25
Arcobacter	5	Enterococcus	83	Oenococcus	5	Synechocystis	7
Arthrobacter	12	Erwinia	8	Paenibacillus	45	Thermoanaerobacter	10
Azospirillum	6	Erythrobacter	5	Pandoraea	11	Thermococcus	18
Bacillus	283	Escherichia	225	Pantoea	14	Thermotoga	13
Bacteroides	17	Eubacterium	12	Paraburkholderia	9	Thermus	8
Bartonella	36	Flavobacterium	16	Pasteurella	12	Thioalkalivibrio	5
Bdellovibrio	5	Francisella	59	Pectobacterium	10	Treponema	40
Bibersteinia	5	Frankia	5	Pediococcus	6	Ureaplasma	10
Bifidobacterium	79	Fusobacterium	15	Planococcus	9	Veillonella	5
Bordetella	29	Gardnerella	6	Porphyromonas	7	Vibrio	67
Borrelia	28	Geobacillus	18	Prevotella	14	Weissella	5
Borreliella	6	Geobacter	11	Prochlorococcus	16	Wolbachia	8
Brachyspira	9	Haemophilus	39	Propionibacterium	26	Xanthomonas	57
Bradyrhizobium	14	Halomonas	5	Proteus	9	Xenorhabdus	7
Brucella	164	Helicobacter	108	Providencia	5	Xylella	7
Buchnera	20	Hymenobacter	7	Pseudoalteromonas	8	Yersinia	66
Burkholderia	183	Janthinobacterium	5	Pseudomonas	197	Archaeoglobus	5
Caldicellulosiruptor	8	Klebsiella	149	Psychrobacter	9	Blattabacterium	8
Campylobacter	74	Lactobacillus	108	Pyrobaculum	8	Hydrogenobaculum	5
Candidate	5	Lactococcus	18	Pyrococcus	8	Taylorella	5
Candidatus	176	Legionella	17	Ralstonia	17	Zymomonas	7
Caulobacter	5	Leifsonia	8	Rhizobium	23	Myxococcus	5
Cedecea	6	Leptolyngbya	5	Rhodobacter	6	Hyphomicrobium	5
Cellulophaga	5	Leptospira	16	Rhodococcus	20	Aerococcus	6
Chlamydia	147	Leuconostoc	12	Rhodopseudomonas	7	Cyanothece	6
Chlamydophila	7	Listeria	81	Rickettsia	57	Lysobacter	5
Chlorobium	7	Mannheimia	15	Riemerella	8	Pseudonocardia	5
Chryseobacterium	6	Marinobacter	11	Ruminococcus	9	Clavibacter	5
Citrobacter	16	Mesorhizobium	10	Saccharomonospora	6		

TABLE 3.1: Listed are the 179 Ensembl Bacteria genera with the number of genomes after filtering and which were used in the inter-genera study.

Model Organism	Genome Size (Mbp)	Genes **[Density]**
Bacillus subtilis ***(B. subtilis)*** BEST7003	4.04	4,133 **[88.91%]**
Caulobacter crescentus ***(C. crescentus)*** CB15	4.02	3,875 **[90.60%]**
Escherichia coli ***(E. coli)*** *K-12 ER3413*	4.56	4,257 **[86.28%]**
Mycoplasma genitalium ***(M. genitalium)*** *G37*	0.58	559 **[92.03%]**
Pseudomonas fluorescens ***(P. fluorescens)*** *UK4*	6.06	5,266 **[84.75%]**
Staphylococcus aureus ***(S. aureus)*** *502A*	2.76	2,556 **[83.93%]**

TABLE 3.2: An overview of genome composition for the 6 model organisms selected to evaluate StORF-Reporter compiled from data held by Ensembl Bacteria. Number of Ensembl annotated genes (coding or non-coding) is reported and the genome density is in bold square brackets. Note the relative differences in genome size (0.58 - 6.06 Mbps) and gene density (percentage covered with annotation, 83.93% - 92.03%).

3.3.2 StORF-Reporter

The StORF-Reporter methodology was developed as an additional step to be performed after a 'traditional' genome annotation had been undertaken. StORF-Reporter is comprised of two distinct parts which can be used independently or together as they were in this study: UR_Extractor(.py) and StORF_Finder(.py). Further to this, both tools have been designed to work together using their default parameters. The processes of both are described in the following sections and the code is available at `https://github.com/NickJD/StORF-Reporter`.

3.3.2.1 Unannotated Region – Extractor (UR-Extractor)

To facilitate the extraction of URs from different prokaryotic genomes (and the various interpretations of the GFF format), UR_Extractor was developed. It allows user-defined genomic features (coding and non-coding genomic elements) to be used to identify the boundaries of genomic elements. Written in Python3 (Van Rossum and Drake, 2009) and using the feature coordinates presented in GFF3 format, regions of DNA without annotation were extracted from the provided DNA sequence file. GFF3 and FASTA files with the details of the sequence and loci for each UR were reported for further analysis (see Appendix Subsection B.1 for the command line menu of UR_Extractor.py).

When recovering URs from a genome it is important to consider that unannotated CDSs (to be captured by StORF-Finder) may overlap with annotated genes (coding and non-coding) in either the same or an alternative frame (Sabath, Graur, and Landan, 2008). Therefore, it was necessary to determine the extent to which

the URs should be extended into the annotated regions. The 6,223 genomes from Ensembl Bacteria, including the six model organisms (MOs) in table 3.2 were studied to identify representative parameters for the extraction of URs across this large set of Ensembl Bacteria genomes. The median gene overlap observed across the six genomes was 3 nt and the 75th percentile was below 50 nt, as can be seen in the left half of Figure 3.3. Furthermore, the median distance from a gene's start codon to the nearest in-frame upstream stop codon (representing the untranslated region of a StORF) was 39 nt and in 77.4% of the CDS genes it was less than or equal to 100 nt (see Figure 3.3). The lengths of overlap between genes (both coding and non-coding) across the 6,223 prokaryotic genomes were investigated and as can be seen in Figure 3.4, the gene overlap lengths are similar across the majority of the Ensembl annotated genomes. These findings further illustrate that these model bacterial genomes are relatively compact and demonstrate that StORFs often contain very little extraneous upstream DNA. To account for the gene and StORF features observed, URs were only extracted if they were at least 30 nt long but were extended by 50 nt at both ends (into annotated genes) by the UR Extractor tool (see Figure 3.1), totaling a minimum length of 130 nt. This extension of 50 nt at both the 5-prime and 3-prime ends, was to allow for the capture of the parts of StORFs which may overlap with existing gene annotations. This process took into account the untranslated region between the start of the StORF and putative start codon of the protein sequence it captured and was necessary as the direction of a potential StORF is not known in advance (see Figure 3.2 for an illustration of how a StORF captures upstream DNA).

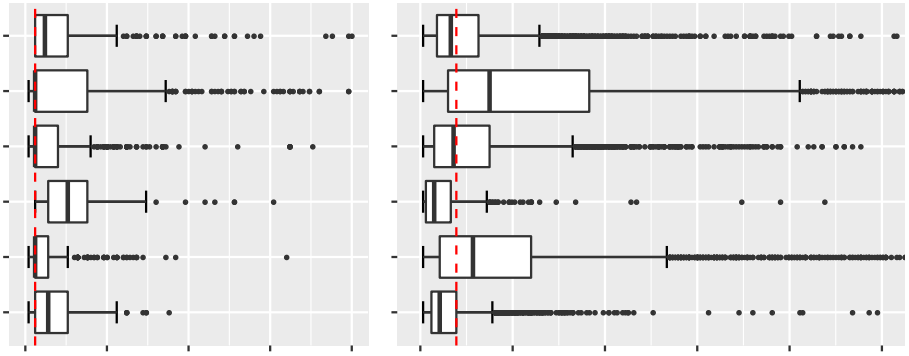


FIGURE 3.3: This double plot reports the analysis of the 6 model organism which was used during the parameterisation of StORF-Reporter. Figure A reports the distributions of the Ensembl gene overlap lengths for each model organism with a dotted red line representing the overall median (3 nt) with the x-axis truncated at 100 nt. Figure B reports the distance between an Ensembl gene's start codon and the first in-frame upstream stop codon for the selected model organisms with the x-axis is truncated at 500 nt. The dotted red line represents the overall median (39 nt). These plots indicate that the extension of 50 nt from each end of the extracted intergenic regions is often enough to capture both the true overlap between an annotated gene and the putative gene identified by a StORF, including the small amount of upstream non-coding DNA which the StORF will contain.

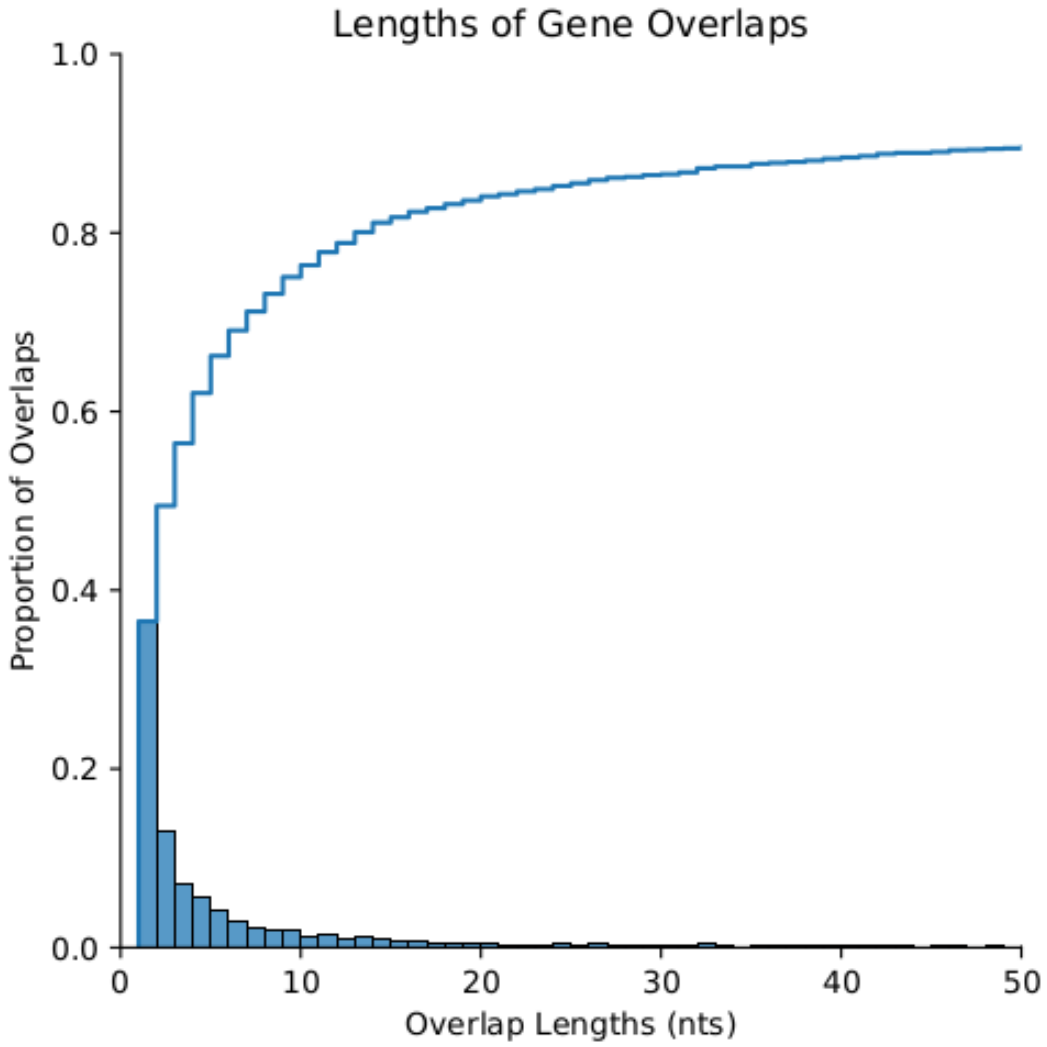
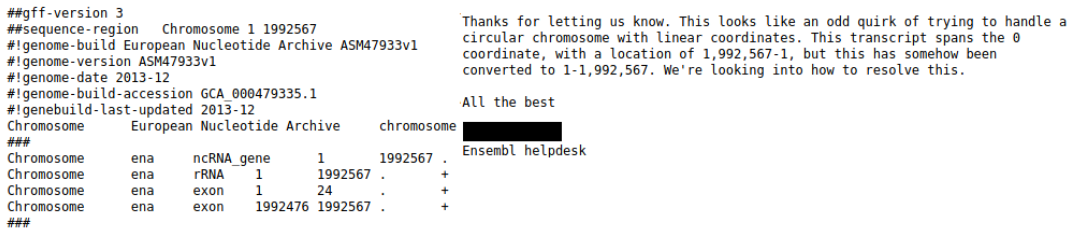


FIGURE 3.4: Shown here are the proportional overlap lengths in nucleotides between all genes (coding and non-coding) from the 6,223 filtered genomes from Ensembl Bacteria. The blue line reports the cumulative proportion of gene overlaps increasing very little after 10-20 nt.

All genomic repositories, including Ensembl Bacteria, are known to contain varying levels of assembly and annotation error due to a number of constraints. One such example can be seen in Figure 3.5, where a rRNA gene has been given the same coordinates as the entire length of the chromosome (genome). This would make the extraction of URs impossible for this genome. It can also be the case that genomes have missing annotations as a result of human or software error, or limitations. Therefore, UR Extractor has a maximum UR length cutoff of 100kb. A genome with a 100 kb UR either warrants further investigation for biological interest or most likely contains annotation error.



```
##gff-version 3
##sequence-region   Chromosome 1 1992567
#!genome-build European Nucleotide Archive ASM47933v1
#!genome-version ASM47933v1
#!genome-date 2013-12
#!genome-build-accession GCA_000479335.1
#!genebuild-last-updated 2013-12
Chromosome      European Nucleotide Archive     chromosome
###
Chromosome      ena     ncRNA_gene      1       1992567 .
Chromosome      ena     rRNA    1       1992567 .       +
Chromosome      ena     exon    1       24      .       +
Chromosome      ena     exon    1992476 1992567 .       +
###
```

```
Thanks for letting us know. This looks like an odd quirk of trying to handle a
circular chromosome with linear coordinates. This transcript spans the 0
coordinate, with a location of 1,992,567-1, but this has somehow been
converted to 1-1,992,567. We're looking into how to resolve this.

All the best

[redacted]
Ensembl helpdesk
```

FIGURE 3.5: This correspondence from the Ensembl Help desk represents an example of annotation error due to the automated and 'hands-off' nature of systematic genome annotation. A rRNA gene has been given the same coordinates as the entire length of the chromosome (genome) as explained in the response from a representative from Ensembl.

3.3.2.2 Stop - Open Reading Frame – Finder (StORF-Finder)

To identify StORFs, StORF-Finder starts by scanning the previously extracted URs (or any DNA sequence) for loci encapsulated by in-frame, user-defined stop codons (see Figure 3.2 for an example of a StORF and Appendix Subsection B.2 for the command line menu of StORF_Finder.py)). By the nature of nucleotide triplet abundance, influenced by GC content and other constraints, the three canonical stop codons; TAG, TGA and TAA, are frequently present throughout both out-of-frame genic and intergenic DNA (Wong et al., 2008). To investigate whether StORF-Finder should itself select which stop codons to use, I performed a comparison of the presence of the triplets with their usage as stop codons across the Ensembl Bacteria genomes. The results of this analysis (see Figure 3.6) show that while there are differences between the three codons, they are not universal and it is likely not beneficial to make generalisations.

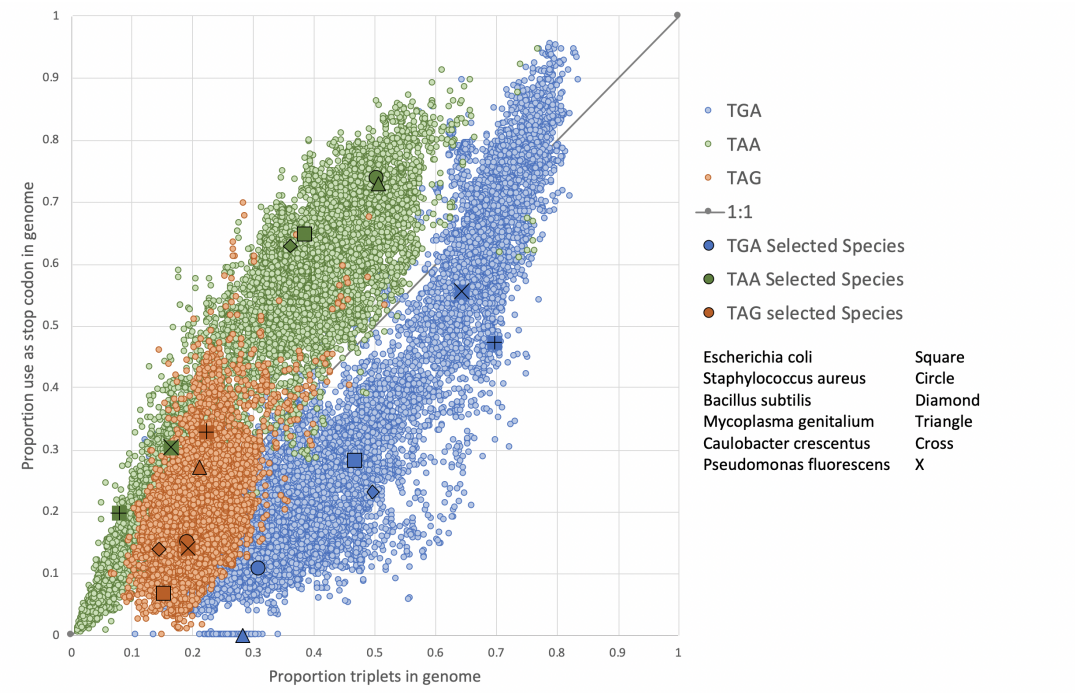


FIGURE 3.6: Graph showing stop codon usage and proportion of triplets in the 44,048 genomes from Ensembl Bacteria. For each of the 6 model organisms, this show the distribution across the entire genome of the three triplets TAA, TAG and TGA in all six reading frames and stop codon usage (i.e. the actual relative usage in Ensembl CDS genes of the three different stop codons).

Additionally, as with start codons, stop codons are differentially preferred by groups of genes and gene families both from within the same and across different genomes (Ivanova et al., 2014). For example, we can see in Figure 3.6 that the codon TAA is used more frequently than its distribution across the genomes suggests. Similarly, TGA is used less frequently than its distribution across the genome would suggest in all 6 model organisms (with TGA never used as a stop codon in *M. genitalium*.

For TAG, the usage is in line with the overall distribution across the genomes. While there is evidence of stop codon preference and even re-coding for amino acids in species such as *M. genitalium*, it can be assumed that for the majority of prokaryotic genomes, the three canonical stop codons can be treated equally.

While being able to effectively circumvent start codon identification, StORFs do have their own drawbacks. As we are not able to identify which codon is the start of transcription, this being ATG or otherwise, when translating an identified StORF to amino acid, the universal codon table is used for all sequences. This is a problem as "The biosynthesis of all proteins from all living cells begins with methionine" (Sherman, Stewart, and Tsunasawa, 1985). Therefore, although an alternative start codon may be used, it is still translated as Methionine in the final amino acid sequence (Lobanov et al., 2010). As a countermeasure against this, StORF-Finder reports both the DNA and amino acid (using the universal genetic code codon table 4) sequences in its output.

As with traditional CDS prediction methods, without any form of filtering, StORF-Finder can report a very high number of overlapping and nested StORFs, even for short regions of DNA. Therefore, a filtering step was developed to reduce the number of StORFs reported. This is done by first ordering all StORFs by their length in descending order and then iteratively removing the nested or shorter StORFs in cases where two StORFs overlap beyond the defined length of 50 nt. This is done individually for each set of StORFs for each UR. This filtration method is applied to all StORFs equally and is intended to prioritise capturing as much of an UR as is possible, while reducing the number of nested and overlapping StORFs. The remaining StORFs are reported in FASTA (DNA and/or amino acid form) and GFF format with both locus coordinates for their original position within the original genome sequence, and their position relative to the UR in which they were reported.

3.3.3 Extracting StORFs from Prodigal and Ensembl Annotations of the Six Model Organisms

To investigate the use of StORF-Reporter to expand contemporary annotations, URs and StORFs were extracted from both the annotations and predictions provided by Ensembl Bacteria and Prodigal (Hyatt et al., 2010) for the 6 model organism genomes listed in Table 3.2. The default parameters of UR_Extractor (-gene_ident "ID=gene") were used to record the complete set of genomic elements (coding and non-coding) which was then used to extract URs from the Ensembl provided annotations. Prodigal was applied to the complete DNA sequences of the six MOs and a novel CDS prediction was performed using the tool's default parameters. As Prodigal only predicted CDSs, UR_Extractor was only able to use the CDS coordinates to extract URs (-gene_ident "CDS").

3.3.4 Extracting StORFs from 6,223 Ensembl Genomes

The StORF-Reporter methodology was applied to the collection of 6,223 filtered Ensembl Bacteria genomes using the Ensembl annotations for the UR extraction. The diversity of assembly and annotation quality of these genomes presented a number of obstacles. Unlike the six model organism genomes, even after filtering, the genomes in this study were often in a fragmented state with a number of low quality regions containing large sections of ambiguous nucleotides. With this in mind, URs and StORFs were extracted individually for each of the 6,223 genomes using their default parameters. The resulting StORF-Finder output FASTA files were combined into a single FASTA file with their original species name appended to the start of each sequence header. For example, the full genome name of 'Enterobacter_cloacae_ecwsu1.asm23997v1' was appended to the start of the protein name 'AEW71445' to make 'Enterobacter_cloacae_ecwsu1.asm23997v1|AEW71445' The custom Python3 script **genome_FASTA_Combiner.py** was used for this.

To investigate the pangenomic spread of StORFs found in the URs across a widely studied prokaryotic species, the URs and StORFs extracted from *E. coli* genomes were studied separately. Many of the Ensembl Bacteria genomes were unlabelled at the species level, therefore to make sure the correct genomes were extracted, only those labelled specifically as *'Escherichia_coli'* were extracted. This resulted in a group of 219 *E. coli* genomes from the original filtered group of 6,223.

3.3.5 Validation of Recovered StORFs

To validate the recovered StORFs from the Prodigal and Ensembl annotations of the six model organisms, both the work of Chapter 2 and homology signatures to known proteins were utilised.

For the StORFs identified between the Prodigal annotations, an extension to the 'ORForise' platform was developed which originally was designed to compare annotations from different CDS prediction tools to a reference annotation and introduced in Chapter 2. This extension, 'StORForise', was used to identify the Ensembl CDS genes missed by Prodigal which StORF-Reporter was able to recover. The default parameters of what classified a gene as 'detected' were taken from the ORForise package (a minimum of 75% coverage of an in-frame CDS prediction for a reference CDS gene). StORF-Reporter was only given the extracted URs according to the Prodigal CDS predictions. Mispredictions by Prodigal result in either elongation of an Ensembl CDS gene prediction, or prediction of a CDS where no Ensembl annotation existed. These would constrain the available regions that could be searched for StORFs. Therefore, as part of the ORForise platform, two additional scripts **'StORF_Undetected.py'** and **'non_vitiated_Missed_Genes.py'** were developed. The first script was used to identify the Ensembl genes which Prodigal missed. The next identifies the number of missed Ensembl genes which were non-vitiated

(not corrupted or contaminated) by the mispredictions of Prodigal (false positives, overlapping Ensembl genes by more than 50 nt).

Although the coding start sites are not identified by StORF-Finder, the coding frame is, and so the upstream region will include the start codon in the same frame (unless mutations are present). This indicates that StORFs can be directly translated into amino acid sequences, and then undergo alignment and identity analysis to find homologous proteins already reported across other genomes and protein databases. The protein sequence aligner Diamond (Buchfink, Xie, and Huson, 2015) (version v0.9.30.131), allows for the creation of an alignment database from any set of protein sequences. This feature was used to align the reported StORFs from both Prodigal and Ensembl annotations to the Ensembl proteomes for the six MOs and to the Swiss-Prot protein sequence database (UniProt Consortium, 2019) (release 2021_01, downloaded 05/03/2021). The sequence alignments were performed with the Diamond blastp option with two separate threshold runs, one with a minimum bitscore of 60 and the second with the same bitscore but also a subject coverage cut-off of >=80%. The same Swiss-Prot database and both parameter searches were used throughout this study. StORFs which show high (>=80%) levels of sequence identity to annotated CDS genes in the Ensembl annotations are likely candidates of gene duplication and therefore should be annotated as such in the canonical annotations.

To investigate the functional profiles of the identified StORFs and how they may differ to the CDS genes from Ensembl, Clusters of Orthologous Groups (COG) functional categories from EggNOG were used. This functional analysis of the representative sequences from each cluster was performed with EggNOG-Mapper version 2.15 (Cantalapiedra et al., 2021) with the latest databases available as downloaded on the 01/08/2021 and with default parameters.

The full 'ORForise' package along with the StORForise and all other additional scripts are available at https://github.com/NickJD/ORForise and https://github.com/NickJD/Bioinformatic_Scripts.

3.3.6 *Escherichia coli* Pangenome Analysis

To investigate the impact that StORFs may have on a well-studied prokaryotic model organism pangenome, the Ensembl Bacteria protein sequences and StORFs extracted from the 219 *E. coli* strains were studied.

The methods by which a pangenome is curated and studied are as varied as the pangenomes themselves. These include, but are not limited to, SNP profiles, gene loci and shared k-mer sizes and gene function (Decano and Downing, 2019; Carlos Guimaraes et al., 2015). In three separate pangenomic studies of *E. coli*, utilising 17, 22 and 7 (including 2 *Shigella* isolates, considered to be *E. coli* strains) genomes, pangenomes consisting of 2,200, 2,800 and 2,865 core genes were reported respectively (Rasko et al., 2008; Fukiya et al., 2004; Chen et al., 2006). In these works, the

descriptions of the methods used to categorise each gene family were brief and in one case, the entire section reporting such methods was only a few lines long, omitting many of the key details. Thus, it is difficult to perform comparative studies on or with pangenomic data. Tools do exist and are widely used for pangenomic study. One such example can be found in Roary (Page et al., 2015). However, these tools use heuristic approaches in building their pangenomes and gene loci information to identify core and paralog genes. Many of the URs which harbor the CDS genes that StORF-Reporter identifies, are unlikely to be in the same location across all genomes. The fragmented state of the majority of genomes held in Ensembl Bacteria (or most genome repositories in general) further adds to the necessity to use alternative techniques to gather the detail of the pangenome.

Amino acid sequence alignment or identity has long been used to group genes of similar function together in gene family analysis, and has been used in this study. CD-Hit (Fu et al., 2012) is a sequence clustering tool which does not require additional information such as loci or genome data and can create gene families based solely on sequence identity. As CD-Hit does not require metadata such as gene loci and function which is often missing or imperfect in the Ensembl Bacteria genomes, it has been shown in this study to be a useful and comparable method.

The Ensembl annotated amino acid sequences *(.pep.all.fa)* were combined into one FASTA file and gene clustering was then undertaken with CD-Hit (version 4.8.1) (Fu et al., 2012). The gene clustering was performed with the following parameters: aligned sequence identity threshold of 0.9 (90%), length difference cutoff of 0.6 (60%), and the '-g' option was set to the 'more accurate' option (see Appendix B.3 for more details). The 0.6 length difference cutoff between clustered sequences allowed for the instances where the StORF sequences contained additional upstream non-coding DNA which would have hindered the clustering of the matching coding regions. The strict sequence identity threshold of 0.9 determined that the resulting clusters were very similar across the regions where they did align. While there are different parameters available, due to the goal of reporting only the most similar gene clusters, while allowing for the known length differences of CDS sequences within gene families, the more stringent cutoffs were used.

The output of CD-Hit consists of two datafiles, a '.clstr' file containing the full sequence cluster metadata (which is a notoriously difficult file format to work with) and a FASTA file containing the representative ID and sequence for each cluster. The resulting CD-Hit cluster file which consisted of the identified Ensembl gene clusters, was used as a baseline against which to compare the additional StORF predictions. The results of these clusters are complex and have been classified differently in respect to the sequences they have clustered. As such they will be referred to as follows:

- **Ensembl-Only**, which refers to the clusters with only Ensembl annotated protein sequences.
- **Ensembl-StORF**, which refers to the clusters which contain at least one sequence from both Ensembl annotated protein sequences and one StORF amino acid sequence.
- **StORF-Only**, which refers to the clusters which solely contain StORF amino acid sequences.

These names are used throughout the rest of Chapter 3 (with minor changes in and Chapter 4). *E. coli* StORF sequences were then combined with the previous CD-Hit output (FASTA) from the Ensembl proteins, which consisted of one representative sequence for each cluster (representatives are often the longer of the sequences in any one cluster). This combined FASTA file then underwent another round of CD-Hit clustering with the same parameters and produced the final *E. coli* pangenome datafiles.

To interpret the output from CD-Hit, a number of Python3 scripts were built. **single_Genera_CD_Hit_StORF_Reporter_Core_Extension.py** was built to first identify the core gene family groups of the 219 *E. coli* strains and then identify whether any StORFs clustered with these representative gene cluster groups, but also whether there were any StORF-only gene clusters spanning large numbers of *E. coli* strains. This script works by first loading in the CD-Hit .clstr output file made from the protein sequences annotated in Ensembl and for each cluster reported in that .clstr file, building a dictionary which consists of the cluster id (number) as key and list of *E. coli* genomes as values. This allows us to count the number of unique strains that make up each cluster and therefore the ability to calculate each cluster's pangenomic status (core, soft-core, accessory). Next, the .clstr file produced from the combined Ensembl representative sequences and full StORF collection was loaded in to identify whether the Ensembl gene clusters had been added to by StORFs or whether there were any StORF only gene clusters. This was done by using the CD-Hit picked representative sequence IDs from the Ensembl only .clstr file to link to the combined CD-Hit run and count any additional StORF sequences which had clustered to those representative sequences. The StORF-Finder output sequence ID tag 'Stop-ORF' was used to distinguish between the Ensembl and StORF sequences. Singleton clusters

were not recorded as part of this study as their presence in the combined clustering strongly indicated that they without sequence identity to sequences in either the Ensembl or StORF datasets. Therefore, they were not used in this study. Lastly, any clusters with only StORF sequences were identified and the number of strains they spanned were recorded.

It is difficult to predict what the result of clustering many thousands of protein sequences together could be. Through introducing hard-coded computation parameters into the natural world, it is inevitable that certain cases are unaccounted for. For example, multiple representatives from separate Ensembl gene clusters can be combined into a new single cluster with the addition of StORF sequences. To account for this, the Ensembl clusters which were combined when StORF sequences were added were first recorded separately in the Ensembl gene cluster results, and then additionally recorded in their new combined clusters with the additional StORFs (see results rows 1 and 2 respectively in Table 3.11).

3.3.7 Inter-Genera Gene Clustering

An important aspect of gene family research concerns their diversity and spread across different genera. To investigate whether StORFs may also be shared across diverse prokaryotic genera, the Ensembl Bacteria protein sequences and StORFs extracted from the URs of the 6,223 filtered genomes were both studied. The same CD-Hit parameters and workflow were undertaken as for the *E. coli* pangenomic study. A modification of the previously developed single species script was built to handle multiple genera, **multiple_Genera_CD_Hit_Con_StORF_Core_Extension.py**. This script records multiple different levels of taxa information for each of the sequences such as genus, species and strain. However, as the study on *E. coli* had already covered StORF-Reporter's ability to capture gene sequences at the species level, only the genus specific to each sequence in this study were recorded.

3.4 Results

3.4.1 Unannotated Regions

Model Organism	Number of Ensembl Genes	Number of Ensembl URs	Longest Ensembl UR Length	Median Ensembl UR Length **[SD]**
B. subtilis	4,133	2,711	1,407	226 **[137.79]**
C. crescentus	3,875	2,321	3,477	221 **[172.85]**
E. coli	4,257	2,743	6,275	243 **[353.37]**
M. genitalium	559	157	4,922	185 **[673.16]**
P. fluorescens	5,266	3,509	20,088	244 **[633.74]**
S. aureus	2,556	1,666	2,591	307 **[235.16]**

TABLE 3.3: This table presents the results of running UR_Extractor on the Ensembl annotations for the six model organisms. Each UR is extended with 50nt at each end. All lengths in nt. Standard deviation is abbreviated as **[SD]**.

Model Organism	Number of Prodigal Genes	Number of Prodigal URs	Longest Prodigal UR Length	Median Prodigal UR Length **[SD]**
B. subtilis	4,016	2,619	6,259	225 **[311.49]**
C. crescentus	3,704	2,394	6,594	231 **[250.10]**
E. coli	4,263	2,743	4,055	242 **[247.13]**
M. genitalium	995	636	2,646	233 **[221.24]**
P. fluorescens	5,421	3,524	4,264	239 **[237.28]**
S. aureus	2,534	1,650	12,332	303 **[492.45]**

TABLE 3.4: This table presents the results of running UR_Extractor on the Prodigal CDS predictions for the six model organisms. Each UR is extended with 50nt at each end. All lengths in nt. Standard deviation is abbreviated as **[SD]**.

UR-Extractor was first applied to both the canonical Ensembl annotations and the Prodigal predictions of the six model organisms, as shown in Tables 3.3 and 3.4. The number of URs reported for each model organism were similar between the Ensembl and Prodigal annotations, except for *M. genitalium* which were 157 and 636 respectively. This was likely due to the inability of Prodigal to account for reassignment of the 'UGA/TGA' stop codon and its resultant misreporting of a number of truncated ORFs. The Ensembl annotations also included non-coding genes which were treated the same as coding genes by the UR-Extractor tool. As Prodigal does not predict non-coding genes, it is likely to report additional URs compared to Ensembl. This can explain some of the higher numbers and longer UR lengths in the Prodigal analysis. However, this was not consistent across all MOs. For example, while the longest UR length in *S. aureus* was 2,591 in the Ensembl annotation, this

increased to 12,332 in the Prodigal annotation. However, the longest UR in *P. fluorescens* decreased from 20,088 to 4,264 for Ensembl and Prodigal annotations respectively.

By studying the URs extracted with either the annotations of Prodigal or Ensembl, there were clearly a number of URs with lengths long enough to contain complete CDS genes. While some of the URs of Prodigal are likely to contain a number of known (in Ensembl) non-coding genes and other genomic elements, as StORFs are specifically designed to identify coding genes, this should not be a problem.

3.4.2 StORF-Reporter Recovers Ensembl Genes Missed by Prodigal

Model Organism	Number of StORFs	Recovered **[Non-vitiated]**
B. subtilis	2,472	12 **[51]**
C. crescentus	1,827	26 **[100]**
E. coli	2,866	22 **[72]**
M. genitalium	587	1 **[6]**
P. fluorescens	3,227	14 **[51]**
S. aureus	2,123	6 **[16]**

TABLE 3.5: Table containing the number of Prodigal StORFs and the number of non-vitiated Ensembl genes recovered by StORF-Reporter which Prodigal missed. Non-vitiated genes are those which had an overlap of less than 50 nt with a Prodigal predicted CDS, thus allowing for them to be included in an extracted UR.

An analysis of the StORFs extracted from the URs of the six MO genomes separately, was undertaken using the annotations from Prodigal. StORF-Reporter was able to recover Ensembl genes that were missed by Prodigal, and also found many other potential genes that had sequence identity to the Swiss-Prot database or proteins already annotated in the MO genomes.

The ORForise platform reported that Prodigal was able to identify the vast majority of Ensembl genes from each of the MOs, except for *M. genitalium*. The genomic regions containing no CDS predictions were extracted with UR-Extractor and StORFs were reported from these URs using StORF-Finder. For each of the MOs StORF-Reporter identified a high number of StORFs, as can be seen in Table 3.5. However, as noted, StORF-Reporter is impeded by the mispredictions of Prodigal. For each model organism, Prodigal reported a number of CDSs which either had no representative in the Ensembl annotation, were elongated versions of Ensembl genes or were too inaccurate to be classified as 'detected', according to the ORForise platform. This meant that StORF-Reporter was only able to search for StORFs in the non-vitiated regions of the genome. Between 1-26 non-vitiated missed Ensembl genes were recovered by using StORF-Reporter with the Prodigal annotations.

The StORFs from each of the model organisms were DIAMOND (Buchfink, Xie, and Huson, 2015) blastp searched against databases created from both the Swiss-Prot

protein database and the proteomes of the respective model organism (see Table 3.6). From these results, it appears that while the ability of StORF-Reporter to recover Ensembl genes which were missed by Prodigal is limited, the number of StORFs which were observed with a high level of sequence identity to both proteins in the curated Swiss-Prot database and from those already identified in the Ensembl annotations requires further investigation.

Model Organism	Number of StORFs	Swiss-Prot **[Coverage >=80%]**	Intra-Genome **[Coverage >=80%]**
B. subtilis	2,472	45 **[30]**	39 **[33]**
C. crescentus	1,827	6 **[4]**	60 **[48]**
E. coli	2,866	75 **[52]**	34 **[29]**
M. genitalium	587	184 **[5]**	182 **[2]**
P. fluorescens	3,404	16 **[5]**	42 **[35]**
S. aureus	2,053	19 **[1]**	25 **[13]**

TABLE 3.6: Table containing the number of Prodigal StORFs which were reported with a hit to either the Swiss-Prot or Intra-Genome protein databases. Intra-Genome is the proteome of the same model organism. DIAMOND blastp hits are recorded with a minimum of a 60 bit score and in bold are reported with a subject coverage of >=80% .

The triplet abundance of the three canonical stop codons, irrespective of their usage as a stop codon for a CDS gene, has, for the most part, little influence on their subsequent use as CDS gene stop codons. However, as shown in Figure 3.6, this is not consistent. As such and due to the innate importance of stop codons for the StORF-Reporter methodology, the stop codons used in the Prodigal predicted CDS genes and StORFs (identified from within the URs reported by Prodigal) for each of the 6 model organisms have been inspected. As seen in Table 3.7, the relationship between the genome-wide abundance and usage by the Prodigal predicted CDS genes for each of the three canonical stop codons is also unclear. On the other hand, the reported StORF stop codon usage is much closer to their reported abundance, for each respective genome. Lastly, while the triplet abundance and Prodigal StORF stop codon usages were closer than the triplet abundance compared to the Prodigal CDS gene stop codon usages, they were still statistically significantly different.

3.4.3 StORF-Reporter Finds Complete Genes Not Present in Ensembl Annotations

StORF-Reporter found potential genes not just in the URs reported by Prodigal, but also in the URs from the curated Ensembl annotation. A high number of StORFs have been identified in the Ensembl annotated URs of each of the six MOs. Each of these StORFs have the potential to contain not only undiscovered genes but also historic gene fragments or other functional units waiting to be characterised.

	Genome Triplet Abundance			Prodigal Gene Stop Usage				Prodigal StORF Stop Usage			
Genomes	TGA [%]	TAG [%]	TAA [%]	TGA [%]	TAG [%]	TAA [%]	$\tilde{\chi}^2$ p-value	TGA [%]	TAG [%]	TAA [%]	$\tilde{\chi}^2$ p-value
B. subtilis	180,347 **[49.57]**	52,378 **[14.39]**	131,084 **[36.03]**	932 **[23.21]**	563 **[14.02]**	2,521 **[62.77]**	<0.00001	1,025 **[41.46]**	424 **[17.15]**	1,023 **[41.38]**	<0.00001
C. crescentus	102,367 **[69.85]**	32,635 **[22.27]**	11,541 **[7.87]**	1,735 **[46.84]**	1,230 **[33.21]**	739 **[19.95]**	<0.00001	1,059 **[57.96]**	458 **[25.07]**	310 **[16.97]**	<0.00001
E. coli	164,560 **[46.64]**	53,119 **[15.05]**	135,187 **[38.31]**	1,232 **[28.90]**	332 **[7.79]**	2,699 **[63.31]**	<0.00001	1,093 **[38.14]**	486 **[16.96]**	1,287 **[44.91]**	<0.00001
M. genitalium	25,382 **[28.26]**	18,982 **[21.13]**	45,456 **[50.60]**	612 **[61.51]**	110 **[11.06]**	273 **[27.44]**	<0.00001	261 **[44.46]**	100 **[17.04]**	226 **[38.50]**	<0.00001
P. fluorescens	189,251 **[64.36]**	56,411 **[19.18]**	48,377 **[16.45]**	3,010 **[55.52]**	770 **[14.20]**	1,641 **[30.27]**	<0.00001	1,644 **[50.95]**	750 **[23.24]**	833 **[25.81]**	<0.00001
S. aureus	119,798 **[30.87]**	73,821 **[19.02]**	194,474 **[50.11]**	268 **[10.58]**	379 **[14.96]**	1,886 **[74.46]**	<0.00001	512 **[24.12]**	429 **[20.21]**	1,182 **[55.68]**	<0.00001

TABLE 3.7: Presented in this table are the following: the triplet abundance of the three canonical stop codons found throughout the six model organism genomes (totaled from both forward and reverse strands), the stop codons used in the Prodigal predicted CDS genes, and the end stop codon used in the StORFs identified from within the URs reported by Prodigal, both from the 6 model organisms which have been inspected. A chi squared test was performed on each model organism: triplet abundance vs Prodigal gene stop codon usage and triplet abundance vs StORF stop codon. Each test resulted in a rounded p-value of <0.00001.

Model Organism	Number of StORFs	Swiss-Prot **[Coverage >=80%]**	Intra-Genome **[Coverage >=80%]**
B. subtilis	2,322	26 **[20]**	8 **[2]**
C. crescentus	1,554	8 **[5]**	15 **[3]**
E. coli	2,798	148 **[107]**	57 **[13]**
M. genitalium	168	63 **[4]**	50 **[0]**
P. fluorescens	3,404	73 **[46]**	173**[42]**
S. aureus	2,503	23 **[4]**	17 **[3]**

TABLE 3.8: Table containing the number of StORFs reported from the URs recovered from the Ensembl annotations with a hit to either the Swiss-Prot or Intra-Genome protein databases. Intra-Genome is the proteome of the same model organism. DIAMOND blastp hits are recorded with a minimum of a 60 bit score and in bold are reported with a subject coverage of >=80%.

A number of StORFs had a high sequence identity to known protein coding genes in the Swiss-Prot protein database. These StORFs could be full length or fragments of genes which are missing in the Ensembl data. There were also a number of StORFs which, while of considerable length, did not have a high level of sequence identity. This could indicate that these sequences are not yet present in the databases or are fragmented in the genome.

The number of StORFs which attained a match to a protein sequence from the curated Swiss-Prot database are reported in Table 3.8. There were also a number of StORFs, which while many were of notable length (>100 amino acids), did not obtain a match in the Swiss-Prot database (see Figure 3.7). These StORFs, representing

long in-frame and non-interrupted regions of assumed intergenic DNA, could indicate that these sequences are not yet present in the database or are in fragmented form in the genome and thus do not pass the sequence alignment cut-offs.

To study the possibility that StORFs may capture instances of gene duplication, for each of the MOs, the reported StORFs were aligned to protein sequence databases built from each of the Ensembl annotations. Between 8-173 StORFs were observed to contain a high quality alignment to putative duplicate 'genic' sequences from within the same genome. These StORFs are likely to represent genes that have undergone duplication and either subsequent loss mutation or have not been detected in the original genome annotation process. While *M. genitalium* observed 50 intra-genome StORFs 'hits', it was the only MO not to have a StORF alignment >=80% of an Ensembl, likely due to the 'UGA/TGA' reassignment. The results are reported in Table 3.8.

Table 3.9 reports that the genome-wide triplet abundance of the canonical stop codons and their usage by the Ensembl predicted CDS genes is significantly different. The same level of difference is also reported for the StORFs extracted from the Ensembl URs.

This is to be expected as the high level of agreement of stop codon usage between the Ensembl and Prodigal annotated CDS genes has already been reported in Chapter 2. Additionally, stop codon usages for Ensembl and Prodigal CDS genes are very similar, often within 1% variance (apart from *M. genitalium*) (see Tables 3.7 and 3.9). A similar low variance was also observed between the StORFs reported from the Ensembl annotated and Prodigal annotated URs. Interestingly, while not reported, the triplet usage of both the 5' and 3' end of each set of StORFs varies less than 1% compared to 3' stop codon for the same set of StORFs. Both of these low variances further indicate that there is an underlying signal influencing StORF triplet and stop codon usage which is yet unclear. Lastly, while the variances between the triplet abundance and Ensembl StORF stop codon usages were closer than compared to the Ensembl gene stop codon usages, they were still statistically different.

	Genome Triplet Abundance			Ensembl Gene Stop Usage				Ensembl StORF Stop Usage			
Genomes	TGA [%]	TAG [%]	TAA [%]	TGA [%]	TAG [%]	TAA [%]	$\tilde{\chi}^2$ p-value	TGA [%]	TAG [%]	TAA [%]	$\tilde{\chi}^2$ p-value
B. subtilis	180,347 **[49.57]**	52,378 **[14.39]**	131,084 **[36.03]**	927 **[23.11]**	560 **[13.96]**	2,524 **[62.93]**	<0.00001	951 **[40.96]**	374 **[16.11]**	997 **[42.94]**	<0.00001
C. crescentus	102,367 **[69.85]**	32,635 **[22.27]**	11,541 **[7.87]**	1,770 **[47.36]**	1,225 **[32.78]**	742 **[19.86]**	<0.00001	916 **[58.94]**	391 **[25.16]**	247 **[15.89]**	<0.00001
E. coli	164,560 **[46.64]**	53,119 **[15.05]**	135,187 **[38.31]**	1,151 **[28.41]**	279 **[6.89]**	2,621 **[64.70]**	<0.00001	1,096 **[39.17]**	437 **[15.62]**	1,265 **[45.21]**	<0.00001
M. genitalium	25.382 **[28.26]**	18.982 **[21.13]**	45.456 **[50.60]**	0 **[0]**	129 **[27.10]**	347 **[72.90]**	<0.00001	80 **[47.62]**	20 **[11.90]**	68 **[40.48]**	<0.00001
P. fluorescens	189,251 **[64.36]**	56,411 **[19.18]**	48,377 **[16.45]**	2,869 **[55.41]**	734 **[14.18]**	1,575 **[30.42]**	<0.00001	1,756 **[51.58]**	761 **[22.35]**	887 **[26.06]**	<0.00001
S. aureus	119,798 **[30.87]**	73,821 **[19.02]**	194,474 **[50.11]**	271 **[10.84]**	379 **[15.16]**	1,850 **[74.00]**	<0.00001	477 **[23.23]**	370 **[18.02]**	1,206 **[58.74]**	<0.00001

TABLE 3.9: Presented in this table are the following: the triplet abundance of the three canonical stop codons found throughout the six model organism genomes (totaled from both forward and reverse strands), the stop codons used in the Ensembl annotated CDS genes, and both end stop codons used in the StORFs identified from within the URs reported by Ensembl, both from the 6 model organisms which have been inspected. A chi squared test was performed on each model organism: triplet abundance vs Ensembl gene stop codon usage and triplet abundance vs StORF stop codon. Each test resulted in a rounded p-value of <0.00001.

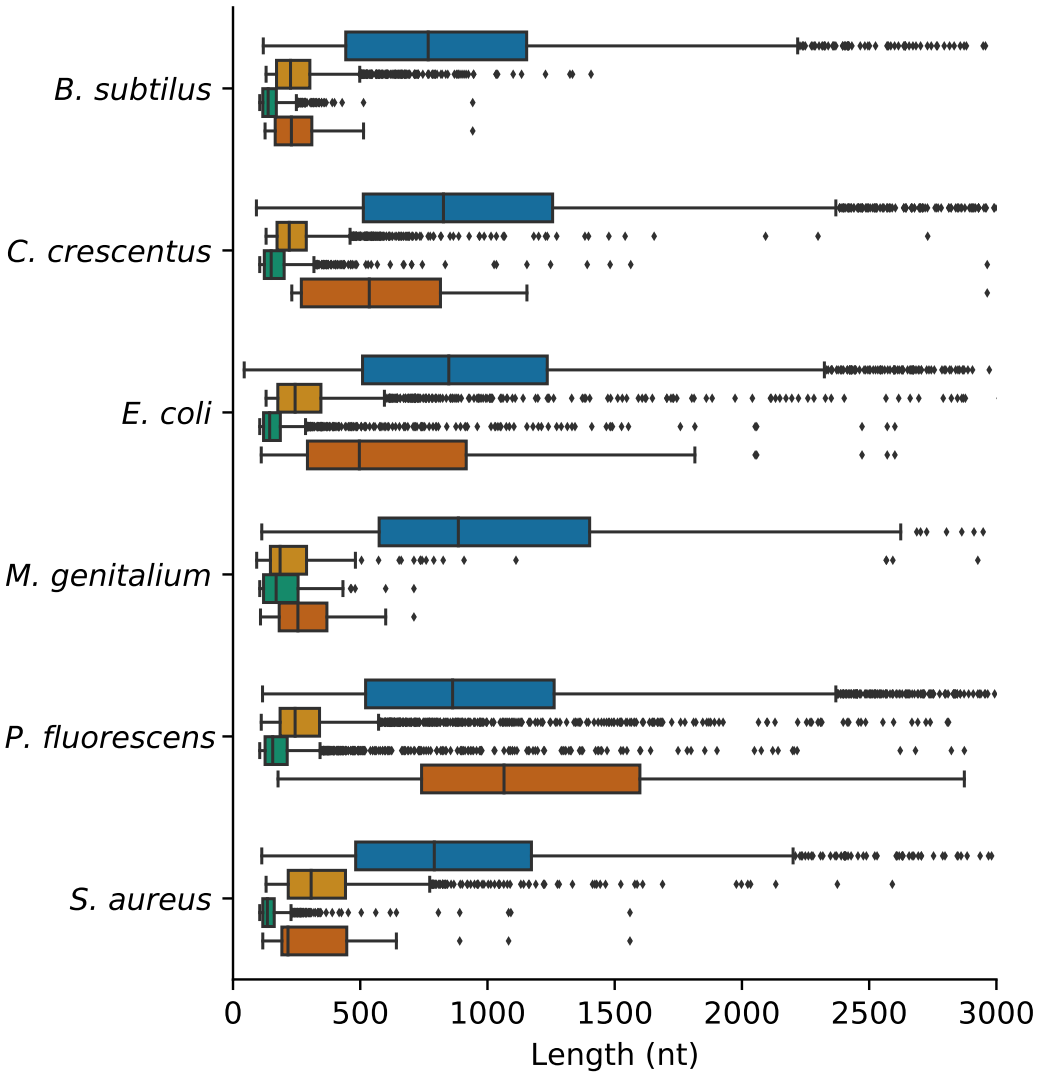


FIGURE 3.7: Shown here are the nucleotide lengths of the Ensembl genes (blue), unannotated regions (URs) extracted from the Ensembl annotations of each of the six model organisms (light orange), the StORFs identified from the URs (green) and the StORFs which had a high sequence identity to known protein coding genes in Swiss-Prot (>=60% bitscore) (dark orange). X axis truncated at 3,000 nt.

3.4.4 Extending the *Escherichia coli* Pangenome

StORF-Reporter found StORFs that both extend the core gene set and create novel core gene clusters which has the potential to extend our knowledge of the *E. coli* pangenome.

The 6,223 Ensembl Bacteria collection was filtered to give a set of 219 *E. coli*. From the 1,042,068 Ensembl protein sequences annotated in these 219 genomes, 34,737 gene clusters were formed, of which 20,676 were non-singletons. A median number of 3,038 URs and 2,958 StORFs were identified for each the 219 genomes for a total of 673,136 URs and 652,056 StORFs (see Table 3.10 for more detail).

Data	Unannotated Regions	StORFs
Number of Sequences	673,136	652,056
Median Number Per Genome	3,038	2,958
Longest Sequence (nt)	45,683	14,334
Median Sequence Length (nt)	234	141
[SD]	**[292.97]**	**[177.48]**

TABLE 3.10: Presented here are the numbers and lengths of unannotated regions (URs) and StORFs extracted from the 219 *Escherichia coli* genomes. While there was variability in the genome quality across this set of genomes, the numbers reported here are similar to those reported for the 6 model organisms. Standard Deviation is reported as **[SD]**.

The representatives from each of the Ensembl-Only clusters (all 34,737) were combined with the 652,056 StORFs identified from the same 219 *E. coli* genomes and the same CD-Hit sequence clustering was applied. This resulted in a total of 86,579 clusters, of which 31,676 were non-singletons. 51,929 clusters were formed of only StORFs (StORF-Only clusters) and 28,094 clusters were formed of only Ensembl genes (Ensembl-Only clusters). The remainder clustered StORFs together with Ensembl genes (Ensembl-StORF clusters). Some of the clusters containing StORFs and Ensembl genes contained multiple Ensembl cluster representatives, thus combining multiple gene families which had previously been split apart. This was a result of StORF sequences effectively bridging the gap between multiple Ensembl sequences (defined henceforth as StORF-Combined-Ensembl clusters). Out of a total of 6,643 Ensembl gene families that clustered with StORF sequences and extended them into additional *E. coli* genomes, 87 were formed where the clustered StORF sequences did not extend the gene family into additional genomes. Figure 3.8 shows that the distribution of the unique *E. coli* genomes present in the Ensembl-Only, Ensembl-StORF and StORF-Only clusters both followed a similar reverse bell curve. Interestingly, the size of the clusters which were extended by the StORFs was often increased by an substantial amount. For example, the proportion of clusters with 10 or fewer unique genomes was reduced from just below 60% to 40% with the addition of the StORF sequences.

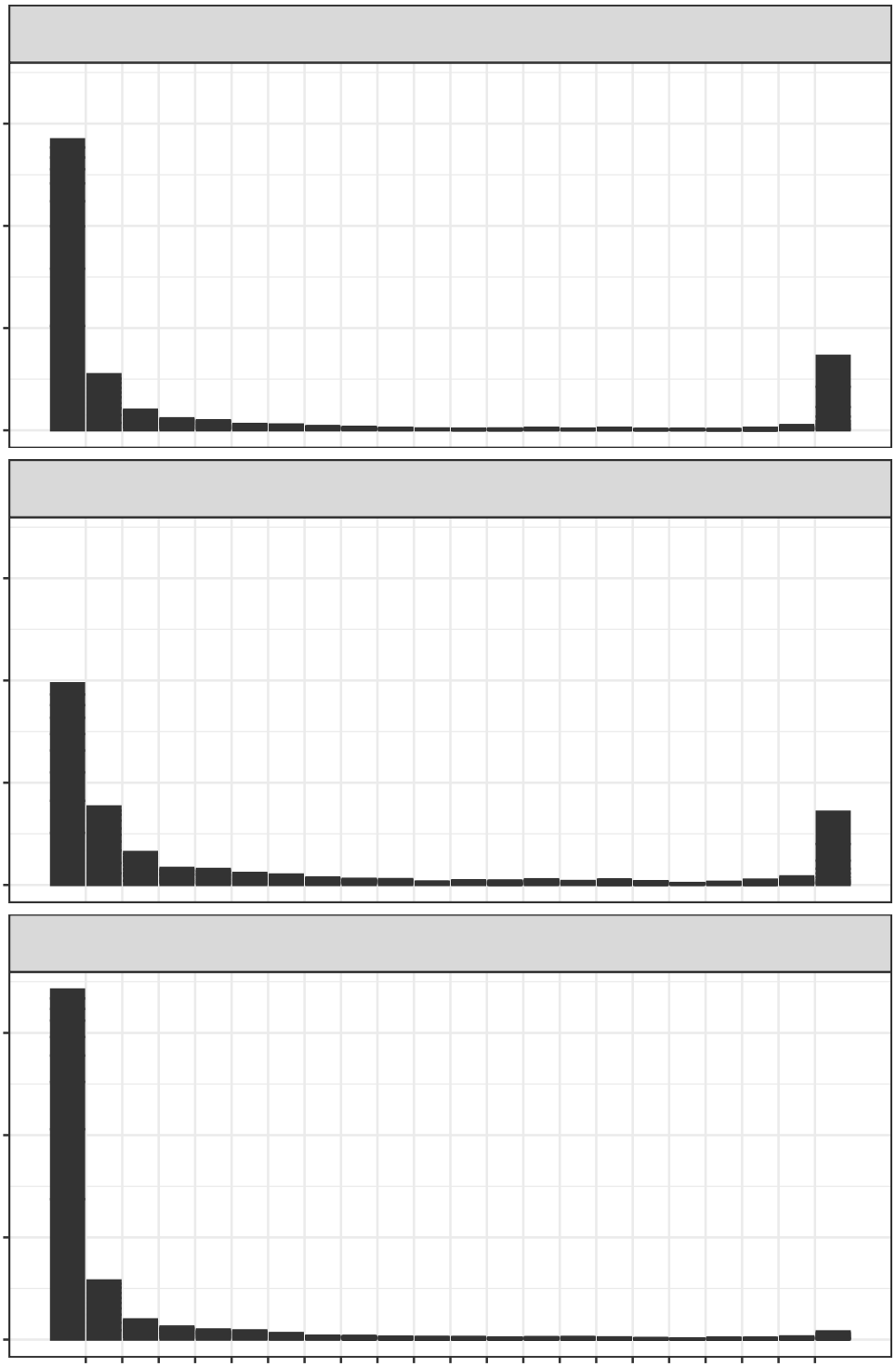


FIGURE 3.8: The distributions of gene families across the 219 *E. coli* pangenome for the Ensembl-Only, Ensembl-StORF and StORF-Only clusters are plotted here. The reverse bell curve is consistent throughout the three cluster types with Ensembl-StORF containing slightly larger gene family clusters as expected due to the added StORF sequences as compared to Ensembl-Only. While the distribution is more towards the lower end for StORF-Only, the same reverse bell curve is observed.

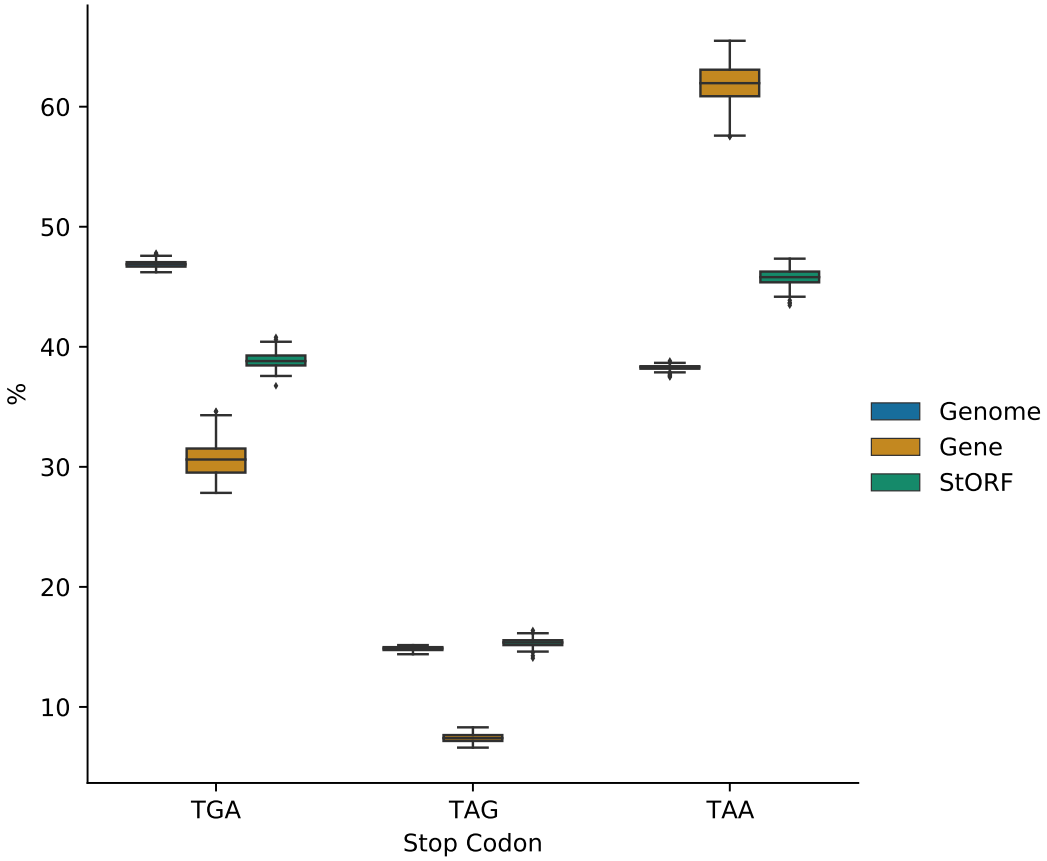


FIGURE 3.9: Presented in this boxplot is the proportional presence of each stop codon for the 219 *E. coli* pangenome genomes (combined from both forward and reverse strands), their usage in Ensembl annotated CDS genes and the stop codons used for the StORFs identified in the Ensembl Unannotated Regions (URs). Both the first and last stop codons used for each StORF are reported here.

The stop codon presence, Ensembl CDS and Ensembl StORF stop codon usages for the 219 *E. coli* genomes were studied. As seen in Figure 3.9, while a similar divergence of codon presence and stop codon usage for Ensembl CDS gene and StORFs was observed as in the analysis on the 6 model organisms, there was some variation between the 219 genomes. Two clear examples of this are the 27.48-34.63% and 57.50-65.49% ranges of TGA and TAA stop codon usages for the Ensembl CDS gene stop codons. Fundamentally, these results show that even within a single species, stop codon usages can vary.

The 2,612 core gene (found in >= 99% genomes) gene families consisting of only Ensembl genes is a similar figure to that of the number of core genes found in other studies of *E. coli* strains (Maddamsetti et al., 2017). Taking the varying levels of genome assembly and annotation quality into consideration, a large number of gene clusters (with and without StORF sequences) were found to contain genes from more than 95% of the *E. coli* genomes (see Table 3.11). Interestingly, some clusters which fell below this 95% level were extended into the 95% and 99% thresholds by the

addition of their clustered StORF sequences. There were also StORF-Only clusters with sequences from more than 95% and 99% of the *E. coli* genomes.

Cluster Types	Core	Soft-Core	Accessory
Ensembl-Only	2,612	455	2,597
Ensembl-StORF	178	67	610
StORF-Combined-Ensembl	0	1	11
StORF	9	15	648
StORF-Only	239	216	3,426

TABLE 3.11: *Escherichia coli* gene families calculated from the set of 219 strains can be extended by the addition of StORFs (likely missed genes) found by the StORF-Reporter methodology. Definitions of the gene families are as follows: Core genes >= 99%, Soft-core genes >= 95% to < 99% and Accessory genes >= 15% and < 95%. Gene families are only counted once. For example, a gene family which is in the Core gene group is not also part of the Soft-Core gene group. Ensembl-Only, Ensembl-StORF and StORF-Only, have been described earlier. The third group 'StORF-Combined-Ensembl', reports the number of gene families where StORF sequences combined at least 2 or more Ensembl cluster representatives together. The fourth group 'StORF', reports the size of clusters in the Ensembl-StORF group but with only the StORF sequences being counted. This allows for the reporting of Ensembl-StORF clusters where it is the StORF sequences driving the distribution across the genomes.

The inherently limiting process of using hard-coded sequence identity and length cutoffs to distinguish gene families has inevitably brought forward a number of interesting cases. One example can be seen in the combining of multiple Ensembl protein representative sequences, and thus clusters, into one new cluster by the addition of StORFs. The 71 amino acid representative protein AKD71933 of Ensembl cluster 4,714 (consisting of 59 sequences and spanning 24 strains, some of which have duplicates or paralogs within the same genome), was clustered with the 66 amino acid protein AHM40952 of Ensembl cluster 6,112 (consisting of 26 sequences from 26 strains) when the StORFs sequences were included in the CD-Hit clustering. Previously these proteins were distinct enough to be clustered separately. The two Ensembl representatives were clustered together in combined Cluster 4 with 335 StORFs from 99 strains. The StORFs in this cluster are longer than the Ensembl genes (around 100 aa), and so CD-Hit chose a StORF to be its representative sequence for the combined cluster. As can be seen in the Clustal Omega (Sievers and Higgins, 2018) multiple sequence alignment of the representative sequences of the above-mentioned clusters, their sequence identity was high $\geq$ 90% (the minimum CD-Hit percentage identity for clustering) along the regions in which they aligned 3.10.

There were also examples of Ensembl gene clusters combined together with StORF sequences, which could completely change the dynamics of the Ensembl gene families. As seen in Figure 3.11, the three Ensembl representatives sequences, AHM40736, AIT36070 and AFS84250, are combined into a new cluster with a single

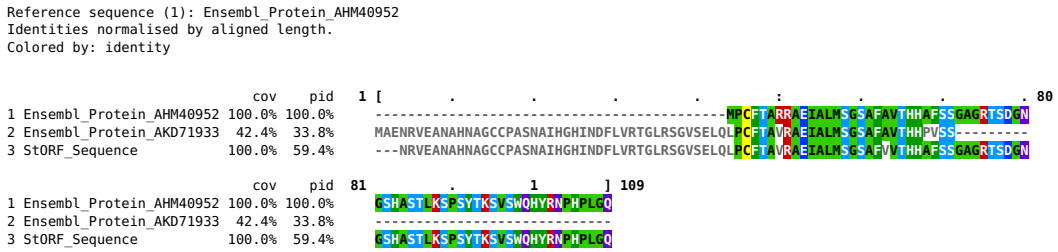


FIGURE 3.10: Clustal Omega multiple sequence alignment of the two Ensembl representative sequences, AHM40952 and AKD71933 with the *E. coli* pangenome cluster 4 StORF representative sequence.

StORF sequence (Cluster 17,575). As we can observe from their alignment, they are in fact very similar and are further examples of the limitations we impose on ourselves when attempting to categories natural elements into computationally-friendly representations. There were 77 of these types of combined clusters which together had clustered 164 Ensembl representative sequences together. These clusters are presented separately in Table 3.11. Interestingly, a number of these combined clusters contained more unique *E. coli* strains via the StORF sequences than via the Ensembl genes. Of the clusters that combined more than one previous Ensembl cluster, there were no cases of core (>= 99%) gene families but 1 example of soft-core (>= 95% to < 99%) and 11 accessory gene families (>= 15% to < 95%).

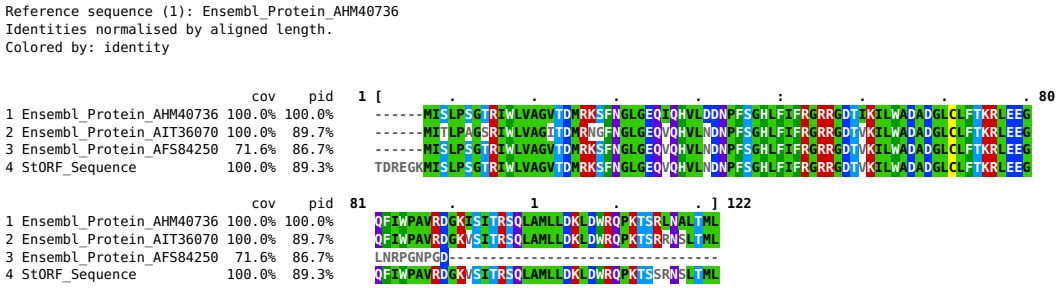


FIGURE 3.11: ClustalO multiple sequence alignment of the three Ensembl representatives sequences, AHM40736, AIT36070 and AFS84250 with the single *E. coli* pangenome cluster 17,575 StORF representative sequence. These three Ensembl representative sequences and their clusters have been grouped together by a single StORF sequence, thus changing the dynamics of this set of gene families completely. The sequences sequence are in their respective order in the alignment output.

As can be seen in Table 3.11, each of the three categories of the overall pangenome of *E. coli* have been extended by the addition of the StORF sequences. Among the most interesting results are the 179 gene families which were extended to the core of the pangenome by the sequences identified by StORF-Reporter. Additionally, there were cases where clusters which contained both Ensembl and StORF sequences, were classified as core gene families, on the basis of their StORF sequences alone. While they were grouped with additional Ensembl genes, the Ensembl genes did

not contribute to the distribution of the cluster (9 were core, 15 were soft-core and 648 were accessory). Most intriguingly, there were a large number of clusters which were made up entirely of StORF sequences (StORF-Only gene families) and spanned all three pangenome gene categories (239 were core, 216 were soft-core and 3,426 were accessory). Lastly, there were 162 Ensembl clusters which although did cluster with StORF sequences, none of the StORFs were from additional genomes and as such were not counted in the above data.

EggNOG COG functional categories were identified for the representative sequences from the initial set of 34,737 gene clusters in order to investigate whether the functional profiles are different between Ensembl and StORF clusters. The sequences which obtained a COG annotation were separated into three groups: those which did not cluster with a StORF sequence (Ensembl-Only), those which contained both Ensembl and StORF sequences (Ensembl-StORF) and those which only contained StORF sequences (StORF-Only). This resulted in 15,296, 23,419 and 6,481 clusters, respectively. While some singleton Ensembl-Only and StORF-Only clusters did have COG annotations, only those clusters which had sequences from at least 2 different genomes were reported. The 20 COG functional categories were grouped together into their 4 respective domains and are presented in Table 3.12. As the COG group 'Information Storage & Processing' was observed with the largest difference and increase between the Ensembl-Only and StORF-Only clusters, the individual COG categories were extracted and reported in Table 3.13. While the number of Ensembl-Only sequences which obtained a COG classification are much higher than for StORF-Only, the reported distribution of COG categories are similar in both. Both 'A' and 'B' are observed in very low numbers (9 'A', 1 'B' for Ensembl-only and 0 'A', 0 'B' for StORF-Only respectively). StORF-Only sequences have less 'K' but more 'L' than Ensembl-Only sequences, possibly hinting at a functional difference of missing gene function from canonical genome annotations.

COG Group	Ensembl-Only [%]	Ensembl-StORF [%]	StORF-Only [%]
INFORMATION STORAGE & PRO'	2,334 **[21.21%]**	832 **[23.09%]**	219 **[32.21%]**
CELLULAR PROCESSES & SIG'	2,861 **[26.00%]**	888 **[24.64%]**	129 **[18.97%]**
METABOLISM	2,447 **[22.24%]**	885 **[24.56%]**	132 **[19.41%]**
POORLY CHARACTERIZED	3,361 **[30.55%]**	999 **[27.72%]**	200 **[29.41%]**
With COGs/Total Sequences	10,165/15,296 **[66.46%]**	3,299/6,643 **[49.66%]**	642/24,767 **[2.60%]**

TABLE 3.12: The COG functional categories assigned to Ensembl-Only, Ensembl-StORF and StORF-Only cluster representative sequences with EggNOG Mapper for the *E. coli* pangenome analysis. Some sequences were observed to have more than one COG functional category. In these instances, the sequence is only counted once in the 'With COGs/Total Sequences' column but each individual COG is counted separately for the 4 groups. While some singleton Ensembl-Only and StORF-Only clusters did have COG annotations, only clusters which had sequences from at least 2 different genomes are reported here. Chi squared statistic tests reported a p-value of 0.000169 for Ensembl-only compared to Ensembl-StORF and <0.00001 for Ensembl-Only compared to StORF-Only. Further to this, the 'POORLY CHARACTERIZED' and 'INFORMATION STORAGE & PROCESSING' categories were identified with the highest chi-square statistic in each comparison.

COG Function	Ensembl-Only [%]	StORF-Only [%]
[J] Translation, ribosomal structure and biogenesis	245 **[10.50%]**	9 **[4.11%]**
[A] RNA processing and modification	9 **[0.39%]**	0 **[0%]**
[K] Transcription	823 **[35.26%]**	61 **[27.85%]**
[L] Replication, recombination and repair	1,256 **[53.81%]**	149 **[68.04%]**
[B] Chromatin structure and dynamics	1 **[0.04%]**	0 **[0%]**

TABLE 3.13: The COG functional categories assigned to Ensembl-Only and StORF-Only cluster representative sequences for the group 'Information Storage and Processing' of the *E.coli* pangenome analysis. While the number of Ensembl-Only sequences which obtained a COG classification are much higher than for StORF-Only, the reported COG categories are similar in both. Both 'A' and 'B' are observed in very low numbers (9, 0 and 1, 0 for Ensembl-Only and StORF-Only respectively). The proportion of StORF-Only sequences with 'K' was less than Ensembl-Only sequences but more had 'L', possibly hinting at a functional overview of missing gene function from canonical genome annotations.

3.4.5 StORFs Identified Within and Across Multiple Genera

Data	Unannotated Regions	StORFs
Number of Sequences	14,221,482	13,301,175
Median Number Per Genome	2,305	1,981
Longest Sequence (nt)	86,235	47,790
Median Sequence Length (nt)	240	147
[SD]	**[366.07]**	**[213.24]**

TABLE 3.14: Presented here are the numbers and lengths of unannotated regions (URs) and StORFs extracted from the 6,223 genomes from Ensembl Bacteria. While there was variability in the genome quality across this set of genomes, the numbers reported here are similar to those reported for the 6 model organisms and the *E. coli* pangenome analysis. Standard deviation is abbreviated as **[SD]**.

The previous analyses of URs and StORFs have been conducted on the same species. It could be assumed that even in the case of the *E. coli* pangenome, many of the StORFs which have been identified are remnants from a zmultitude of different genomic factors such as assembly error, genome structural elements or additional processes we are yet to understand. In this study, all 21,503,164 protein sequences were utilised from the 6,223 genomes from Ensembl Bacteria (see Tables 3.1).As with the *E. coli* pangenome analysis, a substantial number of URs and StORFs were identified across the genomes studied. A median number of 2,305 and 1,981 URs and StORFs were identified for each of the 6,223 genomes for a total of 14,221,482 and 13,301,175 respectively (see Table 3.14 for more detail). Additionally, Ensembl-Only, Ensembl-StORF and StORF-Only clusters, were identified spanning multiple genera across the 179 genera from Ensembl Bacteria. As would be expected, the distribution of genera in the Ensembl-Only clusters is wide and many clusters are reported with more than six different genera (see Table 3.15). Furthermore, not only have Ensembl-Only clusters been added to and extended by StORFs (creating Ensembl-StORF clusters), but also substantial numbers of StORF-Only clusters have been found with StORFs from multiple genera.

Many clusters were identified with both Ensembl and StORF sequences from the same genera and genomes. While these are likely to represent important candidates for gene duplication studies, they were not the aim of this study.

StORFs have changed the diversity and spread of gene families by extending the canonical Ensembl gene clusters into additional genera which were not annotated in the Ensembl annotations. One example of this, from the Ensembl-Only clusters, is a singleton cluster (Cluster 1,845,585) which did not cluster with any other Ensembl gene from any of the 6,223 genomes. It consisted of a 35 amino acid *E. coli* Ensembl gene (reported in GenBank as a hypothetical protein - *E. coli* kte42 - gene ELF73632). This singleton sequence, when combined with the StORF sequences from all 6,223 Ensembl genomes, was clustered into the 4th largest combined cluster (Ensembl and StORF combined 'Cluster 3'), with StORFs spanning multiple different

Cluster Type	1 Genus	2 Genera	3 Genera	4 Genera	5 Genera	6 Genera	>6 Genera
Ensembl-Only	1,569,403	6,9884	6,323	2,491	1,338	803	1,427
Ensembl-StORF	0	0	142	69	38	19	37
Ensembl-StORF-Combined	0	0	0	1	0	0	5
StORF	125,357	2,530	247	63	29	10	25
StORF-Only	1,087,592	27,079	2,313	813	367	172	265

TABLE 3.15: Presented here are the number of clusters which have sequences from multiple genera. The five cluster types are; (1) Ensembl-Only, (2) Ensembl-StORF, which are the clusters which have been extended into their respective genera group by the addition of StORF sequences, (3) Ensembl-StORF-Combined, which reports the number of gene families where StORF sequences combined at least 2 or more Ensembl cluster representatives together, (4) StORF, which are the same clusters as Ensembl-StORF but are counted only by their number of StORF sequences and (5) StORF-Only, which are the clusters which only contain StORF sequences and thus did not cluster with any Ensembl sequence. StORF-Only clusters with a single sequence were not included in these results.

genera. This cluster consisted of 1,162 StORF sequences from a total of 149 genomes and 5 genera (*Shigella, Escherichia, Pantoea, Klebsiella, Enterobacter*). The fact that the only organism to have this sequence annotated was *E. coli* may not be a coincidence.

Another example is a large Ensembl gene family (Ensembl Cluster 31), reported in Genbank to be protein 'CQR83579' which is a '30S ribosomal protein S18' (Sengupta, Agrawal, and Frank, 2001), which spanned 26 genera and 1,030 sequences. This gene family was extended to a Ensembl-StORF cluster by StORF sequences in the combined cluster 277,221. The Ensembl representative protein 'CQR83579' from an *E. coli* k12 strain was clustered with 4 StORFs from three strains of *Yersinia pestis* and one *E. coli*. These StORFs came from genomes which were not represented in the original Ensembl cluster 31. The four StORF sequences were 81 amino acids long as opposed to the 80 amino acids of the Ensembl representative and had between 96% and 100% sequence identity. These results were interesting for the specific reason that without the addition of the StORF sequences, analysis on those four genomes could incorrectly conclude that this likely essential ribosomal gene is not present. This may lead to the assumed dynamics of the genomes and their subsequent pangenomes being incorrect or at the very least incomplete.

As a final example, the small Ensembl Cluster 83,472, which consisted of 32 sequences between 63-70 amino acids long and spanning 4 genera (*Escherichia, Streptococcus, Yersinia* and *Shigella*) was extended significantly by 726 StORF sequences from 7 additional genera (*Pantoea, Klebsiella, Citrobacter, Enterobacter, Serratia, Salmonella*

and *Bacillus*) as part of the Ensembl-StORF combined Cluster 14. The function of this protein is unclear.. A BLAST search against the NCBI non-redundant protein database reported a number of different annotations which indicate that it could be either part of a transposase family, a plasmid stability protein or something else entirely.

As with the *E. coli* pangenome study, the hard-coded cutoffs used for the CD-Hit gene family analysis produced a number of clusters which combined multiple Ensembl representative sequences which were previously clustered separately. Six Ensembl-StORF combined clusters were found to contain more than one Ensembl representative sequence and together combined 14 Ensembl-Only clusters. Additionally, many of the StORF sequences were found in genera which were not present in the original Ensembl-Only clusters. An example of one such cluster, Ensembl-StORF cluster 124,470, combined three Ensembl-Only representatives together with seven StORF sequences. Figure 3.12 shows the phylogenetic tree built with ClustalO (Sievers and Higgins, 2018) and FastTree (Price, Dehal, and Arkin, 2010) and plotted with iTOL (Letunic and Bork, 2021). This tree clearly reports the diversity and at the same time the identity of the StORFs and Ensembl annotated sequences. Arguably, as with the combined clusters in the *E. coli* pangenome study, these StORF sequences bridge the gap between the three separate Ensembl gene families and as such have changed the dynamics of these inter-genera gene families. The multiple sequence alignment for these sequences is shown in Figure 3.13 and shows how the 3 Ensembl genes have been clustered together with the StORF sequences. While the percentage identities are lower than 90% between the 3 Ensembl genes (the reason they did not cluster together), they are more than 90% when compared to some of the StORF sequences. Additionally, this multiple sequence alignment depicts how for some amino acid positions, StORF sequences can have a higher level of sequence identity to one or more Ensembl gene as they are compared to each other. These type of clusters were included in the results of Table 3.15, separately in row 'Ensembl-StORF-Combined'.

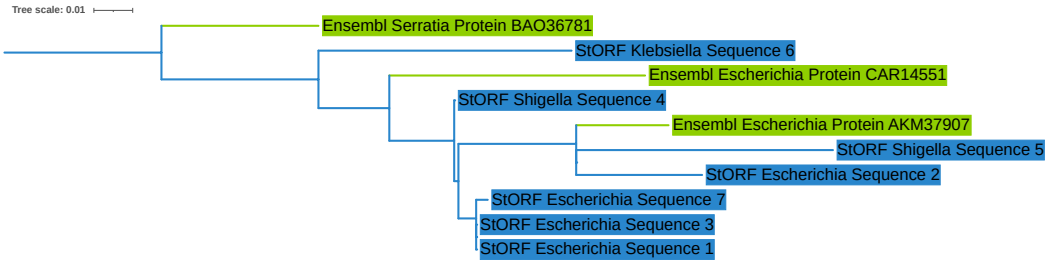


FIGURE 3.12: This is a phylogenetic tree built from the amino acid sequences of combined Cluster 124,470. This cluster consists of three Ensembl cluster representatives (clusters 1,013,244, 5,364 and 382,322) and the seven StORF sequences which clustered to those representatives. This tree was created using ClustalO, Fasttree and rooted at the Ensembl_Serratia_ Protein_BAO36781 sequence.

FIGURE 3.13: ClustalO multiple sequence alignment from the amino acid sequences of combined Cluster 124,470. This cluster consists of three Ensembl cluster representatives (clusters 1,013,244, 5364 and 382,322) and seven StORF sequences.

As seen in the types of clusters reported in Table 3.15, StORF-Reporter was also able to identify a number of clusters containing only StORF sequences that spanned many different genera. The biggest of these, combined cluster 15,543, spans 22 genera and consists of 81 genomes and 92 sequences. At 380 amino acids long, the representative of this cluster had a 90% percent identity and 55% query coverage to the 'fatty-acid-CoA ligase fadD18' protein in GenBank. This large, widespread protein was mostly observed as a single gene in each genome but in 5 species it was found duplicated with between 2 and 6 copies in non-contiguous URs.

The clusters that were found in at least two or more genomes were investigated for COG functional categories using the EggNOG-Mapper tool. The COG functions assigned to the cluster representatives are presented in Table 3.16. Although the low proportion of StORF-Only clusters with a COG annotation (2.97%) does limit these results, the more than 10% COG category changes for 'METABOLISM' and 'INFORMATION STORAGE & PROCESSING' compared to Ensembl-Only, are still informative of the functional differences between the two groups. As the COG category 'INFORMATION STORAGE & PROCESSING' was reported with the largest proportional increase for and Chi squared statistic, this specific COG group was further investigated in Table 3.17. While the low number of StORF-Only COGs does limit these results, there is still a clear upwards shift to 'Replication, recombination and repair' (L). This is interesting as genes assigned with this COG are often perceived to be essential.

COG Group	Ensembl-Only [%]	Ensembl-StORF [%]	StORF-Only [%]
INFORMATION STORAGE & PRO'	173,709 **[18.41%]**	17,776 **[24.28%]**	10,899 **[30.58%]**
CELLULAR PROCESSES & SIG'	229,026 **[20.86%]**	15,855 **[21.65%]**	7,189 **[20.17%]**
METABOLISM	261,742 **[39.41%]**	22,741 **[31.06%]**	10,028 **[28.14%]**
POORLY CHARACTERIZED	139,526 **[21.33%]**	16,855 **[23.02%]**	7,525 **[21.11%]**
With COGs/Total Sequences	1,353,421/1,573,739 **[86.00%]**	67,606/128,261 **[52.71%]**	33,178/1,118,600 **[2.97%]**

TABLE 3.16: The COG functional categories assigned to Ensembl-Only, Ensembl-StORF and StORF-Only cluster representative sequences with EggNOG-Mapper for the inter-genera analysis. Some sequences were observed with more than one COG functional category. In these instances, the sequence is only counted once in the 'With COGs/Total Sequences' column but each individual COG is counted separately for the 4 groups. Clusters are reported here irrespective of whether they were extended into new genera by StORF sequences. Chi squared statistic tests reported a p-value of <0.00001 for Ensembl-Only compared to both Ensembl-StORF and StORF-Only. As identified in the *E. coli* analysis, the POORLY CHARACTERIZED and INFORMATION STORAGE & PROCESSING' categories were reported with the highest chi-square statistic in each comparison, respectively.

COG Function	Ensembl-Only **[%]**	StORF-Only **[%]**
[J] Translation, ribosomal structure and biogenesis	69,641 **[25.77%]**	1,054 **[9.67%]**
[A] RNA processing and modification	332 **[0.12%]**	10 **[0.09%]**
[K] Transcription	125,354 **[46.39%]**	2,354 **[21.60%]**
[L] Replication, recombination and repair	74,399 **[27.53%]**	7,477 **[68.60%]**
[B] Chromatin structure and dynamics	488 **[0.18%]**	4 **[0.04%]**

TABLE 3.17: Presented here are the COG functions assigned to Ensembl-Only and StORF-Only cluster representative sequences for the group 'Information Storage and Processing'. While the number of Ensembl-Only sequences which obtained a COG classification are much higher than for StORF-Only, the reported COG categories are similar in both. Both 'A' and 'B' are observed in very low proportions (0.12%, 0,09% and 0.18%, 0.04$ for Ensembl-Only and StORF-Only respectively). The proportion of StORF-Only sequences with 'K' was less than half that of Ensembl-Only sequences, but 'L' was reported nearly two and a half times more, possibly hinting at a functional overview of missing gene function from canonical genome annotations.

3.5 Discussion

3.5.1 The Unannotated Regions of Prokaryote Genomes

The understanding that there is a great deal of work yet to be done in the area of prokaryotic genome annotation was reaffirmed in the first two chapters of this thesis. This work endeavored not only to report the current state of genome annotation, but also to identify specific weaknesses and highlight possible avenues to be targeted for improvement. A number of known but often overlooked aspects of genomics, such as the distribution of out of frame stop codons (Tse et al., 2010) and the potential for alternative use of start codons across prokaryotes, were used to direct the development of a homology-free method of gene discovery to further compliment our current genomic knowledge. Additional genomic features such as the high level gene density observed in canonical prokaryotic annotations have been "perceived as evidence of adaptive genome streamlining" (Sela, Wolf, and Koonin, 2016), but are at odds with the observations that many parokoryote genomes contain large numbers of long URs (see Tables 3.10 and 3.14). As shown in this chapter (see sections 3.4.3 and 3.4.3 for detail), the six model organisms exhibited thousands of URs which were often overlooked and contain genes which are missed by state-of-the-art gene prediction methods and are also missing from the canonical Ensembl annotations. We can only speculate as to whether these genes identified by StORF-Reporter are in fact expressed. Considering experimental evidence of expression in the model organism *E. coli* is missing from large portions of their gene collections (Ghatak et al., 2019), it will likely require a large community effort.

This work has clearly identified a number of novel CDS genes not in the current genome annotations. It is difficult to know whether the proportion of the StORFs identified in this study are a result of the current ineffectiveness of genome annotation methods or in fact genes which are no longer expressed due to mutation, possibly of the upstream region (putative pseudogenes). However, evidence that we are finding real genes can be seen in the 'Intra-Genome' results in Table 3.8; StORF-Reporter identified the presence of duplicates of genes already present in the Ensembl annotations. While pseudogenes are often ignored, or worse, misnamed as 'junk DNA', even in human disease research (Pink et al., 2011), their presence is important and their absence from annotation is limiting to future studies. This is clear considering the vast numbers of clusters identified with high numbers of additional StORFs sequences which have been identified with high levels of sequence identity to Ensembl annotation genes across many different genera and with high levels of sequence identity to SwissProt proteins and functional COG groups. However, as these genes are already annotated elsewhere and are most likely examples of incomplete genome annotation, it could be argued that the most important findings of this chapter were in fact the StORF only clusters which did not have any identity to any

known genes. Interestingly, the stop codon abundances and usages presented in Tables 3.7 and 3.9, suggest that StORFs without homology to known CDS genes may be at least partially validated with stop codon usage analysis. As coding and non-coding regions of prokaryote genomes are believed to evolve separately, reacting to intra- and inter-genomic pressures independently (Rogozin et al., 2002), SNP and mutation rate studies could be used as evidence in determining whether these are still functional genes.

Many novel StORFs formed clusters which spanned more than 10 genera and as such are prime examples of the current level of omissions in genome annotation. As such, irrespective of whether they are CDS sequences or other more cryptic genomic elements, they are nonetheless target sequences that future developments in genome annotation and experimental work should investigate.

3.5.2 The 'Stop - Open Reading Frame'

The concept of a StORF, developed through the work undertaken throughout the first two chapters of this thesis shows that despite all the advancements in computational biology, there are still clear problems in how genome annotation is undertaken. The findings of Chapter 2 confirmed that genome annotation methods in use today such as Prodigal (Hyatt et al., 2010) accurately detect the majority of CDS genes (in genomes which use the 'universal' codon table). However, in general, while all these tools overpredicted the number of genes, there was still large (>=10-20%) portions of the genomes studied were without any annotation (see the method aggregation study, section 2.4.7).

The specific use of stop codons to identify CDS genes dates back to the early days of automated gene prediction where stop-to-stop regions were identified and then studied for 'signal' (Borodovsky and McIninch, 1993). While it could be thought that StORFs are exceedingly common in prokaryotic genomes, it has been long known that out-of-frame stop codons (OSC) appear across prokaryote genomes (both intergenic and genic) more often than GC content and nucleotide abundance would confer alone (Tse et al., 2010). These OSCs provide a mechanism of early termination of translation in incorrect reading frames so that the metabolic cost and potential toxic bi-products associated with frameshift events can be reduced. This function of OSCs may have been selected for during the course of genome evolution to act against unintended frameshift occurrences. As such, the higher levels of stop codons observed in out-of-frame genic, intergenic and UR regions, lends further credence to the gene-length StORFs found in them.

As discussed in the introduction to this chapter, while some intergenic DNA in any prokaryotic genome is to be expected, the levels observed in the canonical annotations were at odds with the evolutionary theory of genome streamlining (Lynch,

2006). Additionally, the 'random' mutation which happens to non-conserved genomic elements would most likely be visible when presented on a phylogenetic tree. However, as could be seen in Figure 3.12, the distribution and position of the StORFs which were not separated from the canonical Ensembl sequences indicate a level of conservation of these StORFs sequences. It can be assumed, therefore, that as these StORFs are unlikely to have undergone 'random' or deleterious mutation, they represent functional CDS genes in 'canonical URs'. Therefore, not only was there a need for a tool to improve upon contemporary annotations, but it was also a fundamental requirement that the URs be the target for investigation. Subsequently, through the development and use of the StORF-Reporter methodology, the established doctrine of what intergenic DNA is, has been undermined in this chapter.

3.5.3 Identifying the True Intergenic Regions

Clearly, the aim of this work was not to present yet another tool to annotate genomes, but instead a tool to be used in conjunction with other tools. As such the URs and StORFs identified by StORF-Reporter across the 179 genera from Ensembl Bacteria, can be interpreted and used in a number of different ways. However, once putative StORFs have been identified, the remaining regions in the collection of URs for any one genome are not without significance. For example, if we consider the interaction between bacteriophages and their hosts in isolation, it could be imagined that this continuous flow of genetic material, irrespective of whether it is coding or not, can be indicative of both the host and host-environment history (Hendrix, 2003). More recent studies have shown that it is not only phage-related and broken genetic material being passed, but also functional genes involved in host-related processes such as antibiotic resistance (Colavecchio et al., 2017). The genesis of this genetic material and the mutations it accumulates can help decipher both past and future organism interactions. Traditional methods of homology searching do not work well here as many of these genomic elements have either never been characterised or are in too fragmented state. However, the use of **UR-Extractor.py** does make it easier to study these cryptic regions as the resulting regions after the extraction of StORFs, could be classified as close to 'true IRs'. Further to this, these true IRs, should be the target for future studies aiming to complete prokaryote genome annotations. Lastly, the investigation of URs is limited by a number of factors, which have not been investigated exhaustively here but must in any case be acknowledged in any study using the StORF-Reporter methodology.

3.5.4 Supplementing Contemporary Annotations

The annotations provided by Ensembl Bacteria and Prodigal for the six model organisms (see Table 3.2) can be taken to represent the pinnacle of the current state of prokaryotic genome annotation. To compare traditional genome annotation methods such as Prodigal to that of StORF-Reporter is an unfair comparison. Furthermore, their intended purposes are at odds. Prodigal is a complex, multifaceted tool which uses a number of advanced computing techniques and involves the building of its own model trained on every genome sequence with which it is presented. This is so that it can adjust its parameters according to the features such as GC specific to each genome sequence. Additionally, Prodigal predicts genes for the entire length of the target genome.StORF-Reporter on the other hand is a one size fits all tool which not only specifically studies URs, but also has parameters that are designed to detect the types of genes routinely missed by other tools (short, overlapping and genes which use alternative start codons). Therefore, as supported by the results of sections 3.4.2 and 3.4.3 which investigate StORF-Reporter's ability to identify genes not annotated by Prodigal or Ensembl, it is likely that one of the most beneficial ways to use StORF-Reporter is to generate additional (and likely missed) annotations for a genome (a 'GFF+', discussed in the introduction). With the GFF_Adder tool, described in the methods section 2.3.5 of Chapter 2, StORFs can be used to supplement existing genome annotations. The *E. coli* pangenome study clearly presents the opportunity for this process to deliver improvement to contemporary annotations without sacrificing the hard work already undertaken. For example, while the *E. coli* genomes presented in the paper "A comprehensive and high-quality collection of Escherichia coli genomes and their genes" (Horesh et al., 2021) have undergone extensive curation they still contain URs capable of holding genes. Reannotating these genomes would undoubtedly lose much of the work and knowledge used by the authors in the process. Therefore, the ability to investigate only those regions of DNA without annotation and then combine the curated annotations with the additional StORF sequences is a clear advantage over traditional methods of reannotation.

3.5.5 Extending Pangenomes

It has been known for some time that a single genome is not enough to characterise the functional profile of a species as this speed and scope of prokaryotic speciation continues to highlight the need for the study of pangenomics. The *E. coli* pangenome could be argued to be the most studied and characterised of any prokaryotic organism (Rasko et al., 2008). One recent study estimated the pangenomic gene collection at more than 13,000 genes which clearly has "tremendous implications in terms of the diversity and pathogenesis of the species *E. coli* and its ability to colonize and cause disease in the human host." (Rasko et al., 2008). However, the three studies of *E. coli* URs and StORFs in this chapter have shed light on the potential for yet more diversity of not only accessory but also core genes.

> *"The essence of the species is linked to the core genome. However, the majority of the genetic traits linked to virulence, capsular serotype, adaptation, and antibiotic resistance pertain to the dispensable [accessory] genome. Therefore, sequencing [and therefore annotation] of multiple strains is necessary to understand the virulence of pathogenic bacteria and to provide a more consistent definition of the species itself." (Tettelin et al., 2005).*

In the 6 model organism study, *E. coli* StORFs reported the highest levels of sequence identity to either SwissProt or intra-genome proteins (see Table 3.8). Those results, and the high levels of StORFs found in Ensembl-StORF and StORF-Only clusters in the *E. coli* pangenome has highlighted the possible true number of coding and pseudogenised genes missing from the annotations of one of the most studied organisms. The importance of these potential inter-genome genes is yet to be fully explored. While numerous theories could be deduced from their presence and distribution, fundamentally, they are likely candidates for the study of how a species' pangenome can adapt to ever-changing environments (Rasko et al., 2008). These StORFs, functional or pseudogenised genes or not, are additional genetic blueprints an organism continues to hold onto (or is in the process of losing), against the pressures for genome streamlining. Additionally, some of these StORF-Only clusters were observed with higher levels of the COG group 'Information Storage & Processing', compared to Ensembl-Only (as seen in Table 3.12), traditionally which have thought of as being core functions. Investigating this COG group specifically, there is another shift to the COG function 'Replication, recombination and repair' for StORF-Only clusters (see Table 3.13). Interestingly this COG function covers genes involved in a number of essential processes and as such suggests the importance of a number of the StORF-Only clusters missing from the Ensembl genome annotations.

As with many of the results of this work, StORFs continue to present a complicated picture of their presence across the *E. coli* pangenome. Fundamentally, while the identification of these StORFs have provided additional genomic knowledge previously left in the dark corners of their genomes' URs, to truly utilise the nascent information they may harbor, much expanded experimental work is needed.

3.5.6 Extending Intra and Inter Genera Gene Collections

The results of the StORFs reported in the six model organisms and across the *E. coli* pangenomes have shown wide levels of diversity and spread. However, it could be possible that many of these StORFs are artifacts of genome assembly and species-specific structural elements and not primarily CDS genes. Therefore, the inter-genera analysis of the 6,223 genomes from Ensembl Bacteria was key to not only studying the presence of StORFs across further diverse species, but to also provide evidence to further validate them as putative CDS genes. An EggNOG COG functional analysis of the inter-genera results identified that over half of the Ensembl-StORF clusters had a COG function annotation (52.71%) comparable to the Ensembl-only

clusters (86.00%, see Table 3.16) suggesting the StORFs in these clusters were real (but missed) genes. Interestingly, even though only 2.60% and 2.97% of the *E. coli* pangenome and inter-genera StORF-Only clusters (respectively) had COG annotations, the most common functions in both, were related to 'Information Storage & Processing'. Further to this, from that COG function, in both the inter-genera and *E. coli* pangenome StORF-Only clusters, it was 'Replication, recombination and repair' (L) which was the most common COG function. StORFs were found across many different genera at high sequence similarities (>= 95%) (see Table 3.15). As with the *E. coli* pangenomic study, many StORFs from the same cluster were found in different regions across the genomes they were found in, or in the case of duplication, in different regions of the same genome. This adds to the evidence that StORFs can capture real CDS genes (even those without functional annotations). The presence of StORFs, sometimes duplicates found within the same genome but in different URs vastly spaced apart, adds credence to the likelihood that they are not structural elements or other genomic artifacts but instead functional (past or present) CDS genes. Fundamentally, this comparison clearly shows that not only are StORFs capturing known functional CDS genes, but also that their function may be unique to species.

As such, the observed variances in lengths of StORFs clustered together within and across the 179 genera in this study provide further justification of the need for species-agnostic gene prediction methodologies which do not rely on established model organisms or databases. Discovering the reason for their omission may help future tool advancements. However, the inadequacies in genome annotation tools and how they are used, as found in Chapter 2, no doubt have a part to play. While these results may not fundamentally change the understanding of and structure of pangenomes or functional profiles of any one species, they could drastically change the landscape of the functional and phylogenetic interconnections between different genera.

3.5.7 Are Some StORFs Pseudogensied Genes?

The potential diversity of all possible protein sequences is vast (Alberts et al., 2002). A typical protein length of about 300 amino acids can have more than 20^{300} different possible polypeptide chains, yet current protein databases only contain a tiny fraction of this, such as SwissProt, which contains a little over 500,000 protein sequences. Many of these are duplicates or very similar to each other and many of these are not experimentally validated. Adding to this complexity are the vast numbers of pseudogenes, which were likely functional proteins in the recent past, identified through the extensive genome sequencing of the last few decades. While mutation rates have long been studied in prokaryotes (Luria and Delbrück, 1943; Rosche and Foster, 2000), there are still unresolved questions surrounding the speed at which pseudogenes are created and subsequently removed from the genome. Some studies have shown that pseudogenes are more likely to be a result of failed horizontal

gene transfer (Liu et al., 2004). Additionally, while archaea and most non-pathogenic bacteria exhibit greater retention of ancient gene remnants, obligate pathogenic bacteria tend to have younger pools of pseudogenes (Liu et al., 2004). As such, the types of genes pseudogenised and the rate at which they are mutated are highly variable between species and likely are related to environmental specific cues.

However, as outlined in Chapter 2, current methods fail to address this problem, therefore, the true extent of possible pseudogenisation through mutation is likely to remain unclear for the foreseeable future. The StORF-Reporter methodology has the potential to address this issue. As a StORF reports potential ORFs regardless of the likelihood of them being functional and it captures a portion of the 5′ untranslated region of a CDS gene, it also has the potential to capture pseudogene-creating mutations, such as start codon or ribosomal binding sites mutations. As shown in Figure 3.14, which was originally reported in Current Opinion in Microbiology by Goodhead *et al* (Goodhead and Darby, 2015), of the 6 different process that a gene can be pseudogenised, a StORF has the potential to capture all of them. The most easily identified is when pseudogenisation occurs because of a premature stop. While a single StORF could only report one fragment of a gene pseudogenised by a premature stop codon, the combination of multiple StORFs has the potential to recover the entire sequence of the pseudogenised gene (this is investigated in further detail in Chapter 4).

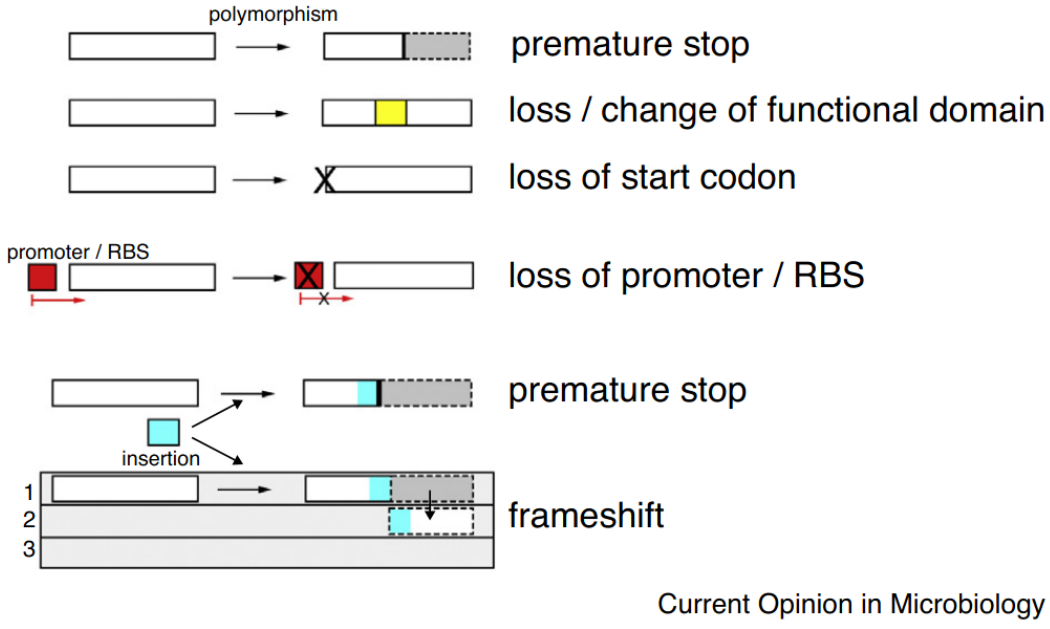


FIGURE 3.14: This figure, originally reported in Current Opinion in Microbiology by Goodhead *et al* (Goodhead and Darby, 2015), depicts a collection of currently understood methods of gene pseudogenisation. The third category, loss of start codon, shows the type of gene pseudogenisation which may be captured by StORFs. (Elsevier license number: 5182450343370)

3.5.8 Conclusion

Genome annotation continues to be a developing field. With each new sequencing project, the proportion of uncharacterised sequences found can vary significantly depending on habitat (Zhang et al., 2020b), but often hundreds of novel genomes are discovered with little identity to previously sequenced taxa (Kowarsky et al., 2017) suggesting that there is much gene diversity yet to be discovered. Exacerbating this issue (as discussed in the 1 and 2 of this thesis), both historic and systematic biases and errors mean that some types of genes will always be missed in these studies, resulting in many tools reporting the same untranslated regions in genomes. This might lead users to the false security of consensus of methods, but as we have demonstrated in this chapter, the StORF-Reporter methodology can enhance current annotations, reporting missed ORFs in these regions.

One major limiting factor in our analysis in Chapter 2, was the use of Ensembl genome annotations as a ground truth (under the assumption that they are a 'complete reference'). In this chapter, we challenged that assumption of completeness through the use of StORF-Reporter to investigate the URs in over 6 thousand Ensembl annotations. In direct contradiction to the idea of genome annotation completeness, a high number of UR StORFs were homologous to known proteins. It is true that the use of the term 'complete' in regard to genome annotation often incurs a level of uncertainty in regard to the annotation of peripheral or accessory genes. However, the relatively high number core genes identified by StORF-Reporter, is simply not compatible with that uncertainty. Furthermore, StORFs without homology to Ensembl proteins, were often found across multiple genera with similar predicted functional profiles to "known" genes. While both these groups of StORFs are therefore prime candidates to be added into their respective genome annotations, the majority of StORFs across all genomes were without homology to known proteins across any database examined. The processes which influence the emergence, distribution and subsequent loss of both novel and modified genes such as gene duplication, neofunctionalisation, 'genome streamlining' and purifying selection (Ohno, 2013; Levasseur and Pontarotti, 2011; Giovannoni, Thrash, and Temperton, 2014), suggests substantial pressure to 'lose' non-functional gene sequences. These factors make the resulting sequences which do persist (especially those found across multiple genera), regardless of homology to known genes, more likely to be functionally important and/or active. Accessory genes, those most likely to be captured by StORFs (See Table 3.11), are now thought to often carry out important and essential functions (Sela, Wolf, and Koonin, 2016), often for adapting to environmental niches and changes (Jiao et al., 2018). Investigating this relationship will be a priority for future work.

Chapter 4

StORF-Reporter Reveals General Misconceptions of 'Stop Codons' Across Prokaryotes

4.1 Chapter Summary

In the previous chapter, over six thousand prokaryote genomes from Ensembl Bacteria were shown to exhibit high numbers of Unannotated Regions (URs). Through the development and use of StORF-Reporter, I was able to identify missing genes across a wide selection of these prokaryotic URs. Many of the genes identified from these URs, exhibited high levels of sequence similarity to full-length functional genes from within the same set of genomes and to the Swiss-Prot protein database. However, many were reported as fragmented hits, only aligning partially to known protein sequences. This led to the idea that a proportion of Stop-ORFs (StORFs) from any one genome may represent recently pseudogenised genes.

There are many processes which can facilitate gene pseudogenisation and as with missed genes, the identification of pseudogenised genes is an important step in expanding our knowledge of prokaryotic genomics. Therefore, to capture these, StORF-Reporter was extended to enable the discovery of one class of pseudogenised gene, fragmentary gene sequences which feature deleterious mutations in the form of internal in-frame stop codons, named 'premature stop' in Figure 3.14. These 'Consecutive-StORFs' (Con-StORFs) follow the same parameters as StORFs but are found consecutively connected together by their in-frame stop codons, and are shown in Figure 4.1.

I report putative pseudogenes forming novel core gene families in the *E. coli* pangenome and extending known and novel gene families across a wide range of prokaryotic genera. Pseudogenised genes like these may contain invaluable information regarding the recent evolutionary history of their genomes, exposure to environmental changes, other species interactions, and genome reorganisation.

This chapter therefore follows closely on from the work of Chapter 3 and builds upon the StORF-Reporter platform, available at `https://github.com/NickJD/StORF-Reporter`. As such, much of the methodology has already been described and so in this chapter I have only presented the parts of the methods which differ from Chapter 3.

4.2 Introduction

Pseudogenised genes, defined as once functional genes that have acquired mutation resulting in loss of function, are important markers of their genome's history, environment change, genomic interaction and organisation (Goodhead and Darby, 2015). Their impact is still currently under investigation and while they have been associated with complex multi-faceted processes such as pathogen-host adaptation, further study is required (Langridge et al., 2015). Further to this, the fluidity of prokaryotic gene content, even between strains (Van Rossum et al., 2020), is a strong indicator that the genomic graveyard, the genomic regions comprised of pseudogenised genes, may have consequences we are yet to fully understand. Moreover, pseudogenised genes have been found to mutate at different rates (both negative and positive) compared to other nonfunctional genomic regions, further suggesting that their loss is at odds with the notion of neutral selection of non-functional elements (Kuo and Ochman, 2010).

Pseudogenes have been studied in both eukaryotes and prokayotes for decades and yet only some of the processes behind gene pseudogenisation are currently known, mostly from observation after the fact (Petrov and Hartl, 2000; Mahmudi et al., 2015; Avni et al., 2018; Chu et al., 2021). Further to this, pseudogenes are inherently difficult to validate without expression analysis and as mutation rate differences have been observed between strains of the same species, an almost unlimited search space which to identify pseudogenes exists (Elena et al., 2005). Additionally, due to the industrial and clinical importance of prokaryotes, a large number of experimental studies into their metabolic processes has been undertaken and as such, there exists a bias for these types of functions, potentially leading to an overrepresentation of pseudogenes annotated with these functions (Valdés et al., 2008). Combined with the fact that pseudogenes are inherently less likely to have high-quality homologous genes in genomic repositories, the technical hurdles involved have made pseudogene classification a daunting task (Karro et al., 2007; Alves et al., 2020).

> *"The dominant limitation in advancing the investigation of pseudogenes now lies in the trappings of the prevailing mindset that pseudogenic regions are intrinsically non-functional." (Cheetham, Faulkner, and Dinger, 2020).*

While contemporary genome annotations provide a snapshot of a single organism's functional gene set, pseudogenised genes are often not reported by contemporary genome annotation prediction tools and there is currently little progress in identifying them without homology searches. The misreporting and lack of pseudogenised genes in genomic annotations, can have wider effects than just single-genome gene composition. Pangenomes are often studied in a binary manner - genes are either present or absent. Through genus-wide pseudogene detection, it will be possible to gain a better understanding of the evolutionary avenues undertaken by

subsets of individuals from a species. Pseudogenised genes can give deeper insights into and understanding of the history of both the individuals and the genus they are shared across.

Previously in Chapter 3, the unannotated regions (URs) of 6,223 prokaryotic genomes were investigated to find CoDing Sequences (CDSs) which had been missed by contemporary genome annotation techniques. Here I investigate the ability to systematically identify putative pseudogenised genes from the URs of prokaryotic genomes. As StORFs are primarily designed to capture missed genes due to incorrect genome annotation, they are able to identify pseudogenised genes, specifically those which have undergone 5' region mutations such as promoter loss or start codon mutation. However, pseudogenised genes are not the primary target of StORFs and the quantity of StORFs detected from each genomes' set of URs is substantial and these are often complete CDS gene sequences simply missed by the previous annotation. The aim of this work is to target the identification of one specific type of pseudogene, those pseudogenised by in-frame stop mutation, in order to better understand their diversity and distribution. Additionally, the specificity of Consecutive - Stop Open Reading Frames (Con-StORFs), which have been designed to specifically search for psueodogenised genes broken via in-frame stop codon mutations, decreases the potential for false positives compared to that of StORFs. This is necessary as one of the central aims of this work is to present putative pseudogenised genes, those which are more likely to require a level of scrutiny that functional genes do not. Further to this, Identifying an in-frame stop codon mutation is a computationally feasible task without the need for wet-lab or homology based analysis.

Through this work, a novel method has been developed to recover pseudogenised CDS gene sequences from the URs of canonical genome annotations. This method can be applied to a standard genome annotation in GFF format, as provided by Ensembl Bacteria. The resultant sequences identified by Con-StORFs can form the basis of a number of different studies, however the cross-genus pangenomic graveyard, which is defined here as the collection of pseudogenes shared within and across genera, may allow for novel insights into a genome's evolutionary geography and 'timescape'.

4.3 Methods

4.3.1 Data Preparation

The same data preparation processes were applied here as were used in Chapter 3 subsection 3.3.1. Therefore, in the following sections it is only explained when the process of data preparation differs.

Table 4.1 originally from Chapter 3 is presented for ease of reference.

Model Organism	Genome Size (Mbp)	Genes **[Density]**
Bacillus subtilis ***(B. subtilis)*** BEST7003	4.04	4,133 **[88.91%]**
Caulobacter crescentus ***(C. crescentus)*** CB15	4.02	3,875 **[90.60%]**
Escherichia coli ***(E. coli)*** *K-12 ER3413*	4.56	4,257 **[86.28%]**
Mycoplasma genitalium ***(M. genitalium)*** *G37*	0.58	559 **[92.03%]**
Pseudomonas fluorescens ***(P. fluorescens)*** *UK4*	6.06	5,266 **[84.75%]**
Staphylococcus aureus ***(S. aureus)*** *502A*	2.76	2,556 **[83.93%]**

TABLE 4.1: An overview of genome composition for the 6 model organisms selected to evaluate StORF-Reporter compiled from data held by Ensembl Bacteria. The number of Ensembl annotated genes (coding and non-coding) is reported and the genome density is in bold square brackets. Note the relative differences in genome size (0.58 - 6.06 Mbp) and gene density (percentage covered with annotation, 83.93% - 92.03%).

4.3.2 Consecutive – Stop Open Reading Frames

Consecutive-Stop Open Reading Frames (Con-StORFs) are Stop Open Reading Frames (StORFs) which are consecutively connected together by an internal in-frame stop codon. This connecting together of two or more StORFs allows for the identification of more than just complete CDS genes, but specifically CDS genes with in-frame stop codons and thus were only partly reported by StORFs in Chapter 3.

This subsection focuses on the development and overview of an extension to the StORF-Reporter methodology described previously in Chapter 3. Specifically, an additional process was implemented within the StORF-Finder subroutine to report Con-StORFs. As can be seen in Figure 4.1, Con-StORFs are designed on top of the StORF concept to target the capture of genes that are unannotated due to pseudogenisation through in-frame stop codons or alternative codon usage (see sections 1.3.1.2 and 2.5.5 for further detail). While the **UR-Extractor.py** program was unchanged, two user-definable parameters and subsequent processes were implemented in **StORF-Finder.py**. As can be seen in the menu system in Appendix

Figure B.2, '-con_storfs' and '-con_only' now enable a user to report Con-StORFs both alongside StORFs and in isolation. The reporting of Consecutive-StORFs (Con-StORFs) follow the same parameters as StORFs but are consecutively connected by their in-frame stop codons, as shown in Figure 4.1.

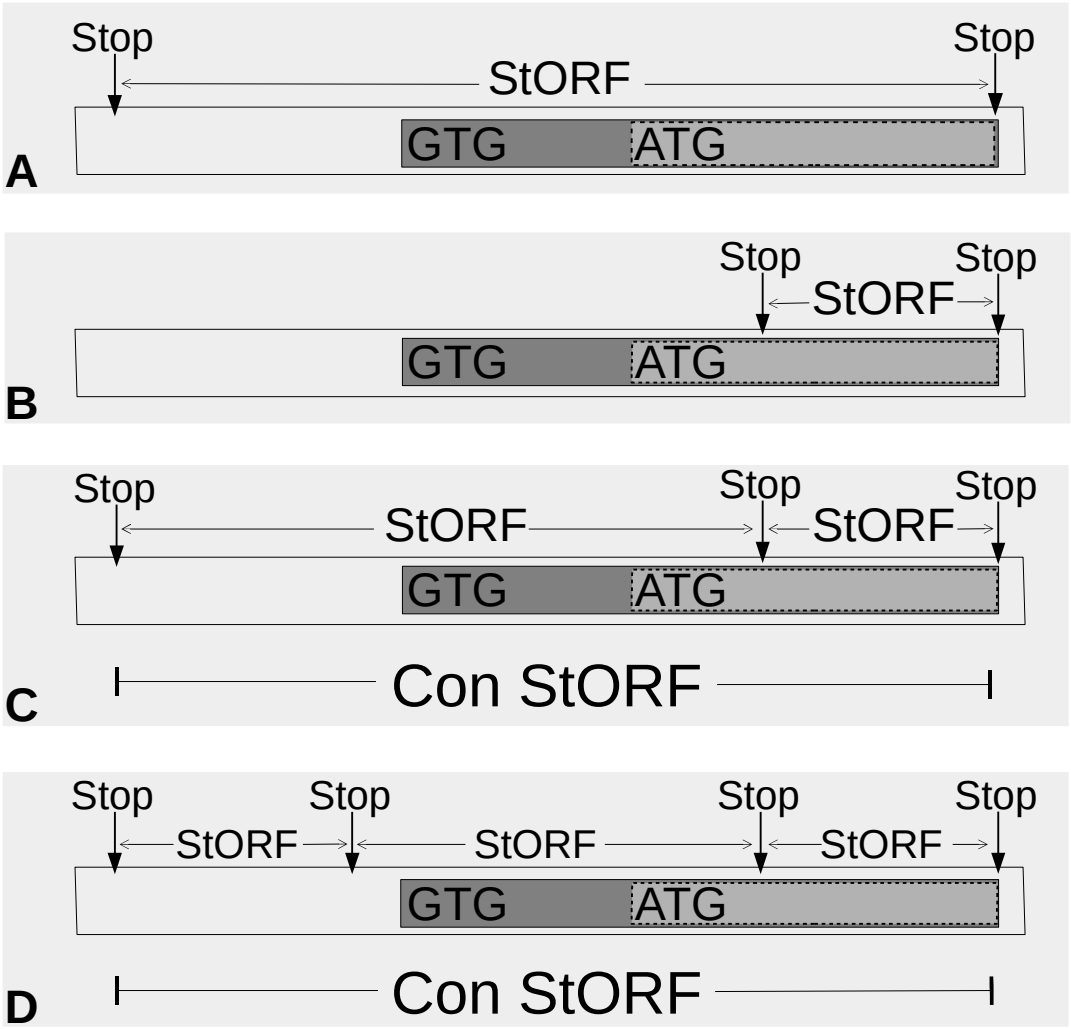


FIGURE 4.1: Visual representation of a Con-StORF and how it can capture multiple potential start codons for a single gene in an unannotated region. While a StORF can consist of only a partial segment of a gene if that gene either recodes a canonical stop codon or has had an in-frame stop codon mutation, Con-StORFs can capture the additional segments of the sequence. Image A depicts a StORF capturing the two possible start positions/codons for a gene and image **B** shows how a StORF can comprise of only a partial segment of a gene if that gene either recodes a canonical stop codon or has had an in-frame stop codon mutation. Image C depicts a Con-StORF capturing a gene which has an in-frame stop codon and image D is an example of how not all of the internal StORFs of a single Con-StORF may capture a gene.

4.3.3 Additions to the Reporting of COG Functional Categories

Further to the EggNOG-Mapper functional analysis descried in Chapter 3 Subsection 3.3.5, it was discovered that there were 'newly' added Mobilome' (X) and 'Defense mechanisms' (V) EggNOG COG categories (Galperin et al., 2019) which are not reported. However, as the output of EggNOG-Mapper reports the COG identifiers, such as COG0675 for 'Transposase', the list of X and V assigned COG identifiers were downloaded from `https://www.ncbi.nlm.nih.gov/research/cog` and were used to record the sequences assigned to each category.

4.4 Results

4.4.1 StORF-Reporter Finds Pseudogenised Genes Not Present in Ensembl Annotations

Model Organism	Number of Ensembl Genes	Number of Ensembl URs	Longest Ensembl UR Length	Median Ensembl UR Length **[SD]**
B. subtilis	4,133	2,711	1,407	226 **[137.79]**
C. crescentus	3,875	2,321	3,477	221 **[172.85]**
E. coli	4,257	2,743	6,275	243 **[353.37]**
M. genitalium	559	157	4,922	185 **[673.16]**
P. fluorescens	5,266	3,509	20,088	244 **[633.74]**
S. aureus	2,556	1,666	2,591	307 **[235.16]**

TABLE 4.2: This table, originally from Chapter 3, presents the result of running UR-Extractor on the Ensembl annotations for the six model organisms. Each UR is extended with 50nt at each end. All lengths in nt. Standard Deviation is reported as **[SD]**

.

The same URs of the six model organisms, according to their Ensembl annotations, extracted previously in Chapter 3 (see Table 4.2 for reference) were used in this section. The modified version of StORF-Finder, which was set to only report Con-StORFs, was applied to these URs. The number of Con-StORFs is much smaller than those of StORFs (see Tables 3.8 and 4.3). Each of the six model organisms has at least 1 Con-StORF with homology to either the Swiss-Prot protein database or proteins from the same genome. *P. fluorescens* and *E. coli*, had the first and second highest number of Con-StORFs with 356 and 199 respectively. Of these, 14 and 23 Con-StORFs obtained alignments to Swiss-Prot and 17 and 13 aligned to intra-genome CDS genes respectively. Further to this, *B. subtilis*, *C. crescentus* and *S. aureus* which themselves had 97, 103 and 103 Con-StORFS respectively, obtained very low numbers of Swiss-Prot or intra-genome alignments. Specifically, in the case of *S. aureus*, only 1 Con-StORF aligned to a Swiss-Prot protein. However, these results are not as clear as they may first seem. *M. genitalium* for example had the highest proportion of its Con-StORFs with alignments to either Swiss-Prot or intra-genome sequence (25

and 21 respectively, for its 32 Con-StORFs). Lastly, these results show that even with the targeted approach taken by the use of Con-StORFs in these 6 model organisms, which themselves constitute a collection of the most studied prokaryotic genomes, a number of interesting genomic elements within their URs continue to be uncovered.

Genome	Num of Con-StORFs	Swiss-Prot		Intra-Genome	
		Hits	Coverage >=80%	Hits	Coverage >=80%
B. subtilis	97	6	4	3	1
C. crescentus	103	2	1	1	1
E. coli	199	23	20	13	4
M. genitalium	32	25	6	21	0
P. fluorescens	356	14	11	17	11
S. aureus	103	1	0	2	1

TABLE 4.3: Table containing the number of Con-StORFs found in the URs recovered from Ensembl annotations for six model organisms. The numbers of Con-StORFs which had a high sequence similarity and >=80% subject hit to a protein in Swiss-Prot and Ensembl proteome is listed.

The stop codons used as the internal dissecting codons of each of the reported Con-StORFs are reported in Table 4.4. While each of the 6 model organisms do seem to exhibit at least one preferred codon, it is interesting that the genome with the lowest number of Con-StORFs is presented as the only one with a clear preference, TGA for *M. genitalium*. Also reported are the numbers of 'Multi Con-StORFs', which are Con-StORFs consisting of more than one internal stop codon.

Genome	Con-StORFs **[Multi]**	Internal Stops		
		TGA	TAG	TAA
B. subtilis	97 **[6]**	51	8	44
C. crescentus	103 **[6]**	61	25	23
E. coli	199 **[11]**	76	43	94
M. genitalium	32 **[12]**	38	4	2
P. fluorescens	356 **[39]**	184	108	117
S. aureus	103 **[4]**	24	22	62

TABLE 4.4: Presented here is the stop codon usage for the in-frame stops of the Con-StORFs for each of the 6 model organisms. In cases where there are more than one internal stop codon (**[Multi]**), the codons are counted individually.

Across the six model organisms, only one of these multi Con-StORFs was observed with sequence similarity to an existing gene. This 2,268 nucleotide, three internal stop codon Con-StORF, reported in *E. coli*, aligned in its entirety (except for the first 15 bp or 4 amino acids due to the first stop codon in the Con-StORF) to the Swiss-Prot protein "Putative uncharacterized protein YgaQ (YgaQ)". Intriguingly, as seen in the multiple sequence alignment presented in Figure 4.2, not only does the entire Con-StORF sequence align with YgaQ from Swiss-Prot, but also all three

of the internal stop codons (note red boxes) are reported as tryptophan (W) in YgaQ. Additionally, the Swiss-Prot record for YgaQ is listed with a number of 'Sequence cautions' due to "Erroneous termination. Truncated C-terminus". More interesting are the internal codons themselves which are all TAG and not TGA, as found in other examples of tryptophan-encoding stop codon reuse.

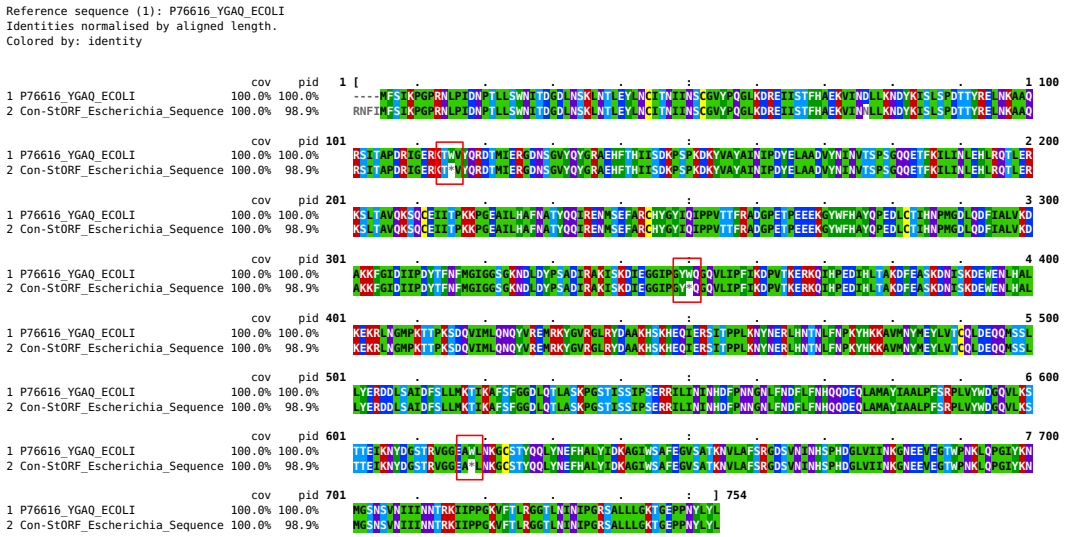


FIGURE 4.2: ClustalO multiple sequence alignment of a multi Con-StORF reported in *E. coli* which aligned to the Swiss-Prot sequence, "Putative uncharacterized protein YgaQ (YgaQ)". This alignment showcases the three internal stop codons (all TAG and highlighted in red boxes) identified by the Con-StORF, all coding for tryptophan (W).

Lastly, out of the 12 multi Con-StORFs reported in *M. genitalium*, one was of particular interest. This multi Con-StORF contained 2 internal in-frame stop codons but used TAA for the first, second and fourth stop codons and only used the 'expected' TGA codon once as the internal third codon. Unfortunately this sequence was not found to have similarity to either a Swiss-Prot or intra-genome protein in its reported coding frame. However, when using the 'blastx' option in DIAMOND to allow for all six coding frames to be translated, the sequence obtained an alignment to the Swiss-Prot protein "Uncharacterized protein MG288" in frame -2 (frame 6). This Con-StORF has presented a number of perplexing elements which, whilst being at odds with some of the assumptions made of what constitutes a Con-StORF, are nonetheless extremely interesting. The presence of three in-frame stop TAA codons to the sole TGA codon was in itself interesting in a *M. genitalium* genome. Next, in frame -2 which aligned to the Swiss-Prot protein, it was indeed TGA which was the internal stop codon which coded unsurprisingly for tryptophan in the aligned Swiss-Prot protein. Lastly, as can be seen in Figure 4.3, although the Con-StORF only aligned to 38.4% of the Swiss-Prot protein, the alignment was of a high quality with 99.4% identity.

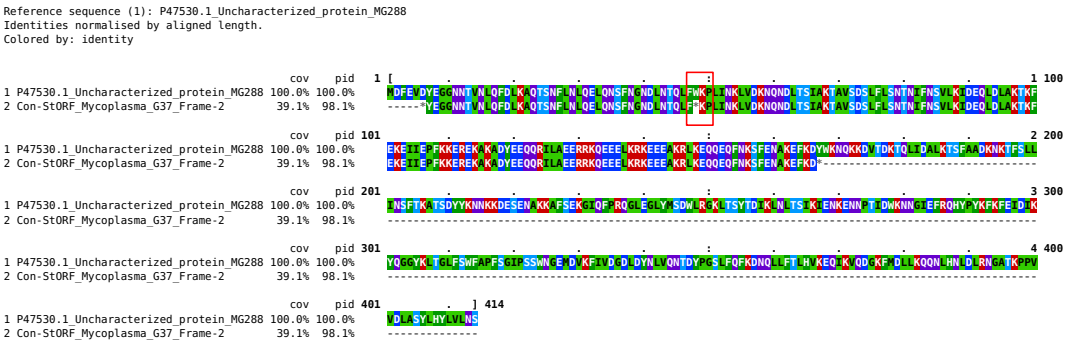


FIGURE 4.3: Clustal Omega multiple sequence alignment of a multi Con-StORF reported in *M. genitalium* which aligned to the Swiss-Prot sequence, "Uncharacterized protein MG288". DIAMOND blastx was used to align the Con-StORF DNA sequence's negative frame -2 (frame 6) to the Swiss-Prot protein as the reported frame reported no hit. Although the alignment coverage is low at 38.4%, the alignment was of a high quality with 99.4% identity. Highlighted in the red box, the in-frame stop codon of the Con-StORF was reported as the amino acid Phenylalanine in the Swiss-Prot protein.

4.4.2 *Escherichia coli* Historic-Pangenome

The investigation into the presence of pseudogenised genes across the *E. coli* pangenome began with the same set of URs as was used in the previous chapter (see Chapter 3, subsection 3.3.6 for reference). As seen in Table 4.5, 40,945 Con-StORFs were identified from within the 673,136 URs reported in the 219 *E. coli* genomes from the previous chapter. The median number of Con-StORFs for the genomes were significantly lower, 175 as opposed to 3,038 StORFs. Longest and median sequence lengths were also lower for Con-StORFs, 3,164 and 95 nt as opposed to 14,334 and 141 nt for StORFs respectively. The much reduced number of Con-StORFs reflect the more targeted objective of this addition to StORF-Reporter and subsequently the reduced search space though the specific requirement for consecutive in-frame stop codons, compared to those for StORFs.

Data	Unannotated Regions	Con-StORFs
Number of Sequences	673,136	40,945
Median Number Per Genome	3,038	175
Longest Sequence (nt)	45,683	3,164
Median Sequence Length (nt)	234	95
[SD]	**[292.97]**	**[110.11]**

TABLE 4.5: Presented here are the numbers and lengths of unannotated regions (URs) and Con-StORFs extracted from the 219 *Escherichia coli* genomes. While there was inconsistency in the genome quality across this set of genomes, the numbers reported here are similar to those reported for the 6 model organisms. Standard Deviation is reported as **[SD]**.

As presented in subsection 3.4.4 of Chapter 3, 34,737 gene clusters were identified from 1,042,068 Ensembl proteins, of which 20,676 were non-singletons. The representatives from these clusters were combined with 40,945 Con-StORFs identified from the same 219 genomes and CD-Hit (Fu et al., 2012) sequence clustering was applied. This resulted in a total of 41,903 clusters, of which 3,513 were non-singletons. The resulting clusters have been classified differently in respect to the sequences they have clustered and follows the structure put forward in Chapter 3, subsection 3.3.6. As such they will be referred to as follows:

- **Ensembl-Only**, which refers to the clusters with only Ensembl-annotated protein sequences.
- **Ensembl-Con-StORF**, which refers to the clusters which contain at least one sequence from both Ensembl-annotated protein sequences and one Con-StORF amino acid sequence.
- **Con-StORF-Only**, which refers to the clusters which solely contain Con-StORF amino acid sequences.

The CD-Hit clustering of the combined Ensembl and Con-StORF sequences produced 7,179 Con-StORF-Only clusters and 34,068 Ensembl-Only clusters. The reduction observed in the total number of clusters can be attributed to the much reduced number of Con-StORFs compared to StORFs, so we see a reduction from 51,929 StORF-Only clusters to 7,179 Con-StORF-Only clusters (see Tables 3.10 and 4.5 for comparison). Further to this, out of a total of 34,724 Ensembl gene families which clustered with Con-StORF sequences, 669 Ensembl gene families were extended by additional *E. coli* sequences, of which 650 were extended by at least one additional *E. coli* strain. Additionally, 29 gene families were extended to become part of the core gene group of $>= 99\%$ by the addition of Con-StORFs, and 3 core gene families were formed from only Con-StORF sequences. Investigating these 3 core gene families further, we find that they were the largest 3 clusters (clusters 0, 1 and 2) and were of gene-like length: 81, 113 and 140 amino acids. Additionally, the representative sequences of these 3 clusters were all reported with alignments to 'conserved hypothetical proteins' in other species (without in-frame stop codons) in the NCBI nr database. Out of the 7,179 Con-StORF-Only clusters, 2,794 consisted of Con-StORFs from at least 2 different genomes which overall spanned an average of 12 genomes. These Con-StORF clusters, consisting of entirely 'lost' sequences, could reshape what we know about *E. coli* and its pangenome.

The low number of Ensembl-Con-StORF clusters compared to Ensembl-StORF clusters reported in the previous chapter does make the resultant distributions difficult to compare. However, while there is a much more pronounced one-sidedness to the lower end (left) of the distribution of the Con-StORF-Only clusters (see Figure 4.4), it is the opposite for Ensembl-Con-StORF clusters. The distribution of Ensembl-Con-StORF clusters, while consisting of a much reduced number, is almost balanced between the lower and upper ends with a small bias towards the lower end of the distribution. This indicates that while there are far fewer Con-StORFs clustering with Ensembl cluster representatives, those which do are still likely to cluster either with Ensembl representatives from soft-core core gene families, or, through their addition, extend those clusters into soft-core and core gene families.

Once again taking genome assembly and annotation error into consideration, a small number of gene clusters that had been extended by the addition of Con-StORFs to become core gene families were reported across the 219 *E. coli* genomes (see Table 4.6). Interestingly, some clusters which fell below this level were extended into the 95% threshold by the addition of their Con-StORFs. There were also Con-StORF-Only clusters with sequences from more than 95% of the *E. coli* genomes, representing conservation of genes either missed by traditional annotation methods or which have been pseudogenised across all genomes. While there were 10 Ensembl-Con-StORF clusters which were formed with more than one Ensembl cluster representative, with 3 of these clustering 3 separate Ensembl representative sequences together, none of these were reported in the pangenome groups in Table 4.6. Lastly,

there were 19 Ensembl clusters which, while they did cluster with Con-StORF sequences, none of the Con-StORFs were from additional genomes and as such were not counted in the Ensembl-Con-StORF data.

Cluster Types	Core	Soft-Core	Accessory
Ensembl-Only	2,612	455	2,597
Ensembl-Con-StORF	29	9	10
Con-StORF-Combined-Ensembl	0	0	0
Con-StORF	0	0	4
Con-StORFs-Only	3	12	208

TABLE 4.6: *Escherichia coli* gene families calculated from the set of 219 strains can be extended by the addition of Con-StORFs (possible pseudogenised genes) found by the StORF-Reporter methodology. Definitions of the gene families are as follows: Core Genes >= 99%, Soft-Core Genes >= 95% to < 99% and Accessory Genes >= 15% and < 95%. Gene families are only counted once. For example, a gene family which is in the Core Genes group is not also part of the Soft-Core Genes group. The third group 'Con-StORF' reports the number of clusters in the Ensembl-Con-StORF group but with only the StORF sequences being counted. This allows for the reporting of Ensembl-Con-StORF clusters where it is the Con-StORF sequences driving the distribution across the genomes.

Between 3 and 11 Con-StORF sequences were added to each of these Ensembl-Con-StORF gene families with the median number of added sequences being 4. The median length of the representative Con-StORFs of these clusters was 452 nt, with the longest being 1,316 and the shortest 170. Of these 29, 16 were reported as multi Con-StORFs.

Figure 4.5 shows the alignment of the cluster containing one of the most interesting of these multi Con-StORF clusters, whose representative was a very large sequence (3,960 nt including stop codons). The alignment of the clustered Ensembl representative (nitrate reductase subunit alpha) for this multi Con-StORF started at 210 nt, which was past the first internal stop codon. As such, these multi Con-StORF are still technically single internal stop Con-StORFs. Nevertheless, the shared internal stop codon, TAG, was reported as Glutamine (Q) in the Ensembl protein sequence (highlighted in the red box). Additionally, the 5 Con-StORFs all exhibited a number (>10) of single-nucleotide synonymous mutations which were not in the Ensembl sequence (one of them being the in-frame stop codon - TGG/TAG). These results show us that Con-StORFs can present some very interesting, albeit, unexpected results (see the blue highlighted box in the zoomed in Figure 4.6 for finer detail on amino acid differences between the Ensembl and Con-StORF sequences).

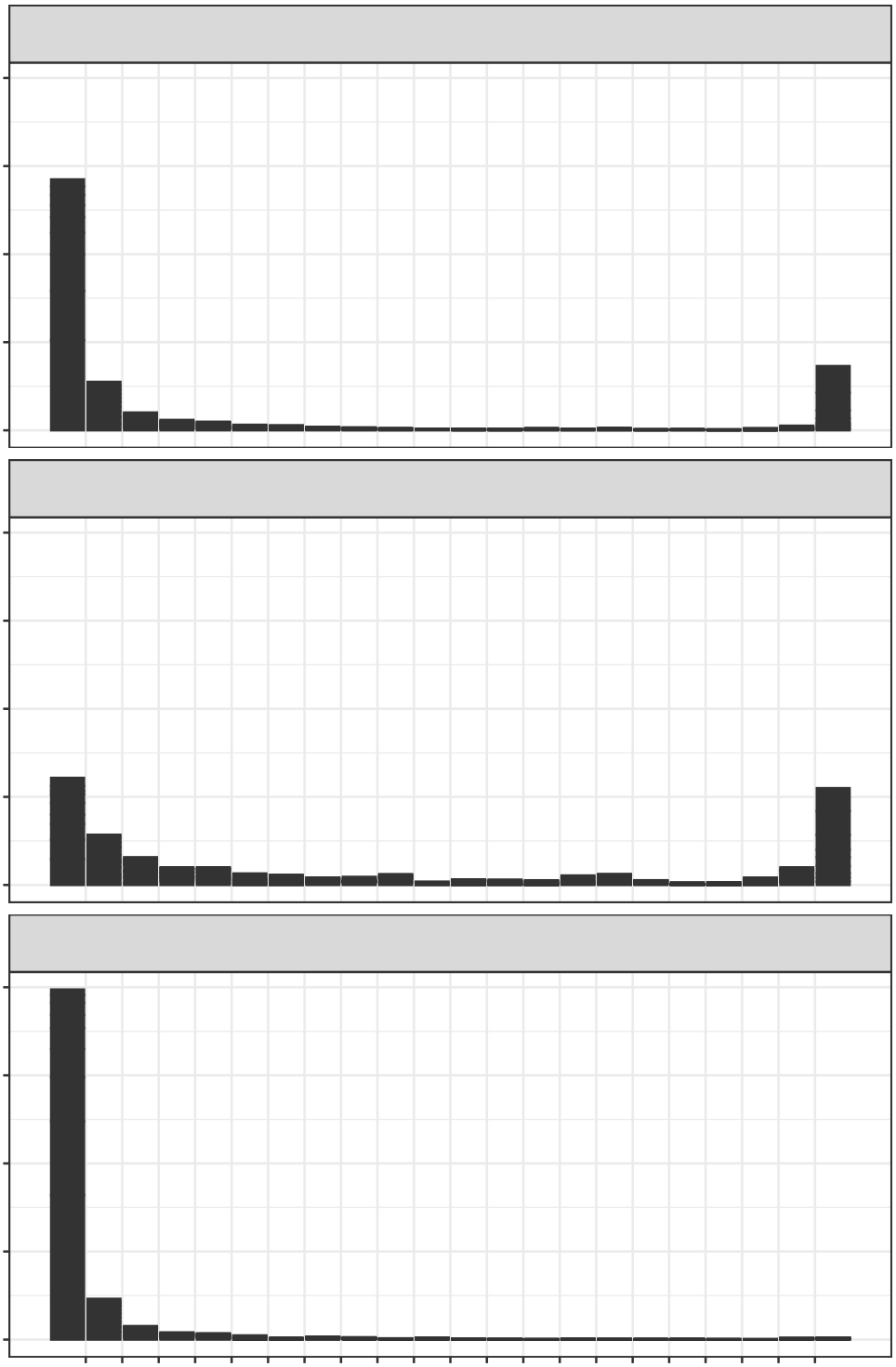


FIGURE 4.4: The distributions of gene families across the 219 *E. coli* genomes forming the pangenome study for the Ensembl-Only, Ensembl-Con-StORF and Con-StORF-Only clusters are plotted here. The reverse bell curve is less pronounced here but is still observable in the Ensembl-Only and Ensembl-Con-StORF data. While the distribution of Ensembl-Con-StORF clusters is weighted more towards the upper end, it is the opposite for Con-StORF-Only.

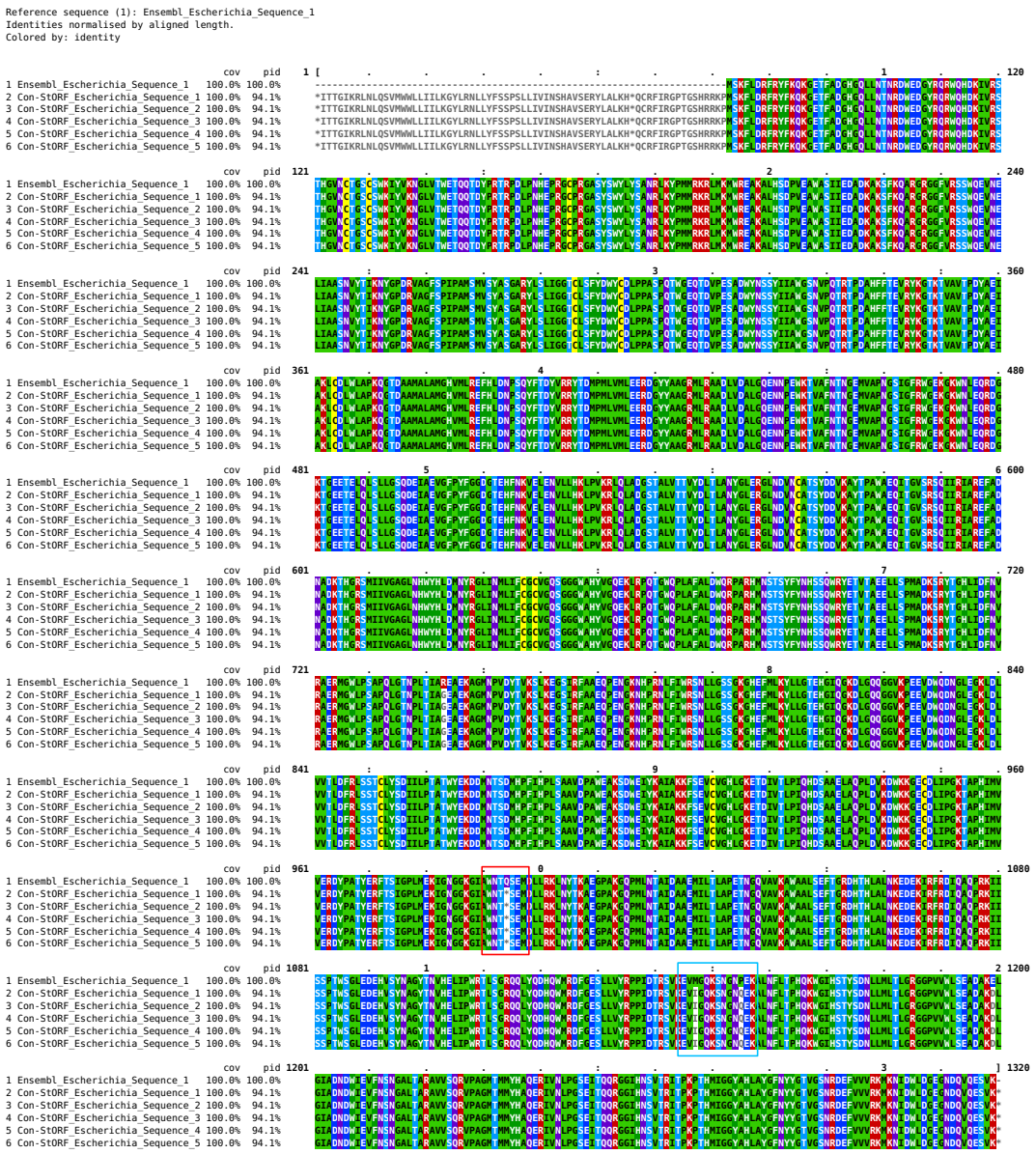


FIGURE 4.5: Clustal Omega multiple sequence alignment of the Ensembl and Con-StORF sequences from Ensembl-Con-StORF cluster 1,043. While aligning across only one of its internal stop codons (TAG) to the Ensembl protein sequence EQZ27181 (nitrate reductase subunit alpha), it was reported as Glutamine (Q) (red box). Additionally, there are also other mutations resulting in the same amino acid present in all Con-StORFs but different in the Ensembl sequence (examples in blue box).

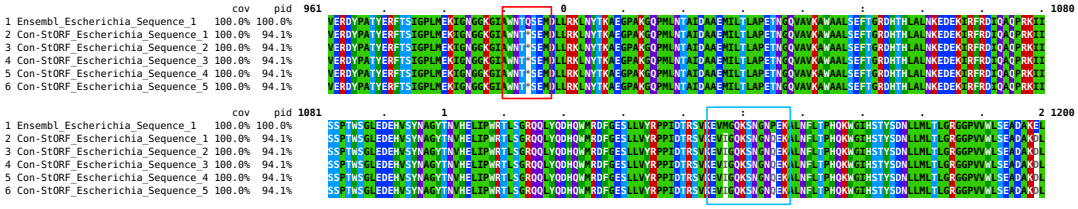


FIGURE 4.6: An expanded view of the Clustal Omega multiple sequence alignment of the Ensembl and Con-StORF sequences from Ensembl-Con-StORF cluster 1,043. Reported in the red box is the conserved internal stop codon (TAG) of the Con-StORFs aligning to the Glutamine (Q) amino acid of the Ensembl protein sequence EQZ27181 (nitrate reductase subunit alpha). Two examples of non-stop codon related synonymous codon changes are reported in the blue box. These mutations which have resulted in the same amino acid present in all Con-StORFs but different in the Ensembl sequence.

The EggNOG COG functional categories reported for Ensembl-Only, Ensembl-Con-StORF and Con-StORF-Only cluster groups in this exploration of the *E. coli* pangenome were investigated and reported in Table 4.7. From these results, a number of clear differences between the three cluster groups can be seen. Firstly, although the number of Ensembl-Con-StORF clusters which reported a COG function is relatively low at 614, compared to the 3,299 for Ensembl-StORF (see Table 3.12 from Chapter 3), there are still observable differences in the distribution of the COG functional groups. The most obvious of these are the large increase of 20% in 'METABOLISM' and a near 10% decrease in the group 'POORLY CHARACTERIZED' from Ensembl-Only to Ensembl-Con-StORF clusters. Next, with even a smaller number of COG assigned representatives, the Con-StORF-Only cluster group was observed to have a very large increase of 'INFORMATION STORAGE & PROCESSING' of nearly 30% compared to Ensembl-Only. This came with the decrease of representatives reported for the other 3 COG categories of between 8-10%. Lastly and quite interestingly, a higher proportion of Ensembl-Con-StORF clusters were reported with a COG category from EggNOG-Mapper compared to those from Ensembl-Only (91.78% and 53.48% respectively).

The juxtaposition observed between the StORF and Con-StORF COG functional categories reported for the representative sequences from the three cluster groups (Ensembl-Only, Ensembl-StORF/Con-StORF and StORF/Con-StORF-Only) showcased a number of similarities and differences (see Tables 3.12 and 3.13 from Chapter 3). While not as severe, the StORF results shown in Table 3.12 were also observed here, with a shift to 'INFORMATION STORAGE & PROCESSING' from the other 3 COG groups. The large differences in the number of Ensembl-Only, StORF-Only and Con-StORF-Only clusters which were reported with a COG functional annotation make comparisons difficult. Nonetheless, there were clear differences observed, specifically in the number of clusters reporting a COG from the 'INFORMATION STORAGE & PROCESSING' group. An from 32.21% to 49.23% was reported

between the StORF-Only and Con-StORF-Only clusters. Additionally, the proportion of Con-StORF-Only clusters which obtained a COG function, were almost 3 times higher than that of StORF-Only (6.63% and 2.60% respectively).

Both the StORF-Only and Con-StORF-Only cluster groups were reported with more COGs from the 'INFORMATION STORAGE & PROCESSING' category than from the Ensembl-Only clusters. As such, this COG category was investigated further and the results are presented in Tables 3.13 and 4.8. Although at low numbers, the same three COG categories, Translation, ribosomal structure and biogenesis (J), Transcription (K) and Replication, recombination and repair (L) were the only three COG functions reported for StORF-Only and Con-StORF-Only. Both cluster groups were also reported with notably higher numbers of function L compared to their respective Ensembl-Only clusters. Lastly, the recently added 'Mobilome' (X) and 'Defense mechanisms' (V) EggNOG COG categories were reported with 12.61%, 41.2% for Ensembl-Only and 3.96%, 1.62% for Con-StORF-Only.

COG Group	Ensembl-Only	Ensembl-Con-StORF	Con-StORF-Only
INFORMATION STORAGE & PRO' [%]	2,932 **[21.51%]**	117 **[17.75%]**	96 **[49.23%]**
CELLULAR PROCESSES & SIG' [%]	3,468 **[25.44%]**	119 **[18.05%]**	31 **[15.90%]**
METABOLISM [%]	3,111**[22.82%]**	286 **[43.40%]**	25 **[12.82%]**
POORLY CHARACTERIZED [%]	4,122 **[30.24%]**	137 **[20.79%]**	43 **[22.05%]**
With COGs/Total Sequences	12,585/19,824 **[63.48%]**	614/669 **[91.78%]**	185/2,794 **[6.62%]**

TABLE 4.7: The COG functional categories assigned to Ensembl-Only, Ensembl-Con-StORF and Con-StORF-Only cluster representative sequences with EggNOG Mapper for the *E. coli* pangenome analysis. Some sequences were observed to have more than one COG functional category. In these instances, the sequence is only counted once in the 'With COGs/Total Sequences' column but each individual COG is counted separately for the 4 groups. While some singleton Ensembl-Only and Con-StORF-Only clusters did have COG annotations, only clusters which had sequences from at least 2 different genomes are reported here. The large differences in the number of Ensembl-Only and Con-StORF-Only clusters which were reported with a COG functional annotation make comparisons difficult. Chi squared statistic tests reported a p-value of <0.00001 for both Ensembl-Only compared to Ensembl-StORF and Con-StORF-Only separately. Further to this, the 'METABOLISM' and 'INFORMATION STORAGE & PROCESSING' COG groups were identified with the highest chi-square statistic in each comparison respectively.

COG Functional Category	Ensembl-Only [%]	Con-StORF-Only [%]
[J] Translation, ribosomal structure and biogenesis	315 **[10.74%]**	1 **[1.04%]**
[A] RNA processing and modification	10 **[0.34%]**	0 **[0%]**
[K] Transcription	1,015 **[34.62%]**	16 **[]16.67%]**
[L] Replication, recombination and repair	1,591 **[54.26%]**	79 **[82.29%]**
[B] Chromatin structure and dynamics	1 **[0.03%]**	0 **[0%]**

TABLE 4.8: The COG functional categories assigned to Ensembl-Only and Con-StORF-Only cluster representative sequences for the group 'INFORMATION STORAGE & PROCESSING'. While the number of Ensembl-Only sequences which obtained a COG classification are much higher than for Con-StORF-Only, there is still a difference in some of reported COG categories. Both 'A' and 'B' are observed in very low numbers (10, 0 and 1, 0 for Ensembl-Only and Con-StORF-Only respectively). Con-StORF-Only sequences have proportionally less 'J' and 'K' but more 'L' than Ensembl-Only sequences, possibly hinting at a functional overview of pseudogenised gene function.

4.4.3 Con-StORFs Identified Within and Across Multiple Genera

In the previous analysis, a small but robust collection of Con-StORFs were found spanning the *E. coli* pangenome. Many of these Con-StORF sequences, found in Ensembl core gene families and forming their own potential novel core gene families, have exhibited a wide range of lengths and functional diversity. As noted in the previous chapters of this thesis, the utility of *E. coli* in a study is often a paradox, due to its over-study and the assumed level of understanding of its genomics within the community (Hunter, 2008b). As such, it was important that both StORFs (see subsection 3.4.5) and Con-StORFs were investigated across a wide selection of prokaryotic genera. To this end, an inter-genera analysis of Con-StORFs was undertaken on the same set of 6,233 prokaryotic genomes from Ensembl Bacteria as done in subsection 3.4.5 of Chapter 3.

While the same URs were used to extract Con-StORFs for both this and the last analysis, as can be seen in the Tables 3.14 and 4.9, there were far fewer Con-StORFs reported than StORFs (1,981 and 148 StORFs and Con-StORFs per genome respectively). However, the Con-StORFs which were reported are more often longer (147 and 330 median length for StORFs and Con-StORFs respectively). Meanwhile, the StORF results still reported the longest sequence of 47,790 nt compared to 33,024 nt for the Con-StORFs.

In further contrast to the StORF results, there are far fewer Ensembl clusters which were extended by Con-StORFs. While there are still a number of Con-StORF and Con-StORF-Only clusters found to be spanning multiple genera (see Table 4.10), only 14 Ensembl-Con-StORF clusters have been extended into additional genera by Con-StORFs. This is in comparison to the 305 Ensembl-StORF clusters as reported in the previous chapter (see Table 3.15). Lastly, out of the resultant 528,369 clusters, 102,683 were non-singletons.

Data	Unannotated Regions	Con-StORFs
Number of Sequences	14,221,482	1,049,855
Median Number Per Genome	2,305	148
Longest Sequence (nt)	86,235	33,024
Median Sequence Length (nt)	240	330
[SD]	**[366.07]**	**[437.58]**

TABLE 4.9: Presented here are the numbers and lengths of unannotated regions (URs) and Con-StORFs extracted from the 6,223 genomes from Ensembl Bacteria. While there was inconsistency in the genome quality across this set of genomes, the numbers reported here are similar to those reported for the Con-StORF analysis of the 6 model organisms and the *E. coli* pangenome. Standard Deviation is reported as **[SD]**.

Cluster Type	1 Genus	2 Genera	3 Genera	4 Genera	5 Genera	6 Genera	>6 Genera
Ensembl-Only	1,569,111	6,9884	6,323	2,491	1,338	803	1,427
Ensembl-Con-StORF	0	0	6	3	1	1	4
Con-StORF-Combined-Ensembl	0	0	0	0	0	0	2
Con-StORF	12,231	136	7	4	3	0	1
Con-StORF-Only	83,963	1,943	169	65	25	12	16

TABLE 4.10: Presented here are the number of clusters which have sequences from multiple genera. The five cluster types are; Ensembl-Only, Ensembl-Con-StORF which are the clusters which have been extended into their respective genera groups by the addition of Con-StORF sequences, Con-StORF-Combined-Ensembl which reports the number of gene families where StORF sequences combined at least 2 or more Ensembl cluster representatives together, Con-StORF which are the same clusters as Ensembl-StORF but are counted only by their number of Con-StORF sequences and lastly StORF-Only which are the clusters which only contain Con-StORF sequences and thus did not cluster with any Ensembl sequence. Con-StORF-Only clusters with a single sequence were not included in these results.

The reduced number of Con-StORF sequences and subsequent relatively low number of Ensembl-Con-StORF and Con-StORF-Only clusters compared to those of the StORF analysis, did not reduce the impact of this work. In fact, a number of interesting clusters, both Ensembl-Con-StORF and Con-StORF-Only, have been reported in this analysis. Firstly, the Con-StORF-Only cluster 1,326, which consisted of 61 sequences at 100% sequence similarity along their entire 71 amino acid length, spanned the highest number of genera for a Con-StORF-Only cluster. The twelve genera in this cluster were - *Salmonella*, *Aeromonas*, *Klebsiella*, *Enterobacter*, *Providencia*, *Pantoea*, *Acinetobacter*, *Pseudomonas*, *Citrobacter*, *Shewanella*, *Corynebacterium* and *Escherichia*. This cluster obtained a homology result to the UniProt 'Unreviewed (TrEMBL (Bateman et al., 2020)) - Computationally analyzed' database and not to Swiss-Prot. However, it was reported with 100% sequence similarity and 95.1% coverage, to a nucleoside triphosphate (NTP) binding protein (D3W47_09045) (see Figure 4.7 for multiple sequence alignment). Interestingly, the alignment of this protein was only to the first segment, or effectively the first StORF making up the Con-StORF. Even though this alignment was technically to a StORF, it was only reported as such 24 times in 4 genera, (20 *Klebsiella*, 3 *Pseudomonas*, and 1 for *Enterobacter* and *Escherichia* respectively) in the StORF results in Chapter 3. There are a number of reasons why this could happen. The most likely of these is that as the first Con-StORF segment is quite short at 41 amino acids, these small StORFs are being filtered out by larger StORFs. To counteract this, StORFs and Con-StORFs should be investigated together. Additionally and quite surprisingly, not only were all sequences 71 amino acids long, but also their DNA sequences were exactly the same and therefore they all had the same in-frame stop codon, TAA, in the same position, as seen in Figure 4.8.

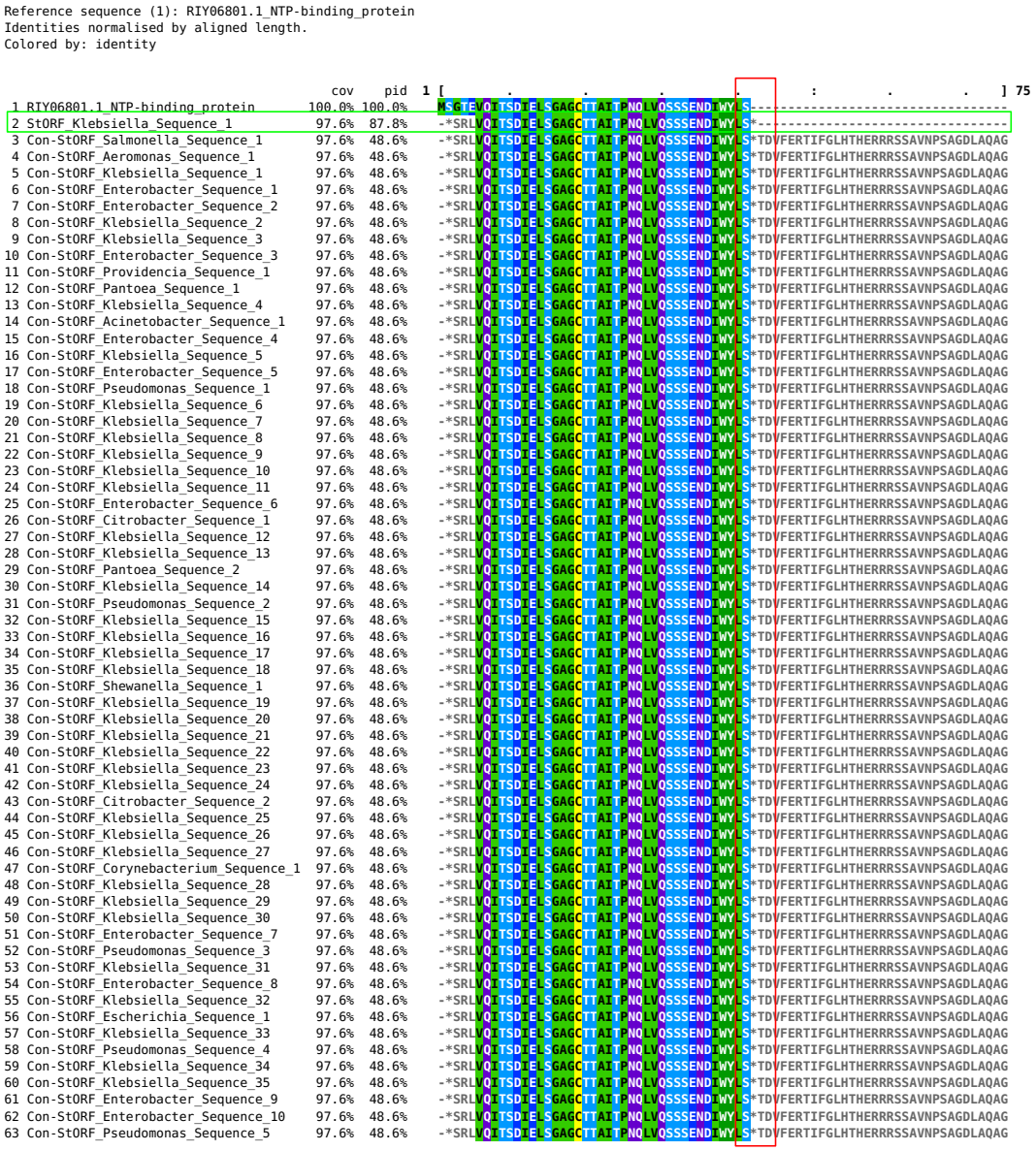


FIGURE 4.7: Clustal Omega multiple sequence alignment from the amino acid sequences of cluster 1,326 and a representative sequence from the StORF analysis. The StORF sequence (highlighted in green) is added to this alignment to report how the aligned Con-StORF aligned solely along the first StORF segment. This Con-StORF-Only cluster spanned the highest number of genera and consists of 61 sequences of 71 amino acids in length, with 100% sequence similarity to each other and spanned 12 genera - *Salmonella*, *Aeromonas*, *Klebsiella*, *Enterobacter*, *Providencia*, *Pantoea*, *Acinetobacter*, *Pseudomonas*, *Citrobacter*, *Shewanella*, *Corynebacterium* and *Escherichia*. Highlighted in red is the conserved stop position of both the NTP-binding protein and StORF sequence which is reported as an 'internal' stop position across all 61 Con-StORF sequences.

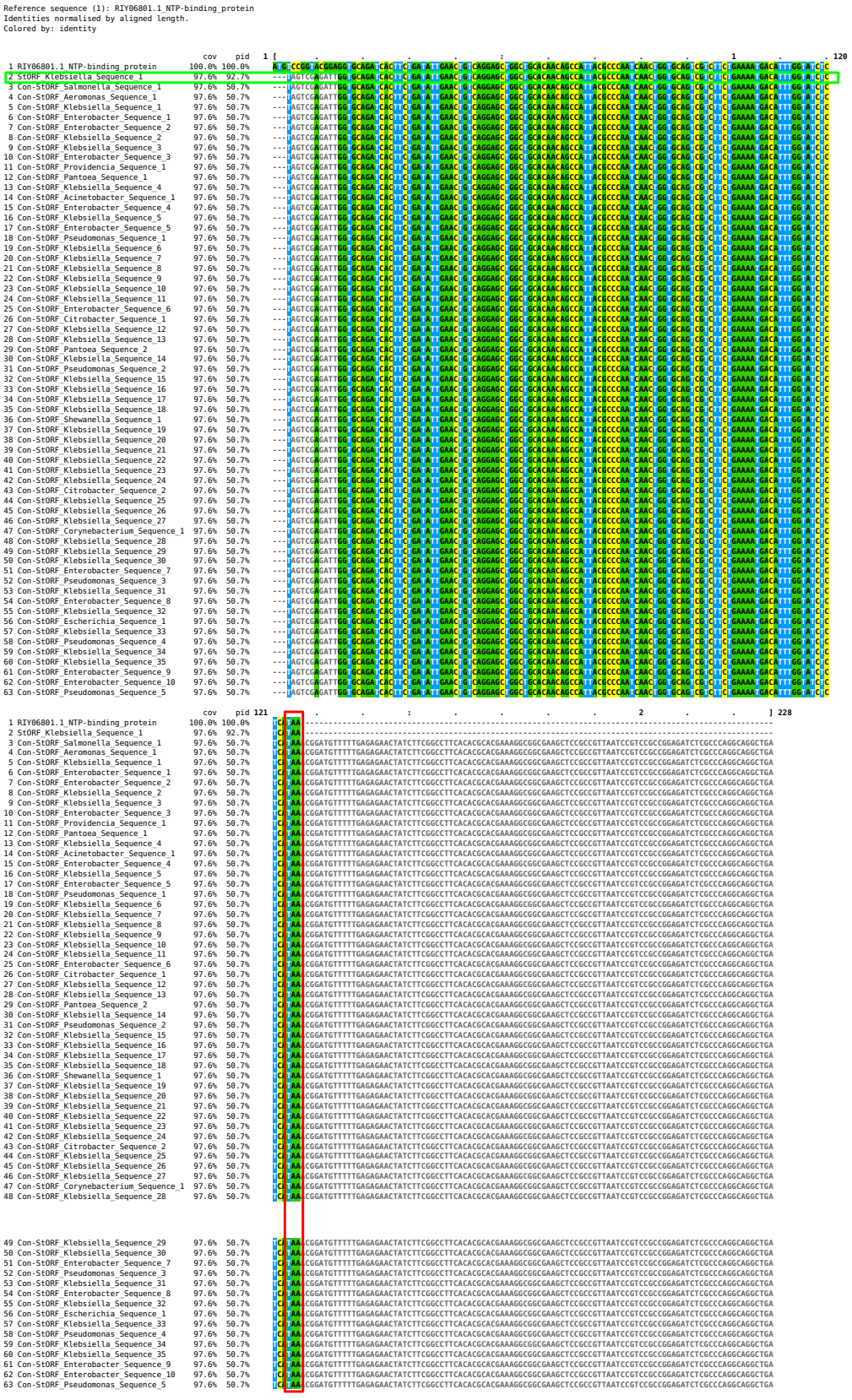


FIGURE 4.8: Clustal Omega multiple sequence alignment of the DNA sequences from the Con-StORF-Only cluster 1,326 which consists of 61 sequences of 71 nucleotide bases. This Con-StORF-Only cluster was found in the highest number of different genera (12), all with exactly the same DNA sequence, and therefore, in-frame stop codon and position. Highlighted in green is the StORF sequence and in red is the conserved position of the internal 'TAA' stop codon.

There were two Ensembl-Con-StORF clusters which were each formed with two different Ensembl representative sequences, respectively. These 'Combined' clusters, as described previously, are examples of the limitations of this analysis imposed by computational techniques, and as such, are reported in Table 4.10, 'Ensembl-Combined-Con-StORF'. One of these clusters, cluster 11,195, was of particular interest and was formed from the two representative sequences from Ensembl clusters 103177 and 674 which contained 23 and 681 sequences from 4 and 6 genera respectively. As seen in the ClustalO multiple sequence alignment in Figure 4.9, while the lengths of the two Ensembl representative sequences are similar at 1,248 and 1,276 amino acids, except for the *Shigella* sequence (at 1,310 amino acids) Con-StORFs are 1,316 amino acids long. Interestingly, while the 5 longer *Escherichia* Con-StORFs are all multi Con-StORFs with 2 internal dissecting stop codons, the shorter *Shigella* Con-StORF is the only non-multi Con-StORF. For all 6 Con-StORFs, the first dissecting stop codon is upstream of the region of alignment for both Ensembl proteins sequences, thus the *Shigella* Con-StORF does not contain an in-frame internal stop codon. There are additional differences between the *Escherichia* and *Shigella* Con-StORFs. One such example is clearly visible in the 3rd from the bottom alignment grid where a 'Q' is present for the *Shigella* and '-', indicating no amino acid (stop codon), is present for the other 5 *Escherichia* Con-StORFs (highlighted in the orange box - see Figure 4.9 and Figure 4.10 for finer detail). This may indicate that the *Shigella* Con-StORF may still be at an earlier stage or has undergone a different route of pseudogenisation such as ribosomal binding site loss, compared to the other 5 *Escherichia* Con-StORFs. Further to this, the red box highlights another position where it is the *Shigella* Con-StORF which contains an amino acid difference. Additionally, as shown in finer detail in the zoomed in Figure 4.10, the purple box highlights positions where the Ensembl *Cronobacter* sequence differs from all other sequences and the green box highlights where it is the Ensembl *Enterobacter* sequence which differs from all other sequences.

While Con-StORFs have been designed to identify pseudogenised genes, these results may present evidence of not only gene pseudogenisation, but also the even more enigmatic process of alternative codon usage. In addition to this, the observed amino acid differences across the two genera for this gene family is supported by the process of genetic code differences and codon optimisation between genera and even species (Gouy and Gautier, 1982; Ikemura, 1985; Bulmer, 1987). The juxtaposition of the differential presence and absence of amino acids in these two positions only further compounds the complexity of our limited and incomplete understanding of protein sequences.

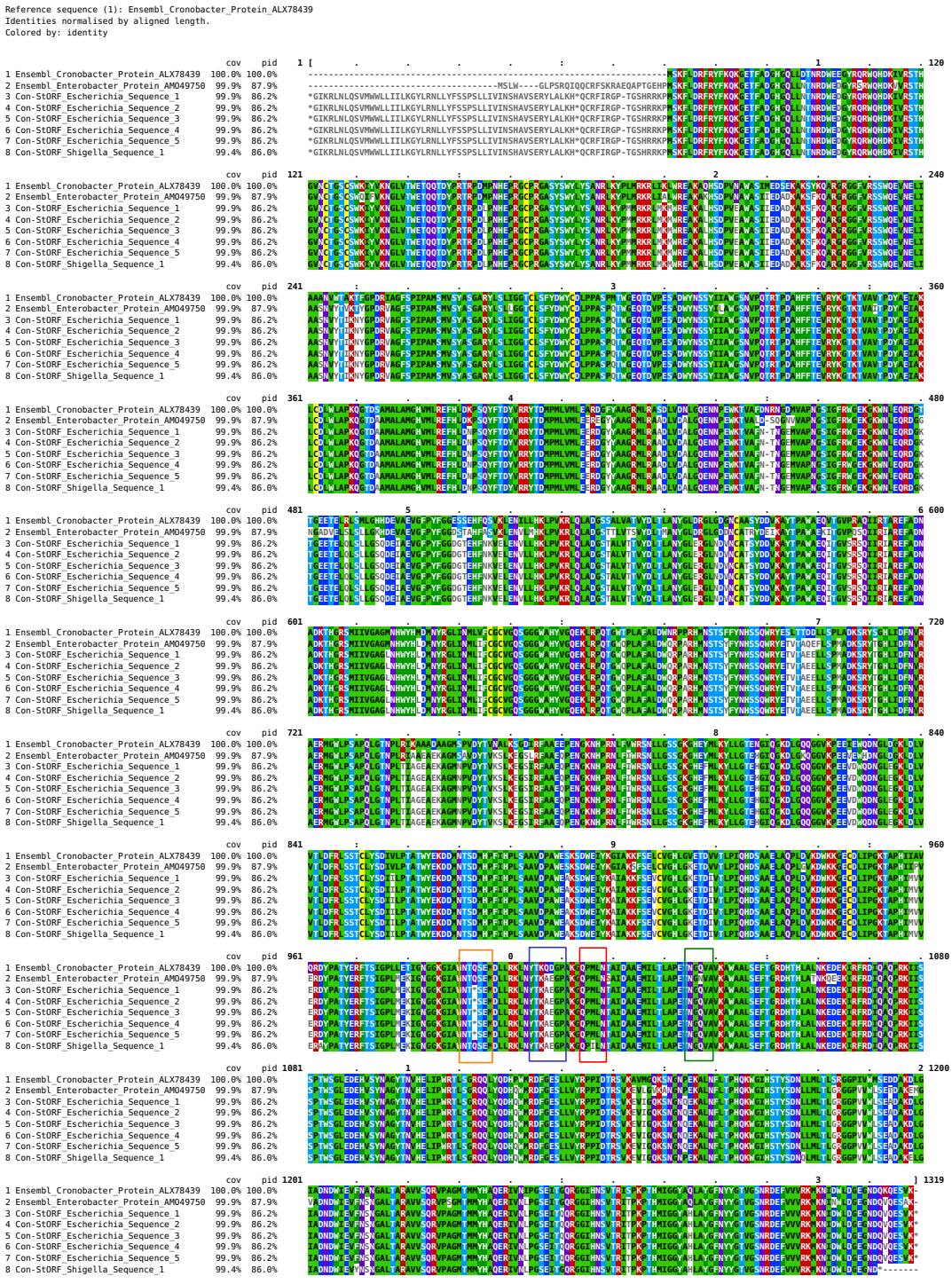


FIGURE 4.9: Clustal Omega multiple sequence alignment from the amino acid sequences of Ensembl-Combined-Con-StORF cluster 11,195. The lengths of the two Ensembl representative sequences are 1,248 and 1,276 amino acids respectively and except for the *Shigella* sequence which is at 1,310 amino acids, the five remaining *E. coli* Con-StORFs are 1,316 amino acids long. Additionally, while the 5 longer *Escherichia* Con-StORFs are all multi Con-StORFs with 2 internal dissecting stop codons (with the in-frame internal stop position shown in the orange box), the shorter *Shigella* Con-StORF is the only non-multi Con-StORF. The red box highlights another position where it is the *Shigella* Con-StORF which contains an amino acid difference. The purple box highlights positions where the Ensembl *Cronobacter* sequence differs from all other sequences and the green box highlights where it is the Ensembl *Enterobacter* sequence which differs from all other sequences.

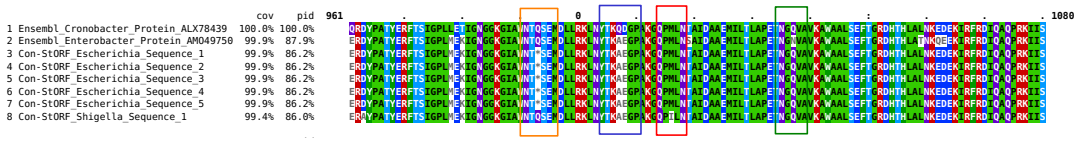


FIGURE 4.10: An expanded view of the Clustal Omega multiple sequence alignment from the amino acid sequences of Ensembl-Combined-Con-StORF cluster 11,195. While the 5 longer *Escherichia* Con-StORFs are all multi Con-StORFs with 2 internal dissecting stop codons (with the in-frame internal stop position shown in the orange box), the shorter *Shigella* Con-StORF is the only non-multi Con-StORF. The red box highlights another position where it is the *Shigella* Con-StORF which contains an amino acid difference. The purple box highlights positions where the Ensembl *Cronobacter* sequence differs from all other sequences and the green box highlights where it is the Ensembl *Enterobacter* sequence which differs from all other sequences.

The Con-StORF-Only clusters which were found in at least two or more genomes were checked for COG functional categories using the EggNOG-Mapper tool. The representatives and the COG categories are presented below in Table 4.11. Unfortunately, while only a few Con-StORF-Only cluster representative sequences reported a COG functional category (77 out of 1,467), as reported in all previous analysis of StORF and Con-StORF COG results, 'INFORMATION STORAGE & PROCESSING' was again the most often reported category. Lastly, the recently added 'Mobilome' (X) and 'Defense mechanisms' (V) EggNOG COG categories were reported with 2.17%, 24.60% and 3.67%, 6.07% for Ensembl-Only and Con-StORF-Only respectively.

COG Group	Ensembl-Only	Ensembl-Con-StORF	Con-StORF-Only
INFORMATION STORAGE & PRO' [%]	280,350 **[18.54%]**	2,432 **[24.84%]**	1,514 **[36.06%]**
CELLULAR PROCESSES & SIG [%]	315,705 **[20.88%]**	2,159 **[22.05%]**	745 **[17.74%]**
METABOLISM [%]	592,687 **[39.19%]**	3,332 **[34.03%]**	1,031 **[24.55%]**
POORLY CHARACTERIZED [%]	323,483 **[21.39%]**	1,869 **[19.08%]**	909 **[21.65%]**
With COGs/Total Sequences	1,394,325/1,642,303 **[84.90%]**	8,937/12,382 **[72.18%]**	3,922/86,193 **[4.55%]**

TABLE 4.11: The COG functional categories assigned to Ensembl-Only, Ensembl-Con-StORF and Con-StORF-Only cluster representative sequences with EggNOG-Mapper for the inter-genera analysis. Some sequences were observed to have more than one COG functional category. In these instances, the sequence is only counted once in the 'With COGs/Total Sequences' column but each individual COG is counted separately for the 4 groups. Clusters are reported here irrespective of whether they were extended into new genera by Con-StORF sequences. Chi squared statistic tests reported a p-value of <0.00001 for both Ensembl-Only compared to Ensembl-StORF and Con-StORF-Only separately. Further to this, the 'INFORMATION STORAGE & PROCESSING' COG group which was identified with the highest chi-square statistic in both comparisons.

COG Functional Category	Ensembl-Only **[%]**	Con-StORF-Only **[%]**
[J] Translation, ribosomal structure and biogenesis	71,532 **[25.52%]**	88 **[5.81%]**
[A] RNA processing and modification	347 **[0.12%]**	0 **[0%]**
[K] Transcription	128,612 **[45.88%]**	293 **[]19.35%]**
[L] Replication, recombination and repair	79,365 **[28.31%]**	1,133**[74.83%]**
[B] Chromatin structure and dynamics	494 **[0.18%]**	0 **[0%]**

TABLE 4.12: Presented here are the COG functional categories assigned to Ensembl-Only and Con-StORF-Only cluster representative sequences for the group 'Information Storage and Processing'. The number of Ensembl-Only sequences which obtained a COG classification are much higher than for Con-StORF-Only. The reported COG categories are reported proportionally. Both 'A' and 'B' are observed in very small proportions (0.12%,0% and 0.18%,0%, for Ensembl-Only and Con-StORF-Only respectively). Additionally, Con-StORF-Only sequences have less than half 'K' but more than twice 'L', as many as the Ensembl-Only sequences. These results are similar to the earlier studies of COG functions.

4.4.4 Validation of Con-StORFs

The results of the previous 3 sections showcased a number of different Con-StoRFs and how they can lead to the reshaping of a genome's gene-collection. As with StORFs, Con-StORFs have been designed to detect specific types of genes missed from canonical annotation. While pseudogenes are the target of Con-StORFs, not all Con-StORFs are pseudogenes, therefore, it important to validate these further. Therefore, to further investigate the Con-StORFs reported and to identify potential differences between validated and non-validated Con-StORFs, validation was undertaken for the Con-StORFs reported in the three previous result subsections.

While true validation of a Con-StORF is impossible without experimental evidence, through this naive approach, we can at least ascertain whether a Con-StORF has a high quality sequence alignment to a protein in protein databases such as Swiss-Prot (Bateman et al., 2020) and that this alignment spans across the Con-StORF's internal in-frame stop codons. Henceforth such Con-StORFs will be called "validated" (see Figure 4.11).

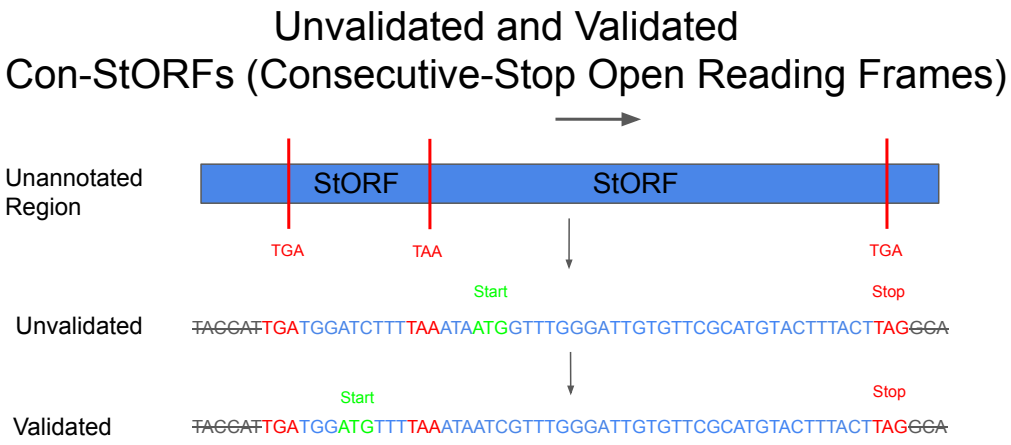


FIGURE 4.11: Visual representation of an unvalidated and validated Con-StORF. While both report a gene, in the unvalidated Con-StORF, the gene sequence does not extend past the internal dissecting stop codon. This is important as the identified gene is not an in-frame stop pseudogenised gene or an alternative codon using gene. The validated Con-StORF has captured a CDS gene sequence on either end of its internal dissecting stop codon ('TAA').

Sequence alignments identified with either BLAST (Altschul et al., 1990) or DIAMOND (Buchfink, Reuter, and Drost, 2021) are reported with both query and subject alignment data. As such, **check_IF_Con-StORF.py** has been developed to take the default tabulated output of either tool (DIAMOND was used in this analysis) and compare the alignment between the query Con-StORF and subject Swiss-Prot protein sequences. The number of internal stop codons spanned by a subject Swiss-Prot (or any aligned protein) protein are reported. As seen in Figure 4.1 section D, a Con-StORF can consist of multiple internal stop codons which may or may not

intersect the subject sequence. 'qstart' and 'qend', the 7th and 8th column of the alignment results from BLAST and DIAMOND, report which parts of the query sequence (Con-StORF) aligned to the subject sequence (Swiss-Prot protein) and as such are used to determine the validity of the Con-StORF.

It is true that some 'real' pseudogenised genes will likely be missed by this computational-only approach to validation. However, those Con-StORFs which are not validated but do align to known proteins, are still at a minimum, StORFs which have captured potential CDSs in the URs of canonical genome annotations. Therefore, not only are we able to validate Con-StORFs, but also inspect the differences between them and their homologs identified in previous studies.

4.4.4.1 Validating Con-StORFs: Six Model Organisms

Genome	Num of Con-StORFs	Swiss-Prot		Intra-Genome	
		Hits **[#]**	Coverage >=80% **[#]**	Hits **[#]**	Coverage >=80% **[#]**
B. subtilis	97	6 **[1]**	4 **[0]**	3 **[1]**	1 **[0]**
C. crescentus	103	2 **[0]**	1 **[0]**	1 **[0]**	1 **[0]**
E. coli	199	23 **[8]**	20 **[6]**	13 **[8]**	4 **[2]**
M. genitalium	32	25 **[23]**	6 **[6]**	21 **[19]**	0 **[0]**
P. fluorescens	356	14 **[6]**	11 **[4]**	17 **[3]**	11 **[3]**
S. aureus	103	1 **[0]**	0 **[0]**	2 **[2]**	1 **[1]**

TABLE 4.13: Table containing the number of Con-StORFs found in the URs recovered from Ensembl annotations for six model organisms. The numbers of Con-StORFs which had a high sequence similarity and >=80% subject hit to a protein in Swiss-Prot and Ensembl proteome is listed. **[#]** is used to indicate the number of Con-StORFs which were identified to have at least one central stop codon placed within the subject protein sequence (validated). The results of this table were not calculable for a chi squared statistical test.

The 6 model organisms have exhibited a wide range of URs, Con-StORFs and internal dissecting stop codon usages. As seen in Table 4.13, apart from *M. genitalium*, a small proportion of each set of Con-StORFs were validated, as described in subsection 4.4.4. As already discussed, the specific case of *M. genitalium* may be explained by the known recoding of TGA for tryptophan and as such, many of the Con-StORFs reported may be as a result of canonical annotation error and not only gene pseudogenisation. Subsequently, out of the 103 Con-StORFs reported in *S. aureus* (the genome with the lowest reported number of Con-StORF alignments), of the 3 which did exhibit homology to either the Swiss-Prot and intra-genome proteins, the Swiss-Prot alignment was a non-validated Con-StORF and the 2 intra-genome alignments were validated Con-StORFs. *P. fluorescens* and *E. coli*, having the highest and second highest number of Con-StORFs, had only 8 and 6 validated Con-StORF alignments to Swiss-Prot and 8 and 3 for intra-genome respectively. Lastly, even

though the proportion of validated Con-StORFs is relatively low, they are nonetheless worth investigating. See Figure 4.3 for an example of a non-validated, non-aligned Con-StORF protein sequence.

The types of internal in-frame stops could .. help us understand whether these genes are .. not pseudogenses and instead are alternative coding genes. It could also be possible that the processes behind pseudogenisation are selecting for certain stop codons over others.

It is difficult to draw any conclusions of the usage of internal in-frame stop codons due to the low number of validated Con-StORFs reported in Table 4.14. These results, however, continue to present the codon usage differences observed previously across 'All Con-StORFs'. Interestingly, while there were clear differences in the use of the three canonical stop codons as internal in-frame stop codons within each of the six model organisms, it was not consistent between them. While the 32 Con-StORFs reported in *M. genitalium* included 4 TAG and 2 TAA internal in-frame stop codons (irrespective of whether they were validated Con-StORFs or not), only one TAG was part of a validated Con-StORF (see Table 4.14). Additionally, apart from *M. genitalium* and its use of TGA, the distribution of internal in-frame stop codons was not replicated in the 'Validated Con-StORFs' for the other genomes. For example, in *E. coli*, while TAG was the least used codon in 'All Con-StORFs', it was the most used in the 'Validated Con-StORFs' Unfortunately, the limited number of validated Con-StORFs in this analysis impedes our ability to discern codon preference. The only other apparent preference shown was for TAG and TAA in the results for *E. coli*, but due to the low number of 'Validated Con-StORFs', this is not evincible.

Genome	Con-StORFs **[Multi]**	All Con-StORF Internal Stop Codons			Validated Con-StORFs Internal Stop Codons		
		TGA	TAG	TAA	TGA	TAG	TAA
B. subtilis	97 **[6]**	51	8	44	1	0	0
C. crescentus	103 **[6]**	61	25	23	0	0	0
E. coli	199 **[11]**	76	43	94	1	6	5
M. genitalium	32 **[12]**	38	4	2	33	1	0
P. fluorescens	356 **[39]**	184	108	117	2	3	3
S. aureus	103 **[4]**	24	22	62	1	1	0

TABLE 4.14: The stop codon usage for the internal stops of the Con-StORFs for each of the 6 model organisms. In cases where there are more than one internal stop codon (**[Multi]**), the codons are counted individually. If Con-StORF were found to have both Swiss-Prot and Intra-Genome hits, they are only recorded once in the validated internal numbers. A chi squared statistic test reported a p-value of <0.177768 for the 'All Con-StORF' compared to 'Validated Con-StORF' internal stop codon counts, so the differences are not significant.

StORF Collection	Con-StORFs **[Multi]**	Validated **[Multi]**	All Con-StORF Internal Stop Codons			Validated Con-StORFs Internal Stop Codons		
			TGA	TAG	TAA	TGA	TAG	TAA
Core Gene Ensembl-Con-StORFs	29 **[8]**	26 **[1]**	13	13	11	8	12	7
All Con-StORFs – Swiss-Prot	40,946 **[1,941]**	1,077 **[17]**	19,283	15,561	8,259	245	461	371

TABLE 4.15: The internal stop codon usage for the Con-StORFs for each of the *E. coli* pangenome clusters. In cases where there are more than one internal stop codon (**[Multi]**), the codons are counted individually. 'Core Gene Ensembl-Con-StORFs' are validated against their respective Ensembl-Only cluster representatives and 'All Con-StORFs – Swiss-Prot' are separately validated against the Swiss-Prot protein database. Chi squared statistic tests reported a p-value of 0.177768 for the 'All Con-StORF' compared to 'Validated Con-StORF' internal stop codon counts for the 'Core Gene Ensembl-Con-StORFs', and a p-value of <0.00001 for the 'All Con-StORFs – Swiss-Prot' comparison.

4.4.4.2 Validating Con-StORFs: *Escherichia coli* Pangenome

The previous results have shown that the process of validating Con-StORFs from a single genome is relatively straightforward as each Con-StORF has a single protein to which it has been aligned. However, it is a very different problem to validate Con-StORF sequences which have clustered with a number of different Ensembl annotated proteins from different strains of the same species, as in the case of the *E. coli* pangenomic analysis. As such, each set of Con-StORF sequences to be validated requires a level of manual grouping which the analysis for the 6 model organisms did not. Two such groupings of validated Con-StORFs are presented in Table 4.15. The first reports that of the 29 Ensembl-Con-StORF clusters which were extended into core gene families by the addition of the Con-StORF sequences, 26 were validated as true Con-StORFs. Additionally, there were 16 multi Con-StORFs from the set of 26 validated Con-StORFs. The second reports a contrast to the high proportion of validated core Ensembl-Con-StORFs reported in the previous analysis. Of the complete collections of 40,946 Con-StORFs reported for all 219 *E. coli* genomes, only 1,077 were validated against the Swiss-Prot database. While this validation was performed outside the CD-Hit clusters, the Swiss-Prot protein sequence alignments can still give us an overview of the likely proportion of true psuedogenised genes, discovered by Con-StORFs across the *E. coli* pangenome. Through these results, an enigmatic but interesting picture of Con-StORFs and the history of pseudogenes emerges.

4.4.4.3 Validating Con-StORFs: Inter-Genera Analysis

Unlike the cluster sequences from the *E. coli* pangenome analysis, the 15 'Genera Extended Ensembl-Con-StORFs' and 1,049,855 'All Con-StORFs – Swiss-Prot' Con-StORFs, as shown in Table 4.16 are, for the most part, not validated against sequences

Data	Con-StORFs **[Multi]**	Validated **[Multi]**	All Con-StORFs Internal Stop Codons			Validated Con-StORFs Internal Stop Codons		
			TGA	TAG	TAA	TGA	TAG	TAA
Genera Extended Ensembl-Con-StORFs	15 **[0]**	5 **[0]**	8	4	3	1	4	0
All Con-StORFs Swiss-Prot	1,049,855 **[74,887]**	13,617 **[629]**	500,381	257,998	376,739	4,071	4,885	4,661

TABLE 4.16: The internal stop codon usage for the Con-StORFs for each of the 6,223 Ensembl Bacteria clusters. In cases where there are more than one internal stop codon (**[Multi]**), the codons are counted individually. 'Core Gene Ensembl-Con-StORFs' are validated against their respective Ensembl-Only cluster representatives and 'All Con-StORFs – Swiss-Prot' are separately validated against the Swiss-Prot protein database. Chi squared statistic tests reported a p-value of 0.010435 for the 'All Con-StORF' compared to 'Validated Con-StORF' internal stop codon counts for the 'Genera Extended Ensembl-Con-StORFs' and a p-value of $<$0.00001 for the 'All Con-StORFs – Swiss-Prot' comparison.

from the same species. As such, the results of validation are more difficult to interpret. While the 'Genera Extended Ensembl-Con-StORFs' were validated by taking the representative Con-StORF (1 of possible $\geq$1) and comparing that against the Ensembl protein it had clustered with (1 of possible $\geq$1), the 'All Con-StORFs – Swiss-Prot' consisted of the entire collection of inter-genera Con-StORFs. These were aligned to the Swiss-Prot database and the 'best' result was used for the validation. However, irrespective of whether the validation of these Con-StORFs is consistent, the results nonetheless suggest that these sequences are found as coding genes in a curated protein database. There are some clear differences between the three stop codons which do suggest some level of preference, even across multiple genera. For example, while TAG is used in the fewest internal stop codons for 'All Con-StORFs (257,998), it is the most used validated internal stop codon (4,885). The exact opposite can be seen for TGA. Used in nearly double the number of 'All Con-StORF' stop codons compared to TAG (500,381), it was the least used in the group of validated Con-StORFs (4,071).

While utilising Swiss-Prot as a curated protein database does enable a higher level or confidence in the Con-StORFs which align to it, the fact that the database does not contain the DNA sequence for their proteins makes further investigation difficult. Even with the protein names, due to the reuse of codons, it is unlikely that it would be possible to find the exact DNA sequence which coded for a specific Swiss-Prot protein. These results once again reiterate the importance of experimental work and state the importance of recording both the DNA and amino acid sequences of each individual CDS gene recorded in genomic databases. As such, with the current data available, it is impossible to determine which of the Con-StORF sequences are using alternative codon usage (stop codon coding for amino acid) or are pseudogenised versions of the sequences in Swiss-Prot.

4.5 Discussion

The results of the previous chapter showcased the level of genomic knowledge still held in the URs of prokaryotic genomes (see Chapter 3, Subsections 3.5.1 and 3.5.2). Through that work, a major impediment for the use of the StORF-Reporter methodology and its acceptance as an (additional) annotation platform was identified as the high number of StORFs reported for each genome (see Tables 3.14). Additionally, there were various types of CDS genes which were being identified by StORFs: Complete CDS identified from complete genomes (see Table 3.5), (potentially pseudogenised) CDSs found across different genera with amino acid changes in different positions (see Figure 3.13), and pseudogenised CDSs which were likely a result of gene duplication or a gene acquisition and subsequent deleterious IFS mutation (see Table 3.8). The focus of this chapter has been to further understand the putative pseudogenes identified. This greatly reduced the number of sequences to be analysed and therefore scale of the analysis allowing a deeper investigation into a too often understudied genomic element. Regardless of the small numbers of sequences reported (see Table 4.9), we found high numbers of Con-StORFs which formed large numbers of clusters with both Ensembl and other Con-StORF sequences, indicating that they were recently functional genes.

4.5.1 The 'Consecutive Stop - Open Reading Frame': Pseudogene Detection May Reveal Recent Functional History

Pseudogenes have long been seen as a paradigm of neutral mutation, with the belief that their removal from a genome after pseudogenisation is neither beneficial nor detrimental (Li, Gojobori, and Nei, 1981). However, much more recently, it has been observed in bacteria that some pseudogenes are rapidly deleted from their genomes, "suggesting that their presence is somehow deleterious" (Kuo and Ochman, 2010). This corresponds to the processes of gene duplication (Magadum et al., 2013) and the delicate balance of gene expression or 'dosage dependency' (Klumpp, Zhang, and Hwa, 2009; Birchler and Veitia, 2012), which both have competing influences on the creation and destruction of novel function. With this in mind, it is surprising how little work has been done to investigate putative pseudogenised genes which have been retained longer and at a higher level of conservation than expected. Pseudogenes and the processes behind their creation are as diverse as prokaryotic genes themselves (see Figure 3.14). It has been speculated that these conserved pseudogenes may be involved in regulation of their "parent gene" and it has been observed that some pseudogenes are even transcribed into RNA (Tutar, 2012). Interestingly, some pseudogenes have been reported to possess the same mutations across different genera (Tutar, 2012). Possible examples of this can be seen in Figures 4.9 and 4.8.

Further to this, pathogenic prokaryotes have been observed with higher levels of pseudogenised genes (Liu et al., 2004), most often with functions involved in metabolism which is likely due to higher levels of redundancy required for their more specialised environments (Carlile, 1982). In non-pathogens, those pseudogenes which are reportedly involved in metabolism, could be explained by the existence of a high level of "metabolic redundancy [in prokaryotes] caused by known alternative metabolic routes" (Xavier, Patil, and Rocha, 2018). However, our knowledge of the functions of pseudogenes is influenced by the types of genes which are of interest clinically and industrially (Valdés et al., 2008; Lobb et al., 2020). It is likely, however, that pseudogenes represent a much wider variety of functions. Routine pseudogene detection is most often carried out with homology searches (Lerat and Ochman, 2005; Zhang et al., 2006b; Tanizawa, Fujisawa, and Nakamura, 2018) which encounter the same types of biases and omissions as reported in the Background and Chapter 2. Supporting this, detailed investigations of parental gene function of pseudogenes have portrayed a wide functional profile (Liu et al., 2004) including families such as DNA transposition, metabolism, nucleotide processing and repair or replication (Lerat and Ochman, 2005). However, over half of the pseudogenes were reportedly produced from genes whose original functions were annotated as 'hypothetical' or 'unknown' (Lerat and Ochman, 2005), demonstrating that there is still a lot of work to be done.

The literature diverges on many of the core characteristics of pseudogenes. These include both the processes behind their formation and which genes they most often arise from, essential, unessential or transiently essential. Due to failed horizontal gene transfer (HGT) being reported as the most common initiator of gene pseudogenisation (Liu et al., 2004), the 'Complexity Hypothesis' (Jain, Rivera, and Lake, 1999) which describes many of the factors behind HGT, has often been indirectly used to explain the processes behind pseudogenisation. However, while the observation that "extensive horizontal transfer has occurred for operational genes (those involved in housekeeping), whereas informational genes (those involved in transcription, translation, and related processes) are seldom horizontally transferred." (Jain, Rivera, and Lake, 1999), it could be suggested that 'operational' functions should be the most common in pseudogenised genes. This is supported by what has been reported in functional studies (Liu et al., 2004; Lerat and Ochman, 2005). On the other hand, a delicate balance exists between the essentialness of a gene and the complexity of its interactions (Ning et al., 2010) and in a more recent review of the Complexity Hypothesis, it was found that "... the biological function of a gene family is an insignificant factor in the determination of transferability.... In contrast, we found that connectivity is an important and statistically significant factor in determining transferability" (Gilbert et al., 2018; Cohen, Gophna, and Pupko, 2011). As even less is known of gene interactions, compared to their function (Kahan et al., 2021), it is unlikely that the complete picture of gene pseudogenisation will be resolved in the near future.

Our results found Con-StORF-Only clusters to have much more of the EggNOG COG function Mobilome (recently added to the EggNOG database), which includes transposable elements (TEs), than Ensembl-Only clusters. This large proportional shift was from 12.61% to 41.2% and 2.17% to 24.60% for the *E. coli* pangenome and inter-genera studies respectively (see Figure 4.12). Additionally, genes within the Mobilome COG category were previously reported as part of the 'Replication, recombination and repair' (L) COG function which is part of the 'INFORMATION STORAGE & PROCESSING' COG group which was found to be the most common group for Con-StORF-Only clusters (see 4.4.4 for details on how EggNOG COGs were reported). This large increase in 'INFORMATION STORAGE & PROCESSING' and thus Mobilome related sequences in Con-StORF-Only clusters contradicts the predictions made by the Complexity Hypothesis (Jain, Rivera, and Lake, 1999). While we are unable to investigate the degree of interactions of these genes, the functional findings are generally in agreement with the more recent analysis of the "revisited complexity hypothesis" (Cohen, Gophna, and Pupko, 2011). Furthermore, as TEs themselves have often been observed as the force behind gene pseudogenisation (Babakhani and Oloomi, 2018), it could be inferred that EggNOG-Mapper is annotating the TEs inserted into pseudogenes and not the 'core' pseudogene sequence itself. In contrast to the Mobilome results and again at odds with what has been reported in the literature (Lerat and Ochman, 2005), the COG category 'METABOLISM' was observed to be consistently reported at lower proportions of Con-StORF-Only clusters, than Ensembl-Only, for both the *E. coli* pangenome and inter-genera studies (22.82% to 12.82% and 39.19% to 24.55%, for Ensembl-Only and Con-StORF-Only respectively – see Tables 4.7 and 4.11). However, further complicating this, Ensembl-Con-StORF clusters were reported to have much higher levels of the 'METABOLISM' COG category than Con-StORF-Only clusters (from 12.82% to 43.40% for *E. coli* pangenome and 24.55 to 34.03% for inter-genera clusters).

The Con-StORFs investigated in this chapter have at times been both in direct contradiction and consensus with different sections of the literature regarding pseudogenes, in which little consensus can be found. While there does seem to be some functional biases in Con-StORFs, it is not consistent across cluster types (Ensembl-Con-StORF or Con-StORF-Only), and this confusing picture is being driven by the historical biases in genome annotation and function analysis present in the literature (see Background and Chapter 2). Additionally, further understanding the reasons why some Con-StORFs are 'entirely' conserved across some genera, whereas others are not (see Figures 4.9, 4.7 and 4.8), requires further investigation. The uncertainty and complexity presented in these results only add to the intrigue of Con-StORFs and reinforces the need for additional development of both StORF-Reporter and methods of utilising this nascent genomic knowledge.

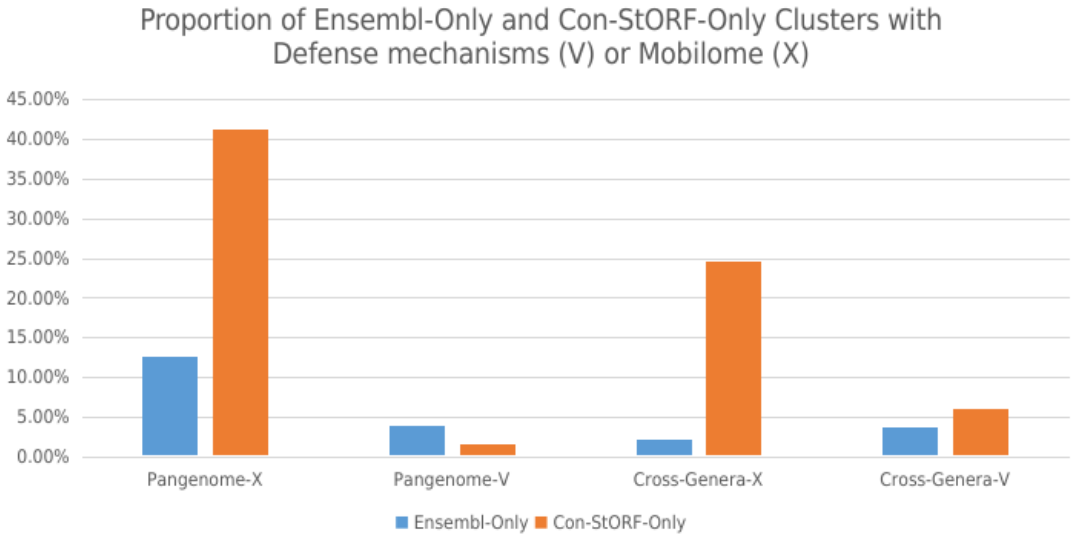


FIGURE 4.12: This plot reports the two recently added 'Mobilome' (X) and 'Defense mechanisms' (V) EggNOG COG categories for Ensembl-Only and Con-StORF-Only clusters for both the *E. coli* pangenome and inter-genera studies. The 'Mobilome' assigned COG function is more commonly reported for Con-StORF-Only clusters but this pattern is not observed for the 'Defense mechanisms' COG.

4.5.2 Con-StORFs are Distributed Widely Across the *Escherichia coli* Pangenome

We have already seen that Con-StORFs, both as part of Ensembl-Con-StORF clusters and as Con-StORF-Only clusters, are widely distributed across a large number of genera. The pangenome study of *E. coli* was undertaken to identify whether Con-StORFs formed part of the core, soft-core or accessory genes.

The distribution of Ensembl-Con-StORF clusters across the *E. coli* pangenome followed closely that of the Ensembl-Only clusters (even more so than seen in the Ensembl-StORF analysis (see Chapter 3) - see Figures 3.8 and 4.4). Further to this, while the majority of Con-StORF sequences were reported as accessory genes, we observed 29 extended core gene families and 3 Con-StORF-Only core gene families found in >=99% of the 219 *E. coli* genomes (see Table 4.6). These 3 gene families, already reported in the literature but missing from the Ensembl annotations used, have only been reported as conserved hypothetical proteins (see Subsection 4.4.2) and are prime candidates for experimental functional analysis. These gene families have also been reported in other species, also as hypothetical proteins. Lastly, in contrast to what has been reported in the literature regarding negative selection of pseudogenes (Sun, Hinnebusch, and Darby, 2008; Kuo and Ochman, 2010), the Con-StORFs detected in this analysis were often of gene-like length and were observed with only minor levels of mutation, often only in-frame stop mutations (see the sequence similarity parameters used in Chapter 3, Subsection 3.3.5). This may indicate that either the Con-StORFs reported are in the 'early' stages of pseudogenisation, or have been conserved because they are still functional.

4.5.3 Are Many Pseudogenes Functional Genes with an Alternative Genetic Code?

The detailed study of the 'non-standard' genetic code 4 using *M. genitalium* genome in our 6 model organism analysis allows us to identify what the in-frame stop codon ratio would look like in a situation when one stop codon (specifically TGA) has been entirely re-purposed to code for an amino acid. In this case (as expected) we observed a much higher frequency use of 'TGA' as validated in-frame stop codons compared to 'TAA' or 'TAG' (see Table 4.13). The other 5 model organism genomes show even (but relatively low) use of all three stop codons as in-frame stops, but this observation is hampered by the low number of validated Con-StORFs identified in each genome. When we examined the ratio of in-frame stop codons across the pangenomes of 219 *E. coli* genomes however, we observed that the ratios of stop codon usage in validated Con-StORF in-frame stop codons significantly differed from the underlying stop codon usage and 'all Con-StORFs', in the genomes (see Table 4.15). This bias (specifically toward 'TAG' in this case) may be representative of occasional alternative coding of this stop codon in this species complex.

Experimental evidence of alternative expression for the 3 canonical stop codons has existed for decades. In a two-decade old study by Kannan *et al*, the expression of *Mycoplasma* genes with internal TGA codons as full-length proteins (while at relatively low efficiency) in recombinant wild-type *Bacillus* genomes was observed (Kannan and Baseman, 2000). Subsequent studies investigated the potential for a single codon to code for different amino acids "determined by a specific 3' untranslated region structure and location of the dual-function codon within the messenger RNA" (Turanov et al., 2009). While many such studies state the rarity of this 'duality' of codon coding, our results may indicate a more extensive presence of this phenomenon in nature. As observed by Lobanov *et al* in 2010.

> *"...instead of complete codon reassignment or competition with translation termination, the same [stop] codon encoded two amino acids at internal positions of proteins. UGA was found to code for both Cys and Sec, and the dual function of UGA may occur even within the same gene." (Lobanov et al., 2010).*

Therefore, in addition to the potential for codons to code for alternative amino acids from the standard set of 20, it has been known for some time that although rare and of niche function, proteins exist which require an additional amino acid outside of that set of 20. Further to the reporting of Lobanov *et al*, Sec or selenocysteine, is canonically known as the 21st amino acid (Stadtman, 1996) and plays important roles in both pathogenicity of many bacteria and their metabolic maintenance. Sec is co-translationally inserted into specific selenocysteine proteins by recoding their opal or T/UGA codons (Zhang et al., 2006a). Additionally, the 22nd amino acid, pyrrolysine or Pyro, is universally observed to be encoded by the amber or T/UAG codon (Srinivasan, James, and Krzycki, 2002). Unlike Sec however, Pyro has only

been identified in bacteria and archeaa (Dinman, 2012), most often in a unique class of methanogenic enzymes (Ho et al., 2021). Much is still unknown in regard to the function and genesis of genes using these additional amino acids. Nevertheless, studies continue to discover that such genes are more common than once thought (Ho et al., 2021) and have revealed possible explanations in the form of symbiotic links between microbiome and host (Zhang and Gladyshev, 2007). While the results in this chapter are not able to confirm whether the Con-StORFs identified, validated or not, are definitely using these two additional amino acids, they are most likely a useful tool for their continued discovery.

As with much in biology, there are likely many reasons behind the observation of potential stop codon recoding, especially within and across different genera as seen in this chapter. Another such explanation may be found in the concept of recurrent mutation. Recurrent mutation has been seen as an important factor in evolution for nearly a century (Haldane, 1933), allowing for population level changes and adaptation. These mutations, formed independently in the same positions in genes shared across species and genera, highlighting the possibility that mutational hotspots are being identified by Con-StORFs. A number of pressures may be influencing the selection for in-frame stop mutations in these regions rather than others. It has already been observed that single-nucleotide mutations (sometimes in-frame stop or large deletions) arise repeatedly in independently evolved populations in response to changes in environment and metabolic necessity (Saxer et al., 2014). Throughout this chapter, a number of Con-StORFs were identified with in-frame stop codons whose positions were shared across both species and genera. Examples of this are shown in Figures 4.9 and 4.8. These observations may indicate that Con-StORFs are identifying additional genomic elements such as non-/functional CDS genes with alternative stop codon usages. If this is true, as alternative codon usage has often been assumed to be both codon and species-specific, it is interesting that this study has reported cases of apparent alternative amino acid coding for all three stop codons (in comparatively large numbers), across the different genera studied (see Tables 4.15 and 4.16).

The number of Con-StORFs observed with such mutations are indicative of the complexity of prokaryotic genomics, which is too often downplayed and clearly adds to the counterarguments being made to the many assumptions surrounding the 'simplicity' of prokaryotic organisms (Hunter, 2008a). Those Con-StORFs observed with shared in-frame stop positions may then be important adaptational signatures required by 'all' genomes, irrespective of phylogeny. Therefore, it could be concluded that pseudogenes, although the initial target of the Con-StORFs, were not the only type of CDS gene found in the Con-StORFs.

4.5.4 Is There In-Frame Stop Codon Preference?

The three validation analyses of Con-StORFs carried out in Section 4.4.4 revealed that all three stop codons were not used in equal frequency for in-frame stop codons.

The validated Con-StORFs for the 6 model organisms reported no clear codon preference apart from TGA for *M. genitalium* (see Table 4.13), which is explained by the well known TGA recoding in that organism. Interestingly, the narrow preference for TAG and then TAA noted in the model organism results for *E. coli* was also observed in the pangenomic study where TAG, TAA and TGA were used in preference, respectively. Additionally, TAG was recorded with the lowest triplet abundance. Interestingly, as can be seen in Table 3.9, even though TGA was the least used validated in-frame stop in both the model organism and pangenomic study for *E. coli*, it was the most common triplet (among TAG, TAA, TGA) found in this species. This is inconsistent with the usage of TGA the second most common CDS terminator in the Ensembl annotations of *E. coli* (at 28.41%).

Our inter-genera analysis of nearly two hundred genera revealed that as with the previous analysis, TGA was reported as the most common in-frame stop codon for all 40,946 pre-validation Con-StORFs but was the least commonly reported in the validated Con-StORFs. Similarly, TAA and TGA were reported as the second and third most common pre-validated in-frame stop codons respectively in the inter-genera analysis but were the second and first in the validated con-StORFs (as with the *E. coli* pangenome analysis - see Tables 4.15 and 4.16). While these results may indicate a signal which could allow identification of in-frame stop codon preference on a species level, neither the triplet abundance studies of Chapter 3 or the counts of un/validated in-frame stops report a clear pattern which could be used to validate Con-StORFs without alignment to known proteins.

4.5.5 Conclusion

Many of the misconceptions of prokaryotic genome annotations have been reassessed in Chapters 3 and 4. Through the reexamination of the too-often ignored and misleadingly named 'intergenic regions' of prokaryotic genomes, we have shown that there is still much genomic knowledge waiting to be unearthed. However, it implausible that we will be able unearth all CDS genes, pseudogenised or otherwise, without further understanding the limits of the current methods and databases. StORF-Reporter provides an opportunity to address this.

There is a perception in the field that pseudogenes are simple non-functional genomic elements. However, the very first study of a pseudogene (Jacq, Miller, and Brownlee, 1977), in addition to coining the term itself, reported that not only was pseudogenisation caused by early truncation (possibly from an in-frame stop mutation), but also observed evidence of expression and stated that ".. the pseudogene must be conserved as closely as the gene region...". As with many areas of science, it appears that subsequent studies of pseudogenes have forgotten much of the original knowledge and only now are we reaffirming what was already discovered. Pseudogenes are largely still excluded from functional experimentation and genomic analyses (Goodhead and Darby, 2015) or undergo homology only analysis (Tanizawa, Fujisawa, and Nakamura, 2018). With an ever-growing number of putative pseudogenes found to exhibit biological function, there is an emerging risk that the premature dismissal of these regions as 'pseudogenic', may lead to important functions being overlooked (Cheetham, Faulkner, and Dinger, 2020). Therefore, in the consideration of which genomic elements are assessed for biological impact, the development and facilitation of pseudogene annotation is paramount. Through the Con-StORF extension to StORF-Reporter, a homology independent method of pseudogene detection has been developed which will help reshape the functional and interactional history of prokaryotic genomes.

One of the most interesting findings of this chapter was the intriguing possibility that a number of genes are repurposing stop codons for amino acid coding, previously believed to be uncommon and present in only a few species. We found these (putative) genes utilising stop codons to code for amino acids across multiple different genera but in the same position, alluding to selective pressure in these regions and not random mutation (see Figures 4.8 and 4.9). Specifically in the example of Con-StORF cluster 1,326, while the Swiss-Prot protein alignment only covers the 'first' StORF of the Con-StORF sequence, the entire sequence across all 61 examples from 12 genera are conserved at 100% identity at the DNA level, including the internal stop codon 'TAA'. This Con-StORF, highly conserved on both of its 'StORF segments', is yet another example of not only the utility of StORF-Reporter, but also a measure of the magnitude of genomic 'Genomic Dark Matter' we are yet to uncover (Goldstein et al., 1975). These in-frame stops have long been postulated as evolutionary remainders from a period when termination was less efficient

(Lu and Rich, 1971; Alff-Steinberger and Epstein, 1994) and while it is thought that 'modern' prokaryotes no longer need them, it is unlikely they still exist without function. Further analysis into those Con-StORFs which share stop codon positions across genomes and genera, may lead to redefining the current understanding of 'stop codons' across prokaryotes. The true extent of their existence is yet unknown, because, as discussed, they are routinely reported as either truncated CDS genes or omitted from the annotations entirely.

The combination of the last two chapters have shown that targeting those regions of prokaryotic genomes which are most often overlooked will further enrich our genomic knowledge. As such, in order to remain at the forefront of genome annotation for the foreseeable future, platforms such as NCBI's prokaryotic genome annotation pipeline (PGAP) (Tatusova et al., 2016), will need to continue to expand and diversify its collection of different annotation tools. We propose StORF-Reporter to be one such tool.

Chapter 5

FrameRate: Assembly-Free Coding Sequence Profiling

Chapter Summary

Throughout this thesis I have endeavored to provide an overview of the complex and developing field of prokaryotic genome annotation. While this work has resulted in the identification and subsequent addressing of a number of limitations, the analysis has been conducted entirely on cultured and pre-assembled genomes. There is a mantra in computer science which states that an algorithm is only as good as the data it is given - 'garbage in, garbage out'. This continues to hold true for bioinformatics and especially in regards to genomic and metagenomic data.

The field of genome assembly, and, in particular, metagenomic assembly, is vast and will not be investigated here. However, in an attempt to overcome the many known limitations in metagenomic assembly, presented here is an exploratory study into the application of a rudimentary machine learning model to classify metagenomic reads as coding or non coding, thus circumventing the needs for genome assembly. Named 'FrameRate', this model can predict the coding frame(s) from an unassembled DNA sequencing reads without the need for homology based inference or any pre-computed database. Utilising the eggNOG-mapper function annotation tool, the predicted coding and non-coding frames were functionally compared to full-length protein sequences identified through a contemporary metagenome assembly and gene prediction process, produced from the same metagenomic sample.

This chapter has two aims: first, to overcome the computational and temporal resources required for large scale metagenomic functional profiling. Secondly, to study the results of the predictions to better understand the biological signature of real amino acid peptides compared to mistranslated DNA sequences and subsequently improve the utilisation of future machine learning algorithms in this space.

The concept for this chapter was developed during a research visit to King Abdullah University of Science and Technology (KAUST) in Saudi Arabia. Additionally, the initial development of the model presented here was undertaken with significant support from Wang Liu-Wei and his continued advice was crucial for the outcome of this chapter.

Software Availability: https://github.com/NickJD/FrameRate

5.1 Introduction

The advent of next-generation sequencing (NGS) has drastically reduced time and cost required for the sequencing of large, complex and niche genomes and metagenomes, from increasingly diverse environments. However, this rapid influx of DNA sequences has also presented a number of qualitative and quantitative challenges to the complex processes of genome assembly and annotation. As discussed in the Background and Chapter 2, contemporary research and computational advancements in *de novo* genome assembly and annotation have struggled to keep pace with this revolution. Subsequently, metagenomics, the process of sequencing the DNA sampled from an environment, irrespective of which species are present, has become a de facto method to overcome the many complications associated with microbial culture (Forbes et al., 2017). In these complex environments, the traditional formula of extraction, isolation and culture is often not feasible and therefore, the resultant assemblies can be a fusion of DNA sequences from different species. Metagenome-Assembled Genome (MAG) assembly has also become more prominent as a method of building individual genomes from these multi-species environments (Stewart et al., 2018). MAGs have a number of caveats inherent to genome assembly itself which are yet to be resolved, but also harbor a number of additional complications, often not encountered by single-species assembly (Alneberg et al., 2018; Chen et al., 2020b). Assembled metagenomic datasets are now in the spotlight which has highlighted the impact of these problems.

Techniques to undertake MAG assembly and annotation are improving, however the speed at which metagenomic datasets can be assembled and studied is still slow and resource heavy when compared to the rate at which new metagnomic data is being produced (Lapidus and Korobeynikov, 2021). Additionally, while many state-of-the-art methods have been designed to predict genes from metagenomes (data which is likely to contain high levels of error and fragmentation) (Noguchi, Park, and Takagi, 2006; Rho, Tang, and Ye, 2010; Zhu, Lomsadze, and Borodovsky, 2010; Yok and Rosen, 2011; Salamov and Solovyevand, 2011; Kelley et al., 2012), the resulting gene predictions are often poor quality, fragmented, of a limited number. These shortcomings make it difficult to fully exploit the high number of metagenomes present in public repositories (Klassen and Currie, 2012). Further to this, genome annotation performed on complete and high quality genomes continues to be a developing field, with many known and unknown shortcomings yet to be addressed (see Chapter 2 (Dimonaco et al., 2021)). The limitations of MAGs, combined with the increased production of metagenomic data has motivated researchers to investigate new methods for metagenomic sequence characterisation.

> *"Despite these efforts there are still no ideal laboratory methods for working with metagenomic DNA, nor a universal assembler for metagenomic data. Without a quality assembly, however, it is impossible to identify all of the members of the microbial communities being sampled, and especially those that are represented in minor amounts. Meanwhile, these yet undiscovered members play an important role not only in the life of the community itself, but also in the ways they affect the life and condition of the associated host and habitat." Lapidus et al (Lapidus and Korobeynikov, 2021).*

The historic impact of sequence read-loss, miss-assembly and miss-annotation on contemporary genome annotation methodologies, has led to a number of exploratory studies to discover novel ways for directly characterising sequence reads (without the need for assembly) from both cultured and metagenomic samples. Such examples include the taxonomic classification of DNA reads by Phymm (Brady and Salzberg, 2009), MG-RAST (Keegan, Glass, and Meyer, 2016) and Kraken (Wood, Lu, and Langmead, 2019) as successful attempts in effectively side-stepping the need for genome assembly. These methods, while making the assembly of reads unnecessary, and bypassing gene prediction, still require knowledge from large pre-computed databases. These resource intensive databases are known to contain errors and omissions which restricts their utility. Additionally, "metagenomes from less studied habitats will likely find less similarities in the databases" (Tamames, Cobo-Simón, and Puente-Sánchez, 2019), .

> *"Although differences exist, taxonomic profiles are rather similar between raw read assignment and assembly assignment methods... Regarding functional annotation, analysis of raw reads retrieves more functions..." Tamames et al (Tamames, Cobo-Simón, and Puente-Sánchez, 2019).*

It is estimated that 80-90% of all prokaryotic DNA is protein coding (Lobb et al., 2020), thus it can be inferred that the majority of metagenomic DNA reads (from prokaryotic rich environments) are also protein coding. This is further supported by the fact that (unlike eukaryotes) the frequency of introns in translated bacterial genes is very low (Belfort et al., 1995; Lamolle and Musto, 2018). However, as a DNA sequence could be transcribed in 6 different frames (3 forward and 3 reverse), it is difficult to predict whether a DNA sequence is protein coding without knowing the correct frame in advance. Previous work relating to the identification of coding frames involved the inspection of the chemical makeup and patterns of individual regions of DNA (Fickett, 1982; Staden, 1984; Gribskov, Devereux, and Burgess, 1984; Tramontano and Macchiato, 1986). This was a manually (without computers) process, routinely done in the early days of DNA sequencing. Nonetheless, in the subsequent decades, the introduction of more advanced computing methods and the availability of [meta]genome assemblies, led to the identification of entire Open Reading Frames (ORFs) being more effective.

The cost of DNA sequencing was once the limiting factor in genomics. Today, DNA sequencing is so cheap and accessible that the analysis of the resulting data is the limiting factor to knowledge discovery (Harris et al., 2019; Maguire et al., 2020). Many bioinformatic tools are now are being specifically developed to be run on personal computers. Many large-scale studies, which previously could only be conducted by large research institutions with access high-performance-computers (HPCs), are now accessible to everyone. However, metagenomic analysis which utilises vast amounts of sequencing data, is still mostly reserved to those with access to HPCs. The advent of affordable hardware designed for machine learning, often available in off-the-shelf laptops and desktops, along with the ability to process big data, has contributed to substantial progress in the development and utilisation of machine learning techniques in genomic analysis (Friedman, 2006; Jordan and Mitchell, 2015; Goodfellow, Bengio, and Courville, 2016; Jumper et al., 2021). This has resulted in the genomics community taking advantage of one specific sub-field of machine learning, known as 'deep learning' (LeCun, Bengio, and Hinton, 2015). A wide variety of machine learning techniques are now in use in practically all fields of research, and are believed to be pivotal for the future of genomics, even if not in their current form (Manrai et al., 2016; Raza, 2020; Auslander, Gussow, and Koonin, 2021).

Nature's data storage mediums, DNA and protein sequence, are paradigms of big data and as such have been greatly utilised by machine learning methods (Jumper et al., 2021). This should come as no surprise, as the vast potential of machine learning comes from its ability to analyse large-scale data and learn from such data without human intervention. However, an often common human-made obstacle in machine learning, is that the very first attempt at solving a problem involves the bypassing of many 'lesser' steps (incorrectly) deemed unnecessary. This is most commonly presented in the form of utilising training data which has already been prepared, often by various groups and processes. While this may make sense in some fields, in biology, it does not. For example, protein function prediction, arguably the most investigated application of machine learning in biology, is conducted on full-length protein sequences which have already been identified by gene prediction tools (Makrodimitris, Van Ham, and Reinders, 2020). As already discussed, not only do these tools themselves have their own limitations and biases (see Chapter 2 (Dimonaco et al., 2021)),these tools also require at least partially assembled genomes. The impact that poorly assembled metagenomes have had on the accuracy of machine learning models is seldom studied (Yang et al., 2020). While this is likely due to the inherent difficulty in identifying inaccuracies in genomic data, such error negatively affects the utility of machine learning algorithms in this area of research.

During a research visit to King Abdullah University of Science and Technology in 2019, I collaborated with Wang Liu-Wei and colleagues, on a novel deep learning approach for predicting protein–protein interactions (PPI) between viruses and humans. This method, DeepViral (Liu-Wei et al., 2021), utilised a number of different data-types or 'features' for learning, such as: phenotype, taxonomic information and protein sequence. However, the reliance on and the combination of assumed 'high quality', multi-labelled data from different curated sources makes the results difficult to interpret and limits its subsequent utility. Further to this, deep learning algorithms are inherently 'black boxes' by nature, meaning it is difficult, if not impossible, to understand the decisions they are making during the training and classifying processes. Nonetheless, and quite interestingly, one of the outcomes of the development of DeepViral was that when the model was trained on different combinations of features, sequence data was the most significant feature.

In response to these findings, I started the development of a convolutional neural network (CNN) model. Initially adapted from DeepViral (Liu-Wei et al., 2021), which itself was adapted from DeepGOPlus (Kulmanov and Hoehndorf, 2020), to bypass the need for [meta]genomic assembly for functional analysis of raw sequence reads. This model, named'FrameRate', was constructed to identify coding sequences from unassembled DNA reads by learning patterns from the coding and non-coding amino acid sequences or 'frames', from individual CDS DNA sequences. Once trained, the model then classifies each of the 6 amino acid frames from a single DNA read as either a Coding Frame (CF) or Non-Coding Frame (NCF). The rudimentary input data (only DNA sequencing reads) and binary model classification (coding or non-coding), allows for greater inspection of the decisions made by the model and subsequently their biological reasoning. The validation of such a tool is difficult. Therefore, to evaluate the performance of FrameRate predictions with real-world data, a systematic comparison was performed between the CF/NCF classified frames and CDS genes predicted from a metagenomic assembly generated from the same raw sequence reads given to FrameRate. As used in Chapters 3 and 4, the eggNOG-mapper (Cantalapiedra et al., 2021) function annotation tool was used here to annotate the sequences from both approaches. The reported COG (Clusters of Orthologous Genes) protein functions from EggNOG (Huerta-Cepas et al., 2019) were then used to compare the two approaches. Similar comparisons have been made previously between different techniques for taxonomic classification of metagenomic samples, direct from reads only or via metagenome assembly (Harris et al., 2019).

Although [meta]genome assembly techniques are improving, they are often unable to assemble a significant proportion of input sequenced reads, even if these proportions are not routinely reported in the literature (Wilkins et al., 2019; Meziti et al., 2021) (see ENA submission guide without any requirement to report the level of assembly for the upload of genomes or MAGs `https://ena-docs.readthedocs.io/`

en/latest/submit/assembly.html). The genomic information represented by these unassembled reads is typically not available for further analysis. To investigate these reads, FrameRate was used to predict the correct coding frame, allowing the analysis of reads which could not be assembled. Additionally, as metagenomic assemblies are constructed from raw DNA reads sampled from a community of species and strains, the individuality of gene function can be lost through chimeric assembly which typically ignores individual single nucleotide variants in favour of consensus based graph assembly (Arroyo Mühr et al., 2020) FrameRate reports a functional profile more similar to that of pure cultured genome assemblies. Each DNA read undergoes the same analysis, irrespective of similarity to other reads. Non-synonymous single nucleotide differences between reads are not lost as is typical in the construction of consensus assemblies.

5.2 Methods

This section will outline the development and application of two approaches applied for the functional profiling of a metagenomic DNA sample: (1) A traditional metagenomic assembly and CDS gene prediction, (2) the coding and non-coding classification of raw reads via a machine learning model. As such, much of the preparation of data is the same in both approaches and when this is the case, it is described below in Subsection 5.2.1. The metagenomic assembly process is described in Subsection 5.2.2 and the FrameRate machine learning approach is described in Subsection 5.2.3. An overview of the two approaches can be seen in Figure 5.1.

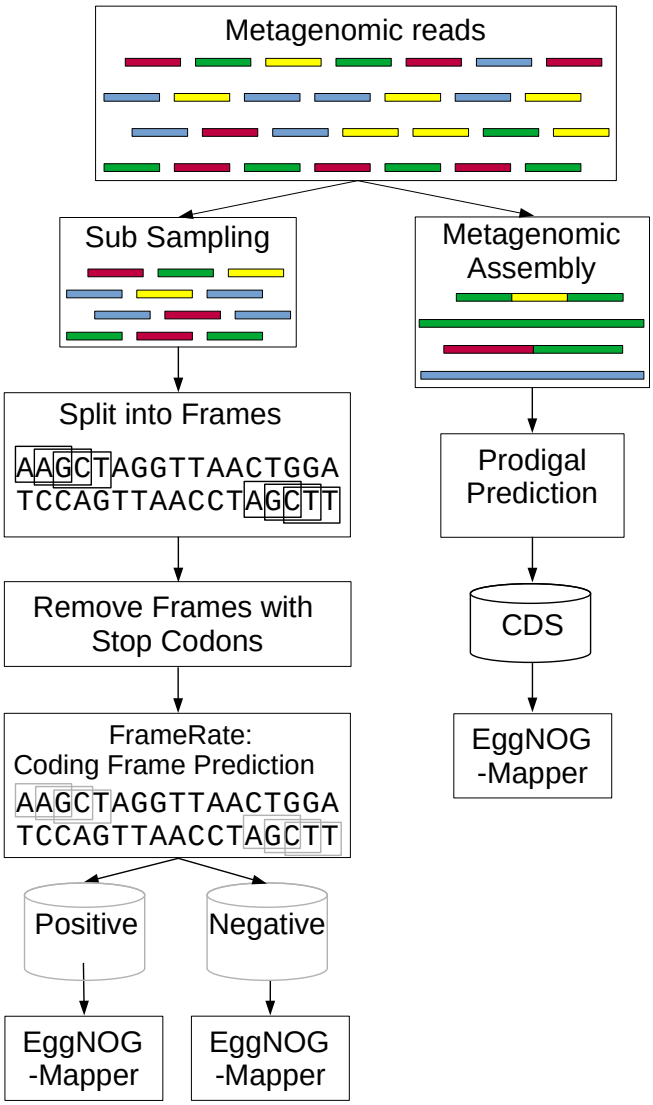


FIGURE 5.1: Presented here is an overview of the two approaches compared in this chapter for functionally profiling a metagenomic sample. The EggNOG COG functional annotations of the Coding and Non-Coding frames classified by FrameRate are compared individually to those identified from the 'traditional' CDS gene predictions by Prodigal undertaken on the MEGAHIT metagenomic assembly.

5.2.1 Metagenomic Sequence Data

To allow for a comparative study, it was important that the metagenomic data used in this chapter was not only from an environment which has been well studied, but also from one which is known to have been previously assembled and studied, complete or otherwise. The Shi *et al* sheep ruminant methane yield project (Shi et al., 2014) produced metagenomic datasets which have been used for a number of different studies (Kamke et al., 2016; Maman et al., 2020).

The metagenomic reads from sample SRR873595_1/2 were selected from NCBI project no. PRJNA202380 of the Shi *et al* study. These two samples were sequenced using an Illumina HiSeq 2000 with high-throughput sequencing from both ends of the DNA fragments (2 × 150 bp). This paired-end sequencing produced two fastq files (Cock et al., 2010), SRR873595_1.fastq and SRR873595_2.fastq. Each Fastq file consisted of 224,630,639 reads, each containing important data such as the Illumina adapters, quality scores and pairing information for each pair of reads. Trimmomatic (Bolger, Lohse, and Usadel, 2014) was used to remove these adapters and trim the reads according to default quality control parameters. This resulted in the two fastq files trimmed_paired_SRR873595_12.fastq.gz with 214,611,607 reads each (see Appendix Subsection C.1 for more detail). PandaSeq (Masella et al., 2012) was then used to pair-end join the reads from the two trimmed fastq files using default parameters. A single FASTA file, trimmed_paired_SRR873595_combined.fasta, was produced with 186,941,580 paired-end reads with the median length of 224 nt. This workflow represents a standard analysis utilising both well-known and used tools, with default parameters which is most often how genomes and metagenomes would be processed.

5.2.2 Metagenomic Assembly and CDS Gene Prediction

The paired-end FASTA file created by PandaSeq was used as the input data for the metagenomic assembly. Using the default parameters for a paired-end metagenomic assembly, the *de Bruijn* graph based metagenome assembler MEGAHIT (Li et al., 2015), was used to create the baseline metagenome for this study. MEGAHIT was chosen for a number of reasons. The most important were its ability to handle sequence paired-end reads in FASTA format, coupled with its reported fast run time and sparing resource use which has resulted in MEGAHIT being widely used in metagenomic studies (Vollmers, Wiegand, and Kaster, 2017). The assembly was not computationally possible on a stand-alone desktop computer and was therefore executed on a 32 core, 512GB memory node on the Aberystwyth University IBERS HPC Cluster (`https://bioinformatics.ibers.aber.ac.uk/wiki/index.php/Main_Page`).

CDS gene prediction was performed using Prodigal (Hyatt et al., 2010), which as discussed in Chapter 2 is not only a leading tool but also has been widely used in metagenomic and pangenomic studies. Default parameters were used, and the predictions were carried out on a stand-alone Linux desktop with 4 CPU cores and 24GB of memory.

5.2.3 FrameRate Model: Convolutional Neural Network

Neural networks are built from layers of neurons which represent weights between nodes in a graph, allowing for data to flow through different paths depending on the interaction between the data and the neuron weights. The FrameRate model is a CNN or convolutional neural network, which is a type of neural network originally designed for pattern recognition in digital image data (O'Shea and Nash, 2015). However, instead of the pixels of every line in an image being read in as one long sequence, the model was presented with a single line, or sequence, of amino acids. Additionally, whereas 'traditional' CNNs traverse along 2 dimensions of image data (2 dimensional CNN or Conv2D - think of an image width and height), as sequence data is linear or in 1 dimension, FrameRate traverses the input data along a single dimension (1 dimensional CNN or Conv1D).

5.2.3.1 Data Preparation for Training

To produce a machine learning model, the first step is to identify and prepare the data which will be used to train it. As the FrameRate model is a binary classifier, meaning its predictions are represented on a scale between 0 - 1 (< 0.5 is non-coding and ≥ 0.5 is coding), its training data must be of the same configuration. The True Positive (TP == CF) and True Negative (TN == NCF) amino acid sequence data required for the training of the model was acquired from Ensembl Bacteria (Howe et al., 2020).

From the 6,223 genomes used in the previous chapters of this thesis (described in Section 3.3 in Chapter 3), one genome (full collection of CoDing Sequences (CDS) from a single genome) from each of the 179 genera from Table 3.1 in Chapter 3 was chosen for training. Ensembl Bacteria provides both DNA and amino acid sequences for the CDS genes of their genomes. As the CFs and NCFs were required for the training of the model, the CDS DNA data files for each chosen genome were used for training.

The Python script **Get_CoDing.py** was developed to select the single genome to use from each genus and then translate the CDSs of the chosen genome into the 6 possible coding frames (1 TP and 5 TN) in their respective amino acid sequences. It did this by first scanning all 6,233 genomes and ranking each genome by file size for each genus. The largest file (containing the largest cohort of CDS sequence data from these often fragmented and incomplete genomes) was selected for each genus.

The specific genomes used as part of the training data set are listed in Appendix Subsection C.2. The DNA sequence was reported in the CDS file in the correct coding frame (the TP frame) and then the 5 alternative TN frames were also translated from this sequence. Therefore, the 727,696 DNA sequences from these selected genomes were each translated into the 6 coding frames using the universal codon table, using the original frame 0 as a basis for translation. Duplicate amino acid sequences were filtered out (from the set of all sequences from all 6 frames) during this process and the resulting FASTA file contained 4,366,176 unique amino acid sequences. Sequence similarity filtering was carried out using CD-Hit (Fu et al., 2012) at 60% length and percentage identity to further reduce the number of similar sequences. This resulted in 3,869,043 (a reduction of 11.39%) amino acid sequences available for training and the resulting FASTA file was then converted to a comma separated file with four fields of information for the FrameRate model.

The four fields of the training data file can be seen in Figure 5.2. The first field reports the genus and Ensembl provided protein ID with converted frame (0-5) number. The second field is the unique numeric ID for each set of amino acid sequences (the same ID for all frames of each CDS gene). The third field is the TP/TN (CF/NCF - 1/0) denominator so that the model knew how to correctly partition the data. The last field contains the converted amino acid sequence.

```
Mycobacterium_AGB26977_0,0,1,RTVTEARNGFNALLADAAHGINTHVIRGAKVAAHIVPAGAPI
Mycobacterium_AGB26977_1,0,0,RLRRAMGSTRCSPMPLTATRMSSGAPRLRPTSCQPARLSSTI
Mycobacterium_AGB26977_2,0,0,NGDGAQWVQRAARRCRSRHKHACHPGRQGCGPHRASRRAYHR
Mycobacterium_AGB26977_3,0,0,AKPAAGVELGDVTIAEVGGPRQQYVPFAKIDRSPGLLTECLV
Mycobacterium_AGB26977_4,0,0,LSRRRRASSWGMSLSLRWAVNPGSNTSHSSQRSIDRPVCSLS
Mycobacterium_AGB26977_5,0,0,AGVGGRRVGGCHYRGGRLTQAAIRPIVRKDRSIARFAHVPCG
```

FIGURE 5.2: Presented here is an example of a single CDS gene which has been prepared in the format needed for training the FrameRate model. Each row is a comma separated entry with important information needed for the model to interpret. The first field reports the genus and Ensembl provided protein ID with converted frame (0-5) number, the second field is the unique numeric ID for each set of amino acid sequences (1 for each CDS gene), the third field is the coding/non-coding (1/0) denominator for the model to correctly partition the data and the last field is the converted amino acid sequences.

As discussed and investigated throughout this thesis, alternative start codon usage is likely more common than is generally reported in genomic data. As such, it must be accounted for in the training data. Whether it is ATG or another codon which is used to signal transcription initiation, the codon is still translated as Methionine by the initiator tRNA (Sherman, Stewart, and Tsunasawa, 1985). However, in the 'universal' coding table, only ATG is routinely encoded as 'M'. To account for this, and for the fact that 'all' coding amino acid sequences would technically start with 'M', the first amino acid was trimmed from all TP and TN training sequences.

This was essential, as if all coding sequences started with M, the model could 'incorrectly' learn that all sequences which started with M are coding. Complicating this further but not requiring additional changes, M can often appear in the middle of a coding sequence and also in out of frame non-coding sequences (by an out-of-frame ATG codon). Lastly, the universal identifier of stop codons in amino acid sequences, '*', if present in the out of frame non-coding sequences, were removed from the training data.

5.2.3.2 Building and Training the Model

The balance of a dataset is an important factor in the ability for any machine learning algorithm to accurately model the the entirety of the data (Hall et al., 2009), not just those which are overrepresented (Japkowicz and Stephen, 2002). There are 5 TN frames for every 1 TP frames in the training data, resulting in an unbalanced data set. This level of bias can often lead to overfitting (Dietterich, 1995) which is when the model learns from a too closely similar or limited set of data points, producing a model which is likely to fail to fit additional data or predict future observations reliably (Batista, Prati, and Monard, 2004). To help mitigate this, during the process of loading in the training data for the model, only 1 of the possible 5 TN frames are included for every TP frame. This is done by uniformly at random selecting a number between 1-5 for each CDS gene and using that number to select the frame, 1-5, to be loaded into the model.

To train the model on the median expected paired-end Illumina sequencing read length of 224 nucleotides (post-trimming), the input length of each amino acid sequence was set to 75 ($224 \div 3 = 74.6$). A sliding window was used to extract amino acid sequences of length 75 from the longer sequences, with a step size of 50 to move the sliding window, which would better replicate the real positions of reads. For those CDSs which were shorter than 75 amino acids, two methods of 'sequence padding' were evaluated. The first was to repeat the sequences (Repeat-Sequence-Padding or RSP) up to the maximum length of 75 and the second was the more standard '0-End-Padding' (0EP). Sequence padding is a developing field in bioinformatic machine learning and there are many studies which have attempted to investigate the influence of various sequence padding methods on their resulting models (Rio et al., 2020). The results of the two approaches undertaken in this study are discussed below.

The input to the neural network model of FrameRate is a one-hot encoded matrix. This matrix contains the numerical representation of each amino acid sequence (see Figure 5.3). Each amino acid is given a numeric value between 0-19 and is used by the one-hot encoding preprocessor to compute the matrix (see Listing 5.1).

```
aaindex = {'K': 0, 'H': 1, 'T': 2, 'V': 3, 'Q': 4, 'L': 5, 'D': 6, 'R': 7,
    'A': 8, 'G': 9, 'Y': 10, 'F': 11, 'S': 12, 'N': 13, 'P': 14, 'C': 15,
    'M': 16, 'I': 17, 'W': 18, 'E': 19}
```

LISTING 5.1: Amino acid to numeric identifier conversion table used for one-hot encoding matrices

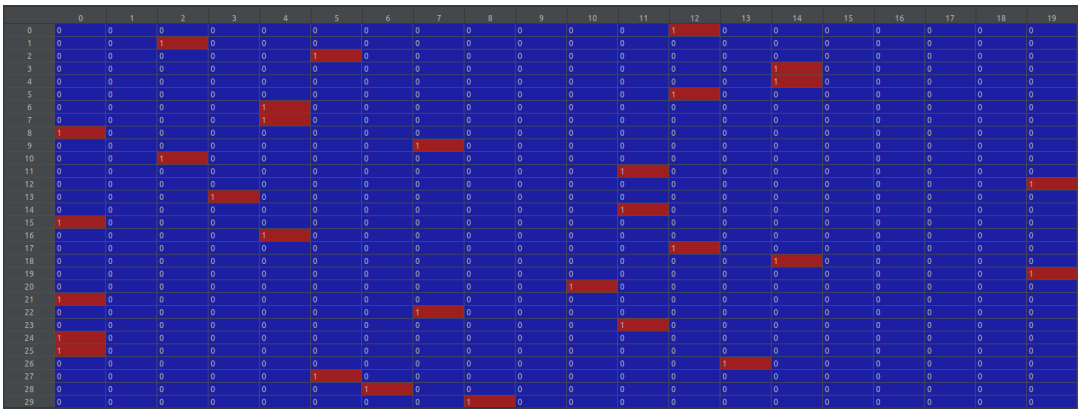


FIGURE 5.3: An example of an amino acid sequence represented as a matrix after one-hot encoding. Each amino acid position (vertical 0-74 but displayed here cut down to 0-29) is encoded with either a 0 (blue cell) or 1 (red cell) for each of the 20 canonical amino acids (horizontal 0-19)

The FrameRate model, adapted from DeepGOPlus and DeepViral, is a convolutional neural network (CNN) which consists of a 1-dimensional convolution (Conv1D) that uses max pooling (facilitates downsampling or 'chunking' of the data) with fully connected layers (a 'brute-force' approach which allows each layer of the neural network to observe the entirety of the previous layer which, while computationally expensive, allows for learning of all features of the previous layer). Additionally, as a binary classifier, FrameRate classifies each input sequence on a scale between 0-1 (< 0.5 is non-coding and ≥ 0.5 is coding).

To tune the model, the following hyperparameters were trialed to evaluate which groups of settings performed the best: the maximum size of the convolution filters (defines the range of kernel sizes used to extract motif level information from the sequence - i.e. 17, 33 and 65), the number of the filters (defines the diversity of possible motifs - i.e. 8 and 16), the size of the max pooling layers (i.e. 1, 10 and 20) and the number of neurons in the fully connected layers (the number of decisions per fully connected layer - i.e. 8, 16 and 32). The following hyperparameters were then fixed throughout all model training and experiments: convolution filter size of 17, 8 filters, a max pooling layer size of 1 and 8 neurons for the dense

layers. Additionally, to prevent the neural network from overfitting, dropouts (Srivastava et al., 2014) were used for the convolutional and dense layers with a rate of 0.5. Dropout is a technique where randomly selected neurons are switched off during training to prevent the model from only learning from a small segment of the data. LeakyReLU (Nair and Hinton, 2010) was used as the activation function (used to calculate whether a given neuron should be activated when presented with an input) for the dense layer with an alpha set to 0.1. The model was trained using the Keras library (Chollet et al., 2015) and performed on a Nvidia GTX 1060 GPU. Model evaluation was performed using the Scikit-learn library model 'accuracy_score' (Pedregosa et al., 2011) on the testing subset of sequences the model had not seen. The full set of sequences was split as follows: 80% for training, 10% for validation during the training of the model and 10% for testing the model after it has finished training. The model did not see the 10% testing data until after training.

5.2.3.3 Classifying

The classifying of sequencing reads consists of 3 stages: (1) converting the DNA sequencing reads into all 6 potential amino acid sequences and applying stop codon filtering, (2) loading the pre-trained model and (3) performing the classification on each amino acid sequence independently.

The DNA sequence reads are read in one at a time. Conversion of each of its 6 possible amino acid sequence frames is done using the universal codon table. Each frame is then truncated to a maximum of 75 amino acids and checked for stop codon positions (*). The frames which have stop positions between > 9 and < 64 are filtered out (first and last 10 amino acids, counting from 0-74). This allows for a minimum of 55 amino acids if a sequence has two stops at the ends of that range, which is still nearly double the Short-ORF length used in Chapter 2. For each DNA read, the remaining frames are then recorded with their frame identification, 0 to 5 (respective of their original order - 0 indicating the frame in which the DNA sequence was originally reported by the sequencer) and stored in a Python dictionary, where the key is the unique DNA read name and the value is a list of the remaining converted frames. Next, the pre-trained FrameRate model is loaded using the Keras library. Once the model has been initialised (either for CPU or GPU classification with a parameter switch), the remaining amino acid frames are classified. Once each of the amino acid sequences have been through the classifier and have been given a confidence score between 0-1, using the unique IDs of each DNA sequence, the scores are then assigned back to the frames in the dictionary made earlier.

With the frames from each DNA sequence read now having their own confidence score, they can be written to disk as either coding or non-coding amino acid sequences. This is done in an iterative process for each set of frames for each DNA read by writing out to one of two FASTA files, coding or non-coding, depending on the confidence score for each frame. First, the frame which has the highest score is

identified and depending on its confidence score (< 0.5 non-coding and $>= 0.5$ coding), written out in FASTA format with its frame and score recorded in the sequence identifier (see Listing 5.2. The remaining frames are then filtered using the same binary classification and then written out to FASTA with the same additions made to their sequence identifiers. The confidence scoring system allows for more than one frame per DNA read to be classified as coding. Where this is the case, the frames classified as coding but which were not the highest scoring, are also written out to the same 'coding' FASTA file and also an additional file: 'multi-coding'. This file *'multi-coding.txt'*, is used to record the DNA sequences which have been predicted to have more than one coding frame and are written out with the DNA sequence's highest scoring frame and the additional frame(s) for later inspection.

```
>SRR873595:::1167:0:0:0:;0.999013_Frame:4_Score:0.58
ERTHQNAPSLFVPEPKTLHYPPSLPFEEEVEIFFAIQMRKKYHLLLDCLDRLCLAGYTQSGDGRVTCLFLF
>SRR873595:::2486:0:0:0:;0.998817_Frame:1_Score:0.91
YLCTMFVVMMTLFVIVLVGYGAGKLGYLGGDFDRQLSRLVINMTCPALILSSAMTGELPDREYILPLLLISVVTY
>SRR873595:::3064:0:0:0:;0.999312_Frame:5_Score:0.99
INIAGAERYRAITTSHIRNADGAYLVYDITNSSTFENIGFWLETVKKATDDNIVIYLVGNKADLIDSSGRNRRVT
>SRR873595:::4456:0:0:0:;0.986443_Frame:6_Score:0.56
IQSGESDKNGRMVKHGSSALRCVLMRCADSFALHNPVVYEYKLKKMNEGKFFRVALSHVAKKLIRTIYTLEKNDL
>SRR873595:::5428:0:0:0:;0.989339_Frame:2_Score:0.82
LQHPKDIVEGSEAWDAVPDLFLVLVSEASNTSLSPALSLRVYIIYIPVFLSYSLPPFFSFF
>SRR873595:::7972:0:0:0:;0.999263_Frame:2_Score:0.74
IFACRNKTSLMLDRTQTIEKLKSTRQYFSEHYGVSSMLLFGSVARNEQKEDNDVDVCVEMKPNLFKQAGVK
>SRR873595:::9738:0:0:0:;0.999549_Frame:6_Score:0.92
ELEASVSLLDETTSTLEEDRSADSATNVALTSTSKFGISKAQLPSTAVLPIPHIPGFVTLPGSTVVQGTIVLESR
>SRR873595:::9738:0:0:0:;0.999549_Frame:1_Score:0.88
VNRSLDSKTIVPCTTVEPGNVTNPGMWGMGSTAVEGSCAFDMPNFDVEVSATFVAESADLSSSSVEVVSSSSETL
>SRR873595:::9884:0:0:0:;0.987154_Frame:5_Score:0.63
TERRWAV*TALSKTEFRSGEGRGCNLSGSGRYPGNEAVIFDCSELRWHHASSQTNELPSLQQDGSFLFALAPV
>SRR873595:::11158:0:0:0:;0.998169_Frame:6_Score:0.58
KRACA*SWARLSDTPSERRAGWNGLLGSAATQRDPRLPPIGRYGWVQGFLPPALSNIWSNREKRAALVLDRG
```

LISTING 5.2: An example output of the FASTA file for the frames classified to be coding by FrameRate. As can be seen even in this small selection, there is a large distribution in confidence scores, ranging from 0.58 to 0.99.

5.2.4 Preparing Data for Comparisons

In this study, a number of different comparisons were conducted between the CDS genes predicted by Prodigal and the unassembled metagenomic DNA reads.

Firstly, a direct comparison between the CDS genes and a subset of DNA sequences which aligned to them was made. To do this, a Bowtie2 (Langmead and Salzberg, 2012) index was built from the MEGAHIT assembly of contigs longer than 1,000 bp (the same which were given to Prodigal to annotate). According to the analysis undertaken in Chapter 2 and the literature, the majority of CDS genes are longer than 1,000 bp (Konstantinidis and Tiedje, 2004; Xu et al., 2006). Additionally, this cutoff has often been used in important studies of metagenomic data (Walt et al., 2017). Once this index was built, a Sequence Alignment/Map (SAM) file (Li et al., 2009) was created from the file containing the filtered and pair-ended reads, *'trimmed_paired_SRR873595_combined.fasta'* and the Bowtie2 index. Using IntersectBed (Quinlan and Hall, 2010), the reads which aligned (or intersected) with the GFF file produced by Prodigal were reported as a BAM file. This file, *'megahit_assembly_contigs_Min_1000_aligned_gff_aligned.bam'*, was then converted to a FASTA file with samtools. This final FASTA file contained 132,254,283 of the original 186,941,580 (70.75%) pair-ended reads.

Next I undertook the same process but instead of using the GFF file provided by Prodigal, I used the entire collection of MEGAHIT assembled contigs and extracted the reads which did not align to that assembly. This resulted in 50,412,461 reads which were not part of the metagenome analysis. In order to produce a baseline set of requirements for FrameRate, two core subsampling routines were undertaken using the Seqtk (HengLi, 2018) toolkit. The first consisted of a collection of 1 million reads which were randomly subsampled from the complete set of 186,941,580. The second consisted of 10% of the complete set of reads and was also randomly subsampled. Additionally, to study the 'Dark Matter' of this metagenome, a set of 1 million reads were subsampled from the set of reads which did not align to the metagenome assembly. These three 'shallow' subsamples were also classified with FrameRate and were functionally compared to the metagenome CDS genes predicted by Prodigal. Further information for these datasets can be seen in Table 5.2.

5.2.5 EggNOG COG Functional Annotation

To perform the functional profiling of both the CDS genes predicted from the metagenome assembly and the FrameRate predicted coding frames, eggNOG-mapper was used (Cantalapiedra et al., 2021) (see Subsection 3.3.5 from Chapter 3 for further detail).

eggNOG-mapper was installed and run locally. It utilised either the blastx or blastp version of the DIAMOND sequence alignment tool (Buchfink, Reuter, and Drost, 2021). In this study, the blastx option translates all 6 amino acid frames from each DNA sequence and aligns them all to a protein database and returns the frame with the 'best hit'. The blastp option is used to align pre-translated amino acid sequences to the same protein database. These alignments were done to identify homologs through sequence similarity using default parameters and relies on the EggNOG database (Huerta-Cepas et al., 2019; Galperin et al., 2021) of orthologous groups (OGs), which covers thousands of bacterial, archaeal, and eukaryotic organisms. This provided pre-computed DIAMOND database contains phylogenies inferred for each OG which has recently been optimised for metagenomic data sets. eggNOG-mapper can output a number of different annotation types such as, but not limited to, predicted protein name, KEGG pathways (Kanehisa and Goto, 2000), modules, orthologs, Gene Ontology labels (Ashburner et al., 2000), and COG functional categories (Tatusov et al., 2000). The output format is tab separated values and a Python script has been written to extract the COG categories and calculate the proportion of sequences with a hit to a COG and the distribution of COG categories.

5.2.6 Metagenome and Hungate Collection CDS Gene Alignment

A set of previously assembled prokaryotic genomes and their predicted CDS genes, which are known to live in the same ruminant environment as our metagenomic sample, were used to undertake another comparison between our two sets of sequences.

The Hungate collection of 493 cultured bacteria and archaea genomes is believed to "... represent ~75% of the genus-level bacterial and archaeal taxa present in the rumen." (Seshadri et al., 2018). To compare our results to another rumen metagenomic sample, the CDS genes were identified with Prodigal (see Table 5.2 for more detail) in the Hungate Collection. These were used to quantify the level of alignment of our set of metagenome CDS gene and of our set of FrameRate sequences. To do this, a DIAMOND protein database was created from the amino acid sequences of the Hungate genes and then using the DIAMOND blastp option with default parameters and a minimum bit score of 60, it was possible to align our metagenome CDS genes and FrameRate amino acid sequences. The same process was undertaken for the metagenome CDS genes where a DIAMOND protein database was created from the metagenome CDS gene protein sequences.

5.3 Results

The results section of this chapter is separated into three main parts. First, technical aspects of the FrameRate model classification process are reported to provide an overview of the training and utility of the method. Second, a comparative metagenomic profiling analysis between the FrameRate and traditional metagenomc assembly approaches. This includes a sequence alignment analysis between the FrameRate classified CFs and NCFs of the metagenome sample to the CDS genes identified from the assembled metagenome and Hungate Collection of cultured genomes. A functional analysis was also conducted which consisted of the predicted CFs and NCFs compared to the CDS genes identified from the metagenome. The third section reports the time and computational resources required to undertake the different analyses as to investigate the utility of FrameRate as a method for rapid and resource sparse metagenomic profiling.

5.3.1 FrameRate Classifier Overview

5.3.1.1 Parameterisation

Machine learning algorithms could be described as collections of parameters or rules, which through an iterative procedure, are used to search for patterns in a given dataset to 'learn' from the data. The established mechanism used to select these rules is through iteratively making minor changes (undertaken by the algorithm itself) to the set of rules to 'find' a selection which allow for the model to learn the most.

Often in neural networks, there are 2 main different types of these rules: (1) parameters, those the model chooses itself throughout the training process, (2) hyperparameters, those we can choose at the start of the training process. As reported in Subsection 5.2.3.2, the hyperparameters investigated in this study were: convolution filter size, number of filters, max pooling layer size, number of neurons in the fully connected layers and lastly, the sequence padding. These parameters were chosen via an iterative process where the model was trained on the same set of sequences and training split (train, validate and test) as described in Subsection 5.2.3.2. For each training session, different parameters were selected to allow for all possible combinations.

Interestingly, although also possibly to be expected due to the simplicity of the model, it was only the max pooling size and sequence padding which showed any discernible difference during the parameterisation of the model. Max pooling, or pooling layer size, is (in our specific case) the dimension or step size that the model uses when scanning the input data (amino acid sequence) for patterns. Fundamentally, the smaller the pooling size, the fewer amino acids are inspected by the model at any time. The importance of max pooling sizes and padding on sequence data has somewhat been observed in previous studies (Koo and Eddy, 2019; Rio et al., 2020).

As can be seen in Figure 5.4, depending on the length of the amino acid sequence, the pooling size and the type of sequence padding utilised (Repeat-Sequence-Padding or 0-End-Padding), notable differences in the performance of FrameRate are observed. The Repeat-Sequence-Padding produced a more accurate model for the larger max pooling sizes, but once max pooling is reduced to 1, allowing for a more detailed overview of the sequences, 0-End-Padding became the more accurate padding option. While the model peaks at 74.24% accuracy for Repeat-Sequence-Padding, it was able to achieve 91.57% accuracy with 0-End-Padding and all other hyperparameters were set the same (max pooling of 1). This result is somewhat replicated in the accuracy of the shorter sequences (less or equal to 50 amino acids) which are those which undergo the most padding. However, while the accuracy of prediction for the short sequences reduces with lower max pooling sizes for Repeat-Sequence-Padding, it increases for 0-End-Padding. These results are computed from the model after it has been trained on the stated parameters and then evaluated against the same testing dataset which the model is not shown during training (see Subsection 5.2.3.2 for further details). Lastly, while requiring further analysis, not only do these results showcase the importance of parameterisation of a neural network, but also how the combination of parameters can have noticeable impacts on the resulting performance.

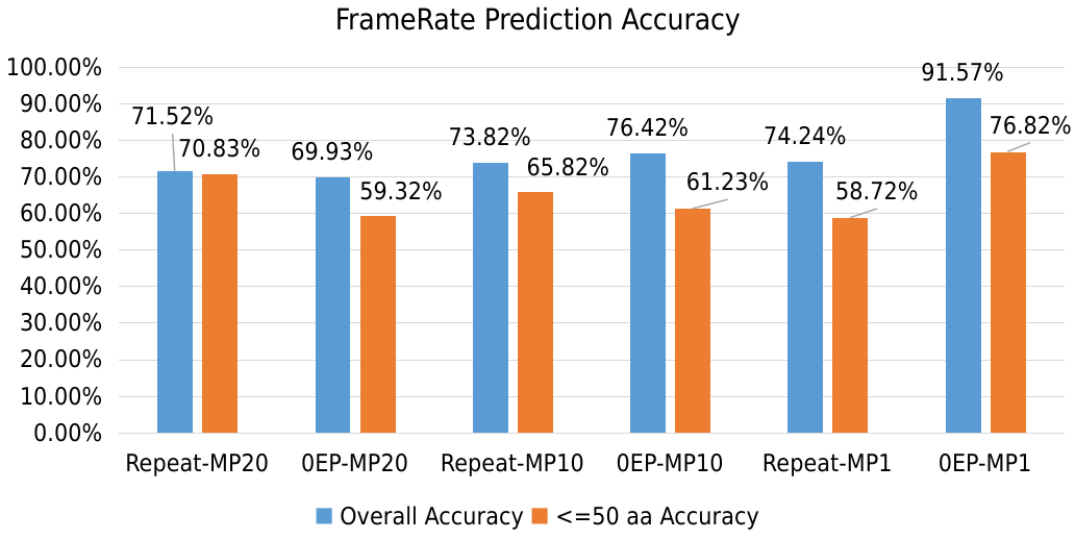


FIGURE 5.4: Presented here are the parameterisation results of the FrameRate model. Each bar plot reports the results of the model accuracy on the same training, testing and validation data but with different parameters. Parameters are reported here as: Repeat is the 'Repeat-Sequence-Padding', 'MP' is the 'max pooling size', 0EP is the '0-End-Padding'. 'Short Read Accuracy' reports the accuracy for sequences less or equal to 50 amino acids.

5.3.1.2 Tuning of Classification Scores

Although FrameRate is a binary classifier (classifying amino acid sequences as either 1 (CFs) or 0 (NCFs)), the actual output of the model is a confidence score between 0-1 (< 0.5 for 0 and $>= 0.5$ for 1). As seen in Figure 5.5, there is a difference between the reads aligned to Prodigal CDS and the unassembled reads that are classified as CFs. The CDS-aligned sequences are classified with higher confidence scores for both CFs and NCFs compared to the unassembled reads (1 is 'highest' for CFs and 0 is 'highest' for NCFs). Interestingly, there was also a difference observed in the NCF scores, with the unassembled reads reporting higher (less clear cut) scores.

Confidence scores are often split into categories such as 'high-confidence' and 'low-confidence' to help reduce the number of false predictions. For example, by examining Figure 5.5, a minimum score of 0.75 and a maximum score of 0.15 could be used to filter out CFs and NCFs, respectively. Additionally, it is interesting to note that the model is seemingly more confident with its predictions of NCFs than CFs. However, without solid ground truth data, it is not clear where in the observed distributions, the false positive and false negative predictions lay. Nonetheless, with further analysis, the differences observed here could potentially be used to not only improve prediction accuracy, but also help distinguish sets of sequences without homology signatures to known proteins.

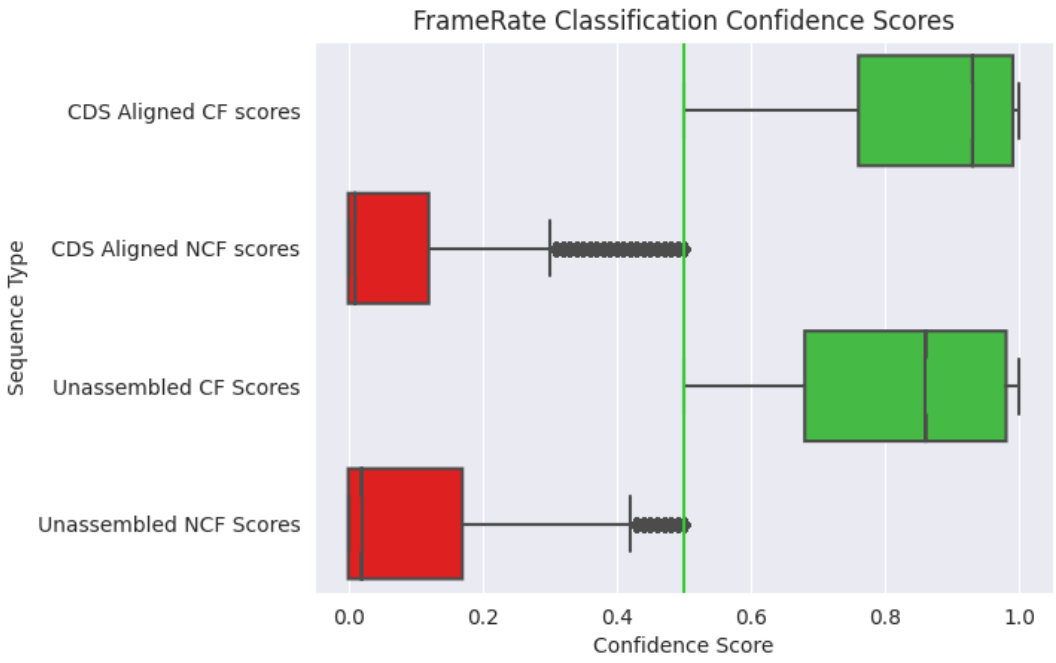


FIGURE 5.5: Presented here are the FrameRate model confidence scores for the reads which aligned to the Prodigal CDS genes from the metagenome assembly and those from the set of unassembled reads. There is a subtle difference between the CDS aligned and unassembled sequences. The scores are reported by the model between 0-1, where 0 is the best confidence score for Non-Coding Frames (NCFs) and 1 is highest confidence score for Coding Frames (CFs).

5.3.1.3 Proportion of Coding Frames per Read

Fundamentally, FrameRate was built to identify the set (between 0 to 6) of CFs from the six possible frames for each read, since a read could potentially contain more than one coding frame, possibly from overlapping genes. The first step in identifying this set of CFs was to remove the frames which had internal stop codons (see Subsection 5.2.3.3 for further details). Interestingly, as can be seen in Table 5.1, the majority of the 6 possible frames for the reads in all sets, including those which aligned to the Prodigal CDSs were filtered out because of internal stop codons. In fact, many entire reads were removed because all 6 frames contained internal stop codons. As discussed in Chapter 4, out-of-frame stop codons are very common, both in and outside of CDS sequences, and may explain these results. Additionally, apart from the '1m - Subsample Raw Reads' set, the proportion of frames classified as CFs per read (after filtering) was similar across the other 3 set of reads. Therefore, these two findings could indicate that filtering out frames with stop codons is likely to be filtering out a notable number of CFs.

Reads	Total Reads	FrameRate CFs	FrameRate Frames Post-Filtering (mean)	FrameRate CFs per Read (mean)	FrameRate NCFs
20% - Prodigal CDS Aligned Reads	26,445,973	24,910,739	1.75	1.20	12,978,414
10% - Subsample Raw Reads	18,691,790	10,085,622	1.54	1.26	5,899,564
1m - Subsample Raw Reads	1,000,000	119,092	1.39	1.08	157,404
Unassembled Reads	50,412,462	43,103,775	1.1	1.25	32,848,511

TABLE 5.1: The number of Coding Frames (CFs) predicted by FrameRate for each set of reads (column 1), the proportion of frames which remain after filtering for each read (0-6), the proportion of CFs classified for each read and finally the number of Non-Coding Frames.

It could be assumed that the vast majority of reads would only have one CF and at most 2. As shown in Figure 5.6, the proportion that FrameRate classified as CFs from the reads which aligned to the Prodigal CDS genes followed this expectation. Interestingly, there were 5 reads which were each reported with 5 CFs. One of these 5 reads, '>SRR873595:::126731668:0:0:0:;0.999093' was given FrameRate confidence scores of 1.0, 0.79, 0.85, 0.54 and 0.51. Using the minimum score of 0.75 to filter less likely CFs, this read would be left with 3 CFs.

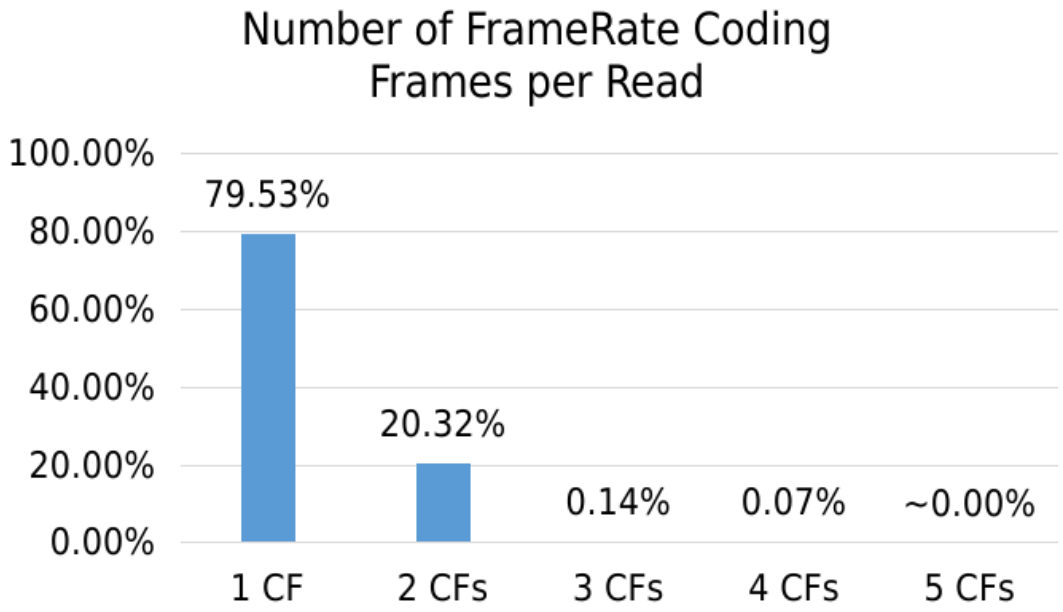


FIGURE 5.6: The proportion of 20% subsampled reads which aligned to the metagenome predicted CDS genes with either 1, 2, 3, 4 or 5 (out of the possible 6) Coding Frames predicted by FrameRate. These results reflect what would be expected when considering that all of these reads were reported as aligning to CDS genes, and the possibility that a proportion of these reads could contain two or more genes overlapping each other. To reduce the number of false positive predictions, the FrameRate model confidence scores could be used as a filter. The number of reads in each category are: '1CF' - 16,426,573, '2 CFs' - 4,197,439, '3 CFs' - 29,565, '4 CFs' - 142, and '5 CFs' - 5.

5.3.2 FrameRate vs Metagenome Assembly: Metagenomic Profiling

This section of the results focuses on the sequence function and alignment analysis of the CFs and NCFs predicted by FrameRate, and the CDS genes identified in the metagenome assembly. In order to undertake a comprehensive and comparative analysis, a ground truth functional profile was needed. A metagenome assembly was created from the same set of reads that will be classified by FrameRate and then CDS genes were identified with Prodigal.

Of the 186,941,580 paired reads which remained after the preprocessing described in Subsection 5.2.1, MEGAHIT was able to assemble 73.03% into the metagenome assembly used through this chapter (see Table 5.2 for detailed number of reads and sequences used in this study). This assembly consisted of 539,021 contigs with a median length of 1,566 bp and a N50 of 2,999 bp. Prodigal identified 1,647,050 CDS genes from these contigs. Additionally, Prodigal identified 1,436,646 CDS genes from the Hungate collection of 493 genomes. These CDS genes are later used as another set of protein sequences to which the frames classified by FrameRate are aligned.

Data	Number of Seqs	Median Length **[SD]**	Min Length	Max Length
Paired Reads	186,941,580	224 **[25.04]**	38	298
1m Random Subsample Raw Reads	1,000,000	226 **[15.82]**	46	298
10% Random Subsample Raw Reads	18,691,789	224 **[25.05]**	46	298
MEGAHIT Assembly (Min 1,000bp)	539,021	1,566 **[4,230.01]**	1,000	285,919
Unassembled Reads	50,412,462	218 **[38.02]**	39	298
Metagenome Prodigal CDSs	1,647,050	606 **[630.41]**	60	45,444
Prodigal CDS Aligned Reads	132,254,283	226 **[13.80]**	39	298
20% - Prodigal CDS Aligned Reads	26,445,972	226 **[13.79]**	39	298
Hungate Prodigal CDSs	1,436,646	855 **[730.53]**	87	45,585

TABLE 5.2: The number of sequences (paired reads, contigs or CDSs) for each dataset used in this chapter separated into three groups. (1) This first group of 3 rows describes the raw reads without the input of any metagenome assembly: the complete set of paired reads which were used to form the metagenomic assembly, 1 million randomly subsampled reads used in the shallow profiling study, and 10% randomly subsampled reads which were also used in the shallow profiling study. (2) This group reported the reads and CDS genes reported from processing the metagenome assembly: First is the complete set of contigs formed during the metagenomic assembly with a minimum length of 1,000 bp. Second is the set of raw reads which were not assembled into the metagenome assembly. Third is the set of CDS genes predicted by Prodigal from the metagenome contigs. Fourth is the number of reads which aligned to the Prodigal CDS gene sequences. Fifth is the 20% subset of reads which aligned to the Prodigal CDS gene sequences which were used later in this study.(3) the number of Prodigal CDS genes predicted from the Hungate collection of genomes. Standard deviation is abbreviated as **[SD]** and all sequence lengths are reported in nucleotides.

5.3.2.1 Alignment of FrameRate-Classified Frames

Machine learning algorithms require ground truth data to undergo a fair and comparative analysis. The core ground truth data in this study is the set of CDS genes predicted by Prodigal from the MEGAHIT metagenome assembly. The Coding and Non-Coding frames which aligned to those CDS genes underwent a sequence similarity alignment using DIAMOND.

5.3.2.1.1 Alignment of FrameRate-Classified Frames: Metagenome CDS Genes

As can be seen in Table 5.3, there is an extreme disparity between the proportion of CFs which aligned to the Prodigal predicted CDS genes and the proportion of NCFs which aligned. While taking into account the large numerical difference in the number of sequences between the two groups, 24,910,739 and 12,978,414 respectively, these results suggest that FrameRate is accurately classifying the majority of its frames. Furthermore, this analysis reaffirms the utilisation of tools such as DIAMOND and eggNOG-mapper (which uses DIAMOND) in this chapter for aligning sequence data of these short lengths (~75 amino acids) for functional annotation.

Type	Total Reads	FrameRate Coding [%]	FrameRate Non-Coding [%]
Prodigal CDS genes (20%)	26,445,973	19,888,300/24,910,739 **[79.84%]**	445,746/12,978,414 **[3.43%]**

TABLE 5.3: The proportion of classified Coding and Non-Coding Frames which aligned using DIAMOND blastp (protein-protein sequence alignment) to the full set of metagenome Prodigal predicted CDS genes. The frames were classified from the same 20% subset which has been used elsewhere in this study.

Observing Figure 5.7, it is clear that, with the caveat that the sequence alignment method used is heuristic, the number of False Negatives (when a CF is incorrectly classified as a NCF), often a problem in machine learning, is very low here. Additionally, the possibility of overlapping genes should be taken into account when analysing these results. A read which contains two or more genes overlapping each other will potentially contain enough of a 'signal' for FrameRate to classify both frames as coding and to be aligned by DIAMOND blastp. On the other hand, a read may contain enough 'signal' for one method but not the other. However, there are no clear cutoffs or 'minimum level of signal' known for either approach.

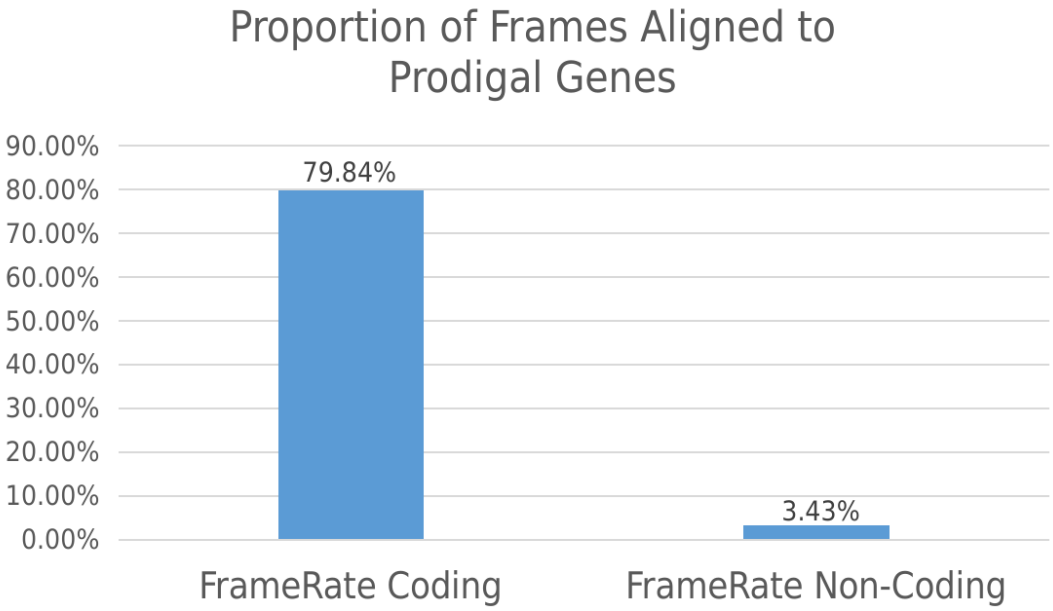


FIGURE 5.7: The striking difference in the proportion of Coding and Non-Coding Frames as classified by FrameRate that aligned to the Prodigal CDS genes predicted from the metagenome assembly is presented in this bar chart. Clearly shown is both the high level of correct Coding predictions and low level of incorrect Non-Coding predictions, according to the metagenome CDS gene alignment.

5.3.2.1.2 Alignment of FrameRate Classified Frames: The Hungate Collection CDS Genes So far, all analysis has been performed on the same set of metagenomic reads, whether it be conducted on the metagenome assembly created with them or through their direct translation into amino acid sequences with FrameRate. Therefore, to independently evaluate the three sets of sequences (the Prodigal CDSs from the metagenome assembly, and the FrameRate CFs and NCFs from the 20% subset of CDS aligned reads) the CDS genes reported in the Hungate collection of 493 genomes were used. The three sets of sequences were aligned to the Hungate CDS genes using DIAMOND blastp and as can be seen in Tables 5.4 and Figure 5.8, the CDS genes and CFs were observed with a difference of less than 12%. This difference could be in part explained by the inequality of aligning the full-length CDS sequences compared to the 75 amino acids from FrameRate. Additionally, the 1.85% of NCF sequences classified by FrameRate which did report an alignment is similar to that observed in Figure 5.7, which reported the FrameRate alignments to the CDS genes predicted from the metagenome assembly.

Type	Total Sequences	MG Prodigal CDSs	FR Coding [%]	FR Non-Coding [%]
Hungate Collection	14,36,647	1,131,932/1,647,050 **[68.72%]**	14,183,017/24,910,739 **[56.94%]**	240,064/12,978,414 **[1.85%]**

TABLE 5.4: The proportion of classified Coding and Non-Coding Frames which aligned using DIAMOND blastp (protein-protein sequence alignment) to the full set of metagenome Prodigal predicted CDS genes. The frames were classified from the same 20% subset which has been used elsewhere in this study.

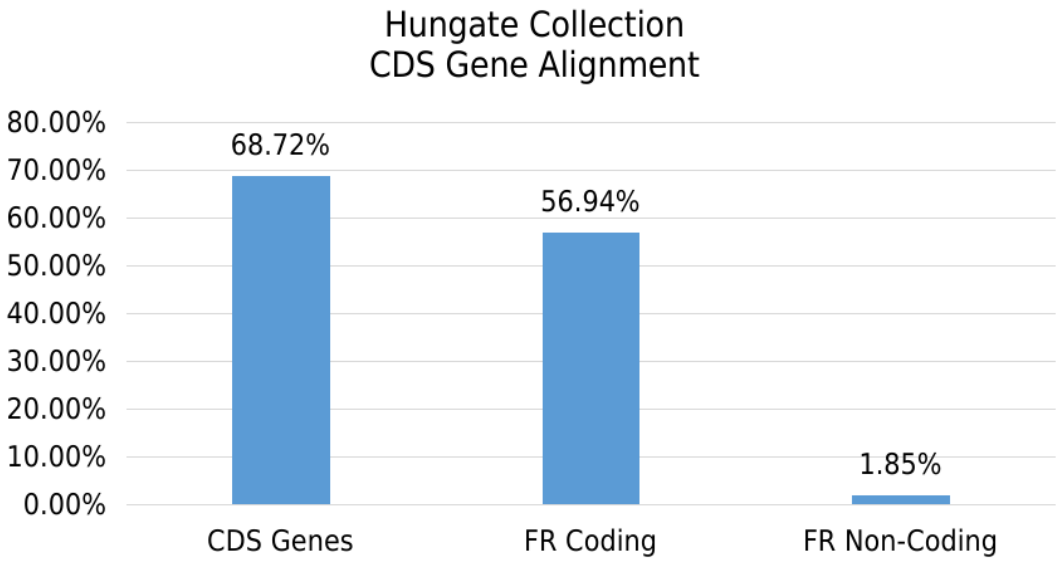


FIGURE 5.8: The differences observed between the proportions of metagenome assembly Prodigal CDS genes, FrameRate Coding and Non-Coding Frames, which aligned to the CDS genes from the Hungate Collection. Clearly shown is both the high level of correct Coding predictions (True Positives) and low level of incorrect Non-Coding predictions (False Negatives).

5.3.3 Functional Profiling

As observed in Chapters 3 and 4 sequence alignment tools such as DIAMOND are able to sufficiently align sequences of 75 amino acids or less to 'full-length' genes. This result is important as the functional analysis tool, eggNOG-mapper, uses DIAMOND to align target sequences to their ortholog sequences for annotation transfer.

5.3.3.1 Functional Profiling: FrameRate vs DIAMOND blastx

eggNOG-mapper was used to annotate the metagenomic CDS genes predicted by Prodigal and the different sets of FrameRate-predicted frames (see Subsection 5.2.5 for further detail). The results of the functional annotations revealed similar assignments to each of the functional categories, whether the data was pre-processed with FrameRate or whether the reads were submitted as DNA to be used with the DIAMOND blastx option which would try to find the best match from all 6 amino acid frames (see Table 5.5).

Type	Prodigal CDSs	20% blastx	20% FR Coding	20% FR Non-Coding
INFORMATION STORAGE & PRO'	265,030 **[21.59%]**	4,260,472 **[21.31%]**	3,353,754 **[20.74%]**	68,625 **[20.86%]**
CELLULAR PROCESSES & SIG'	297,488 **[24.24%]**	4,816,697 **[24.08%]**	4,004,139 **[24.76%]**	69,734 **[21.19%]**
METABOLISM	417,949 **[34.05%]**	7,590,829 **[37.96%]**	6,200,018 **[38.35%]**	119,229 **[36.23%]**
POORLY CHARACTERIZED	246,978 **[20.12%]**	3,328,638 **[16.645%]**	2,611,079 **[16.15%]**	71,463 **[21.72%]**
With COGs/Total Sequences	1,247,128/1,647,050 **[75.72%]**	18,880,192/26,445,973 **[71.39%]**	15,253,301/24,910,739 **[61.23%]**	310,613/12,978,414 **[2.39%]**

TABLE 5.5: The COG functional categories assigned to: (1) the Prodigal CDS genes predicted from the metagenome assembly, (2) the 20% subsample of reads which aligned to the Prodigal CDS genes using the DIAMOND blastx option, (3 and 4) the coding frames and non-coding frames classified by FrameRate from the same 20% subsample of reads which aligned to the Prodigal CDS genes. Chi-square tests between the Prodigal CDS genes and each set of subsampled reads all returned significant p-values of <0.00001. The chi-square test conducted on the blastx and FrameRate CFs blastp reads also reported a significant p-values of <0.00001. While all tests reported highly significant p-values, the large number of COGs assigned to each category make such results difficult to interpret.

Although the large number of COG assignments within each group make statistical analysis difficult to interpret, a chi squared analysis of the functional profiles reported that they were significantly different from each other even though they differed by no more than 4% in any category. In particular, this 4% difference was observed as a shift from 'POORLY CHARACTERIZED' to 'METABOLISM' in the FrameRate CFs results compared to the CDS genes. However, some of the differences may be driven by the nature of the sequences themselves. For instance, the Prodigal CDS genes were predicted from consensus assemblies of the sequencing data while blastx and FrameRate results were based on the reads, which may be closer to the true haplotypes in the community (Rubino et al., 2017). Furthermore, the differences observed between the FrameRate-classified reads and the blastx-classified reads could partly be explained by the use of blastp vs blastx respectively. While the blastx approach of eggNOG-mapper will only return an annotation for

the frame (1/6) which has the 'highest scoring alignment', in theory, FrameRate can return all putative coding frames for a specific read which then individually can be aligned by the blastp option. For example, if a read spans two different CDS genes, FrameRate can identify the coding signal from each independently and report both frames as coding for functional analysis. This is important when considering the number of CDS genes which overlap in different frames (see Chapter 2 for further details on overlapping genes), but which may not be reported correctly or to their full extent by Prodigal.

5.3.3.2 Functional Profiling: Shallow Sampling of Metagenomic Reads

The one million subsampled reads described in Subsection 5.2.4, were annotated by the same eggNOG-mapper tool as used in the previous Subsection 5.3.3.1. Shallow sampling of reads has been used before in metagenomic profiling and has shown to be a successful method of efficiently identifying the majority of the taxonomic profile of a sample (Hillmann et al., 2018). Table 5.6 reports an often used random subsample size of 10% of the reads from the entire metagenomic sample. This subsampling reports a very close EggNOG COG category functional profile to that of the FrameRate CF profile of the 20% subsampled reads which aligned to the metagenome CDS genes (see Table 5.5). Therefore, it could be inferred that not only are we observing the abovementioned functional difference between metagenomically assembled reads and raw read assignment, but also that only a 10% random subsample of an entire metagenomic DNA sample is required to produce a comparative functional profile.

Type	Prodigal CDS Genes	FrameRate 10% CF	FrameRate 10% NCF
INFORMATION STORAGE & PRO'	265,030 **[21.59%]**	1,274,473 **[20.89%]**	36,955 **[20.37%]**
CELLULAR PROCESSES & SIG'	297,488 **[24.24%]**	1,504,667 **[24.67%]**	37,333 **[20.58%]**
METABOLISM	417,949 **[34.05%]**	2,341,900 **[38.40%]**	70,199 **38.70%]**
POORLY CHARACTERIZED	246,978 **[20.12%]**	977,099 **[16.02%]**	36,904 **[20.35%]**
With COGs/Total Sequences	1,247,128/1,647,050 **[75.72%]**	5,751,883/10,085,622 **[57.03%]**	171,337/5,899,564 **[2.90%]**

TABLE 5.6: The EggNOG COG functional categories assigned to the Prodigal CDS genes predicted from the metagenome assembly and a random subsample of 10% of the reads from the entire metagenomic read dataset.

A much reduced subsample size of 1 million reads was also conducted. As can be seen in Table 5.7, this considerably shallow sampling has resulted in a comparable functional profile of COG groups when compared to the previous analysis presented in Subsection 5.3.3.1. Interestingly, there is no observable difference between the larger 10% subsample and the 1 million random subsample of the reads. It could have been assumed that such a low level of subsampling from this complex microbial environment could have led to taxonomic and therefore functional differences due to the overrepresentation of certain taxa and thus functions in the sample.

Nevertheless, these results indicate that this level of subsampling may be as efficient at functionally profiling complex metagenomic data as the broader approach undertaken in the previous Subsection.

Type	Prodigal CDS Genes	FrameRate 1m CF	FrameRate 1m NCF
INFORMATION STORAGE & PRO'	265,030 **[21.59%]**	114,766 **[20.92%]**	8,553 **[22.03%]**
CELLULAR PROCESSES & SIG'	297,488 **[24.24%]**	135,041 **[24.61%]**	8,781 **[22.61%]**
METABOLISM	417,949 **[34.05%]**	210,955 **[38.45%]**	13,258 **[34.14%]**
POORLY CHARACTERIZED	246,978 **[20.12%]**	87,892 **[16.02%]**	8,237 **[21.21%]**
With COGs/Total Sequences	1,247,128/1,647,050 **[75.72%]**	517,666/899,657 **[57.54%]**	36,644/640,707 **[5.72%]**

TABLE 5.7: The EggNOG COG functional categories assigned to the Prodigal CDS genes predicted from the metagenome assembly and a random subsample of one million reads from the complete set of metagenomic reads. The EggNOG COG functional assignments are similar to the 20% subsample of reads taken from the set of reads which aligned to the Prodigal predicted CDS genes. This suggests that this level of shallow sampling is sufficient for functional profiling.

5.3.3.3 Functional Profiling: Unassembled Reads

The assembly of metagenomic samples is a complex and yet unresolved task which often leads to many fragmented and poor quality assemblies (Ayling, Clark, and Leggett, 2020). Although in this analysis, 73.03% of the reads were assembled into the metagenomic assembly (see Subsection 5.3.2), there were still over 50 million reads that were not able to be assembled. The convention is often to ignore these reads and not undertake any further analysis of them. While this mentality has begun to change recently, with developments such as using additional rounds of assembly to utilise assembled contigs to help assemble these unassembled reads, there is still no universally implemented solution (Li et al., 2015; Wick et al., 2017; Hitch and Creevey, 2018). Nevertheless, genomes assembled from metagenomic data are still often of low quality and incomplete (Chen et al., 2020b). They also often contain high levels of contamination and a number of challenges exist which are yet to be overcome (Baptista and Kissinger, 2019). FrameRate allows for the analysis of the exact sequence, as it is in nature. As with the results of the assembled reads, when comparing to the COG function annotations from the Prodigal predicted CDSs, the unassembled reads from the metagenomic assembly reported slightly more 'METABOLISM' and less 'POORLY CHARACTERIZED' compared to the CDS genes (Table 5.8).

Interestingly, the COG functional profiles of each of the FrameRate subsamples, from assembled and non-assembled reads, are more similar to each other than they are to the Prodigal CDS genes in Table 5.5. While not clear, this may potentially indicate some level of bias or overpresentation either in the Prodigal or FrameRate methods and annotations. As mentioned above, there may be differences between

the community derived assembly and individual reads which are being detected by eggNOG-mapper.

Type	Prodigal CDS Genes	Unassembled FR CF	Unassembled FR NCF
INFORMATION STORAGE & PRO'	265,030 **[21.59%]**	4,554,088 **[21.09%]**	238,168 **[19.75%]**
CELLULAR PROCESSES & SIG'	297,488 **[24.24%]**	5,307,474 **[24.58%]**	245,849 **[20.39%]**
METABOLISM	417,949 **[34.05%]**	8,406,417 **[38.94%]**	502,412 **[41.66%]**
POORLY CHARACTERIZED	246,978 **[20.12%]**	3,321,544 **[15.39%]**	219,577 **[18.21%]**
With COGs/Total Sequences	1,247,128/1,647,050 **[75.72%]**	20,343,696/43,103,775 **[47.20%]**	1,138,511/32,848,511 **[3.47%]**

TABLE 5.8: The EggNOG COG functional categories assigned to the Prodigal CDS genes predicted from the metagenome assembly and the set of unassembled metagenomic reads classified by FrameRate (FR). CF and NCF stand for Coding Frames and Non-Coding Frames respectively.

Shallow sampling has been presented as a proven method in the above results. However, as can be seen in Table 5.9, the type of reads sampled can have a large impact on the results of FrameRate and eggNOG-mapper. It is important to remember that both the metagenome assembly and Prodigal CDS prediction methods involve a number of biases and errors. As such, a read which does not align to the CDS predictions is not inherently non-coding. Nevertheless, as clearly seen, there are substantial differences in the reported COG functional categories for the unassembled subsample compared to the CF predictions for the shallow sample (1 million and 10%) and the Prodigal-predicted CDS genes. Additionally, both the number of CFs reported by FrameRate from the 1 million reads randomly subsampled from the complete dataset compared to the 1 million reads sampled from the set of unassembled reads is drastically different at 899,657 (57.54% with COG functions) and 119,092 (17.71% with COG functions) respectively. What can be extracted from these results is that the unassembled reads are likely to be from organisms which are underrepresented in the training data for FrameRate and the EggNOG database. This underrepresentation may also explain the inability for MEGAHIT to assemble these reads as the organisms they belong to may contain specific differences in important elements such as k-mer makeup which is used by the assembly method.

Type	Prodigal CDS Genes	FrameRate 1m CF	FrameRate 1m NCF
INFORMATION STORAGE & PRO'	265,030 **[21.59%]**	7,038 **[32.01%]**	699 **[12.65%]**
CELLULAR PROCESSES & SIG'	297,488 **[24.24%]**	2,993 **[13.61%]**	91 **[1.65%]**
METABOLISM	417,949 **[34.05%]**	5,125 **[23.31%]**	480 **[8.68%]**
POORLY CHARACTERIZED	246,978 **[20.12%]**	6,828 **[31.06%]**	4,256 **[77.02%]**
With COGs/Total Sequences	1,247,128/1,647,050 **[75.72%]**	21,100/119,092 **[17.71%]**	5,360/157,404 **[3.41%]**

TABLE 5.9: The EggNOG COG functional categories assigned to the Prodigal CDS genes predicted from the metagenome assembly and a random subsample of one million reads from the set of metagenomic reads which did not align to the Prodigal CDS genes. The EggNOG COG functional assignments are clearly quite different at this very limited and non-aligned overview of shallow profiling.

5.3.4 Compute Time and Resources

One important factor in contemporary genomic analysis is the computational resources required. Another is whether the potential outcome of such analysis justifies the use of those resources. This is even more true with metagenomic analysis, which most often cannot be performed on local machines. As such, the time and computational resources needed to compute the different segments of this analysis are listed in Table 5.10.

Fundamentally, the computational and time requirements required to process the metagenomic assembly are in line with other similarly sized studies (Walt et al., 2017). However, while the metagenomic assembly clearly requires access to High Performance Computing (HPC) infrastructure, the FrameRate analysis on a 10% random subsample of all reads from the metagenomic sample can be performed on a consumer-grade laptop computer. The most time and resource intensive stage in the training or use of the FrameRate classifier is the initial conversion of the DNA sequencing reads to their respective one-hot encoded matrices for entry into the neural network (see Subsections 5.2.3.2 and 5.2.3.3 for further detail on how the amino acids are prepared for the model). This stage includes the DNA to amino acid conversion and stop codon filtering. Therefore, the largest proportion of the compute time and resources are reserved for this preprocessing step and the actual classification of the matrices (frames) only requires a few minutes to process many millions. However, while the FrameRate model is less than 1MB in size and only needs to use system memory during runtime, the MEGAHIT assembly requires substantial disk storage. This disk storage is needed for the computation of the intermediate assemblies of different k-mer sizes which are then computed and combined to produce the final assembly.

The 1 million and 10% random subsampled sets of reads constitute a more realistic entry point for using FrameRate on a novel meta/genomic sample. These reads were computed and classified by FrameRate in 4-6 minutes and 1-3 hours respectively (multiple runs for consensus), and the majority of this time was taken up by the IO (input/output) processing of the reads. The subsequent initialisation of the model and classification were the quickest processes. The classification in particular only took a matter of seconds for the entire 1 million set of reads. Lastly, the entire classification procedure was completed on a 4 core CPU on an 11 year old Linux desktop computer and utilised no more than 4.4GB of ram.

The identification of CDS genes or CFs/NCFs is only the first part of functional analysis, although an important one. Now that the coding sequences from both sides of the analysis have been created, the functional profiling can be undertaken with the eggNOG-mapper tool. As described in Subsection 5.2.5, substantial resources were required for this and they are presented in Table 5.11. While each of these eggNOG-mapper analyses could be computed on a consumer-grade laptop,

Analysis	Compute Time **[CPU Cores]**	Memory Requirements	Storage Requirements
MEGAHIT Assembly	46 hours **[32]**	~400GB	~100GB
Prodigal CDS Prediction	0.2 hours **[4]**	~2GB	N/A
FrameRate - Unassembled Reads	2-7 hours* **[4]**	~60GB-10GB *	1MB
FrameRate - 20% CDS Aligned Reads	1-4 hours* **[4]**	~60GB-10GB *	1MB
FrameRate - 10% Subsampled Reads	1-3 hours **[4]**	~50GB-8GB*	1MB
FrameRate - 1m Subsampled Reads	5.5 minutes **[4]**	~4.4GB	1MB

TABLE 5.10: The time and computation resource requirements are presented here for the analyses as follows: (1) MEGAHIT metagenomic assembly, (2) Prodigal CDS gene prediction and (3) FrameRate classification approaches. Column 'FrameRate 20%' reports the resources needed to run FrameRate on 20% of the reads which aligned to the Prodigal CDS genes. Where '*' is shown, the time and memory requirements are on a scale. For example, FrameRate can compute the unassembled reads in 1 hour while using 60GBs of memory or 7 hours using 10GBs of memory. The Storage Requirements listed here are the maximum disk space needed during the runtime of each method.

there were substantial differences in both their time and storage requirements for the blastx and blastp approaches.

EggNOG Function Analysis		Compute Time **[CPU Cores]**	Memory Requirements	Storage Requirements
Prodigal CDS Genes	**[blastp]**	4 hours **[4]**	~16GB	~8GB
20% CDS Aligned Reads	**[blastx]**	53 hours **[4]**	~16GB	~80G
FrameRate CF - 20% CDS Aligned Reads	**[blastp]**	6 hours **[4]**	~16GB	~40G
FrameRate CF - 10% Random Subsample Reads	**[blastp]**	4 hours **[4]**	~16GB	~35G
FrameRate CF - 1 million Random Subsample Reads	**[blastp]**	2 hours **[4]**	~8GB	~1GB

TABLE 5.11: The time and computation resource requirements are presented here for the eggNOG-mapper analyses separately for both the metagenomic assembly and FrameRate approaches. 'blastx' and 'blastp' relate to the options in the DIAMOND sequence alignment section of eggNOG-mapper. The listed Storage Requirements are those needed during runtime of each analysis and are released after completion.

CD-Hit clustering could be used to reduce the number of redundant amino acid coding reads. For example, clustering at 99% sequence and length similarity reduced the number of the 20% subsampled coding frames by 25% and non-coding by 20%. This output of the CD-Hit clustering was not used in this chapter.

5.4 Discussion

The original concept for this chapter was formulated during my visit to King Abdullah University of Science and Technology (KAUST) as a visiting member of the Bio-Ontology Research Group (https://cemse.kaust.edu.sa/borg). This team, which primarily worked on the intersection between machine learning and biology, often utilised a fusion of data such as the Gene Ontology (Ashburner et al., 2000), phenotype (Köhler et al., 2019), protein-protein interactions (Szklarczyk et al., 2019) and protein sequences in their methods (Kulmanov and Hoehndorf, 2020; Liu-Wei et al., 2021). However, even though many of their findings pointed towards sequence data being the most significant feature in their training, other data such as taxonomy and protein-protein interactions were the most developed. This is indicative of the wider machine learning research community where utilisation of different data sources is the norm, even when it results in more noise and marginal changes in accuracy in the resulting models (You et al., 2018; You, Huang, and Zhu, 2018).

The analysis from the previous chapters has all pointed towards two classes of genes that require more study (see Chapters 2, 3, 4), which can be summarised as the 'known unknowns' and 'unknown unknowns' (Logan, 2009). Here, the 'known unknowns' are the set or type of genes which have been identified previously but are still routinely missed by state-of-the-art genome annotation methods and as such are often missing from canonical annotations. The 'unknown unknowns', are those genes which we simply have not yet identified in any study reported in the literature. Both classes are bound to be under-represented in the databases. This, combined with the high level of identified genes with only 'hypothetical' or 'function unknown' annotations, even in model organisms, only adds to the challenge of using machine learning approaches to functionally annotate unknown sequences. Therefore, in this chapter I have specifically avoided trying to predict function from sequence data but instead investigated the potential of machine learning to answer one specific question in genomics: is any given DNA sequence protein coding?

5.4.1 Machine Learning can Detect Coding Potential Through Patterns in Protein Sequences

Various complex statistical and machine learning algorithms have been widely and successfully utilised in biology for over two decades (Yang et al., 2020) such as Clustal (Sievers and Higgins, 2018), BLAST (Altschul et al., 1990), and AlphaFold (Jumper et al., 2021). From the very start, neural networks have been utilised for their innate ability to quickly identify representative patterns from seemingly incomprehensible data, either due to its size or complexity. It therefore seems natural that a neural network should be useful when applied to large collections of amino acid sequence data.

The scale and diversity of all 'possible' protein sequences is vast (Alberts et al., 2002). Of the more than 20^{300} (typical protein length $\tilde{3}00$ amino acids) different polypeptide chains that could theoretically be made, the true number that could viably be made and expressed *in vivo* is still currently undetermined. However, important and 'somewhat curated' databases such as SwissProt contain only marginally more than 500,000 proteins and only 10% of these proteins are listed with SwissProt's highest 'annotation score' as of 04/2021 (see https://www.uniprot.org/help/annotation_score for further detail). On the other hand, this may not itself be as big a problem as first thought as it has more recently been postulated that the full spectrum of viable functional protein sequences is likely to exist in current lifeforms (Dryden, Thomson, and White, 2008). Nonetheless, while the true search space of functional amino acid sequences and thus signatures, is possibly a lot smaller than we think, the current databases will continue to become less useful as we continue to sequence and assemble more data from novel environments and organisms.

As discussed in the Introduction, there are patterns in both the DNA and amino acid sequence of a protein coding regions of a genome (Staden, 1984; Gribskov, Devereux, and Burgess, 1984; Fickett, 1982). For example, the non-optimum selection of the initial codons within a CDS gene is crucial in the regulation of expression (Eyre-Walker and Bulmer, 1993) and subsequent studies have suggested this type of selection is also present in the amino acid sequence (Bentele et al., 2013). The identification and exploitation of these patterns or rules have led to a number of advances in computational techniques such as the AlphaFold platform which uses rules inherent to 'all' protein sequences to determine structure (Jumper et al., 2021). Similarly we found that sequence padding and max pooling, which is still often overlooked in the literature (Rio et al., 2020), was the parameters which had teh largest effect on overall accuracy observed in the results of training the model on different combinations of sequence padding and max pooling sizes (see Figure 5.4). Specifically we found that when the max pooling size is larger, the Repeat-Sequence-Padding performed better because there was more data to analyse. However, when the max pooling size was smaller, the Repeat-Sequence-Padding accuracy only increased marginally and in fact dropped by over 12% for the sequences less than or equal to 50 amino acids. These results may in part be explained because this approach pads the 3' end of the sequence with amino acids from the 5' end, it may lead to noise for the model. These out of place padded sequences may never present in nature (or at least in the Ensembl data) and as such, the model is uncertain of how to classify them. Additionally, the impact that we observed of narrower windows defined by smaller max pooling sizes reflects what has already has been reported in the literature up to a point. For example, from as far back as 1969, one study suggested that "... protein folding is guided by the rapid formation of local interactions which then determine the further folding of the polypeptide." (Levinthal, 1969). What this suggests is that it is at the local level, or small groups of amino acids, that the validity of a protein sequence is decided.

There is clearly more to learn about what makes an amino acid sequence protein coding or not. Additionally, while many of the processes behind species-specific codon and amino acid selection have been long known (Ikemura, 1981; Lithwick and Margalit, 2003; Dumontier, Michalickova, and Hogue, 2002), we still do not know the underlying rules which must be followed for a protein sequence to be both viably expressed and folded (Dill et al., 2008). During the early stages of the development of FrameRate, the model was trained on a small set of CDS sequences (their amino acid frames) from a selection of *E. coli* genomes. The trained models which resulted were not as accurate as the final version of FrameRate (on their own testing data) but also more interestingly, their accuracy was very poor when tested on amino acid sequences from other species. This poor performance was also found when uses larger training set of genes from 10 different genera, leading us to utilise the dataset of 179 genomes from 179 genera as presented earlier. As noted, neural networks are difficult, if not impossible to interpret, and the exact reasons for these results may never truly be known without further investigation of the specific sequences which was incorrectly classified. Additionally, although the number of FrameRate classified Non-Coding Frames which were reported with an EggNOG functional annotation was low in each study, future analysis of their sequence could help both understand the processes of FrameRate and the underlying biology. Lastly, these results suggest that with a data-centric approach, where the biology is kept at the forefront of development, machine learning algorithms will continue to make important progress in the field of genomics.

5.4.2 Profiling Metagenomic Samples Without Assembly

Every time data is processed, it is possible that the results incurs a level of bias and loss inherent to the method used. The preprocessing of genomic read data ([meta]genome assembly and annotation) involves a number of filtering steps and the resulting output is often an consensus of the original reads (see Subsections 5.2.1 and 5.2.2 for metagenome preprocessing). Therefore, it is important that we continue to develop methodologies which can characterise genomic data in its rawest and most natural form. Although the validation of FrameRate which has been conducted in this chapter was undertaken on a set of pair-ended reads (for the comparison to the MAG), FrameRate can be utilised on any DNA sequence whether it be paired, unpaired, assembled or unassembled. Furthermore, unlike [meta]genome assembly, the preprocessing conducted with FrameRate is itself non-destructive and all reads and frames can be classified as they are in nature.

As with all genomic studies, it is important to identify and account for the fact that at each level of annotation, there are different biases imposed from each data source and method utilised which are bypassed by Frame rate as it processes the raw reads directly. However, the functional annotation of the resulting protein

sequences brings with it both benefits and challenges. As reported in Results Subsection 5.3.3.1, FrameRate processed blastp and blastx raw reads reported similar assignments (less than 1% difference) to each of the functional categories (see Table 5.5). However, both read-based approaches were observed with the same ~4-5% shift from the COG category 'POORLY CHARACTERIZED' to 'METABOLISM' compared to the CDS genes predicted from the MAG. This shift was observed in each analysis between FrameRate classified CFs and the CDS genes, including from the 10% shallow sampling study (but not for the unrealistic 1 million shallow sampling - see Subsection 5.3.3.2). As discussed above, these differences may be driven by the nature of the sequences themselves. The consensus sequence produced by the metagenomic assembly, combined with the biases imposed by prodigal CDS genes (see Chapter 2 (Dimonaco et al., 2021)) are both possible reasons behind this shift and has been recognised by the community through the use of the term 'metagenomic strain' (Anyansi et al., 2020) to define the set of strains from an environmental samples (Quince et al., 2017; Quince et al., 2021). The strains being lost due to the consensus may contain slight differences which are being picked up by the FrameRate approach (Rubino et al., 2017). Furthermore, the less than 1% difference observed in each COG functional category between the blastp and blastx approaches may have been due to bias from the blastx 'best hit frame' and FrameRate trained on Ensembl. For example, whereas FrameRate is able to report each frame of two genes positioned over a single read, thus allowing for the reporting of the function from both, blastx will only return a single frame. Even though evidence exists that genes positioned close together (also those overlapping) have been frequently observed to contain similar functions, it is not always the case (Mihelčić, Šmuc, and Supek, 2019). Nevertheless, as the results and the apparent biases of the methods used have been consistent throughout this chapter, profiling metagenomic samples ustilisng FrameRate and eggNOG-mapper is as good as the community standards for this type of method.

There is more than one way to characterise a metagenome. The validation studies of FrameRate reported by the two CDS gene alignment studies conducted in Subsections 5.3.2.1.1 and 5.3.2.1.2 produced arguably the most striking results of the chapter with 79.84% of FrameRate CFs reporting an alignment to the metagenome CDS genes compared to only 3.43% of the NCFs (see Figure 5.7). This disparity was also observed in Figure 5.8 which reported 68.72%, 56.94% and only 1.85% of the metagenome CDS genes, FrameRate CFs and FrameRate NCFs aligned to the Hungate Collection CDS genes. While noting that both the metagenome and Hungate Collection CDSs contain their own limitations, both of these alignment studies not only showcase the high level of FrameRate CFs which have correctly been classified, but also report that its NCFs are significantly less likely to report an alignment to known CDS genes. These results could be used to help filter or align DNA reads without any homology to known genes, reducing the reliance on assembled genomes for identifying coding sequences.

5.4.3 FrameRate Reduces the Resources Required for Metagenomic Profiling

Assembly and annotation techniques are not keeping up with the growing number and size of metagenomic studies being conducted (Lapidus and Korobeynikov, 2021). As a direct consequence of the vast amount of time and resources required to carry out the analysis, many years can pass by between environmental sampling and the publication of the resulting data and analysis (Haroon et al., 2016; Sunagawa et al., 2020). This is detrimental to science as not only are we required to wait for this data, but also by the time of publication, the data and techniques used may no longer be as relevant and therefore useful as they were at the inception of the study.

Machine learning at its core promises a number of benefits for genomics, with their relatively small time and computational resource requirements being considered as major factors (Jordan and Mitchell, 2015; Sarker, 2021). As seen in Table 5.10 from Subsection 5.3.4, the computation of the MEGAHIT MAGs was not only impossible on a personal computer, but also required nearly 2 days on a HPC with 32 CPU cores and around 400GB of RAM. However, FrameRate was able to report a similar functional profile to the metagenomic CDS genes in only a fraction of the time (also note - 4 vs 32 CPU cores required for FrameRate vs MEGAHIT respectively) and computational resources (see Table 5.5). Furthermore, unlike many other computational methods such as [meta]genome assembly and sequence alignment, the vast majority of time and computation resources are only required during the initial training of a neural network, and not the application of it. As FrameRate reported the same functional profile when processing the 10% shallow sampled reads as reported in Table 5.6 in Subsection 5.3.3.2, it could be proposed that metagenomic assembly and gene annotation could be bypassed entirely. Fianlly, the results of Table 5.11 which reports the computational requirements for the different approaches to function annotation, reported that the blastx option in eggNOG-mapper requires nearly 9 times the time and twice the storage capacity of FrameRate blastp on the same set of reads.

5.4.4 Limitations and Future Work

There are several ways in which we could improve FrameRate. These changes would include, but are not limited to: (1) training on genus-level pangenomes, (2) labeling the edges of overlapping CDS genes so that the model could learn how to account for reads with more than one coding signal, (3) ncluding parts of the non-coding 5' and 3' untranslated regions of each CDS gene and other 'true' non-coding DNA from the same set of genomes to allow for better representation of 'real-world' sequencing data, (4) using alternative codon to amino acid translation tables for both learning and classifying, (5) investigating the internal stop codon filtering to report whether it is too strict and is possibly filtering out too many frames and especially CFs, (6) producing alternative versions of the model to allow the classifications of

sequence data longer than 75 amino acids. Nonetheless, it can be assumed from the results of this chapter that the (non-exhaustive) listed possible improvements and thus problems yet to be resolved, have had little effect on the overall performance of the FrameRate model.

There are also some challenges presented in the results for which the answer is yet unknown. One of these challenges, which itself transcends across genomics, is the phenomenon that proteins of the same function can exhibit as low as 25% sequence similarity at the amino acid level which has been named the 'Twilight Zone' of sequence function (Chung and Subbiah, 1996). However, in recent studies, it was found that "the sequence similarity between two proteins must be quite high for them by have high function similarity" (Higdon, Louie, and Kolker, 2010) and there may be sequence similarity cutoffs which can be used to distinguish functional divergence (Sangar et al., 2007). For example, one study which examined the homology between genes from yeast and humans reported aligned genes with less than 20% identity had BLAST (Altschul et al., 1990) bit scores ranging from 55 – 170 bits which is higher than the minimum bitscore required for eggNOG-mapper (60 bit) to return a functional annotation (Cantalapiedra et al., 2021). These findings, although incomplete, do suggest that it may be impossible to create training and testing data for FrameRate which does not contain similar sequence signatures in both. Additionally, while the sequence window step (of 50 amino acids) utilised should help mitigate any negative impact (see Subsection 5.2.3.2), but may facilitate the domain level signature transfer to the model and not the signature of the entire protein sequence. Therefore, as domains are more likely to be shared between more sparsely related protein sequences than non-functional segments, the true separation of sequence and function and its impact on the results presented in this chapter and on the training of FrameRate is yet another 'unknown unknown'.

We often refer to machine learning algorithms as artificial intelligence. However, in complete opposition to that definition, these methods can not (accurately) make decisions on data they have yet to see. As such the seemingly excellent results of Subsection 5.3.2.1 may hide a bias because the Ensembl database, which was used to train FrameRate, shares some level of similarity with the EggNOG database and the dataset which Prodigal was trained on. A number of NCFs may be coding but share no similarity to known proteins and therefore the FrameRate model is enriching for protein sequences with known function. It is also possible that some of the CDS gene and FrameRate sequences with EggNOG COGs annotated as 'POORLY CHARACTERIZED' are not actually functional genes. Finally, the EggNOG COG function predictions are likely to be effected by the same type of bias influenced by the crossover between sequence and function.

The current form of FrameRate presents a number of exciting future possibilities and avenues for further uses of the classifier. Some examples include: (1) the ability to quickly identify the level of coding potential from a set of unassembled

reads (or any DNA sequences), (2) to offer insight into the coding potential of the putative sequences, such as StORFs, without any Shine-Delgano or other expected motifs (Omotajo et al., 2015), (3) investigate eukaryote and prokaryote reads from the same environmental samples to determine whether there are different amino acid signatures which could be utilised for their separation in environmental datasets, (4) to study different segments of predicted genes in both prokaryotes and eukaryotes to investigate whether it is possible to label those segments which are likely to be introns, or even entire genes which may not be coding, such as rhibosomal genes, (5) to use the classified amino acid sequences to produce protein level assemblies.

Fundamentally, none of the 6 reported challenges require significant reworking of the FrameRate model and through limited changes to the core mechanics of the model, variations of FrameRate could be developed to investigate the 4 future uses mentioned above.

5.4.5 Conclusion

The ability to investigate genomic data at the community level was one of the major promises of metagenomics (Thomas, Gilbert, and Meyer, 2012). However, what has been gained by the ability to study environments at the community level, has resulted in the loss of resolution of the individual genome. Through the use of FrameRate, we are able to quickly characterise practically every sequence read, whether it assembles or not. Additionally, whereas chimeric sequences are a major problem for metagenomically assembled genomes (Haas et al., 2011; Arroyo Mühr et al., 2020), with FrameRate, we can investigate the sequence as it is in nature.

Overall there is still for machine learning to shape the future of bioinformatics. While a number of conclusions can be drawn from this work, I am specifically surprised by the success of such a simple model and variety of genomic assumptions during the development and training of the FrameRate model (see Subsection 5.2.3). Even though we have identified a large list of problematic assumptions, there are likely more to be discovered (see Subsection 5.4.4). As such, similarly to the investigation of the different parameters of the model undertaken in Subsection 5.4, it is clear that further investigation of the model is required.Although the additional work needed to formalise FrameRate and the classification procedure itself will take time, I believe that this approach is a valid and useful addition for the characterisation of large metagenomic datasets without the need for HPCs. In particular, FrameRate may be best utilised on subsampled data, such as conducted in Subsection 5.3.3.2, to produce an almost immediate and comparable profile of metagenomic sequencing data.

Chapter 6

Computational Analysis of SARS-CoV-2 and SARS-Like Coronavirus Diversity in Human, Bat and Pangolin Populations

Chapter Summary

In 2019, a novel coronavirus, SARS-CoV-2/nCoV-19, emerged in Wuhan, China, and has been identified as the causative pathogen responsible for the ongoing 2021 COVID-19 pandemic. The evolutionary origins of the virus remains a topic of debate (Wu et al., 2021; Holmes et al., 2021) and understanding its complex mutational signatures could help guide vaccine design and development. As part of the international "CoronaHack" in April 2020, a collection of contemporary methodologies were employed to compare the genomic sequences of coronaviruses isolated from human (SARS-CoV-2;n=163), bat (bat-CoV;n=215) and pangolin (pangolin-CoV;n=7).

While there has been recent progress of virus-specific genome annotation methods (Shean et al., 2019), they still rely heavily on homology to reference genomes and RNA expression data. As such, the nuance of the SARS-CoV-2 genome advocated the use of StORF-Reporter (see Chapter 3) which without homology, provided additional ORF predictions to the canonical genome annotation. These additional annotations subsequently were observed with homology to known proteins in other SARS-like viruses. As a result of this, a hybrid genome annotation approach, combining *de novo* gene prediction with homology searching, was employed to produce comparative annotations across the different sets of virus genomes. This approach allowed for analyses including a gene-gene similarity network, codon usage preference and variant discovery. Strong host-associated divergences were noted in ORF3a, ORF6, ORF7a, ORF8 and S, and in codon usage preference profiles. Lastly, several high impact variants have been characterised (inframe insertion/deletion or stop gain) in bat-CoV and pangolin-CoV populations, some of which are found in the same amino acid position and may be highlighting loci of potential functional relevance.

This work is now published in the journal Viruses (Dimonaco, Salavati, and Shih, 2021). **Software Availability:** https://github.com/coronahack2020/final_paper

6.1 Introduction

The continued and increasing occurrence of pandemics that threaten worldwide public health due to human activity is often considered to be inevitable (Patz et al., 2000; Madhav et al., 2017). The COVID-19 (2019-current) pandemic caused by the emergence in Hubei, China, of what has now been identified as Severe Acute Respiratory Syndrome Coronavirus 2/Novel Coronavirus 2019 (SARS-CoV-2/2019-nCoV) by The Coronaviridae Study Group (International et al., 2020), has brought a number of questions regarding its transmission, containment and treatment to the urgent attention of researchers and clinicians. It is acknowledged in these studies that while these questions will likely ultimately be answered, we are in need of answers now (Piplani et al., 2020). The urgency of such questions has spurred a number of atypical approaches and collaborations between experts of different fields and as such, this study was carried out as part of a "CoronaHack" hackathon event in April 2020 where my team gained access to genomes and related metadata available at the time (Dec 2019 - April 2020).

Viruses of the Coronaviridae family have long been studied and while there have been great advances in our understanding, each new emergence has brought about its own questions. The sub-family Coronavirus consists of four genera, Alphacoronavirus (Alpha-CoV), Betacoronavirus (Beta-CoV), Gammacoronavirus and DeltaCoronavirus. Coronaviruses are a family of single-stranded, enveloped and extremely diverse RNA viruses which are known to have come into contact with humans numerous times over the past few decades alone (Weiss, 2020). At around 30kb, they exhibit at least six Open Reading Frames (ORFs), ORF1a/b comprising of approximately 2/3 of the genome which encodes up to 16 non-structural replicase proteins through ribosomal frame-shifting, and four structural proteins: membrane (M), nucleocapsid (N), envelope (E) and spike (S) glycoprotein (Perlman and Netland, 2009). Coronaviruses have developed a number of different strategies to infiltrate their host-cells. In human-associated CoVs, it has been shown that different parts of the human Angiotensin Converting Enzyme 2 (hACE2) can be bound to by their respective S proteins. Previous examples of human-infecting Coronaviruses such as SARS-CoV-1 (Severe Acute Respiratory Syndrome Coronavirus) and MERS-CoV (Middle East Respiratory Syndrome Coronavirus) have shown Coronaviruses to be capable of presumed efficient adaptation to their human host and exhibit high levels of pathogenicity (Amer et al., 2018; Hung, 2003). Interestingly, SARS-CoV-1 and MERS, which along with SARS-CoV-2 are both Beta-CoVs, exhibit only 79.5% and 50% sequence similarity at the whole genome level to SARS-CoV-2, whereas SARS-CoV-2-like coronaviruses found in pangolins (pangolin-CoVs) and bat coronavirus (bat-CoV) SARSr-Ra-BatCoV-RaTG13 (RaTG13) are 91.02% and 96% respectively (Zhu et al., 2020). The relationship of SARS-CoV-2 to other SARS-like coronaviruses, the possible role of bats and pangolins as reservoir species and the role of recombination in its emergence, are clearly of great interest and importance (Boni

et al., 2020). Speculations around other intermediary hosts are also at play, which might have affected the ability for zoonotic transmission for SARS-CoV-2 to its human host (Zhang and Holmes, 2020). Crucially, this evolutionary relationship between SARS-CoV-2 and its lineage may prove to be an important factor in the eventual management or containment of the virus. Moreover, the mutation events along the evolutionary timeline of SARS-CoV-2 are of importance in the discovery of possible adaption signatures within the viral population. At the time of the hackathon, there were two main suspected SARS-like reservoir host-species; bat and pangolin (named bat-CoV and pangolin-CoV).

At the stage of the pandemic this work was carried out, it was acknowledged that it would be unlikely to acquire the necessary information needed to adequately pinpoint the exact origin of the SARS-CoV-2 virus. With this in mind, this study aimed to systematically compare a broad selection of contemporary available SARS-CoV-2, bat-CoV and pangolin-CoV at genome, gene, codon usage and variant levels, without preference for strains or sub-genera. This was comprised of 46 SARS-CoV-2 genomes isolated early in the pandemic from Wuhan, China (Late 2019-Early 2020), 117 SARS-CoV-2 genomes isolated in Germany, representing the later stage of global transmission, 215 bat-CoV genomes of Alpha-CoVs and Beta-CoVs and 7 pangolin-CoV genomes, of which 5 were annotated as Beta-CoVs. Furthermore, during the hackathon, it was recognised that potential biases can arise from directly comparing SARS-CoV-2 to a wide repertoire of coronaviruses at varying stages and quality of genome annotation. In any genomic study, the vast majority of genes or other functional genetic elements predicted are unlikely to ever be experimentally characterised, and even with the international concerted effort in response to SARS-CoV-2, we are unlikely to acquire this knowledge during the current stages of the pandemic. As reported in Chapter 2, it is of paramount importance that the limitations of gene prediction methodologies are understood and accounted for, especially as our reliance on them is likely to only increase during future public health emergencies. Furthermore, an exploratory StORF-Reporter analysis of unannotated regions (URs) of the SARS-CoV-2 genome (according to Ensembl consensus) presented a number of additional Open Reading Frames (ORFs) with homology to known proteins in other SARS-like viruses. These ORFs were in addition to the canonical but fluid genome annotations which the international community were continuing to refine. However, in acknowledgment of the potential differences between annotation methodologies, the inherent time constraints of the CoronaHack, the varying expertise each member of the team brought to the project, and that at the time of this analysis, the SARS-CoV-2 reference was still under-review (for example, the coordinates and expression evidence of ORF10 continue to be under debate (Koyama, Platt, and Parida, 2020; Kim et al., 2020)), we sought to carry out an exploratory analysis using systematic and strain agnostic methods that minimise the use of reference genome or annotation where possible.

Through examination of the inherent sequence diversity between this comprehensive collection of SARS-CoV-2, bat-CoV and pangolin-CoV, this study's aim was to highlight naturally occurring high impact genomic variations that can potentially introduce a change in the resulting protein, such as the insertion or deletion of an amino acid or early termination of the sequence. To further validate mutational adaptations which may have facilitated the zoonotic transmission of SARS-CoV-2, a codon usage analysis was carried out between the SARS-CoV-2 reference genes and the genes identified using the aforementioned approaches. In addition, codon usage preference was profiled across the data set, as in the process of host adaptation, viruses can evolve to express different preferential codon usages (Jitobaom et al., 2020; Kumar et al., 2018; Chen et al., 2020a). Understanding the stability and variability of these positions may potentially aid future design of vaccines or treatments and identify potential avenues for future zoonotic transfer. For instance, an amino acid position where insertion or deletion is commonly found in a coronavirus affecting other species may indicate that its alteration does not have a dramatic impact on the overall protein folding, or that the position is important for transmission to a new host (Forni et al., 2021). Furthermore, this work is differentiated through the systematic approach used to process this broad selection of viral genomes from public repositories. This application of a wide range of contemporary methodologies allowed for an unbiased overview of contemporary Coronavirus genomic data. Understanding the current limitations of annotation pipelines and the available quality of curated SARS-CoV-2 genomes was the main driver of the approach undertaken in this work. Additionally, providing a comprehensive gene and variant annotation for viral genomes collected from multiple hosts will further bridge knowledge gaps in the literature.

In previous pandemics such as SARS-CoV-1 (SARS), many lessons were learned as regards to potentially important studies. Indeed, the majority of articles on SARS were submitted after the World Health Organisation had declared the pandemic over (Xing et al., 2010). These studies contained data that would have been relevant to public health authorities during the pandemic. Contemporary, established and standardised protocols for analysis are vital to the prompt publication and therefore usefulness of such studies. This is why validated methodologies were chosen for the development of this study and the results from StORF-Reporter were not included in the published version of this chapter. However, for completeness, they have been included here.

6.2 Methods

This chapter was primarily carried out as part of a team effort in the 2020 Coronahack and subsequent publication. As first author, the majority of the analysis and reporting was performed by myself with some guidance from the other two authors, Barbara B. Shih and Mazdak Salavati. The 'Gene Relationship Network Graph' section was undertaken by Barbara B. Shih, while the 'Coding Usage' and 'Variant Analysis' sections were performed by Mazdak Salavati.

6.2.1 Genomes

Historically, genomes held in public databases have been fragmentary, resulting in multiple collections with overlapping examples with alternative naming schemes and annotations (Lathe et al., 2008). Fortunately, a large collection of virus genomes of the Coronaviridae family (Coronavirus) deposited in databases such as the Virus Pathogen Resource (ViPR) (Pickett et al., 2012) and NCBI (Coordinators, 2018) have been provided with both genomic sequence and metadata which has been examined for redundancy and comparative annotation. Coronavirus genomes isolated from humans, bats and pangolins used in this study were collected from multiple repositories and grouped by their host and source. The databases and groups are listed in Table 6.1.

Host–Source	No. Genomes	Database
SARS-CoV-2 Wuhan isolates	20	DataBiology
SARS-CoV-2 Wuhan isolates	26	GISAID-Charite (Elbe et al., 2017)
SARS-CoV-2 German isolates	117	GISAID-Charite (Elbe et al., 2017)
SARS-CoV-2 Ensembl Wuhan Ref	1	Ensembl (Yates et al., 2020)
Bat	139	DataBiology
Bat	76	ViPR (Pickett et al., 2012)
Pangolin	5	DataBiology
Pangolin	2	NCBI (Coordinators, 2018)

TABLE 6.1: Coronavirus genomes were collected from the various database resources listed by host and source categories. Using taxonomic data made available by the Virus Pathogen Database and Analysis Resource (ViPR) (Pickett et al., 2012), 70 bat-CoVs were identified as *Betacoronavirus* and 84 were *Alphacoronavirus*. 5 pangolin-CoVs were identified as *Betacoronavirus*. The remaining bat-CoV and pangolin-CoV genomes did not have a family identification. All genomes were downloaded in May 2020 and consisted of the contemporary available and open datasets at the time. The NCBI listed genomes and their respective ID's are currently available through NCBI (Oct 2020). In cases where two groups contained the same genome (Possibly with a different name), only one representative was taken.

6.2.2 Genome Annotation

RNA viruses such as SARS-1 and other coronaviruses have been characterised as having the ability to utilise programmed ribosomal frameshifting for a number of important genes (Dinman, 2010). Identification of such genes is complex and often requires high quality RNA expression evidence. Due to this and the complexity of genome annotation, especially in novel viral genomes such as SARS-CoV-2, a number of approaches were examined and undertaken to identify the set of genes for each of the genomes in this study.

In this regard, for defining genes, we first employed PROKKA (Rapid Prokaryotic Genome Annotation) (Seemann, 2014) to curate the genes for each of the coronavirus genomes. PROKKA utilises Prodigal (Hyatt et al., 2010) to initially find ORFs with protein coding potential. Prodigal is an unsupervised *ab initio* prediction tool which while has been trained on previous examples of prokaryotic genes, does not rely on previous knowledge of the input genome to predict ORFs. This is unlike sequence homology based tools such as BLAST (Altschul et al., 1990), which require highly similar and previously annotated sequence data to identify potential genes. However, as with all software, the predictions it makes are defined by a set of rules or parameters which are influenced from previous studies of specific genomes (see Chapter 2 for more detail). After the initial ORF prediction, PROKKA then employs additional steps to identify non-coding genic regions with tools such as Infernal (Kolbe and Eddy, 2011) and functionally annotate identified genes with a custom BLAST search.

Irrespective of the competence of the PROKKA platform, a number of URs were present across both the Wuhan and Charite genome annotations it produced. Due to this and the nuance of the SARS-CoV-2 genome, UR-Extractor and StORF-Reporter (see Chapter 2) was run on these URs and detected a number StORFs of gene-like lengths. These findings, including the capture of the elusive ORF10 protein by a StORF (see Figure 6.2, demonstrated the need for additional methods for genome annotation. Therefore, to overcome the aforementioned limitations and intricacies of contemporary *ab initio* genome annotation techniques, BLAST was used to identify additional genes with strong homology to those present in the early SARS-CoV-2 reference genome released by Ensembl v100 (SARS-CoV-2 ref) *ASM985889v3* (Yates et al., 2020) (`https://covid-19.ensembl.org`). The additional BLAST annotation was performed with a BLAST percentage identity threshold of $\geq$ 80% and are labelled separately where annotation methodologies may have an impact. This combined approach was used to avoid solely relying on any single method, especially BLAST's agnostic approach to coding frame detection.

While the results from StORF-Reporter were fundamental to the eventual design of the hybrid annotation approach employed, due to the cross-host-species comparative approach of this study, they were not used in further analysis.

6.2.3 Phylogenetic Trees

A Phylogenetic tree was produced from the genomes of the SARS-CoV-2 Wuhan isolates, Ensembl Wuhan reference and the bat-CoV and pangolin-CoV genomes to examine their evolutionary relationships at the genomic level. Clustal Omega 1.2.4 (Sievers and Higgins, 2018) was used to perform a multiple sequence alignment for each of the genomes with default parameters. The phylogenetic tree was inferred from the multiple sequence alignment with RAxML (Stamatakis, 2014) using default parameters apart from the GTRGAMMA option and bootstrapping set to 20. The tree was plotted using the following packages in R: Midpoint-root and ladderisation were carried out using phytools(Revell, 2012) and ape (Paradis and Schliep, 2019), and ggtree (Yu, 2020) was used for the visualisation. The subgenus information for Betacoronavirus were curated and clades labelled based on consensus of the majority (i.e. if > 85% of the samples in the clade are labelled and have the same subgenus annotation). For labelling the bat-CoVs host genera and species information, a list of host genera and species was curated. Host species with $>= 10$ bat-CoV genomes were labelled, followed by host genera with more $>= 10$ bat-CoV genomes. The remaining bat-CoVs were grouped into a single group "other".

6.2.4 Gene Relationship Network Graph

Genes identified by PROKKA from each host-set were collated and together with the additional sequences from the BLAST-alignment to the SARS-CoV-2 ref genome as aforementioned, an all-against-all search was performed with BLAST. This was done with all gene sequences as both the reference and the query as input. A network graph was generated using Graphia Enterprise (Freeman et al., 2020) by treating each gene as a node and generating edges between nodes with significant BLAST alignments. A significant BLAST alignment was defined to have a bit-score ≥ 60, a query coverage $\geq 80\%$ and a percentage identity $\geq 80\%$. Components with less than 5 nodes were removed from the graph. The same procedure was carried out using amino acid sequences as input (Figure 6.4). Where the amino acid sequences were not generated by PROKKA, the matched sequences extracted from BLAST were translated into amino acid sequences, provided that the sequences contained inframe start and stop codons.

6.2.5 Codon Usage

Codon usage metrics for every gene in the SARS-CoV-2 reference gene catalogue were calculated in all available genome sets. Gene sequences identified by PROKKA and BLAST searches against the SARS-CoV-2 ref genes; genes that have a BLAST result were included and annotated with the SARS-CoV-2 gene name (where correct frame was present) were collated. For each set of genes annotated with a SARS-CoV-2 gene, those substantially shorter than the average (< mean length - 2 standard deviation) were removed from codon usage analysis. For ORF6 and ORF8,

the BLAST filter criteria yielded few bat-CoV (11 and 3) or pangolin-CoV (1 and 6) genes. Therefore, in addition to the BLAST selected genes, bat-CoV and pangolin-CoV genes labelled as ORF8 and ORF6 in the network analysis (Figure 6.3) were incorporated in the codon usage analysis. For pangolin-MP789, the PROKKA output from an additional assembly (MT121216.1) was included in the codon usage analysis. As part of the team, custom Python scripts (available at Github (https://github.com/coronahack2020/final_paper.git) were developed to summarise the frequencies of each of the codons for each gene. Non-standard codons, start, stop codons were discarded, along with the codon TGG as it is the only codon coding for tryptophan. Principle Component analysis (PCA) was performed on the Relative Synonymous Codon Usage (RSCU) values, and kmean clustering was used on the first 10 PCs to group the genomes into 3 clusters. RSCU was calculated as the ratio of the observed frequency of codon to the expected frequency under the assumption of equal usage between synonymous codons for the same amino acids (Sharp, Tuohy, and Mosurski, 1986).

6.2.6 Variant Analysis

For this analysis, the teams aim was to highlight naturally occurring and population-wide viable variants, defined as being different to the SARS-CoV-2 ref and have an impact on coding potential. Variant calling was carried out for all available genome sets against the reference SARS-CoV-2 genome released by Ensembl v100 *ASM985889v3*. The allelic counts and variant effect prediction was carried out in order to identify variants with high impact changes (in-frame deletion, in-frame insertion, frameshift, or stop gain) within or between viruses collected from different host-species.

Briefly, multiple genome fasta input files were mapped against the SARS-CoV-2 ref assembly using minimap2 (Li, 2018) with the following flags (minimap2 –cs -cx asm20 INPUT REF > OUT.paf). The generated PAF (pairwise alignment format) files were subsequently used for variant calling through the paftools.js module in minimap2. Haplotype aware variant consequences were generated using VEP (Variant Effect Predictor) (McLaren et al., 2016) (Dunnen et al., 2016)) and BCFtools/csq (Danecek and McCarthy, 2017). The complete set of scripts for this pipeline can be found in https://github.com/coronahack2020/final_paper.git.

[0]sort -k6,6 -k8,8n OUT.paf | paftools.js call -l 200 -L 200 -q 30 -f REF.fa

6.3 Results

6.3.1 Data Collection and Phylogenetic Analysis

215 bat-CoV genomes of varying families (Alphacoronaviruses and Betacoronaviruses) were able to be collated with only one exhibiting a small proportion or genomic uncertainty (presence of 0.45% 'N' nucleotide). However, only 7 pangolin-CoV genomes, of which 5 were annotated as Betacoronaviurs, were available at the start of this study. 3 pangolin-CoV genomes also contained levels of the ambiguous 'N' nucleotide, two of them at high levels (6.88 and 8.19%). A population of post-outbreak SARS-CoV-2 genomes from Charite (Elbe et al., 2017), Germany, were also collated for further analysis, except for the phylogenetic tree. For the phylogenetic analysis, the complete set of 269 Wuhan and pre-Wuhan genomes (7 pangolin-CoV, 47 Wuhan SARS-CoV-2, including the reference genome, and 215 bat-CoV) were examined. The phylogenetic tree produced at the whole genome level showed a clear separation between the SARS-CoV-2 and the bat-CoV genomes, with the exception of bat-RaTG13 which has been placed adjacent to the SARS-CoV-2 clade (Figure 6.1). bat-RaTG13 and its similarity to SARS-CoV-2 has previously been reported (Zhou et al., 2020). While more distantly related to SARS-CoV-2 than bat-RaTG13, MG772933 and MG772934 (bat-SL-CoVZC45 and bat-SL-CoVZXC21 isolates) are more closely related to SARS-CoV-2 than the remaining bat-CoV (Figure 6.1). Six of the 7 pangolin-CoV genomes are grouped together and closest to the SARS-CoV-2 clade, other than bat-RaTG13. One pangolin-CoV, MT084071.1 (MP789 isolate; referred to as pangolin-MP789), is placed in a branch closer to SARS-CoV-2 than the remaining pangolin-CoV in the tree (Figure 6.1). The tree produced was used as an analytical anchor for which we could use to refer to in the results from variant analysis. High impact variants were annotated on the tree to show their distribution across the different clades along the topology of the tree.

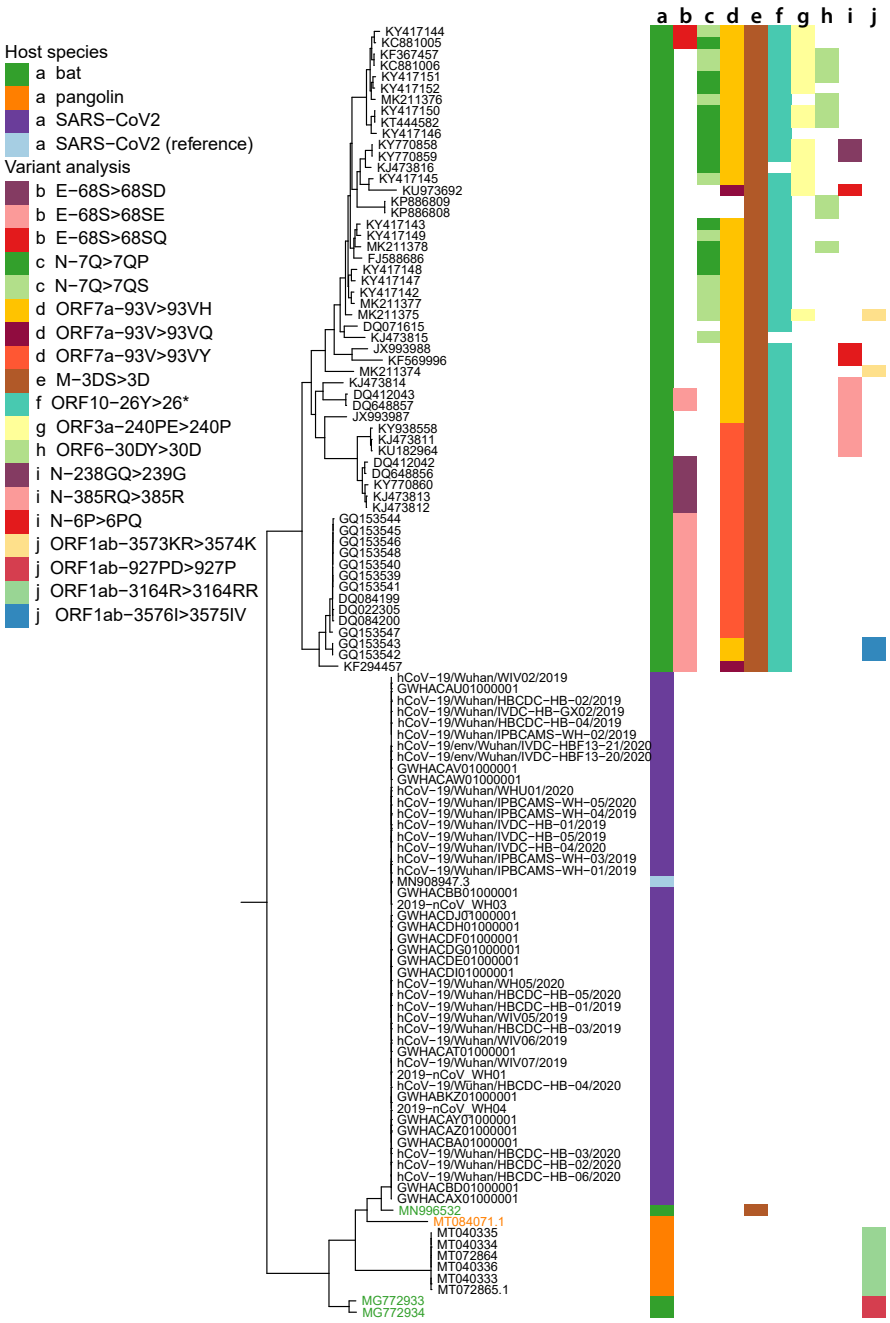


FIGURE 6.1: Phylogenetic tree showing relationship between bat-CoV, pangolin-CoV and SARS-CoV-2. This is the Sarbecovirus clade from Figure 6.9, the phylogenetic tree made with all bat-CoV, all pangolin-CoV and SARS-CoV-2 (Wuhan dataset and SARS-CoV-2 reference) used in this study. Along with the a) host organisms, results from the variant analysis are annotated, showing b-d) positions with multiple amino acid changes, e-h) positions with a single amino acid change (in >10 genomes), and i-j) other variants. The genes and amino acid changes involved in each of the annotated inframe insertion, inframe deletion or stop gain (*) are indicated in the figure legend. The names of four genomes are highlighted, including 3 bat-CoV, MN996532 (bat-RaTG13), MG772933 (bat-SL-CoVZC45), and MG772934 (bat-SL-CoVZXC21), and 1 pangolin-CoV, MT084071.1 (pangolin-MP789), as they are more closely related to SARS-CoV-2 than the other bat-CoV or pangolin-CoV in the tree.

6.3.2 Cross-Host Comparative Genome Annotation

6.3.2.1 StORF-Reporter

The UR-Extractor and StORF-Reporter methodology was used to search for potential protein coding genes which had been missed by the PROKKA pipeline. Specifically, due to the compactness of viral genomes, it was assumed that the URs reported by PROKKA could still contain putative CDSs (CoDing Sequences) that had been missed by PROKKA. Furthermore, although small, PROKKA reported four URs ranging from 291 to 403 nt (lengths extended by UR-Extractor), throughout the 2019-nCoV_WH01 SARS-CoV-2 genome.

Shown in Figure 6.2 with the Interactive Genome Viewer (IGV) (Robinson et al., 2011), the PROKKA (blue) annotations did not report any genomic feature (coding or non-coding) between positions 29,508 and 29,660. Taking into account for the 50 nt extension for both the 5′ and 3′ prime ends made by UR-Extractor, the UR (green) extends between the 11th and 12th genomic feature reported by PROKKA. Applied to the extended UR of 29,458–29,710, StORF-Reporter reported the StORF 29,505–29,649 (red). The protein sequence of this StORF was BLAST-P searched against the non-redundant protein sequence (nr) database at NCBI (O'Leary et al., 2016) and returned a single hit to a single pangolin-CoV protein named ORF10 with 97.37% percentage identity (GenBank Protein Accession: QIG55954.1). Since the first discovery of this StORF in April 2020, the RefSeq database now contains many examples of the ORF10 protein, including those annotated from the SARS-CoV-2 genome.

This finding specifically led to the teams decision to employ the hybrid annotation approach described in the following section of this chapter.

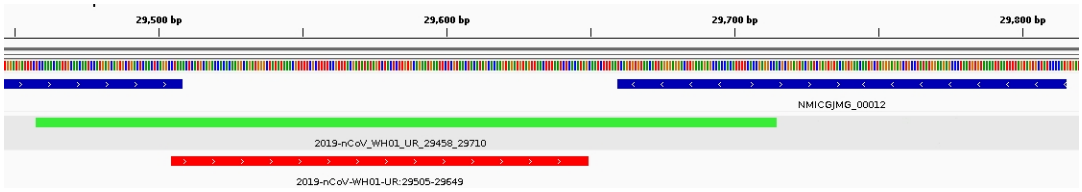


FIGURE 6.2: This 5′ prime section of SARS-CoV-2 genome contains two PROKKA (blue) annotations between positions 29,508 and 29,660. Additionally with 50 nt extensions for both the 5′ and 3′ prime ends undertaken by UR-Extractor, the unannotated region (green) extends between these two PROKKA annotations. StORF-Reporter reported a StORF between position 29,505–29,649 (red) which exhibited 97.37% sequence similarity to the ORF10 gene in a pangolin-CoV.

6.3.2.2 Hybrid Genome Annotation

For each viral genome, a complementary hybrid annotation approach using both PROKKA (Seemann, 2014) and BLAST (Altschul et al., 1990) was employed for identifying genes highly similar to those in the SARS-CoV-2 reference genome released by Ensembl v100 (SARS-CoV-2 ref). The breakdown of this result is shown in Table 6.2, and Table 6.3 presented a detailed account of the genes annotated in each genome and their corresponding annotation tools (PROKKA or BLAST).

Dataset	Min.	Median	Mean	Max.	Sample Count
Wuhan	7	11	11	13	46
Charite	9	11	11	12	117
Bat	2	9	9	12	215
Pangolin	10	11	12	17	7

TABLE 6.2: This table presents the distribution of the number of predicted genes for each dataset. Bat-CoV exhibit the widest distribution of gene count, and pangolin-CoV has the highest number of gene count, with one genome having 17 predicted genes. These outliers have low sequence or assembly quality. In the case of the pangolin-CoV genome reporting 17 genes, it has low quality ('NNNN') nucleotide regions spanning the centre of genes, which causes PROKKA to identify the two ends of one gene. The variance observed only in the median gene count of bat-CoVs, is likely attributable to the large phylogenetic variation exhibited across the bat-CoVs.

Host – Dataset	**No. Genomes**	**No.Genes**	**No. PROKKA**	**No. BLAST**
SARS-CoV-2 WI	46	681	591	90
SARS-CoV-2 GI	117	1736	1495	241
SARS-CoV-2 EWR	1	12	N/A	N/A
Bat	215	2427	2365	62
Pangolin	7	97	95	2

TABLE 6.3: Table containing the total number of genomes and sequences matching genes for each host-species group. Gene-sets listing number of sequences matching genes identified by either PROKKA or BLAST. SARS-CoV-2 group names shortened as; WI: Wuhan Isolates, GI: German Isolates, EWR: Ensembl Wuhan Reference. Listed is the total number of all PROKKA genes identified and the number of BLAST genes which matched an Ensembl reference gene with 80% percentage identity.

PROKKA, which is an alignment-free method, was unable to capture some genes in some of the genomes; BLAST-alignment was used to address this. This has enabled the characterisation of E and ORF10 in many genomes. Genes utilising ribosomal frameshifting such as the aforementioned ORF1ab, are inherently difficult to identify correctly without extensive analysis involving additional techniques and *in situ* evidence such as RNA expression analysis. For the majority of genomes studied, PROKKA was able to identify two large ORFs spanning almost the entire length of the ORF1ab locus and detect a central coronavirus frame-shifting stimulation element (named Corona_FSE and separating the two ORFs) which is a conserved stem-loop of RNA found in coronaviruses that can promote ribosomal frameshifting (Baranov et al., 2005). The gene sequences generated by PROKKA and BLAST (E and ORF10) were used for downstream analysis, including gene-gene network graph, codon usage preference analysis, and a gene-presence summary table. The gene-presence summary table notates whether SARS-CoV ref genes were found ($\geq$ 80% percentage identity and $\geq$ 50% sequence coverage) in each genome; this table is available in the GitHub project `https://github.com/coronahack2020/final_paper/tree/master/host-data`. Supplementary files for each host (in each folder) are named as $*$*_genome_metrics.csv*.

6.3.3 Gene Relationship Network Graph

A gene-gene similarity network analysis was used to compare genes across the SARS-CoV-2, bat-CoV and pangolin-CoV genomes. The advantage of using a 3D network approach to visualise this information was that it can simplify complex connections as visible and therefore easier to interpret patterns. Genes sharing high sequence similarity form independent clusters. In cases where there is a high degree of dissimilarity in a gene for different host-species, a distribution of 2 or more distinct clusters would take place, with each cluster comprised of genes derived from samples of the same host-species. In genes where there is a medium level of dissimilarity across host-species, two or more cluster would appear fused and potentially break apart into distinct clusters if the edge thresholds were increased. Both of these patterns are observed within this dataset. Distinct separation by host-species are seen in ORF1a, ORF3a, ORF6, ORF7a, ORF8 and S (Figure 6.3). The strongest host-species separation observed were between SARS-CoV-2 and bat-CoV; pangolin-CoV always group closer to SARS-CoV-2 than to bat-CoV, with the exception of bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13. In the cases of ORF3a, ORF8 and S, complete separation was observed between bat-CoV and human SARS-CoV-2 (Figure 6.3B & C). Bat-RaTG13 was more similar to SARS-CoV-2 and pangolin-CoV than the remainder of the bat-CoV for S (Figure 6.3C). For ORF3a, bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13 clustered together with SARS-CoV-2 and pangolin-CoV rather than with the remainder of the bat genomes (Figure 6.3). These same three genomes are the only bat-CoV with ORF8 that co-cluster with SARS-CoV-2 ORF8 under the percentage identity threshold ($\geq$80%) set for building the network graph.

Other bat-CoV ORF8 were so distinct from SARS-CoV-2 ORF8 that they do not form edges with SARS-CoV-2 ORF8. Interestingly, within the cluster of ORF8 sequences, the ORF8 for pangolin-MP789 shares an average of 92.14% identity to SARS-CoV-2 ORF8, whilst the ORF8 for remaining pangolin-CoV do not share a strong similarity to the SARS-CoV-2 ref ORF8 (no BLAST result). An average percentage of identity between SARS-CoV-2 ORF8 and bat-CoV ORF8 are 97.05% (bat-RaTG13) and 88.58% (bat-SL-CoVZC45 and bat-SL-CoVZXC21).

To investigate whether it is possible that gene transfer or recombination may have come from a more distantly related bat-CoV, we sought for unusual co-clustering between genes characterised from bat-CoV and SARS-CoV-2. We did not observe such pattern; bat-RaTG13 co-clusters with SARS-CoV-2 for many genes and is also the most similar bat-CoV to SARS-CoV-2 at a genome level. Two additional genes identified by PROKKA, Corona FSE, a non-coding frame-shift stimulation element within ORF1ab and s2m, a stem-loop II-like motif (Robertson et al., 2004) have both been shown to be highly conserved and important for SARS-2-like coronaviruses. s2m has been identified as a mobile genetic element which has been described in a number of single-stranded RNA virus and insect families and has also been shown to be important for viral function (Tengs and Jonassen, 2016; Tengs, Delwiche, and Jonassen, 2020).

In summary, the use of gene-gene network analysis enables us to determine groups of closely related genes, which not only highlights genes showing strong host-species separation, but also characterise clusters of related genes that may be absent or highly different from the reference genome of interest, such as ORF8. 6 genes, ORF1ab, ORF3, ORF6, ORF7a, ORF8 and S, showed a strong host-species separation in the network graph. In particular, with the exception of S, where bat-SL-CoVZC45, bat-SL-CoVZXC21 clustered closer to bat-CoVs, the bat genomes, bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13, clustered together with SARS-CoV-2 than the remainder of the bat-CoV for these 5 genes.

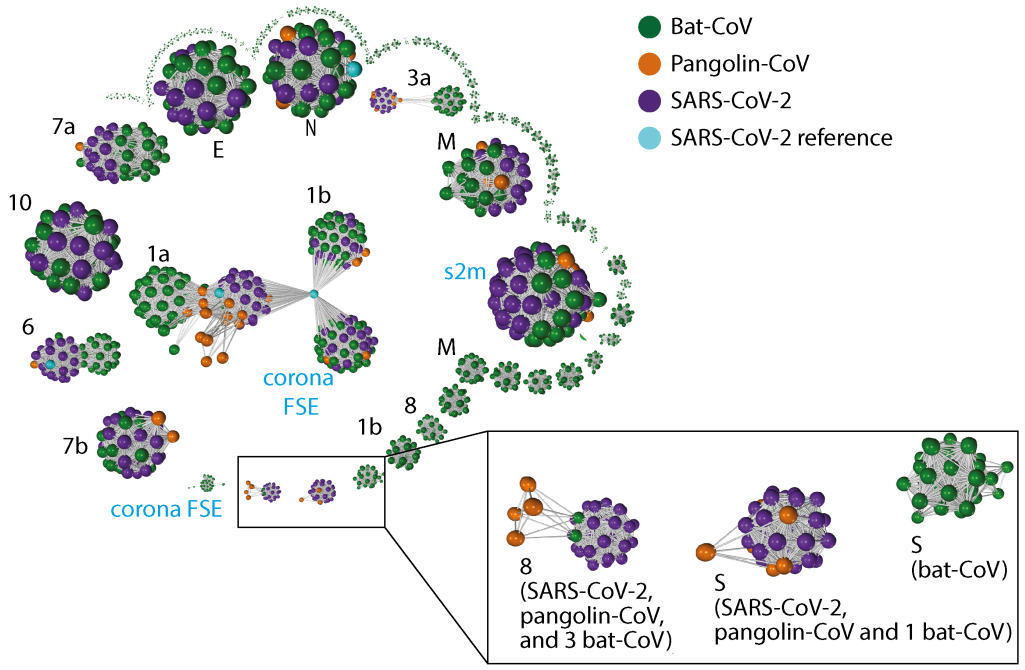


FIGURE 6.3: Gene-gene similarity network analysis. Each node represents a gene defined by PROKKA or a DNA segment similar to genes from the SARS-CoV-2 reference genome. The nodes were compared against each other using BLAST, and nodes with high similarity (bit-score $\geq$ 60 and a query coverage $\geq$ 80%) were connected with an edge. The network graph is labelled with host-species. The black font in the graph indicates the corresponding SARS-CoV-2 gene names ("ORF" omitted) for the larger clusters, whereas blue font indicate additional non-coding sequences defined by PROKKA. Instead of the full length ORF1ab (21k in length), ORF1a and ORF1b were defined by PROKKA as two separate genes. Notably ORF1a, ORF3a, ORF6, and ORF8 and S, show strong separations between nodes from different species. ORF8 from 3 bat-CoV co-cluster with ORF8 from SARS-CoV-2 (RaTG13, bat-SL-CoVZC45 and bat-SL-CoVZXC21 respectively). The remaining bat-CoV ORF8 do not co-cluster with SARS-CoV-2 ORF8 even without the edge filtering threshold. For S, the bat-CoV RaTG13 co-cluster with COVID-19 and pangolin. A cluster of bat-CoVs break off for ORF1b and M, suggesting a large amount of variation amongst bat-CoV for these genes.

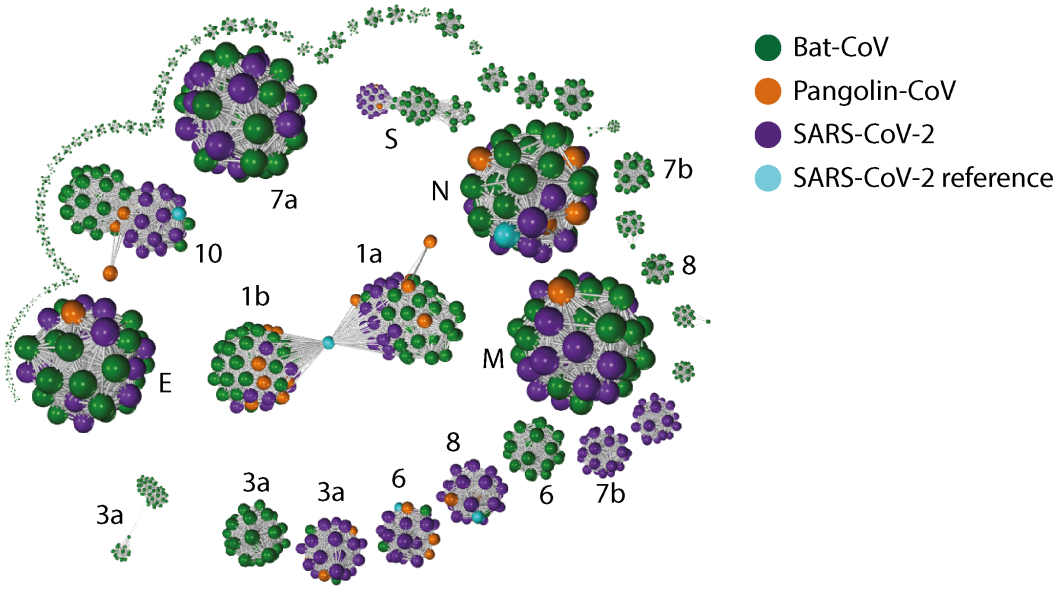


FIGURE 6.4: Gene-gene similarity network analysis. Each node represents an amino acid sequence defined by PROKKA or BLAST (ORF10 and E). The nodes were compared against each other using BLAST, and nodes with high similarity (bit score $\geq$ 60 and a query coverage $\geq$ 80%) were connected with an edge. The network graph is labelled with SARS-CoV-2 gene names ("ORF" omitted). When the network graph is coloured by host species, genes showing higher degree of variability across species are highlighted. Similar to the network analysis on nucleotide sequences (Figure 6.3). Genes ORF3a, ORF6, ORF7b, ORF8, ORF10 and S show strong separation between nodes from different species. The degree of separation in ORF1ab are stronger than ORF10 in the nucleic acid network graph; the reverse is true for the amino acid network graph.

6.3.4 Codon Usage Preference

The RSCU was calculated across SARS-CoV, bat-CoV and pangolin-CoV genomes for each SARS-CoV-2 ref gene. PCA using RSCU showed a strong host-species separation; the first principle component (PC1) accounts for 55.62-85.38% of variation (Figure 6.5), predominately separating SARS-CoV-2 from bat-CoV. Bat-RaTG13, bat-SL-CoVZC45 and bat-SL-CoVZXC21 and pangolin-CoV are usually placed between SARS-CoV-2 and other bat-CoV. With the exception of ORF7b, Pangolin-MP789 is placed closer to SARS-CoV-2 than all other pangolin-CoV (Figure 6.5) with regards to the variation described by PC1 and PC2.

K-means clustering was used to group the genomes into 3 clusters for each gene using the first 10 PCs, which have grouped pangolin-MP789 with SARS-CoV-2 for ORF1a, ORF8, ORF7a, E, ORF6 and N (one of two assemblies). For M and ORF3a, pangolin-MP789 clustered with bat-SL-CoVZC45 and bat-SL-CoVZXC21.

A summary of the synonymous codon ratios (the number of codons divided by the total number of codons coding for the same amino acid), sorted by amino acids, are shown in Figure 6.6.

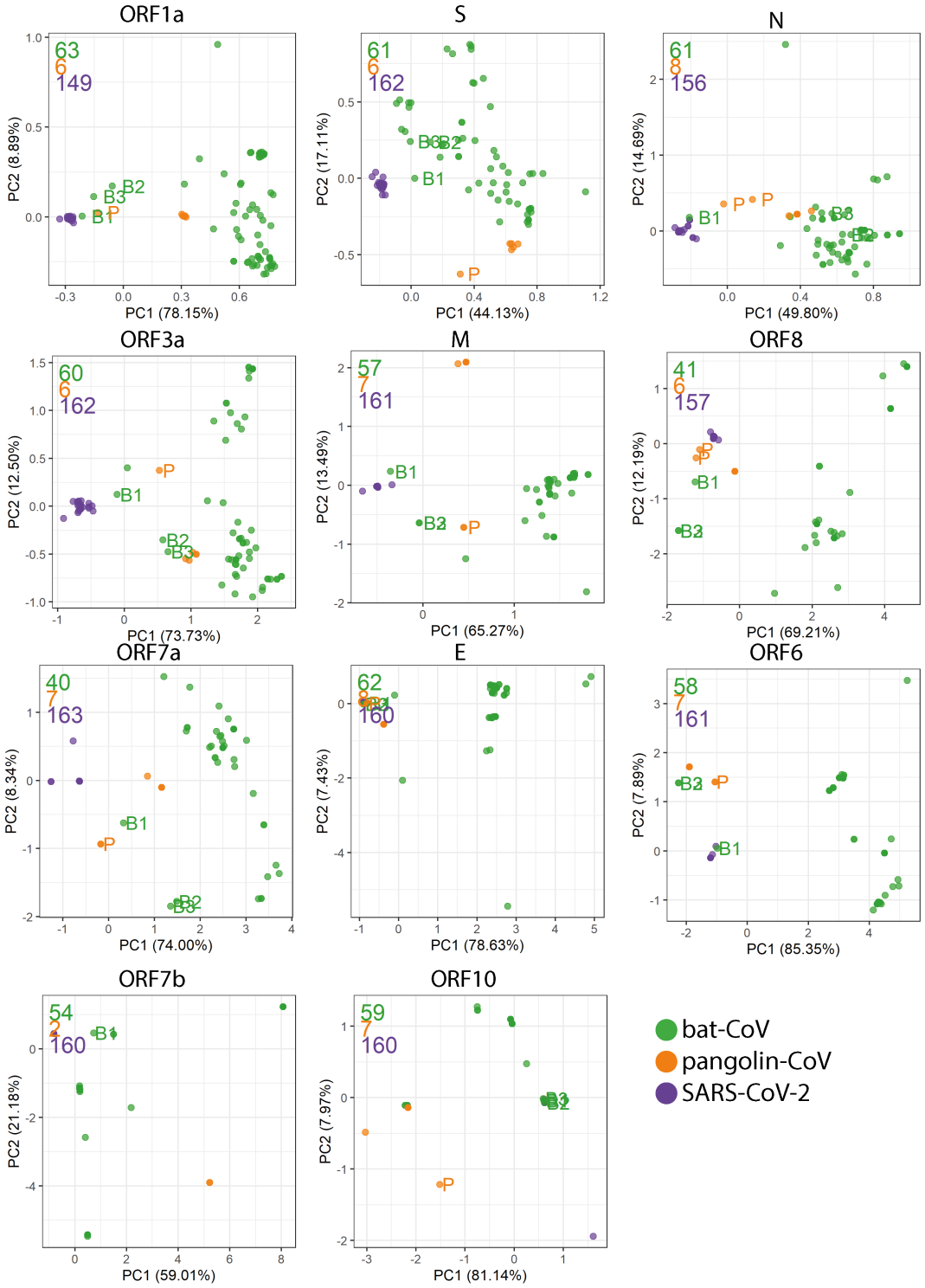


FIGURE 6.5: Relative synonymous codon usage (RSCU) was calculated as the ratio of the observed frequency of codon to the expected frequency under the assumption of equal usage between synonymous codons for the same amino acids. For each gene, Principal Component Analysis (PCA) was carried out on the RSCU values. The first two Principal Components (PCs) are plotted. The total number of genomes used in each plot are indicated in the top left corner in the corresponding colour. In order, they are bat-CoV (green), pangolin-CoV (orange), and SARS-CoV-2 (purple). Four isolates are labelled: bat-RaTG13 (B1), bat-SL-CoVZC45 (B2), bat-SL-CoVZXC21 (B3), and pangolin-MP789 (P; MT121216.1 and MT084071.1).

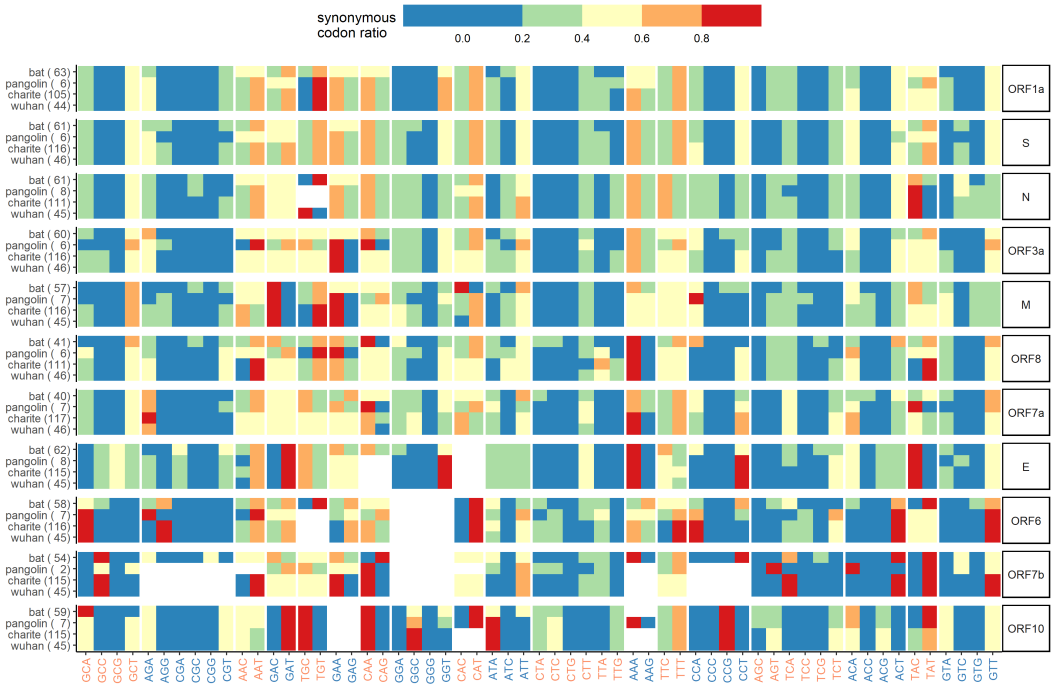


FIGURE 6.6: Synonymous codon ratios are the ratio between the number of a given codon divided by the total number of codon coding for the same amino acid. By sorting this ratio in blocks of synonymous codons, this heatmap illustrates the preferential codons for each amino acid for each dataset across all genes. A number of codon usage preference are consistent across most genes and datasets. For instance, GCT is preferentially used for Alanine and GTT for Valine. On the whole, there appears to be less of a preferential codon use for bat-CoV, especially in longer genes or when multiple genes are accounted for, as indicated by the higher frequency of more evenly distributed codons within each amino acid (i.e. for the bat-CoV dataset, the heatmap colours are of a similar level within each amino acid). Codons with GCs are generally underrepresented, such as in Arg (Arginine), Pro (Proline) and Ser (Serine). * The values in this row were generated by combining codons from multiple genes, E, N, S, ORF1ab, ORF3a, ORF10.

6.3.5 Variant Analysis

Haplotype aware variant calling and variant effect prediction of all genomes in the study has been summarised in Figure 6.7. There are a total of 1,127 variants that are missense, inframe deletion, inframe insertion, stop gained, stop lost, as can be seen in Figure 6.8. We have removed missense from further analysis and came to a total of 24 high impact variations in 8 genes were when comparing bat-CoV and pangolin-CoV genomes against the SARS-CoV-2 ref. We have annotated the majority (with the exception of the NC045512_27675A>ACAG) of these variation in Figure 6.1, and found that some of these variations, such as variants identified in E, ORF7a and ORF3a, appear to exhibit some degree of clade specificity. The only stop gain variant (i.e. NC045512_29635) was present in ORF10 gene of 57 bat-CoV genomes (29635 bp position C>A) which was only representing a synonymous variant in the same position of 6 pangolin-CoV genomes. This variant affected 26Y>26* (Tyrosine to STOP codon TAC>TAA) in bat-CoV ORF10. Assuming the direction of host-selection from bat and pangolin to human, this variant could explain the presence of a longer ORF10 isoform in the 2 latter hosts in comparison to bat-CoV. From the variant Table 6.7, four in-frame insertions were identified as follows:

- **ORF1ab** gene at position 9757 (NC045512_9757 T>TAGA 3164R>3164RR) of all pangolin-Cov genomes which represents an extra Arginine.
- **E** gene at position 26448 (NC045512_26448 T>TGAA 68S>68SE) in 33 bat-Cov genomes which caused an addition of Glutamine.
- **ORF7a** gene at position 27672 (NC045512_27672 T>TCAC 93V>93VH) in 24 bat-Cov genomes by addition of an Histamine.
- **N** gene at position 28293 (NC405512z_28293 A>AACC 7Q>7QP) in 13 bat-Cov genomes by addition of a Proline.

Two in-frame deletions were also identified in ORF3a and M genes. A single Glutamine deletion in ORF3a at position 26,111 was present in 14 bat-Cov genomes (NC045512_26111 CTGA>C 240PE>240P) and a Serine deletion in M gene at position 26,530 (NC045512_26530 ATTC>A 3DS>3D) was present in 57 bat-Cov genomes. The same position presented a missense mutation of 3D>3A (in 2 bat-Cov [bat-SL-CoVZC45 and bat-SL-CoVZXC21] and 1 pangolin-Cov) and 3D>3G in 6 pangolin-Cov genomes.

CHROM	POS	REF	ALT	VAC	consequence	gene_name	amino_acid_change	dna_change	AF	host
NC_045512	3045	CAGA	C	2	inframe_deletion	ORF1ab	927PD>927P	3045CAGA>C	0.01739130	Bat
NC_045512	9757	T	TAGA	6	inframe_insertion	ORF1ab	3164R>3164RR	9757T>TAGA	0.05217391	Pangolin
NC_045512	10983	AAAG	A	2	inframe_deletion	ORF1ab	3573KR>3574K	10983AAAG>A	0.01739130	Bat
NC_045512	10993	C	CGTT	2	inframe_insertion	ORF1ab	3576I>3575IV	10993C>CGTT	0.01739130	Bat
NC_045512	26111	CTGA	C	14	inframe_deletion	ORF3a	240PE>240P	26111CTGA>C	0.12173913	Bat
NC_045512	26447	C	CTGA	5	inframe_insertion	E	68S>68SD	26447C>CTGA	0.04347826	Bat
NC_045512	26448	T	TGAG	16	inframe_insertion	E	68S>68SE	26448T>TGAG	0.13913043	Bat
NC_045512	26448	T	TGAA	33	inframe_insertion	E	68S>68SE	26448T>TGAA	0.28695652	Bat
NC_045512	26448	T	TCAA	2	inframe_insertion	E	68S>68SQ	26448T>TCAA	0.01739130	Bat
NC_045512	26530	ATTC	A	57	inframe_deletion	M	3DS>3D	26530ATTC>A	0.49565217	Bat-Pangolin
NC_045512	27289	GATTAC	GAC	7	inframe_deletion	ORF6	30DY>30D	27289GATTAC>GAC	0.06086957	Bat
NC_045512	27291	TTAC	T	2	inframe_deletion	ORF6	30DY>30D	27291TTAC>T	0.01739130	Bat
NC_045512	27671	T	TTTA	11	inframe_insertion	ORF7a	93V>93VY	27671T>TTTA	0.09565217	Bat
NC_045512	27671	T	TTCA	10	inframe_insertion	ORF7a	93V>93VH	27671T>TTCA	0.08695652	Bat
NC_045512	27672	T	TCAC	24	inframe_insertion	ORF7a	93V>93VH	27672T>TCAC	0.20869565	Bat
NC_045512	27672	T	TTAC	8	inframe_insertion	ORF7a	93V>93VY	27672T>TTAC	0.06956522	Bat
NC_045512	27672	T	TCAG	2	inframe_insertion	ORF7a	93V>93VQ	27672T>TCAG	0.01739130	Bat
NC_045512	27675	A	ACAG	2	inframe_insertion	ORF7a	94Q>94QQ	27675A>ACAG	0.01739130	Bat
NC_045512	28291	C	CCAA	3	inframe_insertion	N	6P>6PQ	28291C>CCAA	0.02608696	Bat
NC_045512	28293	A	AACC	13	inframe_insertion	N	7Q>7QP	28293A>AACC	0.11304348	Bat
NC_045512	28293	A	AATC	11	inframe_insertion	N	7Q>7QS	28293A>AATC	0.09565217	Bat
NC_045512	28987	CCAA	C	2	inframe_deletion	N	238GQ>239G	28987CCAA>C	0.01739130	Bat
NC_045512	29428	ACAG	A	7	inframe_deletion	N	385RQ>385R	29428ACAG>A	0.06086957	Bat
NC_045512	29635	C	A	57	stop_gained	ORF10	26Y>26*	29635C>A	0.49565217	Bat

FIGURE 6.7: High impact variants identified across bat-CoV and pangolin-CoV genomes using the variant calling pipeline based on SARS-Cov-2 Ensembl reference genome. The variants with allele frequency > 0.1 and predicted to have HIGH impact using VEPTools are listed: **CHROM** Contig name, **POS** Position, **REF** Reference allele in Ensembl Human SARS-Cov2, **ALT** Alternative allele(s) found in non-human genomes, **VAC** Alternative variant allele counts and **AF** Allele frequency.

FIGURE 6.8: The coordinate map of all variants called against the human reference SARS-Cov-2 genome. Each horizontal track shows the variants present in the host-species group. The colours show the gene annotation origin of the variant and the shape consequence

6.4 Discussion

During the 5 day hackathon, we endeavoured to utilise the genomic data aggregated by the scientific community and undertook a multifaceted and comprehensive exploration of the genomic sequences (or "similarities and differences") of coronaviruses infecting bat and pangolin hosts, available at the time. We have compared SARS-Cov-2 to all bat-CoV and pangolin-CoV genomes from the listed data repositories (NCBI, VIPR and Databiology) without selecting for strains to represent any specific genera, species or sub-strain. The comparisons spanned across several levels: whole-genome, genes, codons and individual variants.

The origin of SARS-CoV-2 is still unknown and a number of coronaviruses from different hosts have been proposed as potential common ancestors (Lau et al., 2020; Malaiyan et al., 2020). However, bats, especially horseshoe bats, are often linked to SARS-like viruses capable of zoonotic host transfer due to their unique niche as viral reservoirs. This is often characterised by their physiology relatively unaffected under varying viral loads and their natural proximity to human habitation (Li et al., 2005; Banerjee et al., 2019). Furthermore, recombination has been suggested as an avenue for host-transfer for a number of RNA viruses such as SARS-CoV-1 and MERS (Su et al., 2016; Yi, 2020).

Throughout the early stages of the global effort to understand SARS-CoV-2, the number, position, function and importance of its genes was of top priority. Being from the same family as other recent pandemic viruses, initial genomic annotations of SARS-CoV-2 were extrapolated from SARS-CoV-1. Many important genes such as ORF1ab, M, N, and Spike were easily identifiable by analysing such homologs in SARS-CoV-1 and MERS etc. However, ORF10, which until early 2020 was believed to be only found in SARS-CoV-2 (Koyama, Platt, and Parida, 2020), was not annotated by any DNA-only ORF prediction methodology. ORF10 annotation was obtained via RNA expression evidence which still today, its validity in expression and function is still under debate (Jungreis, Sealfon, and Kellis, 2021). The early attempts of annotating the SARS-CoV-2 genome proved difficult with competing gene-sets from the different tools used by the myriad of research teams working on it at the beginning of 2020. As with other Coronaviruses, SARS-CoV-2 contains 10 core genes which were quickly identified when using ORF predictions tools and homology searches across the entire length of the genome. However, due to the large ORF1ab frame-shift gene at the 5′ end of the 30kb genome, the 3′ prime region of Coronavirus genomes often exhibited the highest level of gene-presence variation across species and strain level. This often results in a number of 3′ regions with differing CDS predictions from a number of widely-used, top-performing CDS prediction tools. Understanding the assumed compact nature of RNA viruses in particular, I performed a StORF-Reporter analysis of the 3′ prime region to see whether it was able to identify putative CDSs previously undetected by contemporary techniques.

The results of this analysis show that not only does the StORF-Reporter methodology enable fast, homology-free analysis of novel genomes, but also that annotations, both old and contemporary can be improved with the method (see Figure 6.2). Once the SARS-CoV-2 genome and gene annotation of reference was established, we were able to cross-examine our methodologies and provide comparative overviews of gene composition.

The phylogenetic tree inferred from genomes studied in this work presents a picture of vast bat-CoV diversity and its topology is similar to those of previous studies carried out on pangolin and bat associated coronaviruses when compared to the SARS-CoV-2 genome (Lopes, Mattos Cardillo, and Paiva, 2020). Previous phylogenetic profiling has noted that bat-RaTG13 bares the closest resemblance to SARS-CoV-2 across 55 SARS-like coronavirus genomes (Fahmi, Kubota, and Ito, 2020). Of the the 222 SARS-like coronavirus genomes we have constructed the phylogenetic tree with, bat-RaTG13 remains the closest to SARS-CoV-2, followed by pangolin-MP789, the remaining 6 pangolin-CoV, and then bat-SL-CoVZC45 and bat-SL-CoVZXC21. The relationships between pangolin-MP789 and the 3 aforementioned bat-CoVs have been described (Liu et al., 2020a), but it has not yet been highlighted that pangolin-MP789 is closer to SARS-CoV-2 than the other known pangolin-CoV (Figure 6.1). This relationship has previously been reported and a recombination event between pangolin-CoVs and bat-RaTG13 has been theorised (Xiao et al., 2020).

As well as at genome level, the similarity of bat-RaTG13 and pangolin-MP789 to SARS-CoV-2 is also evident at gene level, in particular, across ORF8 sequences. Only a few closely related SARS-CoV-2 ORF8 orthologues have been identified within bat-betacoronavirus lineages (Ceraolo and Giorgi, 2020; Pereira, 2020). This work has shown the pangoling-MP789 and bat-RaTG13 ORF8 gene exhibit $\geq$ 90% sequence identity to the SARS-CoV-2 ref ORF8. The exact function of ORF8 remains to be elucidated, although studies on ORF8 from SARS-CoV-2 and ORF8ab and ORF8b from SARS-CoV-1 have suggested a role in immune modulation through the interferon signalling pathway (Li et al., 2020b; Wong et al., 2018) and inducing strong antigen response (Hachim et al., 2020). Although the origin or function of the SARS-related coronavirus ORF8 remains unresolved, a 29-nucleotide deletion in ORF8 is often found in SARS-CoV-1, when compared to civet-CoV, suggesting that ORF8 may be important for interspecies transmission (Lau et al., 2015).

Other genes that show strong host-species separation in the gene-gene network analysis include ORF1a, ORF3a, ORF6 and S. It has been previously shown that pangolin-CoV and SARS-CoV-2 S protein were highly similar to each other (97.5%) (Zhang, Wu, and Zhang, 2020). Furthermore, it has been shown that the overall structure of S protein in bat-RaTG13 is highly similar to those in SARS-CoV-2 (Wrobel et al., 2020). This is significant as the S protein plays an important role in the initial penetration and infection of host cells and are often host-specific (Wrapp et al.,

2020). Viruses, through co-evolution with the host have high degrees of flexibility in their receptor usage and capacity to reach binding efficiencies via mutations (Baranowski, Ruiz-Jarabo, and Domingo, 2001; Baranowski et al., 2003) Several human coronaviruses, including SARS-CoV-2, SARS-CoV-1 and human coronavirus NL63 (hCoV-NL63), penetrate the host cell by binding to the host ACE2 through the receptor binding domain (RBD) of S protein (Wu et al., 2011; Hoffmann et al., 2020). It would appear that despite the S protein being more similar between pangolin-CoVs and SARS-CoV-2, the S protein in bat-RaTG13 is still more similar to that of SARS-CoV-2 than other bat-CoVs in our study (Figure 6.3C). This raises the possibility that the most recent common ancestor of SARS-CoV-2 (be of pangolin-CoV or bat-CoV origins) is yet to be sequenced.

Codon usage preference across the species-host range may show signs of preferential codon mutation which have occurred during the complex process of host interaction and transfer (Jitobaom et al., 2020; Kumar et al., 2018). The knowledge of nucleotide profiles and subsequent codons during the human-virus co-evolution could be invaluable to the design of vaccines and their continuous development over the years to come (Rice et al., 2020). On the whole, the codon usage profiles are highly different between SARS-CoV-2 and the majority of bat-CoV, with bat-RatG-13, bat-SL-CoVZC45, bat-SL-CoVZXC21 and panolin-CoV positioned between the two groups. Similar to the analysis by Gu et al. (2020), it was found that codon usage profiles in bat-RaTG13 are most similar to SARS-CoV-2 on the whole (Gu et al., 2020). However, this study included 6 additional pangolin-CoV isolates and found pangolin-MP789 exhibited consistently more similar codon usage profiles to SARS-CoV-2 than the remaining pangolin-CoV at the gene level, which is also reflected in the genome-level phylogenetic tree. These observations highlighted the variation within pangolin-CoV and the closer resemblance between pangolin-MP789 and SARS-CoV-2; pangolin-MP789 is an isolate collected in 2019, whereas all other pangolin isolates were collected prior to 2019. Our codon usage analysis has focused on the overall comparison of RSCU for each gene across bat-CoV; other studies have compared gene sequence characteristics such as GC content and CpG dinucleotide (Nambou and Anakpa, 2020; Alonso and Diambra, 2020; Digard et al., 2020).

Next, the focus was on variants that could potentially have a more profound impact on the amino acid substitution or early stop codon gains (i.e. truncation). Population level viral mutation is a complex process, involving a number of pressures, and while RNA viruses often exhibit some of the highest mutation rates of all viruses, conserved variants can exhibit important functional changes such as the ability to evade immunity more efficiently (Sanjuán and Domingo-Calap, 2016). Furthermore, unlike the vast majority of RNA viruses, coronaviruses encode a complex RNA-dependent RNA polymerase that has a 3' exonuclease domain (Smith, Sexton, and Denison, 2014), effectively proofreading mutational events and therefore are less error-prone. Therefore the mutations observed across populations have undergone

an error-correction process which means they are more likely to be functionally beneficial to the virus. Several of such variants have been observed (allele frequencies > 0.1) that are at consistent loci across different bat-CoV clades as shown in Figure 6.1. Some of these variants are seen in the majority of the bat-CoV samples (which align to the SARS-CoV-2 ref), including a stop-gain for ORF10 and an inframe deletion for M, whilst others, such as the variants seen in ORF7a and E appear to be clade specific (Figure 6.9). Several of these variants affect the same amino acid positions, including E (inframe insertion of *Asp* (Aspartic acid), *Glu* (Glutamic acid) or *Gln* (Glutamine) at at positions 68), N (inframe insertion of *Pro* (Proline) or *Ser* (Serine) at position 7) and ORF7a (inframe insertion of *His* Histidine, *Gln* or *Tyr* (Tyrosine) at position 93) (Figure 6.9). Notably, the stop-gain was identified at amino acid position 26 in ORF10 for 57 of the 59 bat-CoV genomes with ORF10 that had $\geq$ 80% similarity to the SARS-CoV-2 ref. The absence of this stop codon in the pangolin (which exhibited synonymous mutations at the same locus) and SARS-CoV-2 viruses could result in a longer isoform of the ORF10 or fundamental changes in its function and expression levels. In a previous study of SARS-CoV-2 and pangolin-CoV genomes, position 26 was also identified as a region of population level variation from *Tyr* and *His* which significantly modifies the secondary structure of the coil region of the protein (Hassan et al., 2020).

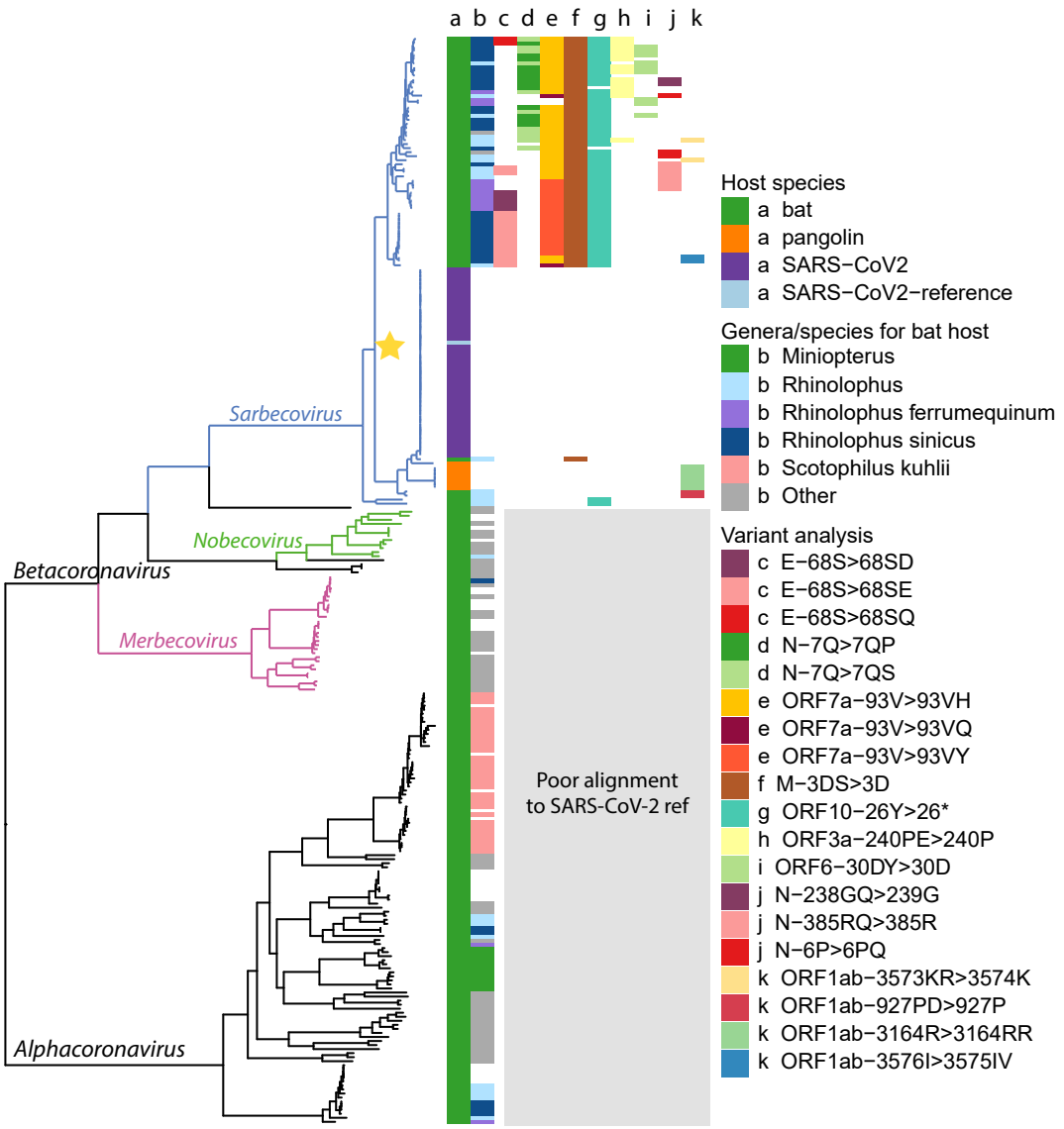


FIGURE 6.9: Ladderised phylogenetic tree of bat-CoV, pangolin-CoV and SARS-CoV-2 (Wuhan dataset and reference) genomes. The hosts for each genome are indicated in a) and host genera or species in b) for bat-CoV. The majority of the Sarbecovirus affect the bat genus Rhinolophus (column b, light blue, dark blue and purple), whereas a much smaller proportion of the Alphacoronavirus are found in bats of this genus. Some clades overlap with specific bat species, including Rhinolophus ferrumequinum, Rhinolophus sinicus and Scotophilus kuhlii. Several high impact variants (inframe insertion, inframe deletion or stop gain) identified from variant analysis overlap with the clades in the phylogenetic tree. The annotation indicates c-e) amino acid positions with multiple variants, f-h) amino acid positions with a single change and found in > 10 genomes, k-l) other variants. The genes and amino acid changes involved in each of the annotated inframe insertion, inframe deletion or stop gain (*) are indicated in the figure legend. Star highlights the clade in Figure 6.1.

There has been little research on ORF10 function, and its expression has been the subject of debate. Whilst Kim et al. (2020) found little evidence of ORF10 expression (0.000009% of viral junction-spanning reads) in cell culture (Vero cells) (Kim et al., 2020), Liu et al (2020) found it to be abundantly expressed in severe COVID-19 patient cases but barely detectable in moderate cases (Liu et al., 2020b). Besides the single ORF10 variant that is observed in the majority of the bat-CoV genomes, 3 different amino acid insertions (4 different nucleotide changes) were observed at position 68 of E gene in 4 different clades of bat-CoVs.

The small envelope E protein is the smallest of coronaviruses' major structural proteins, but also one of the least described (Schoeman and Fielding, 2019). E gene has been shown to be highly expressed inside infected cells and the viruses which are formed without E exhibit reduced levels of viral maturation and tropism. Expression of the E product was essential for virus release and spread, thus demonstrating the importance of E in virus infection and therefore vaccine development (DeDiego et al., 2007). The 68th amino acid position highlighted in this study is in the c-terminal domain, which coincides with the previously reported motif in SARS-CoV-1 (also at 68th amino acid position) that binds to the host cell PALS1 protein to facilitate infection (Teoh et al., 2010).

In one study of 3,617 SARS-CoV-2 genomes, less than 0.5% have been found to have non-synonymous mutation in E, and of these, 20% are at the 68^{th} amino acid position (Hassan, Choudhury, and Roy, 2020). These changes in amino acid may alter the hydrophobicity at the locus, thus possibly influencing the protein functions and interactions (Hassan, Choudhury, and Roy, 2020). Two of the E variants highlighted use different codons for the same amino acid (GAG or GAA for *Glu*), which potentially suggests interplay between the selection pressures of codon optimisation and amino acid insertion into the protein product.

Through this study, a number of inframe insertions have been characterised at the amino acid position 93 in ORF7a across 55 bat-CoV genomes, and at position 94 reported in 2. As with position 68 in E, position 93 in ORF7a has multiple codon insertions coding for the same amino acid but in two groups. In these two groups of bat-CoVs, an additional *His* is encoded for by two different codons and secondly, so is *Tyr* in another group. Intriguingly, ORF7a in SARS-CoV-1 has been shown to regulate the bone marrow stromal antigen 2 which inhibits the release of virions of human infecting viruses (Taylor et al., 2015).

N is another gene for which has been shown to exhibit multiple inframe insertion variants for the same amino acid position. The N protein is highly expressed during an infection, and plays a key role in promoting viral RNA synthesis and incorporating genomic RNA into progeny viral particles (Cong et al., 2020). In gene N, two inframe insertions were observed at amino acid position 7 for *Ser* or *Pro*

from two groups of bat-CoVs (13 and 11 respectively), as well as two inframe deletions at positions 238 and 385. For M in 57 bat-CoV and pangolin-CoV, there is an inframe deletion at position 3, which removed the amino acid *Ser*. At this amino acid position, a missense mutation of (Asp) to Glycine (Gly) is seen in 2 bat-CoV (bat-SL-CoVZC45 and bat-SL-CoVZXC21) and pangolin-MP789, and (Asp) to *Arg* in the remaining 6 pangolin-Cov genomes. Bat-SL-CoVZC45, bat-SL-CoVZXC21 and pangolin-MP789 have been shown to be more similar to SARS-CoV-2 than other coronavirus of the same host on other comparative metrics. M plays an important role in its interactions with both E and S to incorporate virions into the host-cells.

The amino acid positions highlighted through the variant analysis may constitute important differences in the function or folding potential of the protein product. Summarised is the polymorphism along with respective allele frequencies and amino acid consequences in Figure 6.1.

A previous study by Weber et al. (2020) interrogated 572 SARS-CoV-2 genomes isolated worldwide and characterised 10 distinct mutation hotspots that have been found in up to 80% of the viral genomes they examined (Weber, Ramirez, and Doerfler, 2020). Whilst our reported variant positions are not 100 % in concordant with these hotspots, some of them display changes on or adjacent to our reported positions.

6.5 Conclusion

Through employing a number of routine genomic analysis methodologies, this study aimed to bring understanding of the diversity across SARS-CoV-2 and SARS-CoV-2-like coronaviruses by comparing a wide selection of available genomes from the (early stages) starting point of the COVID-19 pandemic. At the core of this work is the constant fluidity of genomic annotation standards and quality. The systematic review of prokaryotic genome annotation tools undertaken in Chapter 2 and the use of the StORF-Reporter methodology, lead to the principal decision to develop and use the hybrid annotation approach. While this work did not directly include the annotations made by the StORF-Reporter, this chapter, clearly instigated by the pandemic, has been a proof of concept for many of the ideas and proposals put forward by this thesis.

As a team, we highlighted a high degree of host-species separation in sequence homology for ORF3a, ORF6, ORF7a, ORF8 and S, as well as codon usage. Along with bat-RaTG13, we have highlighted the pangolin-MP789 isolate to bare stronger resemblance to SARS-CoV-2 than other pangolin-CoV in both whole-genome phylogenetic tree and gene-level codon-usage profiling. Furthermore, a number of amino acid positions that demonstrate high impact variants (inframe insertion/deletion or stop gain) have also been identified in various bat-CoV and pangolin-CoV; these are potentially functionally important positions that warrant further research. The as-yet unknown evolutionary road map undertaken by the ancestor of SARS-CoV-2 to cross over to its now human host is to be investigated for understanding its origin.

Chapter 7

General Discussion and Conclusion

7.1 General Discussion and Conclusions

Throughout the research undertaken for this thesis, I have often been surprised by the assumed level of simplicity reported in the literature with regard to prokaryotic genomics and in particular, their annotation (Hunter, 2008a). However, nothing that I have investigated has led to any other conclusion than that assumption is not only outdated, but also detrimental to the field. While it is true that prokaryotes, and in particular bacteria, have small and compact genomes with 'little intergenic DNA', it has recently been reported that only ~2% of the global prokaryotic taxa are represented by currently sequenced genomes (Zhang et al., 2020b). While existing methods and models are assumed to be competent on that 2%, it is unlikely that they will sufficient to identifying all genomic elements present in the vast majority of genomes yet to be sequenced.

The characterisation of an organism is a multifaceted process which is most often undertaken in a linear fashion. The order of this is [meta]genome assembly, genome annotation, and gene function assignment. Each step can be undertaken via a number of different methods. Each of these methods bring with them their own set of biases and limitations which influence downstream analyses. The impact of this influence on our ability to gain meaningful and accurate genomic knowledge from all studies, both small and large, is still unknown. The analysis of historic and contemporary prokaryotic genome annotation techniques undertaken in Chapter 2 (Dimonaco et al., 2021) reported that existing methods are efficient at identifying those genes which we have many examples of in genomic databases. However, those genes which are 'narrowly' outside the general 'model' of what has been defined as a gene (start codon usage, length, overlap etc), are still routinely missed by all tools studied. From those results, there were many potential avenues for study. In particular, when the result of many top performing annotation tools were combined, many tools were in agreement regarding the locations of large 'intergenic regions'. In Chapter 3 it was hypothesised that these regions are not in fact intergenic but instead harbour genes which fall outside of that general 'model', routinely missed

by prediction tools. This investigation led to the redefinition of these 'intergenic regions' as 'unannotated regions' (URs), yet to be sufficiently investigated. I developed the StORF-Reporter platform to allow for a unified approach to investigating URs and for potential CDS genes. The identification of novel CDS genes was done by bypassing the problems reported in Chapter 2 by extracting Stop Open Reading Frames (StORFs), or Open Reading Frames (ORFs) bounded by stop codons. The default parameters for both UR and StORF reporting were selected as a result of analysis of thousands of prokaryotic genome annotations in Ensembl Bacteria. The results presented in Chapter 3 have reinforced the analysis undertaken in Chapter 2 and continue to refute the definition of completeness in prokaryotic genome annotation. For example, the identification of StORFs showed that relying on set rules such as start codon usage, gene overlap length and minimum gene length leads to many putative genes being missed, many with exact matches to known genes. The Con-StORF extension to StORF-Reporter presented in Chapter 4 exposed the possible extent of under-representation of putative pseudogenes and genes utilising alternative codons in the canonical genome annotation databases we rely on. While it is difficult to evaluate the coding potential of the sequences identified, their conservation and spread across multiple genera suggests that their function is important enough to warrant further investigation. Lastly, the investigation of these unannotated regions through Chapters 3 and 4 barely scratched the surface of what we are missing. The novel core, soft-core and accessory genes discovered with StORF-Reporter have the potential to redefine not only species-wide pangenomic and inter-genera gene collections, but also through their addition, may also redefine our understanding of their phylogeny.

Noting the limitations and obstacles presented so far, another route to profile genomic data was investigated in Chapter 5. It has been long suggested that machine learning will play a pivotal role in the future of bioinformatics. However, many machine learning applications in biology fail to gain traction in the literature. While this can be for a number of reasons specific to each method, often it is due to the 'overcomplexity' of the method and the inability to sufficiently explain the decisions taken by it to the wider biology community, and as such limiting its usefulness. Through a collaboration I undertook during my PhD, which involved the development of a machine learning method to predict virus-host interactions (Liu-Wei et al., 2021), I found that the use of multi-labelled data from a variety of sources makes the resulting model difficult to interpret. Therefore, I have attempted to develop a biology-first machine learning model, named FrameRate, to investigate the use of sequence only data to identify the coding potential from unassembled DNA sequences. Unlike the majority of other machine learning methods, FrameRate uses only amino acid sequences without needing complex and error prone multi-labelled training data. Furthermore, this training data and the binary output of the model, either 'Coding' or 'Non-Coding', allows for greater interpretation of

the decisions being made. FrameRate will allow for fast identification of the coding potential of large [meta]genomic DNA samples, which significantly simplifies subsequent downstream analysis such as functional characterisation.

In March 2020 as the world was entering the COVID-19 pandemic I participated in an online 'hackathon' to offer the skills I had learnt during my PhD studies to help understand coronavirus genomes. At the start of the pandemic the SARS-CoV-2 genome was sequenced, assembled and annotated within a matter of days of the samples getting into the hands of scientists. However, the annotation of certain genes proved problematic and in these early days, it continued to be updated. This was time sensitive research as the number of those infected with COVID-19 was growing daily. Therefore, it was imperative that our findings were made available to the scientific community as soon as possible. This highlights a change in the speed of scientific reporting since during the SARS-CoV-1 outbreak in 2003, the majority of articles with clinical importance were not published until many years after the World Health Organisation had declared the pandemic over (Xing et al., 2010). However, it still took a number of months and a large coordinated scientific effort to come to an agreement on not only the gene position, but also whether specific genes were 'real' (Koyama, Platt, and Parida, 2020). During that time, I was able to develop and employ a hybrid annotation approach to the 'at the time' incomplete annotation of the SARS-CoV-2 genome. This hybrid annotation approach which is described in Chapter 6 (Dimonaco, Salavati, and Shih, 2021), allowed for the identification and comparison of genes and mutations of Coronaviruses from 3 separate host species, human, bat and pangolin. Additionally, using StORF-Reporter, I was able to recover the elusive ORF10 ORF without sequence alignment which was not possible with other ORF prediction tools at the time.

7.1.1 Research Limitations

The work presented in this thesis has at times had to contend with a number of limitations, such as the availability and quality of data, methods, my own abilities, and lastly, COVID-19 restrictions. Key limitations are discussed in their respective chapters. These will affect future work if ways to overcome them are not found and therefore resolving these should be a priority for future research.

The use of the Ensembl Bacteria database for Chapters 2, 3, 4 and 5 provided a level of comparability between each study, and also introduced some of the same inherent biases and errors from that dataset. Out of the original 44,048 genomes available from Release 46 of Ensembl Bacteria, only 6,223 contained fewer than 5 contigs, which itself is still a high level of genome fragmentation. This single database relies on other sources such as GenBank to accumulate its genomic knowledge. Therefore, Ensembl Bacteria does not know where its data came from for the most part, or how it was assembled or annotated. However, the canonical nature of Ensembl Bacteria does at least provide a single point of reference which was used to identify common errors reported in Chapter 2. Additionally, the work of Chapters 3 and 4 could also be affected by the biases in Ensembl Bacteria. There has been a community response to this, and NCBI, GenBank and other databases are currently reannotating their genome collections with contemporary annotation methods. However, even though they are for the most part using NCBI's Prokaryotic Genome Annotation Pipeline which is a state-of-the-art tool (likely to be the 'best' option at the moment), it still contains a number of incomplete assumptions and biases. Also, as the tool continues to undergo changes, it is unlikely that all the genomes will be updated at the same time and therefore there will be different levels of annotation for the different genomes.

The complexity and variability in the process of annotating prokaryotic genomes has been demonstrated throughout this work. The 80-20 dilemma, which is defined as when most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganising huge amounts of data, is in part applicable here. The vast majority of time and resources are put into the collection, sequencing and assembly of DNA with little or no consideration for how it will be annotated. While it is likely that many of the tools used to annotate the canonical genome annotations deposited in Ensembl Bacteria were investigated in Chapter 2, we cannot be certain of which. We are even less sure of what tools were used for assembly and whether different assembly methods fundamentally impact annotation tool performance. What has been thoroughly investigated in Chapters 2, 3, 4 is the performance of genome annotation tools on near-complete genomes, which constitutes the best use case possible. In order to account for the fragmented nature of the majority of genomes deposited in public databases, ORForise and StORF-Reporter must be further developed to address these limitations.

The effects of a gene's absence in a genomic annotation has been characterised for a number of model bacterial species (Wood et al., 2012). Attempts have been made to recognise and resolve missing annotations through the use of homology based methods (Dunne and Kelly, 2017). However, these methods still require high-quality sequence databases and can only be used to find genes which have already been discovered in other studies. While StORF-Reporter can identify under-represented and even completely novel genes, to validate them, either homology signatures to previously detected genes or experimental evidence is required. Without similarity to known genes, the novel sequences can only be validated to some extent by studying their presence and conservation across different species. Nonetheless, the StORFs and Con-StORFs reported in Chapters 3 and 4 require *in vivo* experimental analysis for verification, something which has yet to be done.

A number of limitations of the FrameRate model were discussed in Chapter 5. The majority of these had one common factor: the quality and completeness of the initial dataset, which was used for training and testing the model. This dataset was in part shared with that of Chapters 3 and 4, where it was shown to contain a number of problems relating to the quality of CDS gene annotation. As the training for FrameRate relies entirely on that CDS gene data, any biases or omissions were carried across to it during training. The level of bias or gaps in the knowledge base for FrameRate is inherently difficult (if not impossible) to decipher, as without substantial experimental work to verify spurious predictions, we cannot know if any impact has been made. However, as FrameRate was successfully evaluated against two separate metagenomic assemblies, it is likely that the model has been able to learn an underlying rule of what is required for a protein sequence to be biologically valid.

The speed and salient nature of the analysis conducted in Chapter 6 necessary to annotate the Coronavirus genomes required a combination of approaches and datasets. Firstly, while the SARS-CoV-2 data was retrieved from a single source (Charite and Ensembl), the bat and pangoilin genomes were retrieved from ViPR (Virus Pathogen Resource) which itself contains data from multiple other sources. Therefore, while it was possible to annotate the different genomes with a single combinatory method, the genomes themselves were processed and assembled with a number of different methods which may have had some impact on the resulting cross-genome mutations identified. Additionally, at the time of the study, homology-free tools such as Prodigal were not identifying all genes, putative or otherwise in either of the 3 host sets of Coronaviruses. The hybrid annotation I conducted consisted of the Prodigal CDS gene prediction tool and the BLAST sequence aligner, each containing their own strengths and weaknesses. In part, the combination of both tools did help mitigate for the limitations of each method, and produced a comparable annotation platform which was used successfully to identify a number of putative mutational hotspots across the 3 host sets.

The limitations discussed here and throughout this thesis are by no means exhaustive. However, the success and results presented in this work, show that while these limitations are important and must be resolved, they have not had a significant impact on the analysis conducted. It is also important to note that the vast majority of the limitations that have been identified are in the genomic data used in the analysis and not in the methods used and developed. Therefore, as the genomic data improves over time, so will the usefulness and impact of the work reported here.

7.1.2 Recommendations

The Background of this thesis highlighted several areas lacking information and requiring further study. Whilst some of these were addressed by the work conducted in this thesis, others remain unaddressed. Additionally, there are a number of topics that warrant further research, which would be possible with more advanced methods. In particular, there is a lack of observational studies undertaken to investigate the impact that historical genomic annotation has had on contemporary data and methods. For example, further studies should use the findings and methods of this thesis, in particular that of Chapter 2, to investigate systematic omissions in the genomic databases and ideally trace these omissions back to the specific methods used for their annotation. Supplementary annotation tools such as StORF-Reporter could then be used to further complete the knowledge gaps identified in those annotations.

The work conducted for Chapters 2, 3 and 4 produced a number of interesting discoveries, resulting in recommendations on how genome annotation should be carried out in the future However, there was one element which was first tentatively identified in Chapter 2 and was rigorously reinforced in Chapters 3 and 4; This is that the term 'intergenic region' should be replaced with 'unannotated region' for those regions without current annotation or experimental evidence of function. Further to this, the use of the term 'complete' in genome annotation has been undermined consistently throughout this thesis. The large number of putative pseudogenes reported in Chapter 4 are at odds with the routine lack of reported pseudogenes in genome annotations. Additionally, it could be argued that genes without extensive experimental evidence should be listed with their potential other start and even possibly stop positions. The idea of genes having isoforms is the norm in eukaryotes but alternative codon usage and as such isoforms, are still seen as niche in prokaryotes.

Bioinformatics as a research area has for too long been seen quite differently by biologists and computer scientists. Where it is still too often seen as a black box by many biologists who focus on its ability to produce results, computer scientists often overlook the biological meaning of the task at hand and instead focus on arbitrary algorithm performance metrics. Both sides contribute considerable error and knowledge gaps, however, due to my background and the direction of this thesis, I am specifically concerned with the mistakes made by computer scientists. This is especially true in the field of machine learning. While it is a powerful tool, much

development is undertaken without close coordination with biologists and as such, it is often the case that the biological aims are not kept at the forefront of experimental design. Arbitrary accuracy results are the aim of many studies even when such accuracy is only determinable on the genomic data at hand and if that data is false or incomplete, then those results can be meaningless. Black box machine learning methods, those which are difficult, if not impossible to interpret, will continue to be ignored by biologists, and rightly so. We must therefore endeavor for further cooperation between biologists and computer scientists to develop the field of bioinformatics.

The multitude of annotation methods available at the start of the COVID-19 pandemic allowed for the rapid annotation of SARS-CoV-2 genome. However, due to the disregard of the differences between annotation techniques, it was difficult to compare the annotations due to different output formats (incompatible interpretations of GFF or other formats). However, the most important and frustrating issue was that, even during a pandemic, it was still not routine to report the parameters and in some cases, even the annotation methods used to annotate different collections of the SARS-CoV-2 virus. The idea that all tools are the same is a major embarrassment to genomics and time and resources were wasted due to the continued disregard for genome annotation.

7.1.3 The Re-Usability and Informed Use of Bioinformatics Software is a Problem

One of the core concerns during the development of the various pieces of software during my PhD was to ensure that the transparency and therefore interoperability of the methods employed was closely tied to the utility of the tool itself. To this effect, the parameters used within each of the UR and StORF selection and filtration processes are not only presented clearly, but are modifiable by the end-user. This was an important consideration as the majority of genome annotation methods, including NCBI's PGAP (Tatusova et al., 2016; Tatusova et al., 2016), do not present users with either clarification on parameters or complete control in changing many of them. While this has likely been done to ensure some level of consensus throughout all annotations conducted by such tools, they do, however, take away much of the control and most importantly, understanding, from the scientist. Further to this, as was described in Chapter 2, there are still many biases which are systematic to all genome annotation tools. The choice of CDS gene prediction tool and the data that it is being used to analyse, can have a large impact on the resulting gene collections produced.

The importance of open software and informed methods or tool choice was shown during the publication of the work in Chapter 2 (Dimonaco et al., 2021). After the submission of the preprint online, I was quizzed a number of times as to why I had omitted the PROKKA (Seemann, 2014) genome annotation pipeline from my

analysis. This was despite having investigated Prodigal which is the integral CDS gene prediction method of that pipeline. This lockdown of parameters and opaque nondisclosure of algorithm processes, only furthers the harm any such biases, errors or simply nescience, can inflict. As such, not only are ORForise, StORF-Reporter and FrameRate available as open-source software, but also extension and modification by others are encouraged.

7.1.4 Final Remarks

The rate and scale of genomic data being sequenced will continue to increase for the foreseeable future. The techniques we have for using this data are simply not keeping pace. We need new methods which instead of requiring humans to tell them what to do, can learn for themselves from the data we already have. There is now enough genomic knowledge in the databases to allow for the development of the next generation of tools which should be able to process this vast amount of data. However, the analysis of the data we currently hold is still incomplete and tools such as ORForise and StORF-Reporter must be kept at the forefront of this new stage of genomics. They will enable the comparison of new tools and parameters while keeping track of changes to historic and contemporary genomic annotations.

As shown by the development of FrameRate, the scale of genomic data now held in the databases presents us with a great opportunity to capitalise on the numerous advantages of machine learning. While realistic approaches to the integration of machine learning and genomic data should be undertaken in the future, the resulting methods will inevitably continue to help redefine the field of genomics, for better or worse.

Within the first few months of release of the initial Wuhan variant of SARS-CoV-2 to the scientific community, it had already undergone substantial changes to both its genome assembly and annotation. Therefore, the SARS-CoV-2 pandemic was a timely reminder of the paramount importance of not only fast and accurate genome annotation, but also annotation which is easily inspected, compared and improved upon.

In conclusion, the challenge of characterising the vast and increasing quantity of genomic data presented before us, is also an unmissable opportunity to take a step back and reevaluate how it has been done so far. It is therefore up to us, how we as researchers continue in this field. Will we learn from the lessons of the past or continue to make the very same mistakes for the next three decades?

Appendix A

Chapter 2 Appendix

A.1 Model Organisms (Ensembl Bacteria Release 46)

- *Bacillus subtilis (B. subtilis) - Strain BEST7003 - Assembly ASM52304v1: B. subtilis* is a Gram-positive, genetically tractable, non-pathogenic model organism used in the industrial production of enzymes. It is part of the Firmicute phylum and is a useful model in the study of *Mycobacterium tuberculosis*, which is the the causative agent of tuberculosis. The strain BEST7003 with assembly ASM52304v1 was chosen for this study (Itaya et al., 2005).
- *Caulobacter crescentus (C. crescentus) - Strain CB15 - Assembly ASM690v1: C. crescentus* is a Gram-negative, oligotrophic bacterium commonly found throughout freshwater lakes and streams. It is an important model organism for studying the regulation of the cell cycle, asymmetric cell division, and cellular differentiation and is part of the Proteobacteria phylum. The CB15 strain with the ASM690v1 assembly was chosen for this study (Nierman et al., 2001).
- *Escherichia coli (E. coli) K-12 - Strain ER3413 - Assembly ASM80076v1: E. coli* is one of the most extensively studied microorganisms and is part of the Proteobacterium phylum. *E. coli* is Gram-negative and its genome was first completely sequenced in 1997. It was chosen then for its unique biochemical, molecular and biotechnological attributes but it widely studied now due to its tractability. The K-12 ER3413 strain with the ASM80076v1 assembly was chosen for this study (Anton et al., 2015).
- *Mycoplasma genitalium (M. genitalium) - Strain G37 - Assembly ASM2732v1: M. genitalium* is a parasitic bacterium with one of the smallest currently known genomes of any free living bacterium at around 580,000 bps. Due to it being a human pathogen and its unique genome size, *M. genitalium* has been used as a model for a minimal organism in the study of essential genes due to being one of the most streamlined bacterial genomes currently known (Glass et al., 2006). Although *M. genitalium* does not have cell walls, it is believed to have evolved from Gram-positive bacteria which had lost their cell wall and is part of the Firmicute phylum. The G-37 strain with ASM2732v1 assembly was chosen for this study (Hutchison et al., 1999).

- *Pseudomonas fluorescens (P. fluorescens) - Strain UK4 - Assembly ASM73042v1: P. fluorescens* is a rod-shaped, Gram-negative bacterium and is part of the Proteobacteria phylum.The antibiotic Mupirocin can be produced by cultured *P. flurescens* and is used in the treatment of skin, ear and eye disorders and is a model organism for cell cycle, cell division and differentiation. The UK4 strain with the ASM73042v1 assembly was chosen for this study (Dueholm, Danielsen, and Nielsen, 2014).
- *Staphylococcus aureus (S. aureus) - Strain 502A - Assembly ASM59796v1: S. aureus* is Gram-positive bacterium of the Firmicute phylum and is commonly found on the human body, including the nose, skin and the respiratory tract. It has been known to cause diseases such as infective endocarditis and a drug resistant strain is commonly known as Methicillin-resistant *Staphylococcus aureus* (MRSA). The 502A strain with assembly ASM59796v1 was chosen for this study (Parker et al., 2014).

A.2 Prediction Tools

A.2.1 Prediction Tools Run-Parameters

All tools were provided with the same 6 DNA data files each containing the complete genome for an organism in a single sequence. We did not provide genome-specific parameters such as alternative codon tables to the tools as this study aimed to be representative of real-world analysis where such information may not be known. Each tool was run using its default parameters with no user-defined filtering.

Each prediction was performed locally on a 64-bit Linux machine with an i7 2600k CPU with 24GB RAM, however none of the tools required more than a few minutes to run or more than 500 MB of RAM.

While most of the tools were available as online resources, they were downloaded from the links included in their associated publications and the specific versions used are listed below. Where no version number is available, the year of when the tool was last modified is listed.

A.2.2 Prediction Tools

- Model-based group:

  Some tools have been designed for a specific set genomes or strains and require a pre-built model (a rigid set of parameters tuned to a particular organism) to perform predictions. The construction of these models rely heavily on having an accurate and complete set of genes for a particular organism (among other information). While inaccuracies or biases in the data are likely to be present in the final models, model-based gene predictors trained on a particular species are expected to perform well on strains with comparable gene and genome structure. Overfitting can occur, where only similar genes to those previously found are detected at a high sensitivity. However, there can be large differences in gene number, gene length and genome size between strains of the same species. Model-based prediction for certain model organisms where specific strains are often used for scientific and industrial purposes can still be effective as there may be little genetic difference between two isolates of the same strain.

  The model-based tools were provided with two different organism models, *E. coli* K-12 and *S. aureus* - Mu50 (strains selected where possible). These were chosen as both were in the set of six bacteria and were models which were already available for all model-based tools. In addition, Augustus, which was originally developed for eukaryotic gene prediction, was run with the inclusion of the *H. sapiens* model and each individual Coding Sequence (CDS) predicted was retained as an independent predicted CDS.

- **Augustus** Keller et al., 2011 - Version 3.3.3
  Originally published in 2003, Augustus was developed as a eukaryote genome prediction tool combining protein-family-based gene prediction and incorporated knowledge from external sources (pre-computed genome models) to combine them with an *ab initio* prediction to specifically help with exon prediction. Later versions of Augustus included 3 bacterial and 1 archaeal species to the pre-computed model list to allow for a selection of prokaryotic genome annotation.

- **EasyGene** Nielsen and Krogh, 2005 - Version 1.2
  EasyGene 1.2 published in 2005, employs a genome specific Hidden Markov Model (HMM) which after extracting all CDSs above 120 nt, filters them by using a sequence similarity search to a protein database. The resulting genes and their start positions are then used to retrain the HMM. EasyGene produces scores for multiple potential start codons for each gene and selects the one with the highest computed confidence value.

- **GeneMark.hmm** Lukashin and Borodovsky, 1998 - Prokaryote Model Version 3.2.5
  GeneMark.hmm, published in 1998 was developed to be one of the first tools to "improve the gene prediction quality in terms of finding exact gene boundaries". A HMM is used to model gene boundaries as transitions between hidden states along with ribosomal binding site patterns to refine translation initiation codons. The current genome model parameters were derived from the use of GeneMarkS, the successor of GeneMark.hmm.

- **GeneMark** Borodovsky and McIninch, 1993 - Version 2.5
  GeneMark, developed in 1993, was one of the first gene prediction methods to efficiently perform whole-genome annotation, notably for its ability to predict CDSs on both strands of DNA simultaneously. Markedly, GeneMark was used for the first annotation of a completely sequenced bacterium, *Haemophilus influenzae*, and the first completely sequenced archaeon, *Methanococcus jannaschii*. The GeneMark algorithm consists of species-specific inhomogeneous Markov chain models computed from protein-coding DNA sequences and homogeneous Markov chain models of non-coding DNA. Probability of a predicted sequence fragment to be protein coding in one of six possible frames (including three frames in complementary DNA strand) or to be "non-coding" is computed to determine potential genes in the opposite strand of DNA.

- **FGENESB** Salamov and Solovyevand, 2011 - '2020' The FGENESB pipeline identifies protein, tRNA and rRNA genes, potential promoters, terminators and operons and performs an initial prediction of 'long' CDSs as a

starting point for calculating parameters for gene prediction. The gene prediction algorithm is based on Markov chain models of coding regions and their translation and termination sites. Furthermore, operon prediction is performed using distances between CDSs, frequencies of neighboring genes in known bacterial genomes and positions of predicted promoters and terminators. FGENESB, unlike other model-based prediction tools, presents its model selection as "Choose closest organism", rather than "select species/organism", indicating that the developers acknowledge the models may be used as best-fit rather than for exact species prediction.

- *Ab initio* group:

  Self-training tools do not require any previous knowledge of the target genome and predict *ab initio*, directly from sequence. These were developed to be used on different prokaryotic organisms, however, they do rely on broad models either trained on features gathered directly from the input genome or predict CDSs using a set of predefined parameters which may be adapted. The criteria considered while making predictions include but are not limited to, overlapping CDSs, GC content, CDS length, predicted start and stop codons, and distances between CDSs Delcher et al., 1999; Besemer, Lomsadze, and Borodovsky, 2001. Unfortunately, these criteria and their thresholds are still based on prior knowledge as deciding between candidate CDSs still requires a number of assumptions based on previously studied genes and genomes which the developer has embedded into the algorithm.

  Transdecoder, while technically an *ab initio* tool, is unlike the others in this group as it was specifically designed to predict CDS regions in transcript data.

  - **Prodigal** Hyatt et al., 2010 - Version 2.6.2 Prodigal is an unsupervised gene predictor which examines the input genome for the creation of its input-specific training set. 100 prokaryote genomes were selected in the initial development of the algorithm to determine "very general rules about the nature of prokaryotic genes, such as gene size, maximum overlap between two genes... and RBS (ribosomal binding site) motif usage". A number of constants within the algorithm were tuned to the genetic makeup of the 100 genomes. GC is an important statistic for Prodigal and it is used for a number of steps in the prediction process such as coding scores for each gene predicted. Prodigal performs a number of scoring functions on different aspects of each DNA region selected, thus producing a set of putative "most-likely real" genes. These genes are then examined and are used to tune the model before prediction of genes which exhibit lower likelihood scores. Furthermore, Prodigal has been designed

to detect whether genetic code 4 is needed (*Mysoplasma*) and use it instead of the default code 11.

- **GeneMarkS** Besemer, Lomsadze, and Borodovsky, 2001 - Version 4.25 Developed in 2001, GeneMarkS was one of the first *ab initio* gene prediction methods which could learn directly from short (>400) sequences without prior knowledge or pre-trained models. As with other contemporary tools, HMMs were trained on protein-coding sequence data, non-coding DNA samples and modelled on transition and initiation parameters trained from input sequence. Codon frequencies and positional statistics are utilised along with genomic GC content to learn coding potential for identified CDSs. GeneMarkS has become a bedrock for future prediction tools and has been used as part of wider genome annotation pipelines.
- **GeneMarkS 2** Lomsadze et al., 2018 - Version '2020' An advancement over the original GeneMarkS tool, GeneMarkS-2 further utilises a self-derived *ab initio* training model learnt from input sequences for finding species-specific (native) genes. A collection of pre-computed "heuristic" models are utilised to identify harder-to-detect genes (horizontally transferred). GeneMarkS-2 learns distinct sequence patterns inherent to prokaryotic genomes which are involved in gene expression control. The majority of protein-coding regions in prokaryotic genomes are known to carry species-specific codon usage patterns and GeneMarkS-2 learns these patterns and estimates parameters of typical protein-coding regions of a target genome. This process is similar to the one employed by GeneMarkS(1) but extended.
- **GLIMMER 3** Delcher et al., 2007 - Version 3.02 GLIMMER 3, published in 2007 is the third iteration of the GLIMMER microbial gene predictor software. A number of improvements over the previous implementations include improved coding region and start codon detection, along with a reduction in incorrectly reported overlapping genes. GLIMMER 3, as with the previous versions, starts by predicting CDSs with little filtering and then using a number of user defined (or default) parameters (such as start codon selection, CDS length and overlap length). These CDSs are then scored for their coding potential. To overcome the high levels of potential false positive overlapping CDSs, GLIMMER3 uses these scores to select which of any two overlapping CDSs are more likely to be real (in cases where maximum overlap is surpassed). An Interpolated Markov Model (IMM) is used in the prediction process to help identify coding regions and has also been shown to separate DNA between bacterium and host DNA. GLIMMER, along with GeneMarkS, was also used as part of the NCBI prokaryote annotation pipeline (Tatusova et al., 2016).

- **GeneMark Heuristic Approach** Besemer and Borodovsky, 1999 - Version 3.25 As with many other tools from the GeneMark suite, GeneMark Heuristic Approach (GeneMark HA) was developed on the observations made from GeneMark and GeneMark.hmm. The method was designed to build Markov models derived on a minimal amount of DNA information from 17 completed bacterial genomes. Linear regression was performed to approximate relationships between positional and global nucleotide frequencies, relationships between the amino acid frequencies and the global GC% of the bacterial genomes. Amino acids frequencies were calculated mostly from an *E. coli* genome to build constants for the algorithm. The algorithm builds a heuristic model for every sequence longer than 400 nt. GeneMark HA derived models are expected to be applied to the analysis of the input sequence by the GeneMark and GeneMark.hmm programs.
- **TransDecoder** Haas et al., 2013 - 5.5.0 TransDecoder was designed to identify candidate coding regions within transcript sequences, such as those generated by *de novo* RNA-Seq transcript assembly, or constructed based on RNA-Seq alignments to the genome. TransDecoder identifies likely coding sequences based on a minimum ORF length and a computed log-likelihood score >0. The coding score is greatest when the ORF is scored in the 1st reading frame as compared to scores in the other 5 reading frames. The longer of two CDSs is reported if one is encapsulated by the others coordinates. However, a single transcript can report multiple CDSs (allowing for operons, chimeras, etc).
- **FragGeneScan** Rho, Tang, and Ye, 2010 - 1.3.0 FragGeneScan has been specifically designed to improve prediction performance on metagenomic and short-read sequence data with high levels of sequencing errors, but also perform comparably with other contemporary tools on complete genomes. A combination of probabilistic models trained on codon usage and sequence error data, was used to evaluate sections of DNA for their gene encoding potential. This method has shown higher performance for predicting genes on short-reads with high levels of sequence error than other contemporary methods but can be used on complete low-error genomes.
- **MetaGene** Noguchi, Park, and Takagi, 2006 - 2.24.0 MetaGene, one of the first gene prediction tools specifically developed for prediction on fragmented and metagenomic genomes, examines di-codon frequencies estimated by the GC content of a given sequence with other measures such as length, distance between CDSs and start codon distribution. MetaGene can predict a whole range of prokaryotic genes based on the anonymous genomic sequences of a few hundred bases and identify partial CDSs

which have are located on the terminus of the fragmentary genomic sequences.

- **MetaGeneMark** Zhu, Lomsadze, and Borodovsky, 2010 - '2020' The heuristic model behind MetaGeneMark was developed to replace traditional methods of ORF prediction parameter estimation such as supervised training on a set of "validated" genes or unsupervised training on an input sequence. Dependencies which had formed in evolution, between codon frequencies and genome nucleotide composition are utilised to derive patterns of codon frequencies, critical for the model parameterisation, from frequencies of nucleotides observed in a short or metagenomic sequences. An effective method to estimate prediction parameters was derived from the frequencies of oligonucleotides in protein-coding regions and whole-genome nucleotide composition.
- **Meta Gene Annotator** Noguchi, Taniguchi, and Itoh, 2008 - Version '2008/8/19' Published in 2008, MetaGeneAnnotator predicts all kinds of prokaryotic genes from anonymous genomic sequences. It integrates statistical models of prophage, bacterial and archaeal genes, and builds a self-trained model from input sequences for the predictions. This results in the detection of not only "typical genes but also atypical genes, such as horizontally transferred and prophage genes in a prokaryotic genome". The algorithm also includes a novel approach for the analysis of ribosomal binding sites, which has enabled the detection of species-specific patterns, thus allowing for "precise" prediction of translation starts sites.

Metagenomic gene predictors form a subset of *ab initio* self-training tools which primarily rely on the same methods but involve additional parameters. They must contend with a number of additional difficulties common to metagenomic annotation. The dynamics of metagenomic DNA sequences such as chimeric contigs assembled from different organisms, cause a number of problems for even self-learning predictors. Any model or parameters chosen would need to be recalculated for every metagenomic contig as each is likely to be from a different organism and therefore have different characteristics. Metagenomic assemblies often consist of fragmented genomes which can lead to a number of problems for gene prediction. A given contig may only contain a fragment of a gene. Therefore, simply looking for start and stop codons, which may not be present, along with changes in GC content outside of predicted gene regions, will not be as useful to help to distinguish between coding and non-coding regions. These errors are extremely difficult to account for and tools have been produced to tackle them directly Rho, Tang, and Ye, 2010 We have included 3 metagenome prediction tools in this study.

Many of the tools comprise different versions of the same core software but produced differing results. For example, GeneMark, MetaGeneMark, GeneMark

Heuristic Approach, GeneMark Hidden Markov Model, GeneMark S and GeneMark S2 were all from the same suite of tools and have many similarities with each other but are designed for different purposes and produce different results.

It was decided that no specific rules were to be enforced on the tools. Each tool was run using its default parameters and this was to get a baseline for their accuracy with the least amount of human support. Many hard-coded assumptions were consistent across the tools, such as minimum ORF length and the codons allowed to identify the start and end of an ORF. Some of the tools allowed the minimum ORF length to be altered, but the majority fixed the threshold to around 100 nucleotides.

A.3 ORForise User Menus

A.3.1 Annotation_Compare

```
usage: Annotation_Compare.py [-h] -dna GENOME_DNA [-rt REFERENCE_TOOL]
-ref REFERENCE_ANNOTATION -t TOOL -tp TOOL_PREDICTION [-o OUTNAME] [-v {True,False}]

optional arguments:
  -h, --help            show this help message and exit
  -dna GENOME_DNA, --genome_DNA GENOME_DNA
                        Genome DNA file (.fa) which both annotations are based on
  -rt REFERENCE_TOOL, --reference_tool REFERENCE_TOOL
                        What type of Annotation to compare to? -- Leave blank
                        for Ensembl reference- Provide tool name to compare output
                        from two tools (GeneMarkS)
  -ref REFERENCE_ANNOTATION, --reference_annotation REFERENCE_ANNOTATION
                        Which reference annotation file to use as reference?
  -t TOOL, --tool TOOL  Which tool to analyse? (Prodigal)
  -tp TOOL_PREDICTION, --tool_prediction TOOL_PREDICTION
                        Tool genome prediction file (.gff) - Different Tool Parameters
                        are compared individually via separate files
  -o OUTNAME, --outname OUTNAME
                        Define full output filename (format is CSV) - If not provided,
                        summary will be printed to std-out
  -v {True,False}, --verbose {True,False}
                        Default - False: Print out runtime status
```

FIGURE A.1: Command line menu for ORForise Annotation_Compare.py

A.3.2 Aggregate_Compare

```
usage: Aggregate_Compare.py [-h] -dna GENOME_DNA -t TOOLS -tp TOOL_PREDICTIONS
[-rt REFERENCE_TOOL] -ref REFERENCE_ANNOTATION [-o OUTNAME] [-v {True,False}]

optional arguments:
  -h, --help            show this help message and exit
  -dna GENOME_DNA, --genome_DNA GENOME_DNA
                        Genome DNA file (.fa) which both annotations are based on
  -t TOOLS, --tools TOOLS
                        Which tools to analyse? (Prodigal,GeneMarkS)
  -tp TOOL_PREDICTIONS, --tool_predictions TOOL_PREDICTIONS
                        Tool genome prediction file (.gff) - Providefile locations
                        for each tool comma separated
  -rt REFERENCE_TOOL, --reference_tool REFERENCE_TOOL
                        What type of Annotation to compare to? -- Leave blank for
                        Ensembl reference- Provide tool name to compare output from two tools
                        (GeneMarkS)
  -ref REFERENCE_ANNOTATION, --reference_annotation REFERENCE_ANNOTATION
                        Which reference annotation file to use as reference?
  -o OUTNAME, --outname OUTNAME
                        Define full output filename (format is CSV) - If not provided,
                        summary will be printed to std-out
  -v {True,False}, --verbose {True,False}
                        Default - False: Print out runtime status
```

FIGURE A.2: Command line menu for ORForise Aggregate_Compare.py

A.3.3 GFF_Adder

```
usage: GFF_Adder.py [-h] -dna GENOME_DNA [-rt REFERENCE_TOOL] -ref REFERENCE_ANNOTATION
[-gi GENE_IDENT] -at ADDITIONAL_TOOL -add ADDITIONAL_ANNOTATION [-olap OVERLAP] -o OUTPUT_FILE

optional arguments:
  -h, --help            show this help message and exit
  -dna GENOME_DNA, --genome_DNA GENOME_DNA
                        Genome DNA file (.fa) which both annotations are based on
  -rt REFERENCE_TOOL, --reference_tool REFERENCE_TOOL
                        Which tool format to use as reference? - If not provided,
                        will default to standard Ensembl GFF format, can be Prodigal
                        or any of the other tools available
  -ref REFERENCE_ANNOTATION, --reference_annotation REFERENCE_ANNOTATION
                        Which reference annotation file to use as reference?
  -gi GENE_IDENT, --gene_ident GENE_IDENT
                        Identifier used for extraction of "genic" regions from reference
                        annotation "CDS,rRNA,tRNA": Default for is "CDS"
  -at ADDITIONAL_TOOL, --additional_tool ADDITIONAL_TOOL
                        Which format to use for additional annotation?
  -add ADDITIONAL_ANNOTATION, --additional_annotation ADDITIONAL_ANNOTATION
                        Which annotation file to add to reference annotation?
  -olap OVERLAP, --overlap OVERLAP
                        Maximum overlap between reference and additional genic regions
                        (CDS,rRNA etc) - Default: 50 nt
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Output filename
```

FIGURE A.3: Command line menu for ORForise GFF_Adder.py

A.3.4 GFF_Intersector

```
usage: GFF_Intersector.py [-h] -dna GENOME_DNA [-rt REFERENCE_TOOL] -ref REFERENCE_ANNOTATION
[-gi GENE_IDENT] -at ADDITIONAL_TOOL -add ADDITIONAL_ANNOTATION [-cov COVERAGE] -o OUTPUT_FILE

optional arguments:
  -h, --help            show this help message and exit
  -dna GENOME_DNA, --genome_DNA GENOME_DNA
                        Genome DNA file (.fa) which both annotations are based on
  -rt REFERENCE_TOOL, --reference_tool REFERENCE_TOOL
                        Which tool format to use as reference? - If not provided,
                        will default to standard Ensembl GFF format, can be Prodigal or
                        any of the other tools available
  -ref REFERENCE_ANNOTATION, --reference_annotation REFERENCE_ANNOTATION
                        Which reference annotation file to use as reference?
  -gi GENE_IDENT, --gene_ident GENE_IDENT
                        Identifier used for extraction of "genic" regions from
                        reference annotation "CDS,rRNA,tRNA": Default for is "CDS"
  -at ADDITIONAL_TOOL, --additional_tool ADDITIONAL_TOOL
                        Which format to use for additional annotation?
  -add ADDITIONAL_ANNOTATION, --additional_annotation ADDITIONAL_ANNOTATION
                        Which annotation file to add to reference annotation?
  -cov COVERAGE, --coverage COVERAGE
                        Percentage coverage of reference annotation needed to confirm
                        intersection - Default: 100 == exact match
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Output filename
```

FIGURE A.4: Command line menu for ORForise GFF_Intersector.py

A.4 Description of Comparison Metrics

- Number of Predicted CDSs:
  This is the number of CDSs that the tool has predicted. Some tools predict a large number of potential CDSs and then filter them. All 'Predicted CDS' metrics correspond to the remaining predicted CDSs presented to the user after default filtering.
- Percentage Difference of Number of Predicted CDSs: **(M3)**
  This is the percentage change between the number of predicted CDSs and the number of actual reference Genes. *100 * (Number of predicted CDSs - Number of reference Genes) / Number of reference Genes*
- Number of Predicted CDSs that Detect a Gene:
  This is the number of CDSs that correctly detect at least 75% of the nucleotides of a reference Gene and are in the same frame.
- Percentage of Predicted CDSs that Detected a Gene: **(M2)**
  This is the percentage of predicted CDSs that correctly detect at least 75% of the nucleotides of a reference Gene and are in the same frame.
- Number of Genes Detected:
  The number of reference Genes Detected is characterised as the number of predicted CDS which are in frame with a reference gene and has captured at least 75% of its nucleotide sequence.
- Percentage of Genes Detected: **(M1)**
  The percentage of reference Genes Detected is characterised as the percentage of predicted CDS which are in frame with a reference gene and has captured at least 75% of its nucleotide sequence.
- Median Length of All Predicted CDSs:
  Median length of all predicted CDSs, in nucleotides.
- Percentage Difference of Median CDS Length: **(M4)**
  This is the Percentage Difference from the mean length of reference Genes compared to the mean length of all predicted CDSs. *100 * (Median CDS length - reference gene median length) / reference gene median length*
- Minimum Length of All Predicted CDSs:
  The length of the shortest predicted CDS, in nucleotides.
- Minimum Length Difference:
  This is the percentage difference from the shortest reference gene compared to the length of the shortest predicted CDS. *100 * (Minimum CDS length - Minimum reference gene length) / Minimum reference gene length*

- Maximum Length of All Predicted CDSs:
  The length of the longest predicted CDS, in nucleotides.
- Maximum Length Difference:
  This is the percentage difference from the longest reference Gene compared to the length of the longest predicted CDS. *100 * (Maximum CDS length - Maximum reference gene length) / Maximum reference gene length*
- Median GC Content of All Predicted CDSs:
  This median GC content calculated from all predicted CDSs.
- Percentage Difference of All Predicted CDSs Median GC:
  This is the Percentage Difference of the median GC content of all predicted CDSs compared to the median GC content of all reference Genes. *100 * (Median GC content of all CDSs - Median GC content of all reference genes) / Median GC content of all reference genes*
- Median GC Content of Matched Predicted CDSs:
  This median GC content calculated from predicted CDSs that detected a reference gene.
- Percentage Difference of Matched Predicted CDS GC:
  This is the Percentage Difference of the median GC content of predicted CDSs that detected a reference gene compared to the median GC content of all reference genes. *100 * (Median GC content of matched CDSs - Median GC content of all reference genes) / Median GC content of all reference genes*
- Number of Predicted CDSs that Overlap Another Predicted CDS:
  This is the number of predicted CDSs that overlap another predicted CDS by at least one nucleotide base.
- Percentage Difference of Overlapping Predicted CDSs:
  This is the Percentage Difference of overlapping predicted CDSs as compared to the number of overlapping reference Genes. *100 * (Number of overlapping CDSs - Number of overlapping reference genes) / Number of overlapping reference genes*
- Maximum Length of Predicted CDS Overlap:
  This is the maximum length of predicted CDS overlap, in nucleotides.
- Median Length of Predicted CDS Overlap:
  This is the median length of predicted CDS overlap calculated from all CDS overlap lengths.
- Number of Matched Predicted CDSs Overlapping Another Predicted CDS:
  This is the number of predicted CDSs that detected a reference gene that overlap another predicted CDS by at least one base.

- Percentage Difference of Matched Overlapping Predicted CDSs: **(M8)**
  This is the percentage difference of overlapping CDSs that detected a reference gene as compared to the number of overlapping annotated reference genes. *100 * (Number of matched overlapping CDSs - Number of overlapping reference genes) / Number of overlapping reference genes*
- Maximum Length of Matched Predicted CDS Overlap:
  This is the maximum length of matched predicted CDS overlap, in nucleotides.
- Median Length of Matched Predicted CDS Overlap:
  This is the median length of matched predicted CDS overlap calculated from all predicted CDS overlap lengths.
- Number of Short Predicted CDSs:
  This is the number of predicted CDSs that are under 100 nucleotide bases.
- Percentage Difference of Short Predicted CDSs:
  This is the percentage difference of predicted short CDSs as compared to the number of annotated reference short genes (short defined as less than 100 nucleotide bases). *100 * (Number of short CDSs - Number of short genes) / Number of short genes*
- Number of Matched Short Predicted CDSs:
  This is the number of CDSs which detected a reference gene and that are under 100 nucleotide bases.
- Percentage Difference of Matched Short Predicted CDSs: **(M9)**
  This is the percentage difference of short CDSs which detected a gene as compared to the number of reference short genes. *100 * (Number of s short matched CDSs - Number of short genes) / Number of short genes*
- Number of Perfect Matches: **(M5)**
  This is the number of predicted CDSs that have correctly identified the exact start and stop position of a reference gene.
- Percentage of Perfect Matches:
  This is the percentage of CDSs that have correctly identified the exact start and stop position of a reference gene. *100 * Number of CDSs which matched a reference gene - Number of reference genes) / Number of reference genes*
- Number of Perfect Starts:
  This is the number of Matched CDSs that have correctly identified the start position of a reference gene.
- Percentage of Perfect Starts:
  This is the percentage of Matched predicted CDSs that have correctly identified a reference gene and its start position.

- Number of Perfect Stops:
  This is the number of matched predicted CDSs that have correctly identified the stop position of a reference gene.
- Percentage of Perfect Stops:
  This is the percentage of matched CDSs that have correctly identified a reference gene and its stop position.
- Number of Out of Frame Predicted CDSs:
  This is the number of predicted CDSs that covered more than 75% of a reference gene but were out of frame, thus classified as Unmatched.
- Number of Matched Predicted CDSs Extending a Coding Region:
  This is the number of matched predicted CDSs that extend the 3 and 5-prime end of its detected reference gene.
- Percentage of Matched Predicted CDSs Extending a Coding Region:
  This is the percentage of matched CDSs that extend the 3 and 5-prime end of its detected reference gene.
- Number of Matched Predicted CDSs Extending Start Region:
  This is the number of matched predicted CDSs that extend the 5-prime end of its detected reference gene.
- Percentage of Matched Predicted CDSs Extending Start Region:
  This is the percentage of matched CDSs that extend the 5-prime end of its detected reference gene.
- Number of Matched Predicted CDSs Extending Stop Region:
  This is the number of matched CDSs that extend the 3-prime end of its detected reference gene.
- Percentage of Matched Predicted CDSs Extending Stop Region:
  This is the percentage of matched CDSs that extend the 3-prime end of its detected reference gene.
- Number of All Predicted CDSs on Positive Strand:
  This is the number of all predicted CDSs on the positive strand.
- Percentage of All Predicted CDSs in Positive Strand:
  This is the percentage of all predicted CDSs on the positive strand.
- Number of All Predicted CDSs in Negative Strand:
  This is the number of all predicted CDSs on the negative strand.
- Percentage of All Predicted CDSs in Negative Strand:
  This is the percentage of all predicted CDSs on the negative strand.
- Median Start Difference of Matched Predicted CDSs: **(M6)**:
  This is the median difference calculated by taking all matched predicted CDSs

start position differences from the detected reference genes and finding the median of these differences. This is calculated in nucleotides and the closer to 0, the lower the difference or effective error.

- Median Stop Difference of Matched Predicted CDSs: **(M7)**
  This is the median difference calculated by taking all matched predicted CDSs stop position differences from the detected reference genes and finding the median of these differences. This is calculated in nucleotides and the closer to 0, the lower the difference or effective error.
- ATG Start Percentage:
  This is the percentage of all predicted CDSs which begin with the ATG codon.
- GTG Start Percentage:
  This is the percentage of all predicted CDSs which begin with the GTG codon.
- TTG Start Percentage:
  This is the percentage of all predicted CDSs which begin with the TTG codon.
- ATT Start Percentage:
  This is the percentage of all predicted CDSs which begin with the ATT codon.
- CTG Start Percentage:
  This is the percentage of all predicted CDSs which begin with the CTG codon.
- Other Start Codon Percentage:
  This is the percentage of all predicted CDSs which begin with an alternative start codon.
- TAG Stop Percentage:
  This is the percentage of all predicted CDSs which end with the TAG codon.
- TAA Stop Percentage:
  This is the percentage of all predicted CDSs which end with the TAA codon.
- TGA Stop Percentage:
  This is the percentage of all predicted CDSs which end with the TGA codon.
- Other Stop Codon Percentage:
  This is the percentage of all predicted CDSs which end with an alternative stop codon.
- True Positive:
  The true positive value is calculated by dividing the number of reference genes correctly detected by the total number of reference genes (75% detected and in frame). *Number of reference CDSs detected / Number of reference genes*
- False Positive:
  The false positive value is calculated by dividing the number of predicted

CDSs which did not match any reference genes by the total number of reference genes.*Number of unmatched CDSs / Number of reference genes*

- False Negative:
  The false negative value is calculated by dividing the number of reference genes missed by the predicted CDSs by the total number of reference genes.
- Precision: **(M10)**
  The precision value is calculated by dividing the true positive value by the sum of the true positive and false positive values.
- Recall: **(M11)**
  The recall value is calculated by dividing the true positive value by the sum of the true positive and false negative values.
- False Discovery Rate: **(M12)**
  The false discovery rate is calculated by dividing the false positive value by the sum of the false positive and true positive values.
- True Positive (Nucleotide):
  The true positive value is calculated by dividing the number of nucleotides in reference genes correctly detected by the total number of nucleotides in all reference genes.
- False Positive (Nucleotide):
  The false positive value is calculated by dividing the number of nucleotides in predicted CDSs but not in any reference genes by the total number of nucleotides not in any reference genes.
- True Negative (Nucleotide):
  The true negative value is calculated by dividing the number of nucleotides not in any predicted CDSs by the number of nucleotides not in any reference genes.
- False Negative (Nucleotide):
  The false negative value is calculated by dividing the number of nucleotides in reference genes but not in predicted CDSs by the total number of nucleotides in all reference genes.
- Precision (Nucleotide):
  This precision value is calculated by dividing the nucleotide true positive value by the sum of the nucleotide true positive and false positive values.
- Recall (Nucleotide):
  This recall value is calculated by dividing the nucleotide true positive value by the sum of the nucleotide true positive and false negative values.

- False Discovery Rate (Nucleotide):
  This false discovery rate is calculated by dividing the nucleotide false positive value by the sum of the nucleotide false positive and true positive values.
- Predicted CDS Nucleotide Coverage of Genome:
  This is the percentage of nucleotides in all predicted CDSs out of all nucleotides in the genome.
- Correctly Matched CDS Nucleotide Coverage of Genome:
  This is the percentage of nucleotides in Matched CDSs which correctly detected a reference gene out of all nucleotides in the genome.

Appendix B

Chapter 3 Appendix

B.1 UR_Extractor User Menu

```
usage: UR_Extractor.py [-h] -f FASTA -gff GFF [-ident IDENT] [-min_len MINLEN]
                       [-max_len MAXLEN] [-ex_len EXLEN]
                       [-gene_ident GENE_IDENT] -o OUT_PREFIX
                       [-gz {True,False}]

optional arguments:
  -h, --help            show this help message and exit
  -f FASTA, --fasta_seq FASTA
                        FASTA file for Unannotated Region seq extraction
  -gff GFF              GFF annotation file for the FASTA
  -ident IDENT          Identifier given for Unannotated Region output
                        sequences: Default "Input"_UR
  -min_len MINLEN       Minimum UR Length: Default 30
  -max_len MAXLEN       Maximum UR Length: Default 100,000
  -ex_len EXLEN         UR Extension Length: Default 50
  -gene_ident GENE_IDENT
                        Identifier used for extraction of "genic" regions
                        "CDS,rRNA,tRNA": Default for Ensembl_Bacteria =
                        "ID=gene"
  -o OUT_PREFIX, --output_prefix OUT_PREFIX
                        Output file prefix - Without filetype
  -gz {True,False}      Default - False: Output as .gz
```

FIGURE B.1: Command line menu for UR_Extractor.py

B.2 StORF_Finder User Menu

```
usage: StORF_Finder.py [-h] -seq SEQ [-ua {True,False}] [-wc {True,False}]
                       [-ps {True,False}] [-filt [{none,soft,hard}]]
                       [-aa {True,False}] [-con_storfs {True,False}]
                       [-aa_only {True,False}] [-con_only {True,False}]
                       [-stop_ident {True,False}] [-minorf MIN_ORF]
                       [-maxorf MAX_ORF] [-codons STOP_CODONS]
                       [-olap OVERLAP_NT] [-gff {True,False}] [-o OUT_PREFIX]
                       [-lw {True,False}] [-gz {True,False}] [-v {True,False}]

StORF Run Parameters.

optional arguments:
  -h, --help            show this help message and exit
  -f FASTA              Input FASTA File
  -ua {True,False}      Default - Treat input as Unannotated: Use "-ua False"
                        for standard fasta
  -wc {True,False}      Default - False: StORFs reported across entire
                        sequence
  -ps {True,False}      Default - False: Partial StORFs reported
  -filt [{none,soft,hard}]
                        Default - Hard: Filtering level none is not
                        recommended, soft for single strand filtering and hard
                        for both-strand longest-first tiling
  -aa {True,False}      Default - False: Report StORFs as amino acid sequences
  -con_storfs {True,False}
                        Default - False: Output Consecutive StORFs
  -aa_only {True,False}
                        Default - False: Only output Amino Acid Fasta
  -con_only {True,False}
                        Default - False: Only output Consecutive StORFs
  -stop_ident {True,False}
                        Default - True: Identify Stop Codon positions with '*'
  -minorf MIN_ORF       Default - 100: Minimum StORF size in nt
  -maxorf MAX_ORF       Default - 50kb: Maximum StORF size in nt
  -codons STOP_CODONS   Default - ('TAG,TGA,TAA'): List Stop Codons to use
  -olap OVERLAP_NT      Default - 50: Maximum number of nt of a StORF which
                        can overlap another StORF.
  -gff {True,False}     Default - True: StORF Output a GFF file
  -o OUT_PREFIX         Default - False/Same as input name with '_StORF-R':
                        Output filename prefix - Without filetype
  -lw {True,False}      Default - False: Line wrap FASTA sequence output at 60
                        chars
  -gz {True,False}      Default - False: Output as .gz
  -v {True,False}       Default - False: Print out runtime status
```

FIGURE B.2: Command line menu for StORF_Finder.py

B.3 Gene Clustering with CD-Hit

```
cd-hit -i Escherichia\_coli\_PEP.fa -o
    Escherichia\_coli\_PEP.fa\_CD\_c90\_s60 -s 0.6 -c 0.9 -sc 1 -sf 1 -p 1
    -g 1 -d 0 -M 10000 -T 8
```

LISTING B.1: CD-Hit gene clustering was performed with the listed parameters.

Appendix C

Chapter 5 Appendix

C.1 Read Trimming

```
java -jar /home/nick/Software/Trimmomatic-0.39/trimmomatic-0.39.jar PE
    -trimlog trim_log.txt SRR873595_1.fastq.gz SRR873595_2.fastq.gz
    trimmed_paired_SRR873595_1.fastq.gz
    trimmed_unpaired_SRR873595_1.fastq.gz
    trimmed_paired_SRR873595_2.fastq.gz
    trimmed_unpaired_SRR873595_2.fastq.gz ILLUMINACLIP:TruSeq3-PE.fa:2:30:10
    LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36 -threads 8
```

LISTING C.1: Trimmomatic parameters used to pair-end join the reads from the two trimmed fastq files

Mycobacterium_tuberculosis_gca_000736195.ASM73619v1	Borrelia_garinii_pbr.ASM17227v2	Chryseobacterium_antarcticum.ASM72998v1	Thioalkalivibrio_sulfidiphilus_hl_ebgr7.ASM2198v1
Actinomyces_odontolyticus_atcc_17982.ASM15422v1	Prochlorococcus_sp_mit_0604.ASM75784v1	Methanosarcina_barkeri_3.ASM97030v1	Acidithiobacillus_ferrooxidans_atcc_23270.ASM2148v1
Helicobacter_pylori_gca_001653435.ASM165343v1	Geobacter_bemidjiensis_bem.ASM2072v1	Sphingomonas_panacis.ASM171795v1	Saccharomonospora_glauca_k62.ASM24339v3
Riemerella_anatipestifer_ch3.ASM73405v1	Listeria_monocytogenes_gca_000950775.ASM95077v1	Candidate_division_tm6_bacterium_gw2011_gwf2_43_87.ASM100006v1	Selenomonas_sp_oral_taxon_920.ASM171758v1
Staphylococcus_aureus_m0239.Stap_aure_M0239_V1	Rickettsia_bellii_str_rml_an4.ASM96500v1	Coxiella_like_endosymbiont.ASM107771v1	Clostridioides_difficile_atcc_9689_dsm_1296.ASM107753v1
Nostoc_piscinale_cena21.ASM129844v1	Arthrobacter_sp_eprs66.ASM151227v1	Leuconostoc_mesenteroides_subsp_dextranicum.ASM104769v1	Erwinia_sp_ejp617.ASM16581v1
Hymenobacter_swuensis_dy53.ASM57655v1	Vibrio_cholerae_o1_str_2010el_1786.ASM16645v2	Myroides_odoratimimus_ccug_3837.Myro_odor_CCUG_3837_V1	Ureaplasma_urealyticum_serovar_8_str_atcc_27618.ASM16953v1
Moraxella_bovoculi_gca_000988685.ASM98868v1	Bordetella_holmesii_1058.gbhBH006	Methanococcus_vannielii_sb.ASM1716v1	Paraburkholderia_phenoliruptrix_br3459a.ASM30009v1
Paenibacillus_mucilaginosus_knp414.ASM21891v1	Flavobacterium_psychrophilum_jip02_86.ASM6430v2	Stenotrophomonas_maltophilia_k279a.ASM7248v1	Nocardia_farcinica.NCTC11134
Alteromonas_australica_gca_000934525.ASM93452v1	Bacillus_cereus_bag6x1_1.Baci_cere_BAG6X1-1_G13210_V1	Mesorhizobium_ciceri.ASM161882v1	Collimonas_fungivorans.ASM158414v1
Streptococcus_agalactiae_gca_000831105.ASM83110v1	Acinetobacter_oleivorans_dr1.ASM19679v1	Pyrobaculum_neutrophilum_v24sta.ASM1980v1	Azospirillum_sp_b510.ASM1072v1
Klebsiella_pneumoniae_uhkpc05_gca_000709245.UHKPC05v01	Treponema_denticola_aslm.Trep_dent_ASLM_V1	Aeromonas_hydrophila_j_1.ASM81950v1	Borreliella_valaisiana_vs116.ASM17095v2
Serratia_marcescens_sm39.ASM82877v1	Nitrosomonas_communis.ASM100793v1	Ruminococcus_torques_l2_14.ASM21003v1	Alcanivorax_pacificus_w11_5.ASM29933v2
Hyphomicrobium_sp_mc1.ASM25329v1	Campylobacter_jejuni_subsp_jejuni_nctc_11168_k12e5.ASM83077v1	Eubacterium_sp_14_2.Euba_bact_14-2_V1	Altererythrobacter_marensis.ASM102862v1
Corynebacterium_camporealensis_gca_000980815.ASM98081v1	Thermococcus_piezophilus.ASM164708v1	Wolbachia_endosymbiont_of_drosophila_simulans_wha.ASM37660v1	Thermus_oshimai_jl_2.ASM30988v1
Buchnera_aphidicola_str_usda_myzus_persicae_.ASM52152v1	Anaplasma_phagocytophilum_str_apmuc09.ASM96474v1	Spiroplasma_atrichopogonis.ASM102924v1	Brachyspira_hampsonii_30446.ASM31619v1
Caldicellulosiruptor_obsidiansis_ob47.ASM14521v1	Sulfolobus_acidocaldarius_n8.ASM34031v1	Desulfotomaculum_kuznetsovii_dsm_6115.ASM21470v1	Archaeoglobus_fulgidus_dsm_8774.ASM73403v1
Candidatus_kinetoplastibacterium_oncopeltii_tcc290e.ASM34086v1	Actinobacillus_pleuropneumoniae_serovar_7_str_ap76.ASM2040v1	Synechocystis_sp_pcc_6803.ASM972v1	Gardnerella_vaginalis_6420lit.ASM26345v1
Xanthomonas_campestris_pv_raphani_756c.ASM22196v1	Haemophilus_somnus_129pt.ASM1178v1	Legionella_longbeachae_nsw150.ASM9178v1	Methanocaldococcus_sp_fs406_22.ASM2552v1
Sphingobium_japonicum_ut26s.ASM9112v1	Planococcus_sp_l10_15.ASM118615v2	Synechococcus_sp_cc9311.ASM1458v1	Clavibacter_michiganensis_subsp_sepedonicus.ASM6922v1
Salmonella_enterica_subsp_enterica_serovar_paratyphi_a_gca_000985715.PA032	Pseudoalteromonas_sp_sm9913.ASM18406v1	Pasteurella_multocida_subsp_multocida_str_hn06.ASM25591v1	Rhodopseudomonas_palustris_tie_1.ASM2044v1
Prevotella_denticola_f0289.ASM19339v1	Cronobacter_sakazakii_gca_000982825.ASM98282v1	Weissella_ceti_gca_000750515.ASM75051v1	Bradyrhizobium_oligotrophicum_s58.ASM34480v1
Shewanella_pealeana_atcc_700345.ASM1828v1	Citrobacter_koseri_gca_000783445.ASM78344v1	Clostridium_botulinum_cdc_1436.ASM81793v1	Aerococcus_sanguinicola.ASM154314v1
Amycolatopsis_methanolica_239.ASM73908v1	Enterococcus_casseliflavus_ec20.ASM15735v2	Lysobacter_capsici.ASM144278v1	Leptolyngbya_sp_nies_3755.ASM154843v1
Chlamydia_trachomatis_l1_115.ASM31884v1	Thermotoga_maritima_gca_000978535.ASM97853v1	Pectobacterium_sp_scc3193.ASM26092v1	Cupriavidus_necator_n_1.ASM21921v1
Brucella_abortus_ni622.Bruc_abor_NI622_V1	Pandoraea_sputorum.ASM81484v2	Thermoanaerobacter_sp_ys13.ASM80622v2	Methanobrevibacter_smithii_atcc_35061.ASM1652v1
Geobacillus_sp_js12.ASM159239v1	Lactobacillus_plantarum_jdm1.ASM2308v1	Methylobacterium_nodulans_ors_2060.ASM2208v1	Taylorella_asinigenitalis_mce3.ASM22662v1
Burkholderia_pseudomallei_gca_000756145.ASM75614v1	Frankia_symbiont_of_datisca_glomerata.ASM17761v2	Ralstonia_pickettii_12d.ASM2342v1	Chlamydophila_pneumoniae_ar39.ASM9108v1
Sinorhizobium_meliloti_2011.ASM34606v1	Dehalococcoides_mccartyi_btf08.ASM34169v1	Zymomonas_mobilis_subsp_mobilis_zm4_atcc_31821.ASM710v1	Sphingopyxis_macrogoltabida.ASM131432v1
Aeromicrobium_sp_leaf291.Leaf291	Yersinia_pestis_1670.DYA_1	Deinococcus_proteolyticus_mrp.ASM19055v1	Aggregatibacter_actinomycetemcomitans_hk1651.ASM60404v1
Shigella_flexneri_4c.ASM157996v1	Bifidobacterium_angulatum_dsm_20098_jcm_7096_gca_001025155.ASM102515v1	Marinobacter_adhaerens_hp15.ASM16629v1	Veillonella_parvula_dsm_2008.ASM2494v1
Escherichia_coli_abu_83972.ASM14836v1	Desulfovibrio_africanus_str_walvis_bay.ASM21267v2	Edwardsiella_sp_ladl05_105.ASM118621v1	Cyanothece_sp_pcc_7822.ASM14733v1
Chlorobium_phaeobacteroides_dsm_266.ASM1512v1	_clostridium_acidurici_9a.ASM29935v1	Microbacterium_sp_xt11.ASM151367v1	Janthinobacterium_agaricidamnosum_nbrc_102515_dsm_9628.JAG1
Pseudomonas_sp_m1.M1_v2.0	Enterobacter_aerogenes_ea1509e.ASM33451v1	Agrobacterium_tumefaciens_lba4213_ach5_.ASM57651v1	Pyrococcus_abyssi_ge5.ASM19593v2
Achromobacter_xylosoxidans_gca_001558755.ASM155875v1	Rhizobium_leguminosarum_bv_trifolii_wsm1325.ASM2318v1	Caulobacter_segnis_atcc_21756.ASM9228v1	Cellulophaga_baltica_18.ASM46861v2
Blattabacterium_sp_blatta_orientalis_str_tarazona.ASM33440v1	Psychrobacter_alimentarius.ASM160602v1	Pseudonocardia_sp_al041005_10.ASM129460v1	Erythrobacter_sp_nap1.ASM15286v1
Lactococcus_lactis_subsp_lactis_klds_4_0325.ASM47937v2	Rhodobacter_capsulatus_sb_1003.ASM2186v1	Porphyromonas_gingivalis_w83.ASM758v1	Providencia_stuartii.ASM75434v1
Streptomyces_lividans_tk24_gca_000739105.ASM73910v1	Rhodococcus_opacus.ASM73643v1	Cedecea_neteri_gca_000757825.ASM75782v1	Leifsonia_sp_root1293.Root1293
Mycoplasma_fermentans_jer.ASM14862v1	Bibersteinia_trehalosi_usda_ars_usmarc_192.ASM34759v1	Oenococcus_oeni_psu_1.ASM1438v1	Arcobacter_butzleri_ed_1.ASM28435v1
Micromonospora_lupini_str_lupac_08.ASM29739v2	Acetobacter_pasteurianus_subsp_ascendens.ASM176623v1	Bacteroides_thetaiotaomicron.ASM131497v1	Acidovorax_sp_js42.ASM1554v1
Ehrlichia_chaffeensis_str_osceola.ASM63290v1	Neisseria_meningitidis_nz_05_33.ASM19152v1	Bdellovibrio_bacteriovorus_hd100.ASM19617v1	Dickeya_zeae_ec1.ASM81604v1
Proteus_mirabilis_hi4320.ASM6996v1	Mannheimia_haemolytica_gca_000963635.ASM96363v1	Pediococcus_pentosaceus_atcc_25745.ASM1450v1	Myxococcus_stipitatus_dsm_14675.ASM33173v1
Francisella_tularensis_subsp_tularensis_ti0902.ASM24843v1	Halomonas_chromatireducens.ASM154515v1	Bartonella_tamiae_th307.Bart_tami_Th307_V1	Desulfitobacterium_hafniense_dcb_2.ASM2192v1
Fusobacterium_nucleatum_subsp_polymorphum.ASM145512v1	Pantoea_sp_psnih2.ASM78496v1	Hydrogenobaculum_sp_sho.ASM21506v1	Xenorhabdus_poinarii_g6.ASM96817v1
Leptospira_interrogans_serovar_manilae_gca_001047655.ASM104765v1	Propionibacterium_freudenreichii_subsp_shermanii.PFREUDJS1	Methanobacterium_congolense.MCBB	Xylella_fastidiosa_9a5c.ASM672v1

TABLE C.1: Listed are the 179 genomes selected from each of the 179 genera to be used as training data.

C.2 Training Data

Appendix D

Published Papers:

1. Dimonaco, N. J., Aubrey, W., Kenobi, K., Clare, A., & Creevey, C. J. (2021). No one tool to rule them all: Prokaryotic gene prediction tool annotations are highly dependent on the organism of study. *Bioinformatics*, 2021;, btab827, https://doi.org/10.1093/bioinformatics/btab827.

2. Liu-Wei, W., Kafkas, S., Chen, J., Dimonaco, N. J., Tegnér, J., & Hoehndorf, R. (2021). DeepViral: prediction of novel virus-host interactions from protein sequences and infectious disease phenotypes. *Bioinformatics*, Volume 37, Issue 17, 1 September 2021, Pages 2722–2729, https://doi.org/10.1093/bioinformatics/btab147.

3. Dimonaco, N. J., Salavati, M., & Shih, B. B. (2021). Computational analysis of SARS-CoV-2 and SARS-like coronavirus diversity in human, bat and pangolin populations. Viruses, 13(1), 49. https://doi.org/10.3390/v13010049.

Bioinformatics, 2021, 1–10
https://doi.org/10.1093/bioinformatics/btab827
Advance Access Publication Date: 7 December 2021
Original Paper

Genome analysis

No one tool to rule them all: prokaryotic gene prediction tool annotations are highly dependent on the organism of study

Nicholas J. Dimonaco [1,*], Wayne Aubrey [2], Kim Kenobi [3], Amanda Clare [2,†] and Christopher J. Creevey [4,†]

[1]Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, Aberystwyth SY23 3PD, UK, [2]Department of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, UK, [3]Department of Mathematics, Aberystwyth University, Aberystwyth SY23 3BZ, UK and [4]School of Biological Sciences, Queen's University Belfast, Belfast BT7 1NN, UK

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors.
Associate Editor: Tobias Marschall

Received on May 26, 2021; revised on November 13, 2021; editorial decision on November 30, 2021; accepted on December 2, 2021

Abstract

Motivation: The biases in CoDing Sequence (CDS) prediction tools, which have been based on historic genomic annotations from model organisms, impact our understanding of novel genomes and metagenomes. This hinders the discovery of new genomic information as it results in predictions being biased towards existing knowledge. To date, users have lacked a systematic and replicable approach to identify the strengths and weaknesses of any CDS prediction tool and allow them to choose the right tool for their analysis.

Results: We present an evaluation framework (ORForise) based on a comprehensive set of 12 primary and 60 secondary metrics that facilitate the assessment of the performance of CDS prediction tools. This makes it possible to identify which performs better for specific use-cases. We use this to assess 15 *ab initio*- and model-based tools representing those most widely used (historically and currently) to generate the knowledge in genomic databases. We find that the performance of any tool is dependent on the genome being analysed, and no individual tool ranked as the most accurate across all genomes or metrics analysed. Even the top-ranked tools produced conflicting gene collections, which could not be resolved by aggregation. The ORForise evaluation framework provides users with a replicable, data-led approach to make informed tool choices for novel genome annotations and for refining historical annotations.

Availability and implementation: Code and datasets for reproduction and customisation are available at https://github.com/NickJD/ORForise.

Contact: nicholas@dimonaco.co.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Whole genome sequencing, assembly and annotation is now widely conducted, due predominantly to the increase in affordability, automation and throughput of new technologies (Land *et al.*, 2015). The prediction of protein-coding genes, specifically their corresponding CoDing Sequence (CDS) in prokaryote genomes has often been seen as an established routine. This is in part due to a number of assumptions and features, such as the high density (protein-coding genes contribute ~80–90% of prokaryote DNA) and the lack of introns (Lobb *et al.*, 2020; Salzberg, 2019). However, this process involves the complex identification of a number of specific elements, such as: promoter regions (Browning and Busby, 2004), the Shine–Dalgarno (Dalgarno and Shine, 1973) ribosomal binding site and operons (Dandekar *et al.*, 1998), which all contribute to identifying gene position and order. Additionally, the role of horizontal gene transfer (Jain *et al.*, 1999) and pangenomes further complicates an already difficult process and likely contributes to errors and a lack of data held in public databases (Devos and Valencia, 2001; Furnham *et al.*, 2012). Finally, our ability to characterize the functions of regions of DNA [which has been generally reserved for model organisms

(MOs) and core genes (Russell *et al.*, 2017)] is being outstripped by the rate of genomic and metagenomic sequence data generation from non-MOs and non-core gene DNA sequences.

Before the turn of the century, it was understood that a great deal of work was still needed to address these issues. Studies had shown that many existing CDS prediction tools systematically failed to identify or accurately report genes whose features lay outside a rigid set of rules, such as non-standard codon usage, those which overlap other genes or those below a specified length (Burge and Karlinb, 1998; Guigo, 1997). Since then, a systematic overview of 1474 prokaryotic genome annotations in GenBank concluded 'the cause of the high rates of missed genes is less clear, largely due to a lack of information about the annotation methods used' (Wood *et al.*, 2012). Interestingly, while the majority of missed genes reported were under 300 nt, the annotation tools, which performed the incomplete annotations were developed to report CDSs at a minimum length of 110 nt. While there has been much work to address the problem of incomplete annotation, many gene types continue to be absent or under-represented in public databases (Huvet and Stumpf, 2014; Warren *et al.*, 2010), such as short/small-ORFs (short ORFs) (Duval and Cossart, 2017; Storz *et al.*, 2014; Su *et al.*, 2013). This means that CDS prediction methodologies that use information from existing sequences are in turn ill-equipped to identify genes belonging to these underrepresented/missing gene types. It is therefore of paramount importance that we understand the limits of current CDS predictors as our reliance on automated genome annotation of novel genomes continues to increase (Brenner, 1999). Measures to compare both novel and contemporary CDS prediction tools are not well established or universally employed and novel tool descriptions tend to focus on algorithmic improvements rather than carrying out a systematic assessment of where the strengths or weaknesses in their approaches lie. This prevents researchers from gaining meaningful insight into the specific features of genes, which led to them being missed or partially detected, resulting in a lost opportunity to improve our understanding of prokaryote genome content.

Genome annotation is challenging and is not a single step process. CDS prediction, often the first step, is fast, with little user input, but may require augmentation by different methods to supplement the initial predictions. One example is a tool, such as smORFer (Bartholomäus *et al.*, 2021), that specializes in finding short ORFs through the use of RNA-seq, which can detect transcription events under certain environmental conditions. Further examples use sequence conservation scores and homology searches that can use existing database knowledge (Badger and Olsen, 1999; Dunne and Kelly, 2017; ÓhÉigeartaigh *et al.*, 2014). Furthermore, pipelines are constructed [such as PROKKA (Seemann, 2014) and NCBI's PGAP (Tatusova *et al.*, 2016)] to automate these further rounds of annotation. However, the underlying CDS prediction tools are still core components of these pipelines and are still widely used as standalone tools.

Previous studies, which have evaluated prokaryotic CDS predictors generally only compared a small number of tools, focussing on algorithm design, and did not go into depth when reporting prediction accuracy with few other informative metrics used (Al-Turaiki *et al.*, 2011; Mathé *et al.*, 2002). A more recent study, BEACON (Kalkatawi *et al.*, 2015), considered a small range of metrics including genes 'denoted as identical, similar, unique with overlap or unique without overlap' to either a reference annotation or from the output of three pipelines (PGAP, AAMG and RAST). Unfortunately, the types of genes missed were also not investigated further, leading to a lack of understanding of not only why and how they were missed, but also the impact on our biological understanding of the genome as a whole.

Many prediction methods used today are iterations of original concepts and thus are as in flux as the genomic databases themselves. Future development of CDS prediction techniques is now harnessing the recent advances in machine learning and other computational methods. While previous methods involve the construction of models built from organism-specific parameters, such as codon usage, guanine-cytosine (GC), complex motifs and average CDS length (Besemer and Borodovsky, 1999; Stanke and Morgenstern, 2005), opinions are shifting on the use and importance of MOs (Hunter, 2008; Levy and Currie, 2015; Russell *et al.*, 2017). The volume of prokaryotic protein-coding gene sequences have enabled advanced machine-learning approaches, such as neural networks to predict CDSs that share common characteristics with a selection of previously annotated genes. One such example, Balrog (Sommer and Salzberg, 2021) predicts protein-coding genes by training from an array of non-hypothetical protein-CDSs from thousands of bacterial prokaryote genomes and aims to provide gene prediction across diverse species. Machine-learning models can be poor at making predictions for classes (e.g. genes) whose training data exhibit high levels of bias, error, are under-represented for specific groups (e.g. gene families) and groups for which they have not been trained (Schafer and Graham, 2002). In addition to this, prokaryotic gene families are chronically under-sampled (Warren *et al.*, 2010). It is becoming clear that even with these advances in computational approaches, it is unlikely that we will ever be capable of identifying the complete picture of CDS gene diversity without exhaustive experimental work.

To address these concerns, we extensively evaluate a collection of 15 widely used CDS prediction tools that form the basis of most of the annotations deposited in public databases and therefore have largely been used to build the genomic knowledge used by the scientific community. We provide a comparison platform developed to allow researchers to compare 12 primary and a further 60 secondary metrics to systematically compare the predictions from these tools and study the effect on the resulting genome annotations for their species of interest. This allows for in-depth and reproducible analyses of aspects of gene prediction that are often not investigated and allows researchers to understand the impact of tool choice on the resulting prokaryotic gene collection.

2 Materials and methods

2.1 Current Ensembl genome annotations

Six bacterial MOs and their canonical annotations were downloaded from Ensembl Bacteria (Howe *et al.*, 2020) (available at https://github.com/NickJD/ORForise/tree/master/Genomes). *Bacillus subtilis* BEST7003 strain (assembly ASM52304v1), *Caulobacter crescentus* CB15 strain (assembly ASM690v1), *Escherichia coli* K-12 ER3413 strain (assembly ASM80076v1), *Mycoplasma genitalium* G37 strain (assembly ASM2732v1), *Pseudomonas fluorescens* UK4 strain (assembly ASM73042v1) and *Staphylococcus aureus* 502A strain (assembly ASM59796v1) were chosen for their scientific importance, range of genome size, GC content, assumed near complete and high quality genome assembly and annotation provided by Ensembl Bacteria. They are presented in Table 1 and further information regarding these MOs can be found in Supplementary Section S1.

For each of the chosen MOs, two data files were downloaded from Ensembl Bacteria; the complete DNA sequence (**_dna.toplevel.fa*) and the general feature format (GFF) file (**.gff3*) containing the position of each gene. The current collection of CDS genes presented in the MO annotations from Ensembl [Current Ensembl Annotation (CEA)] was taken as the reference annotations for this study. Prokaryotic genomes exhibit high levels of gene density, often with little extraneous DNA, which is 'commonly perceived as evidence of adaptive genome streamlining' (Sela *et al.*, 2016). Unannotated DNA represents between ~10% and 20% of the six MO genomes selected and while an additional 0.38–2.22% is attributed to non-coding annotations, there is still a measurable portion of each genome without any annotation. This study focuses specifically on the identification of CDSs, which constitute the significant majority of annotated genomic regions in the six genomes studied (82.76–90.62%, see Table 1).

The CDSs from each of the six genomes exhibit a range of differences, which are known to impact the ability of prediction tools to identify them. These include, but are not limited to, GC content, codon usage and gene length. The GC content varies from 31.69% to 67.21% for these genomes, and even within a single genome, the CDS GC content varies widely (see Supplementary Fig. S1 for

Table 1. An overview of genome composition for the six MOs selected to evaluate CDS prediction tools compiled from data held by Ensembl bacteria

Model organism [assembly]	Genome size (Mbp)	Genes [CDSs]	Genome density [CDSs]	GC content (%)
B.subtilis BEST7003 **[ASM52304v1]**	4.04	4133 **[4011]**	88.91% **[87.60%]**	43.89
C.crescentus CB15 **[ASM690v1]**	4.02	3875 **[3737]**	90.60% **[90.23%]**	67.21
E.coli ER3413 **[ASM80076v1]**	4.56	4257 **[4052]**	86.28% **[84.35%]**	50.80
M.genitalium G37 **[ASM2732v1]**	0.58	559 **[476]**	92.03% **[90.62%]**	31.69
P.fluorescens UK4 **[ASM73042v1]**	6.06	5266 **[5178]**	84.75% **[84.20%]**	60.13
S.aureus 502A **[ASM59796v1]**	2.76	2556 **[2478]**	83.93% **[82.76%]**	32.92

Note: Data are presented for all genes and CDS genes in bold square brackets. Note the relatively broad differences in genome size, gene density (percentage covered with annotation) and GC content.

distributions). Furthermore, the canonical ATG start codon is used between 68.58% and 90.67% of the genes for the six genomes (see Supplementary Table S1 for more detail).

Additionally, *M.genitalium* uses the codon translation table 4, meaning one of the three universal stop codons (TGA/UGA) is instead used to code for tryptophan (Dybvig and Voelker, 1996), whereas the other five MOs use the universal translation table 11 (see Supplementary Tables S1 and S3 for more detail). While a similar median CDS length is shared across the six genomes, *B.subtilis* and *P.fluorescens* have a number of long genes (>8000 nt, see Supplementary Fig. S2) and *S.aureus* contains the 31,421nt 'giant protein Ebh' (Cheng *et al.*, 2014), which is more than twice the length of the next largest CDS in this study. The diversity across the rest of prokaryotes is likely to be as great as, or greater than, reported here for these six.

The Sequence Ontology (Eilbeck *et al.*, 2005) describes an ORF as 'The in-frame interval between the stop codons of a reading frame which when read as sequential triplets, has the potential of encoding a sequential string of amino acids'. However, it is conventional for ORFs to be reported as regions of DNA encompassed by a start and stop codon as a start codon is expected to indicate the start of DNA transcription (Brent, 2005). We acknowledge the difference in ontological definition and during this study, we refer to the region of DNA between an in-frame start and stop codon that is predicted to encode for an amino acid (protein) sequence, as a predicted CDS.

2.2 Prediction tools

This study specifically investigates CDS predictors, tools which apply complex filtering after the identification of ORFs across a region of DNA. This is different to ORF finders, which return unfiltered ORFs (Stothard, 2000) that meet a set of pre-defined rules, such as length and in-frame start and stop codons. This filtering is unique to each tool and dependent on properties, such as codon usage, GC content, CDS length, overlap and similarity to known genes and other more sophisticated parameters modelled on analysis of previously studied genes and genomes. Without such filtering methods, CDS predictors would typically report many false positives, such as nested or heavily overlapping CDSs. An example of filtering can be found in in the GeneMark (Borodovsky and McIninch, 1993) algorithm, which reports multiple variations of the same CDS with confidence scores. For this study, we chose the longest for each CDS after assessing the results.

We selected 15 different CDS prediction tools, some of which required a model (a rigid set of parameters adjusted to a particular organism), and the others, which predicted *ab initio* from sequence. The tools, which required a model were: GeneMark.hmm with *E.coli* and *S.aureus* models (Lukashin and Borodovsky, 1998); FGENESB with *E.coli* and *S.aureus* models (Salamov and Solovyevand, 2011); Augustus with *E.coli*, *S.aureus* and *Homo sapiens* models (Keller *et al.*, 2011); EasyGene with *E.coli* and *S.aureus* models (Nielsen and Krogh, 2005); GeneMark with *E.coli* and *S.aureus* models (Borodovsky and McIninch, 1993). Those which did not require a model were: GeneMarkS (Besemer *et al.*, 2001); Prodigal (Hyatt *et al.*, 2010); MetaGeneAnnotator (Noguchi *et al.*, 2008); GeneMarkS-2 (Lomsadze *et al.*, 2018); MetaGeneMark (Zhu *et al.*, 2010); GeneMarkHA (Besemer and Borodovsky, 1999); FragGeneScan (Rho *et al.*, 2010); GLIMMER-3 (Delcher *et al.*, 2007); MetaGene (Noguchi *et al.*, 2006); and TransDecoder (Haas *et al.*, 2013). The two groups are referred to as 'model-based' and '*ab initio*' henceforth and can be seen in Table 2. The group *ab initio* included a number of tools, which were designed for fragmentary and metagenomic studies: MetaGeneMark, MetaGene, MetaGeneAnnotator and FragGeneScan. In addition, TransDecoder was developed to predict coding regions within transcript sequences, often in eukaryotes. To emulate the annotation process of a novel or less studied genome or metagenome, each tool was run using its default parameters. More information regarding each group and tool, and the parameters used to run them, can be found in Supplementary Section S3 'Prediction Tools'.

Whole genome annotation 'pipelines', such as PROKKA (Seemann, 2014) and NCBI's PGAP (Tatusova *et al.*, 2016) were not included, but the initial CDS prediction components embedded in these pipelines, such as Prodigal and GeneMarkS-2, were included in the study. Multiple separate tools from the GeneMark family (Besemer and Borodovsky, 2005) were included (some superseded) due to their extensive use and impact on genomic knowledge over the last three decades.

2.3 Comparison method

A systematic software platform ORForise (ORF Authorise) was built to perform a fair, comparative, and informative analysis of the different tools examining different aspects of their predictions. Version 1.0 of the platform, written in Python3 (Van Rossum and Drake, 2009), was used and is freely available at https://github.com/NickJD/ORForise. It has been designed to process the standardized GFF3 format as well as the individual output formats produced by each tool listed in this study.

In this platform, we endeavoured to choose a wide range of metrics that clearly and representatively capture the many intricacies of the predictions. A number of metrics used in previous studies, such as the number of CDSs predicted, accurate identification of start positions or the number of genes correctly detected, can give some indication of the 'accuracy' of each tool. However, it was found during our analysis that there were many complexities in the prediction results, which would not be represented by these high-level metrics. For example, predicted CDS regions may overlap with one or more known CEA genes but be inaccurately extended or truncated on either the 5' or 3' end. It is also common for smaller CEA genes to be mistakenly encompassed by larger predicted CDSs and while the nucleotide regions of these genes are technically within the predicted regions, even if in-frame, they do not represent the true protein-CDS. Furthermore, different types of inaccuracies may be more or less important, depending on the aim of any given study. Therefore, clear and specific measures of accuracy that describe the detection of the entire locus of a gene are needed. Figure 1 illustrates how we determine correct CEA gene detection, but also explains its nuances and complexities. An example of this is the definition of short ORFs, which in prokaryotes are often described as having lengths of 100–300 nt (Duval and Cossart, 2017; Storz *et al.*, 2014; Su *et al.*, 2013). However, due to hard-coded cutoffs in many of the tools, we chose the 'upper-bound' of 300 nt or 100 codons to define short

Table 2. Version number and reference for all tools used in this study

No.	Tool name	Version	Reference
1	Augustus	3.3.3	Keller *et al.* (2011)
2	EasyGene	1.2	Nielsen and Krogh (2005)
3	GeneMark.hmm	3.2.5	Lukashin and Borodovsky (1998)
4	GeneMark	2.5	Borodovsky and McIninch (1993)
5	FGENESB	'2020'	Salamov and Solovyevand (2011)
6	Prodigal	2.6.3	Hyatt *et al.* (2010)
7	GeneMarkS	4.25	Besemer *et al.* (2001)
8	GeneMarkS 2	'2020'	Lomsadze *et al.* (2018)
9	GLIMMER 3	3.02	Delcher *et al.* (2007)
10	GeneMark (H.A)	3.25	Besemer and Borodovsky (1999)
11	TransDecoder	5.5.0	Haas *et al.* (2013)
12	FragGeneScan	1.3.0	Rho *et al.* (2010)
13	MetaGene	2.24.0	Noguchi *et al.* (2006)
14	MetaGeneMark	'2020'	Zhu *et al.* (2010)
15	MetaGene Annotator	2008/8/19	Noguchi *et al.* (2008)

Note: Tools 1–5 inclusive are model-based tools. Tools 6–15 inclusive are *ab initio*-based tools. Where no version number is available, the year when the tool was used is listed in single quotes.

ORFs. We iteratively developed 72 metrics to help provide the most accurate and informative representation of a tool's prediction quality. Additionally, as part of the ORForise platform, we provide a number of Python3 post-analysis scripts developed to aid in the interrogation between the CEA gene annotations and the CDSs predicted by each of the tools studied. These scripts were used to extract characteristics that are useful in the investigation of why specific CEA genes are detected, missed or incorrectly reported.

2.4 Aggregated tool predictions

An extension to the ORForise comparison platform was built (Aggregate_Compare) to investigate whether an aggregation of predictions from a number of top-performing tools would perform better than individual tools. The CDS predictions from the selected tools are combined into a single data structure with duplicate CDSs filtered out, but alternative predictions for the same locus retained and ordered according to start position. The same comparison algorithm could then be employed on the set of unique CDS predictions identified by this union of the outputs of the selected tools (Prodigal, GeneMark-S-2, MetaGeneAnnotator, MetaGeneMark and GeneMark-S—chosen due to their individual performance) and as with the singular tool comparison, for every CEA gene, the CDS, which deviated the least from the correct locus was selected as the closest match.

2.5 Discovering additional ORFs

To enable the aggregation of different CDSs from contemporary and new annotations, we provide GFF_Intersector to create a single GFF representing the intersection of two existing annotations. This also provides an option to allow the retention of CDSs that have a user-defined difference (default minimum 75% coverage and in-frame). Additionally, we also provide the GFF_Adder tool, which produces a new GFF containing CDSs from an existing annotation, plus the new CDSs, filtered to remove any that overlap existing CDSs by more than 50 nt (user definable).

3 Results

3.1 Metrics for comparison of tools

A total of 72 different metrics were chosen for this exhaustive evaluation in order to give the broadest possible scope to compare and contrast the performance of the tools. The full definitions for each

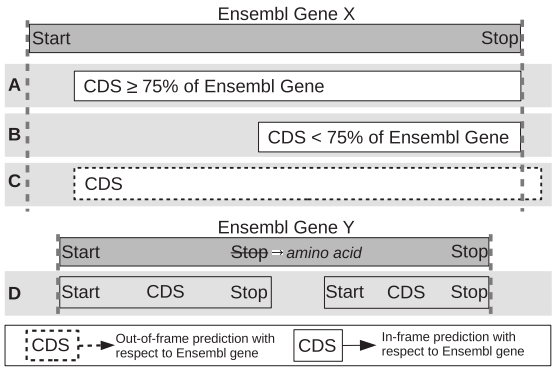


Fig. 1. Illustration of how predicted CDSs are classified as having detected or not detected the CEA genes. Predicted CDSs are compared to the genes held in Ensembl. (**A**) The predicted CDS covers at least 75% and is in-frame with Ensembl gene and therefore it is recorded as detected. (**B**) The predicted CDS covers <75% of the Ensembl gene and therefore is recorded as not detected. (**C**) The predicted CDS covers part of an Ensembl gene but is out of frame (dotted outline) and therefore is recorded as missed. (**D**) The use of alternative stop codons causes the predicted CDS to be truncated or divided into two CDSs that span the Ensembl genes and therefore is recorded as missed

of these metrics can be found in Supplementary Section S5 and are intended to be used as a resource for the community when deciding which tool to apply to both novel and contemporary genome annotation work. The following are 12 of the most informative metrics, selected for their ability to represent both a broad range and depth of different attributes which have been used to distinguish the prediction tools.

- **M1** Percentage of Genes Detected
- **M2** Percentage of Predicted CDSs that Detected a Gene
- **M3** Percentage Difference of Number of Predicted CDSs
- **M4** Percentage Difference of Median Predicted CDS Length
- **M5** Percentage of Perfect Matches
- **M6** Median Start Difference of Matched Predicted CDSs
- **M7** Median Stop Difference of Matched Predicted CDSs
- **M8** Percentage Difference of Matched Overlapping Predicted CDSs
- **M9** Percentage Difference of Matched Short Predicted CDSs
- **M10** Precision
- **M11** Recall
- **M12** False Discovery Rate

M1, Percentage of Genes Detected, is often used as the main indicator of tool performance in other comparisons but interpreted differently between studies. Here, it is characterized as a predicted CDS, which is in frame with a CEA gene and has captured at least 75% of its nucleotide sequence (Fig. 1A). In contrast to M1, which indicates when underprediction (or false negatives) occurs, M2 suggests when overprediction (or false positives) has occurred.

For M3, M4, M8 and M9, *Percentage Difference* was used to identify differences between predicted and CEA metrics: 100*(*Predicted CDS metric—Ensembl Gene Metric*)/*Ensembl Gene Metric*. The best score for a metric using the *Percentage Difference* calculation is 0, as 0 represents no deviation from the CEA annotations. The 'Matched CDSs' identifier used for M6, M7, M8 and M9 represent the CDSs, which have correctly detected an CEA gene. M6 and M7 are calculated by taking the median codon position differences recorded for mispredicted start or stop codons. Metrics, such as the Percentage of Perfect Matches (M5) can give a clearer overview of a tool's 'accuracy' or performance, as it is common for a tool to misidentify either the exact start or stop locus of a detected CEA gene, while metrics, such as Median Start Difference of Matched Predicted CDSs (M6) can help establish the level of inaccuracy.

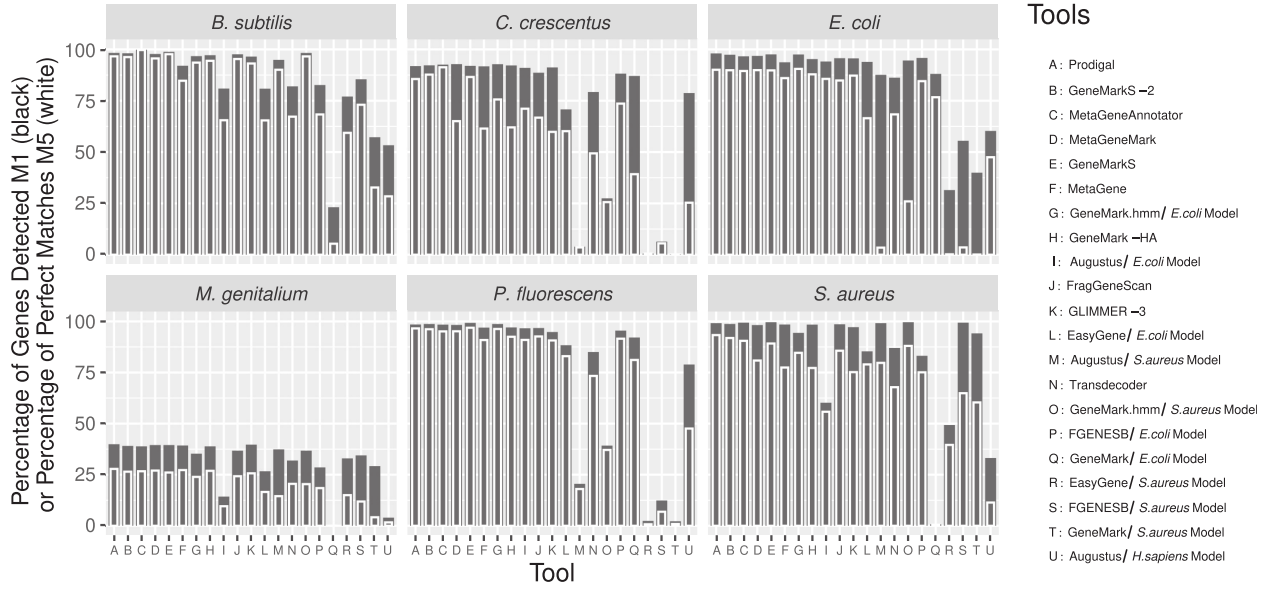


Fig. 2. The result of all 15 gene prediction tools (21 with chosen models) on the 6 MO genomes, ordered by the summed ranks across the 12 metrics. The *Y* axis represents the Percentage of Genes Detected (M1) by each tool in black and the Percentage of Perfect Matches (M5) in white. M5, which represents the ability for a tool to detect the correct start codon, has more variance between the tools than M1. Each column on the *X* axis represents a different tool (some model-based tools were run multiple times). There is considerable variation in how well each tool performs across the different genomes, while all tools perform relatively poorly on the *M.genitalium* genome

The tools were ordered by totalling the rankings for each of these 12 metrics, across the 6 MOs. Supplementary Results S1 contains the results used for the ranking. This ranking, based on a wide range of different performance measures, allows for a comparative overview of contemporary and future tools, and is presented in Figure 2. This figure also shows the Percentage of Genes Detected (M1) with an overlay of the Percentage of Perfect Matches (M5), demonstrating the inconsistency between the two metrics for each tool. Metrics, such as Percentage of Genes Detected (M1) and Percentage of Predicted CDSs that Detected a Gene (M2), are informative and can be representative of a tool's prediction quality, however, they do not convey the complete picture when presented in isolation. This is of particular importance for those working with metagenomic or other fragmentary assemblies, as the likelihood of incomplete fragments and chimeric sequences is higher and can lead to varying mispredictions. Although the overall prediction quality of genes was high across most of the tools and genomes in this study, the additional metrics produced can be used to identify strengths and weaknesses inherent to them. For example, GeneMark.hmm (*S.aureus* model and genome), MetaGeneMark and MetaGeneAnnotator, GeneMarkS were all ranked highest for Percentage of Genes Detected (M1) for at least one MO, while Prodigal and GeneMarkS were ranked highest twice (GeneMarkS and GeneMark.hmm were ranked joint highest for *S.aureus*). However, when inspecting the 12 metrics (Supplementary Fig. S3), it was clear that there were complex differences between the prediction results of not only the highest scoring tools, but also the lower ranked tools, which were often ranked high for some metrics in some of the genomes.

While no tool or group of tools consistently ranked highest or equally across the 12 metrics or MOs, MetaGeneAnnotator ranked best for *B.subtilis* and *M.genitalium*, GeneMarkS-2 ranked best for *C.crescentus* and Prodigal ranked best for *E.coli*, *P.fluorescens* and *S.aureus*.

The combination of multiple metrics can be used to determine which tool should be used between two candidate tools with the same or similar Percentage of Genes Detected (M1). For *M.genitalium*, both GeneMarkS and MetaGeneMark obtained an M1 score of 39.50%, but MetaGeneMark reported a higher Percentage of Perfect Matches (M5) (65.96% compared to 61.17%) than GeneMarkS (see Fig. 2) and is thus more accurate.

In addition, GeneMarkS is ranked first for Percentage of Genes Detected (M1) when applied to *P.fluorescens* with 99.29%, compared to Prodigal, which is ranked 4th with 98.49%. However, Prodigal has the highest Percentage of Perfect Matches (M5), 92.86% versus 87.03% for GeneMarkS, which means that more of the CEA genes identified by Prodigal were exact matches. In this instance, choosing either Prodigal or GeneMarkS as the overall highest performing tool is not arbitrary.

3.2 Model-based versus *ab initio* tools

It was evident that the performance of model-based tools was less consistent across the six MOs than the *ab initio* tools. They could perform as well as or better than a number of *ab initio* tools when the model selected was the same as the genome annotated. However, if genome and model were not the same, they often produced predictions of extremely low quality. For example, GeneMark with the *E.coli* model only predicted 71 CDSs for *S.aureus*'s 2478 CEA genes, of which only 18 CDSs detected a CEA gene. However, while it could be expected that mixing different models and genomes could cause poor quality predictions from model-based tools, there were instances in which both model and genome were the same and the prediction performance was also poor. In particular, in the case of EasyGene using the *S.aureus* model, only 49.31% of *S.aureus* CEA genes were detected, a contrast from the ∼99% detected by the majority of *ab initio* tools.

Intriguingly, Augustus (a model-based tool) when employed with the *E.coli* model, was able to detect 96.64% of *P.fluorescens* genes. Both genomes are *Gammaproteobacteria*, and thus Augustus may be identifying common features of their genes. While this shows that model-based tools can perform well even when their model and target genomes are different, when Augustus was applied to *S.aureus* using the *S.aureus* model, it was only able to detect 20.53%, but unexpectedly detected 78.91% when using an *H.sapiens* model. This is indicative of the inconsistency of model-based prediction tools and the genome models they employ. In contrast, through the ranking approach, we employed, the model-based tool GeneMark.hmm with the *E.coli* model ranked higher (7/21) than a number of *ab initio* tools in both the overall ranking and for individual metrics. Furthermore, GeneMark.hmm with the *S.aureus* model was joint top in detecting the highest number of *S.aureus* CEA genes with GeneMarkS. Additionally, for each of the model-based tools, the *E.coli* model performed better across the six MOs than the *S.aureus* model.

3.3 GC content

No significant variation was observed between the CEA gene median GC content and that of the predicted CDSs from each tool, even for those with poor predictions (see Supplementary Results S2). As can be seen in Supplementary Figure S1, each of the six genomes exhibits CEA genes with a wide range of GC content profiles, irrespective of their genome's median value. We note that the GC content of genes missed by Prodigal is lower for all six MOs, but within the 25–75th percentile range for all CEA genes (Supplementary Fig. S1 and Table S4). Notably, *E.coli* and *P.fluorescens* genes, which were missed by Prodigal are nearly 10% lower in GC content than both detected and partial genes.

3.4 Overlapping CDSs

The overall number of CDSs predicted to have an overlap with another CDS varied across each of the tools and MOs, with cases of both positive and negative percentage differences when compared to the CEA annotations (see Supplementary Results S2 'Full Prediction Metrics'). Proportionally, the number of overlapping CDSs reported by *ab initio* tools are closer to the number of overlapping CEA genes than those reported by the model-based group.

Most model-based tools underpredict the proportion of overlapping CDSs with the exception of GeneMark *E.coli* for *P.fluorescens*, which predicted 2073 overlapping CDSs compared to the 1251 reported by Ensembl (see Supplementary Tables S5 and S6 and result S1 and S2).

Correct detection of CEA overlapping genes is also a problem. By totalling and averaging the Percentage Difference of Matched Overlapping Predicted CDSs (M8), we were able to observe a clear difference between the two tool groups with respect to their ability to detect correct overlapping CEA genes (see Supplementary Tables S5 and S6). The inability of the tools to account for the unusual nature of the *M.genitalium* genome was shown again with an average M8 across all tools of −88.21%, compared to the average of −27.77% for the other five genomes.

Furthermore, when making predictions for *E.coli*, while model-based tools, such as Augustus and EasyGene with the *E.coli* model can closely predict the proportion of overall overlapping CDSs (Percentage Difference of −1.42% and −2.30%, respectively), due to the poorer performance of these tools for correctly detecting CEA genes, their M8 scores for matched overlapping CDSs were substantially lower than the average score of the *ab initio* tools (grouped average of −52.89% as opposed to −23.62%—see Supplementary Table S6). Prodigal exemplifies this difference between the two tool groups. It was able to predict all overlapping CEA from *P.fluorescens* and *S.aureus*, whereas even when paired with the same model and genome, model-based tools continued to perform poorly.

3.5 Short ORFs

The lengths of detected, partially matched and missed CEA genes when predicted by Prodigal are summarized in Supplementary Figure S4. It shows that the CEA genes, which were missed by Prodigal for each genome were substantially shorter in length than the genes, which were detected, except for *M.genitalium*. For the other five MOs, whose combined median length of missed genes is 317, less than half the combined median length of 837.5 of those detected (Supplementary Fig. S4 and Results S2), it is alternative start codon selection, which influences whether a predicted CEA is shortened or elongated.

The proportion of short CEA genes in the six genomes below 300 nt ranged from 4.8% to 13.6% for each of the six MOs. All tools predicted many short CDSs for *M.genitalium* because they were incorrectly truncated due to its alternative stop codon usage. On average, *ab initio* tools were shown to be more likely to correctly detect short CEA genes across the other five MOs (see Supplementary Tables S7 and S8). Interestingly, unlike overlapping genes, short ORFs were more often overpredicted, but few were actually accurate when compared to the CEA. However, *ab initio* tools were much better suited to reporting the correct proportion of short predicted CDSs for all six genomes, often reporting the same proportion (see Supplementary Table S7). While *M.genitalium* does exhibit the highest divergence in proportional reporting of short predicted CDSs, *ab initio* tools were still less divergent (see Supplementary Table S9).

3.6 Partial matches

The number of missed CEA genes was low across the tools studied, with the exception of *M.genitalium* and some outliers from the model-based tools, such as GeneMark, Augustus and EasyGene. However, we also found many genes that were incorrectly reported on the 5' or 3' end. These misannotations, which we have called 'partial matches' if in the correct frame and accounting for ≥75% of a CEA gene, constitute either an elongation or truncation of the protein product of the gene and therefore potentially have an unknown impact on the resultant sequence. A large number of genes were incorrectly reported on the 3' end for *M.genitalium* by each tool. These 3' truncated CDSs are explained by the alternative use of TGA as tryptophan in *M.genitalium* (tools incorrectly assume this encodes a stop codon). The stop codons predicted for *M.genitalium* by all the prediction tools were the same 'TGA, TAG, TAA' as for the CEA genes of the other five MOs. Interestingly, one CEA gene in *E.coli*, which used CTT as a stop codon, was missed by all tools except for FGENESB with its *E.coli* model. FGENESB incorrectly reported the very next codon, a TGA, as the stop position. This 78 nt CEA is the only example, we found of a tool extending a CEA gene not from *M.genitalium*. Augustus with the Human model made a number of non-standard predictions due to its propensity to search for multiple CDSs for each gene but this is to be expected and is not reported in these results. Unlike 3' misprediction, a large number of genes from all six genomes were predicted with alternative start codons (see Supplementary Results S2). This was true for all tools and especially a problem for *C.crescentus* with a relatively low 68.58% ATG start codon usage for all CEA genes. The CEA genes for which Prodigal was unable to obtain a 'Perfect Match' (M5), was just 37.40%. Prodigal used a much higher level of ATG (80.87%) for this set of partially matches genes. This misidentification of start codon usage was a consistent problem among all the tools and genomes studied. However, for *E.coli*, the level of misidentification was lower. As an example, the number of times the correct or incorrect start codons were selected by Prodigal, across all six MOs, including the number of incorrectly chosen instances of the start codon (e.g. a different ATG further upstream of the real ATG) can be seen in Supplementary Table S2.

3.7 Aggregated tool predictions

Combined prediction approaches have previously been utilized to harness the prediction power from multiple tools to increase the number of detected CEA genes (Tatusova *et al.*, 2016; Yok and Rosen, 2011). For each of the MOs, taking the union of the top 5 tool predictions did provide a small increase in the number of Genes Detected (M1) (and a reduction of partial matches) compared to that of the 'best tool' [tool with highest percentage of Genes Detected (M1)] for any particular organism. However, even with this extreme case of using the union of all predicted CDSs, the increase in M1 was negligible (average increase of 0.47%) and came at the expense of predicting a large number of additional incorrect CDSs, as can be seen in Supplementary Table S10. Even in the case of *M.genitalium*, the M1 was not improved more than 0.21% with the union prediction.

3.8 Improving historic annotations

Using the GFF_Adder tool, we investigated the potential of Prodigal to add additional CDSs to the CEA annotations. There are more than 60 additional predicted CDSs that can be found for each of our MOs, and more than 270 for *E.coli* and *P.fluorescens* (see Supplementary Table S11).

4 Discussion

4.1 *Ab initio* tools usually perform better than model-based

We found that *ab initio* tools usually perform better than model-based tools. While no one tool performed the best or worst across all metrics, the *ab initio* tools Prodigal, GeneMarkS-2, MetaGeneAnnotator, MetaGeneMark and GeneMarkS were ranked first–fifth, respectively, across our 12 metric ranking (Supplementary Fig. S3 and Results S1 and Supplementary Rankings).

Strains of the same species can exhibit large intraspecies variation (Van Rossum *et al.*, 2020). Additionally, genes resulting from horizontal transfer, which is more frequent within species (Van Rossum *et al.*, 2020), are likely to contain features from the donor strains, which the rigid model-based methods are unable to recognize. GeneMark, a model-based tool, published in 1993, even when both target genome and model were *E.coli*, was identified as one of the worst performing tools in this study, possibly driven by the well-known large open pangenome of this species (Lukjancenko *et al.*, 2010). The same was observed for *S.aureus*. While model-based tools can perform well even when their model and target genomes are different, in the case of Augustus, when applied to the *C.crescentus* genome using the *S.aureus* model, it was only able to detect 3.93% of CEA genes, but unexpectedly detected 78.75% when using the *H.sapiens* model. Unsurprisingly, model-based predictors have therefore fallen out of development and use over the last decade and *ab initio*-based tools, such as Prodigal, GeneMarkS-2 and GLIMMER3 have become ubiquitous.

4.2 Codon usage has a large influence on accuracy

We found that codon usage has a large influence on accuracy due to its influence on start and stop codon choice, even in MOs.

The re-coding of a stop codon as an amino acid is rare and seems to be taxa specific (Dybvig and Voelker, 1996). While many of the tools offered the ability to change codon tables (often accounting for TGA specifically), the correct codon tables or codon preferences for each genome cannot be known in advance of annotation of a novel organism. Despite this, we would expect that they should be able to predict a significant proportion of genes, even in the absence of the knowledge of a different codon usage table. Some tools, such as Prodigal will assess a genome using both the universal and *Mycoplasma* translation table, however remarkably this did not increase the accuracy of the tool when analysing *M.genitalium* genome (see Fig. 2). Overall TGA was never predicted as tryptophan-coding in this genome by any tool (see Supplementary Results S2).

While ATG is used for 80% of start codons in the canonical annotations for most prokaryote genomes, some species and even some species-spanning gene families have been shown to use very different start codon profiles (Villegas and Kropinski, 2008). The use of different start codons in prokaryote genomes has often been correlated to the genome-wide GC content: at extreme low and high GC (<30% and >80%), ATG and GTG, respectively, are often more prominent. In our study, the extreme example of this was *C.crescentus*, which uses ATG as a start codon only 69% of the time. This is likely driven by its GC content of 67%. All of the tools performed poorly at predicting the correct start codon in this species (Fig. 2). This has been reported in the literature, specifically in relation to the lack of translation initiation sequence motifs traditionally used by prediction tools to identify the frame and start locus of a gene (Schrader *et al.*, 2014). This is not unique to *C.crescentus* and as shown in Supplementary Table S2, for all six MOs incorrect start codon selection resulted in either elongated or truncated CDSs (see Supplementary Results S2). The analysis of *E.coli* exhibited the lowest divergence between CEA and predicted start codon selection (see Supplementary Results S2 for more detail), possibly as a result of its historic use as a MO and having the largest use of the canonical ATG start codon in this study. Studies continue to investigate the possible fluidity of gene start codon selection and how some genes recorded in genomic databases may either have been annotated with the wrong start codon, or even require the annotation of multiple alternative start positions and therefore start codons (Baranov *et al.*, 2015; Meydan *et al.*, 2019; Villegas and Kropinski, 2008).

4.3 Metagenomic annotation approaches are suitable for whole genome sequences

Interestingly, tools made specifically for metagenomic and fragmented genome annotation performed better than most single genome tools (tools ranked third, fourth and sixth were developed for metagenome annotation), possibly indicating that even 'complete' genomes may themselves still harbour elements of sequencing and assembly error which these types of algorithms have been designed to account for. Most genomes submitted to databases, such as the NCBI Genome repository (Haft *et al.*, 2018), are incomplete and can contain hundreds of fragments which can make gene prediction an even more difficult task. As S. Salzberg said in 2019 'Paradoxically, the incredibly rapid improvements in genome sequencing technology have made genome annotation less, not more, accurate' (Salzberg, 2019). This indicates that future annotation work performed on non-model and more diverse organisms may benefit from approaches implemented by metagenomic tools.

4.4 Short genes and overlapping genes are often misreported

We found that short genes and overlapping genes are often misreported and that many tools still have hard-coded limitations and weightings against these types of genes, with model-based tools performing especially poorly.

It has been well established in the literature that short genes are likely under-represented across genomic databases, and therefore, possibly even within the Ensembl data used in this study (Duval and Cossart, 2017; Storz *et al.*, 2014; Su *et al.*, 2013). The growing acceptance that short genes are not only common in prokaryotic genomes but also have important roles (Andrews and Rothnagel, 2014), is at odds with many tools still containing hard-coded limitations for minimum CDS length and algorithmic weights against short CDSs. As might be expected because of its re-coding of TAG, *M.genitalium* proved challenging for all tools to accurately identify CDSs, resulting in the early truncation of a large proportion of CEA genes and an increase in predicted short CDSs. This often led to the tools predicting additional spurious short CDSs in the missed regions (a result that can be seen in the low M10 Precision metric for this genome). However, for the other genomes, most tools also predicted too many short CDSs (9.07% and 39.10%, for *ab initio*- and model-based tools, respectively), but paradoxically still managed to miss a large proportion of Short CEA genes in the Ensembl annotations (missing 26.38% and 53.69% for *ab initio*- and model-based tools, respectively) (see Supplementary Tables S7–S9 and Results S2).

For overlapping genes, while *ab initio* tools performed better than model-based tools (see Supplementary Tables S5 and S6), in general they both under-predicted the number of overlapping CEA genes across the genomes (on average −6.07% and −30.15% for *ab initio*- and model-based tools, respectively) (see Supplementary Tables S5 and S6 and Results S2). No tool was able to correctly detect more than 20% of *M.genitalium's* overlapping CEA genes. Overlapping and nested genes have now become an area of renewed interest for their potential impact on genomic organization and evolution (Huvet and Stumpf, 2014; Krakauer, 2000). For example, mokC in *E.coli*, believed to be a regulatory peptide, completely overlaps hokC and enables hokC expression (Pedersen and Gerdes, 1999) and no tool was able to detect both genes correctly.

Overall, the tools struggled to handle overlapping gene loci, and often returned either only one or neither of the overlapping coding regions in their predictions. This may be due to the manner in which many tools filter multiple candidate ORFs for a single locus leading to sub-optimal predictions. For example, Prodigal reports a CDS in *C.crescentus* on the positive strand at 23 760–24 074 when the CEA CDS is 23 550–24 170 on the negative strand. The unallocated space (24 074–24 170) resulted in Prodigal reporting the next downstream CDS starting at 24 091 instead of 24 133 (as in the Ensembl annotation), erroneously including 5' UTR in the predicted CDS. There are now tools to identify putative short ORFs in both prokaryotes and eukaryotes using additional evidence, such as RNA expression data (Bartholomäus *et al.*, 2021; Ji *et al.*, 2020; Miravet-Verde *et al.*, 2019).

Our results suggest that the identification of short and overlapping CDSs cannot be done independently without the potential for unforeseen consequences for annotation accuracy.

4.5 Historic bias affects gene prediction today

Overall, we have observed an increase in accuracy in tools over time as can be seen with the different versions of GeneMark compared here: the overall rankings of model-based GeneMark (1993) (with *E.coli*/*S.aureus* models), *ab initio* GeneMarkS (2001) and *ab initio* GeneMarkS-2 (2018) are 20/17, 5 and 2, respectively. However, GeneMarkS (2001) performed better than its successor GeneMarkS-2 (2018) for 5 out of the 12 metrics in Supplementary Figure S3 including Percentage of Genes Detected (M1) in *P.fluorescens*, *M.genitalium* and *B.subtilis* (see Supplementary Results S1 and Supplementary Rankings). The performance of GeneMarkS (2001) in M1 may reflect its use for an extended period of time in the NCBI Prokaryote Annotation Pipeline. Possibly as a result of this, many of the CEA genes GeneMarkS (2001) detected were originally identified by the tool itself. Similarly, all model-based tools performed at their best across the 12 metrics and 6 MOs when using their *E.coli* model, hinting at the impact of historical research in this organism. Advances in the realms of machine learning and statistical modelling have the greatest potential to address these issues but are also likely to be the most prone to historical biases in the databases. Many of the rules, such as standard CDS length and codon usage, are inferred from previously identified CDSs. The existence of annotation errors and omissions in various sequence databases is well established and unlikely to be resolved in the near future without significant coordination between repositories (Klimke *et al.*, 2011). Additionally, much of the sequence information derived from MOs will become less relevant as greater numbers of novel organisms are sequenced (Hunter, 2008).

These issues have been raised previously: In 2009, the 'Best Practices in Genome Annotation' meeting report listed a number of areas of concern put forward by attendees (Madupu *et al.*, 2010) including tool choice, strategy to update and correct previous annotations, tracking of changes in databases, prioritization of certain genes for experimental evaluation, documenting processes and keeping up with technological advances. The work presented here addresses the issue of tool choice, but many of the recommendations are yet to be realized. The lack of any previous detailed systematic overview of method performance may also have played a part in these biases not being addressed to date. Our study has shown that tool selection needs to be fully informed by its intended purpose and the tool's weaknesses.

4.6 Current and future techniques are needed to continue annotation improvements

It is clear from both this and previous studies that combinatory approaches are fundamental in bridging the gap to the next stage of genome annotation. This has clearly already begun with pipelines, such as PROKKA and PGAP, which utilize a collection of techniques, most notably, advanced homology searching to complete annotations where traditional CDS predictions fail or produce competing predictions. However, this can also lead to conflicting annotations. As noted, homology searches are only as good as the database being used. The presence or absence of homology does not indicate whether an ORF is a true CDS gene, especially in the nuanced field of alternative ORFs (Orr *et al.*, 2020). Further complications involving alternative ORFs, many of which are overlapping, can be found with new evidence in *E.coli*, where 'Ribosome profiling revealed out-of-frame internal minimal ORFs in 13 *E. coli* genes. Mutation of the start codon... in one gene, yecJ, resulted in an increase in translation of the main-ORF, suggesting that these minimal ORFs also can modulate translation of the main-ORF' (Meydan *et al.*, 2019).

As users of computational genomic techniques, we must realize when we have reached the limit of what is possible with the contemporary data available. This, together with other studies, proposes that the linchpin required for the next step in genome annotation, is not even more techniques reliant on current genomic knowledge, but instead more experimental work and species agnostic approaches. However, the near unlimited scope of growth conditions, environmental pressures et cetera, has made the prospect of experimentally validating all potential CDS regions unfeasible. Finally, while great strides have been made in experimentally validating difficult to characterize gene types, one such study '... suggest[s] substantial numbers of small proteins remain undiscovered in *E. coli*, and existing bioinformatics techniques must continue to improve to facilitate identification' (VanOrsdel *et al.*, 2018).

5 Conclusion

We have presented a comprehensive set of metrics, which distinguish CDS prediction tools from each other and make it possible to identify which performs better for specific use-cases. The ORForise evaluation framework enables users to evaluate new and existing annotations and generate consensus and aggregate gene predictions. We have demonstrated that certain types of genes, such as short genes, overlapping genes and those with alternative codon usage, are still elusive, even to the most advanced *ab initio* techniques. Worryingly, the performance of any tool seems to depend on the genome that is being analysed. For instance, Prodigal, which ranked best overall, was ranked first for *E.coli*, *S.aureus* and *P.fluorescens*, MetaGeneAnnotater was ranked first for *B.subtilis* and *M.genitalium* and GeneMarkS-2 was ranked first for *C.crescentus* (see Supplementary Rankings). Additionally, no individual tool ranked as the most accurate across all genomes for the Percentage of Genes Detected (M1) (the single metric historically used to assess tool performance) or any other individual metric. This is likely to have a measurable impact on downstream genomic and pangenomic studies. However, overall, we found Prodigal to be one of the most well-rounded tools, not only detecting the highest number of CEA genes for two very diverse MOs (*E.coli* and *M.genitalium*), but also performing overall best when ranked across the 12 metric rankings and 6 MOs. It was also overall best for Perfect Matched genes (M5). However, as outlined earlier, it was not always ranked first for all genomes, further suggesting that users should choose tools carefully, based on the organism and question they are studying. Finally, we advise against generating aggregated *ab initio* annotations from multiple tools where no existing annotation is available for the genome, as this results in poor overall performance. However, additional cycles of annotation with tools designed to identify putative CDSs in the unannotated regions, show promise for improving current prokaryotic genomic knowledge.

Author contributions

All authors discussed the conceptualization of the comparison platform and its impact. N.J.D. wrote the code. All authors contributed to the manuscript.

Funding

This work was supported by an Institute of Biological, Environmental and Rural Sciences Aberystwyth PhD fellowship (to N.J.D.). C.J.C. wishes to acknowledge funding from the Biotechnology and Biological Sciences Research Council (BB/E/W/10964A01, BBS/OS/GC/000011B); Department of Agriculture, Food and the Marine Ireland/DAERA Northern Ireland (MethAbate, R3192GFS); and the European Commission via Horizon 2020 (818368, MASTER).

Conflict of Interest: none declared.

References

Al-Turaiki, Israa M., et al. "Computational approaches for gene prediction: a comparative survey." *International Conference on Informatics Engineering and Information Science*. Springer, Berlin, Heidelberg, 2011.

Andrews,S.J. and Rothnagel,J.A. (2014) Emerging evidence for functional peptides encoded by short open reading frames. *Nat. Rev. Genet.*, **15**, 193–204.

Badger,J.H. and Olsen,G.J. (1999) CRITICA: coding region identification tool invoking comparative analysis. *Mol. Biol. Evol.*, **16**, 512–524.

Baranov,P.V. *et al.* (2015) Augmented genetic decoding: global, local and temporal alterations of decoding processes and codon meaning. *Nat. Rev. Genet.*, **16**, 517–529.
Bartholomäus,A. *et al.* (2021) smORFer: a modular algorithm to detect small ORFs in prokaryotes. *Nucleic Acids Res.*, **49**, e89.
Besemer,J. and Borodovsky,M. (1999) Heuristic approach to deriving models for gene finding. *Nucleic Acids Res.*, **27**, 3911–3920.
Besemer,J. and Borodovsky,M. (2005) GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Res.*, **33**, W451–W454.
Besemer,J. *et al.* (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.*, **29**, 2607–2618.
Borodovsky,M. and McIninch,J. (1993) GENMARK: parallel gene recognition for both DNA strands. *Comput. Chem.*, **17**, 123–133.
Brenner,S.E. (1999) Errors in genome annotation. *Trends Genet.*, **15**, 132–133.
Brent,M.R. (2005) Genome annotation past, present, and future: how to define an ORF at each locus. *Genome Res.*, **15**, 1777–1786.
Browning,D.F. and Busby,S.J. (2004) The regulation of bacterial transcription initiation. *Nat. Rev. Microbiol.*, **2**, 57–65.
Burge,C.B. and Karlin,S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.*, **8**, 346–354.
Cheng,A.G. *et al.* (2014) The giant protein Ebh is a determinant of *Staphylococcus aureus* cell size and complement resistance. *J. Bacteriol.*, **196**, 971–981.
Dalgarno,L. and Shine,J. (1973) Conserved terminal sequence in 18S rRNA may represent terminator anticodons. *Nat. New Biol.*, **245**, 261–262.
Dandekar,T. *et al.* (1998) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, **23**, 324–328.
Delcher,A.L. *et al.* (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, **23**, 673–679.
Devos,D. and Valencia,A. (2001) Intrinsic errors in genome annotation. *Trends Genet.*, **17**, 429–431.
Dunne,M.P. and Kelly,S. (2017) OrthoFiller: utilising data from multiple species to improve the completeness of genome annotations. *BMC Genomics*, **18**, 390.
Duval,M. and Cossart,P. (2017) Small bacterial and phagic proteins: an updated view on a rapidly moving field. *Curr. Opin. Microbiol.*, **39**, 81–88.
Dybvig,K. and Voelker,L.L. (1996) Molecular biology of *Mycoplasmas*. *Annu. Rev. Microbiol.*, **50**, 25–57.
Eilbeck,K. *et al.* (2005) The sequence ontology: a tool for the unification of genome annotations. *Genome Biol.*, **6**, R44.
Furnham,N. *et al.* (2012) Current challenges in genome annotation through structural biology and bioinformatics. *Curr. Opin. Struct. Biol.*, **22**, 594–601.
Guigo,R. (1997) Computational gene identification: an open problem. *Comput. Chem.*, **21**, 215–222.
Haas,B.J. *et al.* (2013) *De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nat. Protoc.*, **8**, 1494–1512.
Haft,D.H. *et al.* (2018) RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Res.*, **46**, D851–D860.
Howe,K.L. *et al.* (2020) Ensembl Genomes 2020 – enabling non-vertebrate genomic research. *Nucleic Acids Res.*, **48**, D689–D695.
Hunter,P. (2008) The paradox of model organisms: the use of model organisms in research will continue despite their shortcomings. *EMBO Rep.*, **9**, 717–720.
Huvet,M. and Stumpf,M.P. (2014) Overlapping genes: a window on gene evolvability. *BMC Genomics*, **15**, 721.
Hyatt,D. *et al.* (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, **11**, 119.
Jain,R. *et al.* (1999) Horizontal gene transfer among genomes: the complexity hypothesis. *PNAS*, **96**, 3801–3806.
Ji,X. *et al.* (2020) smORFunction: a tool for predicting functions of small open reading frames and microproteins. *BMC Bioinformatics*, **21**, 1–13.
Kalkatawi,M. *et al.* (2015) BEACON: automated tool for Bacterial GEnome Annotation ComparisON. *BMC Genomics*, **16**, 1–8.
Keller,O. *et al.* (2011) A novel hybrid gene prediction method employing protein multiple sequence alignments. *Bioinformatics*, **27**, 757–763.
Klimke,W. *et al.* (2011) Solving the problem: genome annotation standards before the data deluge. *Stand. Genom. Sci.*, **5**, 168–193.
Krakauer,D.C. (2000) Stability and evolution of overlapping genes. *Evolution*, **54**, 731–739.
Land,M. *et al.* (2015) Insights from 20 years of bacterial genome sequencing. *Funct. Integr. Genomics*, **15**, 141–161.
Levy,A. and Currie,A. (2015) Model organisms are not (theoretical) models. *Br. J. Philos. Sci.*, **66**, 327–348.
Lobb,B. *et al.* (2020) An assessment of genome annotation coverage across the bacterial tree of life. *Microb. Genom.*, **6**, e000341.
Lomsadze,A. *et al.* (2018) Modeling leaderless transcription and atypical genes results in more accurate gene prediction in prokaryotes. *Genome Res.*, **28**, 1079–1089.
Lukashin,A.V. and Borodovsky,M. (1998) GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.*, **26**, 1107–1115.
Lukjancenko,O. *et al.* (2010) Comparison of 61 sequenced *Escherichia coli* genomes. *Microb. Ecol.*, **60**, 708–720.
Madupu,R. *et al.* (2010) Meeting report: a workshop on best practices in genome annotation. *Database*, **2010**, baq001.
Mathé,C. *et al.* (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, **30**, 4103–4117.
Meydan,S. *et al.* (2019) Retapamulin-assisted ribosome profiling reveals the alternative bacterial proteome. *Mol. Cell*, **74**, 481–493.
Miravet-Verde,S. *et al.* (2019) Unraveling the hidden universe of small proteins in bacterial genomes. *Mol. Syst. Biol.*, **15**, e8290.
Nielsen,P. and Krogh,A. (2005) Large-scale prokaryotic gene prediction and comparison to genome annotation. *Bioinformatics*, **21**, 4322–4329.
Noguchi,H. *et al.* (2006) MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic Acids Res.*, **34**, 5623–5630.
Noguchi,H. *et al.* (2008) MetaGeneAnnotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. *DNA Res.*, **15**, 387–396.
ÓhÉigeartaigh,S.S. *et al.* (2014) Searchdogs bacteria, software that provides automated identification of potentially missed genes in annotated bacterial genomes. *J. Bacteriol.*, **196**, 2030–2042.
Orr,M.W. *et al.* (2020) Alternative ORFs and small ORFs: shedding light on the dark proteome. *Nucleic Acids Res.*, **48**, 1029–1042.
Pedersen,K. and Gerdes,K. (1999) Multiple hok genes on the chromosome of *Escherichia coli*. *Mol. Microbiol.*, **32**, 1090–1102.
Rho,M. *et al.* (2010) FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Res.*, **38**, e191.
Russell,J.J. *et al.* (2017) Non-model model organisms. *BMC Biol.*, **15**, 55–31.
Salamov,V.S.A. and Solovyevand,A. (2011) Automatic annotation of microbial genomes and metagenomic sequences. In: Li, R.W. (ed.) *Metagenomics and Its Applications in Agriculture*. Nova Science Publishers, Hauppauge, pp 61–78.
Salzberg,S.L. (2019) Next-generation genome annotation: we still struggle to get it right. *Genome Biol.*, **20**, 92.
Schafer,J.L. and Graham,J.W. (2002) Missing data: our view of the state of the art. *Psychol. Methods*, **7**, 147–177.
Schrader,J.M. *et al.* (2014) The coding and noncoding architecture of the *Caulobacter crescentus* genome. *PLoS Genet.*, **10**, e1004463.
Seemann,T. (2014) Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, **30**, 2068–2069.
Sela,I. *et al.* (2016) Theory of prokaryotic genome evolution. *PNAS*, **113**, 11399–11407.
Sommer,M.J. and Salzberg,S.L. (2021) Balrog: a universal protein model for prokaryotic gene prediction. *PLoS Comput. Biol.*, **17**, e1008727.
Stanke,M. and Morgenstern,B. (2005) AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic Acids Res.*, **33**, W465–W467.
Storz,G. *et al.* (2014) Small proteins can no longer be ignored. *Annu. Rev. Biochem.*, **83**, 753–777.
Stothard,P. (2000) The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences. *Biotechniques*, **28**, 1102–1104.
Su,M. *et al.* (2013) Small proteins: untapped area of potential biological importance. *Front. Genet.*, **4**, 286.
Tatusova,T. *et al.* (2016) NCBI prokaryotic genome annotation pipeline. *Nucleic Acids Res.*, **44**, 6614–6624.
Van Rossum,G. and Drake,F.L. (2009) *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
Van Rossum,T. *et al.* (2020) Diversity within species: interpreting strains in microbiomes. *Nat. Rev. Microbiol.*, **18**, 491–506.
VanOrsdel,C.E. *et al.* (2018) Identifying new small proteins in *Escherichia coli*. *Proteomics*, **18**, 1700064.
Villegas,A. and Kropinski,A.M. (2008) An analysis of initiation codon utilization in the Domain Bacteria–concerns about the quality of bacterial genome annotation. *Microbiology*, **154**, 2559–2661.

Warren,A.S. *et al.* (2010) Missing genes in the annotation of prokaryotic genomes. *BMC Bioinformatics*, **11**, 131.
Wood,D.E. *et al.* (2012) Thousands of missed genes found in bacterial genomes and their analysis with COMBREX. *Biol. Direct*, **7**, 37–15.
Yok,N.G. and Rosen,G.L. (2011) Combining gene prediction methods to improve metagenomic gene annotation. *BMC Bioinformatics*, **12**, 20.
Zhu,W. *et al.* (2010) *Ab initio* gene identification in metagenomic sequences. *Nucleic Acids Res.*, **38**, e132.

Bioinformatics, 37(17), 2021, 2722–2729
doi: 10.1093/bioinformatics/btab147
Advance Access Publication Date: 3 March 2021
Original Paper

Databases and ontologies

DeepViral: prediction of novel virus–host interactions from protein sequences and infectious disease phenotypes

Wang Liu-Wei[1], Şenay Kafkas[1,2], Jun Chen[1], Nicholas J. Dimonaco [3], Jesper Tegnér[1,4] and Robert Hoehndorf [1,2,*]

[1]Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, [2]Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, [3]Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, Wales SY23 3BQ, UK and [4]Biological and Environmental Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

*To whom correspondence should be addressed.
Associate Editor: Peter Robinson

Received on August 13, 2020; revised on January 18, 2021; editorial decision on February 28, 2021; accepted on March 1, 2021

Abstract

Motivation: Infectious diseases caused by novel viruses have become a major public health concern. Rapid identification of virus–host interactions can reveal mechanistic insights into infectious diseases and shed light on potential treatments. Current computational prediction methods for novel viruses are based mainly on protein sequences. However, it is not clear to what extent other important features, such as the symptoms caused by the viruses, could contribute to a predictor. Disease phenotypes (i.e. signs and symptoms) are readily accessible from clinical diagnosis and we hypothesize that they may act as a potential proxy and an additional source of information for the underlying molecular interactions between the pathogens and hosts.

Results: We developed DeepViral, a deep learning based method that predicts protein–protein interactions (PPI) between humans and viruses. Motivated by the potential utility of infectious disease phenotypes, we first embedded human proteins and viruses in a shared space using their associated phenotypes and functions, supported by formalized background knowledge from biomedical ontologies. By jointly learning from protein sequences and phenotype features, DeepViral significantly improves over existing sequence-based methods for intra- and inter-species PPI prediction.

Availability and implementation: Code and datasets for reproduction and customization are available at https://github.com/bio-ontology-research-group/DeepViral. Prediction results for 14 virus families are available at https://doi.org/10.5281/zenodo.4429824.

Contact: robert.hoehndorf@kaust.edu.sa

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Infectious diseases emerging unexpectedly from novel and reemerging pathogens have been a major enduring public health concern around the globe (Jones *et al.*, 2008). Pathogens disrupt host cell functions (Finlay and Cossart, 1997) and target immune pathways (Dyer *et al.*, 2010) through complex inter-species interactions of proteins (Dyer *et al.*, 2008), RNA (Fajardo *et al.*, 2015) and DNA (Weitzman et al., 2004). The study of pathogen–host interactions (PHI) can therefore provide insights into the molecular mechanisms underlying infectious diseases and guide the discoveries of novel therapeutics or provide a basis for the repurposing of available drugs. For example, a previous study of many PHIs showed that pathogens typically interact with the protein hubs (those with many interaction partners) and bottlenecks (those of central locations to important pathways) in human protein–protein interaction (PPI) networks (Dyer *et al.*, 2008). However, due to cost and time constraints, experimentally validated pairs of interacting pathogen–host proteins are limited in number. Therefore, the computational

prediction of PHIs is a useful complementary approach in suggesting candidate interaction partners from the human proteome.

Existing PHI prediction methods for novel viruses typically utilize protein sequence features of the interacting proteins (Alguwaizani *et al.*, 2018 ; Eid *et al.*, 2016; Yang et al., 2020; Zhou et al., 2018). While protein functions have been shown to predict intra-species (e.g. human) PPIs (Guzzi et al., 2012; Jain and Bader, 2010; Pesquita et al., 2009) and such protein specific features exist for some extensively studied pathogens, such as *Mycobacterium tuberculosis* (Huo et al., 2015) and HIV (Mukhopadhyay et al., 2014), for most pathogens, these features are rare and expensive to obtain. As new virus species continue to be discovered (Woolhouse et al., 2012), a method is needed to rapidly identify candidate interactions from information that can be obtained quickly, such as the signs and symptoms exhibited by the host, which may be utilized as a proxy for the underlying molecular interactions between host and pathogen proteins.

The phenotypes elicited by pathogens, i.e. the signs and symptoms observed in a patient, may provide information about molecular mechanisms (Gkoutos *et al.*, 2018). The information that phenotypes provide about molecular mechanisms is commonly exploited in computational studies of Mendelian disease mechanisms (Oellrich et al., 2016), for example, to suggest candidate genes (Hoehndorf *et al.*, 2011; Meehan *et al.*, 2017) or diagnose patients (Köhler *et al.*, 2009), but the information can also be used to identify drug targets (Hoehndorf *et al.*, 2013a) or gene functions (Hoehndorf *et al.*, 2013b). We hypothesize that the host phenotypes elicited by an infection with a pathogen are, among others, the result of molecular interactions, and that knowledge of the phenotypes exhibited by the host can be used to suggest the protein perturbations from which these phenotypes arise.

One major challenge of the novel PHI prediction problem is the lack of ground truth negative data. A recent method, DeNovo (Eid *et al.*, 2016), adopted a 'dissimilarity-based negative sampling': for each virus protein, the negatives are sampled from human proteins that do not have known interactions with other similar virus proteins (above a sequence similarity threshold T). Another method based on protein sequences (Zhou *et al.*, 2018), samples negatives from only the set of host proteins that are less than 80% similar (in terms of sequence similarity) to the host proteins in the positive training data. However, the influence of sequence similarity on function is not uniform and while there is evidence for a number of general evolutionary rules, we are unable to determine cutoffs for any specific protein or function (Ponting, 2001; Whisstock and Lesk, 2003). By construction, these sampling schemes make the human proteins in the negative set different from the positive set; when used not only for training a model but also for evaluating its performance, this sampling scheme has the potential to over-estimate the actual performance for finding novel PHIs. In a more realistic evaluation for a novel virus species, a model would be evaluated on all the host proteins with which it could potentially interact, regardless of sequence similarity.

From these motivations, we developed a machine learning method, DeepViral, to predict potential interactions between viruses and all human proteins for which we can generate the relevant features. Firstly, the features of phenotypes, functions and taxonomic classifications are embedded in a shared space for human proteins and viruses. We then extended a sequence model by incorporating the phenotype features of viruses into the model. We show that the joint model trained on both the sequences and phenotypes can significantly outperform state-of-the-art methods and predict potential PHIs in realistic experimental setups for novel viruses.

2 Materials and methods

DeepViral is a model that predicts potential protein interactions between viruses and human hosts from the protein sequences and feature embeddings of phenotypes, functions and taxonomies. To enable predictions based on such different features we embedded them in a shared representation space. We then combine these feature embeddings with a protein sequence model to predict potential PHIs of novel viruses. The workflow of DeepViral is illustrated in Figure 1.

2.1 Data sources

Interactions between hosts and pathogens were obtained from the Host Pathogen Interaction Database (HPIDB; version 3) (Ammari et al., 2016). The phenotypes associated with pathogens were collected from the PathoPhenoDB (Kafkas et al., 2019), a database of manually curated and text-mined associations of pathogens, infectious diseases and phenotypes. We downloaded the PathoPhenoDB database version 1.2.1 (http://patho.phenomebrowser.net/).

The phenotypes associated with human genes were collected from the Human Phenotype Ontology (HPO) database (Köhler *et al.*, 2019), and the phenotypes associated with mouse genes and the orthologous gene mappings from mouse genes to human genes originated from the Mouse Genome Informatics (MGI) database (Smith *et al.*, 2018). The Entrez gene IDs in HPO and MGI were mapped to reviewed Uniprot protein IDs using the Uniprot Retrieve/ID mapping tool (https://www.uniprot.org/uploadlists) on March 6, 2020. The Gene Ontology annotations of human proteins (release date 2020-02-22) were downloaded from the Gene Ontology Consortium (Ashburner *et al.*, 2000; The Gene Ontology Consortium, 2017). Human PPI networks were downloaded from String (Szklarczyk et al., 2019) and filtered to only include the interactions with experimental evidence. The human protein sequences were obtained from the Swiss-Prot database (UniProt Consortium, 2019).

To add background knowledge from biomedical ontologies of phenotypes and GO classes, we downloaded the cross-species PhenomeNET ontology (Hoehndorf *et al.*, 2011; Rodríguez-García *et al.*, 2017), from the AberOWL ontology repository (Hoehndorf *et al.*, 2015a) on September 13, 2018. We obtained the NCBI Taxonomy classification (Sayers *et al.*, 2009) as an ontology in OWL format (version 2018-07-27) from EMBL-EBI ontology repository (https://www.ebi.ac.uk/ols/ontologies/ncbitaxon).

The SARS-CoV-2 interactions are from a recently released dataset of 332 PHIs from 27 viral proteins (Gordon *et al.*, 2020). The PHIs of other Coronaviruses are from a recently curated dataset of Coronaviridae–host PPIs (Perrin-Cocon *et al.*, 2020). The protein sequences of the Coronaviruses in our study are retrieved from the Swiss-Prot database (UniProt Consortium, 2019).

2.2 Learning feature embeddings

To generate feature embeddings, we used DL2Vec (Chen *et al.*, 2020), a recent method for learning features for entities (in our case, the human proteins and viruses) from their associations to ontological classes. DL2Vec first converted the ontologies and entity associations into a graph, with the classes and entities as the nodes and the associations and ontology axioms as the edges. Then a number of random walks were performed, starting from the entities over to the ontology graph and thereby generating a corpus of walks in the form of sentences capturing the graph neighborhoods and thereby the ontology axioms. After the construction of such sentences, a Word2vec skipgram model (Mikolov *et al.*, 2013) was used to learn an embedding for each entity by learning from the corpus. Following the recommendations of the authors of DL2Vec, we fixed the number of walks to 100, the walk length to 30, the embedding dimension to 100 and the number of training epochs to 30. The embeddings were trained with the Word2Vec library in Julia (version 1.0.4). The resultant embedding was a vector representation of an entity capturing its co-occurrence relations with other entities within the walks generated by DL2Vec. As an example, the walks starting from a virus node explored its graph neighborhood, i.e. its associated phenotypes and its taxonomic relatives, and as an result, its feature embedding captured this information according to the co-occurrence patterns.

2.3 Supervised prediction models and parameter tuning

The neural network model of DeepViral consists of two components: a phenotype model based on the feature embeddings of viruses and human proteins, and a sequence model based on the amino acid sequences of the human and viral proteins. The

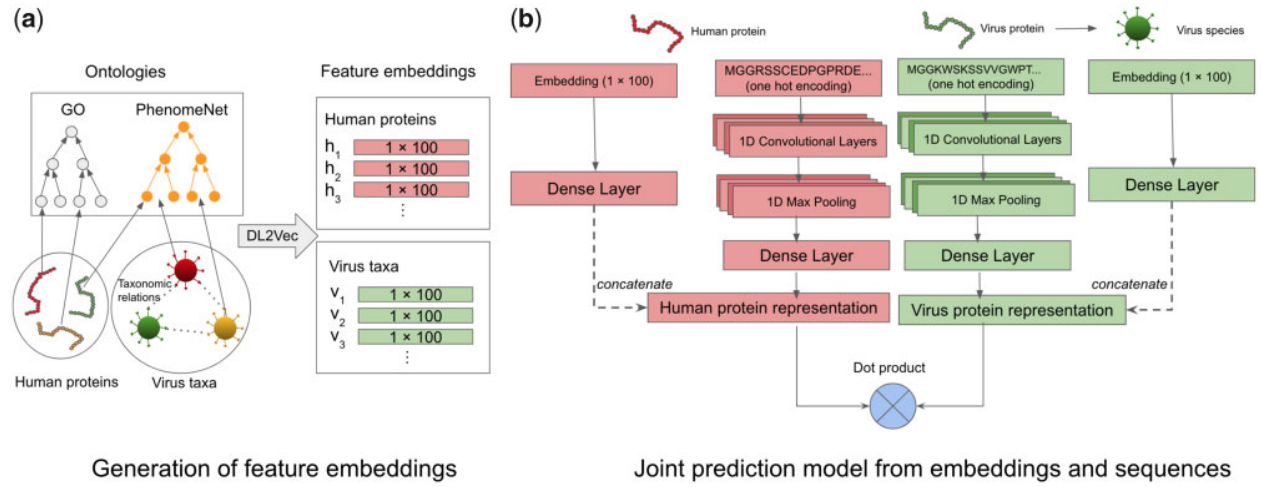


Fig. 1. The workflow of DeepViral. (**a**) Generation of an embedding: the arrows of human proteins and virus taxa represent their annotations to the ontology classes. The dashed lines between viruses represent their taxonomic relations. The annotations, taxonomic relations and ontologies were fed into DL2Vec to generate feature embeddings of dimension 100 for each human protein and virus taxa. (**b**) Joint prediction model: latent representations learned from feature embeddings and protein sequences are concatenated into a joint representation, for human protein and virus protein respectively, on which a dot product is performed to predict interactions

maximum input length of protein sequences is set to 1000 amino acids and all shorter sequences are repeated up to the maximum length. The sequence length cut-off of 1000 is chosen to cover the majority of proteins in the databases from which we constructed our dataset, i.e. 88.2% and 83.7% of the human proteins in Swiss-Prot and HPIDB, respectively, and 91.6% of the virus proteins in HPIDB. The input protein sequences are encoded as a one-hot encoding matrix of 22 rows that represents each amino acid type and the original sequence length (before being repeated), and 1000 columns representing each position of the amino acid sequence.

To predict the likelihood of an interaction between a pair of proteins, we trained the network as a binary classifier, to minimize the binary cross-entropy loss defined below,

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_t \cdot log(y_p) + (1 - y_t) \cdot log(1 - y_p)$$

where N is the total number of predictions, y_t and y_p is the true label and predicted likelihood of y.

We implemented our model using the Keras library and performed training on Nvidia Tesla V100 GPUs. The phenotype model consists of a fully connected layer with the feature embeddings as input. The sequence model, adapted from DeepGOPlus (Kulmanov and Hoehndorf, 2020), is a convolutional neural network (CNN) with the sequences as input and consists of 1-dimensional convolution, max pooling and fully connected layers. We tuned the following hyperparameters of the model through a grid search: the maximum size of the convolution filters (i.e. 16, 32 and 64), the number of the filters (i.e. 8 and 16), the size of the max pooling layers (i.e. 50 and 200) and the number of neurons in the fully connected layers (i.e. 8, 16 and 32). We then fixed these hyperparameters throughout all the experiments: 16 convolutional layers for each filter of 8, 16,. . ., 64 in length, a pool size of 200 and 8 neurons for the dense layers. We also used dropouts (Srivastava *et al.*, 2014) for the convolutional and dense layers with a rate of 0.5 and LeakyReLU as the activation function for the dense layer with an alpha set to 0.1.

3 Results

3.1 Embedding features of viruses and human proteins from phenotypes, functions and taxonomies

We started with the biological hypothesis that phenotypes (i.e. symptoms) elicited by viruses in their hosts can act as a proxy for the underlying molecular mechanisms of the infection, and therefore may provide additional information to the prediction of potential PHIs for novel viruses.

To generate feature embeddings for human proteins and virus taxa, we applied a recent representation learning method DL2Vec (Chen *et al.*, 2020), which learned feature embeddings for entities based on their annotations to ontology classes (see Section 2.2). DL2Vec takes two types of inputs: the associations of the entities with ontology classes (e.g. human proteins and their functions), and the ontologies themselves.

For representing virus taxa through the phenotypes they elicit in their hosts, we used the phenotype associations for viruses from PathoPhenoDB (Kafkas et al., 2019), a database of pathogen to host phenotype (signs and symptoms) associations. To increase the coverage of phenotypes beyond PathoPhenoDB, the taxonomic relations of the viruses were added from the NCBI Taxonomy (Sayers *et al.*, 2009). By adding these taxonomic relations (as annotations of viruses to DL2Vec), we propagated the known phenotypes along the taxonomic hierarchies and learned a generalized embedding for viruses that do not have any phenotype annotations in PathoPhenoDB but have close relatives that do.

Similarly, for representing human proteins, we used the annotations of their associated phenotypes from the Human Phenotype Ontology (HPO) database (Köhler *et al.*, 2019), the phenotypes associated with their mouse orthologs from the Mouse Genome Informatics (MGI) database (Smith *et al.*, 2018), and their protein functions from the Gene Ontology (GO) database (Ashburner *et al.*, 2000; The Gene Ontology Consortium, 2017). We propagated these annotations through the human PPI network, which has been shown to improve prediction for gene-disease associations (Alshahrani and Hoehndorf, 2018).

To provide DL2Vec with structured background knowledge of human and mouse phenotypes as well as protein functions, we used the cross-species phenotype ontology PhenomeNET (Hoehndorf *et al.*, 2011; Rodríguez-García *et al.*, 2017), which is built upon and includes the Gene Ontology (Ashburner *et al.*, 2000; The Gene Ontology Consortium, 2017). These ontologies contain formalized biological background knowledge (Hoehndorf *et al.*, 2015b), which has the potential to significantly improve the performance of these features in machine learning and predictive analyses (Kulmanov *et al.*, 2020; Smaili *et al.*, 2020).

3.2 A joint model for PPI prediction from sequences and phenotypes

DeepViral consists of a phenotype model trained on phenotypes caused by a viral infection and a sequence model trained on protein

sequences, as shown in Figure 1b. The two models take a pair of virus and human proteins as input and predicts the probability score of their interaction. The inputs for a human protein are its feature embedding and its sequence, and the features for a viral protein are its sequence and the feature embedding of the virus species to which it belongs. The sequence model projects the protein sequence into a low dimension vector representation, which is concatenated with the vector projected from the embedding by the phenotype model to form a joint representation of the proteins. A dot product was performed over the two vector representations of the pair of proteins to compute their similarity, which was then used as input to a sigmoid activation function to compute their predicted probability of interaction. In an evaluation where the inputs were not symmetric, e.g. only using the feature embeddings of human proteins but not viruses (or vice versa), an additional dense layer was added to project the longer representation to the same dimension as the other so that the dot product could be performed.

Existing prediction methods for inter-species PPI (e.g. virus–human interactions) have rarely been compared with methods designed for intra-species (e.g. human) PPI prediction. To compare with the existing sequence-based methods for both intra- and inter-species PPI prediction, we evaluated DeepViral and RCNN (Chen *et al.*, 2019), a recent method designed for intra-species prediction, on an existing dataset (Eid *et al.*, 2016) that has been used to evaluate a number of PHI prediction methods (Alguwaizani *et al.*, 2018; Yang *et al.*, 2020; Zhou *et al.*, 2018). The respective model performances and implementation details are shown in Supplementary Section S1. DeepViral trained only on sequences achieves comparable performance with other sequence based methods, while the joint model is able to achieve the best performances in most metrics. However, the evaluation dataset suffers from several drawbacks: (i) negative sampling (to create a balanced dataset) was based on sequence dissimilarity; (ii) the training and test sets only cover 39 viral proteins from 26 virus strains and 11 families, which is highly limited relative to the current size and taxonomic diversity of the PHI databases; (iii) there are overlapping virus proteins (i.e. data leakage) at species level between the training and test sets, which makes it unsuitable for the problem of novel PHI prediction.

3.3 Experimental setup, negative sampling and evaluation metrics for novel viruses

Motivated by the need for more representative datasets to evaluate methods for novel PHI prediction, we constructed a larger dataset from the curated virus–host interactions in HPIDB (Ammari et al., 2016), a database of host–pathogen protein–protein interactions. We constructed our positive set by filtering HPIDB to include all virus–host interactions that (i) are provided with an MIscore, a confidence score for molecular interactions (Villaveces et al., 2015); (ii) are associated with an existing virus family in the NCBI taxonomy (Sayers *et al.*, 2009); (iii) are within 1000 amino acids in length (for both human and viral proteins). After filtering, the dataset includes 24 678 positive interactions and 1066 viral proteins from 14 virus families and 292 virus taxa.

To realistically evaluate the prediction performance, we performed a leave-one-family-out (LOFO) cross validation: at each run, one virus family in our positive set was left out for testing, 20% of the remaining families for validation, and the rest 80% for training. The objective of the LOFO cross-validation is to evaluate the model under a scenario in which the novel virus emerges from a novel virus family—in our study, 'novel' is defined as the situation in which we have no or very little knowledge about its protein interactions and the molecular functions of the viral proteins.

Instead of using 'dissimilarity-based negative sampling' to construct a balanced dataset, we sampled our negatives from all the possible pairwise combinations of human and viral proteins, as long as the pair did not occur in the positive set. Essentially, we treated all 'unknown' interactions as negatives. As the dataset was at this point unbalanced with more negatives than positives, we evaluated the model with the area under the receiver operating characteristic (ROC) curve (Fawcett, 2006). A high ROCAUC indicates the ability of the model to rank the true positive interacting proteins higher than proteins for which no such interaction is known. We computed a ROCAUC for each virus family, and also for each viral protein and virus taxon in that family, for which we reported the mean across them, i.e. macro averages. Each model was evaluated 5 times independently to compute the 95% confidence interval of the ROCAUC, which is bounded by mean$\pm 1.96\times\sigma n$, where n is the sample size and σ is the standard deviation. Additionally, the mean ranks of the true positive proteins were provided as a more interpretable metric: for each viral protein, we ranked all of the 16 627 human proteins in Swiss-Prot (with a length limit of 1000) as its potential interaction partner based on the prediction score and obtained the mean ranks of the true positives.

3.4 Phenotypes improve prediction for novel viruses

With the newly constructed dataset, we further evaluated the existing methods as well as the variants of DeepViral, under the scenario in which a novel virus (from a novel family) emerges and no previous knowledge (except about its protein sequences and the phenotypes elicited in its hosts) is known.

We compared DeepViral with two existing state-of-the-art methods based on protein sequences: Doc2Vec + RF (Yang *et al.*, 2020), a recent method predicting for virus–human interactions; and RCNN (Chen *et al.*, 2019), a recent deep learning based method for intra-species (e.g. human) PPI prediction. To adapt Doc2Vec + RF on our dataset, we used the pretrained Doc2Vec model provided by the authors and the same parameters for the random forest model for training. Similarly, for RCNN, we used the pre-trained embeddings for amino acids and the same model parameters for training. Since the stop criterion for Doc2Vec + RF was to have at most 2 samples at each leaf node, we did not use validation data and trained it with the entirety of the training data, while a validation set was used for both RCNN and DeepViral as described in the experimental setup.

For each model, the summary statistics of the predictive performance are shown in Table 1. For models using only sequence features, DeepViral and Doc2Vec + RF perform on a similar level across the metrics. As the current state-of-the-art method for intra-species PPI prediction, RCNN consistently yields the lowest performances. Adding human or virus embeddings individually shows a slight improvement in most metrics, compared to the sequence-only models, while the joint model with both embeddings achieved the best performances overall. The distributions of the ranks of true positives (Fig. 2) are in general correspondence with the summary statistics, with the joint model having lowest ranks overall.

4 Discussion

4.1 Species-level optimization of DeepViral for novel viruses

The continued emergence of novel viruses is an issue of increasing relevance to global public health (Woolhouse et al., 2012) and economic stability (Chakraborty and Maity, 2020). Accurate prediction of potential PHIs for novel viruses with rapidly obtainable features, such as sequences and phenotypes, would be important for understanding infectious disease mechanisms and the repurposing of existing drugs. The LOFO cross-validation excludes the taxonomic relatives from the same family of the test virus, simulating a challenging scenario where the virus is from an entirely novel family. While this provides a stringent evaluation scheme for DeepViral, it likely leads to an underestimate of performance when applied to real world PHI data as most emerging viruses arise from existing virus families (Woolhouse et al., 2012). To investigate whether the inclusion of data from viruses in the same family can improve DeepViral's ability to predict interactions for viral species, we additionally designed and implemented a leave-one-species-out (LOSO) training and evaluation method. Due to the large number of species, we only applied this method to three viral species from three different RNA virus families, as well as the novel coronavirus SARS-CoV-2 based on a recently released dataset (Gordon *et al.*, 2020).

Table 1. Comparison with the state-of-the-art methods on our dataset to evaluate the performances for novel viruses

Method	Family-wise ROCAUC	Taxon-wise ROCAUC	Protein-wise ROCAUC	Mean rank
RCNN (Chen *et al.*, 2019)	0.726 [0.717–0.734]	0.759 [0.750–0.768]	0.737 [0.731–0.743]	4669
Doc2Vec + RF (Yang *et al.*, 2020)	0.764 [0.763–0.765]	0.768 [0.766–0.770]	0.751 [0.751–0.752]	3740
DeepViral (seq)	0.770 [0.763–0.777]	0.768 [0.759–0.777]	0.749 [0.742–0.756]	4064
DeepViral (seq + human embedding)	0.778 [0.766–0.790]	0.789 [0.776–0.801]	0.757 [0.742–0.771]	4245
DeepViral (seq + viral embedding)	0.788 [0.776–0.801]	0.782 [0.773–0.790]	0.757 [0.746–0.767]	3496
DeepViral (joint)	**0.813 [0.808–0.817]**	**0.829 [0.822–0.836]**	**0.800 [0.797–0.804]**	**3156**

Note: The brackets after DeepViral indicate the features used for the model: seq—protein sequences, joint—both sequences and embeddings of human proteins and viruses. The square brackets behind ROCAUC scores indicate the 95% confidence interval. The bold numbers indicate the best performing method for the respective metrics.

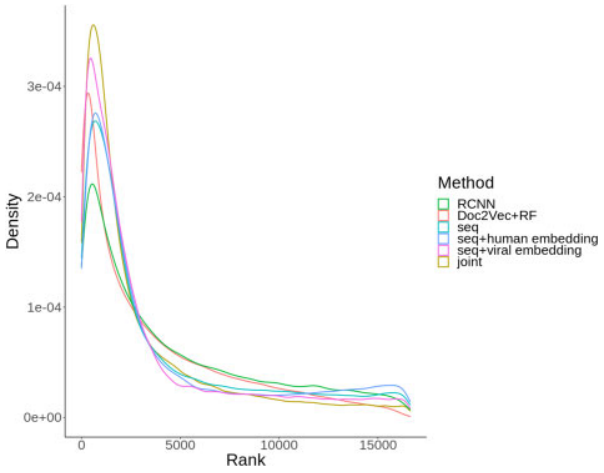


Fig. 2. Density plot of the predicted ranks of true positives for each PHI prediction method. The last four methods correspond to the variants of DeepViral

LOSO is different from LOFO with respect to the training and validation datasets: for each test species, one species from the same family is chosen as the validation set and the rest of the family are all included in the training set. To ensure there is no taxonomic leakage, i.e. identical virus protein sequences among the training, validation and testing datasets, we excluded virus taxa for which proteins have 100% sequence identity.

The comparison between the LOFO and LOSO evaluation is shown in Table 2 and the taxonomic information of the viruses is shown in Supplementary Section S2. When including data from taxonomic relatives (those of the same virus family) in the training and validation sets, the predictive performance of DeepViral improved in all four test cases. The improvements for different viruses exhibited large variability (see Table 2). For example, the Influenza A virus had the largest increase in performance among the four viral species. A similar difference between the virus families can also be observed from the LOFO experiments, as shown in Figure 3. Both the sequence and joint models show similar family-wise variability, with some occasional differences, e.g. Retroviridae performs better than Herpesviridae in the joint model but not in the sequence-based model. The taxon-wise variabilities in both LOFO and LOSO suggest that the features used to predict PHIs may have different generalization and prediction powers across different virus taxa, or PHIs may be characterized to different degrees of completeness. In the future, explainable models (Ribeiro et al., 2016; Lundberg and Lee, 2017) may provide more interpretable insight into this variability.

A contemporary example of a novel virus is the coronavirus SARS-CoV-2, which by the end of 2020 reached more than 83.4 million cases of infections and 1.8 million fatalities globally (Dong *et al.*, 2020) in a timespan of 13 months. In the short time since its emergence, many experimental studies of PHIs between SARS-CoV-2 and human proteins have been published at a historical speed, which enabled biologists to speculate on the infection mechanisms and suggest drug candidates for repurposing (Gordon *et al.*, 2020).

The Coronaviridae M protein constitutes an integral part of the SARS-CoV-2 viral envelope, involved in morphogenesis and assembly via its composite interactions with other structural proteins (Mousavizadeh and Ghasemi, 2020). DeepViral has predicted an interaction between the M protein and the TANK-binding kinase TBK1 (UniProt: Q13158, within top 0.1% of all human proteins). TBK1 plays an important role in the activation of many genes involved in the innate immune response (Fitzgerald *et al.*, 2003; Ran *et al.*, 2016). The interaction between the SARS-CoV-2 M protein and TBK1 was recently validated through affinity capture experiments (Zheng et al., 2020) and proximity-dependent biotinylation methods (Samavarchi-Tehrani *et al.*, 2020). TBK1 has previously been associated with phenotypes related to respiratory distress and respiratory failure through its complex role in amyotrophic lateral sclerosis (Oakes *et al.*, 2017), matching the respiratory phenotypes associated with COVID-19 infections. While the predictions made by DeepViral do not yet allow for a complete understanding of underlying causality, the interaction identified by DeepViral demonstrates how sequence and phenotype information is combined for predicting interactions.

4.2 Using phenotypes to reveal molecular mechanisms of viral infections

DeepViral is, to our knowledge, the first machine learning method that uses clinical phenotypes as a feature to predict PHIs between viruses and human hosts. The use of phenotypes has resulted in a significant improvement ($P < 0.05$; see confidence intervals in Table 1) over methods that rely on sequences alone. Our model avoids the bottleneck of identifying the molecular functions of pathogen proteins by instead introducing a novel and—in the context of infectious diseases—rarely explored type of feature, the phenotypes elicited by pathogens in their hosts, as a 'proxy' for the molecular mechanisms, which in turn eventually produce the observed clinical phenotypes.

One challenge in using phenotypes associated with viral infections or proteins is that they have been derived under different contexts. While phenotypes associated with viral infections are the result of the immune-mediated response and observed in a clinical context (Kafkas et al., 2019), the phenotypes of human proteins are usually associated with a loss or depletion of protein function (Köhler *et al.*, 2019; Smith *et al.*, 2018). However, the phenotypes associated with viral infections obtained from PathoPhenoDB focus on hallmark phenotypes of viral infections that can be used to discriminate between infections of different viruses and thereby de-emphasize the phenotypes resulting from general immune response (Kafkas et al., 2019). Furthermore, the application of neural networks with supervised training can account for differences between observed phenotypes and may even exploit patterns in these differences that are not explicit in the phenotypic representations (Kulmanov *et al.*, 2020; Kulmanov and Hoehndorf, 2020b).

Utilizing phenotypic features observed in humans and mice may have the crucial advantage that we can identify PHIs that may contribute to particular signs and symptoms of infection (Durrant *et al.*,

Table 2. Improvements of DeepViral's predictive performance for four virus species, between leave-one-family-out (LOFO) and leave-one-species-out (LOSO) evaluation

Test virus	Taxon ID	Taxon-wise ROCAUC		Protein-wise ROCAUC		Mean rank	
		LOFO	LOSO	LOFO	LOSO	LOFO	LOSO
SARS-CoV-2	2697049	0.710 [0.680–0.740]	**0.729 [0.708–0.750]**	0.750 [0.716–0.784]	**0.776 [0.756–0.796]**	4683	**4344**
Zika virus	2043570	0.729 [0.699–0.760]	**0.771 [0.752–0.790]**	0.731 [0.716–0.746]	**0.748 [0.731–0.765]**	4516	**4413**
HPV 18	333761	0.747 [0.716–0.779]	**0.801 [0.771–0.831]**	0.820 [0.795–0.846]	**0.890 [0.872–0.907]**	4157	**3240**
Influenza A	644788	0.804 [0.787–0.821]	**0.933 [0.907–0.958]**	0.804 [0.788–0.821]	**0.935 [0.914–0.956]**	3306	**1112**

Note: Taxon identifiers are based on the NCBI Taxonomy Database (Sayers *et al.*, 2009). Each experiment was repeated five times to compute the 95% confidence interval. The bold numbers indicate the better performing method between LOFO and LOSO.

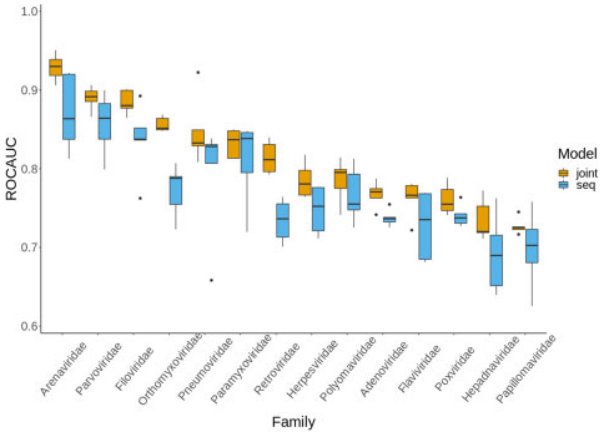


Fig. 3. ROCAUC for each of the 14 virus families from the joint model and the sequence model, respectively, ordered by the ROCAUC of the joint model

2011). For example, our model consistently ranks the RNA helicase protein DDX3X (UniProt: O00571) within the top 0.37% of all human proteins as a potential interaction partner of the non-structural protein 4A (UniProt: A0A024B7W1-PRO_0000443029) of Zika virus (NCBITaxon : 2043570). Infections with Zika virus may result in abnormal embryogenesis and, in particular, microcephaly (Wang *et al.*, 2017). Phenotypes associated with DDX3X in the mouse ortholog include abnormal embryogenesis, microcephaly and abnormal neural tube closure (Chen *et al.*, 2016). While DDX3X has previously been linked to the infectivity of the Zika virus (Doñate-Macián *et al.*, 2018) and can result in intellectual disability (Blok *et al.*, 2015), our model further suggests a role of DDX3X in the development of the embryogenesis phenotypes from Zika virus infections.

4.3 Evaluating predictions for novel viruses

While we have demonstrated a quantitative improvement over existing methods on a previously published dataset (Eid *et al.* 2016; see Supplementary Section S1), we argue that the performance of PHI prediction methods may be over-estimated on datasets where negatives are obtained using a 'dissimilarity-based negative sampling' strategy; when only human proteins that are sufficiently different from known interaction partners of viruses are considered for an evaluation, the prediction task is likely to become too simple to reflect performance in a realistic scenario. To address this challenge, we establish an evaluation strategy in which all host proteins are considered as potential interaction partners for novel viruses. Using this evaluation, the predictive performance is considerably lower than using a dissimilarity-based sampling strategy (see Table 1). Another possible explanation for the decrease in performance is that our negative set likely includes some positive interactions that are (falsely) considered as negatives due to absent knowledge of the interaction; this can potentially result in an underestimation of the actual predictive performance.

We use the mean ranks to evaluate model predictions when challenged with a novel virus from a novel family (LOFO), or with known interactions from its taxonomic relatives (LOSO). However, even the best performing model, i.e. DeepViral jointly trained with phenotypes and sequences, has only been able to rank the known true positive proteins up to a mean rank of 3156 out of all 16 627 human proteins in the LOFO evaluation. While the mean rank is sensitive to predictions at a low rank (see Fig. 2), future work is required to further improve PHI prediction methods, especially in regards to the feature selection and engineering, and evaluation methodologies.

4.4 Limitations and future work

DeepViral has several limitations that can be addressed by future work. One is the scarcity of training data for inter-species PPIs. This challenge may be addressed by transfer learning on the much larger intra-species PPI data available for humans and other model organisms. We also did not utilize other types of PHIs outside virus–human interactions in our current study, such as those of other hosts, e.g. plants and fishes, and other types of pathogens, e.g. bacteria and fungi; both may provide further insights in PHIs and the mechanisms underlying viral infections. In particular, in zoonotic diseases, information from PHIs in animals (if available) may be used to identify or suggest interactions that occur in human hosts (Dimonaco *et al.*, 2020; Li *et al.*, 2020). Furthermore, predicting tissue-specific PHIs would also provide additional insights as proteins of both human hosts (Fagerberg *et al.*, 2014) and viruses (Jarosinski *et al.*, 2012) often have tissue-specific expressions and functions.

Acknowledgements

The authors thank Maxat Kulmanov and Mona Alshahrani for their advice on earlier versions of this work. They also thank Jeffery Law for making public the mappings of the SARS-CoV-2 proteins. They acknowledge the use of computational resources from the KAUST Supercomputing Core Laboratory.

Funding

This work was supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/3790-01-01.

Conflict of Interest: none declared.

References

Alguwaizani,S. *et al.* (2018) Predicting interactions between virus and host proteins using repeat patterns and composition of amino acids. *J. Healthcare Eng.*, **2018**, 1391265.

Alshahrani,M. and Hoehndorf,R. (2018) Semantic disease gene embeddings (SmuDGE): phenotype-based disease gene prioritization without phenotypes. *Bioinformatics*, **34**, i901–i907.

Ammari,M.G. *et al.* (2016) HPIDB 2.0: a curated database for host–pathogen interactions. *Database*, **2016**, baw103.
Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.
Blok,L.S. *et al.* (2015) Mutations in DDX3X are a common cause of unexplained intellectual disability with gender-specific effects on Wnt signaling. *Am. J. Hum. Genet.*, **97**, 343–352.
Chakraborty,I. and Maity,P. (2020) COVID-19 outbreak: migration, effects on society, global environment and prevention. *Sci. Total Environ.*, **728**, 138882.
Chen,C.-Y. *et al.* (2016) Targeted inactivation of murine DDX3X: essential roles of DDX3 in placentation and embryogenesis. *Hum. Mol. Genet.*, **25**, 2905–2922.
Chen,J. *et al.* (2020) Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*, **2020**, btaa879.
Chen,M. *et al.* (2019) Multifaceted protein–protein interaction prediction based on siamese residual RCNN. *Bioinformatics*, **35**, i305–i314.
UniProt Consortium. (2019) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, **47**, D506–D515.
Dimonaco,N.J. *et al.* (2020) Computational analysis of SARS-CoV-2 and SARS-like coronavirus diversity in human, bat and pangolin populations. *Viruses*, **13**, 49.
Doñate-Macián,P. *et al.* (2018) The TRPV4 channel links calcium influx to DDX3X activity and viral infectivity. *Nat. Commun.*, **9**, 2307.
Dong,E. *et al.* (2020) An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect. Dis.*, **20**, 533–534.
Durrant,C. *et al.* (2011) Collaborative Cross mice and their power to map host susceptibility to *Aspergillus fumigatus* infection. *Genome Res.*, **21**, 1239–1248.
Dyer,M.D. *et al.* (2008) The landscape of human proteins interacting with viruses and other pathogens. *PLoS Pathogens*, **4**, e32.
Dyer,M.D. *et al.* (2010) The human-bacterial pathogen protein interaction networks of *Bacillus anthracis*, *Francisella tularensis*, and *Yersinia pestis*. *PLoS One*, **5**, e12089-12.
Eid,F.-E. *et al.* (2016) DeNovo: virus-host sequence-based protein–protein interaction prediction. *Bioinformatics*, **32**, 1144–1150.
Fagerberg,L. *et al.* (2014) Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Mol. Cell. Proteomics*, **13**, 397–406.
Fajardo,T. Jr. *et al.* (2015) Disruption of specific RNA–RNA interactions in a double-stranded RNA virus inhibits genome packaging and virus infectivity. *PLoS Pathogens*, **11**, e1005321–22.
Fawcett,T. (2006) An introduction to ROC analysis. *Pattern Recogn. Lett.*, **27**, 861–874.
Finlay,B.B. and Cossart,P. (1997) Exploitation of mammalian host cell functions by bacterial pathogens. *Science*, **276**, 718–725.
Fitzgerald,K.A. *et al.* (2003) IKKε and TBK1 are essential components of the IRF3 signaling pathway. *Nat. Immunol.*, **4**, 491–496.
Gkoutos,G.V. *et al.* (2018) The anatomy of phenotype ontologies: principles, properties and applications. *Brief. Bioinf.*, **19**, 1008–1021.
Gordon,D.E. *et al.* (2020) A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature*, **583**, 459–468.
Guzzi,P.H. *et al.* (2012) Semantic similarity analysis of protein data: assessment with biological features and issues. *Brief. Bioinf.*, **13**, 569–585.
Hoehndorf,R. *et al.* (2011) PhenomeNET: a whole-phenome approach to disease gene discovery. *Nucleic Acids Res.*, **39**, e119.
Hoehndorf,R. *et al.* (2013a) Mouse model phenotypes provide information about human drug targets. *Bioinformatics*, **30**, 719–725.
Hoehndorf,R. *et al.* (2013b) Systematic analysis of experimental phenotype data reveals gene functions. *PLoS ONE*, **8**, e60847.
Hoehndorf,R. *et al.* (2015a) Aber-OWL: a framework for ontology-based data access in biology. *BMC Bioinformatics*, **16**, 26.
Hoehndorf,R. *et al.* (2015b) The role of ontologies in biological and biomedical research: a functional perspective. *Brief. Bioinf.*, **16**, 1069–1080.
Huo,T. *et al.* (2015) Prediction of host – pathogen protein interactions between *Mycobacterium tuberculosis* and *Homo sapiens* using sequence motifs. *BMC Bioinformatics*, **16**, 100.
Jain,S. and Bader,G.D. (2010) An improved method for scoring protein–protein interactions using semantic similarity within the gene ontology. *BMC Bioinformatics*, **11**, 562.
Jarosinski,K.W. *et al.* (2012) Fluorescently tagged pUL47 of Marek's disease virus reveals differential tissue expression of the tegument protein in vivo. *J. Virol.*, **86**, 2428–2436.
Jones,K.E. *et al.* (2008) Global trends in emerging infectious diseases. *Nature*, **451**, 990–993.
Kafkas,Ş. *et al.* (2019) PathoPhenoDB, linking human pathogens to their phenotypes in support of infectious disease research. *Sci. Data*, **6**, 79.
Köhler,S. *et al.* (2009) Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *Am. J. Hum. Genet.*, **85**, 457–464.
Köhler,S. *et al.* (2019) Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. *Nucleic Acids Res.*, **47**, D1018–D1027.
Kulmanov,M., and Hoehndorf,R. (2020a) DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, **36**, 422–429.
Kulmanov,M. *et al.* (2020) Semantic similarity and machine learning with ontologies. *Brief. Bioinf.*, in press.
Kulmanov,M. and Hoehndorf,R. (2020b) DeepPheno: predicting single gene loss-of-function phenotypes using an ontology-aware hierarchical classifier. *PLoS Comput. Biol.*, **16**, e1008453–22.
Li,X. *et al.* (2020) Emergence of SARS-CoV-2 through recombination and strong purifying selection. *Sci. Adv.*, **6**, eabb9153.
Lundberg,S.M. and Lee,S.-I. (2017) A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, pp. 4768–4777.
Meehan,T.F. *et al.* (2017) Disease model discovery from 3,328 gene knockouts by The International Mouse Phenotyping Consortium. *Nat. Genet.*, **49**, 1231–1238.
Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In Burges,C.J.C. *et al.* (eds.) *Advances in Neural Information Processing Systems*, **Vol. 26**. Curran Associates, Inc., Red Hook, NY, pp. 3111–3119.
Mousavizadeh,L. and Ghasemi,S. (2020) Genotype and phenotype of COVID-19: their roles in pathogenesis. *J. Microbiol. Immunol. Infect.* in press.
Mukhopadhyay,A. *et al.* (2014) Incorporating the type and direction information in predicting novel regulatory interactions between HIV-1 and human proteins using a biclustering approach. *BMC Bioinformatics*, **15**, 26.
Oakes,J.A. *et al.* (2017) TBK1: a new player in ALS linking autophagy and neuroinflammation. *Mol. Brain*, **10**, 5.
Oellrich,A. *et al.* (2016) The digital revolution in phenotyping. *Brief. Bioinf.*, **17**, 819–830.
Perrin-Cocon,L. *et al.* (2020) The current landscape of coronavirus-host protein–protein interactions. *J. Transl. Med.*, **18**, 1–15.
Pesquita,C. *et al.* (2009) Semantic similarity in biomedical ontologies. *PLoS Comput. Biol.*, **5**, e1000443.
Ponting,C.P. (2001) Issues in predicting protein function from sequence. *Brief. Bioinf.*, **2**, 19–29.
Ran,Y. *et al.* (2016) Autoubiquitination of TRIM26 links TBK1 to NEMO in RLR-mediated innate antiviral immune response. *J. Mol. Cell Biol.*, **8**, 31–43.
Ribeiro,M.T. *et al.* (2016) "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 13–17, 2016, pp. 1135–1144.
Rodríguez-García,M.Á. *et al.* (2017) Integrating phenotype ontologies with phenomeNET. *J. Biomed. Semant.*, **8**, 58.
Samavarchi-Tehrani,P. *et al.* (2020) A SARS-CoV-2 – host proximity interactome. 10.1101/2020.09.03.282103..
Sayers,E.W. *et al.* (2009) Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, **37**, D5–D15.
Smaili,F.Z. *et al.* (2020) Formal axioms in biomedical ontologies improve analysis and interpretation of associated data. *Bioinformatics*, **36**, 2229–2236.
Smith,C.L. *et al.* (2018) Mouse genome database (MGD)-2018: Knowledgebase for the laboratory mouse. *Nucleic Acids Res.*, **46**, D836–D842.
Srivastava,N. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.
Szklarczyk,D. *et al.* (2019) STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.*, **47**, D607–D613.
The Gene Ontology Consortium. (2017) Expansion of the gene ontology knowledgebase and resources. *Nucleic Acids Res.*, **45**, D331–D338.
Villaveces,J.M. *et al.* (2015) Merging and scoring molecular interactions utilising existing community standards: tools, use-cases and a case study. *Database*, **2015**, bau131.

Wang,A. *et al.* (2017) Zika virus genome biology and molecular pathogenesis. *Emerg. Microbes Infect.*, **6**, e13.
Weitzman,M.D. *et al.* (2004) Interactions of viruses with the cellular DNA repair machinery. *DNA Repair*, **3**, 1165–1173.
Whisstock,J.C. and Lesk,A.M. (2003) Prediction of protein function from protein sequence and structure. *Q. Rev. Biophys.*, **36**, 307–340.
Woolhouse,M. *et al.* (2012) Human viruses: discovery and emergence. *Philos. Trans. R. Soc. B Biol. Sci.*, **367**, 2864–2871.
Yang,X. *et al.* (2020) Prediction of human–virus protein–protein interactions through a sequence embedding-based machine learning method. *Comput. Struct. Biotechnol. J.*, **18**, 153–161.
Zheng,Y. *et al.* (2020) Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) membrane (M) protein inhibits type I and III interferon production by targeting RIG-I/MDA-5 signaling. *Signal Transduct. Targeted Ther.*, **5**, 1–13.
Zhou,X. *et al.* (2018) A generalized approach to predicting protein–protein interactions between virus and host. *BMC Genomics*, **19**, 568.

viruses

MDPI

Article

Computational Analysis of SARS-CoV-2 and SARS-Like Coronavirus Diversity in Human, Bat and Pangolin Populations

Nicholas J. Dimonaco [1,*], Mazdak Salavati [2,*] and Barbara B. Shih [2,*]

[1] Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, Wales SY3 3FL, UK
[2] The Roslin Institute, Royal (Dick) School of Veterinary Studies, University of Edinburgh, Easter Bush, Midlothian EH25 9RG, UK
* Correspondence: nid16@aber.ac.uk (N.J.D.); mazdak.salavati@roslin.ed.ac.uk (M.S.); barbara.shih@roslin.ed.ac.uk (B.B.S.)

Abstract: In 2019, a novel coronavirus, SARS-CoV-2/nCoV-19, emerged in Wuhan, China, and has been responsible for the current COVID-19 pandemic. The evolutionary origins of the virus remain elusive and understanding its complex mutational signatures could guide vaccine design and development. As part of the international "CoronaHack" in April 2020, we employed a collection of contemporary methodologies to compare the genomic sequences of coronaviruses isolated from human (SARS-CoV-2; n = 163), bat (bat-CoV; n = 215) and pangolin (pangolin-CoV; n = 7) available in public repositories. We have also noted the pangolin-CoV isolate MP789 to bare stronger resemblance to SARS-CoV-2 than other pangolin-CoV. Following de novo gene annotation prediction, analyses of gene–gene similarity network, codon usage bias and variant discovery were undertaken. Strong host-associated divergences were noted in ORF3a, ORF6, ORF7a, ORF8 and S, and in codon usage bias profiles. Last, we have characterised several high impact variants (in-frame insertion/deletion or stop gain) in bat-CoV and pangolin-CoV populations, some of which are found in the same amino acid position and may be highlighting loci of potential functional relevance.

Keywords: coronavirus; hackathon; host-associated divergences; codon usage; variant discovery

Citation: Dimonaco, N.J.; Salavati, M.; Shih, B.B. Computational Analysis of SARS-CoV-2 and SARS-Like Coronavirus Diversity in Human, Bat and Pangolin Populations. *Viruses* **2021**, *13*, 49. https://doi.org/10.3390/v13010049

Academic Editor: Luis Martinez-Sobrido
Received: 3 December 2020
Accepted: 22 December 2020
Published: 30 December 2020

1. Background

The continued and increasing occurrence of pandemics that threaten worldwide public health due to human activity is often considered to be inevitable [1,2]. The COVID-19 (2019–current) pandemic caused by the emergence in Hubei, China, of what has now been identified as Severe Acute Respiratory Syndrome Coronavirus 2/Novel Coronavirus 2019 (SARS-CoV-2/2019-nCoV) by The Coronaviridae Study Group [3], has brought a number of questions regarding its transmission, containment and treatment to the urgent attention of researchers and clinicians. The urgency of such questions has spurred a number of atypical approaches and collaborations between experts of different fields and as such, this study was carried out as part of a "CoronaHack" hackathon event in April 2020 where the authors gained access to genomes and related metadata available at the time (December 2019–April 2020).

Viruses of the Coronaviridae family have long been studied and while there have been great advances in our understanding, each new emergence has brought about its own questions. Coronavirus consists of four genera: *Alphacoronavirus* (Alpha-CoV), *Betacoronavirus* (Beta-CoV), *Gammacoronavirus* and *DeltaCoronavirus*. Coronaviruses are a group of single-stranded, enveloped and extremely diverse RNA viruses which are known to have come into contact with humans numerous times over the past few decades alone [4]. At around 30 kb, they exhibit at least six Open Reading Frames (ORFs), with ORF1a/b comprising of approximately 2/3 of the genome which encodes up to 16 non-structural replicase proteins through ribosomal frame-shifting, and four structural proteins: membrane (M),

nucleocapsid (N), envelope (E) and spike (S) glycoprotein [5]. Coronaviruses have developed a number of different strategies to infiltrate their host cells. In human-associated CoVs, it has been shown that different parts of the human Angiotensin Converting Enzyme 2 (ACE2) can be bound to by their respective S proteins. Pathogens such as SARS-CoV-1 (Severe Acute Respiratory Syndrome Coronavirus) and MERS-CoV (Middle East Respiratory Syndrome Coronavirus) have shown Coronaviruses to be capable of presumed efficient adaptation to their human host and exhibit high levels of pathogenicity [6,7]. Interestingly, SARS-CoV-1 and MERS, which along with SARS-CoV-2 are both Beta-CoVs, exhibit only 79.5% and 50% sequence similarity, respectively, at the whole genome level to SARS-CoV-2, whereas SARS-CoV-2-like coronaviruses found in pangolins (pangolin-CoVs) and bat coronavirus (bat-CoV) RaTG13 (bat-RaTG13) are 91.02% and 96%, respectively [8]. The relationship of SARS-CoV-2 to other SARS-like coronaviruses, the possible role of bats and pangolins as reservoir species and the role of recombination in its emergence are of great interest [9]. Speculations around other intermediary hosts are also at play, which might have affected the ability for zoonotic transmission for SARS-CoV-2 to its human host [10]. Crucially, this evolutionary relationship between SARS-CoV-2 and its lineage may prove to be an important factor in the eventual management or containment of the virus. Moreover, the mutation events along the evolutionary timeline of SARS-CoV-2 are of importance in the discovery of possible adaption signatures within the viral population. At the time of the hackathon, there were two main suspected SARS-like reservoir host species: bat and pangolin (named bat-CoV and pangolin-CoV) .

With this in mind, our study aimed to systematically compare a broad selection of contemporary available SARS-CoV-2, bat-CoV and pangolin-CoV at genome, gene, codon usage and variant levels, without preference for strains or sub-genera. This was comprised of 46 SARS-CoV-2 genomes isolated early in the pandemic from Wuhan, China (Late 2019–Early 2020); 117 SARS-CoV-2 genomes isolated in Germany, representing the later stage of global transmission; 215 bat-CoV genomes of Alpha-CoVs and Beta-CoVs; and seven pangolin-CoV genomes, of which five were annotated as Beta-CoVs. During the hackathon, it was recognised that potential biases can arise from directly comparing SARS-CoV-2 to a wide repertoire of coronaviruses of varying stages of genome annotation. Therefore, we performed a new comparative annotation of all genomes used in this study. To further validate mutational adaptations which may have facilitated the zoonotic transmission of SARS-CoV-2, a codon usage analysis was carried out between the SARS-CoV-2 reference genes and the genes identified using the aforementioned approaches. In addition, we profiled codon usage bias across our data set, as in the process of host adaptation, viruses can evolve to express different preferential codon usages [11–13].

Through examining the inherent sequence diversity between a comprehensive collection of SARS-CoV-2, bat-CoV and pangolin-CoV, we aimed to highlight naturally occurring high impact variations that can potentially introduce a change in the resulting protein, such as the insertion or deletion of an amino acid or early termination of the sequence. Understanding the stability and variability of these positions may potentially aid future design of vaccines or treatments. For instance, an amino acid position where insertion or deletion is commonly found in a coronavirus affecting other species may indicate that its alteration does not have a dramatic impact on the overall protein folding, or that the position is important for transmission to a new host.

Our work is differentiated by the way of a systematic approach was used to process a non-selective group of these viral genomes from public repositories, prior to applying a wide range of contemporary methodologies and genomic knowledge that highlight the variations that exist between different host species. Understanding the current limitations of annotation pipelines and available curated SARS-CoV-2 genomes was the main driver of this approach. Providing a comprehensive gene and variant annotation for viral genomes collected from multiple hosts will bridge this knowledge gap in the literature.

2. Results

2.1. Data Collection and Phylogenetic Analysis

We were able to collate 215 bat-CoV genomes of varying families (Alphacoronaviruses and Betacoronaviruses) with only one exhibiting a small proportion or genomic uncertainty (presence of 0.45% "N" nucleotide). However, only seven pangolin-CoV genomes, of which five were annotated as Betacoronaviurs, were available at the start of this study. Three pangolin-CoV genomes also contained levels of the ambiguous "N" nucleotide, two of them at high levels (6.88 and 8.19%). A population of post-outbreak SARS-CoV-2 genomes from Charite [14], Germany, were also collated for further analysis. For the phylogenetic analysis, we examined the complete set of 269 genomes (seven pangolin-CoV; 47 SARS-CoV-2, including the reference genome; and 215 bat-CoV). The phylogenetic tree produced at the whole genome level showed a clear separation between the SARS-CoV-2 and the bat-CoV genomes, with the exception of bat-RaTG13 which has been placed adjacent to the SARS-CoV-2 clade (Figure 1). The similarity of bat-RaTG13 to SARS-CoV-2 has previously been reported [15]. While more distantly related to SARS-CoV-2 than bat-RaTG13, MG772933 and MG772934 (bat-SL-CoVZC45 and bat-SL-CoVZXC21 isolates) are more closely related to SARS-CoV-2 than the remaining bat-CoV (Figure 1). Six of the seven pangolin-CoV genomes are grouped together and closest to the SARS-CoV-2 clade, other than bat-RaTG13. One pangolin-CoV, MT084071.1 (MP789 isolate; referred to as pangolin-MP789), is placed in a branch closer to SARS-CoV-2 than the remaining pangolin-CoV in the tree (Figure 1). The tree produced was used as an analytical anchor for which we could use to refer to in the results from variant analysis. High impact variants were annotated on the tree to show their distribution across the different clades along the topology of the tree.

2.2. Gene Identification

For each viral genome, a complementary approach using both PROKKA [16] and BLAST [17] was employed for identifying genes highly similar to those in the SARS-CoV-2 reference genome released by Ensembl v100 (SARS-CoV-2 ref). The breakdown of this result is shown in Table 1, and Table A3 presented a detailed account of the genes annotated in each genome and their corresponding annotation tools (PROKKA or BLAST).

PROKKA, which is an alignment-free method, was unable to capture some genes in some of the genomes; BLAST-alignment was used to address this. This has enabled the characterisation of E and ORF10 in many genomes. Genes utilising ribosomal frame-shifting, such as the aforementioned ORF1ab, are inherently difficult to identify correctly without extensive analysis involving techniques and evidence such as RNA expression analysis. For the majority of genomes studied, PROKKA was able to identify two large ORFs spanning almost the entire length of the ORF1ab locus and detect a central coronavirus frame-shifting stimulation element (named Corona_FSE and separating the two ORFs) which is a conserved stem-loop of RNA found in coronaviruses that can promote ribosomal frame-shifting [18]. The gene sequences generated by PROKKA and BLAST (E and ORF10) were used for downstream analysis, including gene–gene network graph, codon usage bias analysis and a gene presence summary table. The gene presence summary table notates whether SARS-CoV ref genes were found ($\geq$80% percentage identity and $\geq$50% sequence coverage) in each genome; this table is available in the GitHub project https://github.com/coronahack2020/final_paper/tree/master/host-data.

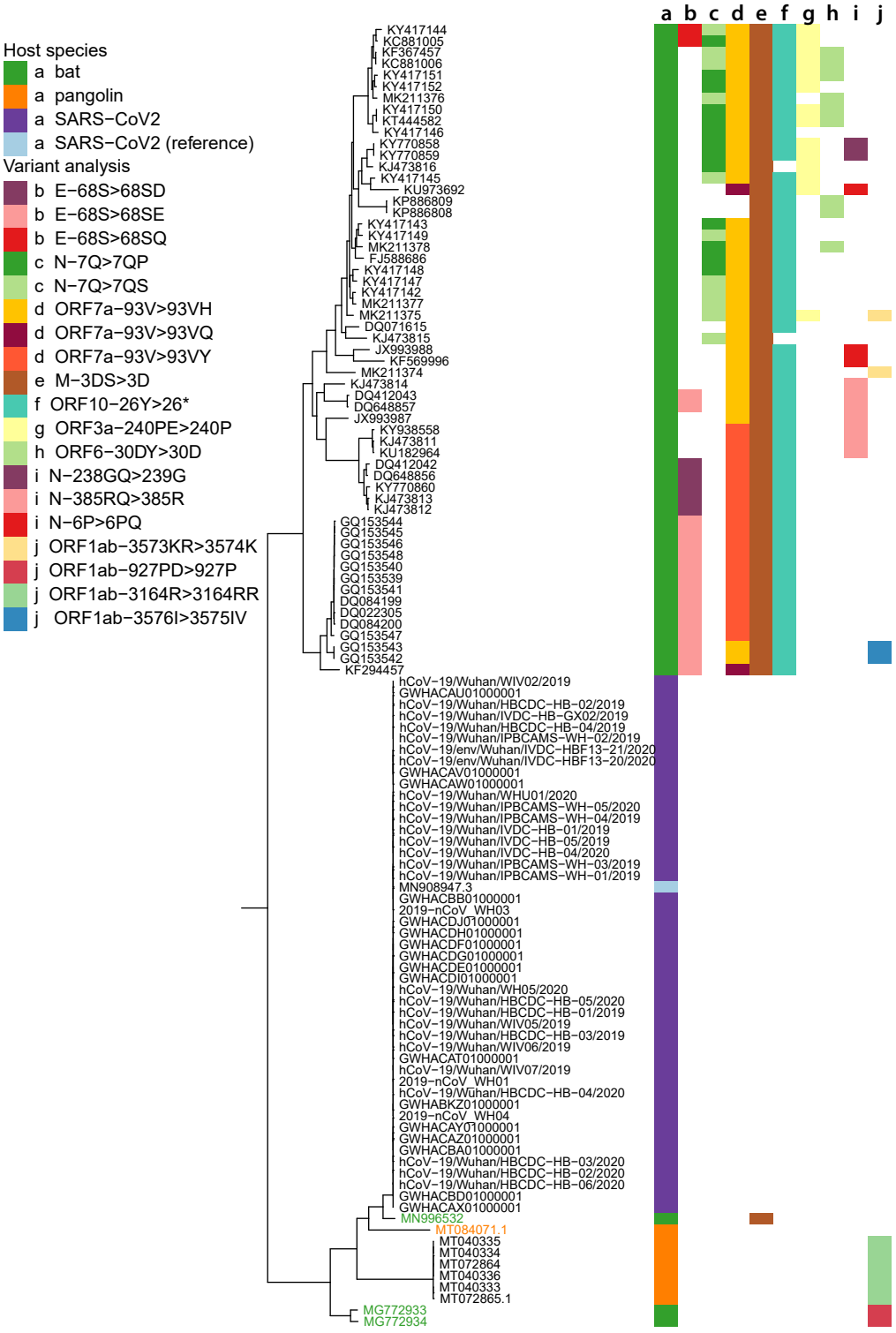


Figure 1. Phylogenetic tree showing relationship between bat-CoV, pangolin-CoV and SARS-CoV-2. This is the Sarbecovirus clade from Figure A5, the phylogenetic tree made with all bat-CoV, all pangolin-CoV and SARS-CoV-2 (Wuhan dataset and SARS-CoV-2 reference) used in this study. Along with the (a) host organisms, results from the variant analysis are annotated, showing (b–d) positions with multiple amino acid changes, (e–h) positions with a single amino acid change (in >10 genomes) and (i,j) other variants. The genes and amino acid changes involved in each of the annotated inframe insertion, inframe deletion or stop gain (*) are indicated in the figure legend. The names of four genomes are highlighted, including 3 bat-CoV—MN996532 (bat-RaTG13), MG772933 (bat-SL-CoVZC45) and MG772934 (bat-SL-CoVZXC21)—and 1 pangolin-CoV, MT084071.1 (pangolin-MP789), as they are more closely related to SARS-CoV-2 than the other bat-CoV or pangolin-CoV in the tree.

Table 1. This table presents the distribution of the number of predicted genes for each dataset. Bat-CoV exhibit the widest distribution of gene count, and pangolin-CoV has the highest number of gene count, with one genome having 17 predicted genes. These outliers have low sequence or assembly quality. In the case of the pangolin-CoV genome reporting 17 genes, it has low-quality ("NNNN") nucleotide regions spanning the centre of genes, which causes PROKKA to identify the two ends of one gene. The median gene count only varying in bat-CoVs, likely attributed to the large phylogenetic variation exhibited across the bat-CoVs.

Dataset	Min.	Median	Mean	Max.	Sample Count
Wuhan	7	11	11	13	46
Charite	9	11	11	12	117
Bat	2	9	9	12	215
Pangolin	10	11	12	17	7

2.3. Gene Relationship Network Graph

A gene–gene similarity network analysis was used to compare genes across SARS-CoV-2, bat-CoV and pangolin-CoV. The advantage of using a 3D network approach to visualise this information was that it simplifies complex information as patterns. Genes sharing high similarity form independent clusters. In cases where there is a high degree of dissimilarity in a gene for different host species, a pattern of 2 or more distinct clusters would take place, with each cluster comprised of genes derived from samples of the same host species. In genes where there is a medium level of dissimilarity across host species, two or more cluster would appear fused and potentially break apart into distinct clusters if the edge threshold were increased. Both of these patterns are observed within this dataset. Distinct separation by host species are seen in ORF1a, ORF3a,ORF6, ORF7a, ORF8 and S (Figure 2). The strongest host–species separation observed was between SARS-CoV-2 and bat-CoV; pangolin-CoV always grouped closer to SARS-CoV-2 than to bat-CoV, with the exception of bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13. In the cases of ORF3a, ORF8 and S, complete separation was observed between bat-CoV and human SARS-CoV-2 (Figure 2B,C). Bat-RaTG13 was more similar to SARS-CoV-2 and pangolin-CoV than the remainder of the bat-CoV for S (Figure 2C). For ORF3a, bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13 clustered together with SARS-CoV-2 and pangolin-CoV rather than with the remainder of the bat genomes (Figure 2). These same three genomes are the only bat-CoV with ORF8 that co-cluster with SARS-CoV-2 ORF8 under the percentage identity threshold ($\geq$80%) set for building the network graph. Other bat-CoV ORF8 were so distinct from SARS-CoV-2 ORF8 that they do not form edges with SARS-CoV-2 ORF8. Interestingly, within the cluster of ORF8 sequences, the ORF8 for pangolin-MP789 shares an average of 92.14% identity to SARS-CoV-2 ORF8, while the ORF8 for remaining pangolin-CoV do not share a strong similarity to the SARS-CoV-2 ref ORF8 (no BLAST result). An average percentage of identity between SARS-CoV-2 ORF8 and bat-CoV ORF8 are 97.05% (bat-RaTG13) and 88.58% (bat-SL-CoVZC45 and bat-SL-CoVZXC21).

To investigate whether if potential gene transfer or recombination that may have come from more distantly related bat-CoV, we sought for unusual co-clustering between genes characterised from bat-CoV and SARS-CoV-2. We did not observe such pattern; bat-RaTG13 co-cluster with SARS-CoV-2 for many genes and is also the most similar bat-CoV to SARS-CoV-2 at a genome level. Two additional genes identified by PROKKA, Corona FSE, a non-coding frame-shift stimulation element within ORF1ab and s2m, a stem-loop II-like motif [19] have both been shown to be highly conserved and important for SARS-2-like coronaviruses. s2m has been identified as a mobile genetic element which has been described in a number of single-stranded RNA virus and insect families and has also been shown to be important for viral function [20,21].

In summary, the use of gene–gene network analysis enables us to determine groups of closely related genes, which not only highlights genes showing strong host–species separation, but also characterise clusters of related genes that may be absent or highly

different from the reference genome of interest, such as ORF8. Six genes—ORF1ab, ORF3, ORF6, ORF7a, ORF8 and S—showed a strong host–species separation in the network graph. In particular, with the exception of S, where bat-SL-CoVZC45, bat-SL-CoVZXC21 clustered closer to bat-CoVs, the bat genomes, bat-SL-CoVZC45, bat-SL-CoVZXC21 and bat-RaTG13, clustered together with SARS-CoV-2 than the remainder of the bat-CoV for these 5 genes.

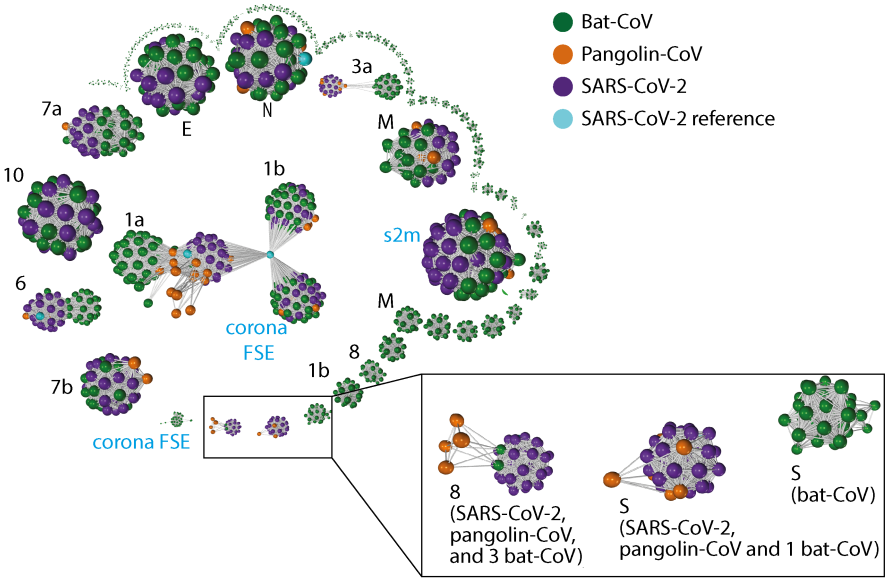


Figure 2. Gene–gene similarity network analysis. Each node represents a gene defined by PROKKA or a DNA segment similar to genes from the SARS-CoV-2 reference genome. The nodes were compared against each other using BLAST, and nodes with high similarity (bit score $\geq$ 60 and a query coverage $\geq$ 80%) were connected with an edge. The network graph is labelled with host species. The black font in the graph indicates the corresponding SARS-CoV-2 gene names ("Open Reading Frame (ORF)" omitted) for the larger clusters, whereas blue font indicate additional non-coding sequences defined by PROKKA. Instead of the full length ORF1ab (21kb in length), ORF1a and ORF1b were defined by PROKKA as two separate genes. Notably, ORF1a, ORF3a, ORF6, and ORF8 and S show strong separations between nodes from different species. ORF8 from 3 bat-CoV co-clusters with ORF8 from SARS-CoV-2 (bat-RaTG13, bat-SL-CoVZC45 and bat-SL-CoVZXC21, respectively). The remaining bat-CoV ORF8 do not co-cluster with SARS-CoV-2 ORF8 even without the edge filtering threshold. For S, the bat-RaTG13 co-cluster with COVID-19 and pangolin. A cluster of bat-CoVs break off for ORF1b and M, suggesting a large amount of variation amongst bat-CoV for these genes.

2.4. Codon Usage Bias

We examined Relative Synonymous Codon Usage (RSCU) across SARS-CoV, bat-CoV and pangolin-CoV for each SARS-CoV-2 reference gene. Principle component analysis (PCA) using RSCU showed a strong host–species separation; the first principle component (PC1) accounts for 55.62–85.38% of variation (Figure 3), predominately separating SARS-CoV-2 from bat-CoV. Bat-RaTG13, bat-SL-CoVZC45 and bat-SL-CoVZXC21 and pangolin-CoV are usually placed between SARS-CoV-2 and other bat-CoV. With the exception of ORF7b, Pangolin-MP789 is placed closer to SARS-CoV-2 than all other pangolin-CoV (Figure 3) with regards to the variation described by PC1 and PC2.

K-means clustering was used to group the genomes into three clusters for each gene using the first 10 PCs, which have grouped pangolin-MP789 with SARS-CoV-2 for ORF1a, ORF8, ORF7a, E, ORF6 and N (one of two assemblies) Figure A7. For M and ORF3a, pangolin-MP789 clustered with bat-SL-CoVZC45 and bat-SL-CoVZXC21 Figure A7.

A summary of the synonymous codon ratios (the number of codons divided by the total number of codons coding for the same amino acid), sorted by amino acids, are shown in Figure A8.

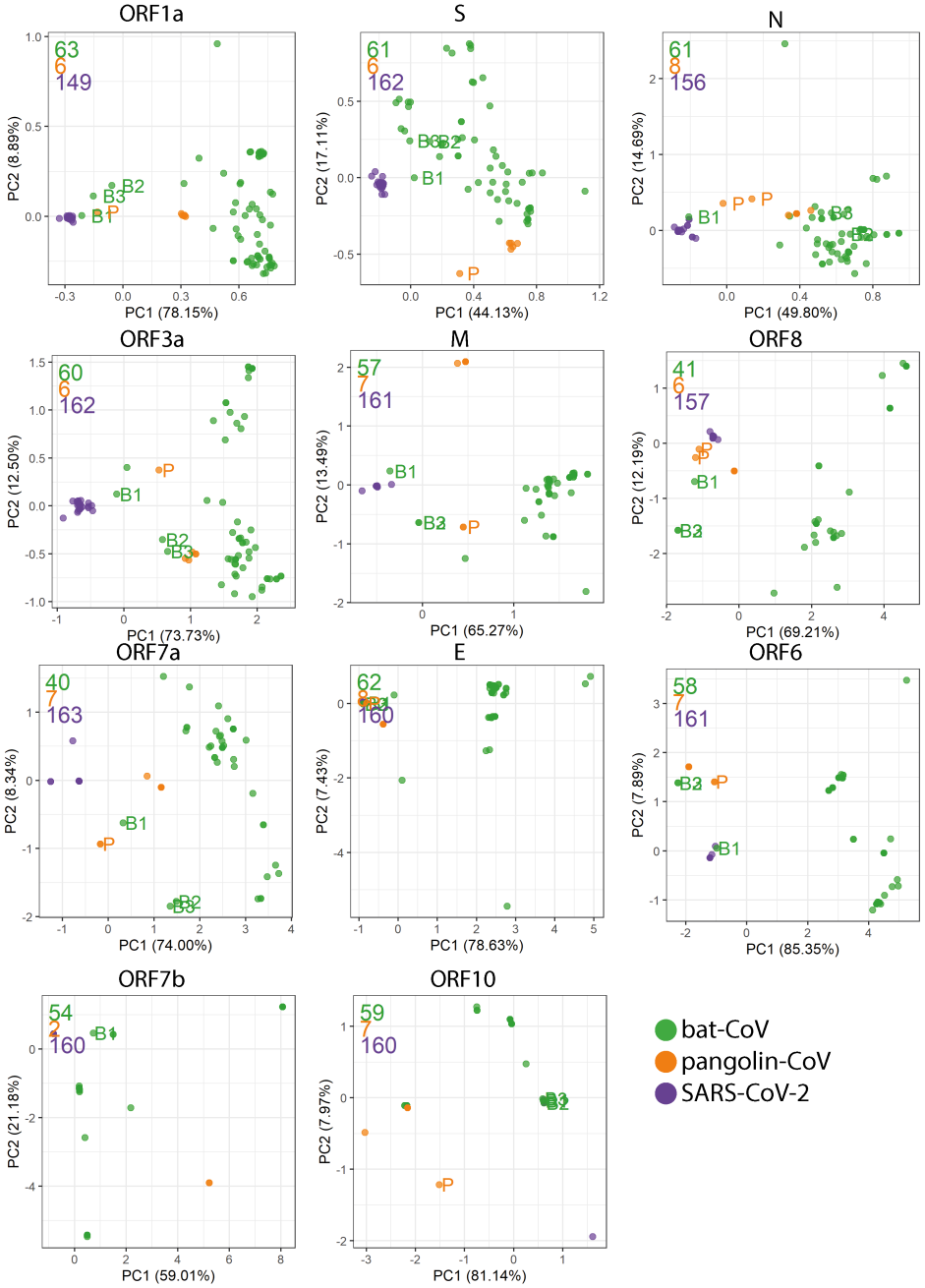


Figure 3. Relative synonymous codon usage (RSCU) was calculated as the ratio of the observed frequency of codon to the expected frequency under the assumption of equal usage between synonymous codons for the same amino acids. For each gene, Principal Component Analysis (PCA) was carried out on the RSCU values. The first two Principal Components (PCs) are plotted. The total number of genomes used in each plot are indicated in the top left corner in the corresponding colour. In order, they are bat-CoV (green), pangolin-CoV (orange) and SARS-CoV-2 (purple). Four isolates are labelled: bat-RaTG13 (B1), bat-SL-CoVZC45 (B2), bat-SL-CoVZXC21 (B3) and pangolin-MP789 (P; MT121216.1 and MT084071.1).

2.5. Variant Analysis

Haplotype-aware variant calling and variant effect prediction of all genomes in the study have been summarised in Figure 4 and Supplementary File. There is a total of 1127 variants that are missense, inframe deletion, inframe insertion, stop gained and stop lost, as can be seen in Figure A9. We have removed missense from further analysis and came to a total of 24 high impact variations in eight genes were when comparing

bat-CoV and pangolin-CoV genomes against the SARS-CoV-2 ref. We have annotated the majority (with the exception of the NC045512_27675A>ACAG) of these variation in Figure 1, and found that some of these variations, such as variants identified in E, ORF7a and ORF3a, appear to exhibit some degree of clade specificity. The only stop gain variant (i.e., NC045512_29635) was present in ORF10 gene of 57 bat-CoV genomes (29,635 bp position C > A) which was only representing a synonymous variant in the same position of six pangolin-CoV genomes. This variant affected 26Y > 26* (Tyrosine to STOP codon TAC > TAA) in bat ORF10. Assuming the direction of host selection from bat and pangolin to human, this variant could explain the presence of a longer ORF10 isoform in the two latter hosts in comparison to bat-CoV. From the list of variants in Figure 4, four in-frame insertions were identified as follows:

- ORF1ab gene at position 9757 (NC045512_9757 T>TAGA 3164R>3164RR) of all pangolin-Cov genomes which represents an extra Arginine.
- E gene at position 26448 (NC045512_26448 T>TGAA 68S>68SE) in 33 bat-Cov genomes which caused an addition of Glutamine.
- ORF7a gene at position 27672 (NC045512_27672 T>TCAC 93V>93VH) in 24 bat-Cov genomes by addition of an Histamine.
- N gene at position 28293 (NC405512z_28293 A>AACC 7Q>7QP) in 13 bat-Cov genomes by addition of a Proline.

Two in-frame deletions were also identified in ORF3a and M genes. A single Glutamine deletion in ORF3a at position 26,111 was present in 14 bat-Cov genomes (NC045512_26111 CTGA > C 240PE > 240P) and a Serine deletion in M gene at position 26,530 (NC045512_26530 ATTC > A 3DS > 3D) was present in 57 bat-Cov genomes. The same position showed a missense mutation of 3D > 3A (in two bat-Cov [bat-SL-CoVZC45 and bat-SL-CoVZXC21] and one pangolin-Cov) and 3D > 3G in six pangolin-Cov genomes.

CHROM	POS	REF	ALT	VAC	consequence	gene_name	amino_acid_change	dna_change	AF	host
NC_045512	3045	CAGA	C	2	inframe_deletion	ORF1ab	927PD>927P	3045CAGA>C	0.01739130	Bat
NC_045512	9757	T	TAGA	6	inframe_insertion	ORF1ab	3164R>3164RR	9757T>TAGA	0.05217391	Pangolin
NC_045512	10983	AAAG	A	2	inframe_deletion	ORF1ab	3573KR>3574K	10983AAAG>A	0.01739130	Bat
NC_045512	10993	C	CGTT	2	inframe_insertion	ORF1ab	3576I>3575IV	10993C>CGTT	0.01739130	Bat
NC_045512	26111	CTGA	C	14	inframe_deletion	ORF3a	240PE>240P	26111CTGA>C	0.12173913	Bat
NC_045512	26447	C	CTGA	5	inframe_insertion	E	68S>68SD	26447C>CTGA	0.04347826	Bat
NC_045512	26448	T	TGAG	16	inframe_insertion	E	68S>68SE	26448T>TGAG	0.13913043	Bat
NC_045512	26448	T	TGAA	33	inframe_insertion	E	68S>68SE	26448T>TGAA	0.28695652	Bat
NC_045512	26448	T	TCAA	2	inframe_insertion	E	68S>68SQ	26448T>TCAA	0.01739130	Bat
NC_045512	26530	ATTC	A	57	inframe_deletion	M	3DS>3D	26530ATTC>A	0.49565217	Bat-Pangolin
NC_045512	27289	GATTAC	GAC	7	inframe_deletion	ORF6	30DY>30D	27289GATTAC>GAC	0.06086957	Bat
NC_045512	27291	TTAC	T	2	inframe_deletion	ORF6	30DY>30D	27291TTAC>T	0.01739130	Bat
NC_045512	27671	T	TTTA	11	inframe_insertion	ORF7a	93V>93VY	27671T>TTTA	0.09565217	Bat
NC_045512	27671	T	TTCA	10	inframe_insertion	ORF7a	93V>93VH	27671T>TTCA	0.08695652	Bat
NC_045512	27672	T	TCAC	24	inframe_insertion	ORF7a	93V>93VH	27672T>TCAC	0.20869565	Bat
NC_045512	27672	T	TTAC	8	inframe_insertion	ORF7a	93V>93VY	27672T>TTAC	0.06956522	Bat
NC_045512	27672	T	TCAG	2	inframe_insertion	ORF7a	93V>93VQ	27672T>TCAG	0.01739130	Bat
NC_045512	27675	A	ACAG	2	inframe_insertion	ORF7a	94Q>94QQ	27675A>ACAG	0.01739130	Bat
NC_045512	28291	C	CCAA	3	inframe_insertion	N	6P>6PQ	28291C>CCAA	0.02608696	Bat
NC_045512	28293	A	AACC	13	inframe_insertion	N	7Q>7QP	28293A>AACC	0.11304348	Bat
NC_045512	28293	A	AATC	11	inframe_insertion	N	7Q>7QS	28293A>AATC	0.09565217	Bat
NC_045512	28987	CCAA	C	2	inframe_deletion	N	238GQ>239G	28987CCAA>C	0.01739130	Bat
NC_045512	29428	ACAG	A	7	inframe_deletion	N	385RQ>385R	29428ACAG>A	0.06086957	Bat
NC_045512	29635	C	A	57	stop_gained	ORF10	26Y>26*	29635C>A	0.49565217	Bat

Figure 4. High impact variants identified across bat and pangolin genomes using the variant calling pipeline based on SARS-Cov-2 Ensembl reference genome. The variants with allele frequency > 0.1 and predicted to have HIGH impact using VEPTools are listed. CHROM: Reference contig name; POS: Position; REF: Reference allele in Ensembl Human SARS-Cov2; ALT: Alternative allele(s) found in non-human genomes; VAC: Alternative variant allele counts; AF: Allele frequency.

3. Discussion

During the 5-day hackathon, we endeavoured to utilise the genomic data aggregated by the scientific community and undertook a multifaceted and comprehensive exploration of the genomic sequences (or "similarities and differences") of coronaviruses infecting bat and pangolin hosts, available at the time. We have compared SARS-Cov-2 to all bat-CoV and pangolin-CoV genomes from the listed data repositories (NCBI, VIPR and Databiology) without selecting for strains to represent any specific genera, species or substrain. Our comparisons spanned across several levels: whole-genome, genes, codons and individual variants.

The origin of SARS-CoV-2 is still unknown and a number of coronaviruses from different hosts have been proposed as the potential common ancestors [22,23]. However, bats are often linked to SARS-like viruses capable of zoonotic host transfer due to their unique niche as viral reservoirs. This is often characterised by their physiology relatively unaffected under varying viral loads and their natural proximity to human habitation [24,25]. Furthermore, recombination has been suggested as an avenue for host transfer for a number of RNA viruses such as SARS-CoV-1 and MERS [26,27].

The phylogenetic tree inferred from genomes studied in this manuscript presents a picture of vast bat-CoV diversity and its topology is similar to those of previous studies carried out on pangolin and bat coronaviruses when compared to the SARS-CoV-2 genome [28]. Previous phylogenetic profiling has noted that bat-RaTG13 bares the closest resemblance to SARS-CoV-2 across 55 SARS-like coronavirus genomes [29]. Of the the 222 SARS-like coronavirus genomes we have constructed the phylogenetic tree with, bat-RaTG13 remains the closest to SARS-CoV-2, followed by pangolin-MP789, the remaining six pangolin-CoV, and then bat-SL-CoVZC45 and bat-SL-CoVZXC21. The relationships between pangolin-MP789 and the three aforementioned bat-CoVs have been described [30], but it has not yet been highlighted that pangolin-MP789 is closer to SARS-CoV-2 than the other known pangolin-CoV (Figure 1). This relationship has previously been reported and a recombination event between pangolin-CoVs and bat-RaTG13 has been theorised [31].

As well as at genome level, the similarity of bat-RaTG13 and pangolin-MP789 to SARS-CoV-2 is also evident at gene level, in particular, across ORF8 sequences. Only a few closely related SARS-CoV-2 ORF8 orthologues have been identified within bat-betacoronavirus lineages [32,33]. We have shown the pangoling-MP789 and bat-RaTG13 ORF8 gene has $\geq$90% sequence identity to the SARS-CoV-2 ref ORF8. The exact function of ORF8 remains to be elucidated, although studies on ORF8 from SARS-CoV-2 and ORF8ab and ORF8b from SARS-CoV-1 have suggested a role in immune modulation through the interferon signalling pathway [34,35] and inducing strong antigen response [36]. Although the origin or function of the SARS-related coronavirus ORF8 remains unresolved, a 29-nucleotide deletion in ORF8 is often found in SARS-CoV-1, when compared to civet-CoV, suggesting that ORF8 may be important for interspecies transmission [37].

Other genes that show strong host-species separation in the gene–gene network analysis include ORF1a, ORF3a, ORF6 and S. It has been previously shown that pangolin-CoV and SARS-CoV-2 S proteins were highly similar to each other (97.5%) [38]. Furthermore, it has been shown that the overall structure of S protein in bat-RaTG13 is highly similar to those in SARS-CoV-2 [39]. This is significant as the S protein plays an important role in the initial penetration and infection of host cells and are often host-specific [40]. Viruses, through co-evolution with the host have high degrees of flexibility in their receptor usage and capacity to reach binding efficiencies via mutations [41,42] Several human coronaviruses, including SARS-CoV-2, SARS-CoV-1 and human coronavirus NL63 (hCoV-NL63), penetrate the host cell by binding to the host ACE2 through the receptor binding domain (RBD) of S protein [43,44]. It would appear that despite the S protein being more similar between pangolin-CoVs and SARS-CoV-2, the S protein in bat-RaTG13 is still more similar to that of SARS-CoV-2 than other bat-CoVs in our study (Figure 2C). This raises the possibility that the most recent common ancestor of SARS-CoV-2 (be of pangolin-CoV or bat-CoV origins) is yet to be sequenced.

Codon usage bias across the species–host range may show signs of preferential codon mutation which have occurred during the complex process of host interaction and transfer [11,12]. The knowledge of nucleotide profiles and subsequent codons during the human–virus co-evolution could be invaluable to the design of vaccines and their continuous development over the years to come [45]. On the whole, the codon usage profiles are highly different between SARS-CoV-2 and the majority of bat-CoV, with bat-RaTG13, bat-SL-CoVZC45, bat-SL-CoVZXC21 and panolin-CoV positioned between the two groups. Similar to the analysis by Gu et al. (2020), we found the codon usage profiles in bat-RaTG13 to be most similar to SARS-CoV-2 on the whole [46]. However, we have included six additional pangolin-CoV isolates in our studies and found pangolin-MP789 exhibited consistently more similar codon usage profiles to SARS-CoV-2 than the remaining pangolin-CoV at the gene level, which is also reflected in the genome-level phylogenetic tree. These observations highlighted the variation within pangolin-CoV and the closer resemblance between pangolin-MP789 and SARS-CoV-2; pangolin-MP789 is an isolate collected in 2019, whereas all other pangolin isolates were collected prior to 2019. Our codon usage analysis has focused on the overall comparison of RSCU for each gene across bat-CoV; other studies have compared gene sequence characteristics such as GC content and CpG dinucleotide [47–49].

Next, we focused on variants that could potentially have a more profound impact on the amino acid substitution or early stop codon gains (i.e., truncation). Population-level viral mutation is a complex process, involving a number of pressures, and while RNA viruses often exhibit some of the highest mutation rates of all viruses, conserved variants can exhibit important functional changes such as the ability to evade immunity more efficiently [50]. Furthermore, unlike the vast majority of RNA viruses, coronaviruses encode a complex RNA-dependent RNA polymerase that has a 3′ exonuclease domain [51], effectively proofreading mutational events and therefore are less error-prone. Therefore, the mutations observed across populations have undergone an error-correction process which means they are more likely to be functionally beneficial to the virus.

We have observed several of such variants (allele frequencies > 0.1) that are at consistent loci across different bat-CoV clades as shown in Figure 1. Some of these variants are seen in the majority of the bat-CoV samples (which align to the SARS-CoV-2 ref), including a stop-gain for ORF10 and an in-frame deletion for M, whilst others, such as the variants seen in ORF7a and E appear to be clade specific (Figure A5). Several of these variants affect the same amino acid positions, including E (in-frame insertion of *Asp* (Aspartic acid), *Glu* (Glutamic acid) or *Gln* (Glutamine) at at positions 68), N (inframe insertion of *Pro* (Proline) or *Ser* (Serine) at position 7) and ORF7a (in-frame insertion of *His* (Histidine), *Gln* or *Tyr* (Tyrosine) at position 93) (Figure A5). Notably, the stop-gain was identified at amino acid position 26 in ORF10 for 57 of the 59 bat-CoV genomes with ORF10 that had $\geq$80% similarity to the SARS-CoV-2 ref. The absence of this stop codon in the pangolin (which exhibited synonymous mutations at the same locus) and SARS-CoV-2 viruses could result in a longer isoform of the ORF10 or fundamental changes in its function and expression levels. In a previous study of SARS-CoV-2 and pangolin-CoV genomes, position 26 was also identified as a region of population level variation from *Tyr* and *His* which significantly modifies the secondary structure of the coil region of the protein [52].

There has been little research on ORF10 function, and its expression has been the subject of debate. Whilst Kim et al. (2020) found little evidence of ORF10 expression (0.000009% of viral junction-spanning reads) in cell culture (Vero cells) [53], Liu et al. (2020) found it to be abundantly expressed in severe COVID-19 patient cases but barely detectable in moderate cases [54]. Besides the single ORF10 variant that is observed in the majority of the bat-CoV, we have observed three different amino acid insertions (four different nucleotide changes) at position 68 of E gene in four different clades of bat-CoVs.

The small envelope E protein is the smallest of coronaviruses' major structural proteins, but also one of the least described [55]. E gene has been shown to be highly expressed inside infected cells and the viruses which are formed without E exhibit reduced levels

of viral maturation and tropism. Expression of the E product was essential for virus release and spread, thus demonstrating the importance of E in virus infection and therefore vaccine development [56]. The 68th amino acid position we highlight in this study is in the c-terminal domain, which coincides with the previously reported motif in SARS-CoV-1 (also at 68th amino acid position) that binds to the host cell PALS1 protein to facilitate infection [57]. Less than 0.5% of 3617 SARS-CoV-2 genomes have been found to have non-synonymous mutation in E, and of these, 20% are at the 68th amino acid position [58]. These changes in amino acid may alter the hydrophobicity at the locus, thus possibly influencing the protein functions and interactions [58]. Two of the E variants we highlighted use different codons for the same amino acid (GAG or GAA for *Glu*), which potentially suggests interplay between the selection pressures of codon optimisation and amino acid insertion into the protein product.

We have characterised a number of in-frame insertions at the amino acid position 93 in ORF7a across 55 bat-CoV genomes, and at position 94 reported in two. As with position 68 in E, position 93 in ORF7a has multiple codon insertions coding for the same amino acid but in two groups. In these two groups of bat-CoVs, an additional *His* is encoded for by two different codons and secondly, so is *Tyr* in another group. Intriguingly, ORF7a in SARS-CoV-1 has been shown to regulate the bone marrow stromal antigen 2 which inhibits the release of virions of human-infecting viruses [59].

N is another gene for which we have shown multiple in-frame insertion variants for the same amino acid position. The N protein is highly expressed during an infection, and it plays a key role in promoting viral RNA synthesis and incorporating genomic RNA into progeny viral particles [60]. In gene N, we observed two in-frame insertions at amino acid position 7 for *Ser* or *Pro* from two groups of bat-CoVs (13 and 11 respectively), as well as two in-frame deletions at positions 238 and 385. For M in 57 bat-CoV and pangolin-CoV, there is an in-frame deletion at position 3, which removed the amino acid *Ser*. At this amino acid position, a missense mutation of (Asp) to Glycine (Gly) is seen in 2 bat-CoV (bat-SL-CoVZC45 and bat-SL-CoVZXC21) and pangolin-MP789, and (Asp) to *Arg* in the remaining 6 pangolin-Cov genomes. Bat-SL-CoVZC45, bat-SL-CoVZXC21 and pangolin-MP789 have been shown to be more similar to SARS-CoV-2 than other coronavirus of the same host on other comparative metrics. M plays an important role in its interactions with both E and S to incorporate virions into the host cells.

The amino acid positions we have highlighted through our variant analysis may constitute important differences in the function or folding potential of the protein product. We have summarised the polymorphism along with respective allele frequencies and amino acid consequences in Figure 1. Weber et al. (2020) have interrogated 572 SARS-CoV-2 genomes isolated worldwide and characterised 10 distinct mutation hotspots that have been found in up to 80% of the viral genomes they examined [61]. While our reported variant positions are not 100 % in concordant with these hotspots, some of them display changes on or adjacent to our reported positions.

Through employing a number of genomic analysis methodologies, this study has aimed to bring understanding of the diversity across SARS-CoV-2 and SARS-CoV-2-like coronaviruses by comparing a wide selection of available genomes from the (early stages) starting point of the pandemic. We have highlighted a high degree of host-species separation in sequence homology for ORF3a, ORF6, ORF7a, ORF8 and S, as well as codon usage. Along with bat-RaTG13, we have highlighted the pangolin-MP789 isolate to bare stronger resemblance to SARS-CoV-2 than other pangolin-CoV in both whole-genome phylogenetic tree and gene-level codon usage profiling. Furthermore, a number of amino acid positions that demonstrate high impact variants (inframe insertion/deletion or stop gain) have also been identified in various bat-CoV and pangolin-CoV; these are potentially functionally important positions that warrant further research. The as-yet unknown evolutionary road map undertaken by the ancestor of SARS-CoV-2 to cross over to its now human host is to be investigated for understanding its origin.

4. Methods

4.1. Genomes

Historically, genomes held in public databases have been fragmentary, resulting in multiple collections with overlapping examples with alternative naming schemes and annotations. Fortunately, a large collection of virus genomes of the Coronaviridae family (Coronavirus) deposited in databases such as the Virus Pathogen Resource (ViPR) [62] have been provided with both genomic sequence and metadata which has been examined for redundancy and comparative annotation. Coronavirus genomes isolated from humans, bats and pangolins used in this study were collected from multiple repositories and grouped by their host and source. The databases and groups are listed in Table 2.

Table 2. Coronavirus genomes were collected from the various database resources listed by host and source categories. Using taxonomic data made available by the Virus Pathogen Database and Analysis Resource (ViPR) [62], 70 bat-CoVs were identified as *Betacoronavirus* and 84 were *Alphacoronavirus*. Five pangolin-CoVs were identified as *Betacoronavirus*. The remaining bat-CoV and pangolin-CoV genomes did not have a family identification. These were downloaded in May 2020 and consisted of the contemporary available and open datasets at the time. All genomes and their respective IDs are currently available through NCBI (Oct 2020). In cases where two groups contained the same genome (Possibly with a different name), only one representative was taken.

Host–Source	No. Genomes	Database
SARS-CoV-2 Wuhan isolates	20	https://doi.org/10.1101/2020.10.22.328864
SARS-CoV-2 Wuhan isolates	26	GISAID-Charite [14]
SARS-CoV-2 German isolates	117	GISAID-Charite [14]
SARS-CoV-2 Ensembl Wuhan Reference	1	Ensembl [63]
Bat	139	https://doi.org/10.1101/2020.10.22.328864
Bat	76	ViPR [62]
Pangolin	5	https://doi.org/10.1101/2020.10.22.328864
Pangolin	2	NCBI [64]

4.2. Genome Annotation

RNA viruses such as SARS and other coronaviruses have been characterised as having the ability to utilise ribosomal programmed frame-shifting for a number of important genes [65]. Identification of such genes is complex and often requires high quality RNA expression evidence. Due to this and the complexity of genome annotation, especially in novel viral genomes such as SARS-CoV-2, two approaches were taken to identify the set of genes for each of the genomes in this study. In this regard, for defining genes, we first employed PROKKA (Rapid Prokaryotic Genome Annotation) to curate the genes for each of the coronavirus genomes. PROKKA utilises Prodigal [66] to initially find ORFs, which ensures that the DNA sequences of the genes found are in-frame and contain the correct amino acid coding potential. Prodigal is an unsupervised ab initio prediction method and therefore does not rely on previous knowledge to predict ORFs, which, unlike sequence homology based tools such as BLAST, does not require previously annotated sequence data to identify potential genes within novel genomes. However, to overcome the limitations and intricacies of contemporary *ab initio* genome annotation techniques, BLAST was used to identify additional genes with strong homology to those present in the SARS-CoV-2 reference genome released by Ensembl v100 (SARS-CoV-2 ref) *ASM985889v3* [63] (https://covid-19.ensembl.org). The additional BLAST annotation was performed with a BLAST percentage identity threshold of $\geq$80% are labelled separately where annotation methodologies may have an impact. This combined approach was used to avoid solely relying on either method, especially BLAST's agnostic approach to coding frame detection.

4.3. Phylogenetic Trees

A Phylogenetic tree was produced from the genomes of the SARS-CoV-2 Wuhan isolates, Ensembl Wuhan reference and the bat and pangolin coronaviruses to examine their evolutionary relationships at the genomic level. Clustal Omega 1.2.4 [67] was used to

perform a multiple sequence alignment for each of the genomes with default parameters. The phylogenetic tree was inferred from the multiple sequence alignment with RAxML [68] using default parameters apart from the GTRGAMMA option and bootstrapping set to 20. The plotted using packages in R. Midpoint-root and ladderized were carried out using phytools [69] and ape [70], and ggtree [71] was used for the visualisation. The subgenus information for *Betacoronavirus* were curated and clades labelled based on consensus of the majority (i.e., if >85% of the samples in the clade are labelled and have the same subgenus annotation). For labelling the bat-CoVs host genera and species information, a list of host genera and species was curated. Host species with >10 bat-CoV genomes were labelled, followed by host genera with more >10 bat-CoV genomes. The remaining bats were grouped into a single group "other".

4.4. Gene Relationship Network Graph

Genes identified by PROKKA from each host set were collated and together with the additional sequences from the BLAST alignment to the SARS-CoV-2 ref genome as aforementioned, an all-against-all comparison was made with BLAST. This was done with all gene sequences as both the reference and the query as input. A network graph was generated using Graphia Enterprise [72] by treating each gene as a node and generating edges between nodes with significant BLAST alignments. A significant BLAST alignment was defined to have a bit score $\geq$ 60, a query coverage $\geq$ 80% and a percentage identity $\geq$ 80%. Components with less than 5 nodes were removed from the graph. The same procedure was carried out using amino acid sequences as input (Figure A6). Where the amino acid sequences were not generated by PROKKA, the matched sequences extracted from BLAST were translated into amino acid sequences, provided that the sequences contained the start and stop codons.

4.5. Codon Usage

Codon usage metrics for every gene in the SARS-CoV-2 reference gene catalogue were calculated in all available genome sets. Gene sequence output of the PROKKA and BLAST searches (where correct frame was present) were collated and BLAST searched against the SARS-CoV-2 ref genes; genes that have a BLAST result were included and annotated with the SARS-CoV-2 gene. For each set of genes annotated with an SARS-CoV-2 gene, those substantially shorter than the average (<mean length − 2 standard deviation) were removed from codon usage analysis.For ORF6 and ORF8, the BLAST filter criteria yielded few bat-CoV (11 and 3) or pangolin-CoV (1 and 6) genes. Therefore, in addition to the BLAST selected genes, bat-CoV and pangolin-CoV genes labelled as ORF8 and ORF6 in the network analysis (Figure 2) were incorporated in the codon usage analysis. For pangolin-MP789, the PROKKA output from an additional assembly (MT121216.1) was included in the codon usage analysis. Custom Python scripts (available on Github (https://github.com/coronahack2020/final_paper.git) were used to summarise the frequencies of each of the codons for each gene. Non-standard codons, start and stop codons were discarded, along with the codon TGG as it is the only codon coding for tryptophan. PCA was performed on the RSCU values, and kmean clustering was used on the first 10 PCs to group the genomes into 3 clusters.

RSCU was calculated as the ratio of the observed frequency of codon to the expected frequency under the assumption of equal usage between synonymous codons for the same amino acids [73].

4.6. Variant Analysis

For this analysis, we aim to highlight naturally occurring and population-wide viable variants, defined as being different to the SARS-CoV-2 ref and have an impact on coding potential. Variant calling was carried out for all available genome sets against the reference SARS-CoV-2 genome released by Ensembl v100 *ASM985889v3*. The allelic counts and variant effect prediction was carried out in order to identify variants with high impact

changes (inframe deletion, inframe insertion, frameshift, or stop gain) within or between viruses collected from different host species.

Briefly, multiple genome fasta input files were mapped against the SARS-CoV-2 ref assembly using minimap2 [74] with the following flags (*minimap2–cs-cx asm20 INPUT REF > OUT.paf*). The generated PAF (pairwise alignment format) files were subsequently used for variant calling through the paftools.js module in minimap2 (*sort-k6,6 -k8, 8n OUT.paf | paftools.js call-l 200-L 200-q 30 -f REF.fa*). Haplotype aware variant consequences were generated using VEP (Variant Effect Predictor) [75,76]) and BCFtools/csq [77]. The complete set of scripts for this pipeline can be found in https://github.com/coronahack2020/final_paper.git.

Supplementary Materials: The following are available online at https://www.mdpi.com/1999-4915/13/1/49/s1.

Author Contributions: Study conceptualisation, methodology and formal analysis was carried out by N.J.D., B.B.S. and M.S. Data curation was carried out by N.J.D. Writing was carried out by N.J.D., B.B.S. and M.S. Visualization was carried out by B.B.S. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The sources for datasets used in this study are detailed in Table 2.

Acknowledgments: This study was carried out with support from DataBiology, MindStreamAI, University of Edinburgh, The Roslin Institute Royal (Dick) School of Veterinary Studies, Institute of Genetics and Molecular Medicine and University of Aberystwyth. Authors of this manuscript were members of the team who one the 3rd joint position in https://medium.com/@pauldowling/accelerating-scientific-collaboration-in-real-time-e1f682f54c87. The full team members were Mazdak Salavati, Barbara B. Shih, Nicholas J. Dimonaco and David A. Parry that contributed equally to the hackathon's outcome. The prize of the Hackathon sponsored by Slack, Fluidstack, Episode 1, Scan Computers, DataBiology, NVIDIA and MindStreamAI (£500) was used towards publication fees of this manuscript. NJD was awarded the Rhiannon Powell Science Bursary by the Old Students' Association of Aberystwyth University in support of his contribution to the manuscript. Please refer to this link for the details of the event: https://www.coronahack.co.uk/. Thanks to Samantha Lycett, Roslin Institute for comments on the manuscript. BBS is supported by a BBSRC Core Capability Grant (BB/CCG1780/1) to the Roslin Institute.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACE2	Angiotensin Converting Enzyme
BLAST	Basic Local Alignment Search Tool
CoV	Coronavirus
DB	DataBiology
E	Envelope
M	Membrane
MERS-CoV	Middle East Respiratory Syndrome Coronavirus
N	Nucleocapsid
NCBI	National Center for Biotechnology Information
ORF	Open reading frame
PCA	Principle component analysis
PC	Principle component
PROKKA	Rapid Prokaryotic Genome Annotation
RaTG13	SARSr-Ra-BatCoV-RaTG13

RBD	Receptor binding domain
RSCU	Relative synonymous codon usage
SARS	Severe Acute Respiratory Syndrome
S	Spike glycoprotein
ViPR	Virus Pathogen Resource

Appendix E. Phylogenetic Tree

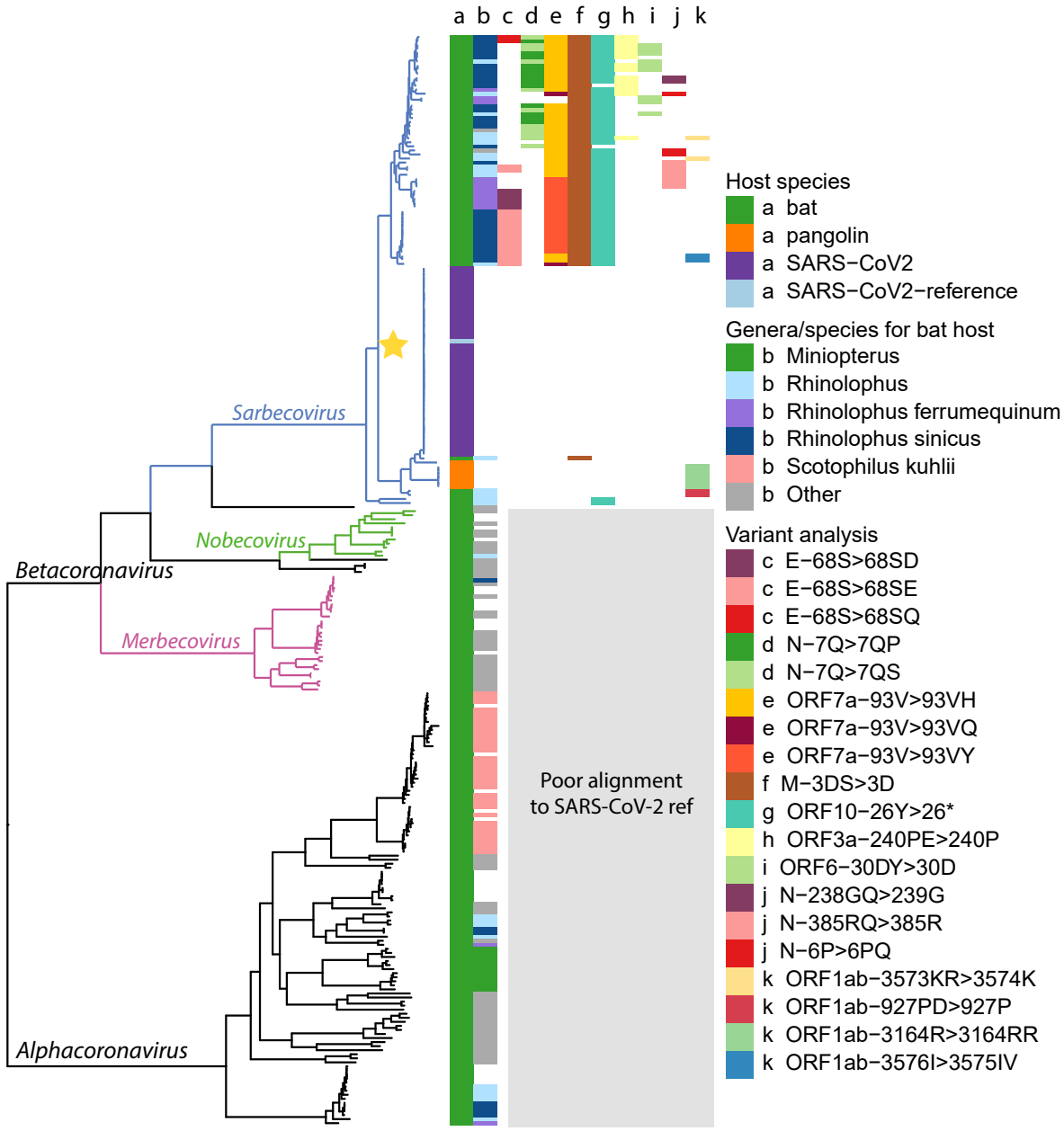


Figure A5. Ladderised phylogenetic tree of bat-CoV, pangolin-CoV and SARS-CoV-2 (Wuhan dataset and reference) genomes. The hosts for each genome are indicated in (a) and host genera or species in (b) for bat-CoV. The majority of the *Sarbecovirus* affect the bat genus *Rhinolophus* (column b, light blue, dark blue and purple), whereas a much smaller proportion of the *Alphacoronavirus* are found in bats of this genus. Some clades overlap with specific bat species, including *Rhinolophus ferrumequinum*, *Rhinolophus sinicus* and *Scotophilus kuhlii*. Several high impact variants (inframe insertion, inframe deletion or stop gain) identified from variant analysis overlap with the clades in the phylogenetic tree. The annotation indicates (c–e) amino acid positions with multiple variants, (f–i) amino acid positions with a single change and found in >10 genomes, (k, l) other variants. The genes and amino acid changes involved in each of the annotated in-frame insertion, in-frame deletion or stop gain (*) are indicated in the figure legend. Star highlights the clade in Figure 1.

Appendix F. Genome Annotation Identified by Source

Table A3. Table containing the total number of genomes and sequences matching genes for each host species group. Gene sets listing number of sequences matching genes identified by either PROKKA or BLAST. SARS-CoV-2 group names shortened as; WI: Wuhan Isolates, GI: German Isolates, EWR: Ensembl Wuhan Reference. Listed is the total number of all PROKKA genes identified and the number of BLAST genes which matched an Ensembl reference gene with 80% percentage identity.

Host–Dataset	No. Genomes	No. Seq Matching Genes	No. by (PROKKA)	No. by (BLAST)
SARS-CoV-2 WI	46	681	591	90
SARS-CoV-2 GI	117	1736	1495	241
SARS-CoV-2 EWR	1	12	N/A	N/A
Bat	215	2427	2365	62
Pangolin	7	97	95	2

Appendix G. Gene–Gene Network Graph Using Amino Acid Sequences

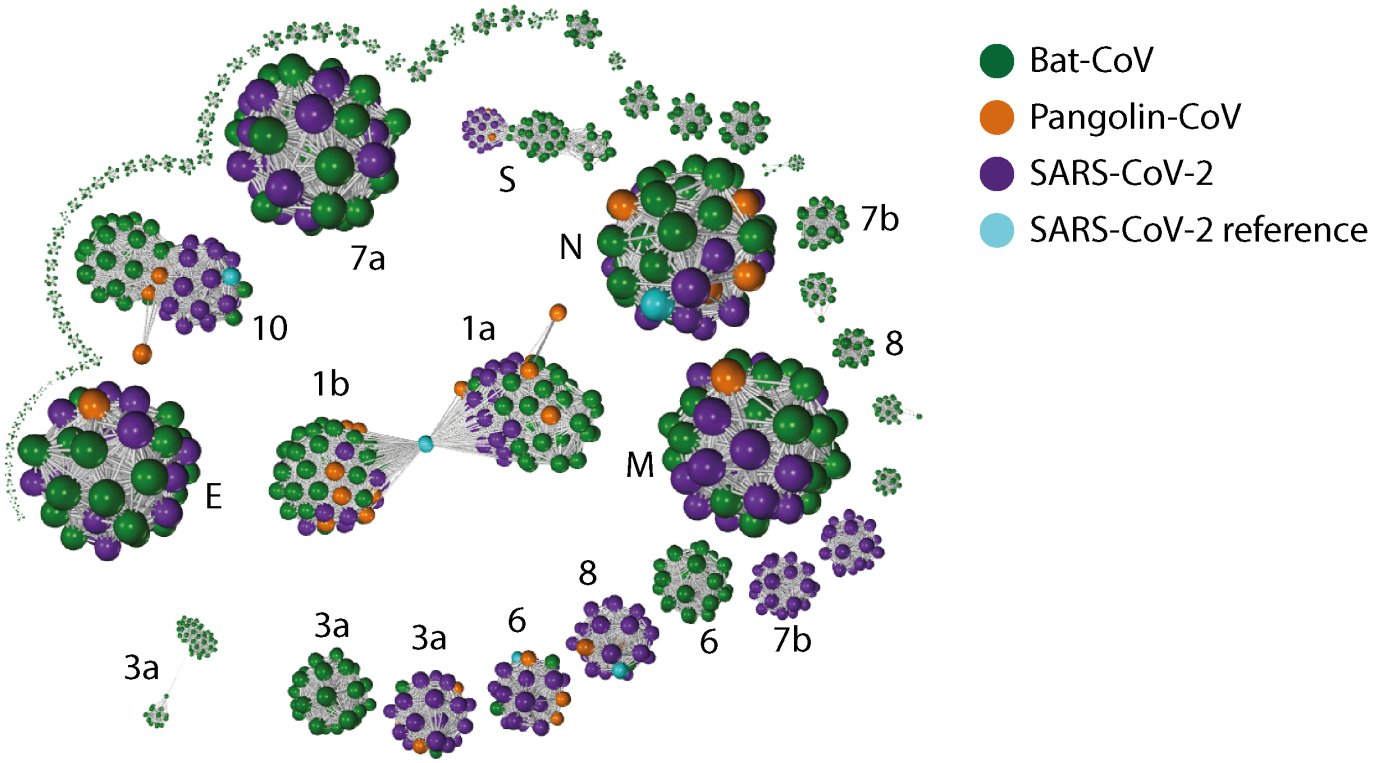


Figure A6. Gene–gene similarity network analysis. Each node represents a amino acid sequence defined by PROKKA or BLAST (ORF10 and E). The nodes were compared against each other using BLAST, and nodes with high similarity (bit score $\geq$ 60 and a query coverage $\geq$80%) were connected with an edge. The network graph is labelled with with SARS-CoV-2 gene names ("ORF" omitted). When the network graph is coloured by host species, genes showing higher degree of variability across species are highlighted. Similar to the network analysis on nucleotide sequences (Figure 2). Genes ORF3a, ORF6, ORF7b, ORF8, ORF10 and S show strong separation between nodes from different species. The degree of separation in ORF1ab are stronger than ORF10 in the nucleic acid network graph; the reverse is true for the amino acid network graph.

Appendix H. PCA Plots Based on the RSCU for Each Gene

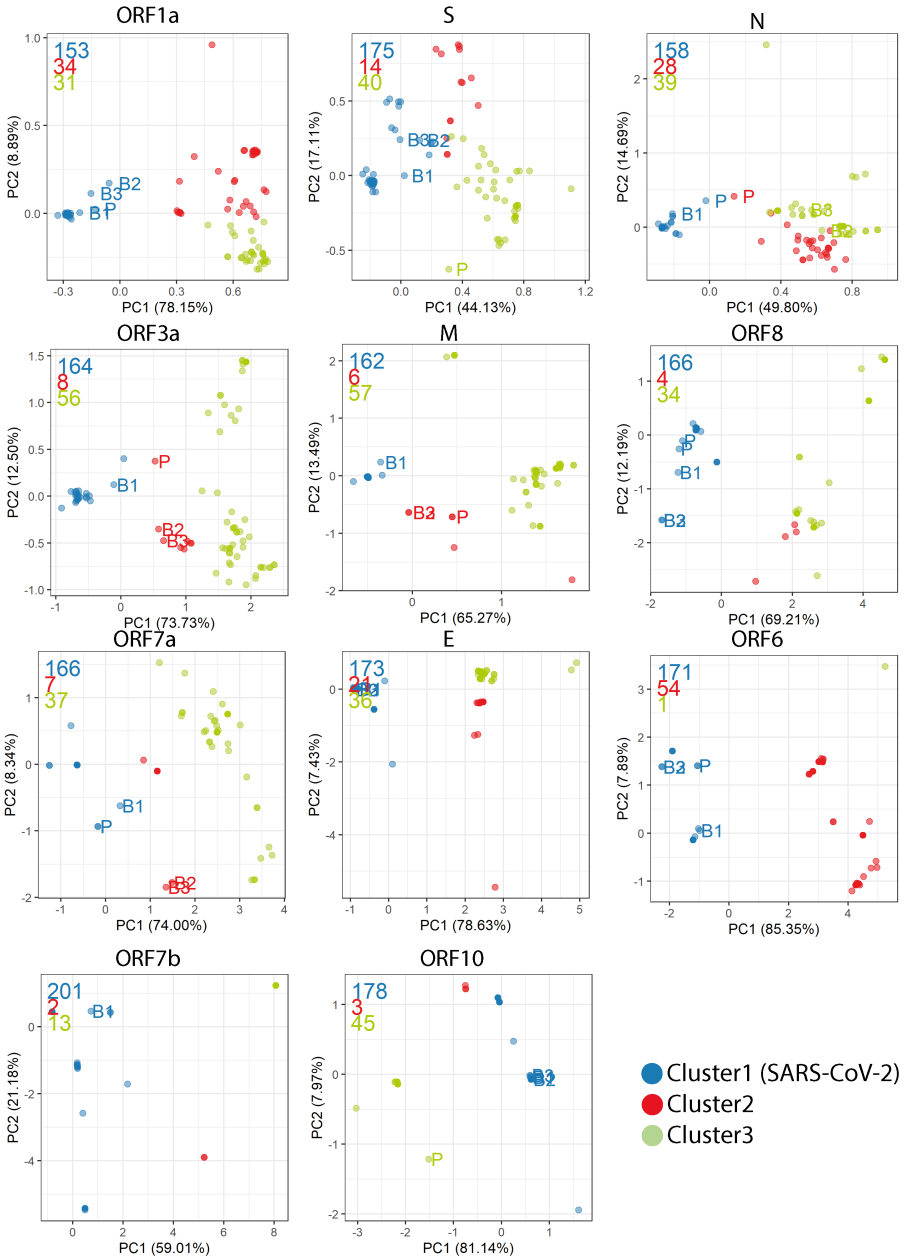


Figure A7. Relative synonymous codon usage (RSCU) was calculated as the ratio of the observed frequency of codon to the expected frequency under the assumption of equal usage between synonymous codons for the same amino acids. For each gene, Principal Component Analysis (PCA) was carried out on the RSCU values, and the first 10 Principal Components (PCs) were used to group the genomes into 3 clusters through kmeans clustering. The cluster with the most number of SARS-CoV-2 genomes is labelled as Cluster 1. Clusters 2 and 3 are assigned according their PC1 and PC2 distance to Cluster 1, with Cluster 2 being closer. Four isolates are labelled: bat-RaTG13 (B1), bat-SL-CoVZC45 (B2), bat-SL-CoVZXC21 (B3), and pangolin-MP789 (P; MT121216.1 and MT084071.1). These 4 isolates are closer to SARS-CoV-2 than other bat-CoV and pangolin-CoV. The number of genomes in Clusters 1, 2 and 3 are indicated in order on the top left corner of each graph.

Appendix I. Synonymous Codon Ratios

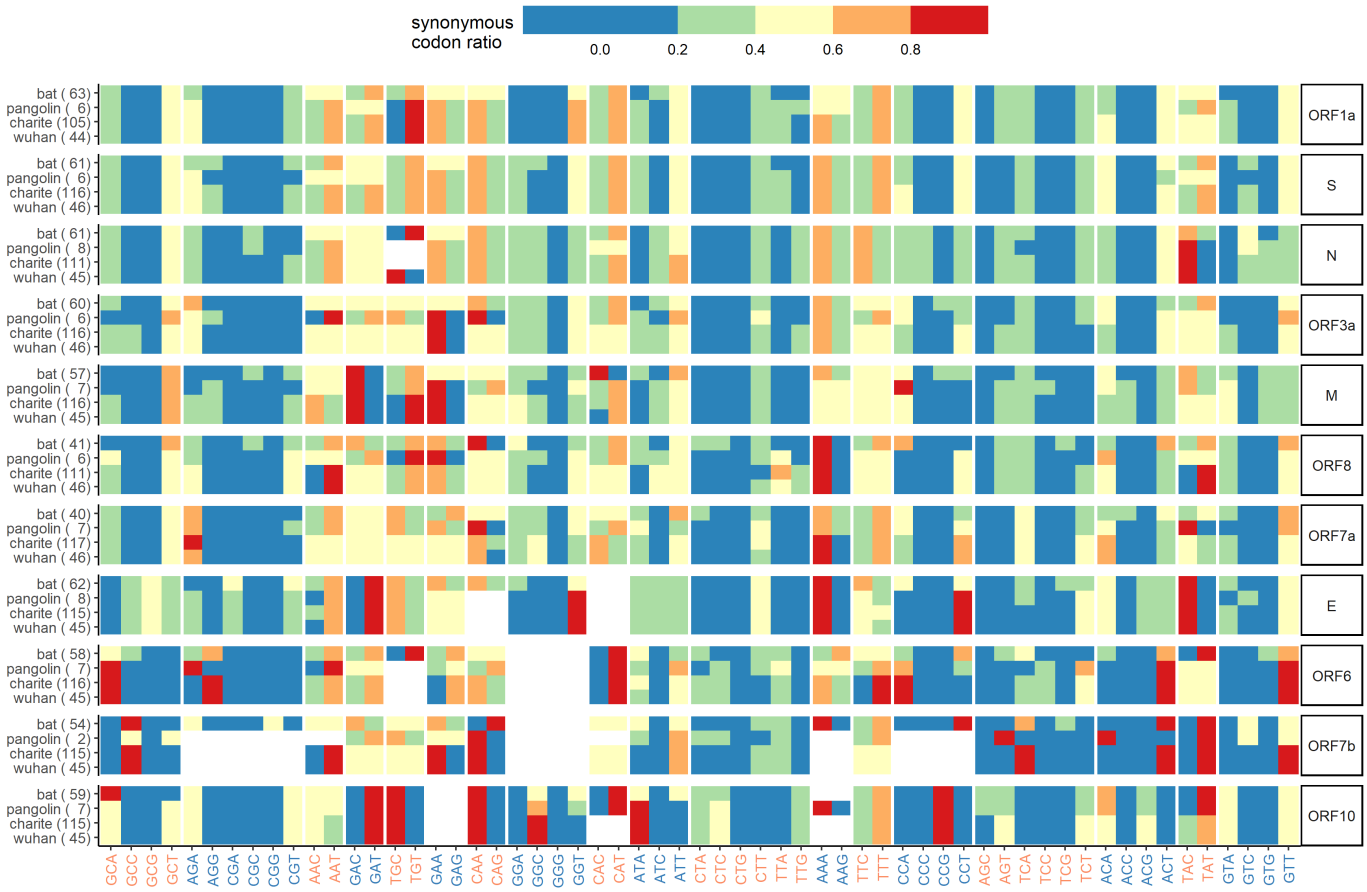


Figure A8. Synonymous codon ratios are the ratio between the number of a given codon divided by the total number of codon coding for the same amino acid. By sorting this ratio in blocks of synonymous codons, this heatmap illustrates the preferential codons for each amino acid for each dataset across all genes. A number of codon usage bias are consistent across most genes and datasets. For instance, GCT is preferentially used for Ala (Alanine) and GTT for Val (Valine). On the whole, there seem to be less of a preferential codon use for bat, especially in longer genes or when multiple genes are accounted for, as per indicated by the higher frequency of more evenly distributed codon within each amino acid (i.e., for the bat dataset, the heatmap colours are of a similar level within each amino acid).

Appendix J. Combined Variant Analysis

Figure A9. The coordinate map of all variants called against the human reference SARS-Cov-2 genome. Each horizontal track shows the variants present in the host-specie group. The colours shows the gene annotation origin of the variant and the shape consequence

References

1. Patz, J.A.; Graczyk, T.K.; Geller, N.; Vittor, A.Y. Effects of environmental change on emerging parasitic diseases. *Int. J. Parasitol.* **2000**, *30*, 1395–1405. [CrossRef]
2. Madhav, N.; Oppenheim, B.; Gallivan, M.; Mulembakani, P.; Rubin, E.; Wolfe, N. *Pandemics: Risks, Impacts, and Mitigation*; The International Bank for Reconstruction and Development/The World Bank: Washington, DC, USA, 2017.
3. Coronaviridae Study Group of the International Committee on Taxonomy of Viruses. The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. *Nat. Microbiol.* **2020**, *5*, 536. [CrossRef]
4. Weiss, S.R. Forty years with coronaviruses. *J. Exp. Med.* **2020**, *217*, e20200537. [CrossRef] [PubMed]
5. Perlman, S.; Netland, J. Coronaviruses post-SARS: update on replication and pathogenesis. *Nat. Rev. Microbiol.* **2009**, *7*, 439–450. [CrossRef]
6. Amer, H.; Alqahtani, A.S.; Alzoman, H.; Aljerian, N.; Memish, Z.A. Unusual presentation of Middle East respiratory syndrome coronavirus leading to a large outbreak in Riyadh during 2017. *Am. J. Infect. Control* **2018**, *46*, 1022–1025. [CrossRef] [PubMed]
7. Hung, L.S. The SARS epidemic in Hong Kong: what lessons have we learned? *J. R. Soc. Med.* **2003**, *96*, 374–378. [CrossRef] [PubMed]
8. Zhu, Z.; Lian, X.; Su, X.; Wu, W.; Marraro, G.A.; Zeng, Y. From SARS and MERS to COVID-19: A brief summary and comparison of severe acute respiratory infections caused by three highly pathogenic human coronaviruses. *Respir. Res.* **2020**, *21*, 1–14. [CrossRef]
9. Boni, M.F.; Lemey, P.; Jiang, X.; Lam, T.T.Y.; Perry, B.; Castoe, T.; Rambaut, A.; Robertson, D.L. Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic. *Nat. Microbiol.* **2020**, *5*, 1408–1417. [CrossRef]

10. Zhang, Y.Z.; Holmes, E.C. A Genomic Perspective on the Origin and Emergence of SARS-CoV-2. *Cell* **2020**, *181*, 223–227. [CrossRef]
11. Jitobaom, K.; Phakaratsakul, S.; Sirihongthong, T.; Chotewutmontri, S.; Suriyaphol, P.; Suptawiwat, O.; Auewarakul, P. Codon usage similarity between viral and some host genes suggests a codon-specific translational regulation. *Heliyon* **2020**, *6*, e03915. [CrossRef]
12. Kumar, N.; Kulkarni, D.D.; Lee, B.; Kaushik, R.; Bhatia, S.; Sood, R.; Pateriya, A.K.; Bhat, S.; Singh, V.P. Evolution of codon usage bias in Henipaviruses is governed by natural selection and is host-specific. *Viruses* **2018**, *10*, 604. [CrossRef] [PubMed]
13. Chen, F.; Wu, P.; Deng, S.; Zhang, H.; Hou, Y.; Hu, Z.; Zhang, J.; Chen, X.; Yang, J.R. Dissimilation of synonymous codon usage bias in virus–host coevolution due to translational selection. *Nat. Ecol. Evol.* **2020**, *4*, 589–600. [CrossRef] [PubMed]
14. Elbe, S.; Buckland-Merrett, G.; Falkename, T.; Thistoo, A. Data, disease and diplomacy: GISAID's innovative contribution to global health. *Glob. Challenges* **2017**, *1*, 33–46. [CrossRef] [PubMed]
15. Zhou, P.; Yang, X.L.; Wang, X.G.; Hu, B.; Zhang, L.; Zhang, W.; Si, H.R.; Zhu, Y.; Li, B.; Huang, C.L.; et al. A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature* **2020**, *579*, 270–273. [CrossRef]
16. Seemann, T. Prokka: Rapid prokaryotic genome annotation. *Bioinformatics* **2014**, *30*, 2068–2069. [CrossRef]
17. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410. [CrossRef]
18. Baranov, P.V.; Henderson, C.M.; Anderson, C.B.; Gesteland, R.F.; Atkins, J.F.; Howard, M.T. Programmed ribosomal frameshifting in decoding the SARS-CoV genome. *Virology* **2005**, *332*, 498–510. [CrossRef]
19. Robertson, M.P.; Igel, H.; Baertsch, R.; Haussler, D.; Ares, M., Jr; Scott, W.G. The structure of a rigorously conserved RNA element within the SARS virus genome. *PLoS Biol.* **2004**, *3*, e5. [CrossRef]
20. Tengs, T.; Jonassen, C.M. Distribution and evolutionary history of the mobile genetic element s2m in coronaviruses. *Diseases* **2016**, *4*, 27. [CrossRef]
21. Tengs, T.; Delwiche, C.F.; Jonassen, C.M. A mobile genetic element in the SARS-CoV-2 genome is shared with multiple insect species. *bioRxiv* **2020**. [CrossRef]
22. Lau, S.K.; Luk, H.K.; Wong, A.C.; Li, K.S.; Zhu, L.; He, Z.; Fung, J.; Chan, T.T.; Fung, K.S.; Woo, P.C. Possible bat origin of severe acute respiratory syndrome coronavirus 2. *Emerg. Infect. Dis.* **2020**, *26*, 1542. [CrossRef] [PubMed]
23. Malaiyan, J.; Arumugam, S.; Mohan, K.; Radhakrishnan, G.G. An update on origin of SARS-CoV-2: Despite closest identity, bat (RaTG13) and Pangolin derived Coronaviruses varied in the critical binding site and O-linked glycan residues. *J. Med Virol.* **2020**. [CrossRef] [PubMed]
24. Li, W.; Shi, Z.; Yu, M.; Ren, W.; Smith, C.; Epstein, J.H.; Wang, H.; Crameri, G.; Hu, Z.; Zhang, H.; et al. Bats are natural reservoirs of SARS-like coronaviruses. *Science* **2005**, *310*, 676–679. [CrossRef] [PubMed]
25. Banerjee, A.; Kulcsar, K.; Misra, V.; Frieman, M.; Mossman, K. Bats and coronaviruses. *Viruses* **2019**, *11*, 41. [CrossRef] [PubMed]
26. Su, S.; Wong, G.; Shi, W.; Liu, J.; Lai, A.C.; Zhou, J.; Liu, W.; Bi, Y.; Gao, G.F. Epidemiology, genetic recombination, and pathogenesis of coronaviruses. *Trends Microbiol.* **2016**, *24*, 490–502. [CrossRef] [PubMed]
27. Yi, H. 2019 novel coronavirus is undergoing active recombination. *Clin. Infect. Dis.* **2020**, *71*, 884–887. [CrossRef]
28. Lopes, L.R.; de Mattos Cardillo, G.; Paiva, P.B. Molecular evolution and phylogenetic analysis of SARS-CoV-2 and hosts ACE2 protein suggest Malayan pangolin as intermediary host. *Braz. J. Microbiol.* **2020**, *51*, 1593–1599. [CrossRef]
29. Fahmi, M.; Kubota, Y.; Ito, M. Nonstructural proteins NS7b and NS8 are likely to be phylogenetically associated with evolution of 2019-nCoV. *Infect. Genet. Evol.* **2020**, *81*, 104272. [CrossRef]
30. Liu, P.; Jiang, J.Z.; Wan, X.F.; Hua, Y.; Li, L.; Zhou, J.; Wang, X.; Hou, F.; Chen, J.; Zou, J.; et al. Are pangolins the intermediate host of the 2019 novel coronavirus (SARS-CoV-2)? *PLoS Pathog.* **2020**, *16*, e1008421. [CrossRef]
31. Xiao, K.; Zhai, J.; Feng, Y.; Zhou, N.; Zhang, X.; Zou, J.J.; Li, N.; Guo, Y.; Li, X.; Shen, X.; et al. Isolation of SARS-CoV-2-related coronavirus from Malayan pangolins. *Nature* **2020**, *583*, 286–289. [CrossRef]
32. Ceraolo, C.; Giorgi, F.M. Genomic variance of the 2019-nCoV coronavirus. *J. Med Virol.* **2020**, *92*, 522–528. [CrossRef] [PubMed]
33. Pereira, F. Evolutionary dynamics of the SARS-CoV-2 ORF8 accessory gene. *Infect. Genet. Evol.* **2020**, *85*, 104525. [CrossRef] [PubMed]
34. Li, J.Y.; Liao, C.H.; Wang, Q.; Tan, Y.J.; Luo, R.; Qiu, Y.; Ge, X.Y. The ORF6, ORF8 and nucleocapsid proteins of SARS-CoV-2 inhibit type I interferon signaling pathway. *Virus Res.* **2020**, *286*, 198074. [CrossRef] [PubMed]
35. Wong, H.H.; Fung, T.S.; Fang, S.; Huang, M.; Le, M.T.; Liu, D.X. Accessory proteins 8b and 8ab of severe acute respiratory syndrome coronavirus suppress the interferon signaling pathway by mediating ubiquitin-dependent rapid degradation of interferon regulatory factor 3. *Virology* **2018**, *515*, 165–175. [CrossRef] [PubMed]
36. Hachim, A.; Kavian, N.; Cohen, C.A.; Chin, A.W.; Chu, D.K.; Mok, C.K.; Tsang, O.T.; Yeung, Y.C.; Perera, R.A.; Poon, L.L.; et al. ORF8 and ORF3b Antibodies Are Accurate Serological Markers of Early and Late SARS-CoV-2 Infection. *Nat. Immunol.* **2020**, *21*, 1293–1301. [CrossRef]
37. Lau, S.K.; Feng, Y.; Chen, H.; Luk, H.K.; Yang, W.H.; Li, K.S.; Zhang, Y.Z.; Huang, Y.; Song, Z.Z.; Chow, W.N.; et al. Severe acute respiratory syndrome (SARS) coronavirus ORF8 protein is acquired from SARS-related coronavirus from greater horseshoe bats through recombination. *J. Virol.* **2015**, *89*, 10532–10547. [CrossRef]
38. Zhang, T.; Wu, Q.; Zhang, Z. Probable pangolin origin of SARS-CoV-2 associated with the COVID-19 outbreak. *Curr. Biol.* **2020**, *30*, 1346–1351.e2. [CrossRef]

39. Wrobel, A.G.; Benton, D.J.; Xu, P.; Roustan, C.; Martin, S.R.; Rosenthal, P.B.; Skehel, J.J.; Gamblin, S.J. SARS-CoV-2 and bat RaTG13 spike glycoprotein structures inform on virus evolution and furin-cleavage effects. *Nat. Struct. Mol. Biol.* **2020**, *27*, 763–767. [CrossRef]
40. Wrapp, D.; Wang, N.; Corbett, K.S.; Goldsmith, J.A.; Hsieh, C.L.; Abiona, O.; Graham, B.S.; McLellan, J.S. Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* **2020**, *367*, 1260–1263. [CrossRef]
41. Baranowski, E.; Ruiz-Jarabo, C.M.; Domingo, E. Evolution of cell recognition by viruses. *Science* **2001**, *292*, 1102–1105. [CrossRef]
42. Baranowski, E.; Ruiz-Jarabo, C.M.; Pariente, N.; Verdaguer, N.; Domingo, E. Evolution of cell recognition by viruses: A source of biological novelty with medical implications. *Adv. Virus Res.* **2003**, *62*, 19. [PubMed]
43. Wu, K.; Chen, L.; Peng, G.; Zhou, W.; Pennell, C.A.; Mansky, L.M.; Geraghty, R.J.; Li, F. A virus-binding hot spot on human angiotensin-converting enzyme 2 is critical for binding of two different coronaviruses. *J. Virol.* **2011**, *85*, 5331–5337. [CrossRef] [PubMed]
44. Hoffmann, M.; Kleine-Weber, H.; Schroeder, S.; Krüger, N.; Herrler, T.; Erichsen, S.; Schiergens, T.S.; Herrler, G.; Wu, N.H.; Nitsche, A.; et al. SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *Cell* **2020**, *181*, 271–280.e8. [CrossRef] [PubMed]
45. Rice, A.M.; Morales, A.C.; Ho, A.T.; Mordstein, C.; Mühlhausen, S.; Watson, S.; Cano, L.; Young, B.; Kudla, G.; Hurst, L.D. Evidence for strong mutation bias towards, and selection against, U content in SARS-CoV-2: Implications for vaccine design. *Mol. Biol. Evol.* **2020**, msaa188.
46. Gu, H.; Chu, D.K.; Peiris, M.; Poon, L.L. Multivariate analyses of codon usage of SARS-CoV-2 and other betacoronaviruses. *Virus Evol.* **2020**, *6*, veaa032. [CrossRef]
47. Nambou, K.; Anakpa, M. Deciphering the co-adaptation of codon usage between respiratory coronaviruses and their human host uncovers candidate therapeutics for COVID-19. *Infect. Genet. Evol.* **2020**, *85*, 104471. [CrossRef]
48. Alonso, A.M.; Diambra, L. SARS-CoV-2 Codon Usage Bias Downregulates Host Expressed Genes With Similar Codon Usage. *Front. Cell Dev. Biol.* **2020**, *8*, 831. [CrossRef]
49. Digard, P.; Lee, H.M.; Sharp, C.; Grey, F.; Gaunt, E. Intra-genome variability in the dinucleotide composition of SARS-CoV-2. *Virus Evol.* **2020**, *6*, veaa057. [CrossRef]
50. Sanjuán, R.; Domingo-Calap, P. Mechanisms of viral mutation. *Cell. Mol. Life Sci.* **2016**, *73*, 4433–4448. [CrossRef]
51. Smith, E.C.; Sexton, N.R.; Denison, M.R. Thinking outside the triangle: replication fidelity of the largest RNA viruses. *Annu. Rev. Virol.* **2014**, *1*, 111–132. [CrossRef]
52. Hassan, S.S.; Attrish, D.; Ghosh, S.; Choudhury, P.P.; Uversky, V.N.; Uhal, B.D.; Lundstrom, K.; Rezaei, N.; Aljabali, A.A.; Seyran, M.; et al. Notable sequence homology of the ORF10 protein introspects the architecture of SARS-COV-2. *bioRxiv* **2020**. [CrossRef]
53. Kim, D.; Lee, J.Y.; Yang, J.S.; Kim, J.W.; Kim, V.N.; Chang, H. The architecture of SARS-CoV-2 transcriptome. *Cell* **2020**, *181*, 914–921.e10. [CrossRef] [PubMed]
54. Liu, T.; Jia, P.; Fang, B.; Zhao, Z. Differential expression of viral transcripts from single-cell RNA sequencing of moderate and severe COVID-19 patients and its implications for case severity. *Front. Microbiol.* **2020**, *11*, 2568. [CrossRef]
55. Schoeman, D.; Fielding, B.C. Coronavirus envelope protein: current knowledge. *Virol. J.* **2019**, *16*, 1–22. [CrossRef]
56. DeDiego, M.L.; Álvarez, E.; Almazán, F.; Rejas, M.T.; Lamirande, E.; Roberts, A.; Shieh, W.J.; Zaki, S.R.; Subbarao, K.; Enjuanes, L. A severe acute respiratory syndrome coronavirus that lacks the E gene is attenuated in vitro and in vivo. *J. Virol.* **2007**, *81*, 1701–1713. [CrossRef] [PubMed]
57. Teoh, K.T.; Siu, Y.L.; Chan, W.L.; Schlüter, M.A.; Liu, C.J.; Peiris, J.M.; Bruzzone, R.; Margolis, B.; Nal, B. The SARS coronavirus E protein interacts with PALS1 and alters tight junction formation and epithelial morphogenesis. *Mol. Biol. Cell* **2010**, *21*, 3838–3852. [CrossRef]
58. Hassan, S.S.; Choudhury, P.P.; Roy, B. SARS-CoV2 envelope protein: non-synonymous mutations and its consequences. *Genomics* **2020**. [CrossRef]
59. Taylor, J.K.; Coleman, C.M.; Postel, S.; Sisk, J.M.; Bernbaum, J.G.; Venkataraman, T.; Sundberg, E.J.; Frieman, M.B. Severe acute respiratory syndrome coronavirus ORF7a inhibits bone marrow stromal antigen 2 virion tethering through a novel mechanism of glycosylation interference. *J. Virol.* **2015**, *89*, 11820–11833. [CrossRef]
60. Cong, Y.; Ulasli, M.; Schepers, H.; Mauthe, M.; V'kovski, P.; Kriegenburg, F.; Thiel, V.; de Haan, C.A.; Reggiori, F. Nucleocapsid protein recruitment to replication-transcription complexes plays a crucial role in coronaviral life cycle. *J. Virol.* **2020**, *94*, e01925-19. [CrossRef]
61. Weber, S.; Ramirez, C.; Doerfler, W. Signal hotspot mutations in SARS-CoV-2 genomes evolve as the virus spreads and actively replicates in different parts of the world. *Virus Res.* **2020**, *289*, 198170. [CrossRef]
62. Pickett, B.E.; Sadat, E.L.; Zhang, Y.; Noronha, J.M.; Squires, R.B.; Hunt, V.; Liu, M.; Kumar, S.; Zaremba, S.; Gu, Z.; et al. ViPR: An open bioinformatics database and analysis resource for virology research. *Nucleic Acids Res.* **2012**, *40*, D593–D598. [CrossRef] [PubMed]
63. Yates, A.D.; Achuthan, P.; Akanni, W.; Allen, J.; Allen, J.; Alvarez-Jarreta, J.; Amode, M.R.; Armean, I.M.; Azov, A.G.; Bennett, R.; et al. Ensembl 2020. *Nucleic Acids Res.* **2020**, *48*, D682–D688. [CrossRef] [PubMed]
64. Coordinators, N.R. Database resources of the national center for biotechnology information. *Nucleic Acids Res.* **2018**, *46*, D8. [CrossRef] [PubMed]

65. Dinman, J.D. Programmed–1 Ribosomal Frameshifting in SARS Coronavirus. In *Molecular Biology of the SARS-Coronavirus*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 63–72.
66. Hyatt, D.; Chen, G.L.; LoCascio, P.F.; Land, M.L.; Larimer, F.W.; Hauser, L.J. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinform.* **2010**, *11*, 119. [CrossRef]
67. Sievers, F.; Higgins, D.G. Clustal Omega for making accurate alignments of many protein sequences. *Protein Sci.* **2018**, *27*, 135–145. [CrossRef]
68. Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **2014**, *30*, 1312–1313. [CrossRef]
69. Revell, L.J. phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.* **2012**, *3*, 217–223. [CrossRef]
70. Paradis, E.; Schliep, K. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* **2019**, *35*, 526–528. [CrossRef]
71. Yu, G. Using ggtree to Visualize Data on Tree-Like Structures. *Curr. Protoc. Bioinform.* **2020**, *69*, e96. [CrossRef]
72. Freeman, T.; Horsewell, S.; Patir, A.; Harling-Lee, J.; Regan, T.; Shih, B.B.; Prendergast, J.; Hume, D.A.; Angus, T. Graphia: A platform for the graph-based visualisation and analysis of complex data. *bioRxiv* **2020**. [CrossRef]
73. Sharp, P.M.; Tuohy, T.M.; Mosurski, K.R. Codon usage in yeast: cluster analysis clearly differentiates highly and lowly expressed genes. *Nucleic Acids Res.* **1986**, *14*, 5125–5143. [CrossRef]
74. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **2018**, *34*, 3094–3100. [CrossRef]
75. McLaren, W.; Gil, L.; Hunt, S.E.; Riat, H.S.; Ritchie, G.R.S.; Thormann, A.; Flicek, P.; Cunningham, F. The Ensembl Variant Effect Predictor. *Genome Biol.* **2016**, *17*, 122. [CrossRef]
76. Den Dunnen, J.T.; Dalgleish, R.; Maglott, D.R.; Hart, R.K.; Greenblatt, M.S.; McGowan-Jordan, J.; Roux, A.F.; Smith, T.; Antonarakis, S.E.; Taschner, P.E. HGVS Recommendations for the Description of Sequence Variants: 2016 Update. *Hum. Mutat.* **2016**, *37*, 564–569. [CrossRef]
77. Danecek, P.; McCarthy, S.A. BCFtools/csq: haplotype-aware variant consequences. *Bioinformatics* **2017**, *33*, 2037–2039. [CrossRef]

Bibliography

Adams, Jerry M and Mario R Capecchi (1966). "N-formylmethionyl-sRNA as the initiator of protein synthesis." In: *Proceedings of the National Academy of Sciences of the United States of America* 55.1, p. 147.

Al-Turaiki, Israa M et al. (2011). "Computational approaches for gene prediction: A comparative survey". In: *International Conference on Informatics Engineering and Information Science*. Springer, pp. 14–25.

Alberts, Bruce et al. (2002). "The shape and structure of proteins". In: *Molecular Biology of the Cell. 4th edition*. Garland Science.

Alff-Steinberger, C and R Epstein (1994). "Codon preference in the terminal region of E. coli genes and evolution of stop codon usage." In: *Journal of Theoretical Biology* 168.4, pp. 461–463.

Alneberg, Johannes et al. (2018). "Genomes from uncultivated prokaryotes: a comparison of metagenome-assembled and single-amplified genomes". In: *Microbiome* 6.1, pp. 1–14.

Alonso, Andres Mariano and Luis Diambra (2020). "SARS-CoV-2 Codon Usage Bias Downregulates Host Expressed Genes With Similar Codon Usage". In: *Frontiers in Cell and Developmental Biology* 8, p. 831. ISSN: 2296-634X. DOI: 10.3389/fcell.2020.00831. URL: https://www.frontiersin.org/article/10.3389/fcell.2020.00831.

Altschul, Stephen F et al. (1990). "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3, pp. 403–410.

Alves, Luís Q et al. (2020). "Pseudo Checker: an integrated online platform for gene inactivation inference". In: *Nucleic Acids Research* 48.W1, W321–W331.

Amann, Rudolf I, Wolfgang Ludwig, and Karl-Heinz Schleifer (1995). "Phylogenetic identification and in situ detection of individual microbial cells without cultivation." In: *Microbiological Reviews* 59.1, pp. 143–169.

Amer, Hala et al. (2018). "Unusual presentation of Middle East respiratory syndrome coronavirus leading to a large outbreak in Riyadh during 2017". In: *American Journal of Infection Control* 46.9, pp. 1022–1025.

Andrews, Shea J and Joseph A Rothnagel (2014). "Emerging evidence for functional peptides encoded by short open reading frames". In: *Nature Reviews Genetics* 15.3, pp. 193–204.

Angiuoli, Samuel V et al. (2008). "Toward an online repository of standard operating procedures (SOPs) for (meta) genomic annotation". In: *OMICS A Journal of Integrative Biology* 12.2, pp. 137–141.

Anton, Brian P et al. (2015). "Complete genome sequence of ER2796, a DNA methyltransferase-deficient strain of Escherichia coli K-12". In: *PloS One* 10.5, e0127446.

Anyansi, Christine et al. (2020). "Computational methods for strain-level microbial detection in colony and metagenome sequencing data". In: *Frontiers in Microbiology* 11, p. 1925.

Arroyo Mühr, Laila Sara et al. (2020). "De novo sequence assembly requires bioinformatic checking of chimeric sequences". In: *PloS One* 15.8, e0237455.

Ashburner, Michael et al. (2000). "Gene ontology: tool for the unification of Biology". In: *Nature Genetics* 25.1, pp. 25–29.

Auslander, Noam, Ayal B Gussow, and Eugene V Koonin (2021). "Incorporating Machine Learning into Established Bioinformatics Frameworks". In: *International Journal of Molecular Sciences* 22.6, p. 2903.

Avni, Eliran et al. (2018). "A phylogenomic study quantifies competing mechanisms for pseudogenization in prokaryotes—The Mycobacterium leprae case". In: *PloS One* 13.11, e0204322.

Ayling, Martin, Matthew D Clark, and Richard M Leggett (2020). "New approaches for metagenome assembly with short reads". In: *Briefings in Bioinformatics* 21.2, pp. 584–594.

Babakhani, Sajad and Mana Oloomi (2018). "Transposons: the agents of antibiotic resistance in bacteria". In: *Journal of Basic Microbiology* 58.11, pp. 905–917.

Badger, Jonathan H and Gary J Olsen (1999). "CRITICA: coding region identification tool invoking comparative analysis." In: *Molecular Biology and Evolution* 16.4, pp. 512–524.

Baek, Jonghwan et al. (2017). "Identification of unannotated small genes in Salmonella". In: *G3: Genes, Genomes, Genetics* 7.3, pp. 983–989.

Banerjee, Arinjay et al. (2019). "Bats and coronaviruses". In: *Viruses* 11.1, p. 41.

Baptista, Rodrigo P and Jessica C Kissinger (2019). "Is reliance on an inaccurate genome sequence sabotaging your experiments?" In: *PLoS Pathogens* 15.9, e1007901.

Baranov, Pavel V, John F Atkins, and Martina M Yordanova (2015). "Augmented genetic decoding: global, local and temporal alterations of decoding processes and codon meaning". In: *Nature Reviews Genetics* 16.9, pp. 517–529.

Baranov, Pavel V et al. (2005). "Programmed ribosomal frameshifting in decoding the SARS-CoV genome". In: *Virology* 332.2, pp. 498–510.

Baranowski, Eric, Carmen M Ruiz-Jarabo, and Esteban Domingo (2001). "Evolution of cell recognition by viruses". In: *Science* 292.5519, pp. 1102–1105.

Baranowski, Eric et al. (2003). "Evolution of cell recognition by viruses: a source of biological novelty with medical implications". In: *Advances in Virus Research* 62, p. 19.

Barrell, Barclay G et al. (1978). *Overlapping Genes in Bacteriophages ϕ X174 and G4*.

Bartholomäus, Alexander et al. (2021). "smORFer: a modular algorithm to detect small ORFs in prokaryotes". In: *Nucleic Acids Research* 49.15, e89–e89.

Bateman, Alex et al. (2020). "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research*.

Batista, Gustavo EAPA, Ronaldo C Prati, and Maria Carolina Monard (2004). "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD Explorations Newsletter* 6.1, pp. 20–29.

Belfort, Marlene et al. (1995). "Prokaryotic introns and inteins: a panoply of form and function." In: *Journal of bacteriology* 177.14, p. 3897.

Belinky, Frida, Igor B Rogozin, and Eugene V Koonin (2017). "Selection on start codons in prokaryotes and potential compensatory nucleotide substitutions". In: *Scientific Reports* 7.1, pp. 1–10.

Belinky, Frida et al. (2018). "Purifying and positive selection in the evolution of stop codons". In: *Scientific Reports* 8.1, pp. 1–11.

Belinky, Frida et al. (2021). "Analysis of Stop Codons within Prokaryotic Protein-Coding Genes Suggests Frequent Readthrough Events". In: *International Journal of Molecular Sciences* 22.4, p. 1876.

Benson, Dennis A et al. (2012). "GenBank". In: *Nucleic Acids Research* 41.D1, pp. D36–D42.

Bentele, Kajetan et al. (2013). "Efficient translation initiation dictates codon usage at gene start". In: *Molecular Systems Biology* 9.1, p. 675.

Besemer, John and Mark Borodovsky (1999). "Heuristic approach to deriving models for gene finding". In: *Nucleic Acids Research* 27.19, pp. 3911–3920.

– (2005). "GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses". In: *Nucleic Acids Research* 33.suppl_2, W451–W454.

Besemer, John, Alexandre Lomsadze, and Mark Borodovsky (2001). "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions". In: *Nucleic Acids Research* 29.12, pp. 2607–2618.

Birchler, James A and Reiner A Veitia (2012). "Gene balance hypothesis: connecting issues of dosage sensitivity across biological disciplines". In: *Proceedings of the National Academy of Sciences* 109.37, pp. 14746–14753.

Bischler, Thorsten et al. (2015). "Differential RNA-seq (dRNA-seq) for annotation of transcriptional start sites and small RNAs in Helicobacter pylori". In: *Methods* 86, pp. 89–101.

Blattner, Frederick R et al. (1997). "The complete genome sequence of Escherichia coli K-12". In: *Science* 277.5331, pp. 1453–1462.

Bohlin, Jon et al. (2017). "The nucleotide composition of microbial genomes indicates differential patterns of selection on core and accessory genomes". In: *BMC Genomics* 18.1, pp. 1–11.

Bolger, Anthony M, Marc Lohse, and Bjoern Usadel (2014). "Trimmomatic: a flexible trimmer for Illumina sequence data". In: *Bioinformatics* 30.15, pp. 2114–2120.

Boni, Maciej F. et al. (July 2020). "Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic". In: *Nature Microbiology*.

ISSN: 2058-5276. DOI: 10.1038/s41564-020-0771-4. URL: https://doi.org/10.1038/s41564-020-0771-4.

Bork, Peer and Amos Bairoch (1996). "Go hunting in sequence databases but watch out for the traps." In: *Trends in Genetics: TIG* 12.10, pp. 425–427.

Borodovsky, Mark and James McIninch (1993). "GENMARK: Parallel gene recognition for both DNA strands". In: *Computers & Chemistry* 17.2, pp. 123–133.

Brady, Arthur and Steven L Salzberg (2009). "Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models". In: *Nature Methods* 6.9, pp. 673–676.

Braun, Burkhard R et al. (2005). "A human-curated annotation of the Candida albicans genome". In: *PLoS Genetics* 1.1, e1.

Brenner, Steven E (1999). "Errors in genome annotation". In: *Trends in Genetics* 15.4, pp. 132–133.

Brent, Michael R (2005). "Genome annotation past, present, and future: how to define an ORF at each locus". In: *Genome Research* 15.12, pp. 1777–1786.

Browning, Douglas F and Stephen JW Busby (2004). "The regulation of bacterial transcription initiation". In: *Nature Reviews Microbiology* 2.1, pp. 57–65.

Buchfink, Benjamin, Klaus Reuter, and Hajk-Georg Drost (2021). "Sensitive protein alignments at tree-of-life scale using DIAMOND". In: *Nature Methods* 18.4, pp. 366–368.

Buchfink, Benjamin, Chao Xie, and Daniel H Huson (2015). "Fast and sensitive protein alignment using DIAMOND". In: *Nature Methods* 12.1, pp. 59–60.

Bulmer, Michael (1987). "Coevolution of codon usage and transfer RNA abundance". In: *Nature* 325.6106, pp. 728–730.

Burge, Christopher B. and Samuel Karlinb (1998). "Finding the genes in genomic DNA". In: *Current Opinion in Structural Biology* 8.3, pp. 346–354. DOI: http://dx.doi.org/10.1016/S0959-440X(98)80069-9. URL: http://www.sciencedirect.com/science/article/pii/S0959440X98800699.

Cantalapiedra, Carlos P et al. (2021). "eggNOG-mapper v2: Functional Annotation, Orthology Assignments, and Domain Prediction at the Metagenomic Scale". In: *bioRxiv*.

Carlile, Michael (1982). "Prokaryotes and eukaryotes: strategies and successes". In: *Trends in Biochemical Sciences* 7.4, pp. 128–130.

Carlos Guimaraes, Luis et al. (2015). "Inside the pan-genome-methods and software overview". In: *Current Genomics* 16.4, pp. 245–252.

Carneiro, Adriana R et al. (2012). "Quality of prokaryote genome assembly: indispensable issues of factors affecting prokaryote genome assembly quality". In: *Gene* 505.2, pp. 365–367.

Carr, Rogan and Elhanan Borenstein (2014). "Comparative analysis of functional metagenomic annotation and the mappability of short reads". In: *PloS One* 9.8, e105776.

Ceraolo, Carmine and Federico M Giorgi (2020). "Genomic variance of the 2019-nCoV coronavirus". In: *Journal of Medical Virology* 92.5, pp. 522–528.

Cheetham, Seth W, Geoffrey J Faulkner, and Marcel E Dinger (2020). "Overcoming challenges and dogmas to understand the functions of pseudogenes". In: *Nature Reviews Genetics* 21.3, pp. 191–201.

Chen, Feng et al. (2020a). "Dissimilation of synonymous codon usage bias in virus–host coevolution due to translational selection". In: *Nature Ecology & Evolution*, pp. 1–12.

Chen, Lin-Xing et al. (2020b). "Accurate and complete genomes from metagenomes". In: *Genome Research* 30.3, pp. 315–333.

Chen, Swaine L et al. (2006). "Identification of genes subject to positive selection in uropathogenic strains of Escherichia coli: a comparative genomics approach". In: *Proceedings of the National Academy of Sciences* 103.15, pp. 5977–5982.

Cheng, Alice G, Dominique Missiakas, and Olaf Schneewind (2014). "The giant protein Ebh is a determinant of *Staphylococcus aureus* cell size and complement resistance". In: *Journal of Bacteriology* 196.5, pp. 971–981.

Chollet, Francois et al. (2015). *Keras*. URL: https://github.com/fchollet/keras.

Chu, Xiao et al. (2021). "Gene loss through pseudogenization contributes to the ecological diversification of a generalist Roseobacter lineage". In: *The ISME Journal* 15.2, pp. 489–502.

Chu, Yongjun and David R Corey (2012). "RNA sequencing: platform selection, experimental design, and data interpretation". In: *Nucleic Acid Therapeutics* 22.4, pp. 271–274.

Chung, Su Yun and S Subbiah (1996). "A structural explanation for the twilight zone of protein sequence homology". In: *Structure* 4.10, pp. 1123–1127.

Cock, Peter JA et al. (2010). "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". In: *Nucleic Acids Research* 38.6, pp. 1767–1771.

Cohen, Ofir, Uri Gophna, and Tal Pupko (2011). "The complexity hypothesis revisited: connectivity rather than function constitutes a barrier to horizontal gene transfer". In: *Molecular Biology and Evolution* 28.4, pp. 1481–1489.

Colavecchio, Anna et al. (2017). "Bacteriophages contribute to the spread of antibiotic resistance genes among foodborne pathogens of the Enterobacteriaceae family–a review". In: *Frontiers in Microbiology* 8, p. 1108.

Cong, Yingying et al. (2020). "Nucleocapsid protein recruitment to replication-transcription complexes plays a crucial role in coronaviral life cycle". In: *Journal of Virology* 94.4.

Coordinators, NCBI Resource (2018). "Database resources of the national center for biotechnology information". In: *Nucleic Acids Research* 46.Database issue, p. D8.

Dabrowski, Maciej, Zuzanna Bukowy-Bieryllo, and Ewa Zietkiewicz (2015). "Translational readthrough potential of natural termination codons in eucaryotes–The impact of RNA sequence". In: *RNA Biology* 12.9, pp. 950–958.

Dalgarno, L and J Shine (1973). "Conserved terminal sequence in 18S rRNA may represent terminator anticodons". In: *Nature New Biology* 245.148, pp. 261–262.

Dandekar, Thomas et al. (1998). "Conservation of gene order: a fingerprint of proteins that physically interact". In: *Trends in Biochemical Sciences* 23.9, pp. 324–328.

Danecek, Petr and Shane A McCarthy (Feb. 2017). "BCFtools/csq: haplotype-aware variant consequences". In: *Bioinformatics* 33.13, pp. 2037–2039. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx100. URL: https://doi.org/10.1093/bioinformatics/btx100.

Dar, Daniel and Rotem Sorek (2018). "Bacterial noncoding RNAs excised from within protein-coding transcripts". In: *MBio* 9.5, e01730–18.

Darnell, James E (1978). "Implications of RNA-RNA splicing in evolution of eukaryotic cells". In: *Science* 202.4374, pp. 1257–1260.

Davies, Julian, Walter Gilbert, and Luigi Gorini (1964). "Streptomycin, suppression, and the code". In: *Proceedings of the National Academy of Sciences of the United States of America* 51.5, p. 883.

Decano, Arun Gonzales and Tim Downing (2019). "An Escherichia coli ST131 pangenome atlas reveals population structure and evolution across 4,071 isolates". In: *Scientific Reports* 9.1, pp. 1–13.

DeDiego, Marta L et al. (2007). "A severe acute respiratory syndrome coronavirus that lacks the E gene is attenuated in vitro and in vivo". In: *Journal of Virology* 81.4, pp. 1701–1713.

Delcher, Arthur L et al. (1999). "Improved microbial gene identification with GLIMMER". In: *Nucleic Acids Research* 27.23, pp. 4636–4641.

Delcher, Arthur L et al. (2007). "Identifying bacterial genes and endosymbiont DNA with Glimmer". In: *Bioinformatics* 23.6, pp. 673–679.

Denton, James F et al. (2014). "Extensive error in the number of genes inferred from draft genome assemblies". In: *PLoS Computational Biology* 10.12, e1003998.

Devos, Damien and Alfonso Valencia (2001). "Intrinsic errors in genome annotation". In: *TRENDS in Genetics* 17.8, pp. 429–431.

Dietterich, Tom (1995). "Overfitting and undercomputing in machine learning". In: *ACM Computing Surveys (CSUR)* 27.3, pp. 326–327.

Digard, P. et al. (2020). "Intra-genome variability in the dinucleotide composition of SARS-CoV-2". In: *Virus Evolution* 6.2, veaa057.

Dill, Ken A et al. (2008). "The protein folding problem". In: *Annu. Rev. Biophys.* 37, pp. 289–316.

Dimonaco, Nicholas J, Mazdak Salavati, and Barbara B Shih (2021). "Computational analysis of SARS-CoV-2 and SARS-like coronavirus diversity in human, bat and pangolin populations". In: *Viruses* 13.1, p. 49.

Dimonaco, Nicholas J et al. (Dec. 2021). "No one tool to rule them all: Prokaryotic gene prediction tool annotations are highly dependent on the organism of study". In: *Bioinformatics*. btab827. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab827. URL: https://doi.org/10.1093/bioinformatics/btab827.

Dinman, Jonathan D (2010). "Programmed–1 Ribosomal Frameshifting in SARS Coronavirus". In: *Molecular Biology of the SARS-Coronavirus*. Springer, pp. 63–72.

– (2012). "Control of gene expression by translational recoding". In: *Advances in Protein Chemistry and Structural Biology* 86, pp. 129–149.

Dixon, Kevin et al. (2007). "Identification of the functional initiation codons of a phase-variable gene of Haemophilus influenzae, lic2A, with the potential for differential expression". In: *Journal of Bacteriology* 189.2, pp. 511–521.

Dryden, David TF, Andrew R Thomson, and John H White (2008). "How much of protein sequence space has been explored by life on Earth?" In: *Journal of The Royal Society Interface* 5.25, pp. 953–956.

Du, Liutao et al. (2009). "Nonaminoglycoside compounds induce readthrough of nonsense mutations". In: *Journal of Experimental Medicine* 206.10, pp. 2285–2297.

Du, Meng-Ze et al. (2018). "The GC content as a main factor shaping the amino acid usage during bacterial evolution process". In: *Frontiers in Microbiology* 9, p. 2948.

Dueholm, Morten Simonsen, Heidi Nolsøe Danielsen, and Per Halkjær Nielsen (2014). "Complete genome sequence of Pseudomonas sp. UK4, a model organism for studies of functional amyloids in Pseudomonas". In: *Genome Announc.* 2.5, e00898–14.

Dumontier, Michel, Katerina Michalickova, and Christopher WV Hogue (2002). "Species-specific protein sequence and fold optimizations". In: *BMC Bioinformatics* 3.1, pp. 1–15.

Dunne, Michael P and Steven Kelly (2017). "OrthoFiller: utilising data from multiple species to improve the completeness of genome annotations". In: *BMC Genomics* 18.1, p. 390.

Dunnen, Johan T. den et al. (2016). "HGVS Recommendations for the Description of Sequence Variants: 2016 Update". In: *Human Mutation* 37.6, pp. 564–569. DOI: 10.1002/humu.22981. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/humu.22981.

Duval, Mélodie and Pascale Cossart (2017). "Small bacterial and phagic proteins: an updated view on a rapidly moving field". In: *Current Opinion in Microbiology* 39, pp. 81–88.

Dybvig, Kevin and LeRoy L Voelker (1996). "Molecular Biology of *Mycoplasmas*". In: *Annual Reviews in Microbiology* 50.1, pp. 25–57.

Edgell, David R, Marlene Belfort, and David A Shub (2000). "Barriers to intron promiscuity in bacteria". In: *Journal of Bacteriology* 182.19, pp. 5281–5289.

Eilbeck, Karen et al. (2005). "The Sequence Ontology: a tool for the unification of genome annotations". In: *Genome Biology* 6.5, R44.

Eisenhaber, Frank (2006). "Prediction of protein function". In: *Discovering biomolecular mechanisms with computational Biology*. Springer, pp. 39–54.

Elbe, Stefan et al. (2017). "Data, disease and diplomacy: GISAID's innovative contribution to global health". In: *Global Challenges* 1.1, pp. 33–46.

Elena, Santiago F et al. (2005). In:

Eyre-Walker, Adam and Michael Bulmer (1993). "Reduced synonymous substitution rate at the start of enterobacterial genes". In: *Nucleic Acids Research* 21.19, pp. 4599–4603.

Fahmi, Muhamad, Yukihiko Kubota, and Masahiro Ito (2020). "Nonstructural proteins NS7b and NS8 are likely to be phylogenetically associated with evolution of 2019-nCoV". In: *Infection, Genetics and Evolution* 81, p. 104272.

Fickett, James W (1982). "Recognition of protein coding regions in DNA sequences". In: *Nucleic Acids Research* 10.17, pp. 5303–5318.

– (1995). "ORFs and genes: how strong a connection?" In: *Journal of Computational Biology* 2.1, pp. 117–123.

Forbes, Jessica D et al. (2017). "Metagenomics: the next culture-independent game changer". In: *Frontiers in Microbiology* 8, p. 1069.

Forni, Diego et al. (2021). "The substitution spectra of coronavirus genomes". In: *Briefings in Bioinformatics*.

Freeman, Thomas et al. (2020). "Graphia: A platform for the graph-based visualisation and analysis of complex data". In: *bioRxiv*. DOI: 10.1101/2020.09.02.279349.

Fricke, W Florian and David A Rasko (2014). "Bacterial genome sequencing in the clinic: bioinformatic challenges and solutions". In: *Nature Reviews Genetics* 15.1, pp. 49–55.

Friedman, Jerome H (2006). "Recent advances in predictive (machine) learning". In: *Journal of Classification* 23.2, pp. 175–197.

Frishman, Dmitrij (2007). "Protein annotation at genomic scale: the current status". In: *Chemical Reviews* 107.8, pp. 3448–3466.

Fu, Limin et al. (2012). "CD-HIT: accelerated for clustering the next-generation sequencing data". In: *Bioinformatics* 28.23, pp. 3150–3152.

Fukiya, Satoru et al. (2004). "Extensive genomic diversity in pathogenic Escherichia coli and Shigella strains revealed by comparative genomic hybridization microarray". In: *Journal of Bacteriology* 186.12, pp. 3911–3921.

Fukuda, Yoko, Yoichi Nakayama, and Masaru Tomita (2003). "On dynamics of overlapping genes in bacterial genomes". In: *Gene* 323, pp. 181–187.

Furnham, Nicholas, Tjaart AP de Beer, and Janet M Thornton (2012). "Current challenges in genome annotation through structural biology and bioinformatics". In: *Current Opinion in Structural Biology* 22.5, pp. 594–601.

Furuno, Masaaki et al. (2003). "CDS annotation in full-length cDNA sequence". In: *Genome Research* 13.6b, pp. 1478–1487.

Galperin, Michael Y et al. (2019). "Microbial genome analysis: the COG approach". In: *Briefings in Bioinformatics* 20.4, pp. 1063–1070.

Galperin, Michael Y et al. (2021). "COG database update: focus on microbial diversity, model organisms, and widespread pathogens". In: *Nucleic Acids Research* 49.D1, pp. D274–D281.

Ghatak, Sankha et al. (2019). "The y-ome defines the 35% of Escherichia coli genes that lack experimental evidence of function". In: *Nucleic Acids Research* 47.5, pp. 2446–2454.

Gilbert, Jack A et al. (2018). "Current understanding of the human microbiome". In: *Nature medicine* 24.4, pp. 392–400.

Giovannoni, Stephen J, J Cameron Thrash, and Ben Temperton (2014). "Implications of streamlining theory for microbial ecology". In: *The ISME Journal* 8.8, pp. 1553–1565.

Glass, John I et al. (2006). "Essential genes of a minimal bacterium". In: *Proceedings of the National Academy of Sciences* 103.2, pp. 425–430.

Goldstein, Gideon et al. (1975). "Isolation of a polypeptide that has lymphocyte-differentiating properties and is probably represented universally in living cells". In: *Proceedings of the National Academy of Sciences* 72.1, pp. 11–15.

Goli, Baharak and Achuthsankar S Nair (2012). "The elusive short gene–an ensemble method for recognition for prokaryotic genome". In: *Biochemical and Biophysical Research Communications* 422.1, pp. 36–41.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). "Machine learning basics". In: *Deep Learning* 1.7, pp. 98–164.

Goodhead, Ian and Alistair C Darby (2015). "Taking the pseudo out of pseudogenes". In: *Current Opinion in Microbiology* 23, pp. 102–109.

Goodwin, Sara, John D McPherson, and W Richard McCombie (2016). "Coming of age: ten years of next-generation sequencing technologies". In: *Nature Reviews Genetics* 17.6, pp. 333–351.

Gouy, Manolo and Christian Gautier (1982). "Codon usage in bacteria: correlation with gene expressivity". In: *Nucleic Acids Research* 10.22, pp. 7055–7074.

Gribskov, Michael, John Devereux, and Richard R Burgess (1984). "The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression". In:

Griffith, Malachi and Marco A Marra (2007). "Alternative expression analysis: experimental and bioinformatic approaches for the analysis of transcript diversity". In: *Genes, Genomes & Genomics* 2, pp. 201–242.

Gu, Haogao et al. (2020). "Multivariate analyses of codon usage of SARS-CoV-2 and other betacoronaviruses". In: *Virus Evolution* 6.1, veaa032.

Guigo, Roderic (1997). "Computational gene identification: an open problem". In: *Computers & Chemistry* 21.4, pp. 215–222.

Guo, Yan et al. (2014). "Three-stage quality control strategies for DNA re-sequencing data". In: *Briefings in Bioinformatics* 15.6, pp. 879–889.

Haas, Brian J et al. (2011). "Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons". In: *Genome Research* 21.3, pp. 494–504.

Haas, Brian J et al. (2013). "De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis". In: *Nature protocols* 8.8, pp. 1494–1512.

Hachim, Asmaa et al. (2020). *ORF8 and ORF3b antibodies are accurate serological markers of early and late SARS-CoV-2 infection*. Tech. rep. Nature Publishing Group.

Haft, Daniel H et al. (2017). "RefSeq: an update on prokaryotic genome annotation and curation". In: *Nucleic Acids Research* 46.D1, pp. D851–D860.

– (2018). "RefSeq: an update on prokaryotic genome annotation and curation". In: *Nucleic Acids Research* 46.D1, pp. D851–D860.

Haldane, JBS (1933). "The part played by recurrent mutation in evolution". In: *The American Naturalist* 67.708, pp. 5–19.

Hall, Mark et al. (2009). "The WEKA data mining software: an update". In: *ACM SIGKDD Explorations Newsletter* 11.1, pp. 10–18.

Hannenhalli, Sridhar S et al. (1999). "Bacterial start site prediction". In: *Nucleic Acids Research* 27.17, pp. 3577–3582.

Haroon, Mohamed F et al. (2016). "A catalogue of 136 microbial draft genomes from Red Sea metagenomes". In: *Scientific Data* 3.1, pp. 1–6.

Harris, Zachary N et al. (2019). "Massive metagenomic data analysis using abundance-based machine learning". In: *Biology Direct* 14.1, pp. 1–13.

Hassan, Sk Sarif, Pabitra Pal Choudhury, and Bidyut Roy (2020). "SARS-CoV2 envelope protein: non-synonymous mutations and its consequences". In: *Genomics*.

Hassan, Sk Sarif et al. (2020). "Notable sequence homology of the ORF10 protein introspects the architecture of SARS-COV-2". In: *bioRxiv*.

Haynes, Winston A, Aurelie Tomczak, and Purvesh Khatri (2018). "Gene annotation bias impedes biomedical Research". In: *Scientific Reports* 8.1, pp. 1–7.

Hecht, Ariel et al. (2017). "Measurements of translation initiation from all 64 codons in E. coli". In: *Nucleic Acids Research* 45.7, pp. 3615–3626.

Hemm, Matthew R et al. (2008). "Small membrane proteins found by comparative genomics and ribosome binding site models". In: *Molecular Microbiology* 70.6, pp. 1487–1501.

Hendrix, Roger W (2003). "Bacteriophage genomics". In: *Current Opinion in Microbiology* 6.5, pp. 506–511.

HengLi (2018). *SeqTK*. https://github.com/lh3/seqtk.

Higdon, Roger, Brenton Louie, and Eugene Kolker (2010). "Modeling sequence and function similarity between proteins for protein functional annotation". In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 499–502.

Hillmann, Benjamin et al. (2018). "Evaluating the information content of shallow shotgun metagenomics". In: *Msystems* 3.6, e00069–18.

Hitch, Thomas CA and Christopher J Creevey (2018). "Spherical: an iterative workflow for assembling metagenomic datasets". In: *BMC Bioinformatics* 19.1, pp. 1–8.

Ho, Joanne ML et al. (2021). "Improved pyrrolysine biosynthesis through phage assisted non-continuous directed evolution of the complete pathway". In: *Nature Communications* 12.1, pp. 1–10.

Hoff, Katharina J et al. (2009). "Orphelia: predicting genes in metagenomic sequencing reads". In: *Nucleic Acids Research* 37.suppl_2, W101–W105.

Hoffmann, Markus et al. (2020). "SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor". In: *Cell*.

Holmes, Edward C et al. (2021). "The origins of SARS-CoV-2: A critical review". In: *Cell*.

Hood, Leroy and Lee Rowen (2013). "The human genome project: big science transforms biology and medicine". In: *Genome medicine* 5.9, p. 79.

Horesh, Gal et al. (2021). "A comprehensive and high-quality collection of Escherichia coli genomes and their genes". In: *Microbial genomics* 7.2.

Howe, Doug et al. (2008). "The future of biocuration". In: *Nature* 455.7209, pp. 47–50.

Howe, Kevin L et al. (2020). "Ensembl Genomes 2020 – enabling non-vertebrate genomic Research". In: *Nucleic Acids Research* 48.D1, pp. D689–D695.

Huerta-Cepas, Jaime et al. (2019). "eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses". In: *Nucleic Acids Research* 47.D1, pp. D309–D314.

Hung, Lee Shiu (2003). "The SARS epidemic in Hong Kong: what lessons have we learned?" In: *Journal of the Royal Society of Medicine* 96.8, pp. 374–378.

Hunter, Philip (2008a). "Not so simple after all: A renaissance of research into prokaryotic evolution and cell structure". In: *EMBO Reports* 9.3, pp. 224–226.

– (2008b). "The paradox of model organisms: the use of model organisms in research will continue despite their shortcomings". In: *EMBO Reports* 9.8, pp. 717–720.

Hutchison, Clyde A et al. (1999). "Global transposon mutagenesis and a minimal Mycoplasma genome". In: *Science* 286.5447, pp. 2165–2169.

Huvet, Maxime and Michael PH Stumpf (2014). "Overlapping genes: a window on gene evolvability". In: *BMC Genomics* 15.1, p. 721.

Hyatt, Doug et al. (2010). "Prodigal: prokaryotic gene recognition and translation initiation site identification". In: *BMC Bioinformatics* 11.1, p. 119.

Ikemura, Toshimichi (1981). "Correlation between the abundance of Escherichia coli transfer RNAs and the occurrence of the respective codons in its protein genes: a proposal for a synonymous codon choice that is optimal for the E. coli translational system". In: *Journal of Molecular Biology* 151.3, pp. 389–409.

– (1985). "Codon usage and tRNA content in unicellular and multicellular organisms." In: *Molecular Biology and Evolution* 2.1, pp. 13–34.

International, Coronaviridae Study Group of the et al. (2020). "The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2". In: *Nature Microbiology* 5.4, p. 536.

Itaya, Mitsuhiro et al. (2005). "Combining two genomes in one cell: stable cloning of the Synechocystis PCC6803 genome in the Bacillus subtilis 168 genome". In: *Proceedings of the National Academy of Sciences* 102.44, pp. 15971–15976.

Ivanova, Natalia N et al. (2014). "Stop codon reassignments in the wild". In: *Science* 344.6186, pp. 909–913.

Jacq, Claude, JR Miller, and GG Brownlee (1977). "A pseudogene structure in 5S DNA of Xenopus laevis". In: *Cell* 12.1, pp. 109–120.

Jain, Ravi, Maria C Rivera, and James A Lake (1999). "Horizontal gene transfer among genomes: the complexity hypothesis". In: *Proceedings of the National Academy of Sciences* 96.7, pp. 3801–3806.

Japkowicz, Nathalie and Shaju Stephen (2002). "The class imbalance problem: A systematic study". In: *Intelligent Data Analysis* 6.5, pp. 429–449.

Ji, Xiangwen, Chunmei Cui, and Qinghua Cui (2020). "smORFunction: a tool for predicting functions of small open reading frames and microproteins". In: *BMC Bioinformatics* 21.1, pp. 1–13.

Jiao, Jian et al. (2018). "Coordinated regulation of core and accessory genes in the multipartite genome of Sinorhizobium fredii". In: *PLoS Genetics* 14.5, e1007428.

Jitobaom, Kunlakanya et al. (2020). "Codon usage similarity between viral and some host genes suggests a codon-specific translational regulation". In: *Heliyon* 6.5, e03915.

Johnson, Zackary I and Sallie W Chisholm (2004). "Properties of overlapping genes are conserved across microbial genomes". In: *Genome Research* 14.11, pp. 2268–2272.

Jordan, Michael I and Tom M Mitchell (2015). "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245, pp. 255–260.

Jumper, John et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589.

Jungreis, Irwin, Rachel Sealfon, and Manolis Kellis (2021). "SARS-CoV-2 gene content and COVID-19 mutation impact by comparing 44 Sarbecovirus genomes". In: *Nature Communications* 12.1, pp. 1–20.

Kahan, Rashi et al. (2021). "Modulators of protein–protein interactions as antimicrobial agents". In: *RSC Chemical Biology* 2.2, pp. 387–409.

Kalkatawi, Manal, Intikhab Alam, and Vladimir B Bajic (2015). "BEACON: automated tool for B acterial GE nome A nnotation C omparis ON". In: *BMC Genomics* 16.1, pp. 1–8.

Kamke, Janine et al. (2016). "Rumen metagenome and metatranscriptome analyses of low methane yield sheep reveals a Sharpea-enriched microbiome characterised by lactic acid formation and utilisation". In: *Microbiome* 4.1, pp. 1–16.

Kanehisa, Minoru and Susumu Goto (2000). "KEGG: kyoto encyclopedia of genes and genomes". In: *Nucleic Acids Research* 28.1, pp. 27–30.

Kannan, TR and Joel B Baseman (2000). "Expression of UGA-containing Mycoplasma genes in Bacillus subtilis". In: *Journal of Bacteriology* 182.9, pp. 2664–2667.

Karp, Peter D (1998). "What we do not know about sequence analysis and sequence databases." In: *Bioinformatics (Oxford, England)* 14.9, pp. 753–754.

Karro, John E et al. (2007). "Pseudogene. org: a comprehensive database and comparison platform for pseudogene annotation". In: *Nucleic Acids Research* 35.suppl_1, pp. D55–D60.

Keegan, Kevin P, Elizabeth M Glass, and Folker Meyer (2016). "MG-RAST, a metagenomics service for analysis of microbial community structure and function". In: *Microbial Environmental Genomics (MEG)*. Springer, pp. 207–233.

Keller, Oliver et al. (2011). "A novel hybrid gene prediction method employing protein multiple sequence alignments". In: *Bioinformatics* 27.6, pp. 757–763.

Kelley, David R et al. (2012). "Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering". In: *Nucleic Acids Research* 40.1, e9–e9.

Kersey, Paul J et al. (2010). "Ensembl Genomes: extending Ensembl across the taxonomic space". In: *Nucleic Acids Research* 38.suppl_1, pp. D563–D569.

Kim, Dongwan et al. (2020). "The architecture of SARS-CoV-2 transcriptome". In: *Cell*.

Klassen, Jonathan L and Cameron R Currie (2012). "Gene fragmentation in bacterial draft genomes: extent, consequences and mitigation". In: *BMC Genomics* 13.1, pp. 1–11.

Klimke, William et al. (2011). "Solving the problem: genome annotation standards before the data deluge". In: *Standards in Genomic Sciences* 5.1, pp. 168–193.

Klumpp, Stefan, Zhongge Zhang, and Terence Hwa (2009). "Growth rate-dependent global effects on gene expression in bacteria". In: *Cell* 139.7, pp. 1366–1375.

Kobayashi, Kan et al. (2012). "Structural basis for translation termination by archaeal RF1 and GTP-bound EF1α complex". In: *Nucleic Acids Research* 40.18, pp. 9319–9328.

Köhler, Sebastian et al. (2019). "Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources". In: *Nucleic Acids Research* 47.D1, pp. D1018–D1027.

Kolbe, Diana L and Sean R Eddy (2011). "Fast filtering for RNA homology search". In: *Bioinformatics* 27.22, pp. 3102–3109.

Konstantinidis, Konstantinos T and James M Tiedje (2004). "Trends between gene content and genome size in prokaryotic species with larger genomes". In: *Proceedings of the National Academy of Sciences* 101.9, pp. 3160–3165.

Koo, Peter K and Sean R Eddy (2019). "Representation learning of genomic sequence motifs with convolutional neural networks". In: *PLoS Computational Biology* 15.12, e1007560.

Korandla, Deepank R et al. (2020). "AssessORF: combining evolutionary conservation and proteomics to assess prokaryotic gene predictions". In: *Bioinformatics* 36.4, pp. 1022–1029.

Korkmaz, Gürkan et al. (2014). "Comprehensive analysis of stop codon usage in bacteria and its correlation with release factor abundance". In: *Journal of Biological Chemistry* 289.44, pp. 30334–30342.

Kowarsky, Mark et al. (2017). "Numerous uncharacterized and highly divergent microbes which colonize humans are revealed by circulating cell-free DNA". In: *Proceedings of the National Academy of Sciences* 114.36, pp. 9623–9628.

Koyama, Takahiko, Daniel Platt, and Laxmi Parida (2020). "Variant analysis of SARS-CoV-2 genomes". In: *Bulletin of the World Health Organization* 98.7, p. 495.

Krakauer, David C (2000). "Stability and evolution of overlapping genes". In: *Evolution* 54.3, pp. 731–739.

Kramer, Emily B and Philip J Farabaugh (2007). "The frequency of translational misreading errors in E. coli is largely determined by tRNA competition". In: *RNA* 13.1, pp. 87–96.

Krogh, Anders, I Saira Mian, and David Haussler (1994). "A hidden Markov model that finds genes in *E. coli* DNA". In: *Nucleic Acids Research* 22.22, pp. 4768–4778.

Kulmanov, Maxat and Robert Hoehndorf (2020). "DeepGOPlus: improved protein function prediction from sequence". In: *Bioinformatics* 36.2, pp. 422–429.

Kumar, Anuj (2009). "An overview of nested genes in eukaryotic genomes". In: *Eukaryotic Cell* 8.9, pp. 1321–1329.

Kumar, Naveen et al. (2018). "Evolution of codon usage bias in Henipaviruses is governed by natural selection and is host-specific". In: *Viruses* 10.11, p. 604.

Kuo, Chih-Horng and Howard Ochman (2010). "The extinction dynamics of bacterial pseudogenes". In: *PLoS Genetics* 6.8, e1001050.

Kwong, Jason C et al. (2015). "Whole genome sequencing in clinical and public health microbiology". In: *Pathology* 47.3, pp. 199–210.

Lamolle, Guillermo and Héctor Musto (2018). "Why Prokaryotes Genomes Lack Genes with Introns Processed by Spliceosomes?" In: *Journal of Molecular Evolution* 86.9, pp. 611–612.

Land, Miriam et al. (2015). "Insights from 20 years of bacterial genome sequencing". In: *Functional & integrative genomics* 15.2, pp. 141–161.

Lander, Eric S et al. (2001). "Initial sequencing and analysis of the human genome". In:

Langmead, Ben and Steven L Salzberg (2012). "Fast gapped-read alignment with Bowtie 2". In: *Nature Methods* 9.4, pp. 357–359.

Langridge, Gemma C et al. (2015). "Patterns of genome evolution that have accompanied host adaptation in Salmonella". In: *Proceedings of the National Academy of Sciences* 112.3, pp. 863–868.

Lapidus, Alla L and Anton I Korobeynikov (2021). "Metagenomic data assembly–the way of decoding unknown microorganisms". In: *Frontiers in Microbiology* 12, p. 653.

Lapierre, Pascal and J Peter Gogarten (2009). "Estimating the size of the bacterial pan-genome". In: *Trends in Genetics* 25.3, pp. 107–110.

Lathe, W et al. (2008). "Genomic data resources: challenges and promises". In: *Nature Education* 1.3, p. 2.

Lau, Susanna KP et al. (2015). "Severe acute respiratory syndrome (SARS) coronavirus ORF8 protein is acquired from SARS-related coronavirus from greater horseshoe bats through recombination". In: *Journal of Virology* 89.20, pp. 10532–10547.

Lau, Susanna KP et al. (2020). "Possible bat origin of severe acute respiratory syndrome coronavirus 2". In: *Emerging Infectious Diseases* 26.7, p. 1542.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lee, David, Oliver Redfern, and Christine Orengo (2007). "Predicting protein function from sequence and structure". In: *Nature Reviews Molecular Cell Biology* 8.12, pp. 995–1005.

Lerat, Emmanuelle and Howard Ochman (2005). "Recognizing the pseudogenes in bacterial genomes". In: *Nucleic Acids Research* 33.10, pp. 3125–3132.

Letunic, Ivica and Peer Bork (2021). "Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation". In: *Nucleic Acids Research* 49.W1, W293–W296.

Levasseur, Anthony and Pierre Pontarotti (2011). "The role of duplications in the evolution of genomes highlights the need for evolutionary-based approaches in comparative genomics". In: *Biology Direct* 6.1, pp. 1–12.

Levinthal, Cyrus (1969). "How to fold graciously". In: *Mossbauer Spectroscopy in Biological Systems* 67, pp. 22–24.

Levy, Arnon and Adrian Currie (2015). "Model organisms are not (theoretical) models". In: *The British Journal for the Philosophy of Science* 66.2, pp. 327–348.

Lewin, Harris A et al. (2018). "Earth BioGenome Project: Sequencing life for the future of life". In: *Proceedings of the National Academy of Sciences* 115.17, pp. 4325–4333.

Li, Dinghua et al. (2015). "MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph". In: *Bioinformatics* 31.10, pp. 1674–1676.

Li, Haoyang et al. (2020a). "Modern deep learning in bioinformatics". In: *Journal of Molecular Cell Biology* 12.11, pp. 823–827.

Li, Heng (May 2018). "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094–3100. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty191. URL: https://doi.org/10.1093/bioinformatics/bty191.

Li, Heng et al. (2009). "The sequence alignment/map format and SAMtools". In: *Bioinformatics* 25.16, pp. 2078–2079.

Li, Jin-Yan et al. (2020b). "The ORF6, ORF8 and nucleocapsid proteins of SARS-CoV-2 inhibit type I interferon signaling pathway". In: *Virus Research* 286, p. 198074.

Li, Wen-Hsiung, Takashi Gojobori, and Masatoshi Nei (1981). "Pseudogenes as a paradigm of neutral evolution". In: *Nature* 292.5820, pp. 237–239.

Li, Wendong et al. (2005). "Bats are natural reservoirs of SARS-like coronaviruses". In: *Science* 310.5748, pp. 676–679.

Lithwick, Gila and Hanah Margalit (2003). "Hierarchy of sequence-dependent features associated with prokaryotic translation". In: *Genome Research* 13.12, pp. 2665–2673.

Liu, Ping et al. (2020a). "Are pangolins the intermediate host of the 2019 novel coronavirus (SARS-CoV-2)?" In: *PLoS Pathogens* 16.5, e1008421.

Liu, Teng et al. (2020b). "Differential expression of viral transcripts from single-cell RNA sequencing of moderate and severe COVID-19 patients and its implications for case severity". In: *Frontiers in Microbiology* 11, p. 2568.

Liu, Yang et al. (2004). "Comprehensive analysis of pseudogenes in prokaryotes: widespread gene decay and failure of putative horizontally transferred genes". In: *Genome Biology* 5.9, pp. 1–11.

Liu-Wei, Wang et al. (2021). "DeepViral: prediction of novel virus-host interactions from protein sequences and infectious disease phenotypes." In:

Lobanov, Alexey V et al. (2010). "Dual functions of codons in the genetic code". In: *Critical Reviews in Biochemistry and Molecular Biology* 45.4, pp. 257–265.

Lobb, Briallen et al. (2020). "An assessment of genome annotation coverage across the bacterial tree of life". In: *Microbial Genomics* 6.3.

Logan, David C (2009). "Known knowns, known unknowns, unknown unknowns and the propagation of scientific enquiry". In: *Journal of Experimental Botany* 60.3, pp. 712–714.

Lomsadze, Alexandre et al. "GeneMarkS-2: Raising Standards of Accuracy in Gene Recognition". In: ().

Lomsadze, Alexandre et al. (2018). "Modeling leaderless transcription and atypical genes results in more accurate gene prediction in prokaryotes". In: *Genome Research* 28.7, pp. 1079–1089.

Lopes, Luciano Rodrigo, Giancarlo de Mattos Cardillo, and Paulo Bandiera Paiva (2020). "Molecular evolution and phylogenetic analysis of SARS-CoV-2 and hosts ACE2 protein suggest Malayan pangolin as intermediary host". In: *Brazilian Journal of Microbiology*, pp. 1–7.

Lu, Jennifer and Steven L Salzberg (2018). "Removing contaminants from databases of draft genomes". In: *PLoS Computational Biology* 14.6, e1006277.

Lu, Ponzy and Alexander Rich (1971). "The nature of the polypeptide chain termination signal". In: *Journal of Molecular Biology* 58.2, pp. 513–531.

Lu, Roujian et al. (2020). "Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding". In: *The Lancet* 395.10224, pp. 565–574.

Lukashin, Alexander V and Mark Borodovsky (1998). "GeneMark.hmm: new solutions for gene finding". In: *Nucleic Acids Research* 26.4, pp. 1107–1115.

Lukjancenko, Oksana, Trudy M Wassenaar, and David W Ussery (2010). "Comparison of 61 sequenced *Escherichia coli* genomes". In: *Microbial Ecology* 60.4, pp. 708–720.

Luria, Salvador E and Max Delbrück (1943). "Mutations of bacteria from virus sensitivity to virus resistance". In: *Genetics* 28.6, p. 491.

Lynch, Michael (2006). "Streamlining and simplification of microbial genome architecture". In: *Annu. Rev. Microbiol.* 60, pp. 327–349.

Macek, Boris et al. (2019). "Protein post-translational modifications in bacteria". In: *Nature Reviews Microbiology* 17.11, pp. 651–664.

Maddamsetti, Rohan et al. (2017). "Core genes evolve rapidly in the long-term evolution experiment with Escherichia coli". In: *Genome Biology and Evolution* 9.4, pp. 1072–1083.

Madhav, Nita et al. (2017). *Pandemics: risks, impacts, and mitigation*. The International Bank for Reconstruction and Development / The World Bank.

Madupu, Ramana et al. (2010). "Meeting report: a workshop on Best Practices in Genome Annotation". In: *Database* 2010.

Magadum, Santoshkumar et al. (2013). "Gene duplication as a major force in evolution". In: *Journal of genetics* 92.1, pp. 155–161.

Maguire, Finlay et al. (2020). "Metagenome-assembled genome binning methods with short reads disproportionately fail for plasmids and genomic islands". In: *Microbial Genomics* 6.10.

Mahmudi, Owais et al. (2015). "Gene-pseudogene evolution: a probabilistic approach". In: *BMC Genomics* 16.10, pp. 1–11.

Makrodimitris, Stavros, Roeland CHJ Van Ham, and Marcel JT Reinders (2020). "Automatic gene function prediction in the 2020's". In: *Genes* 11.11, p. 1264.

Malaiyan, Jeevan et al. (2020). "An update on origin of SARS-CoV-2: Despite closest identity, bat (RaTG13) and Pangolin derived Coronaviruses varied in the critical binding site and O-linked glycan residues". In: *Journal of Medical Virology*.

Maman, Leila Ghanbari et al. (2020). "Co-abundance analysis reveals hidden players associated with high methane yield phenotype in sheep rumen microbiome". In: *Scientific Reports* 10.1, pp. 1–12.

Manrai, Arjun K et al. (2016). "Genetic misdiagnoses and the potential for health disparities". In: *New England Journal of Medicine* 375.7, pp. 655–665.

Martiny, Adam C (2019). "High proportions of bacteria are culturable across major biomes". In: *The ISME Journal* 13.8, pp. 2125–2128.

Masella, Andre P et al. (2012). "PANDAseq: paired-end assembler for illumina sequences". In: *BMC Bioinformatics* 13.1, pp. 1–7.

Mathé, Catherine et al. (2002). "Current methods of gene prediction, their strengths and weaknesses". In: *Nucleic Acids Research* 30.19, pp. 4103–4117.

McInerney, James O, Alan McNally, and Mary J O'Connell (2017). "Why prokaryotes have pangenomes". In: *Nature microbiology* 2.4, pp. 1–5.

McLaren, William et al. (2016). "The Ensembl Variant Effect Predictor". In: *Genome Biology* 17.1, p. 122. ISSN: 1474-760X. DOI: 10.1186/s13059-016-0974-4. URL: https://doi.org/10.1186/s13059-016-0974-4.

Médigue, Claudine et al. (1999). "Detecting and analyzing DNA sequencing errors: toward a higher quality of the Bacillus subtilis genome sequence". In: *Genome Research* 9.11, pp. 1116–1127.

Medini, Duccio et al. (2005). "The microbial pan-genome". In: *Current Opinion in Genetics & Development* 15.6, pp. 589–594.

Meydan, Sezen, Nora Vazquez-Laslop, and Alexander S Mankin (2018). "Genes within genes in bacterial genomes". In: *Regulating with RNA in Bacteria and Archaea*, pp. 133–154.

Meydan, Sezen et al. (2019). "Retapamulin-assisted ribosome profiling reveals the alternative bacterial proteome". In: *Molecular cell* 74.3, pp. 481–493.

Meziti, Alexandra et al. (2021). "The reliability of metagenome-assembled genomes (MAGs) in representing natural populations: Insights from comparing MAGs against isolate genomes derived from the same fecal sample". In: *Applied and Environmental Microbiology* 87.6, e02593–20.

Michard, Céline and Patricia Doublet (2015). "Post-translational modifications are key players of the Legionella pneumophila infection strategy". In: *Frontiers in Microbiology* 6, p. 87.

Mihelčić, Matej, Tomislav Šmuc, and Fran Supek (2019). "Patterns of diverse gene functions in genomic neighborhoods predict gene function and phenotype". In: *Scientific Reports* 9.1, pp. 1–16.

Miller, Jason R, Sergey Koren, and Granger Sutton (2010). "Assembly algorithms for next-generation sequencing data". In: *Genomics* 95.6, pp. 315–327.

Miller, Melissa B and Bonnie L Bassler (2001). "Quorum sensing in bacteria". In: *Annual Reviews in Microbiology* 55.1, pp. 165–199.

Miravet-Verde, Samuel et al. (2019). "Unraveling the hidden universe of small proteins in bacterial genomes". In: *Molecular Systems Biology* 15.2, e8290.

Nagies, Falk SP et al. (2020). "A spectrum of verticality across genes". In: *PLoS Genetics* 16.11, e1009200.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Icml*.

Nakamura, Yoji et al. (2004). "Biased biological functions of horizontally transferred genes in prokaryotic genomes". In: *Nature Genetics* 36.7, pp. 760–766.

Nambou, Komi and Manawa Anakpa (2020). "Deciphering the co-adaptation of codon usage between respiratory coronaviruses and their human host uncovers candidate therapeutics for COVID-19". In: *Infection, Genetics and Evolution* 85, p. 104471.

Nielsen, Pernille and Anders Krogh (2005). "Large-scale prokaryotic gene prediction and comparison to genome annotation". In: *Bioinformatics* 21.24, pp. 4322–4329.

Nierman, William C et al. (2001). "Complete genome sequence of Caulobacter crescentus". In: *Proceedings of the National Academy of Sciences* 98.7, pp. 4136–4141.

Ning, Kang et al. (2010). "Examination of the relationship between essential genes in PPI network and hub proteins in reverse nearest neighbor topology". In: *BMC Bioinformatics* 11.1, pp. 1–14.

Nirenberg, Marshall and Philip Leder (1964). "RNA codewords and protein synthesis: The effect of trinucleotides upon the binding of sRNA to ribosomes". In: *Science* 145.3639, pp. 1399–1407.

Noguchi, Hideki, Jungho Park, and Toshihisa Takagi (2006). "MetaGene: prokaryotic gene finding from environmental genome shotgun sequences". In: *Nucleic Acids Research* 34.19, pp. 5623–5630.

Noguchi, Hideki, Takeaki Taniguchi, and Takehiko Itoh (2008). "MetaGeneAnnotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes". In: *DNA Research* 15.6, pp. 387–396.

Notari, Daniel Luis et al. (2014). "IntergenicDB: a database for intergenic sequences". In: *Bioinformation* 10.6, p. 381.

Odell, Sarah G et al. (2017). "The art of curation at a biological database: principles and application". In: *Current Plant Biology* 11, pp. 2–11.

ÓhÉigeartaigh, Seán S et al. (2014). "SearchDOGS bacteria, software that provides automated identification of potentially missed genes in annotated bacterial genomes". In: *Journal of bacteriology* 196.11, pp. 2030–2042.

Ohno, Susumu (2013). *Evolution by gene duplication*. Springer Science & Business Media.

O'Leary, Nuala A et al. (2016). "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation". In: *Nucleic Acids Research* 44.D1, pp. D733–D745.

Ollivier, Bernard et al. (2018). "Importance of Prokaryotes in the Functioning and Evolution of the Present and Past Geosphere and Biosphere". In: *Prokaryotes and Evolution*. Springer, pp. 57–129.

Olsen, Nikoline S et al. (2020). "Exploring the remarkable diversity of culturable Escherichia coli phages in the Danish Wastewater Environment". In: *Viruses* 12.9, p. 986.

Omotajo, Damilola et al. (2015). "Distribution and diversity of ribosome binding sites in prokaryotic genomes". In: *BMC Genomics* 16.1, pp. 1–8.

Orr, Mona Wu et al. (2020). "Alternative ORFs and small ORFs: shedding light on the dark proteome". In: *Nucleic Acids Research* 48.3, pp. 1029–1042.

O'Shea, Keiron and Ryan Nash (Nov. 2015). "An Introduction to Convolutional Neural Networks". In: *ArXiv e-prints*.

Page, Andrew J et al. (2015). "Roary: rapid large-scale prokaryote pan genome analysis". In: *Bioinformatics* 31.22, pp. 3691–3693.

Pallejà, Albert, Eoghan D Harrington, and Peer Bork (2008). "Large gene overlaps in prokaryotic genomes: result of functional constraints or mispredictions?" In: *BMC Genomics* 9.1, pp. 1–10.

Panicker, Indu S, Glenn F Browning, and Philip F Markham (2015). "The effect of an alternate start codon on heterologous expression of a PhoA fusion protein in mycoplasma gallisepticum". In: *PloS One* 10.5, e0127911.

Paradis, E. and K. Schliep (2019). "ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R". In: *Bioinformatics* 35, pp. 526–528.

Parker, Dane et al. (2014). "Genome sequence of bacterial interference strain Staphylococcus aureus 502A". In: *Genome Announcements* 2.2, e00284–14.

Parkin, Neil T et al. (2020). "Multi-laboratory comparison of next-generation to sanger-based sequencing for HIV-1 drug resistance genotyping". In: *Viruses* 12.7, p. 694.

Patz, Jonathan A et al. (2000). "Effects of environmental change on emerging parasitic diseases". In: *International Journal for Parasitology* 30.12-13, pp. 1395–1405.

Pavy, N. et al. (1999). "Evaluation of gene prediction software using a genomic data set: application to *Arabidopsis thaliana* sequences". In: *Bioinformatics* 15.11, pp. 887–899. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/15.11.887. URL: http://bioinformatics.oxfordjournals.org/cgi/content/long/15/11/887.

Pedersen, Kim and Kenn Gerdes (1999). "Multiple hok genes on the chromosome of Escherichia coli". In: *Molecular Microbiology* 32.5, pp. 1090–1102.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *The Journal of Machine Learning Research* 12, pp. 2825–2830.

Pennisi, Elizabeth (2008). *Proposal to'wikify'GenBank meets stiff resistance*.

Pereira, Filipe (2020). "Evolutionary dynamics of the SARS-CoV-2 ORF8 accessory gene". In: *Infection, Genetics and Evolution* 85, p. 104525.

Perlman, Stanley and Jason Netland (2009). "Coronaviruses post-SARS: update on replication and pathogenesis". In: *Nature Reviews Microbiology* 7.6, pp. 439–450.

Petrov, DA and DL Hartl (2000). "Pseudogene evolution and natural selection for a compact genome". In: *Journal of Heredity* 91.3, pp. 221–227.

Pickett, Brett E et al. (2012). "ViPR: an open bioinformatics database and analysis resource for virology Research". In: *Nucleic Acids Research* 40.D1, pp. D593–D598.

Pink, Ryan Charles et al. (2011). "Pseudogenes: pseudo-functional or key regulators in health and disease?" In: *RNA* 17.5, pp. 792–798.

Piplani, Sakshi et al. (2020). "In silico comparison of spike protein-ACE2 binding affinities across species; significance for the possible origin of the SARS-CoV-2 virus". In: *arXiv preprint arXiv:2005.06199*.

Pisithkul, Tippapha, Nishaben M Patel, and Daniel Amador-Noguez (2015). "Post-translational modifications as key regulators of bacterial metabolic fluxes". In: *Current Opinion in Microbiology* 24, pp. 29–37.

Povolotskaya, Inna S et al. (2012). "Stop codons in bacteria are not selectively equivalent". In: *Biology Direct* 7.1, pp. 1–13.

Price, Morgan N, Paramvir S Dehal, and Adam P Arkin (2010). "FastTree 2–approximately maximum-likelihood trees for large alignments". In: *PloS One* 5.3.

Pritchard, Arthur E et al. (1990). "Analysis of NADH dehydrogenase proteins, ATPase subunit 9, cytochrome b, and ribosomal protein L14 encoded in the mitochondrial DNA of Paramecium". In: *Nucleic Acids Research* 18.1, pp. 163–171.

Pruitt, Kim D, Tatiana Tatusova, and Donna R Maglott (2007). "NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins". In: *Nucleic Acids Research* 35.suppl_1, pp. D61–D65.

Quince, Christopher et al. (2017). "DESMAN: a new tool for de novo extraction of strains from metagenomes". In: *Genome Biology* 18.1, pp. 1–22.

Quince, Christopher et al. (2021). "STRONG: metagenomics strain resolution on assembly graphs". In: *Genome Biology* 22.1, pp. 1–34.

Quinlan, Aaron R and Ira M Hall (2010). "BEDTools: a flexible suite of utilities for comparing genomic features". In: *Bioinformatics* 26.6, pp. 841–842.

Rasko, David A et al. (2008). "The pangenome structure of Escherichia coli: comparative genomic analysis of E. coli commensal and pathogenic isolates". In: *Journal of Bacteriology* 190.20, pp. 6881–6893.

Raza, S (2020). "Artificial intelligence for genomic medicine". In: *Cambridge: PHG Foundation, University of Cambridge*.

Régnier, Philippe and Paulo E Marujo (2003). "Polyadenylation and Degradation of RNA". In: *Translation Mechanisms*, p. 184.

Revell, Liam J. (2012). "phytools: An R package for phylogenetic comparative biology (and other things)." In: *Methods in Ecology and Evolution* 3, pp. 217–223.

Rho, Mina, Haixu Tang, and Yuzhen Ye (2010). "FragGeneScan: predicting genes in short and error-prone reads". In: *Nucleic Acids Research* 38.20, e191–e191.

Ribet, David and Pascale Cossart (2010). "Post-translational modifications in host cells during bacterial infection". In: *FEBS letters* 584.13, pp. 2748–2758.

Rice, Alan M. et al. (2020). "Evidence for strong mutation bias towards, and selection against, U content in SARS-CoV-2: implications for vaccine design". eng. In: *Molecular Biology and Evolution*. PMC7454790[pmcid], msaa188. ISSN: 1537-1719. DOI: 10.1093/molbev/msaa188. URL: https://pubmed.ncbi.nlm.nih.gov/32687176.

Richardson, Emily J and Mick Watson (2013). "The automatic annotation of bacterial genomes". In: *Briefings in Bioinformatics* 14.1, pp. 1–12.

Rio, Angela Lopez-del et al. (2020). "Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction". In: *Scientific Reports* 10.1, pp. 1–14.

Ritter, Deborah I et al. (2019). "A case for expert curation: an overview of cancer curation in the clinical genome resource (ClinGen)". In: *Molecular Case Studies* 5.5, a004739.

Roberts, Jeffrey W (2019). "Mechanisms of bacterial transcription termination". In: *Journal of Molecular Biology* 431.20, pp. 4030–4039.

Roberts, Richard J (2004). "Identifying protein function—a call for community action". In: *PLoS Biology* 2.3, e42.

Robertson, Michael P et al. (2004). "The structure of a rigorously conserved RNA element within the SARS virus genome". In: *PLoS Biology* 3.1, e5.

Robinson, James T et al. (2011). "Integrative genomics viewer". In: *Nature Biotechnology* 29.1, pp. 24–26.

Robison, Keith, Walter Gilbert, and George M Church (1994). "Large scale bacterial gene discovery by similarity search". In: *Nature Genetics* 7.2, pp. 205–214.

Rogozin, Igor B et al. (2002). "Congruent evolution of different classes of non-coding DNA in prokaryotic genomes". In: *Nucleic Acids Research* 30.19, pp. 4264–4271.

Rosche, William A and Patricia L Foster (2000). "Determining mutation rates in bacterial populations". In: *Methods* 20.1, pp. 4–17.

Ross, Michael G et al. (2013). "Characterizing and measuring bias in sequence data". In: *Genome Biology* 14.5, pp. 1–20.

Rubino, Francesco et al. (2017). "Divergent functional isoforms drive niche specialisation for nutrient acquisition and use in rumen microbiome". In: *The ISME Journal* 11.4, pp. 932–944.

Russell, James J et al. (2017). "Non-model model organisms". In: *BMC Biology* 15.1, pp. 1–31.

Ryoji, Masaru, Karen Hsia, and Akira Kaji (1983). "Read-through translation". In: *Trends in Biochemical Sciences* 8.3, pp. 88–90.

Sabath, Niv, Dan Graur, and Giddy Landan (2008). "Same-strand overlapping genes in bacteria: compositional determinants of phase bias". In: *Biology Direct* 3.1, p. 36.

Sakai, Hiroyuki D and Norio Kurosawa (2018). "Saccharolobus caldissimus gen. nov., sp. nov., a facultatively anaerobic iron-reducing hyperthermophilic archaeon isolated from an acidic terrestrial hot spring, and reclassification of Sulfolobus solfataricus as Saccharolobus solfataricus comb. nov. and Sulfolobus shibatae as Saccharolobus shibatae comb. nov." In: *International Journal of Systematic and Evolutionary Microbiology* 68.4, pp. 1271–1278.

Salamov, AA and VV Solovyev (1997). "The Gene-Finder computer tools for analysis of human and model organisms genome sequences". In: *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Halkidiki, Greece*, pp. 294–302.

Salamov, Victor Solovyevand Asaf and A Solovyevand (2011). "Automatic annotation of microbial genomes and metagenomic sequences". In: *Metagenomics and its applications in agriculture. Nova Science Publishers, Hauppauge*, pp. 61–78.

Salzberg, Steven L (2007). "Genome re-annotation: a wiki solution?" In: *Genome Biology* 8.1, p. 102.

– (2019). "Next-generation genome annotation: we still struggle to get it right". In:

Salzberg, Steven L et al. (1998). "Microbial gene identification using interpolated Markov models". In: *Nucleic Acids Research* 26.2, pp. 544–548.

Sangar, Vineet et al. (2007). "Quantitative sequence-function relationships in proteins based on gene ontology". In: *BMC Bioinformatics* 8.1, pp. 1–15.

Sanjuán, Rafael and Pilar Domingo-Calap (2016). "Mechanisms of viral mutation". In: *Cellular and Molecular Life Sciences* 73.23, pp. 4433–4448.

Sarkar, Nilima (1997). "Polyadenylation of mRNA in prokaryotes". In: *Annual review of biochemistry* 66.1, pp. 173–197.

Sarker, Iqbal H (2021). "Machine learning: Algorithms, real-world applications and research directions". In: *SN Computer Science* 2.3, pp. 1–21.

Saxer, Gerda et al. (2014). "Mutations in global regulators lead to metabolic selection during adaptation to complex environments". In: *PLoS Genetics* 10.12, e1004872.

Sboner, Andrea et al. (2011). "The real cost of sequencing: higher than you think!" In: *Genome Biology* 12.8, pp. 1–10.

Schafer, Joseph L and John W Graham (2002). "Missing data: our view of the state of the art." In: *Psychological Methods* 7.2, p. 147.

Schmitt, Emmanuelle et al. (2020). "Recent advances in archaeal translation initiation". In: *Frontiers in Microbiology* 11, p. 2259.

Schnoes, Alexandra M et al. (2009). "Annotation error in public databases: misannotation of molecular function in enzyme superfamilies". In: *PLoS Computational Biology* 5.12, e1000605.

Schnoes, Alexandra M et al. (2013). "Biases in the experimental annotations of protein function and their effect on our understanding of protein function space". In: *PLoS Computational Biology* 9.5, e1003063.

Schoeman, Dewald and Burtram C Fielding (2019). "Coronavirus envelope protein: current knowledge". In: *Virology Journal* 16.1, pp. 1–22.

Schrader, Jared M et al. (2014). "The coding and noncoding architecture of the Caulobacter crescentus genome". In: *PLoS Genetics* 10.7, e1004463.

Seemann, Torsten (2014). "Prokka: rapid prokaryotic genome annotation". In: *Bioinformatics* 30.14, pp. 2068–2069.

Segerman, Bo (2012). "The genetic integrity of bacterial species: the core genome and the accessory genome, two different stories". In: *Frontiers in Cellular and Infection Microbiology* 2, p. 116.

Sela, Itamar, Yuri I Wolf, and Eugene V Koonin (2016). "Theory of prokaryotic genome evolution". In: *Proceedings of the National Academy of Sciences* 113.41, pp. 11399–11407.

Sengupta, Jayati, Rajendra K Agrawal, and Joachim Frank (2001). "Visualization of protein S1 within the 30S ribosomal subunit and its interaction with messenger RNA". In: *Proceedings of the National Academy of Sciences* 98.21, pp. 11991–11996.

Seshadri, Rekha et al. (2018). "Cultivation and sequencing of rumen microbiome members from the Hungate1000 Collection". In: *Nature Biotechnology* 36.4, pp. 359–367.

Shah, Shiraz A et al. (2019). "Comprehensive search for accessory proteins encoded with archaeal and bacterial type III CRISPR-cas gene cassettes reveals 39 new cas gene families". In: *RNA Biology* 16.4, pp. 530–542.

Shariat, Nikki W, Ruth E Timme, and Abigail T Walters (2021). "Phylogeny of Salmonella enterica subspecies arizonae by whole-genome sequencing reveals high incidence of polyphyly and low phase 1 H antigen variability". In: *Microbial Genomics* 7.2.

Sharp, Paul M, Therese MF Tuohy, and Krzysztof R Mosurski (1986). "Codon usage in yeast: cluster analysis clearly differentiates highly and lowly expressed genes". In: *Nucleic Acids Research* 14.13, pp. 5125–5143.

Shean, Ryan C et al. (2019). "VAPiD: a lightweight cross-platform viral annotation pipeline and identification tool to facilitate virus genome submissions to NCBI GenBank". In: *BMC Bioinformatics* 20.1, pp. 1–8.

Shendure, Jay and Hanlee Ji (2008). "Next-generation DNA sequencing". In: *Nature Biotechnology* 26.10, pp. 1135–1145.

Sherman, Fred, John W Stewart, and Susumu Tsunasawa (1985). "Methionine or not methionine at the beginning of a protein". In: *Bioessays* 3.1, pp. 27–31.

Shi, Weibing et al. (2014). "Methane yield phenotypes linked to differential gene expression in the sheep rumen microbiome". In: *Genome Research* 24.9, pp. 1517–1525.

Sieber, Patricia, Matthias Platzer, and Stefan Schuster (2018). "The definition of open reading frame revisited". In: *Trends in Genetics* 34.3, pp. 167–170.

Sievers, Fabian and Desmond G Higgins (2018). "Clustal Omega for making accurate alignments of many protein sequences". In: *Protein Science* 27.1, pp. 135–145.

Smith, Everett Clinton, Nicole R Sexton, and Mark R Denison (2014). "Thinking outside the triangle: replication fidelity of the largest RNA viruses". In: *Annual Review of Virology* 1, pp. 111–132.

Sommer, Markus J and Steven L Salzberg (2021). "Balrog: A universal protein model for prokaryotic gene prediction". In: *PLoS Computational Biology* 17.2, e1008727.

Sridhar, Jayavel et al. (2011). "Junker: an intergenic explorer for bacterial genomes". In: *Genomics, Proteomics & Bioinformatics* 9.4-5, pp. 179–182.

Srinivasan, Gayathri, Carey M James, and Joseph A Krzycki (2002). "Pyrrolysine encoded by UAG in Archaea: charging of a UAG-decoding specialized tRNA". In: *Science* 296.5572, pp. 1459–1462.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Staden, Rodger (1984). "Measurements of the effects that coding for a protein has on a DNA sequence and their use for finding genes". In:

Stadtman, Thressa C (1996). "Selenocysteine". In: *Annual Review of Biochemistry* 65.1, pp. 83–100.

Stamatakis, Alexandros (2014). "RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies". In: *Bioinformatics* 30.9, pp. 1312–1313.

Stanke, Mario and Burkhard Morgenstern (2005). "AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints". In: *Nucleic Acids Research* 33.suppl_2, W465–W467.

Stewart, Robert D et al. (2018). "Assembly of 913 microbial genomes from metagenomic sequencing of the cow rumen". In: *Nature Communications* 9.1, pp. 1–11.

Stoeger, Thomas et al. (2018). "Large-scale investigation of the reasons why potentially important genes are ignored". In: *PLoS Biology* 16.9, e2006643.

Storz, Gisela, Yuri I Wolf, and Kumaran S Ramamurthi (2014). "Small proteins can no longer be ignored". In: *Annual Review of Biochemistry* 83, pp. 753–777.

Stothard, Paul (2000). "The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences". In: *Biotechniques* 28.6, pp. 1102–1104.

Stothard, Paul and David S Wishart (2006). "Automated bacterial genome analysis and annotation". In: *Current Opinion in Microbiology* 9.5, pp. 505–510.

Su, Mingming et al. (2013). "Small proteins: untapped area of potential biological importance". In: *Frontiers in Genetics* 4, p. 286.

Su, Shuo et al. (2016). "Epidemiology, genetic recombination, and pathogenesis of coronaviruses". In: *Trends in Microbiology* 24.6, pp. 490–502.

Sun, Yi-Cheng, B Joseph Hinnebusch, and Creg Darby (2008). "Experimental evidence for negative selection in the evolution of a Yersinia pestis pseudogene". In: *Proceedings of the National Academy of Sciences* 105.23, pp. 8097–8101.

Sunagawa, Shinichi et al. (2020). "Tara Oceans: towards global ocean ecosystems Biology". In: *Nature Reviews Microbiology* 18.8, pp. 428–445.

Sutton, Thomas DS et al. (2019). "Choice of assembly software has a critical impact on virome characterisation". In: *Microbiome* 7.1, pp. 1–15.

Szklarczyk, Damian et al. (2019). "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". In: *Nucleic Acids Research* 47.D1, pp. D607–D613.

Taft, Ryan J, Michael Pheasant, and John S Mattick (2007). "The relationship between non-protein-coding DNA and eukaryotic complexity". In: *Bioessays* 29.3, pp. 288–299.

Tamames, Javier, Marta Cobo-Simón, and Fernando Puente-Sánchez (2019). "Assessing the performance of different approaches for functional and taxonomic annotation of metagenomes". In: *BMC Genomics* 20.1, pp. 1–16.

Tang, Binhua et al. (2019). "Recent advances of deep learning in bioinformatics and computational Biology". In: *Frontiers in Genetics* 10, p. 214.

Tanizawa, Yasuhiro, Takatomo Fujisawa, and Yasukazu Nakamura (2018). "DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication". In: *Bioinformatics* 34.6, pp. 1037–1039.

Tate, John G et al. (2019). "COSMIC: the catalogue of somatic mutations in cancer". In: *Nucleic Acids Research* 47.D1, pp. D941–D947.

Tatusov, Roman L et al. (2000). "The COG database: a tool for genome-scale analysis of protein functions and evolution". In: *Nucleic Acids Research* 28.1, pp. 33–36.

Tatusova, Tatiana et al. (2016). "NCBI prokaryotic genome annotation pipeline". In: *Nucleic Acids Research* 44.14, pp. 6614–6624.

Taylor, Justin K et al. (2015). "Severe acute respiratory syndrome coronavirus ORF7a inhibits bone marrow stromal antigen 2 virion tethering through a novel mechanism of glycosylation interference". In: *Journal of Virology* 89.23, pp. 11820–11833.

Tengs, Torstein, Charles F Delwiche, and Christine M Jonassen (2020). "A mobile genetic element in the SARS-CoV-2 genome is shared with multiple insect species". In: *bioRxiv*.

Tengs, Torstein and Christine M Jonassen (2016). "Distribution and evolutionary history of the mobile genetic element s2m in coronaviruses". In: *Diseases* 4.3, p. 27.

Teoh, Kim-Tat et al. (2010). "The SARS coronavirus E protein interacts with PALS1 and alters tight junction formation and epithelial morphogenesis". In: *Molecular Biology of the Cell* 21.22, pp. 3838–3852.

Tettelin, Hervé et al. (2005). "Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome"". In: *Proceedings of the National Academy of Sciences* 102.39, pp. 13950–13955.

Thomas, Torsten, Jack Gilbert, and Folker Meyer (2012). "Metagenomics-a guide from sampling to data analysis". In: *Microbial Informatics and Experimentation* 2.1, pp. 1–12.

Thompson, Luke R et al. (2017). "A communal catalogue reveals Earth's multiscale microbial diversity". In: *Nature* 551.7681, pp. 457–463.

Thorpe, Harry A et al. (2017). "Comparative analyses of selection operating on non-translated intergenic regions of diverse bacterial species". In: *Genetics* 206.1, pp. 363–376.

Tramontano, A and MF Macchiato (1986). "Probability of coding of a DNA sequence: an algorithm to predict translated reading frames from their thermodynamic characteristics". In: *Nucleic Acids Research* 14.1, pp. 127–135.

Troudet, Julien et al. (2017). "Taxonomic bias in biodiversity data and societal preferences". In: *Scientific Reports* 7.1, pp. 1–14.

Trüper, HG (1992). "Prokaryotes: an overview with respect to biodiversity and environmental importance". In: *Biodiversity & Conservation* 1.4, pp. 227–236.

Tsai, Chen-Hsun et al. (2015). "Genome-wide analyses in bacteria show small-RNA enrichment for long and conserved intergenic regions". In: *Journal of bacteriology* 197.1, pp. 40–50.

Tse, Herman et al. (2010). "Natural selection retains overrepresented out-of-frame stop codons against frameshift peptides in prokaryotes". In: *BMC Genomics* 11.1, pp. 1–13.

Turanov, Anton A et al. (2009). "Genetic code supports targeted insertion of two amino acids by one codon". In: *Science* 323.5911, pp. 259–261.

Tutar, Yusuf (2012). "Pseudogenes". In: *Comparative and Functional Genomics* 2012.

UniProt Consortium (2019). "UniProt: a worldwide hub of protein knowledge". In: *Nucleic Acids Research* 47.D1, pp. D506–D515.

Valdés, Jorge et al. (2008). "Acidithiobacillus ferrooxidans metabolism: from genome sequence to industrial applications". In: *BMC Genomics* 9.1, pp. 1–24.

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Van Rossum, Thea et al. (2020). "Diversity within species: interpreting strains in microbiomes". In: *Nature Reviews Microbiology* 18.9, pp. 491–506.

VanOrsdel, Caitlin E et al. (2018). "Identifying new small proteins in Escherichia coli". In: *Proteomics* 18.10, p. 1700064.

Villegas, Andre and Andrew M Kropinski (2008). "An analysis of initiation codon utilization in the Domain Bacteria–concerns about the quality of bacterial genome annotation". In: *Microbiology* 154.9, pp. 2559–2661.

Vollmers, John, Sandra Wiegand, and Anne-Kristin Kaster (2017). "Comparing and evaluating metagenome assembly tools from a microbiologist's perspective-not only size matters!" In: *PloS One* 12.1, e0169662.

Walt, Andries Johannes Van der et al. (2017). "Assembling metagenomes, one community at a time". In: *BMC Genomics* 18.1, pp. 1–13.

Wang, Jun et al. (2003). "Vertebrate gene predictions and the problem of large genes". In: *Nature Reviews Genetics* 4.9, p. 741.

Wang, Zhong, Mark Gerstein, and Michael Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature Reviews Genetics* 10.1, pp. 57–63.

Wang, Zhuo, Yazhu Chen, and Yixue Li (2004). "A brief review of computational gene prediction methods". In: *Genomics, Proteomics & Bioinformatics* 2.4, pp. 216–221.

Warren, Andrew S. et al. (2010). "Missing genes in the annotation of prokaryotic genomes". In: *BMC Bioinformatics* 11.1, p. 131.

Weber, Stefanie, Christina Ramirez, and Walter Doerfler (2020). "Signal hotspot mutations in SARS-CoV-2 genomes evolve as the virus spreads and actively replicates in different parts of the world". In: *Virus Research* 289, p. 198170.

Weiss, Susan R (2020). "Forty years with coronaviruses". In: *Journal of Experimental Medicine* 217.5.

Wick, Ryan R et al. (2017). "Unicycler: resolving bacterial genome assemblies from short and long sequencing reads". In: *PLoS Computational Biology* 13.6, e1005595.

Wilkins, Laetitia GE et al. (2019). "Metagenome-assembled genomes provide new insight into the microbial diversity of two thermal pools in Kamchatka, Russia". In: *Scientific Reports* 9.1, pp. 1–15.

Wong, Hui Hui et al. (2018). "Accessory proteins 8b and 8ab of severe acute respiratory syndrome coronavirus suppress the interferon signaling pathway by mediating ubiquitin-dependent rapid degradation of interferon regulatory factor 3". In: *Virology* 515, pp. 165–175.

Wong, Tit-Yee et al. (2008). "Role of premature stop codons in bacterial evolution". In: *Journal of Bacteriology* 190.20, pp. 6718–6725.

Wood, Derrick E, Jennifer Lu, and Ben Langmead (2019). "Improved metagenomic analysis with Kraken 2". In: *Genome Biology* 20.1, pp. 1–13.

Wood, Derrick E et al. (2012). "Thousands of missed genes found in bacterial genomes and their analysis with COMBREX". In: *Biology Direct* 7.1, p. 37.

Wrapp, Daniel et al. (2020). "Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation". In: *Science* 367.6483, pp. 1260–1263. ISSN: 0036-8075. DOI: 10.1126/science.abb2507. URL: https://science.sciencemag.org/content/367/6483/1260.

Wrobel, Antoni G et al. (2020). "SARS-CoV-2 and bat RaTG13 spike glycoprotein structures inform on virus evolution and furin-cleavage effects". In: *Nature Structural & Molecular Biology* 27.8, pp. 763–767.

Wu, Kailang et al. (2011). "A virus-binding hot spot on human angiotensin-converting enzyme 2 is critical for binding of two different coronaviruses". In: *Journal of Virology* 85.11, pp. 5331–5337.

Wu, Zhiqiang et al. (2021). "SARS-CoV-2's origin should be investigated worldwide for pandemic prevention". In: *The Lancet* 398.10308, pp. 1299–1303.

Xavier, Joana C, Kiran Raosaheb Patil, and Isabel Rocha (2018). "Metabolic models and gene essentiality data reveal essential and conserved metabolism in prokaryotes". In: *PLoS Computational Biology* 14.11, e1006556.

Xiao, Kangpeng et al. (2020). "Isolation of SARS-CoV-2-related coronavirus from Malayan pangolins". In: *Nature*, pp. 1–4.

Xing, Weijia et al. (2010). "Anatomy of the epidemiological literature on the 2003 SARS outbreaks in Hong Kong and Toronto: a time-stratified review". In: *PLoS Medicine* 7.5, e1000272.

Xu, Lin et al. (2006). "Average gene length is highly conserved in prokaryotes and eukaryotes and diverges only between the two kingdoms". In: *Molecular Biology and Evolution* 23.6, pp. 1107–1108.

Yang, Aimin et al. (2020). "Review on the Application of Machine Learning Algorithms in the Sequence Data Mining of DNA". In: *Frontiers in Bioengineering and Biotechnology* 8, p. 1032.

Yates, Andrew D et al. (2020). "Ensembl 2020". In: *Nucleic Acids Research* 48.D1, pp. D682–D688.

Yi, Huiguang (2020). "2019 novel coronavirus is undergoing active recombination". In: *Clinical Infectious Diseases*.

Yok, Non G and Gail L Rosen (2011). "Combining gene prediction methods to improve metagenomic gene annotation". In: *BMC Bioinformatics* 12.1, p. 20.

Yoshinaka, Y et al. (1985). "Translational readthrough of an amber termination codon during synthesis of feline leukemia virus protease". In: *Journal of Virology* 55.3, pp. 870–873.

You, Ronghui, Xiaodi Huang, and Shanfeng Zhu (2018). "DeepText2GO: Improving large-scale protein function prediction with deep semantic text representation". In: *Methods* 145, pp. 82–90.

You, Ronghui et al. (2018). "GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank". In: *Bioinformatics* 34.14, pp. 2465–2473.

Yu, Guangchuang (2020). "Using ggtree to Visualize Data on Tree-Like Structures". In: *Current Protocols in Bioinformatics* 69.1, e96.

Zaaijer, Sophie et al. (2016). "Using mobile sequencers in an academic classroom". In: *elife* 5.

Zhang, Hong et al. (2020a). "Metabolic stress promotes stop-codon readthrough and phenotypic heterogeneity". In: *Proceedings of the National Academy of Sciences* 117.36, pp. 22167–22172.

Zhang, Tao, Qunfu Wu, and Zhigang Zhang (2020). "Probable pangolin origin of SARS-CoV-2 associated with the COVID-19 outbreak". In: *Current Biology*.

Zhang, Yan and Vadim N Gladyshev (2007). "High content of proteins containing 21st and 22nd amino acids, selenocysteine and pyrrolysine, in a symbiotic deltaproteobacterium of gutless worm Olavius algarvensis". In: *Nucleic Acids Research* 35.15, pp. 4952–4963.

Zhang, Yan et al. (2006a). "Dynamic evolution of selenocysteine utilization in bacteria: a balance between selenoprotein loss and evolution of selenocysteine from redox active cysteine residues". In: *Genome Biology* 7.10, pp. 1–17.

Zhang, Yong-Zhen and Edward C. Holmes (2020). "A Genomic Perspective on the Origin and Emergence of SARS-CoV-2". In: *Cell* 181.2, pp. 223 –227. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2020.03.035. URL: http://www.sciencedirect.com/science/article/pii/S0092867420303287.

Zhang, Zhaolei et al. (2006b). "PseudoPipe: an automated pseudogene identification pipeline". In: *Bioinformatics* 22.12, pp. 1437–1439.

Zhang, Zheng et al. (2020b). "Estimate of the sequenced proportion of the global prokaryotic genome". In: *Microbiome* 8.1, pp. 1–9.

Zhong, Nanshan and Guangqiao Zeng (2006). "What we have learnt from SARS epidemics in China". In: *BMJ* 333.7564, pp. 389–391.

Zhou, Peng et al. (2020). "A pneumonia outbreak associated with a new coronavirus of probable bat origin". In: *nature* 579.7798, pp. 270–273.

Zhu, Wenhan, Alexandre Lomsadze, and Mark Borodovsky (2010). "Ab initio gene identification in metagenomic sequences". In: *Nucleic Acids Research* 38.12, e132–e132.

Zhu, Zhixing et al. (2020). "From SARS and MERS to COVID-19: a brief summary and comparison of severe acute respiratory infections caused by three highly pathogenic human coronaviruses". In: *Respiratory Research* 21.1, pp. 1–14.