Image Representations of DNA allow Classification by Convolutional Neural Networks

Joshua Hope

MSc by Research

University of York

Biology

12/2020

Abstract

In metagenomic analyses the rapid and accurate identification of DNA sequences is important. This is confounded by the existence of novel species not contained in databases. There exist many methods to identify sequences, but with the increasing amounts of sequencing data from high-throughput technologies, the use of new deep learning methods are made more viable. In an attempt to address this it was decided to use Convolutional Neural Networks (CNNs) to classify DNA sequences of archaea, which are important in anaerobic digestion. CNNs were trained on two different image representations of DNA sequences, Chaos Game Representation (CGR) and Reshape. Three phyla of archaea and randomly generated sequences were used. These were compared against simpler machine learning models trained on the 4-mer and 7-mer frequencies of the same sequences. It was found that the simpler models performed better than CNNs trained on either image representation, and that Reshape was the poorest representation. However, by shuffling sequences whilst preserving 4-mer count it was found that the Reshape model had learnt 4-mers as an important feature. It was also found that the Reshape model was able to perform equally well without depending on the use of 4-mers, indicating that certain training regimes may uncover novel features. The errors of these models were also random or in weak disagreement, suggesting ensemble methods would be viable and help to identify problematic sequences.

Contents

Acknowledgements and Declaration

I would like to acknowledge the support of Professor James Chong and the Chong lab throughout the research process.

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Introduction

Microbial Communities

Microbial communities are diverse and important in a range of environments. They are important in the Earth's biogeochemical cycles: in soil they impact nutrient cycling and carbon sequestration (Grosso et al. 2018); at the deep sea floor they cycle buried nutrients (Román et al. 2019); and in the human gut they play roles in nutrition and disease, including autoimmune disorders (Shanahan et al. 2017; Chiang et al. 2019). They also offer a potential reservoir for pharmaceuticals and cosmetics of varying functions (Demain 2014). A better understanding of microbial communities would offer insights into the factors affecting community make-up and the functional mechanisms of metabolism specific to species, groups of species, and the geochemical system as a whole (Ofek-Lalzar et al. 2014). As such, microbial communities necessitated the development of the field of metagenomics, which considers all the genomes in a community (the metagenome), but methods are still being refined and developed. This will have many implications in genomic understanding, the health of humans and animals, the preservation of wild microbial communities and their finessed manipulation in industry.

Anaerobic Digestion

With the increasing global population, the amount of organic waste from food, food production, and wastewater is likely to increase as a result of, or to meet the demands of, the population. Anaerobic digestion (AD) is a process carried out by microbes in the absence of oxygen that breaks down organic matter and produces methane and carbon dioxide as a result, as well as digestate and waste materials. This is an important process in the global cycling of carbon, but due to the value of the end-products, systems have been engineered to not only facilitate the breakdown of organic waste matter, but also recover these products. This represents a potential source of renewable energy, as not only would it be processing a waste product aiding in its disposal, but compared to fossil fuels it has the potential to be a lower source of greenhouse gas emissions (Angelidaki et al. 2011). AD therefore has promise as a technology to help meet the demands from two areas that will grow with population: waste disposal and energy. AD is a multi-stage process, with no one member of the microbial community capable of complete digestion in an anaerobic digester (Sundberg et al. 2013). One key group of organisms in the AD process and the production of methane are called methanogens, all known of which are archaea. A range of feedstocks can be anaerobically digested as a method of treatment, including municipal solid waste (Rao et al. 2000), wastewater and sewage sludge (Gong, Ren, and Xing 2005; Suryawanshi, Chaudhari, and Kothari 2010), and waste from farming and livestock, which can be rich in lignocellulose (Alvarez, Villca, and Lidén 2006).

During the treatment of wastewater, AD enables heavy metals to be removed in leachate (Selling, Håkansson, and Björnsson 2008) and transforms certain toxins (Saini et al. 2003). However, byproducts include nitrous oxide, and methane itself is a greenhouse gas with a global warming effect far greater than carbon dioxide (Paolini et al. 2018). As such, understanding and control of the process is important, at both mechanical and chemical levels.

Methanogens

The production of methane in laboratory conditions was found to be positively related to taxa diversity, and it is suggested that not only do rare species play an important role, but that the communities also have little functional redundancy, hence why loss of taxa reduces methane production (Sierocinski et al. 2018). These are the organisms which perform methanogenesis. This is the final step in the AD process and relies on the reactions of many other species, including bacteria. Hydrolysis, acidogenesis, and acetogenesis make the reactants biologically available for methanogenesis. The production of methane is performed via two main pathways, acetoclastic and hydrogenotrophic methanogenesis, but is also possible through the methylotrophic pathway. Not all species of methanogens possess all three pathways; in fact, it is thought that only those in the order Methanosarcinales do (Yuchen Liu and Whitman 2008). The material being treated is hydrolysed by the microbial community into soluble monomers such sugars or amino acids, which are then fermented.

The exact make-up of the community depends on many factors, including the feedstock being digested, and shows fluctuations over time (Yi et al. 2014). After fermentation the hydrogen is either used with carbon dioxide by hydrogenotrophic methanogens to produce methane, or converted into acetic acid in a process called acetogenesis, performed by strict bacterial anaerobes. It can then be used by acetoclastic methanogens to produce methane and carbon dioxide. Methylotrophic methanogens use one-carbon (C1) compounds such as methylamines to produce methane and various byproducts. These are not the only uses of the intermediate products, for example, acetate can be degraded by syntrophic acetate oxidisers before obligate interspecies hydrogen transfer to a hydrogenotrophic methanogen (Zinder and Koch 1984). However, the overall flow should be towards the production of methane, and the already-mentioned dependencies between members of the community should also be more apparent. It should be noted that the population dynamics can also result in shifts in metabolic activity within genera, such as under essential trace element deprivation, which leads to a shift in *Methanosarcina* from acetoclastic and methylotrophic to hydrogenotrophic methanogenesis (Wintsche et al. 2018).

As mentioned, all known methanogens are archaea. Until recently it was thought that they also belonged to a single phylum inside archaea, Euryarchaeota, and that this is likely where the capability evolved (Y. Liu 2010). However, multiple archaea from outside Euryarchaeota have been shown to possess homologs of the genes required for methanogenesis, for example, the phyla Bathyarchaeota (Evans et al. 2015) and Verstraetearchaeota. Verstraetearchaeota are likely capable of methylotrophic (Vanwonterghem et al. 2016) and hydrogenotrophic (Berghuis et al. 2018) methanogenesis. This has implications for how methanogenesis evolved, but also emphasises that much is unknown about archaeal lineages and their functionality. Indeed, they were initially thought to be species of bacteria (Woese, Kandler, and Wheelis 1990), and it has been largely thanks to new studies using metagenomics that advances in the reconstruction of their phylogeny has been made (Zaremba-Niedzwiedzka et al. 2017; Castelle and Banfield 2018). Even inside Euryarchaeota new classes have been proposed and accepted in recent years (Borrel et al. 2014; Nobu et al. 2016). However, work still remains to discover the full functionality of archaea.

A key step in determining whether a species or group of species is novel or performs a novel function is identification and comparison. This means that relationships over time can be assessed and that differences in genes can inform experimental hypotheses. One way to identify a species is with the use of conserved marker genes.

16s rRNA

16s rRNA is present in the 30S subunit of the ribosome of prokaryotes and is essential in the synthesis of all proteins. Within it there are conserved hypervariable regions. These can be used to identify a species or infer phylogenetic distance. 16s rRNA amplicon sequencing and analysis has been widely used for determining phylogenetic relationships in microbes (Lane et al. 1985). This is due to their universal distribution between species, constant function, and shows a slow rate of change, especially compared with most other proteins (Fox et al. 1980; Karlin, Mrázek, and Campbell 1997). Whilst this analysis is accurate, even with short amplicon reads only a few base-pairs long (100-250bp) (Z. Liu et al. 2007), they rely on databases to determine exact identity. They can however be used to place organisms that are similar enough into operational taxonomic units (OTUs). Many methods used to produce these have been shown to be unstable, meaning the clustered sequences change depending on the number of sequences that are clustered. Therefore, the final identity of an organism could be wrong due to a factor that should not have an affect (He et al. 2015).

Despite the growing size of the 16s rRNA databases, there remain other problems, including bias against certain taxa due to inefficiency of universal primers used in studies to hybridize the templates (Campanaro et al. 2018). The variable region that allows classification is flanked by highly conserved regions that are targeted by the primers, designed to be universal. However, a recent study found that approximately 10% of environmental microbial sequences could evade detection by such marker gene surveys using classical PCR methods (Eloe-Fadrosh et al. 2016). Most of these missed species belong to candidate phyla radiation, a radiation which is estimated to contain over 70 phyla, half of which seem to come from novel lineages, or call for a reorganisation of existing lineages. This is a severe limitation as not only would a large amount of information be missed, but the species missed are uncultured and less well understood, meaning it would be difficult to account for.

A selection of marker genes could be used, since there are other marker genes that have been researched and used in phylogenetic studies, such as the RNA polymerase beta subunit (RpoB) gene, which has similar properties to the 16s rRNA gene but with higher levels of divergence (Mollet, Drancourt, and Raoult 1997). Even with these problems accounted for, the information obtained would only relate to phylogenetic relationships between species. It would not inform anything relevant to their metabolism; some tools exist to infer function based on phylogenetic make-up of a community, but these have been found insufficient to accurately predict the full metabolic potential of communities (Sevigny et al. 2019). Marker gene sequencing focussing on 16s rRNA only seems useful as a validation step after other types of analysis when the goal is to understand the communities at a functional level.

Metagenomics

Given that communities are dynamic and species have recombination and horizontal gene transfer events, there's a lot that could be missed without looking at the whole sequence. One approach to address this that has become standard in metagenomics is whole genome shotgun-sequencing, which aims to capture the combined DNA sequences from the environment without the need for a culturing step. This seems much more desirable as it could capture sequences rich in insights, allowing estimates of metabolic networks in anaerobic digesters to be built (J. Zhang et al. 2017). Such approaches have had much success elsewhere, such as the metabolic activities of organisms in hadal environments (X. Zhang et al. 2018), and was used to characterize antibiotic resistance in pathogens in sewage and beach environments to allow better public health surveillance (Fresia et al. 2019). These studies demonstrate that metagenomics allow us to extract more value from microbial communities, which impact all regions of human life.

Such metagenomic approaches do have their limitations, including cost and genome quality, however solutions to overcome these are becoming increasingly more effective. These affect all stages of a metagenomic pipeline, namely: sequencing, assembly, and further analysis, and will be discussed in their sections. In metagenomics, DNA is extracted from the environment of interest, prepared and then sequenced. The reads are then assembled into contigs. The sequencing approaches can vary, but either short or long-read technologies are used, or a hybrid approach using both. During sequencing errors can occur where one base is misread as being another. With long read technologies error rate this is generally 1-5% (Amarasinghe et al. 2020) and for short read it is less than 1% (Pfeiffer et al. 2018). Using only short read sequencing however will mean that repetitive regions cannot be resolved and will result in more fragments. Using both would increase costs, and despite the falling cost of sequencing, the cost of sequencing and analysis is still a limitation of metagenomics (Schwarze et al. 2019).

Assembly

The aim of the assembly step is to join the read fragments of each individual species produced by the sequencing step into contiguous sequences, termed a contig. As much of the community's metagenome is reconstructed as possible. This is complicated by the sequences themselves, and the sequencing methods used. Approaches using graphs build on the shortest common superstring problem, where you can visualise the sequence as a directed graph with nodes as K-long substrings and edges between being the weighted overlap (how many bases overlap) between two substrings. Then, the path that visits every node whilst maximising the overlap cost is found. This approach is NP-hard (Gallant, Maier, and Astorer 1980), and in the case of repetitive genomes, would collapse the repetitive regions. Overlap-Layout-Consensus (OLC) methods use the overlap graph and then resolve the sequence further with multiple sequence alignments. De Bruijn graphs are one method of assembly that form directed multigraphs from the input sequence using substrings of length K and substrings of K-1 within those. Each K-1 substring is treated as a node and overlap between them as an edge, therefore making a substring of length K. There are various formalisations that could be used to construct a genomic assembly from the graph. One approach is called the Shortest De Bruijn Superwalk problem that aims to find the shortest walk that uses every edge of a De Bruijn graph once. Building on this and using

copy counts estimation the edges can be given multiplicities (how many times that edge must be walked) and then the superwalk that meets the constraints is used as the assembly. Both of these are NP-hard again (Medvedev et al. 2007; Kapun and Tsarev 2013). This means they are all computationally demanding and when sequencing is not perfect can form fragmented assemblies. This requires faster running, but less accurate, approximations of the solutions and assembling the unambiguous regions of graphs into separate contigs.

Additionally, genomic sequences can contain regions with repetitive elements, which can also be shared between different taxa in the metagenomic sample (Cahill et al. 2010; Khan et al. 2018). This problem is greater for short-read technologies, the most popular of which is Illumina. It is possible for the read length to be insufficient to span the repeat area of the sequence, though this is dependent upon the taxa the sequence belongs to (Cahill et al. 2010). This complicates the assembly process as the repeat region could be collapsed, resulting in shorter genomes or chimeric contigs, which are contigs that contain the sequence of more than one species or strain. The problem is made worse by the fact that the abundances of species are unknown in environmental samples, meaning it is not obvious if a low coverage is due to sequencing errors (as assumed in single genomic assembly) or because of a low abundance of the species. As such, abundances are often estimated, with various methods existing, such as the use of the read classifier Kraken followed by Bayesian statistics to calculate probable relative abundances for each species (Lu et al. 2017). Kraken itself is a programme that can classify reads by using a user-specified library of genomes against which a query is performed for every K-mer in the sequences (K by default is 31). Whatever taxa has the most matches is the result, and no matches will result in the sequence being unclassified (Wood and Salzberg 2014).

Other methods use genome databases (Truong et al. 2015; Milanese et al. 2019), and base their estimates of abundance on the coverage of clade specific genes or single-copy, protein-coding phylogenetic marker genes. However, as previously mentioned, database methods are not guaranteed to account for novel taxa, even with the increased data, though this would improve them. Therefore, the abundance estimate may be inaccurate at fine-level classification. Long-read technologies can enable longer contigs by producing reads that are long enough to span repeat regions, in some cases producing reads over 2Mb in length (Payne et al. 2019), but averages can vary, with 13 to 20 kb being reported (Tyson et al. 2018). However, they do have a higher error-rate: Nanopore reports that the modal raw read accuracy for their R9.4.1 and R10.3 pore chemistries are >97%, though results achieved in recent studies have been lower, but still >95% (Amarasinghe et al. 2020).

One advantage long-read has over short-read technologies was that they don’t use PCR, which introduces errors in abundance specific to species by favouring sequences without high or low GC content, as well as differential primer matching (Sipos et al. 2007), which could result from sequence divergence. However, there are methods to account for this (Aird et al. 2011; Krehenwinkel et al. 2017) and methods that don’t use PCR (Huptas, Scherer, and Wenning 2016), as well as a wide variety of tools developed to assemble from short reads (Ayling, Clark, and Leggett 2019). Combining both sequencing technologies produces good results, using the long-read to span problematic regions and the greater accuracy of short-reads to provide a more accurate contig. Accuracy greater than 99.8% has be produced on the human genome (Jain et al. 2018), and community resolution of the human

gut microbiota has been improved through 5x greater base pair accuracy, close to 10x fewer assembly errors, and twice the contiguity than could be achieved with either technology alone (Bertrand et al. 2019).

With cheaper sequencing technologies, the amount of data being produced is likely to keep increasing, which also has implications for not only assembly, but all steps in the process; more data will take more time to process. Not only that, but memory demand will increase, and some popular assemblers require more than 500GB of RAM for certain environmental datasets (Walt et al. 2017). This is expected by some to become a greater problem as the rate data can be produced will exceed the rate of development of memory capacity (Kleftogiannis, Kalnis, and Bajic 2013). In order to compensate for this there are various approaches, such as digital normalisation (C. T. Brown et al. 2012), which reduces redundancies in the data by removing highly covered reads until average coverage reaches a specified value. This does reduce the size of the data, and thus speeds up downstream processing, but can result in more fragmented assemblies.

Assembly methods can be made more memory efficient, such as through the use of bloom filters, which are probabilistic data structures that report whether an element is in a set or might be in a set, and has therefore been used with De Bruijn graph assemblers to reduce the memory requirements. This comes at a cost of accuracy, even with methods that aim to reduce false positives, which are responsible for false branching (branches that don't exist in the real graph) (Chikhi and Rizk 2012). With large compute node clusters available, there are assemblers that can co-assemble terabyte datasets, which has benefits over assembling and merging smaller datasets, such as lower error-rate (Hofmeyr et al. 2020). However, these resources are not yet available to all, and the error-rates still persist, meaning sequences from metagenomic studies would rarely be perfect.

Binning

Another approach to streamline assembly is the partitioning, or "binning", of reads. Binning refers to the clustering or classification of assembled or raw reads into separate groups, or "bins", that represent some meaningful taxonomic difference between them. When used on raw reads to aid assembly, certain methods recently have found binning reads improved results of downstream analysis comparing metagenomic samples (Song, Ren, and Sun 2019).

Binning usually provides more accurate results after an assembly step, due to the longer resulting sequence that can be more informative than a single raw read. Further to binning, the bins could be reassembled to create more accurate contigs or annotated and analysed further. Bins also reduce the size of the data sets that need to be analysed, often improving the quality too, or facilitating the selection of high quality groups. An example is the assembly of 238 unique and high-quality draft genomes from 396 microbial samples from the human gut, one of the most complex microbial communities currently studied (H. B. Nielsen et al. 2014). For this reason accurate binning is vital to discovering the function of groups within the microbial communities; if complete or draft prokaryote genomes result, then that allows inference of taxon-specific activity and the design of experiments to measure their role in the community *in situ*. This has allowed metagenomics to garner many insights, such as the importance of microbes in food from the metagenome associated with the soybean

product Kinema, from which a novel glutamate decarboxylase gene was validated (J. Kumar et al. 2019). Another study detected novel biocatalysts and then experimentally validated two contigs that encoded β-glucosidase and xylanase from geothermal water reservoirs (Kaushal et al. 2018). These have applications in food, textiles and biofuel industries (V. Kumar, Dangi, and Shukla 2018). It has also been used discovering new predator prey genotypes that can be used to analyse evolutionary dynamics (Sangwan et al. 2015).

Binning can be subdivided into two types at a high level; taxonomy binning and genome binning. Taxonomy binning uses similarity-based methods such as sequence similarity between whole reads using BLAST (Altschul et al. 1990), as is used in MEGAN to search databases, like NCBI-NT, and apply a naive lowest common ancestor assignment algorithm. Also used are clade-specific or single copy genes, or the look-up of specific K-mers. These are utilised in Kraken, which uses by default K-mers that are 31 nucleobases long, but can be changed to use any size (Wood and Salzberg 2014). A problem with these similarity-based taxonomy binning approaches is their reliance on databases, as this biases them towards species that have already been identified. It is an often cited figure that only 1% of microbes on the planet have been cultured. How accurate this is is debatable, but it seems that a large majority of prokaryotes remain uncultured across most environments (Steen et al. 2019).

Whilst information of the sequences of species will have also been added to databases from metagenomic studies, the major challenge is that there is a vast amount of information and so the databases will be unrepresentative of the full microbial diversity. This could result in unassigned contigs that lack a reference in the databases, or contigs that are misassigned due to the potentially incomplete nature of the sequence that would result from metagenomic assembly and result in gaps or mismatches. Whilst the assignment may represent roughly the taxonomic position of the species, there is no guarantee. This is a problem for microbial communities as they will contain novel species and are analysed largely metagenomically. As the databases are growing, and the amount of sequence data gathered in experiments increases due to improved methods and decreasing costs, the computational costs of look-up approaches comparing large genomic sequences becomes greater than alternatives (Bonham-Carter, Steele, and Bastola 2014). Methods that depend of marker genes - MetaPhlAn2 and Kraken2 - have also been found to underrepresent the complexity of communities due to high complexity of the presence of novel microbial sequence, highlighting again that such approaches are only accurate when the databases are complete (Lugli et al. 2019). Using draft genomes recovered from the metagenomic data, termed metagenomic assembled genomes (MAGs), helps alleviate this issue, however the quality and completeness of these MAGs is lower when compared to genomes generated from isolates (Van Rossum et al. 2020).

Genome binning is done independently of databases and uses features in the DNA sequence or abundance information to bin, and is much more popular than the alignment based methods currently. These methods have their advantages, particularly when applied to fragments of DNA, as they do not rely on exact alignment or databases. Rather, they use features that are assumed to be specific to taxonomic groups. There are three main approaches that utilise different features; those based on composition, those based on

abundance, and those that take a hybrid approach and utilise both composition and abundance (Sedlar, Kupkova, and Provaznik 2017).

K-mers

A useful set of features used in binning metagenomic reads is the genetic composition of a read, also called its genome signature. Genetic composition refers to the frequencies of the set of substrings present in a read when those substrings are of a fixed alphabet (for DNA this is simply the canonical nucleobases A, C, T, and G) and length. Generally substrings of lengths of 2 (dinucleotides), 3 (trinucleotides/codons) (Goldman 1993), and 4 (tetranucleotides) are used routinely for sequence analysis, though it is not uncommon to use substrings of much greater length (Tang et al. 2014). These are called K-mers or K-lets, where K is the length of the substring. K-mers are counted by moving a sliding window of size K along a sequence one base at a time (hence K-mers are overlapping) until the window reaches the end of the sequence. As such, the number of K-mers obtained from a sequence is equal to the length of the sequence minus K-1, as once the sliding window reaches the final K bases of the sequence, going any further would result in nucleotide substrings less than K. Figure 1 shows the resulting 2-mer counts for a short sequence of DNA. Three 2-mers result from a sequence 4 bases long.

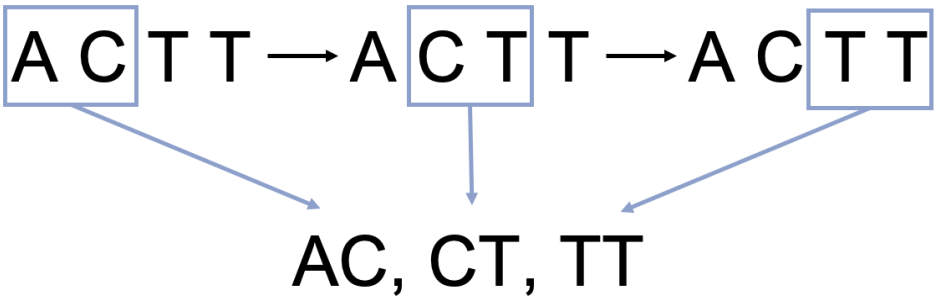

Figure 1. K-mers of length 2 extracted from the sequence ACTT. The sliding window (blue box) extracts the 2 bases and then moves along by one base, repeating this operation until it arrives at the last 2-mer. The extracted 2-mers are listed below this.

The reason these are useful for binning reads is because DNA within a species' genome has relatively stable distributions of K-mer frequencies, as showed with 2-mers using odds ratio values (Karlin and Burge 1995), which normalise the 2-mer counts by the frequency of each mononucleotide that comprises it. It has been shown to hold true for K-mers with greater values of K, even when looking at only 1000bp long regions of DNA (Zhou, Olman, and Xu 2008). This is despite local heterogeneity and from elements such as repetitive extragenic palindromes and origins of replication. These distributions also differ between species (Karlin and Ladunga 1994) and therefore carry phylogenetic signals. For example, 10 kbp fragments have been successfully characterised by using the species-specific distributions of K-mers of length 2, 3, and 4 in SOMs (Abe et al. 2003). Not only this, but species which are more closely related via phylogeny have been found to be more similar in their genomic compositions (Pride et al. 2003).The simplest genetic composition feature is GC content,

which is the percentage of G or C in a DNA sequence, or conversely AT content which is the percentage of A and T. Dinucleotides also vary between species, with CG underrepresented in some thermophilic bacteria (Karlin, Mrázek, and Campbell 1997) and thermophiles such as Sulfolobus, and GC being mainly overrepresented in γ-Proteobacterial sequences (Karlin, Campbell, Mrázek 1998). However, it is not the levels of any one genomic signature that lends them their species resolving capabilities, but a majority of them used in consort, such as shown using a Bayesian classifier and classifying sequences of 400bp in length at 85% accuracy (Sandberg et al. 2001).

As might be expected, reducing a sequence or part of a sequence to a single metric like GC content is oversimplifying the problem of phylogenetic separation and won't give accurate and reliable results. The variation in intragenomic GC content in Prokaryotes would however be less than in Eukaryotes due to high density of coding DNA resulting in a comparatively narrow heterogeneity (Sueoka 1962).

Yet, it should be noted that anaerobic microbes, such as those present in anaerobic digester communities, show greater GC heterogeneity (Bohlin et al. 2010). In one study conducted on the complete genomes of 57 Prokaryotes it was found GC content captured approximately 40% of the species specificity in coding regions and was insufficient for non-coding regions, where the GC content was on average 3.5% lower (Sandberg et al. 2003). It also reported a correlation between GC content and genomic signature, as was previously shown when the first component of a PCA analysing 36 Chaos Game Representation distance matrices was strongly correlated with AT content (Deschavanne et al. 1999). What was important was that Sandberg et al. (2003) also showed there was a large variance in the genomic signatures for any value of GC content, which meant that GC content was not sufficient for species separation and other factors had to be considered. Given what we know of GC content this is perhaps unsurprising; it could be reflecting the environment as much or more than it is an accurate predictor of taxonomy. Environmental factors (Foerstner et al. 2005) such as temperature (Hao Zheng 2010) and the complexity of the environment, and the size of the genome are correlated with GC content. It is also still an active area of research; take genome size, where reduced genome organisms may lack the machinery to repair mutations, and since most are G/C to A/T, a lower GC content might be expected. However, it's been shown that bacterial species have an excess of these mutations (Hildebrand, Meyer, and Eyre-Walker 2010; Galtier 2010), which has important implications for their evolution and genomes, and it's not known how this occurs or if it's the same in archaea. This paints GC content as an insufficient measure alone for determining species to inform binning, but an important feature nonetheless.

GC content also partially determines the frequencies of higher magnitude K-mers; a codon such as GGC could be expected to be more likely to occur in a GC rich organism through simple probability. This would also extend to K-mers and K-mers of length of K-1. Consider the sequence ATGT: its frequencies could be anticipated with knowledge of the frequencies of ATG and TGT which constitute it, and those by the dinucleotides which form them, and those by the individual nucleobases. However, this is assuming that such sequences are random, which they are not, and whilst these assumptions hold partially true, it is the deviations from the expected values by such probabilistic reasoning that would be more informative. Indeed, Markov models have been used for sequence analysis based on that

reasoning (Gelfand, Kozhukhin, and Pevzner 1992; Burge, Campbell, and Karlin 1992; Schbath, Prum, and de Turckheim 1995). It was also found that 4-mer frequencies explain more than GC content trends (Dick et al. 2009), and as such are less affected by GC content variation due to environmental factors and neutral factors such as mutations. This makes 4-mers an appealing feature for use in classification.

However, other values of K could be used, and indeed as K increases, so does the specificity of the K-mer, at the cost of an increased number of possibilities (A^k where A is alphabet size). With this comes a sparser problem space since far fewer of the possible K-mers will be observed, which would lead to an increase in computational time for algorithms processing this information (T.-J. Wu, Huang, and Li 2005). Moreover, using long K-mers is applicable when identifying known species, but since the patterns of them would be so specific at larger lengths, they wouldn't generalise well to help the classification of novel species. Generally then, the choice of K-mer size will depend upon the compute resources available, the size of the sequence being analysed, and how similar those sequences are, as sequences more closely related would require more specific K-mers to separate them. Since 4-mers have been shown to separate reads with the same or similar GC content (Karlin and Ladunga 1994), have been used to train ESOMs (emergent self-organizing maps) to separate reads as short as 500bp (Dick et al. 2009), and agree with 16s rRNA analysis when used in binning experiments of 9054 in-silico-generated fosmid-sized DNA fragments (Teeling et al. 2004), they seem a sufficient point of comparison for this study.

Machine Learning

The advances in sequencing technology, both long and short, and computing power have allowed omics technologies to enter into an era of "Big Data". Indeed, one study predicted the annual sequencing capacity for genomics in 2015 to be more than 35 Pbp (1 Pbp = 1000000 Gbp), and under the most conservative estimate would rise to exabase capacity within a decade (1 Ebp = 1000 Pbp), but stated the possibility for it to increase by four or five orders of magnitude due to the demand of human genome sequencing from governments (Stephens et al. 2015). Metagenomics has benefited from this, with the large amount of data produced in the field, thanks to the increasing popularity of next gen sequencing, and the amount of DNA in the microbiome.

This presents an ideal environment for machine learning methods to be applied. Machine learning is a subset of artificial intelligence that aims to build models purely from data, capable of making predictions and forming decision boundaries. This may seem similar to statistics, and whilst there is some disagreement between where classical statistics ends and machine learning begins, it is generally the focus of the model that differentiates. In classical statistics the model's purpose is for hypothesis testing and inference from this about how a system behaves and how confident we can be in any relationships discovered.

In machine learning the focus is on prediction through recognising complex patterns in the underlying data and finding a model that best explains these by using a learning algorithm and reducing a loss function, some of which require the data to be partitioned into data the algorithm trains on, the "training set", and then unseen data to test how well the model generalises, the "testing set". There is also a "validation" set, which can be used to tune

hyperparameters or update weights in neural networks. This is possible as the model fits on the training set, and is then evaluated against the validation set. Hyperparameters can be tuned this way by trying out the different combinations and seeing which perform best on the validation set. This is improved by a strategy called cross-validation subdivides the training set multiple times into the training and validation data, with the validation datasets not overlapping. The performance over all these subdivisions can then be used to account for random effects in the division of training and validation sets on the performance and in therefore choosing the hyperparameters. The test set is then used to give a final measure of performance on unseen data. Since the testing test cannot provide all possible unseen data, the performance of a model on it is instead an approximation of real world application performance.

Machine learning also makes fewer assumptions about the data, such as its distribution, which helps when applying it to complex problems with many variables, where drawing inferences can also become harder due to the complexity (Bzdok 2017; Bzdok, Altman, and Krzywinski 2018). However, both do share methods, such as bootstrapping and generalised linear models, so the boundary between them isn't completely clear.

There are a variety of machine learning algorithms belonging to two main groups, those that perform supervised learning and those that perform unsupervised learning, however semi-supervised learning is also used. All machine learning algorithms require "features", which represent a variable or multiple variables from the input data; often these have to be selected and engineered to ensure that the algorithm can utilise them. For instance, categorical variables such as gender would have to be encoded with dummy variables, using a different number for each unique gender. The difference between the groups is how the data is presented and what tasks they are used for.

Supervised learning algorithms require a label for each input, the input's class, such as a binary outcome, like cancerous or non-cancerous, or continuous explanatory variable such as height, for which they are trained to predict, so that they can determine a label for unseen data. It is also possible to give multiple labels, called a multiclass classification problem.

Unsupervised learning algorithms don't require labels as they are not intended to make predictions about single data points but instead are used to find any structure underlying the data, and as such are used in cluster analysis, but also dimensionality reduction.

Semi-supervised learning uses both labelled and unlabelled data to perform either a classification or regression task as in supervised learning, or clustering as in unsupervised learning, by making assumptions about the data based on the type of expected interactions between the distribution of marginal data and the posterior distribution (van Engelen and Hoos 2020).

When training models, the learning algorithms may fit the training data too well, which can result in noise and outliers present in the data being encoded in the trained model. This means performance on the training set would be perfect or close to, depending on how much of the noise has been fitted. However, this comes at the expense of a reduced performance on the validation set, since it has been fitted to the unrepresentative points, and

therefore reduces the generalisability of the model. This is called overfitting. As model complexity increases, there may often be improvements in the reduction of the loss function, however, when models get too complex they are more affected by overfitting, which is seen as an increase in performance on testing but a reduction in performance on the validation (H. Wu and Shapiro 2006). It can also be a result of too few data points during training. There exist methods that aim to modify learning algorithms to reduce the error on the validation set and fit less well to the training set, and therefore increase the generalisability and reduce the effect of overfitting (also known as high variance), which are called regularization methods.

Conversely, when a model is too simple it will underfit, as it will be unable to learn all the relationships that are underlying the data and are required for good model performance (Lever, Krzywinski, and Altman 2016a). This can require making the model more complex or cleaning the data to reduce noise so only the most meaningful points remain, however what points are noise and what points aren't may be unknown and difficult to determine. As such, defining the model could be seen as finding a balance between complexity and the data available.

Unsurprisingly, machine learning has been used in metagenomics, including binning, assigning taxonomy to sequences, and predicting gene function. Binning is a largely unsupervised process since it is grouping reads based on some similarity of features, whereas assigning a taxonomy requires the learning of labels and is therefore supervised. Indeed, CONCOCT is an algorithm that uses K-mers and coverage of multiple samples in a gaussian mixture model to cluster contigs (Alneberg et al. 2014). The previously mentioned TETRA uses a linear regression of the Z-scores for 4-mer frequencies for comparative analysis (Teeling et al. 2004). As with binning, assigning a sequence a taxonomic rank can use K-mers as features. It can also use sequence alignment, but the effectiveness of this relies heavily upon the completeness and quality of the database used. Such methods are also slower to use than trained models, which should have a linear run time, which affects their applicability to the data sets of large scale metagenomics; compositional methods can be up to 17.3 times faster (Vervier et al. 2015).

Most of the machine learning taxonomic assignment tools not based on sequence alignment use K-mers, however the value of K is variable. These are effective, yet there may be more complex features or relationships which have not been engineered yet into features for learning algorithms. Exploiting these could result in better performance.

Deep Learning

Deep learning is a subfield of machine learning, based on artificial neural networks. It has both supervised and unsupervised methods, which have the same aims as in machine learning. In recent years deep learning methods have advanced many fields, outperforming previous methods, including non-deep learning machine learning algorithms. Such areas include speech recognition, where they have outperformed the previous methods which used hidden Markov models and Gaussian mixture models (Dahl et al. 2012), and in sequence translation tasks, such as English to French (Sutskever, Vinyals, and Le 2014). One of the most recent and well-publicised works in NLP comes from OpenAI in the form of an autoregressive language model aimed to improve task-agnostic performance, GPT-3.

This model has 175 billion trainable parameters, over ten times the size of the previously largest models, and is trained on a mix of data-sets using over 1000 Petaflop/s-days of compute to do so. As a result it can perform well in a variety of NLP tasks, such as producing text indistinguishable from that produced humans (T. B. Brown et al. 2020). Deep learning represents an exciting area of active research, of which the capabilities will grow as computation becomes cheaper and datasets larger.

Another set of tasks performed by deep learning are those within computer vision, such as image classification or segmentation. Convolutional neural networks (CNNs) became the most popular deep learning approach to tackle these tasks after the success of AlexNet at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky, Sutskever, and Hinton 2017). They train on labelled examples and use their performance on a validation set to gradually reduce their loss over a series of cycles. The final performance is then measured on a test set that the network has not seen.

Neurons and the Network

Neural networks are composed of neurons. These neurons are based upon mathematical reasoning about biological neurons (McCulloch and Pitts 1990), which take one or multiple inputs, find the sum of the weighted input and bias, which is then fed into an activation function that decides whether the neuron 'fires' or not. As such, the output of a single neuron with two inputs can be written as in equation 1.

$$y_1 = f(b_1 + \sum_{j=1}^{2} w_j x_j) \tag{1}$$

Where y_1 is the output, f is the activation function, b_1 the bias, and w and x the weights and inputs respectively. Figure 2 is a diagram of such a neuron. The bias relates to how easy it is to make a certain neuron fire, whilst the weights tell of the importance of each input. The activation function exists to introduce non-linearity into the network, which allows the modelling of complex, non-linear patterns, allowing better performance than linear methods in these cases. The activation function is also what allows changes in weights and biases in a neural network to be modified via an algorithm called backpropagation. A common activation function used in CNNs is Rectified Linear Unit (ReLU), as used in AlexNet (Krizhevsky, Sutskever, and Hinton 2017), which is given by equation 2.

$$f(x) = \begin{cases} 0 \; for \; x \; < 0 \\ x \; for \; x \; \geq 0 \end{cases} \tag{2}$$

ReLU trains faster than saturating activation functions, such as the sigmoid or Tanh, since they will lead to sparser activation. Despite the positive values being linear, the fact that values less than 0 are set to 0 makes it a nonlinear function. It suffers less from the vanishing gradient problem, caused when by a gradient that is close to 0, resulting in backpropagation failing to update the weights. Since ReLU only saturates one direction, this is less of an issue. This is a problem particularly in deeper networks and other architectures such as Recurrent Neural Networks (Hochreiter 1998).

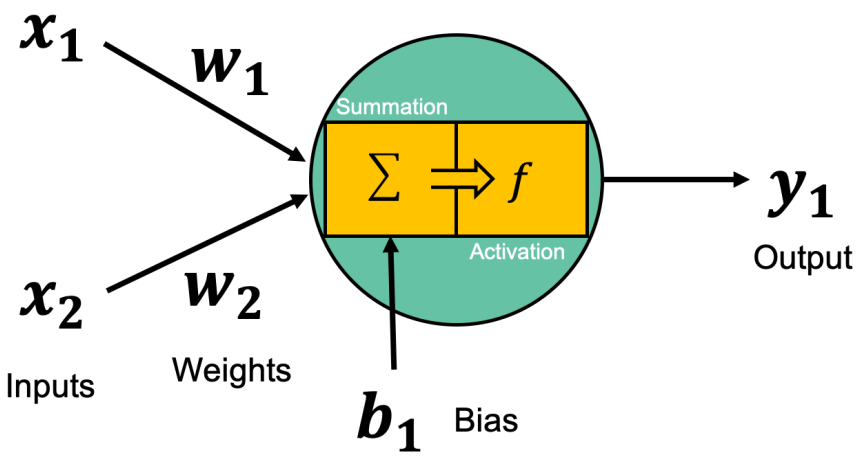

Figure 2. A single neuron as given by equation 1. A node contains the summation and activation functions. The inputs are multiplied by their respective weights, and then summed with the bias. This value then undergoes activation by the chosen function and the result from this is the output.

Single neurons are simple units, but even alone they could be used as a simple binary classifier; if the activation of the sum and the bias is above a threshold it can be classified as one class, and if it's below then it can be classified as the other. Estimating these weights and biases is the process of learning, and can be achieved through backpropagation, discussed later. Neurons are very modular, a property that allows them to be stacked to create large networks, "neural networks", which can take a large number of inputs. In doing this the inputs are received by every neuron in the first layer, and the output of those is sent forward to every neuron in the subsequent layer. Feed-forward neural networks (an example in figure 3) are one type of neural network, and are used in CNNs for supervised learning tasks.

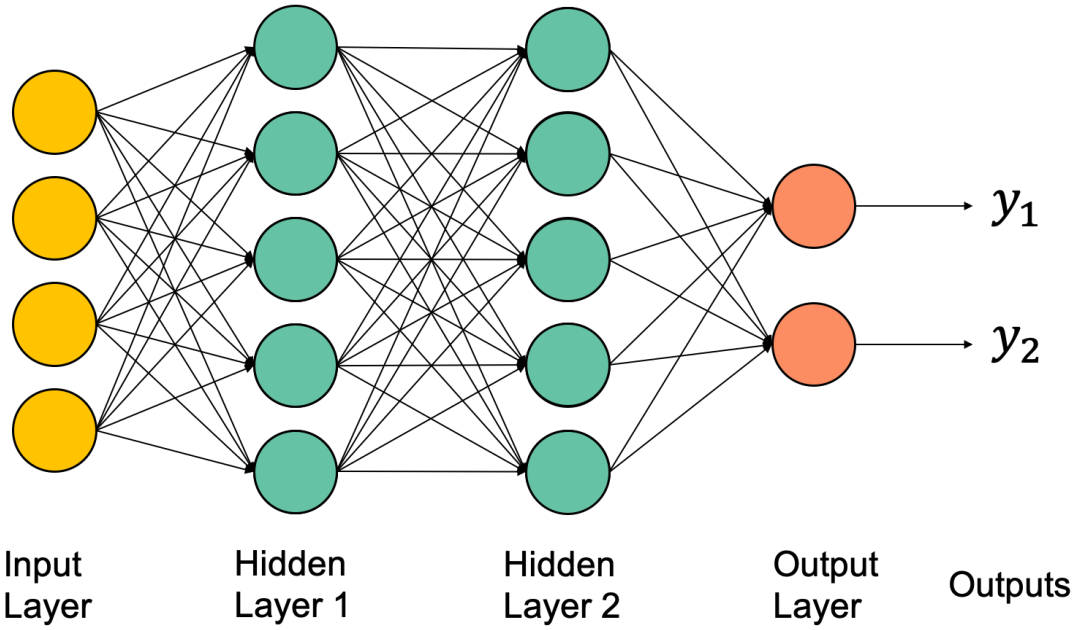

Figure 3. A fully connected feed-forward neural network. The output of each layer is connected to every node in the subsequent layer, hence fully connected. Any single node has only one output; the multiple arrows leaving a node represent this value being fed into multiple subsequent nodes, not multiple outputs. During a pass information flows from left to right until it reaches the output. There can be any number of hidden layers.

Importantly, the hidden layers after the first are using inputs from previous hidden layers, which means they are using non-linearly transformed abstractions of the data, which, when trained, is what allows a neural network to learn complex relationships between the inputs. Whilst there is no universally agreed-upon definition of when a neural network becomes deep, it depends on the task, yet in general deep learning neural networks will have multiple hidden layers between the input and output layer (Schmidhuber 2015).

Learning

For learning to happen, the weights and biases of each input/output and neuron need to be capable of being modified to improve performance, as they determine the importance of an input in the classification. As such, a loss function relating the error in model performance to reality is needed. A general workflow for training is shown in figure 4. The aim is to reach a global minimum of loss from the loss function, using the gradients to inform the descent to the point. This is part of the reason why deep learning needs so many examples to train from. Reality in the case of image classification would be the images available in the validation set. To allow this information about performance flow backwards through the network, the backpropagation algorithm makes use of the chain rule of multivariable calculus to calculate the gradients of the loss function with respect to the weights of the network, which is used to update the biases and weights. This can be repeated in training cycles termed “epochs”.This means activation functions also have to be differentiable, as well as non-linear. For non-binary classification tasks the commonly used loss function is categorical cross entropy, which is the cross-entropy loss after a softmax activation. This calculates the

differences between the probability distributions used to assign classes by the network. Weights are updated at the end of every batch, a small division of the training input, which is determined by batch size (Radiuk 2017). Once a number of samples equal to the set batch size have been processed, the weights are allowed to update.

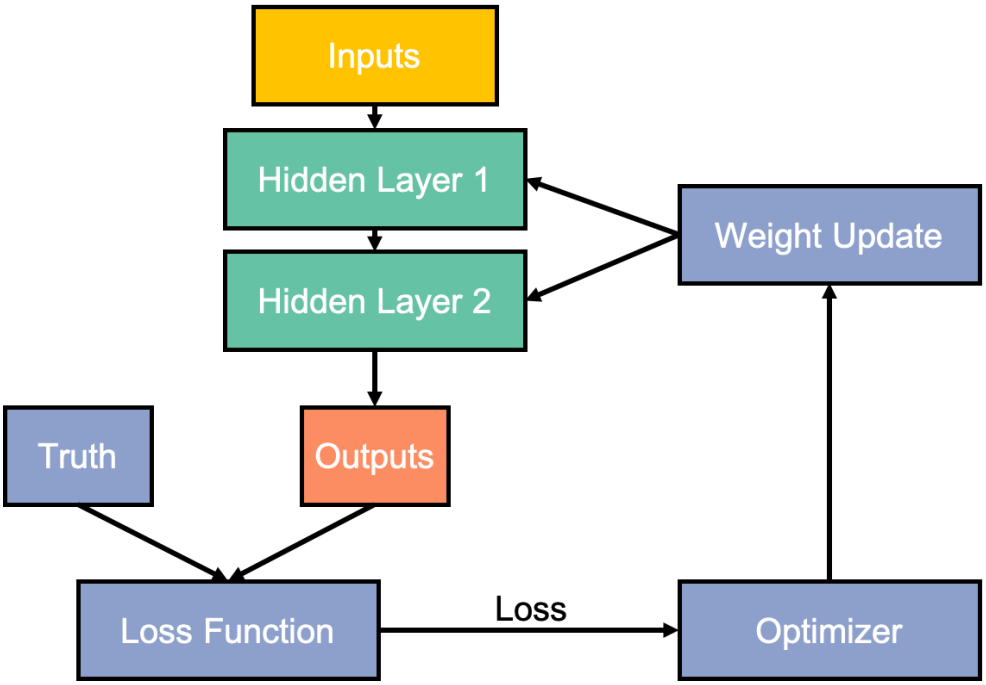

Figure 4. A simplified learning cycle of a deep neural network. Truth is either the true value of data from the training or validation set, for a classification task this would be the true class of that object. The loss function is what compares truth to the obtained outputs. The optimizer uses the loss to calculate how much to update the weights of the hidden layers. The cycle can be run infinitely, though generally training is stopped when validation loss rises and training loss continues to fall, indicating overfitting.

An important component of the training process of a neural network is the optimizer. The optimizer affects how quickly the neural network will learn and also how well, and does so using gradients calculated from the loss function with the backpropagation algorithm. If we think of the problem of learning as a constrained optimization problem, we want to reach the global minima, the point at which the loss will be lowest, and performance on the training and validation sets the best, according to our loss function. One way of doing this is to use gradient descent, which is a basic algorithm, but isn't suited to large datasets as it can get caught in local minima and takes a long time to converge.

However, it does form the basis of reasoning for other methods. As such, a frequently used optimised is Adaptive Moment Estimation (Adam) (Kingma and Ba 2014), which uses a variable learning rate. A learning rate is the rate at which weights in the neural network are updated, and as such it is one of the most important hyperparameters. When the learning rate is too large, the descent of the gradient may miss the minima and as a result increase the error, whereas a learning rate that is too slow will take a lot of time and resources to

train, and may get stuck in local minima. The learning rates in Adam are updated on a per weight basis, by using exponential moving averages of the gradient and squared gradient as estimates of the mean and uncentered variance of that gradient and corrects for bias, which bounds the weight update by the learning rate and makes choosing the scale of the learning rate easier. Two parameters, called betas, determined the rate of decay for these values. This results in faster convergence and oftentimes better performance (figure 5).

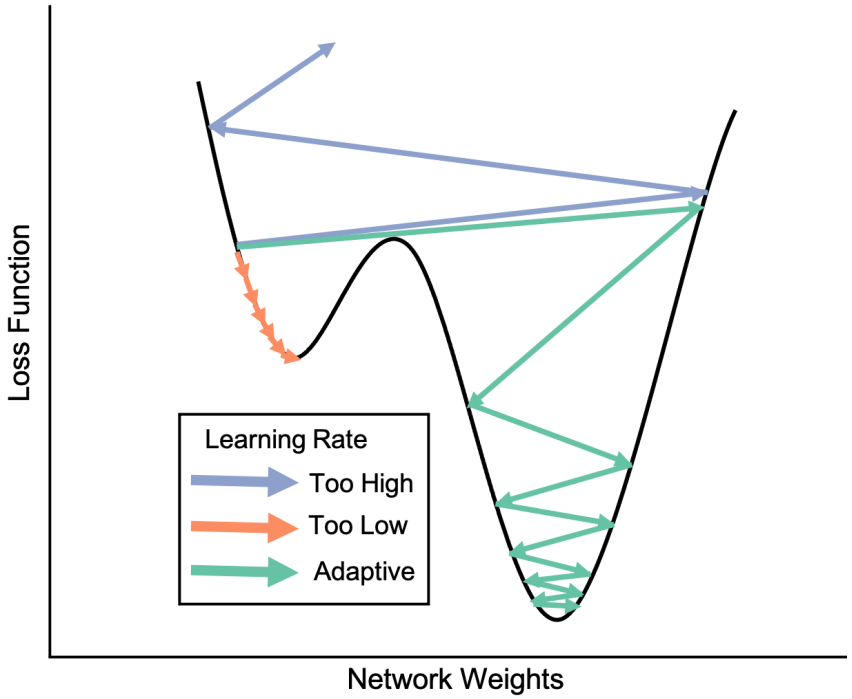

Figure 5. The effect of learning rate on convergence. The black line represents the problem space with the lowest valley representing the global minima, and the ends of arrows the position in the problem space after a weight update. When the learning rate is too high (blue arrow) the weight updates cause a divergence from the minima. The orange arrow shows what happens when a learning rate is too low: it gets stuck in a local minima and takes a large number of updates to do so. The green arrow represents an adaptive learning rate, such as those in Adam, where local minima are avoided and fewers steps are needed due to initially large steps.

Since changing the weights of the layers will also affect the inputs of subsequent layers, training is complicated further, since the distribution of input for each layer is likely to change. To tackle this 'internal covariate shift' is a method called batch normalisation, which speeds up training time (Ioffe and Szegedy 2015). It does this by applying normalization for each training mini-batch, often performed after convolution layers but before activation with a nonlinear function. The result is a smoothening of the optimization landscape, and so gradients behaviour is more stable. This allows training times to be reduced since gradients become more predictive (Santurkar et al. 2019). Two important parameters of this are epsilon and momentum. Momentum affects the rate at which the estimates of mean and variance are updated and epsilon is a constant in the calculation which improves stability.

There are also learnable parameters, γ and β, which are vectors of size equal to input. The calculation which uses all these is shown in equation 3.

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \epsilon}} (\gamma + \beta) \quad (3)$$

Overfitting

As mentioned, neural networks are large and prone to overfitting, but there are ways to combat this. One way is to train for fewer epochs, since the network is fit gradually to the training data by weight updates once in each epoch, it is possible to stop training before the network is fit too closely to the data and performance on the validation set decreases (i.e before overfitting). This is usually done by monitoring the performance of the network on the validation set, and stopping if performance doesn't improve or grows worse, within a timeframe of epochs.

Another way is to increase the amount of training data so that it takes longer to overfit and has more time to learn the relationships between inputs that reduce loss further. One way to do this is to gather more data. Another is called augmentation, which takes data already in the dataset and changes it in some way that has relevance, for the case of images this may be reflecting the image horizontally so that other orientations can be learnt. This works because the class of the image, say for instance, cat, does not change if its orientation is different. This isn't applicable to all cases, and so other augmentations may have to be used.

Regularization methods are changes to the model aimed at reducing overfitting, such as dropout (Hinton et al. 2012), where every hidden neuron has a random chance, defined by a set probability, to be omitted from the network. This means it is temporarily removed from the network, as are its connections to other neurons. The rationale for this is that neurons can fix the mistakes made in others, that could result in co-adaptations developing that then don't generalise well to unseen data and thus have caused the model to overfit. By randomly omitting neurons, neurons in the network can't develop these co-adaptations as each epoch it is not certain what neurons will be present. Weight decay is also a regularization method (Krogh and Hertz 1991). The aim of weight decay is to constrain how fast weights grow in a network, to make it less complex and more generalizable. It adds a cost function that penalizes large weights, so the only weights that will grow large are those where it is important for learning. When used with Adam, weight decay needs to be decoupled due to its use of momentums and adaptive nature (Loshchilov and Hutter 2017).

CNNs

In CNNs the inputs are features extracted arrays which are then used for learning the correct classification by layers similar to the fully connected network described above, making them a feed-forward neural network. Images are a common input, a 2D array, however they can be used on any dimensional array such as 3D arrays, an example of which would be video, or 1D arrays such as sequences. Such networks have been useful in biology, such as in live microscopy, where they have led to detection of live cells with 96% accuracy whilst being 100 times faster than previous methods (Linsley et al. 2020). In bioinformatics they have been widely used, such as for gene expression classification from histone modification data (Singh et al. 2016), predicting splice junctions on pre-mRNA transcripts (Jaganathan et al.

2019), and the prediction of gene expression using proximal and distal promoter and enhancer-promoter interactions (W. Zeng, Wang, and Jiang 2019). They have also been used in metagenomics as part of a methodology to predict genes in ORFs (Al-Ajlan and El Allali 2019) and to improve the classification of fungi (Vu, Groenewald, and Verkley 2020) and bacteria (Liang et al. 2020).

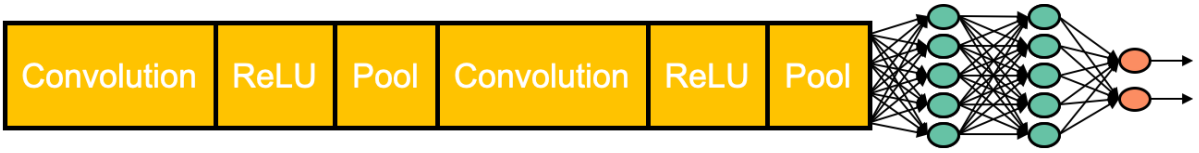

Figure 6. A general diagram of a CNN: convolutional layers are stacked with an activation function (ReLU) and a pooling layer, before entering a fully connected layer or layers that give the output classification. Only the convolution layers and fully connected layers receive weight updates.

The layers in CNNs will vary depending on the model, yet they have a core set of layers common to most. They generally take the form of a series of repeating of convolution, ReLU (or other activation function), and pooling operations, before the fully connected layer or layers, which provide the outputs (figure 6). Since CNNs are used for cases of supervised learning, these outputs are predictions.

In CNNs, the convolution operation is a linear operation that multiplies the inputs by values in a kernel (Goodfellow, Bengio, and Courville 2016). The values of the kernel are what a CNN aims to learn. The kernels are often initialised randomly and then adjusted in a similar manner to the weights in a fully connected feed-forward neural network. That is to say, they are learnt using backpropagation to minimise a loss function. When applied to 2D data, such as images, the kernels are often 2D as well. The kernels move across the image a number of pixels at a time, called a stride. At each step the kernel computes the dot product of the input values and the weights in the kernel (see figure 7A). Since the kernels are usually of size 3x3 or greater, multiple pixels in the input will provide a single output. This is done until the whole image is traversed. The result is a smaller matrix than the original image, which are referred to as feature maps. This is done with multiple kernels. A kernel will have the same weights at every point in the image that it is used. Since weights contributing to the output feature maps are those in the kernel, each pixel will have the same weights as those that occupied the same position in the kernel. This is called weight sharing (Rumelhart, Hinton, and Williams 1986). It means that a kernel will learn a single set of weights and increases the efficiency of the network. As such the kernel will be able to detect the feature it has trained, regardless of where it is located in the image, making this design translation invariant. The use of kernels also means the network has sparse weights, when the kernel is smaller than the image.

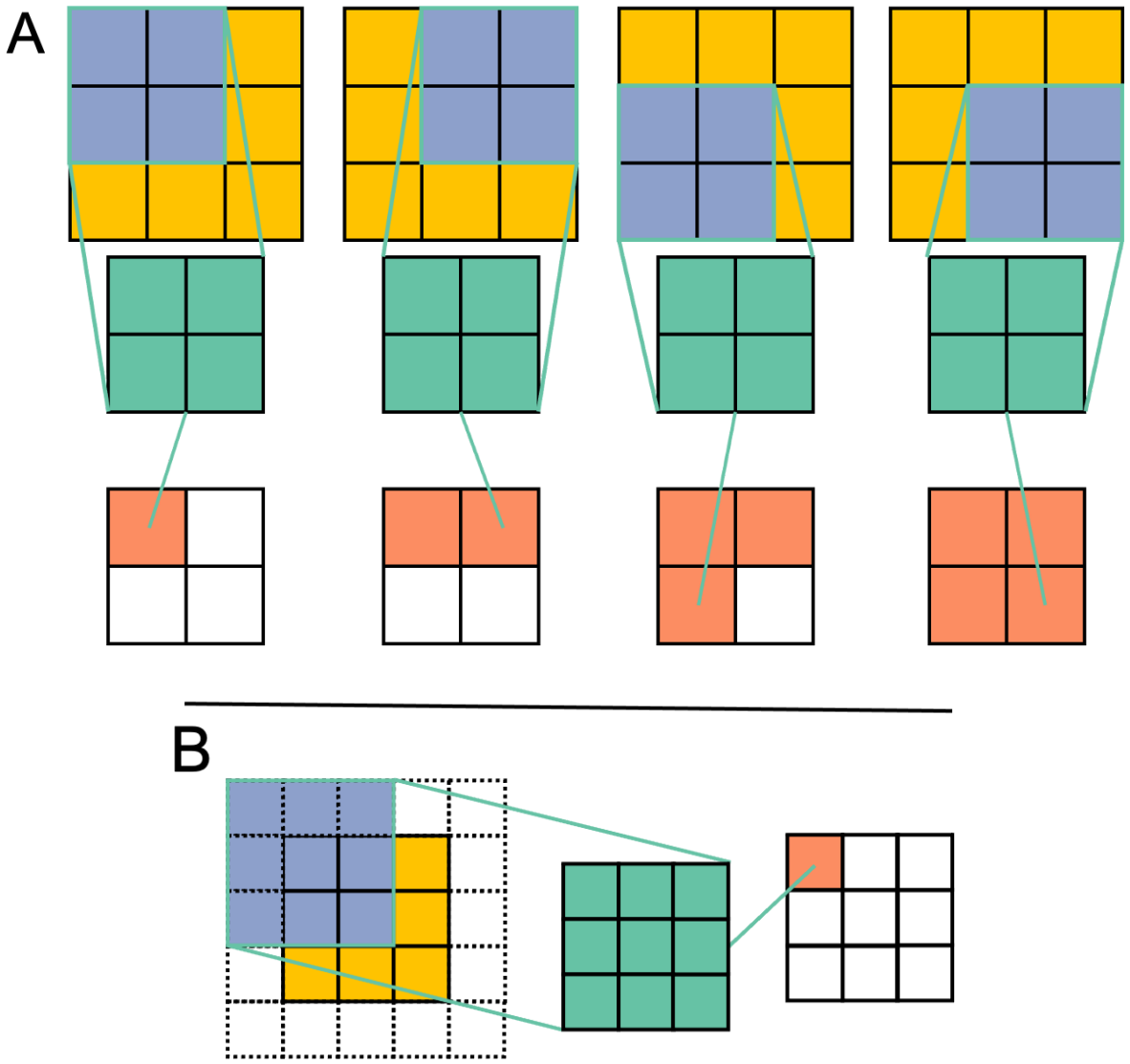

Figure 7. (A) A convolution of a 3x3 matrix without padding. In blue is the receptive field of the input image. The dot product of the receptive field and the kernel (green), to produce a single value output (orange). The receptive field moves by a step of 1 and the dot product is repeated at the remaining 3 possible positions, using the same kernel for each receptive field. The result is a 2x2 feature map. If a similar convolution was applied to the output, the effective receptive field would be 3x3 even whilst only looking at 2x2 pixels since those 2x2 summarise the 3x3. (B) Shows an example of how padding allows retention of dimensions (a 3x3 input gives a 3x3 output) by adding extra values, often 0, before applying the kernel.

This is because every pixel is not connected to every output. Indeed, only the pixels present in the kernel will link to their output, so for a 3x3 image using a kernel size of 2x2, each output will have 4 weights instead of 9. When scaling this up to real images, which have upwards of thousands of pixels, there is an even clearer improvement in efficiency. These two features of the weights in neural networks, that they are shared and sparse, mean that not only is learning computationally feasible, but that a range of simple features can be

learnt by the different kernels and detected at multiple locations in the input image. Not only this, but the receptive field (those inputs that affect an output) grows the deeper the convolution layers are stacked, since they will take the outputs of the previous convolutional layer as inputs, and will therefore be indirectly connected to more of the original input pixels. This means that the features learnt by a layer will become more complex, built from the simpler features (Kriegeskorte 2015; Geirhos et al. 2018). In such a fashion deep outputs can be connected to all or large portions of the input image indirectly and learn increasingly complex features. After the dot product is found, bias, which must also be learnt, is added to each output feature map to reflect the importance of the feature for classification of the input (M. A. Nielsen 2015).

Often, shrinking the input may be undesirable, so zero-padding surrounds the matrix with a border of zeroes, the thickness of which depends on the filter size used. Padding is often employed to allow for more layers to be employed without shrinking the feature map (figure 7B) (Yamashita et al. 2018).

The ReLU function is then used to set negative values in the output feature maps to zero, which speeds up computation since there are then no operations to perform on zero. Additionally it introduces non-linearity. Since convolutions are linear operations it would be possible to collapse all chained convolutions into a single linear operation. By introducing non-linearity this cannot happen, and allows the multi-resolution properties of the CNN (-C. Jay Kuo 2016).

The pooling operation is designed to reduce the size of the input via a function, in order to extract important and representative features, whilst also achieving invariance to translations and removing noise, and make the model more robust against overfitting (Gholamalinezhad and Khosravi 2020). It does this by summarising the features in a small area, or neighbourhood. Common pooling methods include max-pooling (Ranzato, Boureau, and Cun 2008) and mean-pooling (Lecun et al. 1998), with max being better suited to sparse features and giving reduced variability (Boureau, Ponce, and LeCun 2010). Similar to how kernels traversed over the input image, values in a striding area of the output feature maps, a 2x2 area most commonly, are either averaged to find the mean or the maximum value is taken, for mean-pooling and max-pooling respectively. This is effectively downsampling the features to make the exact position matter less, since a small translation in the input will not affect the output of most pooling layers. This also allows the network to become invariant to certain transformations (Goodfellow, Bengio, and Courville 2016).

Since CNNs remove the need for feature engineering, which often requires an expert and thorough understanding of the problem, they can be applied to problems where such knowledge does not yet exist or the systems of study are not fully understood. There is therefore great potential for applying such networks to genomic data, as previously mentioned.

Image Representations of DNA

DNA can be represented as a string of letters. There are several approaches to turn these into a 2D image to train a CNN. Chaos game representation (CGR) is a method of converting a sequence into an image (Jeffrey 1990). It derives its name from the chaos

game, which can be used to generate fractal attractors, such as the Sierpinski triangle, if three points are used. However, CGR uses four points in a square, with each point representing a base, A, C, T or G. For a sequence of DNA a point is drawn halfway between the center and the corner representing the current base. After this a point is drawn halfway between that base and the subsequent base, repeating until the end of the sequence (figure 8). This will create a series of points that give a fractal pattern, due to the non-random nature of the DNA sequence. The exact pattern will be unique to the sequence used and can be reconstructed from it (figure 8).

Importantly though, due to the fractal nature of this representation, K-mers are represented using CGR (Hill, Schisler, and Singh 1992). Considering the most recent K bases to reach a point will give the K-mer at that point in the sequence. It is also true that the most recent K bases will fall inside the area covered by the most recent K-1 bases. See figure 10, where all dinucleotides ending in C fall inside the area covered by C. However, to have each point map to a pixel and retain a necessary square aspect ratio the image would use 4^k total pixels, where K is K-mer length. As such this is infeasible for high values of K long before coming close to reaching the lengths of DNA sequences. Therefore at manageable resolutions images produced by CGR will be a representation of shorter K-mers.

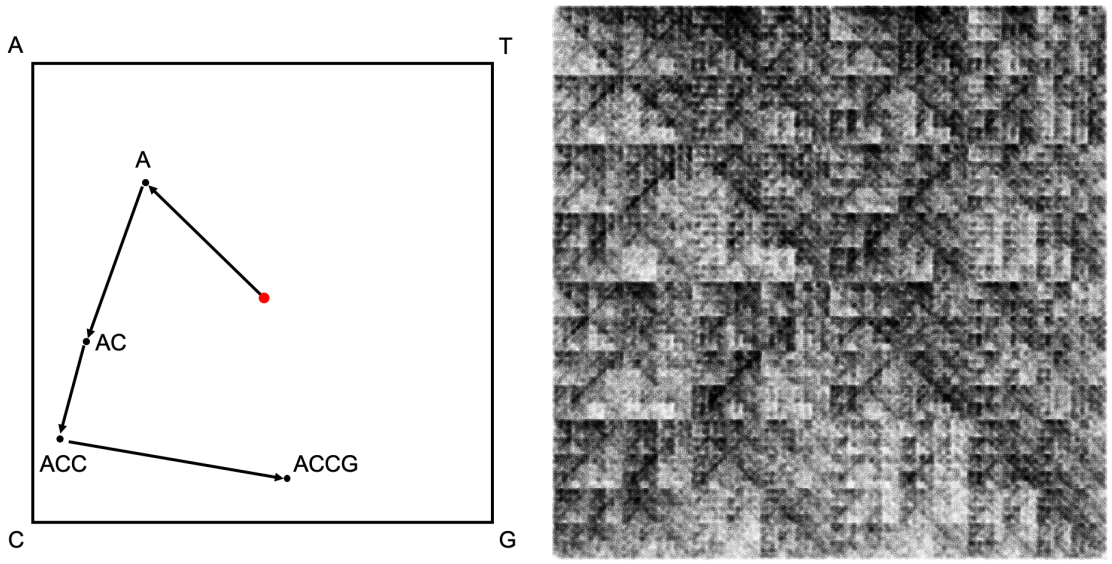

Figure 8. (Left) The chaos game with the 4-mer ACCG. The red dot indicates the center. From here a point is placed halfway between the center and the A corner. This is repeated, placing a new dot halfway between the current point and the corner of the next base. The arrows are to show order of plotting of the dots and are not part of the actual representation. (Right) A fractal pattern resulting from the chaos game with the sequence of an entire species. The corner bases are the same as denoted in the left image. Such an image is an example of chaos game representation (CGR).

Since the resolution limit means CGR creates images which are K-mer summarisations of the sequence, it may be that some important details are lost. Another way of converting a DNA sequence to an image is using space-filling curves, which retain all the bases of the

raw sequence. They do this by mapping each element of the sequence to a pixel in the resulting 2D image (figure 9). This has been done using a variety of space-filling curves, including Hilbert curves and Reshape curves, to predict determinants of chromatin structure (Yin et al. 2018). By using space-filling curves the proximity for the most proximal elements is retained, whilst distal elements are brought into close proximity. This covers short and longer range interactions, and preserves their sequence. CGR does not do this beyond the K bases at each pixel value; the sequence cannot be reconstructed from a CGR image, but it could be from a space-filling curve image. Reshape curves are easier to implement than Hilbert curves, yet provide comparable or better accuracy than other alternative curves (Yin et al. 2018). Not only does this mean that by using Reshape, meaningful long range patterns might be discovered, but that those patterns which are found could be used on other DNA sequences, not limiting the analysis to interactions between K-mers of a set length.

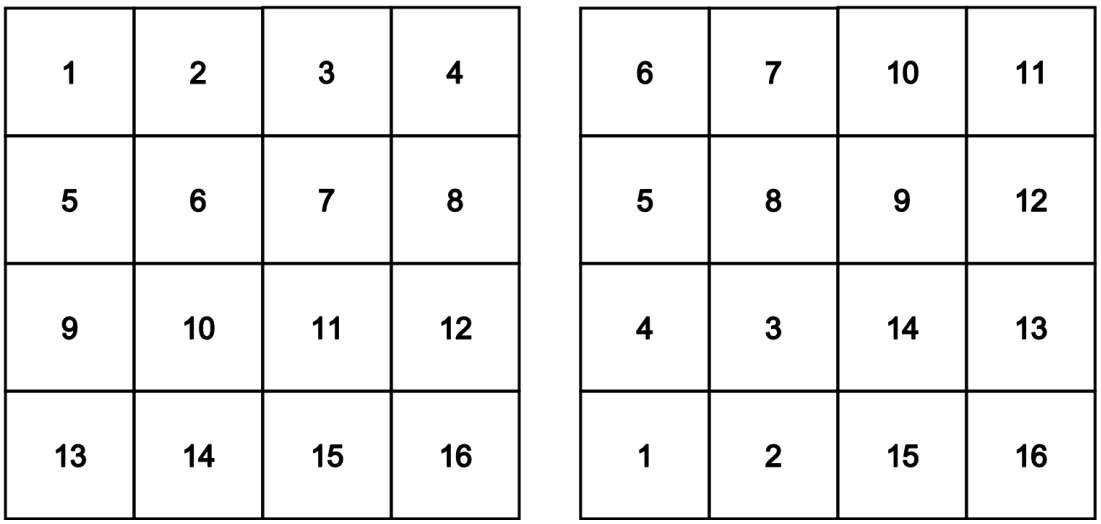

Figure 9. Space-filling curves structure a sequence into a 2D image. Reshape (Left) and Hilbert (right) curves of the same sequence, 1-16. Both retain the order of the sequence, but bring distal elements closer.

Since histones are commonly present in archaea (Henneman et al. 2018), using Reshape could also reflect the importance of spatial configuration which makes interactions between distal regions important. Currently the only lineage not possessing orthologs of H3 and H4, histones important in chromatin assembly, is thought to be Crenarchaota, but even they possess proteins thought to compact DNA by altering its configuration (Driessen et al. 2013), so here too distal elements are likely important. A representation that reflects these differences in distally related elements between lineages, such as might therefore improve classification.

This study aims to leverage CNNs on DNA in an archaeal classification task. By representing DNA sequence as an image and using CNNs it is hoped we can leverage the work already performed on improving model architecture and training techniques to improve classification performance over simpler machine learning models. Using a CNN will allow feature importance to be learnt in a supervised manner. The two image conversion formats

used will also be compared. Reshape encourages the learning of distal interactions, whereas CGR summarises the sequence as K-mers (in this case 7-mers so that resolution is the same). CGR could therefore miss out on important features of the sequence, or it could reduce noise and improve performance. These two approaches are compared to the best performing of a consort of machine learning models that use 4-mers and 7-mers as features. Limitations of the approach used and future work are also discussed.

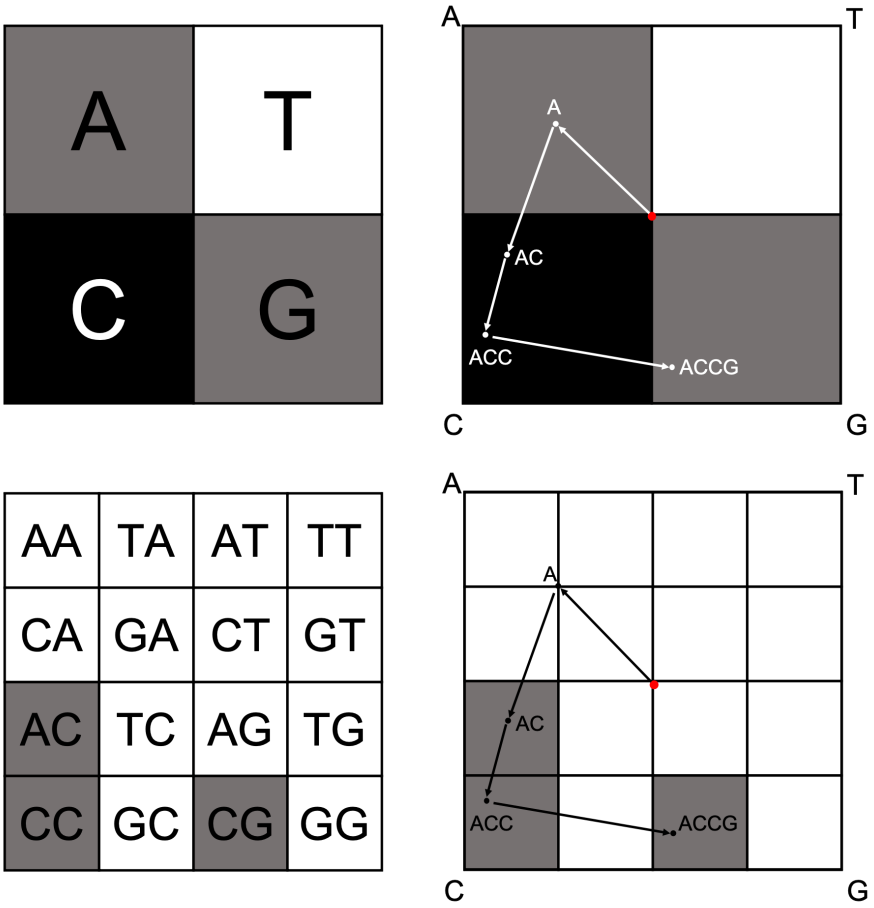

Figure 10. (Top Left) Shows the CGR representation of the sequence ACCG at the resolution of nucleotides, with the nucleotide the square represents overlaid. The pixel intensity for each square corresponds to the number of that nucleotide in the sequence. (Top Right) Shows how to arrive at the top left CGR, with the sequence and chaos game steps overlaid. (Bottom Left) The CGR representation of the same sequence but at the resolution of dinucleotides. The beginnings of the fractal nature can be seen, with each nucleotide square being divided up and the first base having the same relative position inside that square as in the lower resolution grid. For example, AC occupies the top left of the C quadrant, just as A occupies the top left of the nucleotide resolution grid. The pattern is true for all positions and all resolutions. (Bottom Right) Show the chaos game overlaid on the image. Since A is in the middle of the four squares in the top left quadrant it does not count towards any of the dinucleotide counts, so the remaining dinucleotides are of equal intensity since there is only one of each.

Methods

DNA sequences gathered from Genbank or randomly generated, before being converted into images or tables of K-mers, then split into training and test sets. The models were trained on the former and evaluated using the latter. Figure 11 shows a schematic overview of this, and the elements are discussed in further detail in the relevant sections.

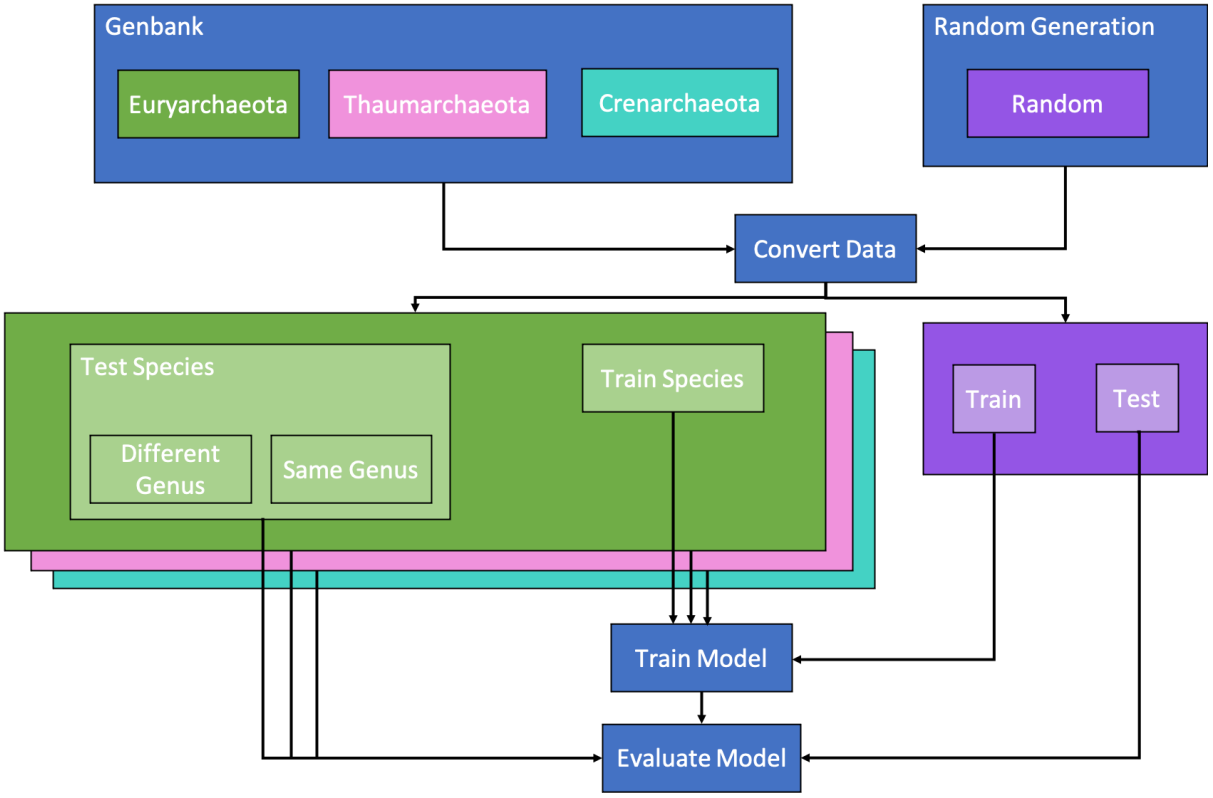

Figure 11. All archaeal sequences were retrieved from Genbank and Random sequences were randomly generated. The sequences were converted into tables of 4-mers, 7-mers, or were converted into images using CGR or Reshape. These were then split into training and testing sets, with the test sets of the archaea containing species of the same and different genera. Training used the species and random sequences in the training datasets, and performance was evaluated on the testing sets.

Datasets

The training data was constructed by randomly selecting the whole genomes of species belonging to Euryarchaeota, Thaumarchaeota, or Crenarchaeota from GenBank. These Orders represent the most well studied archaea with high quality genomes for a variety of species within. In addition, the random set in the training data was made by generating random sequences of bases, with equal probability to pick each base. The training data was split further into training and validation datasets for the CNN models. This was done using a random allotment into training or validation sets. The validation set is what allows the CNN to learn; it provides the ground truth against which performance can be measured and loss calculated, which informs how the weights are updated. Therefore, both are involved in the training process.

The validation set in the context of the other models would serve a slightly different role; it would be used to determine optimal hyperparameters, after which all the data would be used to train on using these. However, hyperparameter tuning was not performed due to time constraints, so the training data was not split before it was used to train the models. This is because multiple models were, many of which would improve little with hyperparameter tuning.

The test set was composed of three main groups: one consisted of species from the same genera as those in the training set, and the second consisted of species of Euryarchaeota, Crenarchaeota, and Thaumarchaeota from different genera to those in the training set. This is to ensure the model is not overfitting to features specific to the genera of the training set, and to see whether the rules learnt by the model allow it to identify closely and more distantly related, unseen species.

The final group in the test set were randomly generated sequences, generated using a different seed to those in the training set, resulting in a different set of random sequences. The random data was used to act as a control as a set of sequences that are obviously different from the others. Table 1 shows the split of the training set.

Table 1. The split for each class, the number of species used and how many sections 16384 bp long that represents.

Class	Partition	Number of species used	Number of 16384 bp sequences
Euryarchaeota	Train	31	9864
Euryarchaeota	Test (Same Genus)	24	3175
Euryarchaeota	Test (Different Genus)	27	4511
Crenarchaeota	Train	21	4808
Crenarchaeota	Test (Same Genus)	21	2604
Crenarchaeota	Test (Different Genus)	18	2968
Thaumarchaeota	Train	10	2451
Thaumarchaeota	Test (Same Genus)	3	508
Thaumarchaeota	Test (Different Genus)	2	704
Random	Train	N/A	999
Random	Test	N/A	3999

Metrics

For a majority of the experiments, F1-score was used since the classes are imbalanced in size. It is the harmonic mean of precision and recall. Precision is the ratio of correctly classified positive results of a class (true positives, tp) to total results classified as that class, consisting of true positives, and the results which were incorrectly predicted to belong to the specific class in question (false positives, fp). Recall is the ratio of true positives of a class to all data points in that class: the true positives and all the examples incorrectly classified as other classes (false negatives, fn). Since F1-score is an aggregate measurement it balances the effect of recall and precision. This is important as recall and precision consider false negatives and false positives respectively, which are important to understand in class imbalanced classification (Lever, Krzywinski, and Altman 2016b). Using a single metric also simplifies comparison. F1-score is given by equation 4.

$$F1 = \frac{2}{precision^{-1} + recall^{-1}} = \frac{tp}{tp + \frac{1}{2}(fn + fp)} \quad (4)$$

Furthermore, as some classes are rarer within the training data, the macro-F1 score was used to assess model performance, which takes the mean average of each class's F1 score with respect to the number of classes, not the number examples. This emphasises model performance on rare classes, removing the effect of class size, as each class should be equally valued. However, where only one class is considered accuracy is used instead.

A multiclass variant of Matthews Correlation Coefficient (Jurman, Riccadonna, and Furlanello 2012) was used to compare the errors of replicate CNN models. It is a metric that ranges between -1 and 1, where -1 represents perfect inverse prediction, 0 represents random prediction, and 1 represents perfect prediction.

Features and Training

To convert the sequence data into a format amenable for machine learning models to train on, features needed to be extracted. K-mers of length four (4-mers) were chosen for this due to their taxonomic resolution capabilities at high levels, such as phyla. These were counted in sections of the sequences 16384 bases long. The count of each 4-mer in these 16384 base regions is treated as a feature to train upon, resulting in 256 features. If a sequence was shorter than 16384 bases, it was excluded from the dataset. Each row was then scaled by the total number of canonical 4-mers counted. As some sequences had unresolved bases, marked with an N, they would not be included, so not all rows totalled to 16381 4-mers.

Models were trained using scikit-learn version 0.23.1 and Python version 3.8.3. Models were trained on the entire training dataset and final performance assessed using performance on the test sets. As mentioned, the performance metric used was F1-Score due to an imbalance between classes.

Feature selection was performed with L1 regularization of a logistic regression model (a logistic LASSO regression). 2, 5, 10, and 25 features were selected using L1 regularization. Feature selection was also performed by removing variables that had a pearson correlation

coefficient greater than 0.8 in the original 256 features as these would most likely be redundant when making a prediction. This yielded 121 features.

Using LASSO regression to perform a continuous subset selection of features is possible due to the nature of the L1 penalty it applies to the logistic regression. It is capable of reducing the coefficients of features to 0. This is a type of regularization. As the penalty is increased, complexity is punished more, so the coefficients of features are reduced. Features which are less important as predictors will have their coefficients reduced to 0 first. Once this happens, a feature can be ignored as it doesn't impact the classification result. The penalty can be increased until the target number of features is met.

A variety of models were then trained on 2, 5, 10, 25, 121 and 256 (all) 4-mers, and the top 5 highest F1-Scores taken. The top 5 performing models differed between the number of features used. All were normalised after feature selection to prevent information leakage. Feature values are a ratio of all selected 4-mers, not the relative frequency when considering all 256. The models used were scikitlearn implementations of following classifiers: Ada Boost, Bagging, ExtraTrees, Gradient Boosting, Random Forest, Gaussian Process, Logistic Regression, Passive Aggressive, Ride, SGD, Preceptron, Bernoulli and Gaussian Naive Bayes, KNN, Linear SVM, Linear Discriminant Analysis, and Quadratic Analysis.

The same sequences used in the training and test sets were converted into images to train the CNN upon. This was done by using a Reshape curve and ordinally encoding each base to an equally spaced pixel value between 0 and 255. Unlike the features used by the other machine learning models, the data used to train the CNN kept the sequences with unresolved bases. Thus there were five different possible values a pixel could take. These sequences of numbers were then shaped into a 128 by 128 square, reading from left to right, and top to bottom. A simple Reshape curve is shown in figure 10. A Reshape curve was chosen since it maintains the sequential nature of the DNA whilst also bringing distal regions close together. It also allowed for simple image augmentation to increase the number of training examples. This was performed by shifting the start of the image 8192 bases forward, so what was the halfway point of the image becomes the starting point of the augmented image, and the end is the halfway point of the image subsequent to the original image (figure 12).

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24

Figure 12. (Left) How a Reshape curve would position the elements of a sequence. (Right)

Shows how the transforming of images was performed to augment the datasets, beginning from the ninth element, the start of the latter half of the previous image.

To train the CNN on the Reshape curve data, the data was split at a 80:20 training:validation ratio. Final performance was assessed against complete test sets. Again, F1-score was used as the performance metric. The model architecture used was based on that of VGG19 (Simonyan and Zisserman 2015) and was implemented in fastai. A batch size of 10, weight decay of 0.0001, were used when training the CNN for 10 epochs on a fit one cycle policy (Smith 2018). The Adam optimiser was used with betas (0.9, 0.99), and categorical cross entropy was used as the loss function. The model uses blocks of 2D convolutions followed by batch normalisation and then ReLU activation. After repetition of these blocks there is max pooling (the number of repetitions before pooling varies). Max pooling uses a kernel size and stride of 2 with no padding. As such, after every max pooling the width and height dimensions are halved and the depth left unaffected, except for the final max pooling layer, which occurs alongside average pooling. After the final convolution layer, average pooling to create a 7x7 size is performed, after which both average and max pooling is performed. The result of this is flattened into a vector and goes through two-fully connected layers before output. Convolutional layers used a kernel size of 3x3 and a stride of 1, with padding (1, 1). Batch normalisation uses an epsilon of 1e-5 and a momentum of 0.1. Further model details can be found in table 2.

Generation of the CGR images was done at the resolution of 7-mers, since this would result in a 128x128 resolution image, as used in the Reshape curves. The images that were shifted along half the length of the image to augment the dataset were not used to generate CGR images, as sequence continuity is not important in such a representation. They could therefore be omitted to improve training times. With this exception, the same 16384 base segments of sequences were used to generate the images. Training on the CGR data was performed using the same architecture, but only for 6 epochs instead of 10 to avoid overfitting.

To generate the sequences of random 4-mers used to perturb the CNN, random sequences were taken from each of the four classes from the training data to form a subset. These 4-mers were counted to provide the 4-mer frequencies that the randomly generated sequences had to meet. Sequence generation was done using a Python implementation of uShuffle (Jiang et al. 2008). uShuffle works by using the Euler algorithm (Kandel et al. 1996) with an updated algorithm for the generation of uniform random rooted directed graphs (arborescences) (Propp and Wilson 1998). This is permuted, and then a Eulerian walk generates the sequence by outputting the sequence along the walk by using each edge. This produces a uniformly random sequence where K-mer counts, such as 4-mers in this instance, can be preserved. These sequences were then converted into images using Reshape curves as previously described. These were then added as a fifth class, meaning the dimensions of the final fully connected layer in the model were changed to (512, 5).

t-SNE was performed to embed the 4-mers used to train the machine learning models into 2D space: using perplexity 30, early exaggeration 12, learning rate 200, with a gradient

calculation algorithm using the Barnes-Hut approximation over 1000 iterations with scikit-learn.

Table 2. The layers of the CNN architecture, (input channels, output channels), and the number of trainable parameters for each layer. Batch normalisation layers and the ReLU that follows each convolution layer is omitted for brevity, but the second value in parameters for each convolutional layer is the number for the associated batch normalization layer.

Input Image (128x128x3)	**Parameters**
Conv (3, 64)	84, 12
Conv (64, 64)	1792, 256
Max pooling	0
Conv (64, 128)	73856, 512
Conv (128, 128)	147584, 512
Max pooling	0
Conv (128, 256)	295168, 1024
Conv (256, 256)	590080, 1024
Conv (256, 256)	590080, 1024
Conv (256, 256)	590080, 1024
Max pooling	0
Conv (256, 512)	1180160, 2048
Conv (512, 512)	2359808, 2048
Conv (512, 512)	2359808, 2048
Conv (512, 512)	2359808, 2048
Max pooling	0
Conv (512, 512)	2359808, 2048
Conv (512, 512)	2359808, 2048
Conv (512, 512)	2359808, 2048
Conv (512, 512)	2359808, 2048
Max pooling	0
Average pooling and max pooling	0
Flatten	0
Batch normalisation (1024)	4096
Drop out (p = 0.25)	0
Fully connected (1024, 512)	524800
ReLU then batch normalisation (512)	2048
Drop out (p = 0.5)	0
Fully connected (512, 4)	2052

Adding in species

4 species of Thermococci were present in the training data. To see how easily these are learnt, and what effect adding more examples had on the overall accuracy of predictions of species of Thermococci, all 4 were removed. The CNN model was then trained three times with random initialization, and made predictions over the test sets and the four species removed. Species of Thermococci were then added one at a time; after a species was added the model was trained again with replicates and made the same predictions. This was repeated until all 4 of the Thermococci originally in the training set were returned.

Results

The 4-mer distributions of the same genus test dataset was found to be significantly different from the other two. Increasing the number of features improved performance with diminishing returns, and at 25 4-mers as features out of 256 performance is only slightly lower. Both CGR and Reshape-trained CNNs performed worse than the machine learning consort trained on 7-mers. Reshape CNNs were able to learn the importance of order in sequence over 4-mer distributions, with increasing ability as shuffled sequences from other orders with retained 4-mer distributions were used. Thaumarchaeota performance was significantly worse than the other Orders. Certain species within the Euryarchaeota class Thermococci resulted in significantly worse performance than others. Further, errors within some other classes of Euryarchaeota showed predictive relationships. Uncertainty in the text body of this section is given as mean ± SEM unless otherwise stated.

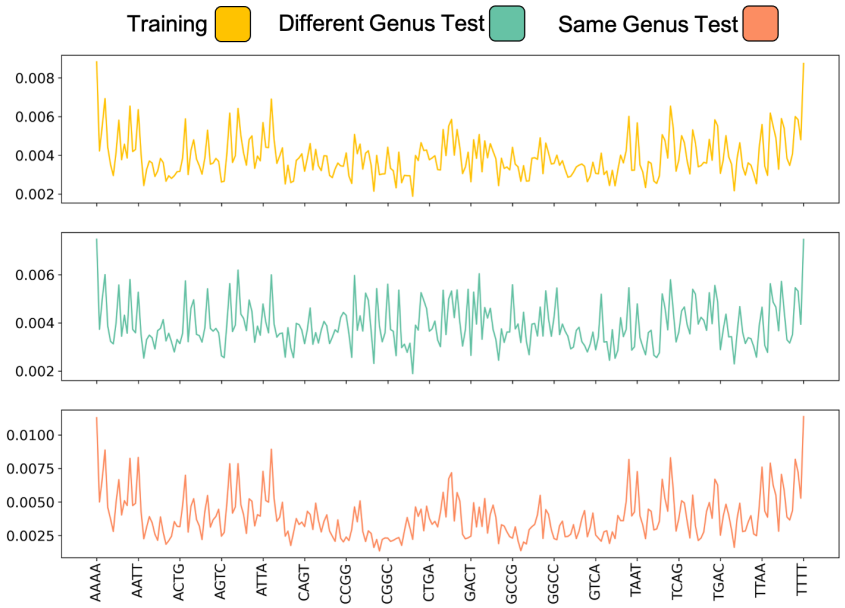

Figure 13. The relative frequency profiles of 4-mers between training and test sets are similar in shape. The 4-mers begin with A, C, G then T, with A-rich 4-mers being leftmost, then C-rich, then G-rich, and then T-rich being rightmost. The difference between the relative frequencies of the same genus set was found to be significantly different from the training (χ^2 = 874.1; df = 255; p = 8.734e-69) and different genus set (χ^2 = 1977.4; df = 255; p = 1.138e-263).

In order to see how similar 4-mer frequencies were between the training set and test sets the 4-mer profiles were compared (figure 13). Whilst there are slight differences in scale, they are all in the same order of magnitude and show similar shape. The most abundant 4-mers by relative frequency are those rich in A and T, in particular AAAA and TTTT are most highly abundant in all sets. To see whether this is an effect of the differences in class size, the

4-mer profiles were also plotted for Euryarchaeota, Crenarchaeota and Thaumarchaeota in the training and two test sets (figure 14). From this it can be seen that the profile shapes are similar between the sets, with the largest difference coming from Euryarchaeota of a different genus, however the peaks and troughs are largely in the same place, it is only the intensity of these that vary. The intra-phyla differences (17.6 ± 2.69%) were found to be significantly less than the inter-phyla differences (38.6 ± 2.44%) (Two Sample t-test, one-tail: t = -4.65; df = 34; p = 2.45e-5).

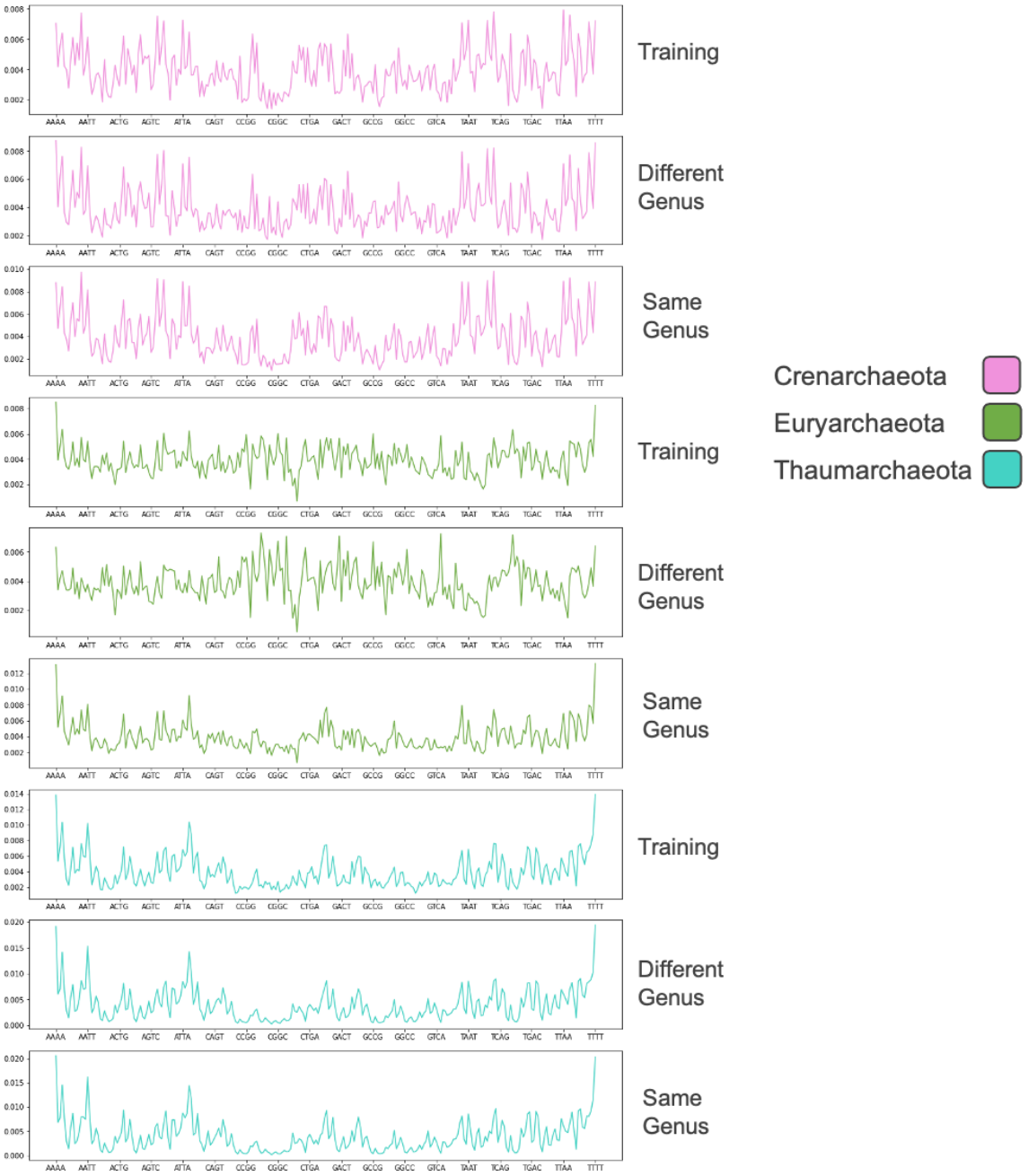

Figure 14. The relative frequency profiles of 4-mers for the three phyla are similar in shape between their respective training and test sets. The 4-mers begin with A, C, G then T, with A-rich 4-mers being leftmost, then C-rich, then G-rich, and then T-rich being rightmost.

PCA us to visualize the multi-dimensional problem in 2D space. Figure 15 shows a PCA of the training data and the two principal components explain 53.89% of the variance. From it we can see a separation between Crenarchaeota and Thaumarchaeota whilst Euryarchaeota overlaps with all other Phyla and Random. The biological data points appear spread out more, with Euryarchaeota showing the most spread. The randomly generated sequences form a fairly central cluster.

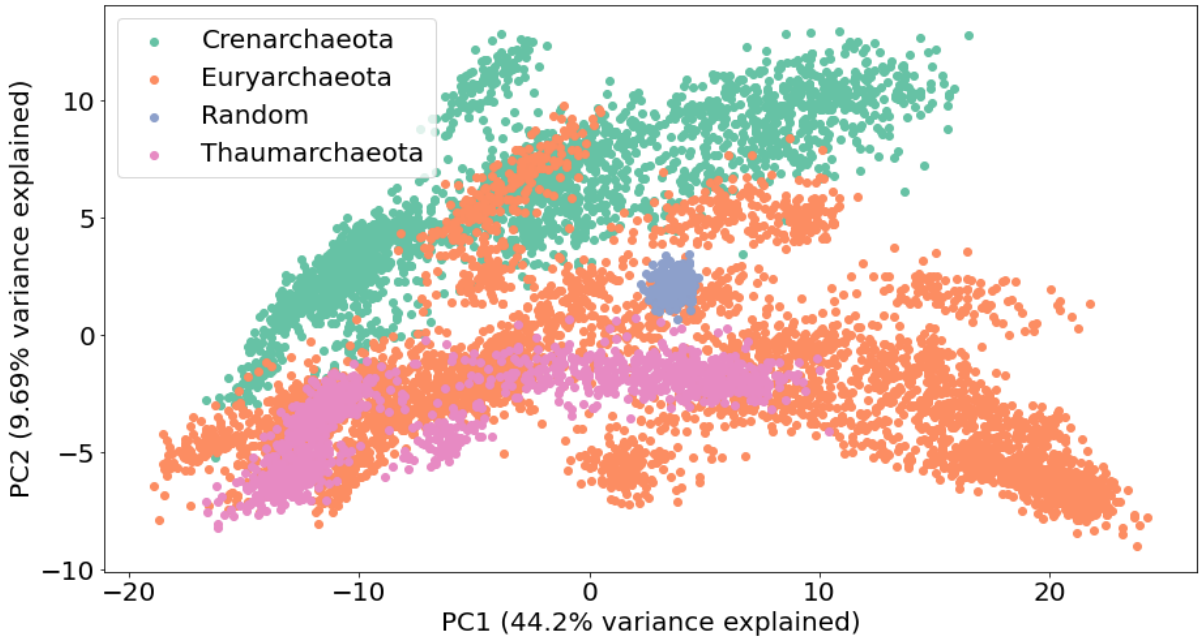

Figure 15. Training data classes form clusters when 256 dimensions are reduced to 2. All sequences from the training dataset were used in a PCA. The most obvious clustering is crenarchaeota, sequences of which form a most separate band above euryarchaeota and thaumarchaeota. The separation of the two largest sets may explain why thaumarchaeota overlaps with euryarchaeota. A minority of euryarchaeota sequences cluster with the crenarchaeota. Random sequences form a uniform, tight cluster in the center of the plot, since the 4-mers in those are fair more evenly distributed than in biological sequences.

To give a better visualisation of the problem space, t-SNE was used to embed the data into 2D space (figure 16). Smaller clusters seem to form within the three Phyla, with each cluster being made up of only data points belonging to one Phylum, apart from a few outliers causing some clusters to be heterogeneous. The number of clusters is close to the number of species used for each class, but less. Regions of clusters appear as though they could be multiple clusters in close proximity, so to investigate it is necessary to look at a level more fine-grain than phyla. This is excluding the randomly generated sequences, which are still forming one cluster.

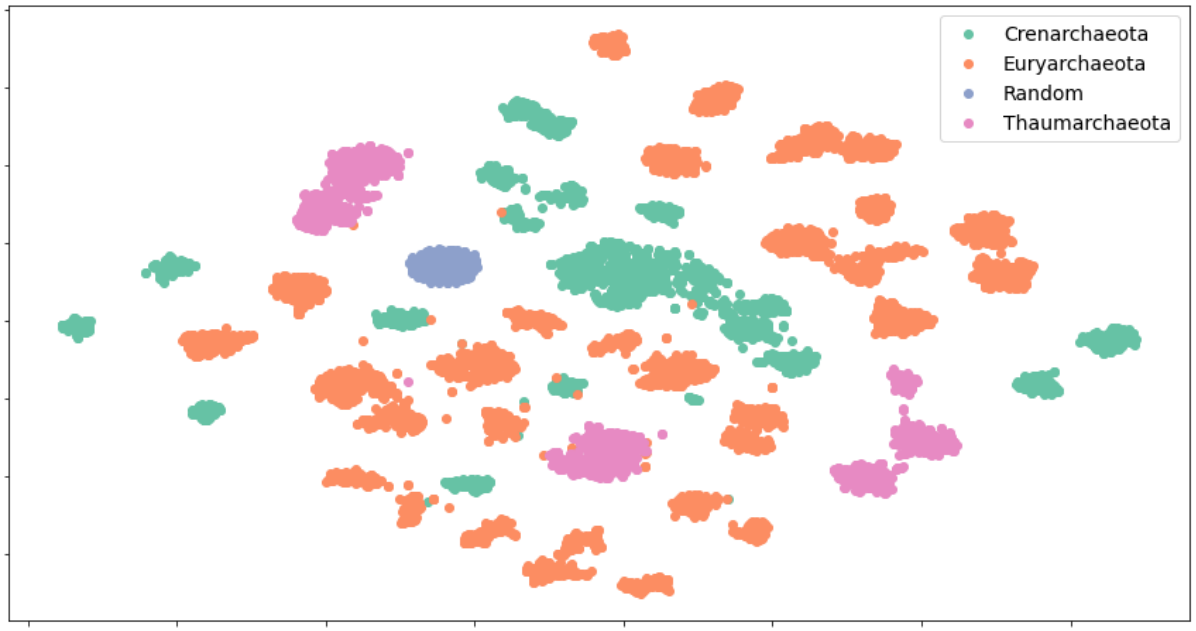

Figure 16. t-SNE of training data shows clusters homogeneous in class. Excluding the randomly generated data, the classes do not form one cluster each; instead multiple smaller clusters form, which are largely homogenous. The t-SNE uses all training data of the 4-mer trained models.

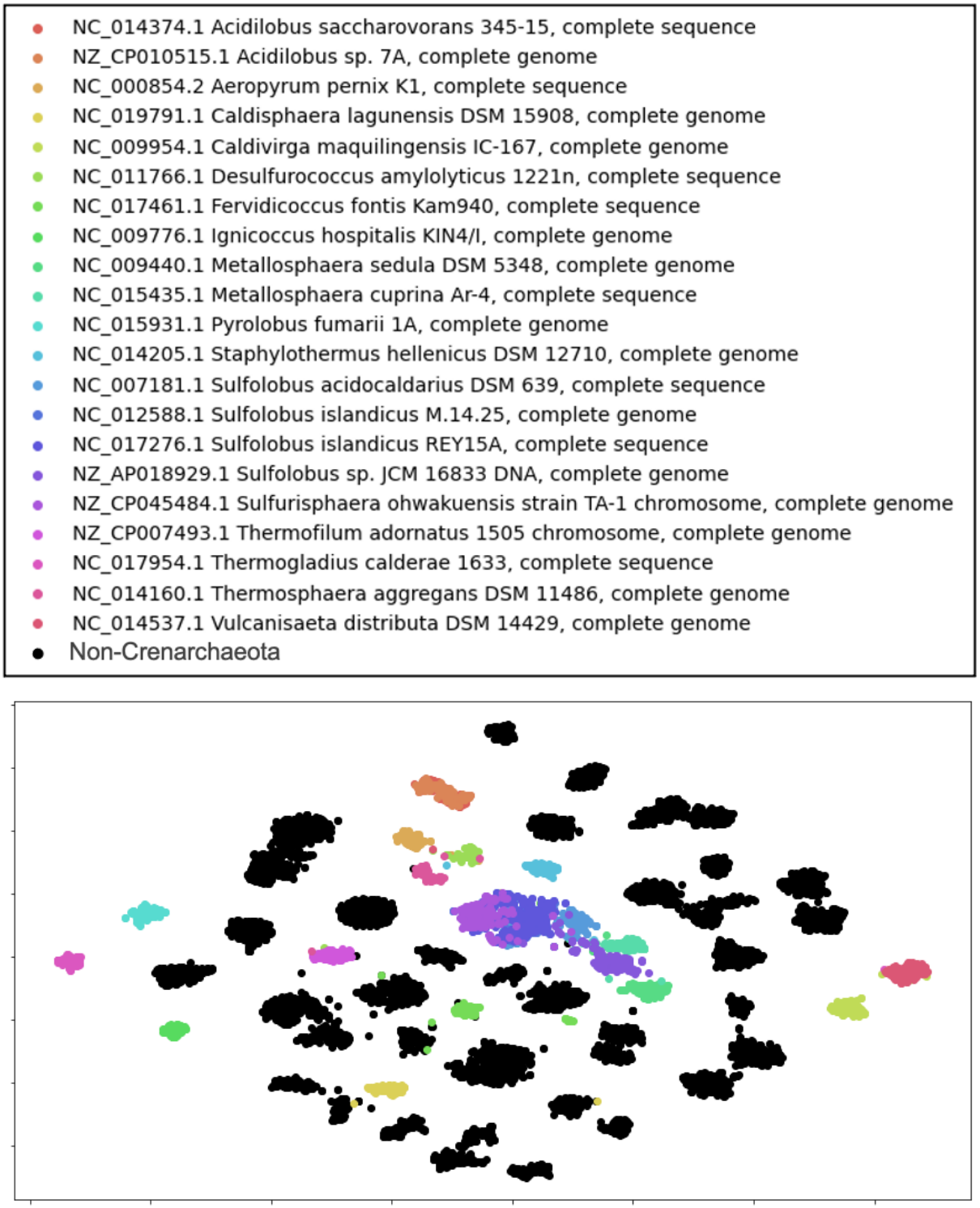

Figure 17. t-SNE clustering of sequence 4-mers corresponds to different genuses and species. All species in black are non-crenarchaeota. Two *Acidolobus* species overlap in orange and red at the top of the plot, whereas *Sulfolobus* form separate clusters within a larger cluster of multiple genuses in the centre of the plot.

Figure 17 shows that these clusters correspond mostly to individual species, however the central cluster is a mix of species from the genus *Sulfolobus* and several other species from the order Sulfolobales. Other closely related species overlap entirely.

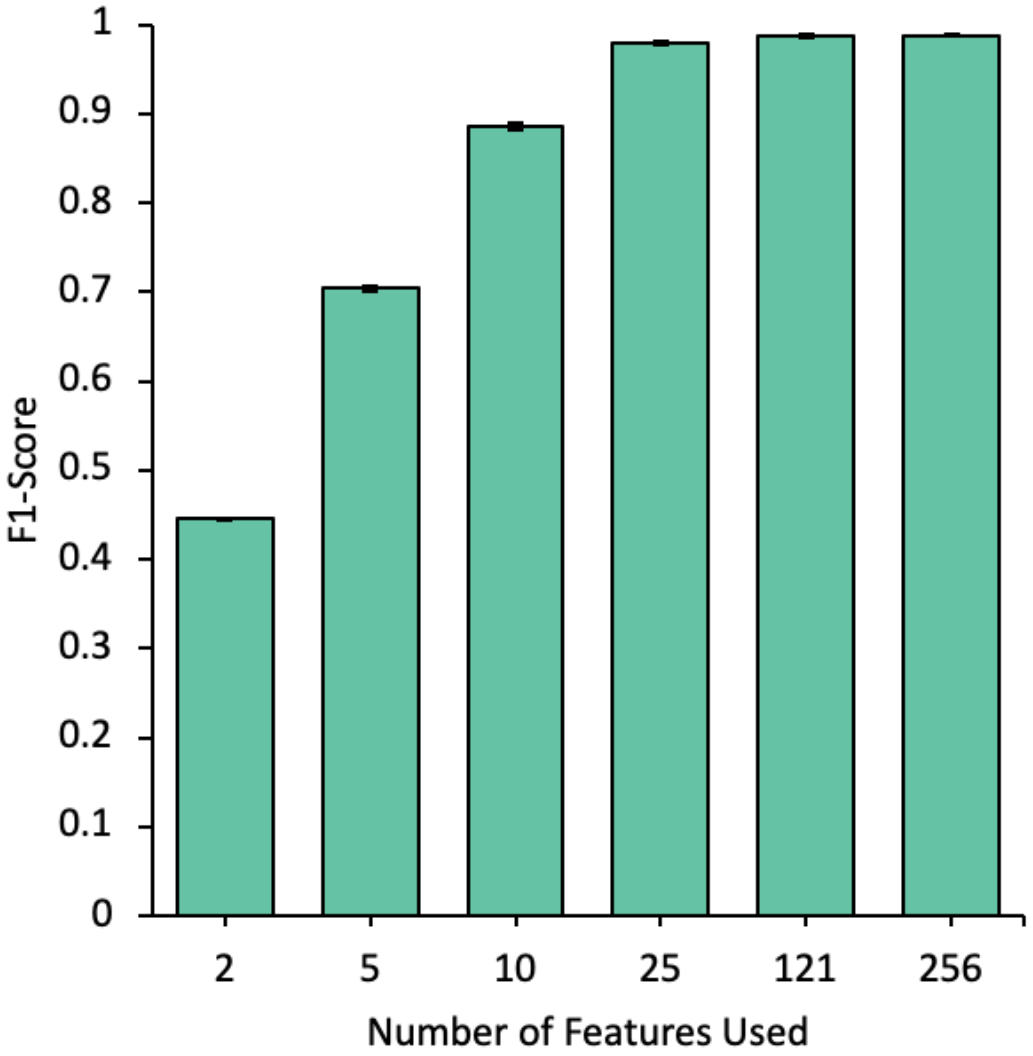

Figure 18. F1-score increases with the number of 4-mers used to train machine learning models. F1-score of the top 5 scoring models on the combined training sets for each number of 4-mers were taken. Error bars show SEM. Performance stops increasing significantly at 121 features (0.988 ± 0.0011).

Classification of sequences at the level of phyla using 4-mers was therefore carried out. To see how much information 16384 bases carries, the number of features used was reduced, since if features are redundant, losing them should not reduce the information carried significantly, and therefore the performance of classification should not be reduced either. Redundancy in the features used is somewhat obvious, as it would be expected for certain 4-mers to be correlated due to sharing part of their sequence. For example, AAAA could be correlated with AAAG. Unsurprisingly, increasing the number of features used to train the model and make predictions from leads to an increase in F1 score. The increase is asymptotic and by 25 features the F1 score is only 0.01 below that obtained from using 121 features and all 256 features, yet this difference is significant (Friedman Test, $\chi^2 = 8.4$; $p = 0.015$). At 121 features there is no significant difference between the F1-Score obtained and

that obtained using 256 features (figure 18). As the number of features increased, more linear models entered the top 5, displacing parametric models.

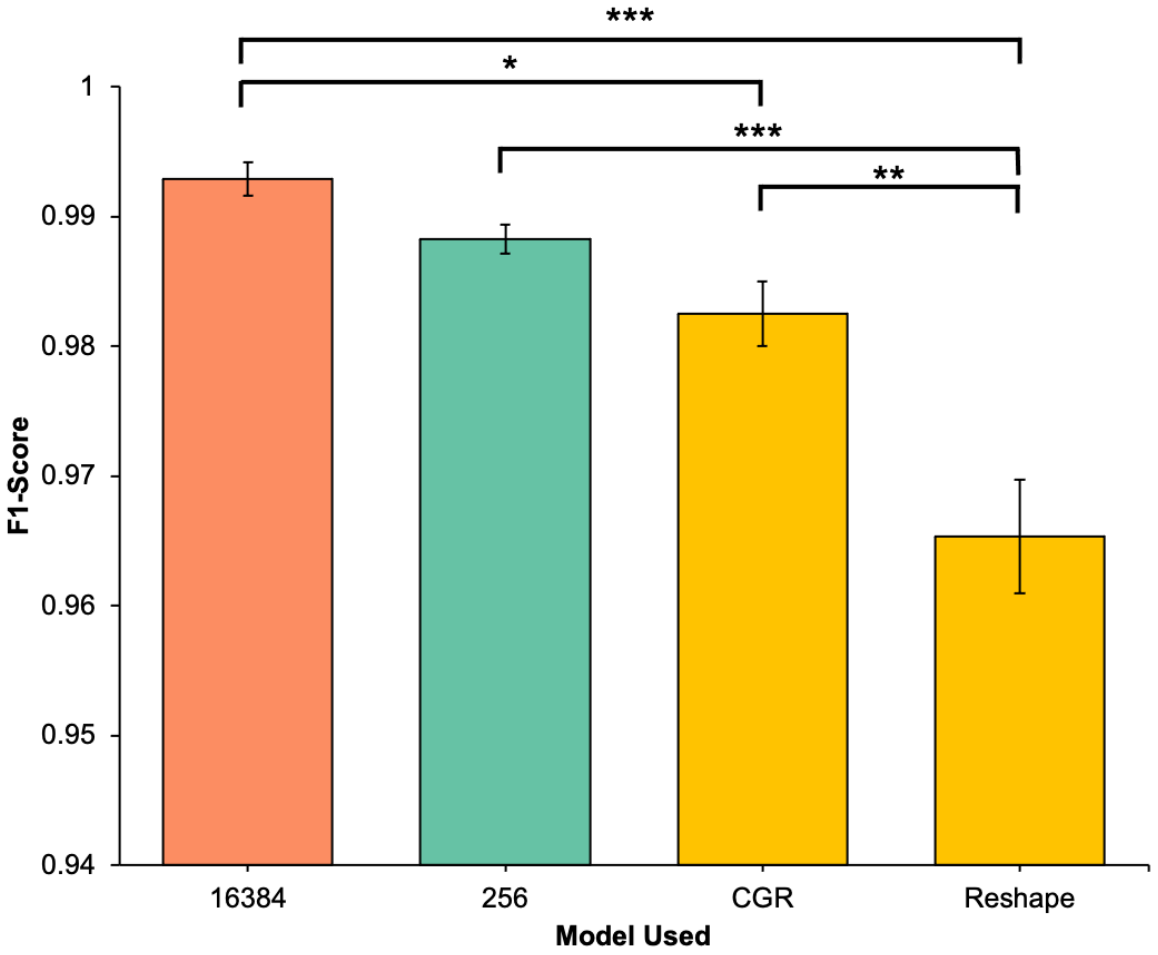

Figure 19. The CNN models perform significantly worse than the 7-mer machine learning models for both image representations. K-mer trained models take the five-best performances and CNNs use the performances of 3 models trained with the same parameters. Additionally, the Reshape-trained CNNs have significantly lower F1-scores than both the CGR-trained CNNs and the machine learning models trained on 4-mers. Error bars show SEM. *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

Since the CGR is learning effectively visual representation of 7-mers, 7-mers were also used as features for machine learning models. When comparing the top 5 machine learning models and both CNNs there is a significant difference in F1-score between the models used (ANOVA: $F = 30.72$; $df = 3, 11$; $p = 1.2e\text{-}5$). Post-hoc testing revealed that the differences were between the model using 16384 features, which had the highest F1-score (0.993 ± 0.00122) and both CNN models (CGR: 0.983 ± 0.0025, Reshape: 0.965 ± 0.00441). Additionally, the F1-score of the Reshape-trained CNN was significantly lower than that of the model using 256 features (0.988 ± 0.0011) and the CNN trained of CGR data (figure 19).

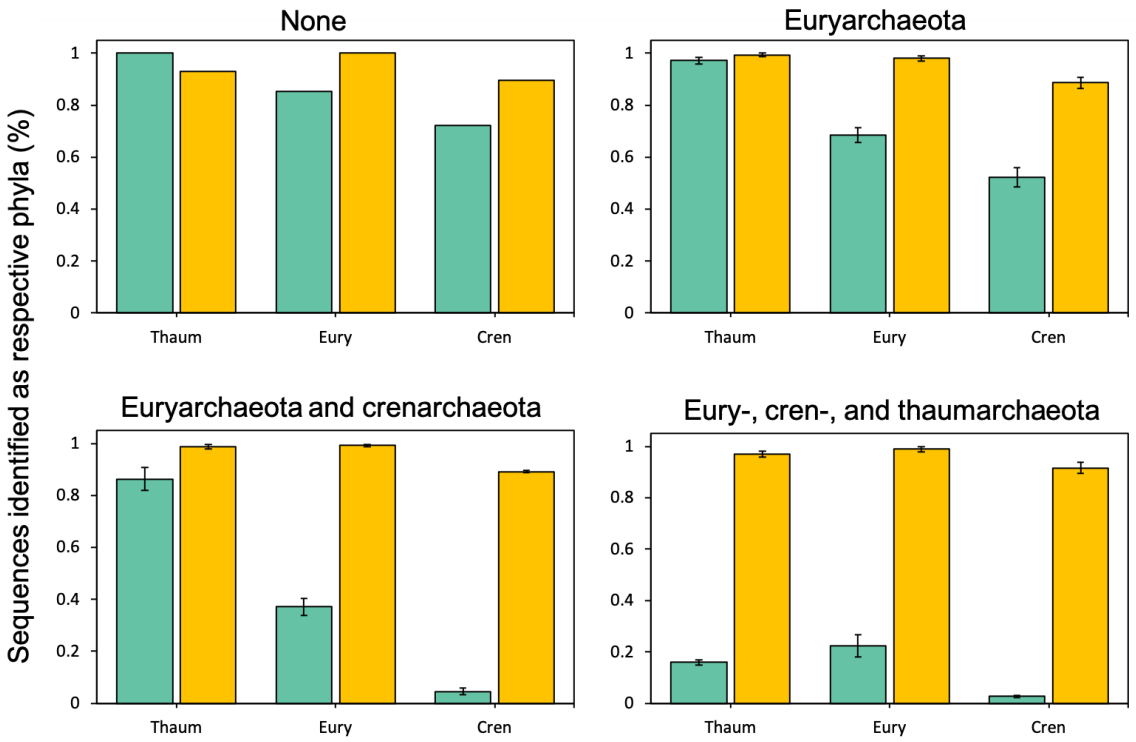

Figure 20. Sequences shuffled into a random order whilst preserving 4-mer count are initially mistaken as sequences from their original phyla. In total, 464 sequences were randomly chosen and shuffled, and used as training sequences, and additional sequences were used to make a test set. Adding an additional class with examples of shuffled sequences added one phyla at a time, there is a reduction in this type of error. Test shuffled sequences are compared to the same sequences without having been shuffled. Error bars show SD, plot titles show phyla from which shuffled sequences have been added to the testing set.

One possible advantage of using sequence over the summarisation with K-mers in that order may be important in certain cases, which only using the raw sequence data would reveal. By changing the order of the sequence but preserving 4-mer count, the sequence is made identical in terms of 4-mers and similar in close K-mers (3 and 5). This can be used to determine the importance of order. Not only that, it assesses the importance of 4-mers in the classification by Reshape-CNN. Sequences were shuffled using ushuffle to preserve 4-mer counts, but generate random and non-biological sequences. Using the same 4-class trained models as previously the shuffled sequences were classified as the phyla of which they were from before shuffling. Once an additional class is added for these shuffled sequences, and examples of shuffled sequences added one phyla at a time, fewer of these sequences were classified as their phyla before shuffling. Figure 20 shows that adding shuffled Euryarchaeota sequences reduces both the percentage of shuffled Euryarchaeota and Crenarchaeota sequences identified as their respective phyla. The percentage of shuffled Thaumarchaeota sequences did not reduce, and only reduced a little after shuffled Crenarchaeota sequences were also added to training data. The most significant reduction in the percentage of shuffled Thaumarchaeota sequences identified as Thaumarchaeota came only after shuffled Thaumarchaeota were added to the training data.

No significant difference on the F1-scores was found between the Reshape-trained models which included randomly shuffled 4-mers in the training data as a separate class and those

that did not include this fifth class. Looking into the Reshape-trained models which used only four classes there was found to be no significant difference in F1-scores between the species in the testing sets that were in the same genus as those in the training data and those that were in different genuses.

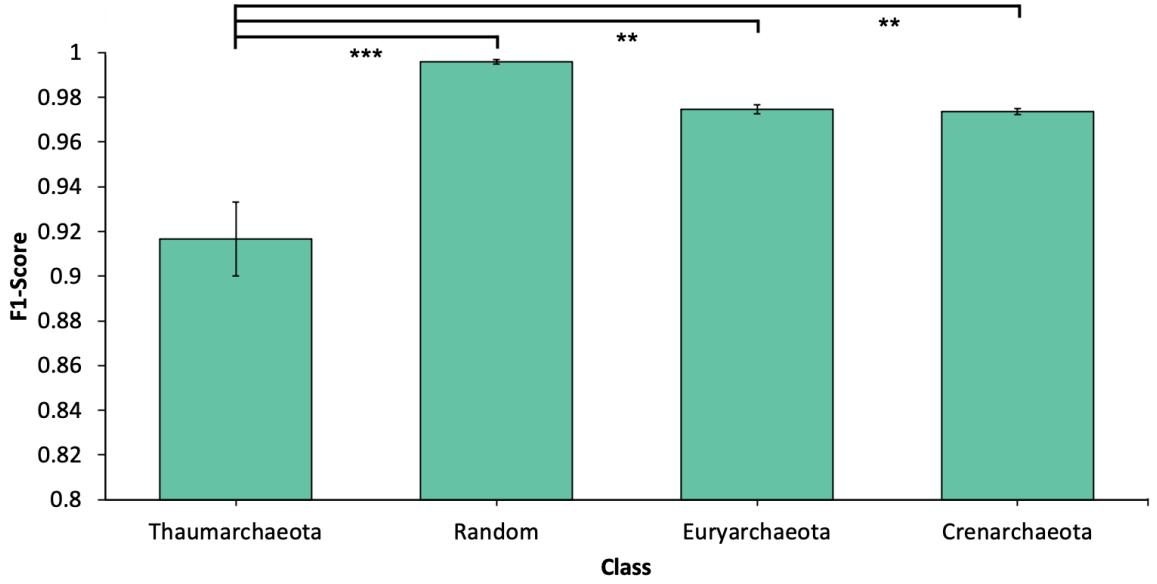

Figure 21. Mean F1-scores for all the classes. The F1-Score is based on the performance of three models on both training sets combined. The mean F1-Score for sequences from Thaumarchaeota was significantly lower than all other classes in the four class model. Error bars show SEM. *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

However, the classification class significantly impacted performance of the Reshape-CNN models on all training examples for each class (ANOVA: $F = 16.74$; $df = 3, 8$; $p = 0.000827$). Post-hoc testing revealed that the differences were between Thaumarchaeota, the class for which performance was lowest (0.917 ± 0.0164), and all other classes (figure 21). There were no significant differences found between phyla for the mean percentage differences of 4-mers between training and test species.

In order to gain insight into why the Reshape-CNN performed worse overall than any other model, the performance on Euryarchaeota, the largest class, was broken down further into performance on phylogenetic class. The performance of most classes is high, with first quartiles around or above 95% accuracy (figure 22). However, there is a significant difference in accuracy between the phylogenetic classes of species classified (ANOVA: $F = 8.69$; $df = 7, 181$; $p = 3.47e\text{-}9$). Post-hoc testing revealed that the differences were between Thermococci, which had the lowest accuracy ($90.6 \pm 2.63\%$), and all other classes, apart from DHVE2 (figure 22).

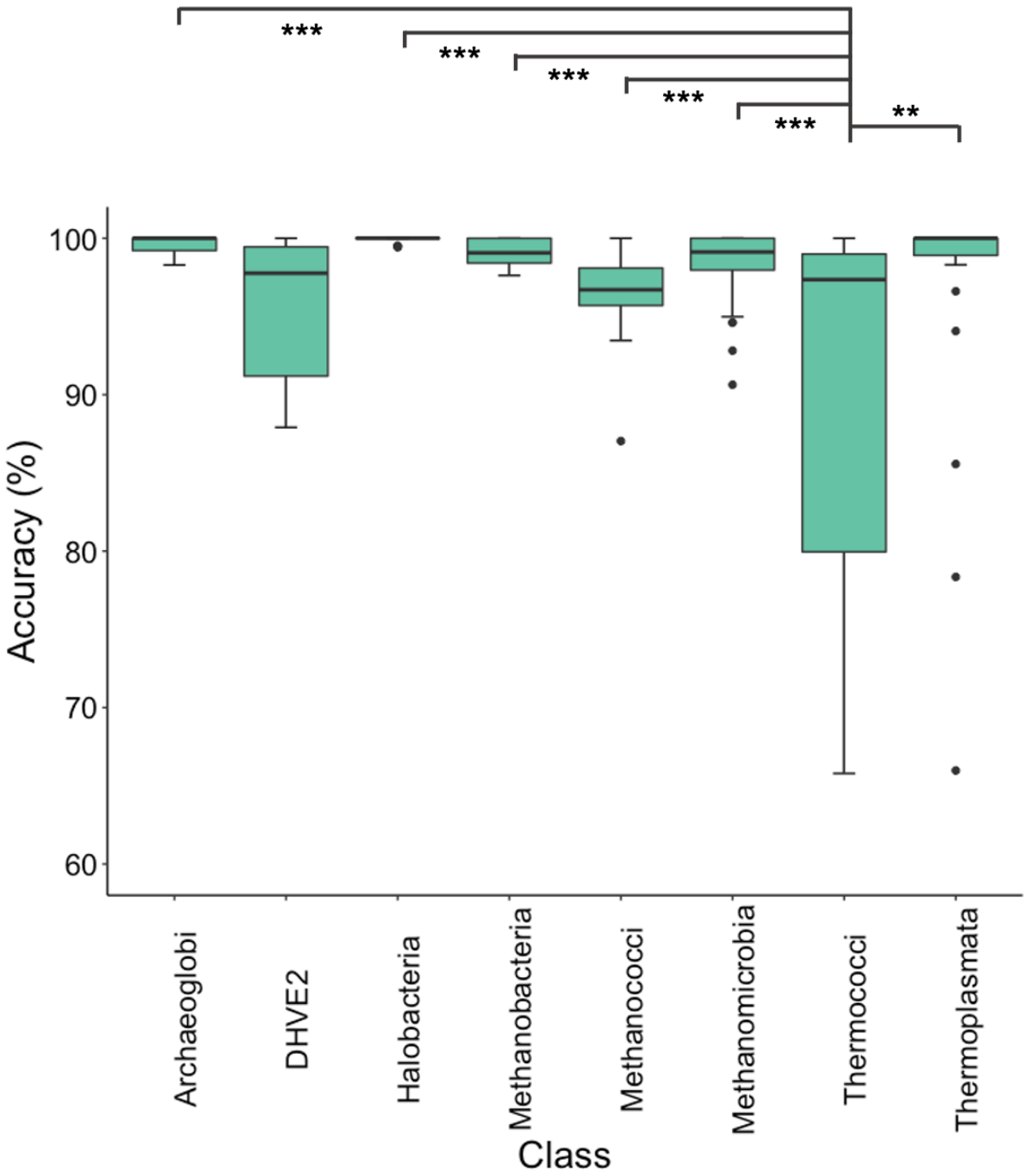

Figure 22. The accuracy of one phylogenetic class, Thermococci, is significantly lower than all but one of the other classes. The performance was of Reshape-CNN models without shuffled-sequences in the training data, meaning they had 4 possible output classes (the three phyla and random). Only DHVE2 sequences were not classified significantly better than those belonging to Thermococci. *** $p < 0.001$, ** $p < 0.01$.

Since this class was performing worse than others it was removed from the training data and the models retrained. Reads from one species were added back in one at a time and the

model retrained after each. From this it was found that there is a significant difference in accuracy of prediction of Thermococci between the models trained on different numbers of Thermococci species (ANOVA: F =20.68; df = 4, 85; p = 6.25e-12). Post-hoc testing revealed that the differences were between models trained on 0 Thermococci, which had the lowest accuracy (46.67 ± 5.37%), and all other models, and between the model trained on one Thermococcus species (74.84 ± 4.29%) and the model trained on 4 (90.6 ± 2.63%) (figure 23).

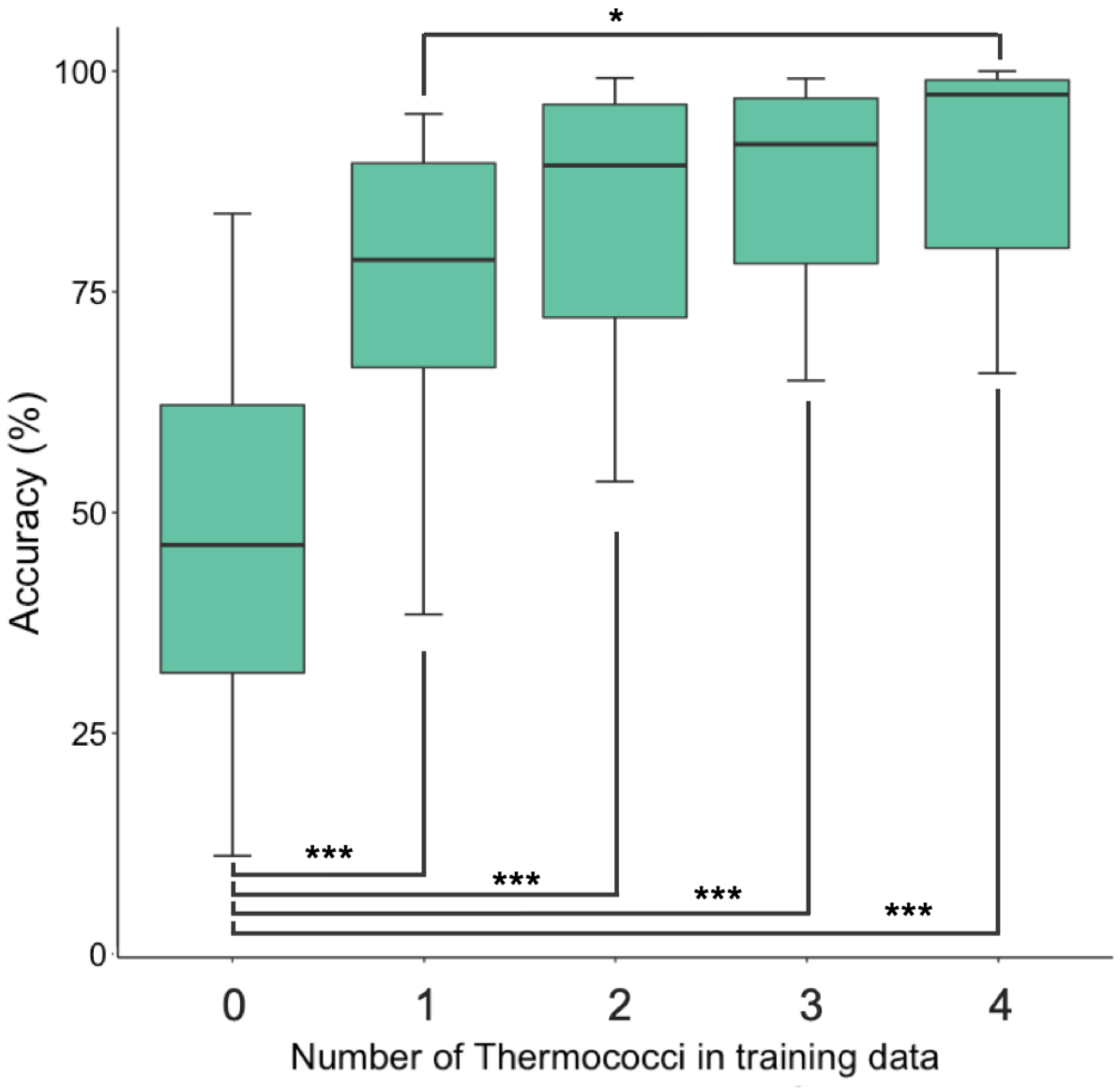

Figure 23. Increasing the number of Thermococci in the training data improves classification accuracy of Thermococci species. The models were retrained after each species was added to the training data. The significant improvements were between 0 and all other numbers of species in the training data, and 1 and 4 (how many there were before removal). *** p < 0.001, ** p < 0.01.

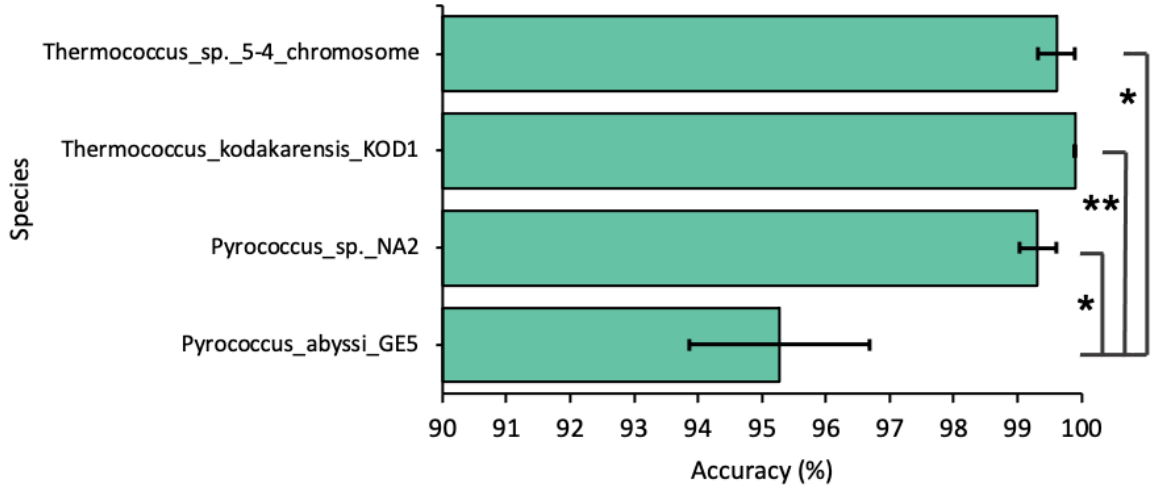

Figure 24. Accuracy on species in the training data. Sequences belonging to *Pyrococcus abyssi GE5* were classified with significantly less accuracy than the other three Thermococci species trained on. Error bars show SEM. ** $p < 0.01$, * $p < 0.05$.

After the final species was added back in, the accuracy of predictions on four species used to train revealed a significant difference between the accuracy of predictions for each species (ANOVA: $F = 8.77$; $df = 3, 8$; $p = 0.00656$). Post-hoc testing revealed that the differences were between *Pyrococcus abyssi GE5*, which had the lowest accuracy (95.4 ± 1.41%) and was significantly lower than the accuracy for all of the other species (figure 24).

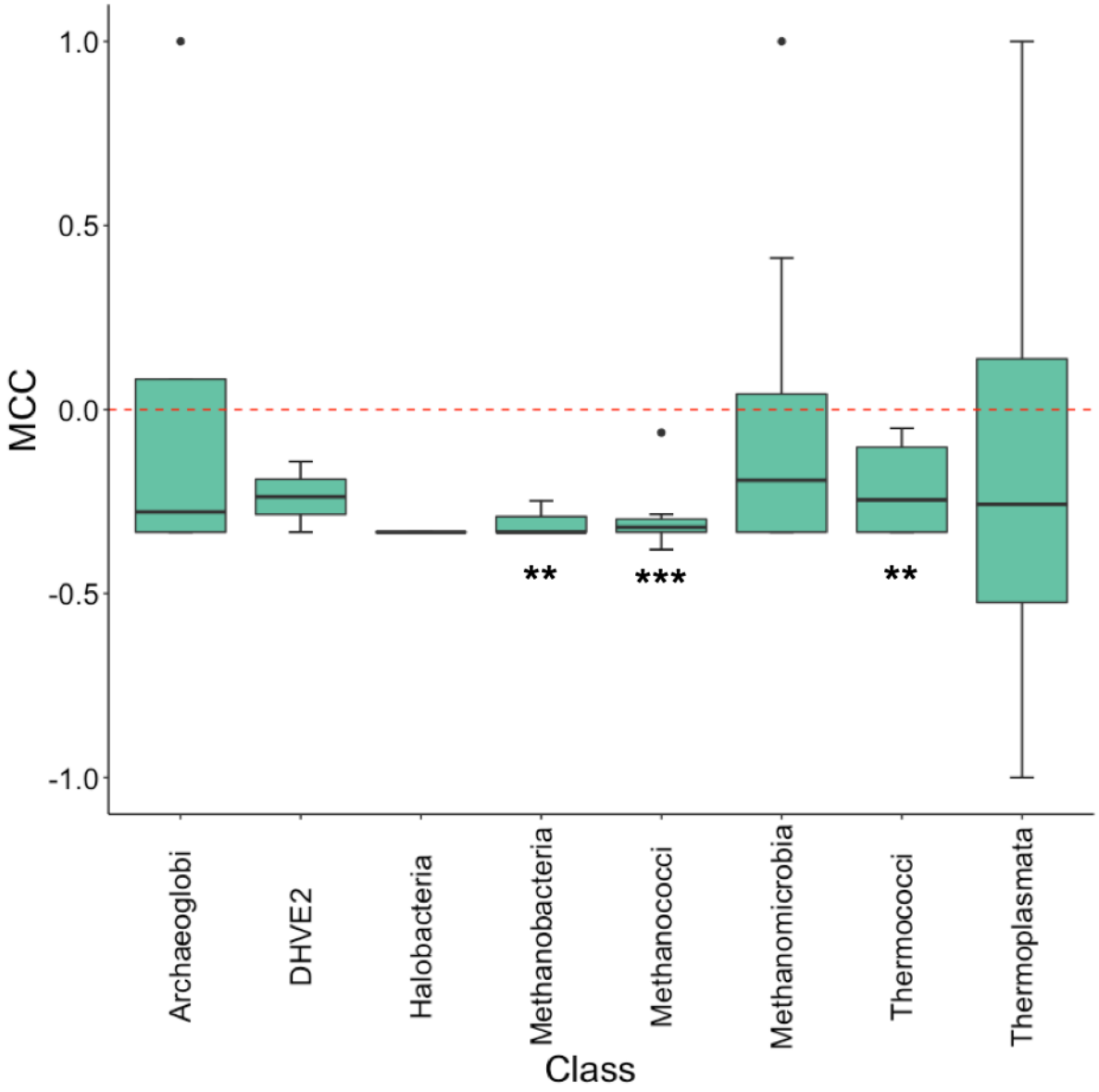

Figure 25. The mean MCC scores for errors in some phylogenetic classes are not significantly different from 0, but Methanobacteria, Methanococci and Thermococci are. MCC was calculated for each pairing of three replicate Reshape-CNNs used. *** $p < 0.001$, ** $p < 0.01$.

To test whether the errors were happening randomly or at the same positions for each duplicate model, the mean Matthews Correlation Coefficient (MCC) was found between each pairing of the models. The mean of the mean Matthews Correlation Coefficient of the classification errors were significantly different from zero for Thermococci (One Sample t -test: t =-3.95; df = 5; p = 0.005425), Methanobacteria (One Sample t -test: t =-10.75; df = 2; p = 0.004275), and Methanococci (One Sample t -test: t =-8.52; df = 7; p = 3.04e-05). Figure 25 shows this. All means were also greater than -1 apart from that of the MCC of Thermoplasmata.

Discussion

4-mer Distributions

The 4-mer distributions are different between the same genus testing set and the other two. This may be due to different amounts of each Order in them, but also due to differences between species of certain genus. Most machine learning models assume that the testing set is drawn from a stationary distribution (Hulten, Spencer, and Domingos 2001). Despite this, the performance on both testing sets was not significantly different, suggesting the difference in 4-mer distributions could be due to make-up rather than differences in DNA sequence of the species used.

The variance in distributions between phyla are greater than within, which is consistent with 4-mers distributions as a taxonomic signal (Karlin and Ladunga 1994). The most common 4-mers are AAAA and its complement TTTT. These 4-mers have been found in binding sites involved in the regulation of heat shock response in hyperthermophilic archaea (Vierke et al. 2003). They also feature in important regions in the regulation of the cold response in some archaea: the pentamer AAAAA is needed to induce cold stress DEAD box RNA helicase in *Thermococcus kodakarensis (Nagaoka et al. 2013)*. The composition may reflect the extremophile nature of the archaea. The variation in 4-mers remaining within phyla is likely due to species differences in the dataset.

The PCA shows separation in two dimensions with approximately 54% of variance explained, which again suggests that the use of 4-mers is sufficient in some degree to separate DNA sequences at the level of phyla, which has been widely accepted (Pride et al. 2003). The clear lack of overlap between Crenarchaeota and Thaumarchaeota suggests that the two phyla could be easier to separate than either from Euryarchaeota, which has data points overlapping on the 2D projection (figure 15). The spread on the projection is proportional to the number of examples used for each Phyla, supporting the idea of 4-mer frequencies being species specific (Teeling et al. 2004). The randomly generated sequences form a relatively tight cluster, which is consistent with the random distribution of 4-mers in each randomly generated sequence, meaning that they are distributed evenly around the expected values of an equal number of each 4-mer, so can't be separated as much as the other non-random classes by these values, so have less spread. Overall they support the idea that even simple linear models should be able to separate the classes. The overlap may result in certain examples being classified incorrectly, or could be because it is a low dimensional representation of a high dimensional feature space, and in the original higher dimensions separation could be easier.

The t-SNE provides an alternate mapping to lower dimensional space. This again is good evidence that the classes could be separated at phylum level, and even below, using only 4-mer counts. However, certain species from the same genus appear to overlap in clusters. Apart from this, the near-homogeneous clusters suggest that they exist in clear divisions in the feature space, though the few outliers may prevent perfect classification by models. The random sequences should be the easiest to identify as they always form the same cluster and have no outliers.

Effect of Features

That F1-Score increases diminishingly with the number of features (4-mers) used to train the machine learning models is unsurprising considering that they were selected to remove the least informative first. A F1-Score of roughly 0.45 is achieved using only the ratio of two 4-mers, which suggests large differences between the phyla. Given that K-mer counts have been used to separate species, it might follow that a smaller subset of these features could separate sequences into phyla. That performance is very close between 25 and 256 features also supports this. It also suggests that the length of sequence used (16384 bp) could possibly be reduced, since using less than 10% of the input features yield results almost as good. This could be because of the relatedness between 4-mers, as can be seen from the removal of highly correlated 4-mers, to leave 121 features, not impacting performance significantly. It should also be noted that the top five highest performing models change, shifting from non-linear decision tree models to linear models, likely due the greater dimensionality from the extra features aiding separation.

As specificity of a K-mer increases with length, the risk of overfitting increases as the training data become sparser. Machine learning models using 7-mers achieve a higher F1-Score than those using 4-mers. However, it seems that 7-mers are still sufficiently generalizable for separating phyla whilst also outperforming 4-mer models. The lengths of K-mers used to bin are often much longer than either, however (Sczyrba et al. 2017).

Representation Performance

The collection of best performing machine learning algorithms using 7-mers also outperforms both CNN approaches. The CGR representation is presenting 7-mers in an image, so contains the same information, yet performance is worse despite this. This may suggest that the models were overfitting or underfitting on the training data. The architecture used was originally designed for 224x224 images for the ILSVRC, which has over 1000 classes and much larger training sets (around one million images before augmentation) than the one used in this study. Since increasing training data size can help reduce the effects of overfitting, using an architecture designed for more data than is provided could therefore lead to overfitting. As such, the architecture may be too deep for the purpose at hand, given the current amount of data. Solutions to this could include changing the regularization methods to make their effects stronger, such as increasing the frequency of dropout or tuning weight decay.

More simply, reducing the number of layers may be a viable approach, as many neural networks produced to process DNA use few layers (Le and Nguyen 2019; Busia et al. 2019). Using the same architecture with reduced layers would be possible, as the paper the architecture is based on also includes shallower models. It is not obvious how the performance will be affected, and alternate shallower models may perform better. Increasing the number of examples to combat overfitting might also work, however when performing classification at phyla level a large array of species are needed across and within genuses to ensure generalization of the model. This is a problem when the number of archaeal sequences freely available is relatively few, especially compared to the much better studied bacteria (Adam et al. 2017). Reducing the resolution of images would be a possible approach, as it may be the lengths used are too long. Indeed, other studies at the level of

species classification have used much shorter sequences (Dick et al. 2009). Reducing image size not only increases the amount of examples at the cost of information in each example, but allows classification of smaller sequences for Reshape curve, for which the minimum length is currently 16384 bp, since every pixel needs to be filled. This would allow better application to metagenomic problems which often produce fragments that can be shorter than this, and also allows for the classification of shorter plasmids. For CGR this would result in a sparse representation that may adversely affect performance.

Changing the image size would demand further changes to the architecture. Generally, the ratio between the split of training and testing data is around 80:20 and 70:30, though on smaller sets cross-validation may be preferential. In this study the testing and training data for biological sequences (i.e. excluding random) was close to 50:50. Alternate ratios would need to be experimented with, but including more species will either reduce the number of species from different genuses available or increase the number of species belonging to a genus already in the training data and may then increase the risk of overfitting to that genus. However, without trialling and tuning this by experiment the effects of changing the split is unclear. The number of epochs was determined by looking at the validation versus test performance and using the configuration before training loss exceeded validation loss, an indicator of overfitting. However, this may not be the ultimate best configuration for this problem, and introducing patience in the training regime could result in better performing models (Hutter, Lücke, and Schmidt-Thieme 2015).

Moreover, various hyperparameters, such as the aforementioned regularization methods, but also the number of epochs to train for and batch size, could be tuned via a systematic search of possible combinations to account for their effects on each other. This could also explore the possibility of underfitting by comparing higher epoch models. Exploring different architectures with this search is an option for further performance improvement. The reason neither part of this was performed was because of the computational expense and time restraints.

The worst performance was that of the Reshape CNN, which may reflect poor suitability of the model to the representation, or a requirement for longer training on that data type. Firstly, the Reshape representation was chosen in an attempt to allow the model to uncover patterns in DNA useful for classification, including those at a distance, in an unsupervised manner (Yin et al. 2018). This is different to supplying the CGR representation, for although that allows patterns in 7-mers to be discovered, it is only a summation of the raw sequence, whereas using Reshape provides the sequence itself. It was unclear whether this would result in learning aberrant features due to the random nature of the training-validation split and the relatively few images that would result in incorrect classification. This is especially true for those only seen in few or one class in the training dataset, but existing in many in the test. The use of raw sequence could also introduce noise that the model can overfit on. Again, only by testing different training regimes and hyperparameter tunings as described above will the root of the difference between representation performances become apparent. CGR may be a better representation for such a high-level classification, since, due to its fractal nature, similar K-mers are close to each other. This means the kernel will pass over

correlated K-mers and this may allow more efficient summarisation than for a raw sequence, and abstractions will be composed of K-mers that are more and more distal.

Additionally, the performance of models using 7-mers and 4-mers not being significantly different suggests diminishing returns in performance improvement from K-mers for increasing values of K, or a need to tune models for different K-mer values. It could also tell of a saturation in performance at the level of phyla, due to being a very high taxonomic level. The errors remaining could be regions of uncharacteristic K-mers or those uniform across class, perhaps as a result of conserved genes and presence in unbalanced classes skewing how such a feature affects classification.

Shuffled Sequences

One possible advantage of using sequence over the summarisation with K-mers is that order may be important in certain cases, which only using the raw sequence data would reveal. From the results on shuffled 4-mers it can be seen that initially order is not valued highly; the amount of shuffled Euryarchaeota, Crenarchaeota, and Thaumarchaeota classified as their non-shuffled respective phyla is very close to the number of actual sequences classified as such. 4-mers are important to the model for classification, since maintaining them but changing the order of the sequence has very little effect on model performance. Indeed, more shuffled Thaumarchaeota are classified as Thaumarchaeota than unshuffled Thaumarchaeota, which lends support to the idea that certain sequences can be deleterious to performance due to noise or aberrant features. The models learn the importance of 4-mers, and when the Thaumarchaeota sequences are shuffled they preserve these, so the shuffling must also remove some arrangement of bases causing confusion with another class.

Adding shuffled Euryarchaeota sequences reduces the number of shuffled Euryarchaeota and Crenarchaeota identified as their unshuffled phyla, without reducing the classification performance on unshuffled sequences. This means that not only the model learning differences between sequences with the same 4-mer counts, but that the features learnt can be applied to other phyla. The reduction is not total, indicating that 4-mers are still important features to the model, perhaps due to the relatively few shuffled sequences added (less than 200 per phyla), but their importance is reduced. However, not all phyla benefit; Thaumarchaeota is unaffected. When shuffled Crenarchaeota are added in addition to shuffled Euryarchaeota, there's a only minor reduction in the amount of shuffled Thaumarchaeota identified as Thaumarchaeota. Crenarchaeota and Euryarchaeota seem to benefit more from having the other's shuffled sequences added than Thaumarchaeota does from either. This could indicate a greater similarity in key 4-mers between the two phyla that are reduced in classification importance for either phyla. It could also be due to a greater similarity of features learnt in the place of 4-mers. This is reinforced by the addition of shuffled Thaumarchaeota to training data reducing the number of shuffled Thaumarchaeota in the test set being classified as Thaumarchaeota but not reducing the performance on unshuffled being correctly classified as Thaumarchaeota. This means that the models can tell the difference between shuffled and unshuffled Thaumarchaeota. Therefore the very slight reductions in misclassification of shuffled Thaumarchaeota sequences when Eury- and Crenarchaeota were added isn't due to a greater difficulty in classifying Thaumarchaeota.

Once shuffled sequences from all phyla are added to the training data it is shuffled Euryarchaeota sequences that have the most classified as Euryarchaeota. This may be due to Euryarchaeota being the largest class in the training dataset, and having a larger variance in 4-mers and sequences, and since equal amounts of shuffled reads were added for each phyla, it may not be representative enough of the whole phyla. Adding more reads could test this, and it may simply be that the randomly shuffled class needs more examples. This could tell us whether 4-mers have been reduced in importance for classification or if specific examples are needed if they present unique or rarer features. The effect of adding more shuffled reads should also be investigated per phyla, to see which produces the greatest rate of reduction in misclassification.

Phylogenetic Differences

Thaumarchaeota also has a significantly lower F1-score than all other classes for the four-class Reshape CNN models, which may reflect similarities between it and the larger classes, or that the number of Thaumarchaeota in the training data are too small to properly generalize. Thaumarchaeota only has 10 species in the training dataset as opposed to 31 and 21 of Euryarchaeota and Crenarchaeota respectively. These models the 4-mers were relatively important features, hence the results on shuffled and unshuffled sequences when no shuffled were in the training set. Given that there were no significant differences between phyla for mean percentage differences of 4-mers between training and test data, it cannot be that the Thaumarchaeota had ‘harder’ test data.

Thaumarchaeota were first classified as Crenarchaeota from SSU rRNA results, and it was not until it was shown that they lacked typical genome signatures of Crenarchaeota using newly available genome sequence data that they were classified as a distinct phyla (Brochier-Armanet et al. 2008). The study points out that *Cenarchaeum symbiosum*, now a Thaumarchaeota, was found to have homologs of EAG3 (Cubonová et al. 2005), an archaeal histone gene, which is present in most Euryarchaeota but not Crenarchaeota. It could be sequences such as these which are affecting the performance of the models on Thaumarchaeota. If such features are only present in the test data, this may explain why they result in incorrect classification, likely as Euryarchaeota. To address this, using different compositions of training and test sets could reflect whether the data is not enough to generalise on, or if there are problematic species, the effect of which will be increased in a smaller test set. This was not performed due to computation and time limitations. Due to the relative lack of Thaumarchaeota sequences, many studies have focussed on assembling the genomes of species from novel environments (Zhong et al. 2020). It is therefore likely that the number of available sequences for unique species will increase. Using these in the future should improve performance or again reveal hard to learn species.

Of all the Euryarchael classes, performance on Thermococci was significantly lower than most others. This may indicate that species belonging to this class are more varied or have features that are harder to learn. The Thermococci genuses Thermococcus and Pyrococcus are both obligate anaerobic chemoorganotrophs (Lee et al. 2008) and can grow at temperatures approaching 100°C (Weinberg et al. 2005). They differ, therefore, to some of the other Euryarchaeota, which include halophiles, methanogens and lithotrophs. However,

they are present in the training data, so these differences in metabolism should be accounted for.

Removing all Thermococci sequences reveals that some features learnt on other classes of Euryarchaeota allow approximately half of the Thermococci sequences to be correctly identified. Repeating this with other species would allow a comparison to determine whether this is high or low, but such analysis was not performed here. Adding sequences back in one species at a time reveals that further features are learnt and increase the accuracy of classification. However, when looking at the Thermococci species trained on, the models were significantly less accurate on predicting reads of *Pyrococcus abyssi GE5.* As the model has already seen these sequences, this suggests something unusual about the sequences. *Pyrococcus abyssi GE5* is a barophile, which does result in differences in amino acid substitutions in orthologous genes when compared to non-barophilic species of the same genus (Di Giulio 2005). It may be possible that this could result in sequence differences that cannot be resolved without increasing loss elsewhere in the models, and hence poor performance on a model species already trained on. The species of Thermococci that were predicted most accurately from the test all belonged to the genus Thermococcus. *Pyrococcus furiosus* was present in the test set also, and is a non-barophilic organism. *Palaeococcus pacificus* was also in the test set, but had no species of the same genus in the test set. These two species had the lowest accuracy predictions. This may suggest that the organisms belonging to Thermococci are relatively diverse. Even within *Palaeococcus*, *Palaeococcus pacificus* is an obligate chemoorganoheterotroph and a strict anaerobe dependent on sulfur (X. Zeng et al. 2013), whereas *Palaeococcus helgesonii* is capable of both microaerobic and anaerobic growth (Amend et al. 2003). With such a diversity of metabolisms and so few species available to train on, it's possible that certain clades of organisms require more data to train on before models can generalise properly. Regardless, further work done should note whether performance is similarly poor on this species, if using different architectures and hyperparameters as previously suggested.

The mean MCC scores of the errors show that the majority of errors present in Euryarchaeota offer random or weak disagreement in prediction of each other. This suggests that by using an ensemble of models it would be possible to improve performance. Since the models have differing errors, likely as a result of the weight initialization, using a large enough ensemble would minimize the effect of these random errors. Not only this, but the regions where many or all of the models produce errors could be investigated and provide insight into why the models are not able to classify these sequences correctly. This would aid future experimental designs and could uncover interesting features.

Limitations

The methods described in this study are useful to a limited extent at comparing the four and five classes. There are a few caveats and limitations that should be addressed directly and considered before they can be implemented as useful tools or pipelines.

The first is something that has been touched on previously. Archaea are a relatively new domain, and new denominations are being created within it. Given the lack of sequencing data for certain clades, it is possible that further shifts are likely to occur. Thaumarchaeota

became a distinct phylum in 2008 (Brochier-Armanet et al. 2008), and the superphylum Asgardarchaeota was proposed in 2017 (Zaremba-Niedzwiedzka et al. 2017). Their discovery currently seems to support the idea that Eukaryotes emerged from within archaea (MacLeod et al. 2019). It is possible the models were therefore trained on species that belonged to the wrong phyla, or were put into the wrong class, which could affect model performance and what features are learnt. The sequences were also taken from GenBank, and as some of the new species used were recovered from metagenomes, there may be certain errors present.

Leading on from this, the models used only consider four classes. This is okay when using the models to compare the four classes, but a severe limitation in leveraging the work to be useful in identification tasks. This is because it would treat classification as a closed-set problem, whereas in reality it is an open-set recognition problem when also considering novel taxa (Scheirer et al. 2013). Therefore, using these models, sequences from other domains or phyla will be classified as one of the four classes, as rejection as unknown is not possible, which is incorrect. Increasing the number of classes requires sufficient examples for each class. This could be tailored to each application; for instance, for use in AD studies models could be trained on all taxa known to be involved in AD. However, this is still a closed-set problem, just with more classes. Therefore, in order to try an utilise the architecture, using a sigmoid function instead of softmax in the final layer and a series of one vs. all binary classification models would allow a probabilistic approach, with a cut-off probability needing to be determined by using out-of-distribution examples (such as unseen phyla).

Intra-class splitting is another method of achieving this, which involves splitting a class into typical and atypical examples, which allows for a tighter and closed decision boundary; the models in this paper use open decision boundaries, which is why the four classes occupy the whole problem space and rejection as unknown is not possible (Schlachter, Liao, and Yang 2019). However, determining atypical and typical examples would likely be very difficult to do manually, so the results from a closed-set model architecture such as the one used here could be used to inform this splitting. Not only this, but the models should also be able to identify lower-level taxonomy, such as class or genus, as phyla is too broad to be of much use. This would help with analysing novel species most.

Determining which features found by the neural network are useful is a difficult task. Many of the neural networks used in image-classification can be fooled by adversarial examples that exploit biases and introduce noise that changes the images imperceptibly to humans, but causes the network's accuracy to significantly drop (Hendrycks et al. 2019). This is due an overreliance on certain features which can cause a model to be less robust. The stresses a problem; the features learnt may be vulnerable to changes in input, such as strain differences, inversions, deletions, translocations and horizontal gene transfer events. There is also the problem of interpreting these features. Methods to do this, such as saliency maps, random and systematic perturbations can change feature importance (Ghorbani, Abid, and Zou 2019). Whether this will occur in such neural networks described or result from the previously mentioned genomic events is unclear, but should be investigated. This creates

another problem as human intuition cannot be applied to an image representation of DNA in the same way as it can to an image of an object.

Given the rapid advance of neural networks, and architectures such as Transformer, which has recently outperformed CNNs in image classification, there seems to be potential for greater performance than demonstrated here (Dosovitskiy et al. 2020). Whether representing DNA as an image is the most efficient representation is not clear, but it is likely that different representations have different virtues, as demonstrated by the performance of CGR over Reshape, but the ability for random 4-mer shuffling to be learnt by Reshape trained models.

Given certain problems that exist in metagenomics, using simulated and well studied data sets would be useful in seeing how the model not only handles other datasets, but would also provide an idea of how sequencing errors affect performance of the model. It could be that the use of a random set in the training data in this instance would cause the sequence to be misclassified if the errors are frequent enough. With current assembly and sequencing technology, having such an error rate seems unlikely, however. Computation time is also a concern within metagenomics, and given the use of CNNs and the need for data conversion, timing experiments would also be a good idea for future work.

Overall this study has helped compare the effectiveness of two representations of DNA for use in CNNs, and has shown that both are effective classifiers at the level of phyla, though less than simpler models using K-mers. However, Reshape representation offers the ability to learn features more complex than K-mers, which could be leveraged in tandem for classification. There remains much work to be done before such models are useful, such as introducing the ability for the model to reject examples as unknown. Given the progress in the field, it is likely that many new discoveries will inform and aid the improvement of this work.

References

Abe, Takashi, Shigehiko Kanaya, Makoto Kinouchi, Yuta Ichiba, Tokio Kozuki, and Toshimichi Ikemura. 2003. “Informatics for Unveiling Hidden Genome Signatures.” *Genome Research* 13 (4): 693–702.

Adam, Panagiotis S., Guillaume Borrel, Céline Brochier-Armanet, and Simonetta Gribaldo. 2017. “The Growing Tree of Archaea: New Perspectives on Their Diversity, Evolution and Ecology.” *The ISME Journal*. https://doi.org/10.1038/ismej.2017.122.

Aird, Daniel, Michael G. Ross, Wei-Sheng Chen, Maxwell Danielsson, Timothy Fennell, Carsten Russ, David B. Jaffe, Chad Nusbaum, and Andreas Gnirke. 2011. “Analyzing and Minimizing PCR Amplification Bias in Illumina Sequencing Libraries.” *Genome Biology* 12 (2): R18.

Al-Ajlan, A., and A. El Allali. 2019. “CNN-MGP: Convolutional Neural Networks for Metagenomics Gene Prediction.” *Interdisciplinary Sciences, Computational Life Sciences* 11 (4). https://doi.org/10.1007/s12539-018-0313-4.

Alneberg, Johannes, Brynjar Smári Bjarnason, Ino de Bruijn, Melanie Schirmer, Joshua Quick, Umer Z. Ijaz, Leo Lahti, Nicholas J. Loman, Anders F. Andersson, and Christopher Quince. 2014. “Binning Metagenomic Contigs by Coverage and Composition.” *Nature Methods* 11 (11): 1144–46.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. “Basic Local Alignment Search Tool.” *Journal of Molecular Biology* 215 (3): 403–10.

Alvarez, René, Saul Villca, and Gunnar Lidén. 2006. “Biogas Production from Llama and Cow Manure at High Altitude.” *Biomass and Bioenergy*. https://doi.org/10.1016/j.biombioe.2005.10.001.

Amarasinghe, Shanika L., Shian Su, Xueyi Dong, Luke Zappia, Matthew E. Ritchie, and Quentin Gouil. 2020. “Opportunities and Challenges in Long-Read Sequencing Data Analysis.” *Genome Biology* 21 (1): 1–16.

Amend, Jan P., D’arcy R. Meyer-Dombard, Seema N. Sheth, Natalya Zolotova, and Andrea C. Amend. 2003. “Palaeococcus Helgesonii Sp. Nov., a Facultatively Anaerobic, Hyperthermophilic Archaeon from a Geothermal Well on Vulcano Island, Italy.” *Archives of Microbiology*. https://doi.org/10.1007/s00203-003-0542-7.

Angelidaki, Irini, Dimitar Karakashev, Damien J. Batstone, Caroline M. Plugge, and Alfons J. M. Stams. 2011. “Biomethanation and Its Potential.” *Methods in Methane Metabolism, Part A*. https://doi.org/10.1016/b978-0-12-385112-3.00016-0.

Ayling, Martin, Matthew D. Clark, and Richard M. Leggett. 2019. “New Approaches for Metagenome Assembly with Short Reads.” *Briefings in Bioinformatics* 21 (2): 584–94.

Berghuis, Bojk A., Feiqiao Brian Yu, Frederik Schulz, Paul C. Blainey, Tanja Woyke, and Stephen R. Quake. 2018. “Hydrogenotrophic Methanogenesis in Archaeal Phylum Verstraetearchaeota Reveals the Shared Ancestry of All Methanogens.” https://doi.org/10.1101/391417.

Bertrand, Denis, Jim Shaw, Manesh Kalathiyappan, Amanda Hui Qi Ng, M. Senthil Kumar, Chenhao Li, Mirta Dvornicic, et al. 2019. “Hybrid Metagenomic Assembly Enables High-Resolution Analysis of Resistance Determinants and Mobile Elements in Human Microbiomes.” *Nature Biotechnology* 37 (8): 937–44.

Bohlin, Jon, Lars Snipen, Simon P. Hardy, Anja B. Kristoffersen, Karin Lagesen, Torunn Dønsvik, Eystein Skjerve, and David W. Ussery. 2010. “Analysis of Intra-Genomic GC

Content Homogeneity within Prokaryotes." *BMC Genomics* 11 (August): 464.
Bonham-Carter, Oliver, Joe Steele, and Dhundy Bastola. 2014. "Alignment-Free Genetic Sequence Comparisons: A Review of Recent Approaches by Word Analysis." *Briefings in Bioinformatics* 15 (6): 890–905.
Borrel, Guillaume, Nicolas Parisot, Hugh M. B. Harris, Eric Peyretaillade, Nadia Gaci, William Tottey, Olivier Bardot, et al. 2014. "Comparative Genomics Highlights the Unique Biology of Methanomassiliicoccales, a Thermoplasmatales-Related Seventh Order of Methanogenic Archaea That Encodes Pyrrolysine." *BMC Genomics* 15 (August): 679.
Boureau, Y-Lan, Jean Ponce, and Yann LeCun. 2010. "A Theoretical Analysis of Feature Pooling in Visual Recognition." In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 111–18. ICML'10. Madison, WI, USA: Omnipress.
Brochier-Armanet, Céline, Bastien Boussau, Simonetta Gribaldo, and Patrick Forterre. 2008. "Mesophilic Crenarchaeota: Proposal for a Third Archaeal Phylum, the Thaumarchaeota." *Nature Reviews. Microbiology* 6 (3): 245–52.
Brown, C. Titus, Adina Howe, Qingpeng Zhang, Alexis B. Pyrkosz, and Timothy H. Brom. 2012. "A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data." http://arxiv.org/abs/1203.4802.
Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. "Language Models Are Few-Shot Learners." http://arxiv.org/abs/2005.14165.
Burge, C., A. M. Campbell, and S. Karlin. 1992. "Over- and under-Representation of Short Oligonucleotides in DNA Sequences." *Proceedings of the National Academy of Sciences of the United States of America* 89 (4): 1358–62.
Busia, Akosua, George E. Dahl, Clara Fannjiang, David H. Alexander, Elizabeth Dorfman, Ryan Poplin, Cory Y. McLean, Pi-Chuan Chang, and Mark DePristo. 2019. "A Deep Learning Approach to Pattern Recognition for Short DNA Sequences." *bioRxiv*. https://doi.org/10.1101/353474.
Bzdok, Danilo. 2017. "Classical Statistics and Statistical Learning in Imaging Neuroscience." *Frontiers in Neuroscience* 11 (October): 543.
Bzdok, Danilo, Naomi Altman, and Martin Krzywinski. 2018. "Statistics versus Machine Learning." *Nature Methods*. https://doi.org/10.1038/nmeth.4642.
Cahill, Matt J., Claudio U. Köser, Nicholas E. Ross, and John A. C. Archer. 2010. "Read Length and Repeat Resolution: Exploring Prokaryote Genomes Using next-Generation Sequencing Technologies." *PloS One* 5 (7): e11518.
Campanaro, Stefano, Laura Treu, Panagiotis G. Kougias, Xinyu Zhu, and Irini Angelidaki. 2018. "Taxonomy of Anaerobic Digestion Microbiome Reveals Biases Associated with the Applied High Throughput Sequencing Strategies." *Scientific Reports* 8 (1): 1926.
Castelle, Cindy J., and Jillian F. Banfield. 2018. "Major New Microbial Groups Expand Diversity and Alter Our Understanding of the Tree of Life." *Cell* 172 (6): 1181–97.
Chiang, Hsin-I, Jian-Rong Li, Chun-Chi Liu, Po-Yu Liu, Hsin-Hua Chen, Yi-Ming Chen, Joung-Liang Lan, and Der-Yuan Chen. 2019. "An Association of Gut Microbiota with Different Phenotypes in Chinese Patients with Rheumatoid Arthritis." *Journal of Clinical Medicine Research* 8 (11). https://doi.org/10.3390/jcm8111770.
Chikhi, Rayan, and Guillaume Rizk. 2012. "Space-Efficient and Exact de Bruijn Graph Representation Based on a Bloom Filter." *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-642-33122-0_19.
C. Jay Kuo, C. 2016. "Understanding Convolutional Neural Networks with A Mathematical Model." *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1609.04112.
Cubonová, L 'ubomíra, Kathleen Sandman, Steven J. Hallam, Edward F. Delong, and John N. Reeve. 2005. "Histones in Crenarchaea." *Journal of Bacteriology* 187 (15): 5482–85.

Dahl, G. E., Dong Yu, Li Deng, and A. Acero. 2012. “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition.” *IEEE Transactions on Audio, Speech, and Language Processing*. https://doi.org/10.1109/tasl.2011.2134090.

Demain, Arnold L. 2014. “Importance of Microbial Natural Products and the Need to Revitalize Their Discovery.” *Journal of Industrial Microbiology & Biotechnology* 41 (2): 185–201.

Deschavanne, P. J., A. Giron, J. Vilain, G. Fagot, and B. Fertil. 1999. “Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences.” *Molecular Biology and Evolution*. https://doi.org/10.1093/oxfordjournals.molbev.a026048.

Dick, Gregory J., Anders F. Andersson, Brett J. Baker, Sheri L. Simmons, Brian C. Thomas, A. Pepper Yelton, and Jillian F. Banfield. 2009. “Community-Wide Analysis of Microbial Genome Sequence Signatures.” *Genome Biology* 10 (8): R85.

Di Giulio, Massimo. 2005. “A Comparison of Proteins from Pyrococcus Furiosus and Pyrococcus Abyssi: Barophily in the Physicochemical Properties of Amino Acids and in the Genetic Code.” *Gene* 346 (February): 1–6.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. 2020. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.” http://arxiv.org/abs/2010.11929.

Driessen, Rosalie P. C., He Meng, Gorle Suresh, Rajesh Shahapure, Giovanni Lanzani, U. Deva Priyakumar, Malcolm F. White, Helmut Schiessel, John van Noort, and Remus Th Dame. 2013. “Crenarchaeal Chromatin Proteins Cren7 and Sul7 Compact DNA by Inducing Rigid Bends.” *Nucleic Acids Research* 41 (1): 196–205.

Eloe-Fadrosh, Emiley A., Natalia N. Ivanova, Tanja Woyke, and Nikos C. Kyrpides. 2016. “Metagenomics Uncovers Gaps in Amplicon-Based Detection of Microbial Diversity.” *Nature Microbiology* 1 (February): 15032.

Engelen, Jesper E. van, and Holger H. Hoos. 2020. “A Survey on Semi-Supervised Learning.” *Machine Learning*. https://doi.org/10.1007/s10994-019-05855-6.

Evans, Paul N., Donovan H. Parks, Grayson L. Chadwick, Steven J. Robbins, Victoria J. Orphan, Suzanne D. Golding, and Gene W. Tyson. 2015. “Methane Metabolism in the Archaeal Phylum Bathyarchaeota Revealed by Genome-Centric Metagenomics.” *Science* 350 (6259): 434–38.

Foerstner, Konrad U., Christian von Mering, Sean D. Hooper, and Peer Bork. 2005. “Environments Shape the Nucleotide Composition of Genomes.” *EMBO Reports* 6 (12): 1208–13.

Fox, G., E. Stackebrandt, R. Hespell, J. Gibson, J. Maniloff, T. Dyer, R. Wolfe, et al. 1980. “The Phylogeny of Prokaryotes.” *Science*. https://doi.org/10.1126/science.6771870.

Fresia, Pablo, Verónica Antelo, Cecilia Salazar, Matías Giménez, Bruno D’Alessandro, Ebrahim Afshinnekoo, Christopher Mason, Gastón H. Gonnet, and Gregorio Iraola. 2019. “Urban Metagenomics Uncover Antibiotic Resistance Reservoirs in Coastal Beach and Sewage Waters.” *Microbiome* 7 (1): 35.

Gallant, John, David Maier, and James Astorer. 1980. “On Finding Minimal Length Superstrings.” *Journal of Computer and System Sciences*. https://doi.org/10.1016/0022-0000(80)90004-5.

Galtier, Nicolas. 2010. “Faculty Opinions Recommendation of Evidence of Selection upon Genomic GC-Content in Bacteria.” *Faculty Opinions – Post-Publication Peer Review of the Biomedical Literature*. https://doi.org/10.3410/f.5579957.5546056.

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2018. “ImageNet-Trained CNNs Are Biased towards Texture; Increasing Shape Bias Improves Accuracy and Robustness.” *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1811.12231.

Gelfand, M. S., C. G. Kozhukhin, and P. A. Pevzner. 1992. “Extendable Words in Nucleotide Sequences.” *Computer Applications in the Biosciences: CABIOS* 8 (2): 129–35.
Gholamalinezhad, Hossein, and Hossein Khosravi. 2020. “Pooling Methods in Deep Neural Networks, a Review.” *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/2009.07485.
Ghorbani, Amirata, Abubakar Abid, and James Zou. 2019. “Interpretation of Neural Networks Is Fragile.” *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v33i01.33013681.
Goldman, N. 1993. “Nucleotide, Dinucleotide and Trinucleotide Frequencies Explain Patterns Observed in Chaos Game Representations of DNA Sequences.” *Nucleic Acids Research* 21 (10): 2487–91.
Gong, M. L., N. Q. Ren, and D. F. Xing. 2005. “Start-up of Bio-Hydrogen Production Reactor Seeded with Sewage Sludge and Its Microbial Community Analysis.” *Water Science and Technology: A Journal of the International Association on Water Pollution Research* 52 (1-2): 115–21.
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. The MIT Press.
Grosso, Felicia, Paola Iovieno, Anna Alfani, and Flavia De Nicola. 2018. “Structure and Activity of Soil Microbial Communities in Three Mediterranean Forests.” *Applied Soil Ecology*. https://doi.org/10.1016/j.apsoil.2018.07.007.
Hao Zheng, Hongwei Wu. 2010. “Gene-Centric Association Analysis for the Correlation between the Guanine-Cytosine Content Levels and Temperature Range Conditions of Prokaryotic Species.” *BMC Bioinformatics* 11 (Suppl 11): S7.
Hendrycks, Dan, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2019. “Natural Adversarial Examples.” *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1907.07174.
Henneman, Bram, Clara van Emmerik, Hugo van Ingen, and Remus T. Dame. 2018. “Structure and Function of Archaeal Histones.” *PLoS Genetics* 14 (9): e1007582.
He, Yan, J. Gregory Caporaso, Xiao-Tao Jiang, Hua-Fang Sheng, Susan M. Huse, Jai Ram Rideout, Robert C. Edgar, et al. 2015. “Stability of Operational Taxonomic Units: An Important but Neglected Property for Analyzing Microbial Diversity.” *Microbiome* 3 (May): 20.
Hildebrand, Falk, Axel Meyer, and Adam Eyre-Walker. 2010. “Evidence of Selection upon Genomic GC-Content in Bacteria.” *PLoS Genetics*. https://doi.org/10.1371/journal.pgen.1001107.
Hill, K. A., N. J. Schisler, and S. M. Singh. 1992. “Chaos Game Representation of Coding Regions of Human Globin Genes and Alcohol Dehydrogenase Genes of Phylogenetically Divergent Species.” *Journal of Molecular Evolution* 35 (3): 261–69.
Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. “Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors.” http://arxiv.org/abs/1207.0580.
Hochreiter, Sepp. 1998. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions.” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. https://doi.org/10.1142/s0218488598000094.
Hofmeyr, Steven, Rob Egan, Evangelos Georganas, Alex C. Copeland, Robert Riley, Alicia Clum, Emiley Eloe-Fadrosh, et al. 2020. “Terabase-Scale Metagenome Coassembly with MetaHipMer.” *Scientific Reports* 10 (1): 10689.
Hulten, Geoff, Laurie Spencer, and Pedro Domingos. 2001. “Mining Time-Changing Data Streams.” In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 97–106. KDD ’01. New York, NY, USA: Association for Computing Machinery.
Huptas, Christopher, Siegfried Scherer, and Mareike Wenning. 2016. “Optimized Illumina PCR-Free Library Preparation for Bacterial Whole Genome Sequencing and Analysis of Factors Influencing de Novo Assembly.” *BMC Research Notes* 9 (May): 269.

Hutter, Frank, Jörg Lücke, and Lars Schmidt-Thieme. 2015. “Beyond Manual Tuning of Hyperparameters.” *KI - Künstliche Intelligenz*. https://doi.org/10.1007/s13218-015-0381-0.
Ioffe, Sergey, and Christian Szegedy. 2015. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1502.03167.
Jaganathan, Kishore, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F. McRae, Siavash Fazel Darbandi, David Knowles, Yang I. Li, Jack A. Kosmicki, et al. 2019. “Predicting Splicing from Primary Sequence with Deep Learning.” *Cell* 176 (3): 535–48.e24.
Jain, Miten, Sergey Koren, Karen H. Miga, Josh Quick, Arthur C. Rand, Thomas A. Sasani, John R. Tyson, et al. 2018. “Nanopore Sequencing and Assembly of a Human Genome with Ultra-Long Reads.” *Nature Biotechnology* 36 (4): 338–45.
Jeffrey, H. J. 1990. “Chaos Game Representation of Gene Structure.” *Nucleic Acids Research* 18 (8): 2163–70.
Jiang, Minghui, James Anderson, Joel Gillespie, and Martin Mayne. 2008. “uShuffle: A Useful Tool for Shuffling Biological Sequences While Preserving the K-Let Counts.” *BMC Bioinformatics* 9 (1): 1–11.
Jurman, Giuseppe, Samantha Riccadonna, and Cesare Furlanello. 2012. “A Comparison of MCC and CEN Error Measures in Multi-Class Prediction.” *PloS One* 7 (8): e41882.
Kandel, D., Y. Matias, R. Unger, and P. Winkler. 1996. “Shuffling Biological Sequences.” *Discrete Applied Mathematics* 71 (1-3): 171–85.
Kapun, Evgeny, and Fedor Tsarev. 2013. “De Bruijn Superwalk with Multiplicities Problem Is NP-Hard.” *BMC Bioinformatics* 14 (5): 1–4.
Karlin, S., and C. Burge. 1995. “Dinucleotide Relative Abundance Extremes: A Genomic Signature.” *Trends in Genetics: TIG* 11 (7): 283–90.
Karlin, S., and I. Ladunga. 1994. “Comparisons of Eukaryotic Genomic Sequences.” *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.91.26.12832.
Karlin, S., J. Mrázek, and A. M. Campbell. 1997. “Compositional Biases of Bacterial Genomes and Evolutionary Implications.” *Journal of Bacteriology* 179 (12): 3899–3913.
Karlin, S., A. M. Campbell, and J. Mrázek,. 1998. “Comparative DNA Analysis Across Diverse Genomes.” *Annual Review of Genetics* 32 (1): 185–225.
Kaushal, Girija, Jitendra Kumar, Rajender S. Sangwan, and Sudhir P. Singh. 2018. “Metagenomic Analysis of Geothermal Water Reservoir Sites Exploring Carbohydrate-Related Thermozymes.” *International Journal of Biological Macromolecules* 119 (November): 882–95.
Khan, Abdul Rafay, Muhammad Tariq Pervez, Masroor Ellahi Babar, Nasir Naveed, and Muhammad Shoaib. 2018. “A Comprehensive Study of De Novo Genome Assemblers: Current Challenges and Future Prospective.” *Evolutionary Bioinformatics Online* 14 (February): 1176934318758650.
Kingma, Diederik P., and Jimmy Ba. 2014. “Adam: A Method for Stochastic Optimization.” http://arxiv.org/abs/1412.6980.
Kleftogiannis, Dimitrios, Panos Kalnis, and Vladimir B. Bajic. 2013. “Comparing Memory-Efficient Genome Assemblers on Stand-Alone and Cloud Infrastructures.” *PloS One* 8 (9). https://doi.org/10.1371/journal.pone.0075505.
Krehenwinkel, Henrik, Madeline Wolf, Jun Ying Lim, Andrew J. Rominger, Warren B. Simison, and Rosemary G. Gillespie. 2017. “Estimating and Mitigating Amplification Bias in Qualitative and Quantitative Arthropod Metabarcoding.” *Scientific Reports* 7 (1): 17668.
Kriegeskorte, Nikolaus. 2015. “Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing.” *Annual Review of Vision Science* 1

(1): 417–46.
Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. “ImageNet Classification with Deep Convolutional Neural Networks.” *Communications of the ACM*. https://doi.org/10.1145/3065386.
Krogh, Anders, and John A. Hertz. 1991. “A Simple Weight Decay Can Improve Generalization.” In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, 950–57. NIPS’91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
Kumar, Jitesh, Nitish Sharma, Girija Kaushal, Sanjukta Samurailatpam, Dinabandhu Sahoo, Amit K. Rai, and Sudhir P. Singh. 2019. “Metagenomic Insights Into the Taxonomic and Functional Features of Kinema, a Traditional Fermented Soybean Product of Sikkim Himalaya.” *Frontiers in Microbiology*. https://doi.org/10.3389/fmicb.2019.01744.
Kumar, Vishal, Arun Kumar Dangi, and Pratyoosh Shukla. 2018. “Engineering Thermostable Microbial Xylanases Toward Its Industrial Applications.” *Molecular Biotechnology* 60 (3): 226–35.
Lane, D. J., B. Pace, G. J. Olsen, D. A. Stahl, M. L. Sogin, and N. R. Pace. 1985. “Rapid Determination of 16S Ribosomal RNA Sequences for Phylogenetic Analyses.” *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.82.20.6955.
Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. “Gradient-Based Learning Applied to Document Recognition.” *Proceedings of the IEEE*. https://doi.org/10.1109/5.726791.
Lee, Hyun Sook, Sung Gyun Kang, Seung Seob Bae, Jae Kyu Lim, Yona Cho, Yun Jae Kim, Jeong Ho Jeon, et al. 2008. “The Complete Genome Sequence of Thermococcus Onnurineus NA1 Reveals a Mixed Heterotrophic and Carboxydotrophic Metabolism.” *Journal of Bacteriology* 190 (22): 7491–99.
Le, Nguyen Quoc Khanh, and Van-Nui Nguyen. 2019. “SNARE-CNN: A 2D Convolutional Neural Network Architecture to Identify SNARE Proteins from High-Throughput Sequencing Data.” *PeerJ Computer Science*. https://doi.org/10.7717/peerj-cs.177.
Lever, Jake, Martin Krzywinski, and Naomi Altman. 2016a. “Model Selection and Overfitting.” *Nature Methods*. https://doi.org/10.1038/nmeth.3968. 2016b. “Classification Evaluation.” *Nature Methods* 13 (8): 603–4.
Liang, Qiaoxing, Paul W. Bible, Yu Liu, Bin Zou, and Lai Wei. 2020. “DeepMicrobes: Taxonomic Classification for Metagenomics with Deep Learning.” *NAR Genomics and Bioinformatics* 2 (1). https://doi.org/10.1093/nargab/lqaa009.
Linsley, Jeremy W., Drew A. Linsley, Josh Lamstein, Gennadi Ryan, Kevan Shah, Nicholas A. Castello, Viral Oza, et al. 2020. “Super-Human Cell Death Detection with Biomarker-Optimized Neural Networks.” *Cold Spring Harbor Laboratory*. https://doi.org/10.1101/2020.08.04.237032.
Liu, Y. 2010. “Taxonomy of Methanogens.” *Handbook of Hydrocarbon and Lipid Microbiology*. https://doi.org/10.1007/978-3-540-77587-4_42.
Liu, Yuchen, and William B. Whitman. 2008. “Metabolic, Phylogenetic, and Ecological Diversity of the Methanogenic Archaea.” *Annals of the New York Academy of Sciences*. https://doi.org/10.1196/annals.1419.019.
Liu, Zongzhi, Catherine Lozupone, Micah Hamady, Frederic D. Bushman, and Rob Knight. 2007. “Short Pyrosequencing Reads Suffice for Accurate Microbial Community Analysis.” *Nucleic Acids Research* 35 (18): e120.
Loshchilov, Ilya, and Frank Hutter. 2017. “Decoupled Weight Decay Regularization.” http://arxiv.org/abs/1711.05101.
Lugli, Gabriele Andrea, Christian Milani, Leonardo Mancabelli, Francesca Turroni, Douwe van Sinderen, and Marco Ventura. 2019. “A Microbiome Reality Check: Limitations of in Silico-Based Metagenomic Approaches to Study Complex Bacterial Communities.”

Environmental Microbiology Reports 11 (6): 840–47.
Lu, Jennifer, Florian P. Breitwieser, Peter Thielen, and Steven L. Salzberg. 2017. “Bracken: Estimating Species Abundance in Metagenomics Data.” *PeerJ Computer Science*. https://doi.org/10.7717/peerj-cs.104.
MacLeod, Fraser, 1 School of Biotechnology and Biomolecular Sciences, The University of New South Wales, Sydney, Australia, Gareth S. Kindler, Hon Lun Wong, et al. 2019. “Asgard Archaea: Diversity, Function, and Evolutionary Implications in a Range of Microbiomes.” *AIMS Microbiology*. https://doi.org/10.3934/microbiol.2019.1.48.
McCulloch, W. S., and W. Pitts. 1990. “A Logical Calculus of the Ideas Immanent in Nervous Activity. 1943.” *Bulletin of Mathematical Biology* 52 (1-2): 99–115; discussion 73–97.
Medvedev, Paul, Konstantinos Georgiou, Gene Myers, and Michael Brudno. 2007. “Computability of Models for Sequence Assembly.” *Algorithms in Bioinformatics. WABI 2007. Lecture Notes in Computer Science, Vol 4645. Springer, Berlin, Heidelberg.* https://doi.org/10.1007/978-3-540-74126-8_27.
Milanese, Alessio, Daniel R. Mende, Lucas Paoli, Guillem Salazar, Hans-Joachim Ruscheweyh, Miguelangel Cuenca, Pascal Hingamp, et al. 2019. “Microbial Abundance, Activity and Population Genomic Profiling with mOTUs2.” *Nature Communications* 10 (1): 1014.
Mollet, C., M. Drancourt, and D. Raoult. 1997. “rpoB Sequence Analysis as a Novel Basis for Bacterial Identification.” *Molecular Microbiology* 26 (5): 1005–11.
Nagaoka, Eriko, Ryota Hidese, Tadayuki Imanaka, and Shinsuke Fujiwara. 2013. “Importance and Determinants of Induction of Cold-Induced DEAD RNA Helicase in the Hyperthermophilic Archaeon Thermococcus Kodakarensis.” *Journal of Bacteriology* 195 (15): 3442–50.
Nielsen, H. Bjørn, Mathieu Almeida, Agnieszka Sierakowska Juncker, Simon Rasmussen, Junhua Li, Shinichi Sunagawa, Damian R. Plichta, et al. 2014. “Identification and Assembly of Genomes and Genetic Elements in Complex Metagenomic Samples without Using Reference Genomes.” *Nature Biotechnology* 32 (8): 822–28.
Nielsen, M. A. 2015. *Neural Networks and Deep Learning*. Determination Press.
Nobu, Masaru Konishi, Takashi Narihiro, Kyohei Kuroda, Ran Mei, and Wen-Tso Liu. 2016. “Chasing the Elusive Euryarchaeota Class WSA2: Genomes Reveal a Uniquely Fastidious Methyl-Reducing Methanogen.” *The ISME Journal* 10 (10): 2478–87.
Ofek-Lalzar, Maya, Noa Sela, Milana Goldman-Voronov, Stefan J. Green, Yitzhak Hadar, and Dror Minz. 2014. “Niche and Host-Associated Functional Signatures of the Root Surface Microbiome.” *Nature Communications* 5 (September): 4950.
Paolini, Valerio, Francesco Petracchini, Marco Segreto, Laura Tomassetti, Nour Naja, and Angelo Cecinato. 2018. “Environmental Impact of Biogas: A Short Review of Current Knowledge.” *Journal of Environmental Science and Health, Part A*. https://doi.org/10.1080/10934529.2018.1459076.
Payne, Alexander, Nadine Holmes, Vardhman Rakyan, and Matthew Loose. 2019. “BulkVis: A Graphical Viewer for Oxford Nanopore Bulk FAST5 Files.” *Bioinformatics*. https://doi.org/10.1093/bioinformatics/bty841.
Pfeiffer, Franziska, Carsten Gröber, Michael Blank, Kristian Händler, Marc Beyer, Joachim L. Schultze, and Günter Mayer. 2018. “Systematic Evaluation of Error Rates and Causes in Short Samples in next-Generation Sequencing.” *Scientific Reports* 8 (1): 1–14.
Pride, David T., Richard J. Meinersmann, Trudy M. Wassenaar, and Martin J. Blaser. 2003. “Evolutionary Implications of Microbial Genome Tetranucleotide Frequency Biases.” *Genome Research* 13 (2): 145–58.
Propp, James Gary, and David Bruce Wilson. 1998. “How to Get a Perfectly Random Sample from a Generic Markov Chain and Generate a Random Spanning Tree of a Directed Graph.” *Journal of Algorithms & Computational Technology* 27 (2): 170–217.

Radiuk, Pavlo. 2017. “Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets.” *Information Technology and Management Science* 20 (December). https://doi.org/10.1515/itms-2017-0003.

Ranzato, Marc\textquotesingle Aurelio, Y-Lan Boureau, and Yann Cun. 2008. “Sparse Feature Learning for Deep Belief Networks.” In *Advances in Neural Information Processing Systems*, edited by J. Platt, D. Koller, Y. Singer, and S. Roweis, 20:1185–92. Curran Associates, Inc.

Rao, M. S., S. P. Singh, A. K. Singh, and M. S. Sodha. 2000. “Bioenergy Conversion Studies of the Organic Fraction of MSW: Assessment of Ultimate Bioenergy Production Potential of Municipal Garbage.” *Applied Energy*. https://doi.org/10.1016/s0306-2619(99)00056-2.

Román, Sara, Rüdiger Ortiz-Álvarez, Chiara Romano, Emilio O. Casamayor, and Daniel Martin. 2019. “Microbial Community Structure and Functionality in the Deep Sea Floor: Evaluating the Causes of Spatial Heterogeneity in a Submarine Canyon System (NW Mediterranean, Spain).” *Frontiers in Marine Science*. https://doi.org/10.3389/fmars.2019.00108.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. “Learning Internal Representations by Error Propagation.” In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, 348–52. Cambridge, MA, USA: MIT Press.

Saini, R., S. S. Kanwar, O. P. Sharma, and M. K. Gupta. 2003. “Biomethanation of Lantana Weed and Biotransformation of Its Toxins.” *World Journal of Microbiology & Biotechnology* 19 (2): 209–13.

Sandberg, Rickard, Carl-Ivar Bränden, Ingemar Ernberg, and Joakim Cöster. 2003. “Quantifying the Species-Specificity in Genomic Signatures, Synonymous Codon Choice, Amino Acid Usage and G+C Content.” *Gene* 311 (June): 35–42.

Sandberg, Rickard, Gösta Winberg, Carl-Ivar Bränden, Alexander Kaske, Ingemar Ernberg, and Joakim Cöster. 2001. “Capturing Whole-Genome Characteristics in Short Sequences Using a Naïve Bayesian Classifier.” *Genome Research* 11 (8): 1404–9.

Sangwan, Naseer, Carey Lambert, Anukriti Sharma, Vipin Gupta, Paramjit Khurana, Jitendra P. Khurana, R. Elizabeth Sockett, Jack A. Gilbert, and Rup Lal. 2015. “Arsenic Rich Himalayan Hot Spring Metagenomics Reveal Genetically Novel Predator-Prey Genotypes.” *Environmental Microbiology Reports* 7 (6): 812–23.

Santurkar, Shibani, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2019. “How Does Batch Normalization Help Optimization?” *arXiv [stat.ML]*. arXiv. http://arxiv.org/abs/1805.11604.

Schbath, S., B. Prum, and E. de Turckheim. 1995. “Exceptional Motifs in Different Markov Chain Models for a Statistical Analysis of DNA Sequences.” *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology* 2 (3): 417–37.

Scheirer, Walter J., Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. 2013. “Toward Open Set Recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7): 1757–72.

Schlachter, Patrick, Yiwen Liao, and Bin Yang. 2019. “Deep One-Class Classification Using Intra-Class Splitting.” *2019 IEEE Data Science Workshop (DSW)*. https://doi.org/10.1109/dsw.2019.8755576.

Schmidhuber, Jürgen. 2015. “Deep Learning in Neural Networks: An Overview.” *Neural Networks: The Official Journal of the International Neural Network Society* 61 (January): 85–117.

Schwarze, Katharina, James Buchanan, Jilles M. Fermont, Helene Dreau, Mark W. Tilley, John M. Taylor, Pavlos Antoniou, et al. 2019. “The Complete Costs of Genome Sequencing: A Microcosting Study in Cancer and Rare Diseases from a Single Center in the United Kingdom.” *Genetics in Medicine: Official Journal of the American College of*

Medical Genetics 22 (1): 85–94.
Sczyrba, Alexander, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, et al. 2017. “Critical Assessment of Metagenome Interpretation-a Benchmark of Metagenomics Software.” *Nature Methods* 14 (11): 1063–71.
Sedlar, Karel, Kristyna Kupkova, and Ivo Provaznik. 2017. “Bioinformatics Strategies for Taxonomy Independent Binning and Visualization of Sequences in Shotgun Metagenomics.” *Computational and Structural Biotechnology Journal* 15: 48–55.
Selling, Robert, Torbjörn Håkansson, and Lovisa Björnsson. 2008. “Two-Stage Anaerobic Digestion Enables Heavy Metal Removal.” *Water Science and Technology: A Journal of the International Association on Water Pollution Research* 57 (4): 553–58.
Sevigny, Joseph L., Derek Rothenheber, Krystalle Sharlyn Diaz, Ying Zhang, Kristin Agustsson, R. Daniel Bergeron, and W. Kelley Thomas. 2019. “Marker Genes as Predictors of Shared Genomic Function.” *BMC Genomics* 20 (1): 1–13.
Shanahan, Fergus, Douwe van Sinderen, Paul W. O’Toole, and Catherine Stanton. 2017. “Feeding the Microbiota: Transducer of Nutrient Signals for the Host.” *Gut* 66 (9): 1709–17.
Sierocinski, Pawel, Florian Bayer, Gabriel Yvon-Durocher, Melia Burdon, Tobias Großkopf, Mark Alston, David Swarbreck, Phil J. Hobbs, Orkun S. Soyer, and Angus Buckling. 2018. “Biodiversity-Function Relationships in Methanogenic Communities.” *Molecular Ecology* 27 (22): 4641–51.
Simonyan, Karen, and Andrew Zisserman. 2015. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1409.1556.
Singh, Ritambhara, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. 2016. “DeepChrome: Deep-Learning for Predicting Gene Expression from Histone Modifications.” *Bioinformatics* 32 (17): i639–48.
Sipos, Rita, Anna J. Székely, Márton Palatinszky, Sára Révész, Károly Márialigeti, and Marcell Nikolausz. 2007. “Effect of Primer Mismatch, Annealing Temperature and PCR Cycle Number on 16S rRNA Gene-Targetting Bacterial Community Analysis.” *FEMS Microbiology Ecology* 60 (2): 341–50.
Smith, Leslie N. 2018. “A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 Learning Rate, Batch Size, Momentum, and Weight Decay.” *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1803.09820.
Song, Kai, Jie Ren, and Fengzhu Sun. 2019. “Reads Binning Improves Alignment-Free Metagenome Comparison.” *Frontiers in Genetics* 10 (November): 1156.
Steen, Andrew D., Alexander Crits-Christoph, Paul Carini, Kristen M. DeAngelis, Noah Fierer, Karen G. Lloyd, and J. Cameron Thrash. 2019. “High Proportions of Bacteria and Archaea across Most Biomes Remain Uncultured.” *The ISME Journal*.
Stephens, Zachary D., Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson. 2015. “Big Data: Astronomical or Genomical?” *PLoS Biology* 13 (7): e1002195.
Sueoka, N. 1962. “On the Genetic Basis of Variation and Heterogeneity of DNA Base Composition.” *Proceedings of the National Academy of Sciences of the United States of America* 48 (April): 582–92.
Sundberg, Carina, Waleed A. Al-Soud, Madeleine Larsson, Erik Alm, Sepehr S. Yekta, Bo H. Svensson, Søren J. Sørensen, and Anna Karlsson. 2013. “454 Pyrosequencing Analyses of Bacterial and Archaeal Richness in 21 Full-Scale Biogas Digesters.” *FEMS Microbiology Ecology* 85 (3): 612–26.
Suryawanshi, P. C., A. B. Chaudhari, and R. M. Kothari. 2010. “Mesophilic Anaerobic Digestion: First Option for Waste Treatment in Tropical Regions.” *Critical Reviews in*

Biotechnology 30 (4): 259–82.
Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. “Sequence to Sequence Learning with Neural Networks.” http://arxiv.org/abs/1409.3215.
Tang, Jie, Keru Hua, Mengye Chen, Ruiming Zhang, and Xiaoli Xie. 2014. “A Novel K-Word Relative Measure for Sequence Comparison.” *Computational Biology and Chemistry* 53PB (December): 331–38.
Teeling, Hanno, Jost Waldmann, Thierry Lombardot, Margarete Bauer, and Frank Oliver Glöckner. 2004. “TETRA: A Web-Service and a Stand-Alone Program for the Analysis and Comparison of Tetranucleotide Usage Patterns in DNA Sequences.” *BMC Bioinformatics* 5 (October): 163.
Truong, Duy Tin, Eric A. Franzosa, Timothy L. Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. 2015. “MetaPhlAn2 for Enhanced Metagenomic Taxonomic Profiling.” *Nature Methods* 12 (10): 902–3.
Tyson, John R., Nigel J. O’Neil, Miten Jain, Hugh E. Olsen, Philip Hieter, and Terrance P. Snutch. 2018. “MinION-Based Long-Read Sequencing and Assembly Extends the Caenorhabditis Elegans Reference Genome.” *Genome Research*. https://doi.org/10.1101/gr.221184.117.
Van Rossum, Thea, Pamela Ferretti, Oleksandr M. Maistrenko, and Peer Bork. 2020. “Diversity within Species: Interpreting Strains in Microbiomes.” *Nature Reviews. Microbiology* 18 (9): 491–506.
Vanwonterghem, Inka, Paul N. Evans, Donovan H. Parks, Paul D. Jensen, Ben J. Woodcroft, Philip Hugenholtz, and Gene W. Tyson. 2016. “Methylotrophic Methanogenesis Discovered in the Archaeal Phylum Verstraetearchaeota.” *Nature Microbiology* 1 (October): 16170.
Vervier, Kévin, Pierre Mahé, Maud Tournoud, Jean-Baptiste Veyrieras, and Jean-Philippe Vert. 2015. “Large-Scale Machine Learning for Metagenomics Sequence Classification.” *Bioinformatics* 32 (7): 1023–32.
Vierke, Gudrun, Afra Engelmann, Carina Hebbeln, and Michael Thomm. 2003. “A Novel Archaeal Transcriptional Regulator of Heat Shock Response.” *The Journal of Biological Chemistry* 278 (1): 18–26.
Vu, Duong, Marizeth Groenewald, and Gerard Verkley. 2020. “Convolutional Neural Networks Improve Fungal Classification.” *Scientific Reports* 10 (1): 1–12.
Walt, Andries Johannes van der, Andries Johannes van der Walt, Marc Warwick van Goethem, Jean-Baptiste Ramond, Thulani Peter Makhalanyane, Oleg Reva, and Don Arthur Cowan. 2017. “Assembling Metagenomes, One Community at a Time.” *BMC Genomics*. https://doi.org/10.1186/s12864-017-3918-9.
Weinberg, Michael V., Gerrit J. Schut, Scott Brehm, Susmita Datta, and Michael W. W. Adams. 2005. “Cold Shock of a Hyperthermophilic Archaeon: Pyrococcus Furiosus Exhibits Multiple Responses to a Suboptimal Growth Temperature with a Key Role for Membrane-Bound Glycoproteins.” *Journal of Bacteriology* 187 (1): 336–48.
Wintsche, Babett, Nico Jehmlich, Denny Popp, Hauke Harms, and Sabine Kleinsteuber. 2018. “Metabolic Adaptation of Methanogens in Anaerobic Digesters Upon Trace Element Limitation.” *Frontiers in Microbiology* 9 (March): 405.
Woese, C. R., O. Kandler, and M. L. Wheelis. 1990. “Towards a Natural System of Organisms: Proposal for the Domains Archaea, Bacteria, and Eucarya.” *Proceedings of the National Academy of Sciences of the United States of America* 87 (12): 4576–79.
Wood, Derrick E., and Steven L. Salzberg. 2014. “Kraken: Ultrafast Metagenomic Sequence Classification Using Exact Alignments.” *Genome Biology* 15 (3): R46.
Wu, Hao, and Jonathan L. Shapiro. 2006. “Does Overfitting Affect Performance in Estimation of Distribution Algorithms.” *Proceedings of the 8th Annual Conference on Genetic and*

Evolutionary Computation - GECCO '06. https://doi.org/10.1145/1143997.1144078.
Wu, Tiee-Jian, Ying-Hsueh Huang, and Lung-An Li. 2005. "Optimal Word Sizes for Dissimilarity Measures and Estimation of the Degree of Dissimilarity between DNA Sequences." *Bioinformatics* 21 (22): 4125–32.
Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. 2018. "Convolutional Neural Networks: An Overview and Application in Radiology." *Insights into Imaging* 9 (4): 611.
Yi, Jing, Bin Dong, Jingwei Jin, and Xiaohu Dai. 2014. "Effect of Increasing Total Solids Contents on Anaerobic Digestion of Food Waste under Mesophilic Conditions: Performance and Microbial Characteristics Analysis." *PloS One* 9 (7): e102548.
Yin, Bojian, Marleen Balvert, Davide Zambrano, Alexander Schönhuth, and Sander Bohte. 2018. "An Image Representation Based Convolutional Network for DNA Classification." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1806.04931.
Zaremba-Niedzwiedzka, Katarzyna, Eva F. Caceres, Jimmy H. Saw, Disa Bäckström, Lina Juzokaite, Emmelien Vancaester, Kiley W. Seitz, et al. 2017. "Asgard Archaea Illuminate the Origin of Eukaryotic Cellular Complexity." *Nature* 541 (7637): 353–58.
Zeng, Wanwen, Yong Wang, and Rui Jiang. 2019. "Integrating Distal and Proximal Information to Predict Gene Expression via a Densely Connected Convolutional Neural Network." *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btz562.
Zeng, Xiang, Xiaobo Zhang, Lijing Jiang, Karine Alain, Mohamed Jebbar, and Zongze Shao. 2013. "Palaeococcus Pacificus Sp. Nov., an Archaeon from Deep-Sea Hydrothermal Sediment." *International Journal of Systematic and Evolutionary Microbiology* 63 (Pt 6): 2155–59.
Zhang, Jingxin, Liwei Mao, Le Zhang, Kai-Chee Loh, Yanjun Dai, and Yen Wah Tong. 2017. "Metagenomic Insight into the Microbial Networks and Metabolic Mechanism in Anaerobic Digesters for Food Waste by Incorporating Activated Carbon." *Scientific Reports* 7 (1): 11293.
Zhang, Xinxu, Wei Xu, Yang Liu, Mingwei Cai, Zhuhua Luo, and Meng Li. 2018. "Metagenomics Reveals Microbial Diversity and Metabolic Potentials of Seawater and Surface Sediment From a Hadal Biosphere at the Yap Trench." *Frontiers in Microbiology* 9 (October): 2402.
Zhong, Haohui, Laura Lehtovirta-Morley, Jiwen Liu, Yanfen Zheng, Heyu Lin, Delei Song, Jonathan D. Todd, Jiwei Tian, and Xiao-Hua Zhang. 2020. "Novel Insights into the Thaumarchaeota in the Deepest Oceans: Their Metabolism and Potential Adaptation Mechanisms." *Microbiome* 8 (1): 78.
Zhou, Fengfeng, Victor Olman, and Ying Xu. 2008. "Barcodes for Genomes and Applications." *BMC Bioinformatics* 9 (1): 1–11.
Zinder, Stephen H., and Markus Koch. 1984. "Non-Aceticlastic Methanogenesis from Acetate: Acetate Oxidation by a Thermophilic Syntrophic Coculture." *Archives of Microbiology*. https://doi.org/10.1007/bf00402133.