PROBABILISTIC MODELING FOR WHOLE METAGENOME PROFILING

David Burks, BSc

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

May 2021

APPROVED:

Rajeev K. Azad, Major Professor
Michael S. Allen, Committee Member
Mauricio S. Antunes, Committee Member
Pamela A. Padilla, Committee Member
Vladimir Shulaev, Committee Member
Jyoti Shah, Chair of the Department of Biological Sciences
Su Gao, Dean of the College of Science
Victor Prybutok, Dean of the Toulouse Graduate School

Burks, David. *Probabilistic Modeling for Whole Metagenome Profiling*. Doctor of Philosophy (Biology), May 2021, 115 pp., 5 tables, 14 figures, (9 supplementary tables and 26 supplementary figures in separate files), references, 125 titles.

To address the shortcomings in existing Markov model implementations in handling large amount of metagenomic data with comparable or better accuracy in classification, we developed a new algorithm based on pseudo-count supplemented standard Markov model (SMM), which leverages the power of higher order models to more robustly classify reads at different taxonomic levels. Assessment on simulated metagenomic datasets demonstrated that overall SMM was more accurate in classifying reads to their respective taxa at all ranks compared to the interpolated methods. Higher order SMMs (9th order or greater) also outperformed BLAST alignments in assigning taxonomic labels to metagenomic reads at different taxonomic ranks (genus and higher) on tests that masked the read originating species (genome models) in the database. Similar results were obtained by masking at other taxonomic ranks in order to simulate the plausible scenarios of non-representation of the source of a read at different taxonomic levels in the genome database. The performance gap became more pronounced with higher taxonomic levels. To eliminate contaminations in datasets and to further improve our alignment-free approach, we developed a new framework based on a genome segmentation and clustering algorithm. This framework allowed removal of adapter sequences and contaminant DNA, as well as generation of clusters of similar segments, which were then used to sample representative read fragments to constitute training datasets. The parameters of a logistic regression model were learnt from these training datasets using a Bayesian optimization procedure. This allowed us to establish thresholds for classifying metagenomic reads by SMM. This led to the development of a Python-based frontend that

combines our SMM algorithm with the logistic regression optimization, named POSMM (Python Optimized Standard Markov Model). POSMM provides a much-needed alternative to metagenome profiling programs. Our algorithm that builds the genome models on the fly, and thus obviates the need to build a database, complements alignment-based classification and can thus be used in concert with alignment-based classifiers to raise the bar in metagenome profiling.

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to everyone involved in my academic career. It can start as something as simple as being noticed, much in the way that Dr. Rebecca Dickstein noticed me hiding in the back of her biochemistry lectures. Four years later, that recognition would lead me back to UNT to begin the journey summarized in these pages. I do not have the words equivalent to my gratitude for the committee members giving me this opportunity. To Drs. Michael Allen, Vladimir Shulaev, and Mauricio Antunes, your mentorship and support is something I will take with me everywhere I go. To Dr. Pamela Padilla, I have felt like an honorary member of your lab, and I sincerely thank you for letting me contribute to the amazing work that you do. I am sure that with each defense the impact is softened, but I hope my committee members realize that they are helping me build a life I never thought possible. I am thankful to the University of North Texas, UNT College of Sciences, UNT Department of Biological Sciences, and the UNT Biodiscovery Institute for giving me the opportunity to teach, research, and contribute with the brilliant minds that have made these institutions their home. I am thankful to my beautiful and patient wife, who has pulled me out of the dark far more times than I can count. None of this is possible without Dr. Rajeev Azad. You are the best mentor that I could ever ask for. Because of you, I have become the scientist I am today, and I will do my best to follow in your footsteps. To my fellow Azad lab members, past and present, thank you for believing in me. I hope I can continue to make each of you proud.

To my mother and my brother, this is for us.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

DATA TABLES AND FIGURES INCLUDED AS SUPPLEMENTARY MATERIALS

Tables

Figures

maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented here as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level).

SUPPLEMENTARY FIG. 4.3 – Graphs illustrating the composition of each cluster generated by different clustering methods. The composition is visualized as the distribution of source genomes (E. coli K12 is recipient and rest are donors) in a cluster, in terms of fractions of a cluster assigned to different sources as well as fractions of source DNAs resident in a cluster (fractions shown in percentages). These cluster composition maps were generated at parametric settings detailed in Table 1 for segment-order (a), hierarchical (b), network (c), and affinity-propagation (d) clustering methods.

SUPPLEMENTARY FIG. 4.4 – Graphs illustrating the composition of each cluster generated by different clustering methods. The composition is visualized as the distribution of source genomes (E. coli K12 is recipient and rest are donors) in a cluster, in terms of fractions of a cluster assigned to different sources as well as fractions of source DNAs resident in a cluster (fractions shown in percentages). These cluster composition maps were generated at parametric settings detailed in Table 2 for segment-order (a), hierarchical (b), network (c), and affinity-propagation (d) clustering methods.

SUPPLEMENTARY FIG. 4.5 – Graphs illustrating the composition of each cluster generated by different clustering methods. The composition is visualized as the distribution of source genomes (E. coli K12 is recipient and rest are donors) in a cluster, in terms of fractions of a cluster assigned to different sources as well as fractions of source DNAs resident in a cluster (fractions shown in percentages). These cluster composition maps were generated at parametric settings detailed in Table 3 for segment-order (a), hierarchical (b), network (c), and affinity-propagation (d) clustering methods.

SUPPLEMENTARY FIG. 4.6 – Graphs illustrating the composition of each cluster generated by different clustering methods. The composition is visualized as the distribution of source genomes (E. coli K12 is recipient and rest are donors) in a cluster, in terms of fractions of a cluster assigned to different sources as well as fractions of source DNAs resident in a cluster (fractions shown in percentages). These cluster composition maps were generated at parametric settings detailed in Table 5 for segment-order (a), hierarchical (b), network (c), and affinity-propagation (d) clustering methods.

SUPPLEMENTARY FIG. 4.7 – Overall sensitivity (SN averaged over all test genomes) in identifying alien nucleotides while preserving donor identity in the genomes for each clustering method at increasing thresholds. The network clustering threshold (MCL inflation parameter) is represented as the percentage of the maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level). Donors are represented by their largest cluster for which they make up the majority of nucleotides. For donors that did not make up the majority of nucleotides in any cluster, the values of metrics SN, PR, and F1 were deemed zero for them.

SUPPLEMENTARY FIG. 4.8 – Overall precision (PR averaged over all test genomes) in identifying alien nucleotides while preserving donor identity in the genomes for each clustering method at increasing thresholds. The network clustering threshold (MCL inflation parameter) is represented as the percentage of the maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level). Donors are represented by their largest cluster for which they make up the majority of nucleotides. For donors that did not make up the majority of nucleotides in any cluster, the values of metrics SN, PR, and F1 were deemed zero for them.

CHAPTER 1

BACKGROUND AND SIGNIFICANCE

1.1 Introduction

Metagenomics is the study of metagenomes, that is, the collection of genomes representing a microbial environment. This is facilitated by simultaneous sequencing of DNA of all microbes dwelling the environment, without the need for isolation or culturing of individual members. Traditionally, metagenomics was restricted to the sequencing of 16S rRNA 'marker' genes, which are regions of the genomes with high taxonomic specificity (Almeida, Mitchell, Tarkowska, & Finn, 2018; Jovel et al., 2016; J. Patel, 2001). By focusing on 16S rRNA, researchers have been able to quickly and reliably identify the individual inhabitants of microbial communities, but beyond "who is there?", it does not inform any further on a microbial community. In particular, such an approach provides no information on "what are they doing" (Brooks et al., 2015; Shah, Tang, Doak, & Ye, 2011; Větrovský & Baldrian, 2013). Advances in next-generation sequencing, which have lowered costs and increased the throughput, have enabled sequencing the entire DNA complement of a microbial community. The whole metagenome shotgun (WMS) sequencing allows obtaining both the taxonomic and the functional profiles of a microbial community (Quince, Walker, Simpson, Loman, & Segata, 2017; Sevim et al., 2019; Venter, 2004). However, analyzing shotgun metagenomic samples is a non-trivial task. Identifying the sources of individual genomic snippets, or 'reads', requires robust databases and advanced algorithms. As the datasets continue to grow in size, the throughput of analysis software also becomes an issue, as many methods for taxonomic inference face the computational bottleneck; without efficient implementations within reasonable times, these methods face the challenge of becoming irrelevant in this age of exponentially growing databases (Ainsworth, Sternberg, Raczy, & Butcher, 2017; Burks & Azad, 2020; Ounit, Wanamaker, Close, & Lonardi, 2015; Wood, Lu, & Langmead, 2019; Wood & Salzberg, 2014). The potential of WMS sequencing is immense, from the discovery of new phyla (Wilson et al., 2014) to the revelation of antibiotic resistance genes (Handelsman, 2004), to name a few. However, the progress in this direction is hampered by the shortcomings of analytical tools and resources. Therefore, there is a pressing need to develop computationally efficient, robust methods for metagenome analysis.

1.2 Significance of WMS Metagenomics

The term metagenome was originally used to describe the collective genomes of soil microbiomes (Handelsman, Rondon, Brady, Clardy, & Goodman, 1998), but has since broadened to represent all microbial environments, as diverse as deep ocean vents (Anantharaman, Breier, & Dick, 2016) to the microenvironments within and on a human body (Lloyd-Price, Abu-Ali, & Huttenhower, 2016; Mukherjee, Beall, Griffen, & Leys, 2018). With each year, thousands of new WMS datasets are deposited to different repositories, such as the European Molecular Biology Laboratory, which currently hosts over thirty-one thousand metagenomic datasets in its MGnify repository (Mitchell et al., 2020). These datasets represent the only current genomic snapshots of uncultivated bacteria, and are believed to represent 70% of all known prokaryotic phyla (Wilson & Piel, 2013). Genomic contents of such organisms, which have not been successfully isolated and cultured, have been used to develop new CRISPR-Cas systems and are believed to bridge the current evolutionary gap between known eukaryotes and prokaryotes (Burstein et al., 2016). The human metagenome project (HMP) has also benefited from deeper insights offered by WMS sequencing, as their 16S rRNA amplicon datasets routinely fail to offer taxonomic resolution at or below the family or genus level (Hillmann et al., 2018). The first problem post WMS is to decode the metagenome to determine "who is there", however, because of the vast amount of unknowns and dependence of analytical methods on knowns (database sequences) to infer unknowns, the challenges abound. The metagenome profiling is based on taxonomic classification of small (~100-250 bp) nucleotide fragments arising from organisms that may or may not be represented in the genome databases.

1.3 Taxonomic Classification of Metagenomic Reads

For 16S rRNA metagenomic studies, phylogenetic classification relies on the quality of sequencing and the taxonomic representation within databases (DeSantis et al., 2006; Lan, Wang, Cole, & Rosen, 2012; Quast et al., 2012). As a universal marker in phylogenetic classification, the 16S rRNA ribosomal gene represents a taxonomic fingerprintthat can typically specify microorganisms down to the family or genus of their lineage. Also known as amplicon sequencing, 16S rRNA sequencing affords researchers advantages in that the targeted sequencing of such a specific region lowers sequencing costs, drastically reduces analysis time, and perhaps yields reasonably reliable classification albeit at higher taxonomic ranks. Despite the advantages, the redundancy of amplicon sequencing is beset with sequencing artifacts and chimeric reads, with an estimated 1 in 20 rRNA sequence records believed to contain 'significant' anomalies (Ashelford, Chuzhanova, Fry, Jones, & Weightman, 2005). Copy number variation of 16S rRNA makes it an unreliable estimator of taxonomic abundance without reference-quality genomic assemblies for all involved microbes (Acinas, Marcelino, Klepac-

Ceraj, & Polz, 2004; Větrovský & Baldrian, 2013). WMS studies attempt to address many of these problems by querying the entire genomic complement of each community member. However, the shotgun approach yields a vast amount of reads from overlapping regions across the genomes of organisms that may or may not have been represented in the sequence databases. When the genomes of such organisms are present in the databases, exact or near exact match of a metagenomic DNA fragment with a database sequence leads to straightforward identification of the source of the metagenomic read. Otherwise, a more sophisticated approach is required (Zielezinski, Vinga, Almeida, & Karlowski, 2017). Alignment, to an extent, does provide a level of taxonomic abstraction. Reads from shared genes of highly similar bacterial strains or species do tend to be similar, often assessed based on percent identity and coverage among other metrics, and this is an established method of inferring homology (Moreno-Hagelsieb & Latimer, 2008). Inferring homology between sequences originating from distantly related organisms, or of rapidly evolving sequences, is more difficult, and alignment-free methods have been developed to address the shortcomings of sequence alignment methods (Ding, Cheng, Cao, & Sun, 2015; Gregor, Dröge, Schirmer, Quince, & McHardy, 2016; Rosen, Garbarine, Caseiro, Polikar, & Sokhansanj, 2008).

1.4 Alignment-Free Profiling via Markov Modeling

Markov models have previously been established as a reliable method for taxonomic profiling of metagenomic reads (Brady & Salzberg, 2009). One of the earlier alignment-free implementations, PhymmBL, leverages the sensitivity of Markov models and specificity of BLAST alignment to assign taxonomic lineages to each of the reads in a metagenomic dataset. This method compares favorably with its component methods, Phymm (alignment-free) and

BLAST (alignment-based), particularly when the read originating species are not represented in the genome database (Brady & Salzberg, 2011, 2009; Wood & Salzberg, 2014). Despite the apparent advantages of PhymmBL, its update and support were discontinued in 2012, and advances in alignment algorithms and rapidly growing nucleotide databases have made it a less attractive option for WMS analysis (Ainsworth et al., 2017). In cases where database representation is not an issue, PhymmBL processing reads at 0.01% the speed of similarly performing classification tools based on exact *k*-mer alignments is a prime drawback of its usage for large-scale data analysis (Ainsworth et al., 2017; Wood & Salzberg, 2014). Furthermore, the most current release calls upon the archived RefSeq databases for model training, failing to run without source code modifications on fresh installations.

A Markov model is characterized by initial (marginal) and transition probabilities estimated from training data. For metagenomic classification, such training data are represented by genome databases. In DNA sequence analysis, Markov models illuminate the short-range dependencies in nucleotide ordering and allow the prediction of a nucleotide based on the preceding oligonucleotide of a length *m* that defines the model order. For a *m*th order Markov model, the model parameters, i.e. the initial and transition probabilities, are estimated from the frequencies of (*m*+1)-mers in the training data. For a sequence *S* of length *N*, the probability of *S* to be generated by a Markov model *M* of order *m* is given as:

$$P(S|M) = p(\alpha_1 \alpha_2 \dots \alpha_m) \prod_{i=m+1}^{N} p(\alpha_i | \alpha_{i-m} \alpha_{i-m+1} \dots \alpha_{i-1}) \qquad \textbf{(Eq. 1.1)}$$

where α_i is the nucleotide α at position *i* in sequence *S*, and *P* (or *p*) denotes probability. With respect to WMS datasets, *S* represents any one of the unassigned, individual DNA reads sequenced from a metagenome; the initial and transition probabilities to be used to compute

$P(S|M)$ are estimated from the training data comprised of full genomes, with a Markov model M built for each genome. In the straightforward application of Markov models to WMS data, the assignment of taxonomic identity to a read S is based on the genome model M that yields the highest probability $P(S|M)$ among all genome models.

1.5 Shortcomings of Alignment-Free Profiling via Markov Modeling

The number of model parameters increases exponentially with the order m of the Markov model. Therefore, for higher order models, the process of modeling can become computationally prohibitive, especially at higher ($m > 8$) orders. While lowering the order can reduce the computation, the predictive poweris diminished. Beyond the computational limitation, higher orders require larger amounts of training data, with an increasing number of the 4^{m+1} oligonucleotides absent in smaller or partial genomes. This can lead to zero counts, and therefore zero values for initial or transition probabilities, leading to a $P(S|M)$ of zero, even though the read may have originated from the species represented by M. Unless addressed, this could severely limit the predictive power of the Markov model-based approach. A Markov model M must adequately account for all possible k-mers, whether present or not in the training data. If a sequence S (e.g. metagenomic read) contains a k-mer not accounted for by the model, this becomes a zero-probability event, and the resultant $P(S|M)$ becomes zero, even if S originates from an organism represented by M. The frequency of such an occurrence increases with higher orders, and is compounded in sequences that may contain base-call errors resulting in zero-probability k-mers that are erroneously reported within S. Variable order and interpolated order Markov models have been developed to mitigate these effects, utilizing probabilities of oligomers of lengths up to and including m. In essence, such models are the result of combining several models, each of a different order but trained on the same data. Methods for weighting models of different orders are based on the counts of oligonucleotides of different sizes, resulting in a combined model that has addressed the zero count events, e.g. by falling back on lower order oligomers (models) to replace rare or non-existent higher order oligomers, and thus obviatingthe zero-probability events. This problem has been extensively studied, resulting in both mathematically rigorous and heuristic modeling algorithms. The Markov model component of PhymmBL, namely Phymm, takes a heuristic approach, interpolating models with weights based on *k*-mer frequencies and a confidence score from a chi-squared test that assesses the difference between oligonucleotide distributions of higher orders and those of lower orders combined (Brady & Salzberg, 2009). Using the algorithmic foundation by GLIMMER (A. Delcher, 1999; Salzberg, Delcher, Kasif, & White, 1998), a nucleotide prediction is made by one of Phymm's interpolated Markov models (IMMs) based on the preceding *m*-mer only if the frequency of the *m*-mer is greater than 400. Otherwise, the models are interpolated from 0^{th} to m^{th} order with weights assigned by the interpolated context model (ICM) of GLIMMER (Brady & Salzberg, 2011, 2009; A. Delcher, 1999; A. L. Delcher, Bratke, Powers, & Salzberg, 2007). The prediction of a nucleotide by an ICM of order *m* based on a mutual information test to determine the most informative positions in the preceding oligomer of length *m*. A full description of the ICM building process is covered in the second GLIMMER publication (A. L. Delcher et al., 2007), as it is the build-icm sub-program of GLIMMER that generates the models used by PhymmBL.

While ICMs are posited to be better predictors than SMMs and are inherently immune to the zero-probability problems observed in probabilistic modeling, the process is computationally expensive. Estimation of 4^k and 4^{k+1} initial and transitional probabilities, from order k=0 to m, is required for a m^{th} order interpolated Markov model. A Chi-square test for IMM building and mutual information test for ICM building brings in additional computational load. Phymm's conception in 2009 might have been timely as the field of metagenomics also emerged just a few years prior to this and the prevalence of partially sequenced prokaryotic genomes might have been a justifiable argument to use interpolation to address the limited training data problem. Furthermore, the code for building ICMs was already established as a part of the GLIMMER program and thus was readily deployed to perform read classification by PhymmBL. As of 2020, there are over 29,000 RefSeq genomes from unique species with complete assembly status. A fully sequenced genome yields the full training set for a particular organism, however, the benefits associated with interpolating models based on complete genomic training data are not clear. Although Phymm was compared to non-Markovian methods, it was not benchmarked against other variants of Markov models. We therefore revisited Markov model application in metagenomics, benchmarked Markov model algorithms, and leveraged this in developing an efficient Markov model algorithm for metagenome profiling.

1.6 Focus and Organization of the Dissertation

The primary goal of this dissertation is to provide a novel alignment-free platform for taxonomic classification of metagenomic sequences, which not just addresses the limitations of existing Markov model-based algorithms, but also offers complementation to state-of-the-art alignment-based profiling. This work involved developing novel Markov model-based algorithms, optimizing existing algorithms to establish robust training datasets, and integrating alignment-free and alignment-based approaches into a metagenomic classification platform that presents the most inclusive, highest performing method for metagenomic sequence classification.

Taking advantage of the influx of thousands of fully sequenced genomes that allows exploiting the power of higher order Markov models, we have devised a slimmer but computationally more efficient Markov model platform for metagenome profiling. Better overall performance of higher order Markov models vis-à-vis computationally expensive interpolated models in our extensive benchmark experiments paved the way towards developing a highly efficient Markov model algorithm for sequence classification. Written in C++, our standard Markov model (SMM) program provides a database-free, high-performance modelling platform that is linearly scalable with multi-core CPUs. In Chapter 2, we present the development of this algorithm, and demonstrate its effectiveness compared to variants of Markov models in metagenomic sequence classification. Using simulated metagenomic datasets, we benchmark the performance, and compare to best-in-class alignment strategies as proof of concept.

Training data, both for modelling and benchmarking, should be regarded with utmost care. Poor quality, unrepresentative data can lead to misleading conclusions, obscuring the performance of even the best methods. In Chapter 3, the algorithmic advances in alignment-free metagenome profiling (Chapters 2) are realized in the form of a more advanced, user-friendly tool, POSMM, a Python-Optimized SMM metagenomic classifier. The core of POSMM analyzes metagenomic reads using the SMM algorithm without constructing a database of genome models. By obviating the need to construct a model database, a first step in all existing methods of this class, it achieves a far higher computational efficiency in metagenomic sequence classification. Markov model-derived read scores are renormalized (scaling between 0 and 1) using logistic regression estimators trained on genomic clusters outputted by a new, more efficient version of segmentation and clustering algorithm. As POSMM develops models on the fly, we demonstrate how our method can be incorporated with the popular alignment-based metagenomic classifier Kraken to raise the accuracy bar to a level that cannot be achieved by using Kraken or POSMM alone. The computationally efficient version of the Markovian Jensen-Shannon Divergence (MJSD) segmentation-clustering algorithm presented in this chapter is adapted from a previous version used in IslandCafe, a program for genomic island prediction. This optimized algorithm, rewritten in C++, allows hyper-segmentation and clustering of a prokaryotic genome within seconds. This algorithm was used to filter out extraneous sequences such as adapter remnants and contaminant DNA often present in microbial genome assemblies. Algorithmic advances in methods for segmentation and clustering of genomic and metagenomic sequences are critical for large-scale, robust analysis of genomes and metagenomes. Although many different clustering algorithms exist for grouping compositionally similar DNA sequences, their relative strengths and weaknesses are not well-understood, particularly of those algorithms that follow segmentation with clustering in many different heuristic ways to group similar segments. Therefore, in Chapter 4, we benchmark various clustering approaches previously used to group compositionally similar segments following the recursive segmentation of a genome. This includes positionally-dependent, hierarchical, graph-based, and affinity-propagation clustering methods. Using artificial chimeric genomes, we reveal the complementary strengths and weaknesses of different clustering approaches and the most effective methods for clustering genomic segments under many different conditions.

Finally, in Chapter 5, we discuss further improvements to POSMM in the future to keep it relevant in this era of big data in genomics and metagenomics.   Our machine-learning derived regression models, stored and implemented in standard JSON, can be customized and updated as they are based on the popular sci-kit Python libraries. This will allow POSMM to evolve well beyond its initial release, building on the foundation we have established herein. We also discuss the shortcomings associated with shotgun metagenomics, and how programs such as POSMM can adapt as the field matures. The new version of MJSD segmentation and clustering algorithm can have applications that extend beyond taxonomic classification, such as in genomic island detection, metagenomic binning, and in the development of gene interaction networks.

CHAPTER 2

HIGHER-ORDER MARKOV MODELS FOR METAGENOMIC SEQUENCE CLASSIFICATION*

2.1 Introduction

Markov models have proved to be an invaluable resource for parsing genomes for information encoded within them. Using the frequencies of oligonucleotides within genomes, researchers have successfully deployed Markov chain models to solve a variety of biological problems, including identification of genes (Besemer, 2001; A. Delcher, 1999; Lukashin, 1998; Salzberg et al., 1998), determination of boundaries between introns and exons, localization of sequence motifs (R. D. Finn, Clements, & Eddy, 2011; Robert D Finn et al., 2015; Quevillon et al., 2005; Wheeler & Eddy, 2013), and inference of the taxonomic origins of fragments in metagenomic datasets (Brady & Salzberg, 2009; Skewes-Cox, Sharpton, Pollard, & DeRisi, 2014).

Among the recent biological applications of the Markov model is in the taxonomic classification of sequence fragments in metagenomic datasets, which entails building as many models as the sequences of distinct genomes in the microbial database (Brady & Salzberg, 2009; Skewes-Cox et al., 2014). The model parameters, namely, the initial and transition probabilities, are learnt from the frequencies of oligomers in the training sequence i.e. the genome where (k+1)-mer frequencies specify the probabilistic parameter estimates of the model of order k (R. K. Azad & Borodovsky, 2004). The number of model parameters increases exponentially with model order, rendering high order models not suitable for modeling microbial genomes which are typically of the order of Mbp in size. While models of order 6 or

*This chapter is reproduced from Burks, D. and Azad, R. K. (2020). Higher-order Markov models for metagenomic sequence classification. Oxford Bioinformatics, 36(14), 4130-4136. Authors retain copyright.

lower were found suitable for modeling coding and non-coding sequences in prokaryotic gene prediction (R. K. Azad & Borodovsky, 2004), no such determination has yet been made in metagenomic sequence classification where information content of whole genome is exploited for building models. Rather, a previously used model in gene prediction, namely interpolated Markov model, was used without regard to the order of the interpolated model optimal for such analysis and without justification of this model vis-à-vis other Markov models in metagenome analysis (Brady & Salzberg, 2009). This class of models exploit the frequently occurring longer oligomers for predicting the succeeding nucleotides, otherwise it falls backs on frequently occurring shorter oligomers (lower order models) for prediction.

Rarely occurring or unencountered oligomers pose a significant challenge to probabilistic methods whose predictive power depends on the reliable estimate of probabilistic parameters. Higher order models are sought for greater predictive power, however, frequencies of oligomers dwindle with increase in order with many becoming too low to be reliable for prediction. Furthermore, for oligomers with low counts, the probabilities estimated based on these may be unrepresentative, especially if the reference genome (training data) is incomplete. On the other hand, oligomers that are not represented in the reference genome data lead to zero probabilities, thus rendering the models incompatible with fragments containing evolutionary variations such as mutations or having sequencing errors. This situation also leads to breakpoints in the Markov chain, a violation of the fundamental assumption that the modelled process does not terminate (Rabiner & Juang, 1986).

Strategies for dealing with low-to-zero count oligomers, commonly referred to as *smoothing*, have been extensively studied in the field of natural language processing (Essen &

Steinbiss, 1992; Kuhn, 1988; Ney & Essen, 1991; Saul & Pereira, 1997). Genomic applications have relied heavily on order interpolation to deal with this problem (Brady & Salzberg, 2009; A. Delcher, 1999; Salzberg et al., 1998). The process is intensive, requiring the calculation of probabilities for all model orders up to and including the designated *k* for k^{th} order interpolated model (Salzberg et al., 1998). Different order models are combined through weights, also called interpolation parameters, allowing the use of longer oligomers (higher orders) only when their representations are deemed adequate, otherwise the model falls back on the shorter contexts (oligomers) for prediction of the succeeding nucleotide. Phymm and PhymmBL (a “hybrid” that integrates Phymm and BLAST results), the Markov model-based software tools for taxonomic classification of sequence fragments, use a decision tree based interpolation of orders first introduced in early versions of the gene-identification program GLIMMER (Brady & Salzberg, 2009; A. Delcher, 1999). However, this model framework, referred to as interpolated context model or ICM, requires a significant computational investment, partly due to calculations along the ICM decomposition tree for mutual information between nucleotide distribution at each position within *k*-mer windows and the distribution at the $(k+1)^{th}$ position (A. Delcher, 1999).

The advent of next-generation sequencing has created a glut of unclassified genomic fragments originating from mixed populations of microbes dwelling different environments. These metagenomic datasets are often very large, each containing millions of short DNA sequences or reads (Mitchell et al., 2018). The need to analyze such vast quantities of reads in a reasonable time frame has led to the adjustment of existing classification methods, such as the local alignment, that are simply too slow or computationally expensive in their most accurate forms (Y. Chen, Ye, Zhang, & Xu, 2015; Menzel, Ng, & Krogh, 2016; Ounit et al., 2015; Wood & Salzberg, 2014). As metagenomic classifiers, these programs employ specialized algorithms that attempt to sacrifice an acceptable margin of accuracy for exponential speed gains (Wood & Salzberg, 2014). Despite being one of the most accurate taxonomic classification methods for short reads (Ainsworth et al., 2017; Brady & Salzberg, 2009; Nalbantoglu, Way, Hinrichs, & Sayood, 2011; Wood & Salzberg, 2014), Markov model-based methods have fallen heavily out of favor mainly due to the significant computational time investments associated with their current implementations (Corvelo, Clarke, Robine, & Zody, 2018; Nalbantoglu et al., 2011).

In this study, we revisited the Markov chain model and investigated its effectiveness in taxonomic classification of metagenomic reads. Comparative assessment was performed for Markov models of different orders, as well as with interpolated models, on synthetic metagenomes of different composition and levels of complexity to identify the optimal model structure for metagenome profiling. Our results show that a simple implementation of Markov models– pseudo-count supplemented higher order models– outperformed complex models such as those implementing interpolation in metagenomic classification of reads as short as 100 nt at all taxonomic ranks, and even longer reads at lower taxonomic ranks. Models of order 9 and higher demonstrated significantly high accuracy in classification (~70% or higher) at higher taxonomic ranks (order or above). Furthermore, we compared the Markov model-based approach to local alignments performed with BLAST (Altschul, Gish, Miller, Myers, & Lipman, 1990) to demonstrate the considerable advantages of using Markov models for classifying reads originating from poorly represented taxa, and the limitations of alignment for higher levels of taxonomic abstraction.

We further emphasize that models of order 9th and above are often not implemented and tested in this domain apparently due to the computational overload in training such models. This bottleneck was overcome via an efficient implementation that allowed learning of parameters for models of orders up to 12. Our novel implementation within a C++ framework complements sequence alignment-based approach and can therefore be used in concert with alignment-based methods to interrogate metagenome datasets to gain better understanding of the microbial communities, perhaps within a reasonable time frame now.

2.2 Materials and Methods

2.2.1 Metagenome Dataset Generation

To empirically benchmark the considered strategies for taxonomic classification, synthetic metagenomes were constructed by utilizing the entire set of fully sequenced prokaryotic genomes available in the NCBI RefSeq database (O'Leary et al., 2016). Assessment at the genus and higher taxonomic level entailed "species masking", wherein matches between a test-read and models were not allowed to be from the same species, that is, the genome model(s) for the test-read originating species were excluded. Assessments were also performed with genome models masked for other taxonomic ranks; masking at a specific taxonomic rank entails excluding genome models representing the specific rank of the test-read. A total of 5716 genomes were considered (with plasmids excluded) and test sets were assembled by sampling sequences from a metagenomic sampling pool comprised of genomes representing multiple members from each taxon (rank). To evaluate the efficacy of each method as a function of nucleotide composition, the metagenomic sampling pool was split into three subpools based on the G+C content (%GC) of each genome. The three subpools represent the genomes with %GC < 40, genomes with %GC between 40-55, and genomes with %GC > 55 respectively, and were determined based on %GC histogram for all considered genomes (SUPPLEMENTARY FIG. 2.1). Metagenomic sets, each of 100,000 reads, were sampled from the metagenomic pool proportional to the abundance of the taxa and genome sizes, and similarly from each subpool, using the metagenomic read generator Grinder v0.5.4 (Angly, Willner, Rohwer, Hugenholtz, & Tyson, 2012). In total, five replicate datasets consisting of reads of size 100 nt and another set with reads of size 250 nt were generated for the metagenomic pool as well as for each subpool. Datasets with simulated sequencing errors were also constructed using the "-md poly4 3e-3 3.3e-8" parameter. Error probability was based on a fourth degree polynomial modeling the Illumina sequencing error rate (Grinder v0.5.4 manual; Korbel *et al.* 2009; Angly *et al.* 2012; Moller and Liang 2017). Interactive Sankey diagrams for metagenomic read datasets are provided as supplemental files (SUPPLEMENTARY FIGS. 2.2-2.5), which illustrate the abundances of different taxa in the simulated metagenomes. Taxonomic annotation was performed using the ncbitax2lin R script (https://github.com/zyxue/ncbitax2lin) in order to associate Genbank accession number with its full taxonomic lineage to each genome.

2.2.2 Whole Genome Markov Models

Variants of Markov model including those that use smoothing strategies to deal with low (or zero) count *k*-mers were assessed on test metagenomes constructed as described above. Standard Markov model (SMM; we refer below LOMs and HOMs collectively as SMMs) of *k*th order was established for each genome based on the counts of (*k*+1)mers in the genome, with pseudocount incorporated by incrementing the count of each (*k*+1)mer by 1 in order to circumvent the problem of zero probability associated with non-occurring (*k*+1)mers in the sequence. The initial probabilities (P_i) and transitional probabilities (P_t) for a *k*th order were estimated as follows. Given that S_x is one of the 4^k possible *k*-mers in set $\{S_1,S_2,\ldots S_m\}$ where $m=4^k$, and $c(S_x)$ is the count of S_x in a genome, the estimate of the initial (or marginal) probability of S_x was obtained as:

$$P_i(S_x) \approx \frac{c(S_x)+4}{\sum_{x=1}^{m} c(S_x)+4\cdot m} \qquad \textbf{(Eq. 2.1)}$$

The estimate for the probability of transitioning from S_x to the just succeeding nucleotide *b*, where $b \in \mathcal{A}$ and $\mathcal{A} \equiv$ {A, T, C, G}, was obtained as:

$$P_t(b|S_x) \approx \frac{c(S_x\,b)+1}{\sum_{z\in\mathcal{A}} c(S_x\,z)+4} \qquad \textbf{(Eq. 2.2)}$$

Interpolated Markov models that use different interpolation strategies to combine lower and higher order models were also built for each genome and compared with SMM for metagenomic sequence classification. These models included Deleted Interpolation Model (DIM) (R. K. Azad & Borodovsky, 2004), Interpolated Markov Model (IMM) (Salzberg et al., 1998), and Interpolated Context Model (ICM) (Delcher 1999; Brady and Salzberg 2009). A total of 5716 genome models were built each for SMM, DIM, IMM, and ICM. The programs used for building IMMs and DIMs have been provided at https://github.com/djburks/SMM in folder SUPPLEMENTAL CODE, with instructions for their compilation and use (see also SUPPLEMENTARY NOTES). Instructions for building non-default (i.e. 8th order) ICMs via PhymmBL have also been provided in the MODELING NOTES file in the SUPPLEMENTARY NOTES folder. The SMM codebase's commit hash is e500859cff05a9a537a8b2263ff8be638e68c5a5.

2.2.3 Markov Model Scoring

A probabilistic score for a metagenomic read is calculated using the initial and transitional probabilities (Eqns. 1 and 2) for each model. For a read sequence *R* of length *N*, the probabilistic score for *R* to be generated by model *M* of order *k* is given as:

$$P(R|M) = P_i(r_1 r_2 \dots r_k) \prod_{i=k+1}^{N} P_t(r_i | r_{i-k} r_{i-k+1} \dots r_{i-1}) \qquad \textbf{(Eq. 2.3)}$$

where r_i denotes the nucleotide *r* at position *i* in read sequence *R*. Here the model *M* refers to any of the considered models, namely, SMM, IMM, DIM, and ICM. While standard probabilistic parameters estimated based on pseudo-count initialized *k*-mer frequencies (Eqns. 1 and 2) were used for SMM of order *k*, the probabilistic parameters for IMM, DIM, and ICM of order *k* were estimated as described in the respective papers (Brady & Salzberg, 2009; Salzberg et al., 1998) and implemented in the respective programs made available by their authors. These latter programs used different smoothing techniques to combine the models up to order *k*, thus attempting to address the problem of encountering scarce or non-existent longer oligomers during the training for higher order models. We refer readers to the published papers on these models for details on their smoothing approaches (R. K. Azad & Borodovsky, 2004; Brady & Salzberg, 2009; A. Delcher, 1999; Salzberg et al., 1998).

2.2.4 Classification and Benchmarking

Markov model variants were benchmarked on synthetic metagenomes constructed as described above. Assessment was performed by masking the Markov models of test read originating species in the genome model database. Assessment was thus performed at the genus and higher taxonomic levels. As only a small fraction of metagenomic reads have their source species represented in the genome database, species masking allows testing of the models' ability to assign the reads correctly to genus or higher levels when the read originating species is yet unknown, i.e., not represented in the database. We therefore included only those genera with two or more species with sequenced genomes in the assessment. The model with the highest probability score (Eqn. 3) was deemed the "best" match, and the taxonomic identity was assigned accordingly to the read. The performance at different taxonomic levels was assessed by obtaining the taxa (genus and higher ranks) for the best match model (genomes) for each read. Note that the lower level misclassification will get corrected at higher level if that lower level taxon lies hierarchically within the same higher level taxon as of the read being classified. Thus, the classification performance either remains the same or improves with higher ranks.

For ICMs, the output of PhymmBL (v4.0) was parsed to find the probability scores for reads for each considered ICM. For SMM, our software implementation of different order models was used for scoring reads and for IMM and DIM, the executables of these models were used to generate the probability scores; these have been made available for download from a GitHub repo at https://github.com/djburks/SMM.

SMM was also benchmarked against BLAST, the local alignment software often used in metagenomic sequence classification. Read alignment was performed using BLASTn, a component of the NCBI-Blast+ 2.8.1 suite, at the default parameter setting. Species masked assessment was done by first performing BLAST alignment of each read against genome database. A read was assigned the taxonomic identity of the genome that yielded the highest bitscore match with the read but representing a different species than of the read. Accuracy was reported as the percentage of reads in a test metagenome that were correctly classified by a method.

In all performance tests, only a single thread was used for each method. All tests were conducted on a Ryzen 1600 with 32GB of RAM on Manjaro Linux with the 4.19.9 kernel. Tests were run independently to avoid I/O interference, and all models were loaded from the same 7200RPM hard drive. Model generation for ICMs was handled by the build-icm program included with the latest version of PhymmBL (v4.0).

2.2.5 Comparative Assessment on Real Metagenomic Data

A shotgun metagenome dataset consisting of 99,933 Illumina Miseq 75 bp single-end reads was downloaded from the NCBI SRA with the accession number ERR965975 (Leinonen, Sugawara, Shumway, & International Nucleotide Sequence Database Collaboration, 2011). The reads, sequenced from mouse fecal samples, were classified using 12th order ICM and SMM. ICM was constructed using PhymmBL v4.0 at its default setting, and the output was parsed to exclude contributions from its BLAST component. The phylogenetic classification of a read was performed by identifying the lineage (taxa) represented by the highest scoring genome model for the read. The genome database was comprised of the 5716 genomes as was used in the benchmarking using the synthetic metagenomes. Prior to classification, an 8 bp barcode sequence was removed from one of the ends of each read. Classifications were visualized through interactive Sankey diagrams at the phylum, class, order, family, and genus levels.

2.2.6 Generation and Assessment of Whole-Taxa SMMs

A combined fasta file for each taxon represented in our RefSeq database was constructed and used to build 12th order SMM models. Each individual genome in our database was appended to a fasta file representing its species, genus, family, order, and class.

The same 100 nt and 250 nt read datasets used in the previously described assessments were classified using these combined models, and masked classification accuracy was calculated as described above. Phylum through genus assignment of a read was derived from the highest scoring whole-species model not representing the originative species of the read. For higher level models, such as whole-genus or whole-class models, only the accuracy for encompassing taxa was derived with masking similarly applied. For example, in assessing family-level models, the phylum, class, and order classification of a read originating from a *Salmonella* genome was derived from the highest scoring whole-family model excluding the *Enterobacteriaceae* model.

2.3 Results

2.3.1 Assessment of Markov Model Variants of 8th Order on Synthetic Metagenomes

IMM of 8th order was previously used for gene prediction as well as for metagenomic read classification (A. Delcher, 1999; Kelley, Liu, Delcher, Pop, & Salzberg, 2012). However, data for comparison with 8th order SMM were not presented. Therefore, here we first present comparative assessment of IMM and SMM, both of 8th order, in classifying reads from synthetic metagenomes. Comparisons were also made with ICM and DIM, also of the 8th order (Figure 2.1, SUPPLEMENTARY FIG. 2.6). Classification was performed by assigning the lineage (genus and higher taxonomic ranks) represented by the highest scoring model for a synthetic metagenomic read. After assigning taxonomic identity to each read, the predicted taxa were tallied with actual taxa for all reads. Accuracy of a model was obtained as the percentage of correct classifications at a taxonomic level, beginning with genus and up to phylum level. Models from the same species as of the read origin were not considered. Species masked assessment tested the ability of a model to correctly assign taxonomic ranks to a read originating from a species for which only genomes representing the corresponding higher taxonomic ranks are available in the database. For reads whose source genomes are represented in the database, classification task is straightforward with local alignment methods, however, these methods do not perform well if the source species for the reads of interest are not represented in the database (Brady & Salzberg, 2009).

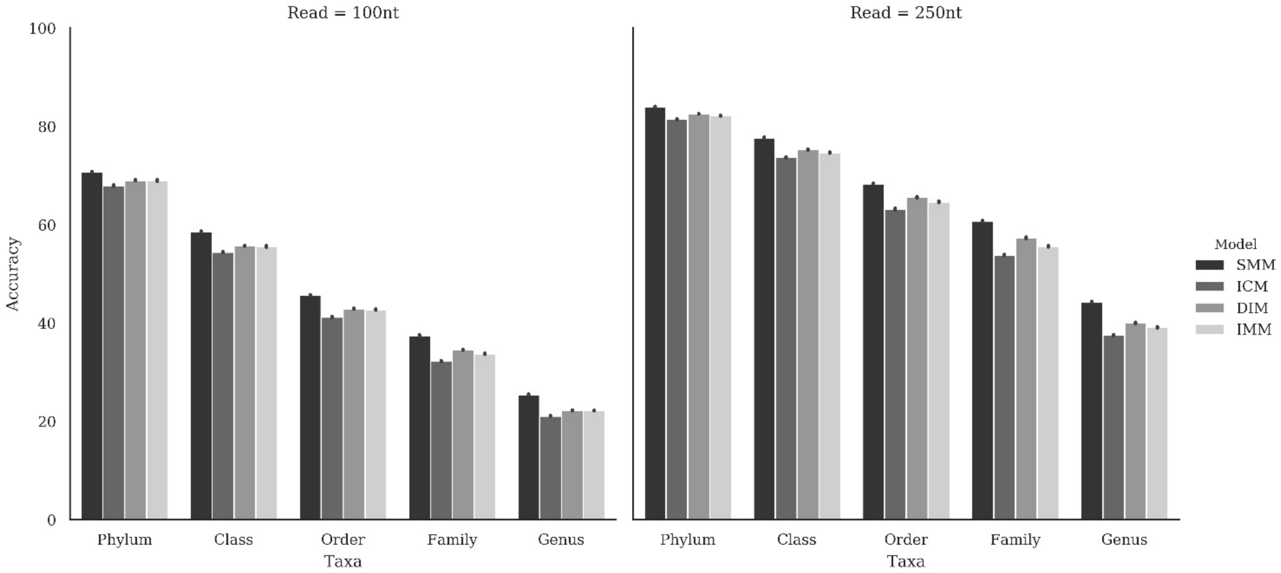

Figure 2.1: Accuracies of different Markov model based methods of 8th order in classifying reads of length 100 nt (left) and 250 nt (right) at different taxonomic ranks.

SMMs produced higher classification accuracies than the variants of interpolated Markov model (DIM, IMM, ICM) at all taxonomic levels for both 100 nt and 250 nt read sets sampled from the full RefSeq dataset (Figure 2.1). This was also true for each %GC subpool, except for genus-level assignment for 250 nt reads generated with Illumina error models for 40-55% GC genomes (SUPPLEMENTARY FIG. 2.7). Unexpectedly, ICM, perhaps among the most complex Markov models, generated lowest classification accuracies in all tests (Figure 2.1, SUPPLEMENTARY FIG. 2.6). This performance gap increased with read length, with ICM lagging behind other models by over 6% in accuracy at the genus level classification. Overall, the relative performance trend was consistent across the test datasets (Figure 2.1, SUPPLEMENTARY FIG. 2.6).

2.3.2 Effects of Increasing Markov Model Order on Classification Accuracy

Previous studies on applications of Markov chain models in sequence analysis have used models of order up to 8th. Applications of HOMs (9th order or higher) have been elusive, primarily due to lack of enough labeled data to train these models and the computational limitations in dealing with such models. However, as the computational power and resources increase, the ability to classify datasets with HOMs becomes more feasible. As Markov chain models were last tested for metagenomic sequence classification nearly a decade ago and both computational capabilities and genome resources have increased quite significantly since then, it was tempting to evaluate HOMs for read classification. We therefore built SMMs of 9th, 10th, 11th, and 12th order and assessed their performance vis-à-vis the 8th order model on the 100 nt and 250 nt full read datasets as well as the different GC range datasets with and without simulated Illumina sequencing errors (Figures 2.1 and 2.2). Notably, the highest order SMM among these (12th order) produced the highest classification accuracy at all taxonomic ranks for the 100 nt reads, but was not the best performer among HOMs at ranks above order for the 250 nt error-free read set where the performance seemed to reach a cap regardless of model order (Figure 2.2). However, on datasets of reads with simulated sequencing errors, models performed better with increasing order for all considered read lengths and at all taxonomic ranks (Figure 2.2, SUPPLEMENTARY FIG. 2.8).

To verify whether SMMs compares favorably with ICM even at higher orders, we generated 12th order ICM and assessed its classification accuracies with those of SMMs. Interestingly, although higher order SMMs, i.e., the HOMs, outperformed the 12th order ICM on 100 nt datasets at all taxonomic ranks (by up to 16.8% in classification accuracy), the ICM either performed similarly or slightly outperformed SMMs at higher taxonomic ranks (order and above) on error-free 250 nt datasets (Figure 2.2). However, when the simulated Illumina sequencing errors were introduced, SMM of order as low as 9 outperformed the 12th order ICM by up to 3.6% in accuracy at all taxonomic levels (SUPPLEMENTARY FIGS. 2.8, 2.9).

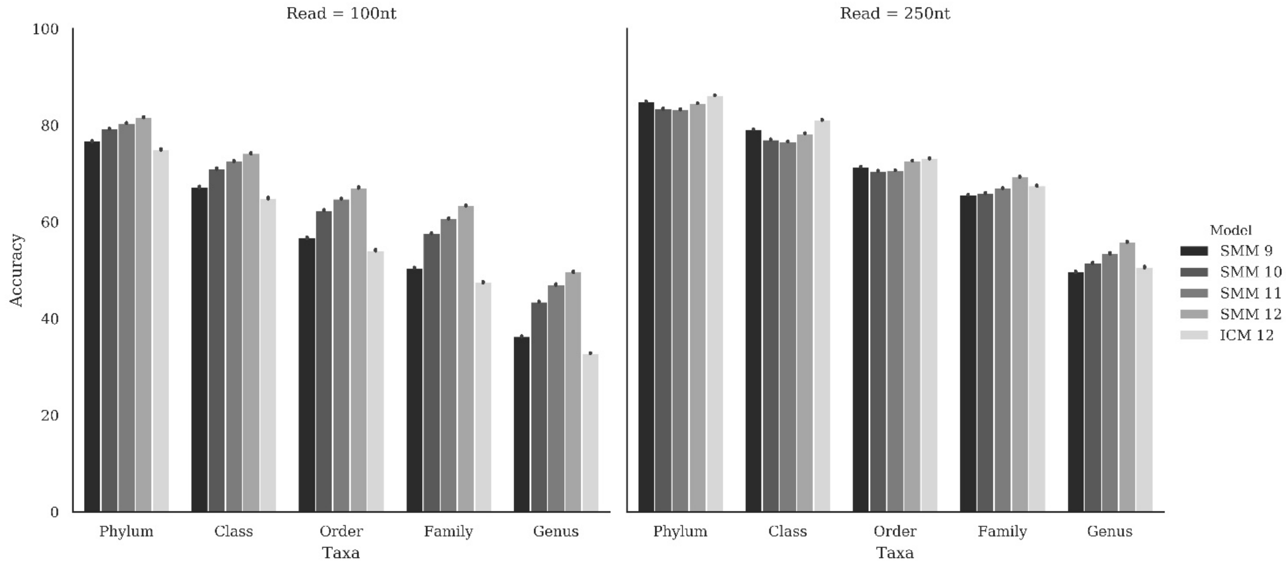

Figure 2.2: Comparative assessment of SMMs of different orders and ICM of 12th order (PhymmBL's default setting) on classification of reads of length 100 nt (left) and 250 nt (right) at different taxonomic ranks.

2.4 Computational Performance on Model Construction and Scoring

Computational performance of any Markov model-based program for taxonomic classification is important given the consistently increasing size of modern datasets. To compare ICM and SMM performance, we constructed 5,716 whole genome ICMs of order 12 for comparison with SMMs of orders 9 through 12. SMM of order 12 used 12 preceding nucleotides for predicting the next nucleotide (the classic 12th order model) whereas ICM interpolated 10 of the 12 preceding nucleotides to predict the next nucleotide (12th order interpolation but selecting the “most informative” 10 of 12 context nucleotides). ICM construction and scoring was performed using the latest version of PhymmBL (v4.0). SMM was compiled with g++ 0.9.2 with O4 optimizations. To assess the computational speed in scoring metagenomic reads, we calculated the runtime of SMM and that of the read scoring stage of PhymmBL on read sets of increasing size (Figure 2.3). PhymmBL incorporates a one-time model building stage for each genome prior to scoring. For our database of 5,716 genomes, this step required over 34 hours. In contrast, the model building and read scoring are a single stage in SMM, allowing users to change the parameters such as model order without requiring a complete rebuild of the database, a requirement for the ICMs of PhymmBL. There is an initial time investment associated with single-stage modeling and scoring. However, the rate of models produced and reads in a dataset scored by these models (Models/Minute, Figure 2.3) is significantly higher for SMM versus PhymmBL (~4 times by SMM of order 12 vs. PhymmBL for a dataset of 100,000 reads, Figure 2.3). This results in much lower runtime for SMM compared to PhymmBL as the number of reads is increased (Figure 2.3). PhymmBL takes over 5 days to score reads of a simulated metagenome consisting of 800,000 reads of size 100 nt; in contrast the 12th order SMM program accomplishes this within 12 hours. It should be noted that PhymmBL used less than 500 Mb RAM during both model building and read scoring. SMM, which loads reads into memory for rapid lookup, uses memory proportional to the dataset being analyzed. A modest read set of 100,000 100-nt reads required ~800 Mb of RAM, whilst a fairly large dataset of 1 million 100-nt reads required ~1.92 Gb of RAM. SMM thus exploits the abundance of memory in modern systems to accomplish read classification in reasonable times. This is in tune with the needs to develop programs that can exploit the memory abundance and accessibility in the current generation of computers to perform big data analysis such as that of metagenomes. Note that the BLAST component of PhymmBL, including database construction and read alignment, was not included in the performance benchmark.

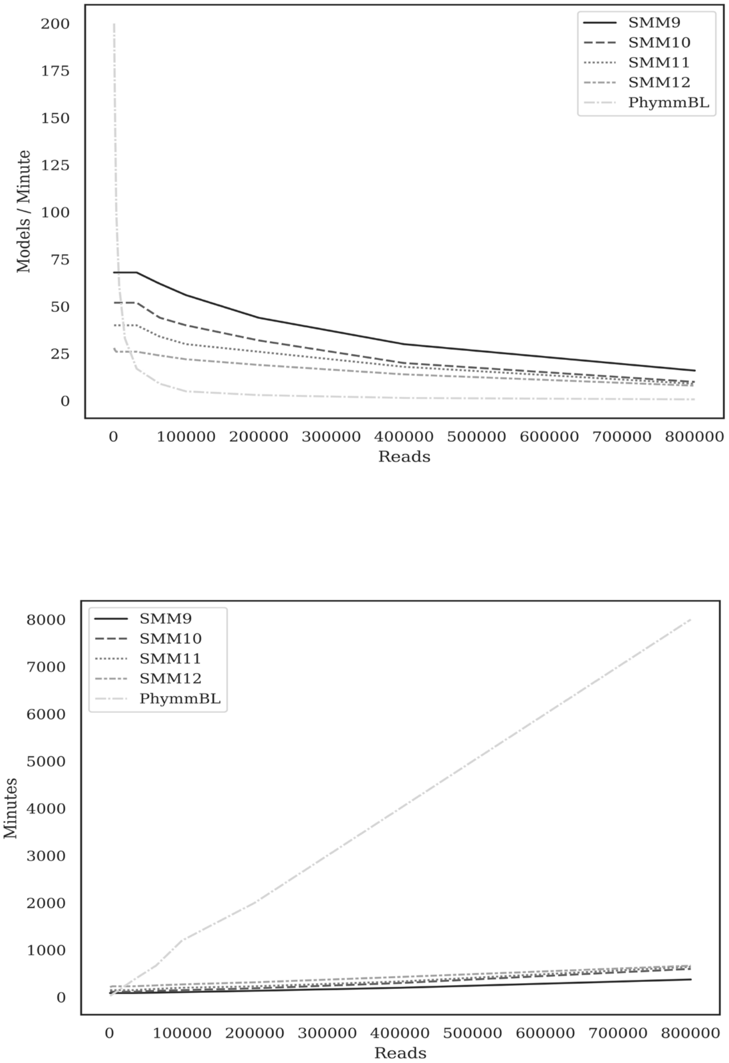

Figure 2.3: (Top) Rate of models produced and reads in a dataset scored (Models/Minute) as function of dataset size, for SMM of different orders and PhymmBL. (Bottom) Algorithm runtime (in minutes) as a function of read dataset size.

In terms of scalability, the SMM algorithm lends itself very well to the additional cores and memory common in modern desktop and high-performance computers. As models and reads are loaded entirely into memory for scoring, I/O bottlenecks are minimized during concurrent runs. A significant reduction in runtime in processing a dataset of 800,000

reads by 12th order SMM was achieved when all 12 threads of a Ryzen 1600 were used and the dataset was split and each instance was run concurrently (from 12 hours to 3.5 hours). Obviating the need to set up a model database, which required ~200 Gb for PhymmBL installation, allows usage of valuable space on high-performance solid-state drives, upon which SMM's performance can be improved even further.

2.4.1 Comparative Assessment of SMM with BLAST

The intended use of our alignment-free approach is for classifying metagenomic reads originating from organisms that are not represented at lower taxonomic levels (e.g. species) but at higher taxonomic levels in the current genomic databases. To demonstrate the limitations of alignment-based approaches in such scenarios, we performed alignment between each read and genomic fasta sequences in the GenBank database using BLASTn and considered the best blast hit as the taxonomic identity predictor for the read (Figure 2.4, SUPPLEMENTARY FIG. 2.10). Species-masking was done by disregarding any best hits resulting from genomic sequences from the same species as of the read being analyzed. SMM12 provided over 20% higher classification accuracy versus BLASTn at the Phylum level for 100nt reads (Figure 2.4), and notably, SMM outperformed BLASTn on both 100 nt and 250 nt datasets at all taxonomic ranks with and in the absence of simulated Illumina sequencing errors (Figure 2.4, SUPPLEMENTARY FIG. 2.10).

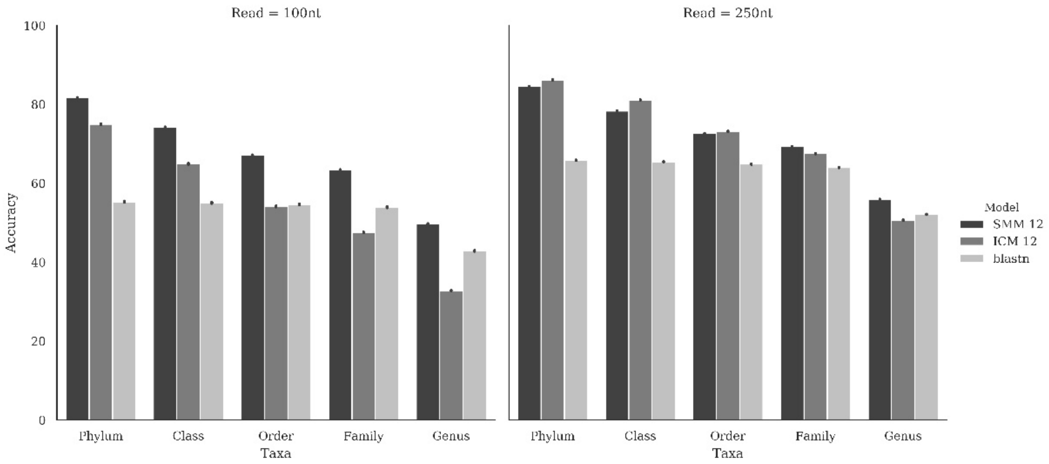

Figure 2.4: Comparative assessment of SMM of 12th order with BLASTn (nucleotide BLAST). Accuracies in classifying reads of length 100 nt (left) and 250 nt (right) are shown at different taxonomic ranks.

In addition to species masking, we assessed the performance by masking at other taxonomic ranks. SMM and ICM performed fairly similarly at higher masking levels, except for family-level masking, though this performance gap diminished significantly when Illumina error was introduced to the reads (SUPPLEMENTARY FIG. 2.11). Classification accuracies for class-level masking, the highest taxonomic rank masking performed, were higher for 12th order SMM compared to 12th order ICM, demonstrating further the ability of SMM to more robustly classify reads that originate from taxa, representing lower to higher taxonomic ranks, that are not represented in the database. Strain level masking highlighted the largest performance difference between the two models, with 12th order SMM outperforming 12th order ICM by up to 40% in classification accuracy (SUPPLEMENTARY FIG. 2.11).

2.4.2 Application to Real Metagenomic Data

12th order ICM and SMM, as well as IMM and DIM at their default settings (8th order), were applied to characterize a mouse fecal metagenome (Leinonen et al., 2011). A vast majority of mouse fecal metagenome reads (> 80%) were classified to the same three phyla by all four model types (SUPPLEMENTARY FIGS. 2.12a-d). The abundances of reads assigned to these phyla, namely, Firmicutes, Proteobacteria, and Bacteroidetes, were also similar between these model types (SUPPLEMENTARY TABLE 2.1). A notable phyletic discrepancy among all four model types was the number of reads assigned to the phylum Chlamydiae (SUPPLEMENTARY TABLE 2.1). SMM's Chlamydiae prediction (81 reads, ~0.08%) is closer to the original study of this microbiota, which didn't identify this phylum in the taxonomic profiling of the metagenome (Dey et al., 2015). DIM assigned the next lowest number of reads to this phylum; however, it was over ten times higher than that by SMM. Significant discrepancies were observed at the genus level, e.g. SMM assigned ~15K reads to *Bacteroides*, more than double the reads assigned to this taxon by ICM, and over 12K reads assigned to *Enterococcus* by SMM compared to ~8K by ICM. Although validation on real datasets is difficult or not even feasible, extensive validation experiments afforded by simulated metagenomes suggests that SMM may likely be classifying metagenomic reads more robustly than ICM; perhaps the genus-level classification accuracy of 12th order SMM on a simulated dataset was over 95%, 25% higher than by 12th order ICM (SUPPLEMENTARY FIG. 2.11a).

2.4.3 Efficacy of Whole-Taxon Models

The speed of SMM allows for the construction of whole-taxa models in a reasonable time frame and memory requirements. To test the potential of such models, a fasta file for each taxon represented in our database was constructed by concatenating every genomic fasta file from an organism belonging to that group. SMMs of order 12 were then constructed using the concatenated fasta files, and used to classify the same 100 nt and 250 nt full RefSeq datasets used in our prior accuracy tests.

Compared to models built from isolated genomes, whole-taxa 12th order SMMs failed to produce higher accuracy regardless of the taxonomic level of the collective model. Models built at the species level performed the closest, approximately ~4-5% less accurate than models built from isolated genomes. More encompassing models, such as those built upon the combined genomes of entire taxonomic genera, performed much worse than isolated genomic and collective species models. For 250 nt reads, the accuracy of read family classification decreased to ~23.4% compared to 64.9% and 69.6% accuracy in collective species and isolated genomic models, respectively. This performance gap appeared to widen, regardless of read length, as the taxonomic level of collective models was increased to represent more genomes (SUPPLEMENTARY TABLE 2.2).

2.5 Discussion

We show here that a simple implementation of a Markov chain model outperforms sophisticated models in taxonomic classification of short metagenomic reads of size 100 nt often generated by next generation sequencing platforms. This was found true in general with 8th order SMM even when the read length was increased to 250 nt, and also when the performance was reassessed for different GC range datasets. ICM's performance was worst among all models, likely because it couldn't obtain enough informative sites in the context sequences with its 8th order implementation to gain better discriminative ability. Extensive benchmark experiments presented here demonstrate the limitations of interpolation techniques in generating superior models for short read classification, rather a simple pseudo-count approach to address zero count oligomers renders SMM amenable to such analysis and our results highlight the advantages of pseudo-count supplemented SMM over other models. In addition to better classification of short reads, a straightforward and simple implementation of Markov model results in significant reduction in computational time. A simplified codebase allows for additional compiler optimization, and further augments the performance of SMM in taxonomic classification.

Our results show that SMM's accuracy increases with model order on 100 nt read sets, with the best performance obtained at order 12, which was the highest order we could implement in this study. It is possible that models of order higher than 12 may perform even better on short read classification; further computational advances could make possible implementation of such models for metagenomic sequence analysis. Our analysis also revealed that among SMMs, models of lower orders (8 – 11) may be optimal for higher taxonomic level classification (Class or above) when analyzing longer reads (Figure 2.2). In fact, the interpolation was found most effective in these instances- ICM of 12th order outperformed SMMs on 250 nt datasets at Order and higher taxonomic levels, specifically when the GC content was <55% (SUPPLEMENTARY FIG. 2.9). This highlights the conditions under which interpolation becomes effective in classification; longer reads may allow the power of interpolation to manifest by accounting for more of *k*-mers in the reads that are rendered reliable predictors of the succeeding nucleotides by such models. However, 9th- 12th order SMMs were more accurate than 12th order ICM in classifying reads generated with Illumina-based error profiles, regardless of the read length or composition (%GC) (SUPPLEMENTARY FIG. 2.8). This reveals the potential limitations of ICM in real scenarios, where simpler models such as SMM may be amenable to more robust classification of metagenomic reads that are degraded by sequencing errors.

For many practical purposes, classifiers that can reliably classify short sequences at lower taxonomic ranks (family or below) are desired. SMMs, particularly HOMs, clearly emerged as the model of choice for classifying metagenomic sequences at lower ranks, as evident from superior performances on both 100 nt and 250 nt datasets at lower ranks. Lower level classification (e.g. family or genus) requires attaining finer resolution for discriminating between apparently similar sequences, which may be provided by more informative HOMs. A substantial increase in accuracy was observed when read size was increased to 250 nt from 100 nt (Figures. 2.1, 2.2). Although there are voluminous metagenomic data with reads of size below 250 nt that call for development of methods for robust analysis of short reads, recent advances in sequence technology are enabling generation of longer reads that may be classified more robustly by HOMs and coupled with computational advances that could enable implementation of HOMs beyond 12th order, we expect the accuracy in read classification at lower ranks to improve substantially in the near future. Use of LOMs gives the advantage of significant reduction in computational complexity that results from exponentially decreasing number of model parameters to be trained as the model order is decreased. Higher rank taxonomic classification may be desirable for certain metagenomic samples inundated with unculturable bacteria, given that most unculturable prokaryotes are believed to belong to yet uncharacterized genera (Hofer, 2018; Lloyd, Steen, Ladau, Yin, & Crosby, 2018).

Notably, alignment-free (e.g. SMM) and alignment-based (e.g. BLAST) methods have complementary strengths in metagenome profiling. Previous studies have discussed the advantages of alignment-based approach in classifying reads when there is representation of read originating genomes in the database, otherwise their performance declines sharply, however, the alignment-free approaches perform well in this scenario (Pham & Zuegg, 2004;

Vinga & Almeida, 2003; Zielezinski et al., 2017). Our results further reinforce these findings from the previous studies. We show here that, on species masked test sets, how the classification accuracy increases with higher ranks through the use of Markov models, but stagnates for alignment (Figure 2.4). Classification using BLASTn shows that accuracy increases up to family level and then begins to cap off from order level onwards. At this level of phylogenetic distance, the homologous nucleotide sequences from related organisms could have diverged to an extent that local alignments are deemed insignificant and therefore not returned as "hits" by BLASTn, or even if returned as hits, their scores could be similar or worse than those from random (wrong) hits and therefore, the misclassified reads from lower taxonomic levels do not get assigned correctly at higher levels by BLASTn. In contrast, alignment-free methods such as SMM are not constrained by such limitations of alignment-based approaches, as is borne out in this study; these methods could encode subtle evolutionary signals as compositional biases (short-range dependencies as in oligomer distributions), bypassing assessment of long-range conservation that may be elusive (or below the detection threshold) at larger evolutionary distances and therefore could work better in scenarios where alignment methods might not.

Current implementation of SMM has made possible application of higher order models for large-scale analysis of metagenomic sequences. Future studies could focus on integration of alignment-free and alignment based methods, as has been demonstrated in PhymmBL, for more robust profiling of metagenomes, and on addressing challenges brought by evolutionary phenomena such as horizontal gene transfer that can result in highly similar nucleotide sequences present in otherwise phylogenetically distant taxa (Boto, 2010; Juhas et al., 2009;

Koonin, 2016). In addition, other informative features may also be incorporated in development of robust programs for metagenome profiling, e.g. use of paired-end reads, where available, to improve classification by examining whether taxon assignment for both reads is consistent, and in cases where not, the plausible scenario of horizontal transfer may be further evaluated. Furthermore, assembling longer sequences from short reads could augment metagenomic sequence classification, still an evolving technology that must account for chimerism in microbial genomes, which renders assembly difficult particularly for prokaryotes.

We have shown here the advantages of using HOMs in metagenomic read classification and how this can complement the frequently used alignment-based methods. A more efficient, rapid implementation of HOMs was presented here, thus contributing a new metagenomic read classifier for large scale analysis of metagenome data. Our comprehensive benchmarking experiments using a full array of genomes available in the NCBI GenBank repository demonstrate the usefulness of this method in lower taxonomic rank classification regardless of the heterogeneity of metagenomes (e.g. GC-content or read length).

CHAPTER 3

POSMM: AN EFFICIENT ALIGNMENT-FREE METAGENOMIC PROFILER THAT COMPLEMENTS ALIGNMENT-BASED PROFILING*

3.1 Introduction

Shotgun metagenomics is becoming increasingly popular in profiling the taxonomic composition of microbial communities. Wherein the early applications sought to identify the members of microbial communities, further advances in sequencing technologies and analysis tools are uncovering new information that shines a light on hitherto unknown facets of microbiotas and at the same time elicits new questions that call for more enquiries into microbiotas and therefore further interrogation of the metagenomic data. In contrast to 16S based approach where the focus is on sequencing only 16S rRNA genes of a community, shotgun metagenomics strives to sequence the entire nucleotide complement of a microbial community. While the debate remains open on the efficacy of metagenomic profiling through 16S sequencing versus whole metagenome (shotgun) sequencing (WMS) (Jovel et al., 2016; Shah, Tang, Doak, & Ye, 2011), it is beyond question that WMS allows for functional profiling by targeting the entire genomic repertoire of culturable and unculturable organisms in a community.

The increased complexity of WMS datasets demands development of more advanced methods for taxonomic profiling. Such tools are tasked with determining the taxonomic identities of individual reads arising from taxa that may or may not have representation in the

*This chapter is reproduced from Burks, D. and Azad, R. K. (2020). POSMM: An efficient alignment-free metagenomic profiler that complements alignment-based profiling. To be submitted. Authors retain copyright.

genome databases. Sequence alignment, the standard approach to inferring the origins of nucleotide fragments, can establish the taxonomic identity unambiguously only if the read originating organism is represented in the database. Despite the limitations, alignment algorithms such as BLAST have remained the mainstay in taxonomic classification of metagenomic reads (Ladunga, 2017). Metagenomic classification through local alignment has been augmented by developing extensions of BLAST, such as HS-BLASTN and DIAMOND, which prioritize speed to handle the increasingly cumbersome size of emerging WMS data (Buchfink, Xie, & Huson, 2015).

Despite the development of ultrafast alternatives to BLAST, the sheer size of metagenomic data has reoriented the focus of alignment towards hyper-fast exact-match for queries of distinct *k*-mers composing the reads (Ounit, Wanamaker, Close, & Lonardi, 2015; Wood & Salzberg, 2014). In 2017, the typical size of a WMS dataset was estimated between 1 and 10 Gbp (Quince, Walker, Simpson, Loman, & Segata, 2017), and has continued to grow as the associated costs and technological hurdles of sequencing shrink. The advent of third-generation sequencing has further amplified this big data problem in metagenomics (Mikheyev & Tin, 2014; Patel et al., 2018; Thakkar, Sabara, & Koringa, 2017), and the traditional alignment-free approaches adapted for use in metagenomic taxonomic classification are continuously being rendered obsolete despite typically offering higher sensitivity across large phylogenetic breadth (Brady & Salzberg, 2009; Menzel, Ng, & Krogh, 2016; Tello-Ruiz et al., 2016; Wood & Salzberg, 2014).

Alignment-free methods offer a more robust higher level of taxonomic abstraction for metagenomic sequences compared to methods based on alignment, particularly when the query read originating genome is elusive (Brady & Salzberg, 2009; Burks & Azad, 2020a). Recent years have seen a resurgence of Markov model based methods for metagenomic classification (Burks & Azad, 2020b; Song, Ren, & Sun, 2019; Wang, Hu, & Li, 2016). While no current Markovian approach outpaces the optimized alignment schema of tools such as Kraken or CLARK in terms of the turnover rate (Ounit et al., 2015; Wood & Salzberg, 2014), new optimized algorithms have brought Markov models back as a realistic alternative with reasonable runtimes in the context of WMS analysis (Burks & Azad, 2020). While classification speed is certainly important, accuracy is paramount, and the metagenomic scientific community should have the options to choose one over the other, or perhaps the best trade-off between the two, based on their priorities and needs.

One of the biggest hurdles in taxonomic classification, particularly for reads where the closest identified relative may not share even a single oligomer of reasonable length, is the estimation of confidence for matches. The probabilistic scoring by Markov models does identify the best matching model (genome) to the read, but offers little beyond this. Whether the best hit represents the source organism the read originated from is always in question as this does not provide insight into the strength of the relationship between a model and the read. A frequently used Markov model based program, PhymmBL, introduced polynomial functions accounting for read length, Markov model order, and taxonomic level to generate confidence scores in later revisions of the software (Brady & Salzberg, 2011a), though the underlying methodology was not clearly laid out. Alignment based program Kraken2 offers a classification score based on the frequencies of taxon-specific *k*-mers, but can vary greatly with the database used, and quickly becomes restrictive, particularly for taxa with highly similar *k*-mer representations (Wood, Lu, & Langmead, 2019a).

Combining complementary methods has seen success as a strategy to raise the accuracy bar in taxonomic classification. PhymmBL is an example of such an approach that exploited the complementary strengths of interpolated context models (ICMs) generated by GLIMMER (Salzberg, Delcher, Kasif, & White, 1998) and the local alignment with BLAST (Altschul, Gish, Miller, Myers, & Lipman, 1990) within an integrative framework to classify reads with higher sensitivity and precision than by either of the standalone programs. Such combinations work best when the strengths of each individual method can address the weaknesses of the other. For modern classifiers built on exact *k*-mer alignment, precision can be very high. Sensitivity, however, is a usual weakness, with tools such as Kraken failing to align over 68% of reads from real metagenomic datasets (Wood & Salzberg, 2014).

In what follows, we introduce and describe a new metagenomic classifier, POSMM, named after Python-Optimized Standard Markov Model algorithm. POSMM leverages higher accuracy of alignment-free, Markov model based approach for taxonomic abstraction as both a standalone program and a component program for WMS read classification. Building Markov models of genomes and scoring of reads by the trained models are executed by our previously published standard Markov model (SMM) based algorithm (Burks & Azad, 2020b), allowing the end-user to select the model order and therefore control the accuracy and CPU time trade-off (computationally demanding higher order models tend to be more accurate, however, this may not be always true). The taxonomic classification of reads is performed based on a regression-based probability score derived from simulated read data. The training dataset was assembled by proportionately sampling from genomic regions with distinct compositional signatures for each prokaryotic species represented in the database; precautions were taken to remove the contaminant sequences that may distort the conclusions of our machine-learning process. This was achieved by employing Segmented Genome Model (SGM) based program, a new C++ incarnation of an integrated segmentation and clustering program that can rapidly segment genomes and group compositionally similar segments into distinct clusters for each genome (Azad & Li, 2013; Jani & Azad, 2019; Jani, Mathee, & Azad, 2016).

3.2 Results

Underlying POSMM is a modified version of the original SMM algorithm previously found superior in both classification accuracy and computational performance to legacy Markov model variants (Burks & Azad, 2020b). The SMM algorithm was used to build higher order Markov models (order 10-12) of each genome. Each read from a metagenomic dataset is then "matched" against the genome models by computing the probability of the read to be generated by each model. Thresholds for predicting the lineage (different taxonomic ranks) were established based on taxon-specific logistic regression models (see Methods). POSMM and Kraken were assessed on both simulated and real metagenomic data, and furthermore, a combined framework of POSMM and Kraken was benchmarked on the same datasets, as described below.

3.2.1 Real, Mock, and Simulated Metagenomes for Classification Accuracy Assessment

To benchmark the performance of alignment-free POSMM relative to alignment-based Kraken2, we used the simulated metagenomic test datasets as used in both the Kraken and CLARK benchmarks (Ounit et al., 2015; Wood & Salzberg, 2014). The HiSeq and MiSeq datasets represent reads assembled from sequencing projects of isolated genomes, whilst the simBA-5 dataset features bacterial and archaeal reads with 5X the error rate expected in metagenomic sequencing.

A predefined mock metagenome, developed as part of a study comparing metagenomic sequencing methods (Sevim et al., 2019), was also used to compare the performance of Kraken2 and POSMM, as well as a hybrid of both the programs. Developed from Illumina sequencing of a synthetic microbial community, the full dataset consists of over 213 million paired-end 151 bp reads. The size of this dataset makes it computationally prohibitive for alignment-free classification methods, such as NBC and PhymmBL (Wood & Salzberg, 2014). Only one of the reads of each pair was used for classification. Comparison was also made to directed alignment performed in the original study using the bwa aligner and the reference genome of each species in the synthetic community (Li & Durbin, 2009; Sevim et al., 2019).

We also analyzed two real human saliva metagenomes that were earlier used in the Kraken and CLARK benchmarks (Ounit et al., 2015; Wood & Salzberg, 2014). As with the simulated and mock metagenomes, over 20% of reads within these datasets were not classified by Kraken2. Using the custom GenBank database, we classified reads in each dataset with Kraken2 and POSMM. Reads that couldn't get classified by Kraken2 were re-analyzed with POSMM to assign taxonomic identities to reads otherwise deemed 'unclassifiable'.

3.2.2 Establishing Score Cutoffs for Classification

Kraken2 provides confidence scores for thresholding. POSMM's thresholding is based on probabilistic scores produced by logistic regression models. To evaluate the effects of score cutoffs on classification precision and sensitivity, reads of each simulated metagenome were classified at different cutoffs, ranging from 0 to 0.75 (Figure 3.1). In all cases, sensitivity and precision were calculated as established in the Kraken and CLARK studies (see Methods; Ounit et al., 2015; Wood & Salzberg, 2014).

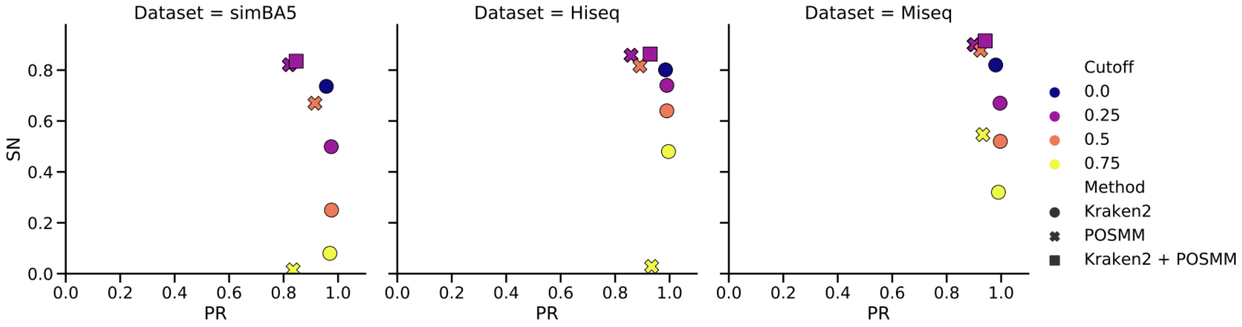

Figure 3.1: Scatterplot of the genus-level SN and PR for POSMM, Kraken2, and a combined Kraken2/POSMM analyses of three simulated metagenomes when using the confidence score of each program as a threshold. For the combined analysis, Kraken2 and POSMM were used with 0 and 0.25 confidence score thresholds, respectively.

For mock and real metagenome analysis, confidence score threshold was not used with Kraken2. For genus-level classification of reads of the simulated and mock metagenomes, Kraken2 performed best without any confidence thresholds (default setting that allows classification to lowest common ancestor based on the number of exact *k*-mer matches in a clade). POSMM performed best with modest cutoffs ranging typically between 0.2 and 0.4. As expected, higher cutoffs increased the precision of POSMM at the expense of sensitivity. For the hybrid of Kraken2 and POSMM, a cutoff of 0.25 was used for POSMM (default, performance at other cutoffs are shown in (Figure 3.1)). When analyzing reads from closely related species, we observed that the 0.25 cutoff did not offer any advantage over no cutoff where the taxon assignment was based on highest scoring genome model. However, when the dataset contains distantly related reads, beyond phyla, the use of cutoff was observed to improve classification. In general, POSMM emphasized sensitivity over precision, whereas Kraken2 emphasized precision over sensitivity. After performing initial analysis with Kraken2, POSMM can be deployed to classify reads that are left unclassified, or to provide more specific classifications to reads assigned only to higher taxa by Kraken2. This approach leverages the complementarity of Kraken and POSMM, that is, the speed and precision of Kraken and the sensitivity and capability to classify at different taxonomic ranks of POSMM.

3.2.3 Simulated Metagenome Classification Accuracy

To assess the classification performance of POSMM, Kraken2, and their hybrid, we used simulated, mock, and real metagenomes. Simulated and mock metagenomes allow for performance reporting, as the read identities are pre-established (simulated metagenomes) or narrowed to known members of the originating synthetic microbial community (mock metagenomes). Real metagenomes offer additional insights into the real-world applicability of POSMM and Kraken2, as well as the benefits of combining both approaches, but offer little in terms of the accuracy of either method.

For a fair assessment, we used the previously established test metagenomes, namely, the simulated metagenomes featured in Kraken and CLARK's original benchmarks (Ounit et al., 2015; Wood & Salzberg, 2014), as well as in other classifier-performance studies (Břinda, Sykulski, & Kucherov, 2015; Metwally, Dai, Finn, & Perkins, 2016). For simulated metagenomes, the precision and sensitivity were computed at different score thresholds as well as without a threshold. As expected, the precision of both POSMM and Kraken2 tended to increase for all three simulated metagenomes as the threshold was increased (Figure 3.1). This relationship was less apparent with Kraken2, which maintained a high precision at all cutoffs. The lowest precision reported by Kraken2, 0.957, was observed with the simBA-5 metagenome with no cutoff. However, this is just 0.018 less than the highest precision of 0.975 reported by Kraken2 for this dataset at 0.25 confidence score cutoff. POSMM, in classifying reads from the same simulated metagenome, had a broad range of precision, with the difference between the highest and lowest precision, 0.095, over 5 times that of Kraken2. The sensitivity metric was much more sensitive to threshold cutoff for both programs, dropping below 0.1 for both programs at the highest tested threshold of 0.75 (Figure 3.1). The lowest sensitivity for Kraken2 (0.08) was encountered when analyzing the simBA-5 metagenome at 0.75 cutoff; POSMM experienced the lowest sensitivity (0.0149) also with the simBA-5 metagenome at 0.75 cutoff. Substantial decline in sensitivity was also observed with the HiSeq metagenome at the 0.75 cutoff, indicating that despite increasing the precision, raising the threshold level leaves bulk of the reads unclassified.

Despite having a confidence score threshold option, the precision remained fairly constant with and without a threshold in place for Kraken2. The impact of the threshold on precision was more apparent with POSMM than Kraken2. We attribute this observation as a characteristic of the underlying platforms, wherein exact alignments are only possible when that level of similarity is present (Kraken), whilst there will always be a maximum Markov model score for a read regardless of the level of similarity.

The hybrid of POSMM and Kraken2 yielded the best overall performance (highest F1-score, the harmonic mean of precision and sensitivity), for all simulated metagenomes (Figure 3.1, Figure 3.2). By first analyzing each simulated metagenome with Kraken2, and then applying POSMM only to reads left unclassified by Kraken2, the highest F1-scores for all three simulated metagenomes could be achieved. The most pronounced improvement in performance was observed with the MiSeq dataset, where the hybrid approach yielded an F1-score 3% higher than the next best performing method (POSMM with a 0.25 cutoff). The effect was less obvious with the HiSeq and simBA-5 datasets (~1% increase in F1-score), but only when compared to Kraken2 without any confidence score thresholding. The increase in F1-score attained through the hybrid approach is mainly due to an increase in sensitivity contributed by POSMM. The hybrid approach produced the highest mean sensitivity and F1-score (averaged over all three simulated metagenomes). Average precision of the hybrid approach was lower than that of Kraken2 by ~7%, but was offset by a gain in sensitivity through POSMM resulting in the superior overall performance (Figure 3.2).

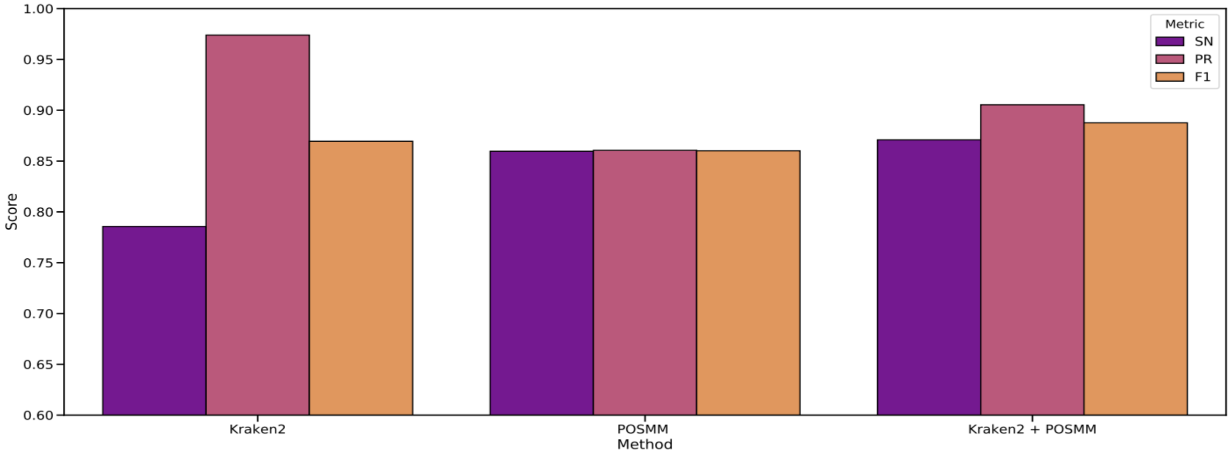

Figure 3.2: Bar plot of the genus-level SN, PR, and F1-score of all three methods averaged across the three simulated metagenomes (Hiseq, Miseq, and simBA-5). Kraken2 results are based on no applied confidence threshold. POSMM results are after applying a 0.25 confidence score threshold for classification. Kraken2 + POSMM results were generated as described in (Methods), wherein initial classification is performed by Kraken2 without cutoffs, followed by POSMM with a 0.25 confidence score threshold on the unclassified (at genus level) reads.

3.2.4 Mock Metagenome Classification Accuracy

Next the performance of POSMM. Kraken2, and their hybrid was assessed on the mock metagenome and also compared with the genome-directed alignments performed in the original study of the mock metagenome (Figure 3.3) (Sevim et al., 2019). No read filtering was performed prior to analysis with POSMM or Kraken2. Similar to the simulated metagenome results, Kraken2 left millions of reads unclassified at several taxonomic ranks as confidence thresholds were introduced. POSMM, as before, helped classify reads that were deemed 'unclassifiable' by Kaken2 or even by the original study (Sevim et al., 2019).

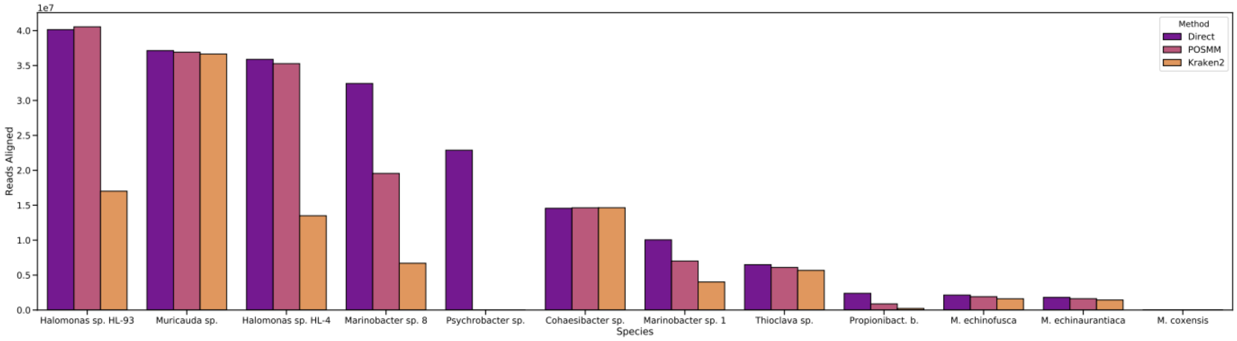

Figure 3.3: The number of reads assigned by POSMM and Kraken2 to each species present in the SRR8073716 mock metagenome. The direct method refers to the number of reads assigned to each genus in the original study using genome-specified bwa alignments. Results from POSMM are after applying a 0.25 confidence score threshold. No threshold was applied to Kraken2.

Kraken2 didn't do well in assigning reads belonging to the *Halomonas* genus, which constituted the ~37% of the mock community, specifically in assignment of the reads to either of the two *Halomonas* species (HL-93, HL-4), or to the genus itself. On the other hand, POSMM's read assignment matched closely with the genome-directed bwa alignments in the original study. The *Psychrobacter* species of the mock community (LV10R520-6) was not represented in the genome database used for Kraken2 and POSMM, and as expected, reads from this species were misclassified at the species level. Genus level classification was also not up to the mark (Figure 3.4), despite the inclusion of 58 unique *Psychrobacter* species in the database. POSMM aligned more reads specifically to the two *Marinobacter* species genomes (LV10MA510-1 and LV10R510-8), and this was also reflected at the genus level, where POSMM shared more read alignment to these taxa with the direct-alignment (~42 million reads) than Kraken2 with the direct-alignment (~11 million reads).

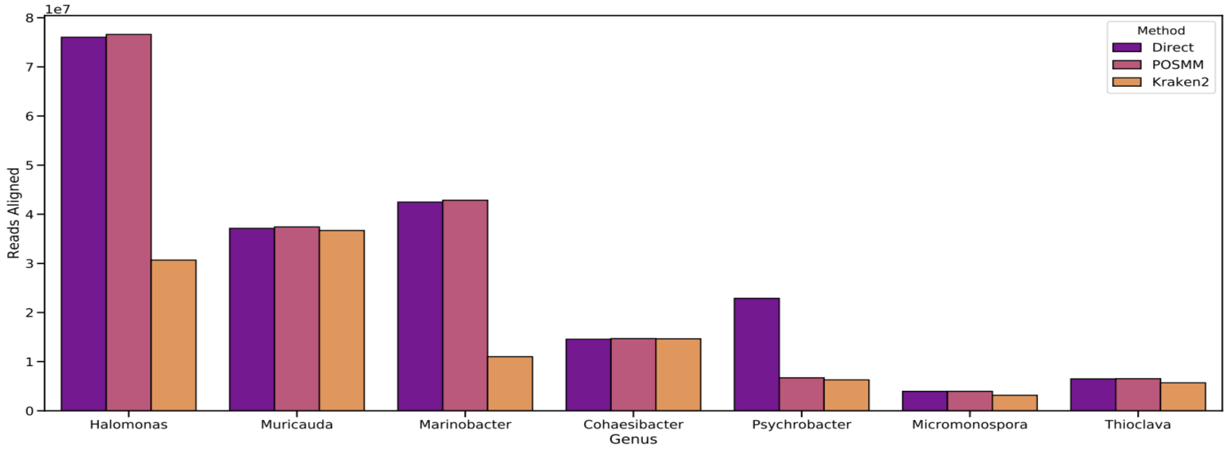

Figure 3.4: The same data as presented in Figure 3.3 but at the genus level. Only species known to be part of the mock metagenome are present in the graph. The Kraken2 + POSMM combined analysis consisted of a full Kraken2 analysis with no confidence score threshold, followed by reanalysis of reads unclassified by Kraken2 at the species level by POSMM with a 0.25 confidence score threshold.

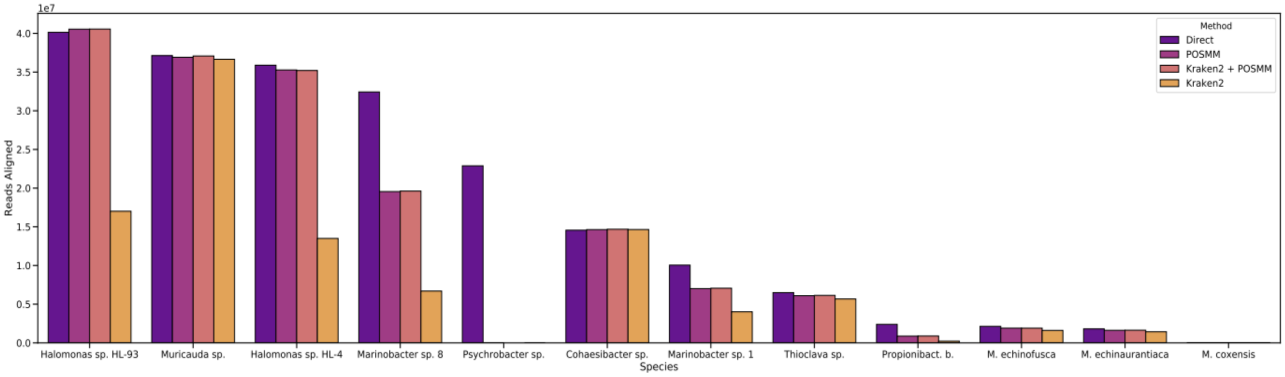

Figure 3.5: The same data as presented in Figure 3.4 but at the species level. Only species known to be part of the mock metagenome are present in the graph. The Kraken2 + POSMM combined analysis consisted of a full Kraken2 analysis with no confidence score threshold, followed by reanalysis of reads unclassified by Kraken2 at the species level by POSMM with a 0.25 confidence score threshold.

Interestingly, combining POSMM and Kraken2 very closely resembled the results of POSMM standalone (Figure 3.5). As before, the entire mock metagenome was first analyzed with Kraken2 without a threshold. Reads that were not classified to a species by Kraken2 were then reanalyzed by POSMM with a 0.25 score cutoff, and taxonomic classifications were merged. Given the size of the dataset, this led to a dramatic decrease in POSMM analysis time, as Kraken's first-pass analysis filtered out over 51% (over 109 million reads) of the dataset.

3.2.5 Real Metagenome Classification Comparison

We used both Kraken2 and POSMM to characterize the communities of two human microbiome samples previously featured in multiple metagenomic classification benchmarks (Ounit et al., 2015; Wood & Salzberg, 2014). Unlike previous assessments, our full GenBank database was used for read classification by both Kraken2 and POSMM. Datasets SRR062462 and SRR062415 are both of human saliva samples, and were filtered for human contaminant reads prior to the analysis. Quality-trimming and additional filtering were performed, removing low quality bases and adapter remnants.

The proportions of genus classifications were similar between Kraken2 and POSMM (SUPPLEMENTARY FIGS. 3.1-3.6). As with the simulated and mock metagenomic datasets, Kraken2 left a significant number of reads unclassified (>277,000 reads, >20%), which were assigned by POSMM. Despite the difference in total read assignments, the proportions of taxon assignments were similar between these classifiers. In agreement with the prior analysis (Wood & Salzberg, 2014), *Streptococcus*, *Haemophilus*, and *Prevotella* genera represented the majority of reads for both programs (SUPPLEMENTARY FIGS. 3.1-3.6).

To investigate the reads left unclassified by Kraken2, we filtered reads that were not assigned to any taxon by Kraken2. These reads were subjected to classification by the hybrid of Kraken2 and POSMM, at POSMM cutoff of 0.25. Taxonomic classification of the formerly unclassified reads was fairly spread out across multiple genera. *Bacillus*, the genus with least number reads assigned to by Kraken2, had now 12,413 additional reads assigned to it by the hybrid program (4.34% of all unclassified reads in the SRR062415 dataset). *Streptomyces* that had only 233 reads assigned to by Kraken2, was assigned 9745 additional reads by the hybrid program (3.41% of all unclassified reads). Interactive diagrams of the classifications by each method, as well as the POSMM classification of Kraken2's unclassified reads, built using Plotly and compatible with modern internet browsers, are provided as supplementary html diagrams (SUPPLEMENTARY FIGS. 3.1A-3.6A).

3.2.6 POSMM Runtime

POSMM runtime is dependent on model order and the number of models used to score the metagenomic reads. We examined POSMM's time to completion on a single core versus all 6 physical cores in the use of a Ryzen 1600 system. The runtime as a function of dataset size, number of models used, and threading is shown in (Figure 3.6). Dataset size (in reads) had little effect on POSMM's total runtime.

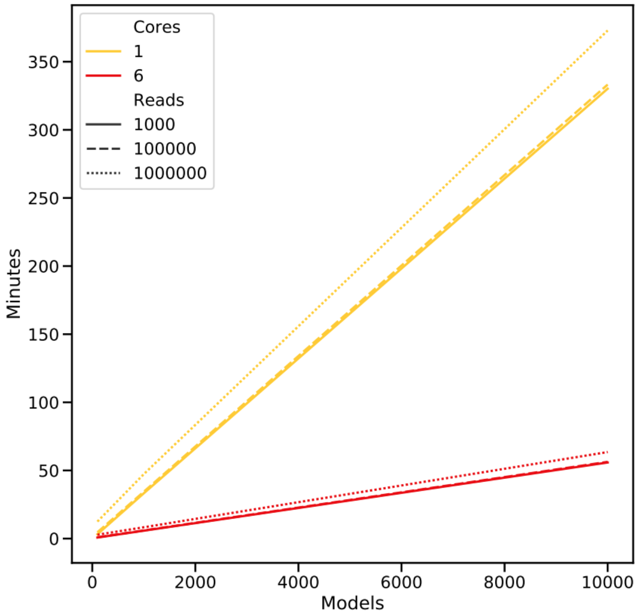

Figure 3.6: Line plot showing the runtime (in minutes) of POSMM based on the number of models, number of 100nt reads to analyze, and core count.

3.3 Discussion

POSMM echoes the higher sensitivity in taxonomic inference of traditional alignment-free metagenomic classifiers (Brady & Salzberg, 2009; Rosen et al., 2008). By simplifying the Markov model based approach to taxonomic classification (Burks & Azad, 2020b), POSMM circumvented the computational time barrier that has made several alignment-free metagenomic classifiers obsolete as the dataset size continues to grow (Wood & Salzberg, 2014). While POSMM lacks the speed of *k*-mer aligners, it does have the speed and scalability to analyze large metagenomic datasets produced by current sequencers. As an accompaniment, POSMM offers to augment the sensitivity of faster though less sensitive alignment based metagenomic classification programs. By obviating the need for establishing model databases, made possible by generating models directly from genomic fasta files on the fly, POSMM ushers in a new approach that can be easily adapted and restructured to fit with specific needs in classification.

POSMM is also highly scalable. The memory footprint is entirely based on the size of the dataset being analyzed, as metagenomic reads are indexed and kept in memory for rapid lookup during score computation. Users with less resources can split datasets as needed, allowing POSMM to run on devices ranging from power-efficient laptops to high-performance computing environments. As the number of CPU cores continues to increase on desktop computers, the potential throughput of POSMM should also scale linearly. The simplified underlying codebase for generating SMMs, written in C++11 and only using standard libraries, is also easily portable to the increasingly common ARM architecture that continues to expand beyond use in mobile phones. The regression score models, which are built using the popular scikit library and stored in JSON, are also easily modifiable. Being able to easily adjust the score models to scale to an ever-changing and rapidly growing databases, POSMM holds the promise to remain relevant in many years to come.

3.4 Materials and Methods

3.4.1 Database Generation

Developing a fully inclusive database is essential for training and testing any taxonomic classification method. Only including the highest quality genomes can give uncharacteristic advantages during benchmarks that may not be reflected in real world applications. While Kraken2 maintains a robust standard database and a prokaryotic database, many of the genomes in the mock shotgun dataset (Sevim et al., 2019) and identified in the real metagenomes were not present in either.

POSMM's speed is dependent upon the number of models (i.e. genomes) being queried. To keep analysis within a reasonable time window and give all species with sequenced genomes equal representation without redundancy, we developed a priority system for collecting representative genomes for all species currently available in NCBI GenBank. First, the archaeal and bacterial assembly summaries were downloaded from the RefSeq release FTP site (https://ftp.ncbi.nlm.nih.gov/refseq/release/). Taxid numbers were used to isolate unique species and then a representative genome for each species was obtained. Using the NCBI RefSeq terminology, included with the assembly summary, we selected 'Reference' genomes where available, otherwise 'Representative' genomes. When neither of Reference and Representative genome was available, the decision was based on the assembly level in the order of 'complete', 'chromosome', 'scaffold', and finally 'contig'. Species with only partial representation of their genomes were not included in our custom database. In the event of a tie, one genome was randomly chosen using a random number generator from the Python standard library.

The custom database is comprised of genomes of 29,870 unique species. These genomes represent various quality levels; partial genome assemblies were not included. Because of variable quality of the genome assemblies, each genome was subjected to filtering for potentially extraneous sequences. The same type of genome set can be downloaded using the POSMM “--runmode setup” and “--gtype bacteria/archaea” parameters.

3.4.2 Real and Mock Metagenome Processing

WGS reads from male human saliva samples (NCBI SRA accessions SRR062462 and SRR062415) were downloaded using the fastq-dump utility from the sratools suite (Leinonen, Sugawara, Shumway, & International Nucleotide Sequence Database Collaboration, 2011). Reads were trimmed to remove low quality bases and adapter sequences using fastp (Chen, Zhou, Chen, & Gu, 2018). Human DNA sequences were removed by aligning reads to the GRCh38 *Homo sapiens* genome using bwa 0.7.17-r1188 (Li & Durbin, 2009). Reads that aligned to the human genome were manually removed using ad-hoc scripts following analysis of the output BAM file. Sunburst diagrams for taxonomic classifications were generated using the plotly library for Python 3.8.

NCBI SRA accession SRR8073716, representing an Illumina-sequenced metagenome from a previously published mock microbial community (Sevim et al., 2019), was also downloaded via fastq-dump. No read processing was performed prior to analysis by either program (Kraken2, POSMM). Direct genome read alignment counts were taken from the supplementary files of the original study (Sevim et al., 2019).

3.4.3 Markov Model Classification Algorithm

POSMM allows users to build standard Markov models or SMMs of orders 10-12 (Burks & Azad, 2020b) for each genome using the genomic sequence fasta files. First, an empty count distribution of the specified order is filled with pseudocounts. At the start of each run, an "empty" probability distribution is also built, representing the initial and transition probabilities for the specified model order *k*. Both initial and transition distributions are kept global and are reset as each genome is modelled. In this way, metagenomic fasta files can be indexed respective to the relevant positions of the global probability distribution by using a vector of memory-address pointers. Maintaining a static location in memory and changing the probabilities per genome minimize the memory footprint and avoid I/O bottlenecks. The average model build time for a prokaryotic genome is typically less than 3 seconds, but can be further minimized by storing genomic data on high-speed NVMe or RAMDisk drives.

To speed up the throughput, POSMM splits genome sets for modelling based on the specified CPU core availability, and run concurrent analyses of the same metagenomic fasta file. This is faster than multi-threading the reads being analyzed, and takes full advantage of the increasing RAM availability of the modern computing environments. The biggest bottleneck of SMM, and by association, of POSMM, is on-the-fly generation of Markov models of genomes, however, splitting this across multiple CPU cores bestows the highest performance gains to the user.

3.4.4 Machine Learning Derived Score

Most alignment-free metagenomic classifiers tend to assign taxonomic identity to all reads regardless of whether the source taxa for the reads are represented or not in the genome database used for classification (Brady & Salzberg, 2009; Rosen, Garbarine, Caseiro, Polikar, & Sokhansanj, 2008). This can lead to an inflation of misclassifications, particularly for reads originating from organisms whose genomes are not represented in the genomic databases. Higher taxonomic level classification could be more accurate as closely related genomes belonging to the same taxon may be represented in the database; however, even higher level classifications are not immune to this as a vast number of reads in a metagenomic sample may not their source representation even at higher taxonomic levels in the database. Alignment based methods have largely avoided this problem as alignment provides a confidence score for the similarity of the query read with a subject sequence in the database. This may reduce misclassifications resulting from ambiguous alignments (Ounit et al., 2015; Wood, Lu, & Langmead, 2019b; Wood & Salzberg, 2014). The developers of alignment-free classifier PhymmBL took a cue and attempted to address by introducing a confidence score akin to the alignment score (Brady & Salzberg, 2011b). Using simulated training data, 3D-curve fitting was applied in order to formulate a similarity score based on the read length, taxonomic level, and Phymm score. Thresholding based on this score was demonstrated to be effective in reducing misclassifications (Wood & Salzberg, 2014). However, later studies have suggested that using this score for thresholding can lower both sensitivity and specificity of metagenomic classification (Lan, Wang, Cole, & Rosen, 2012).

The fidelity of any fitting procedure is dependent upon the quality of the training data.

Poor taxonomic representation, or perhaps taxonomic overrepresentation, could explain why certain datasets seem to benefit the scoring schema of PhymmBL while others do not (Lan et al., 2012; Wood & Salzberg, 2014). To develop a more robust Markov model based scoring schema for phylogenetic classification, we employed logistic regression in combination with Bayesian optimization and cross-validation techniques. Furthermore, training data was sampled from the compositionally distinct fractions within each genome (Jani & Azad, 2019). Representation of compositionally disparate regions within genomes is vital for producing a reliable score. Attempts to generate higher order models of compositionally atypical regions didn't yield desired results as these regions were often relatively much small and therefore did not lend themselves well to generating reliable higher order models. The increase in the number of models also dramatically increased the POSMM's runtime. Isolating these regions, and having their representation in the training data, was deemed an effective approach for incorporating useful evolutionary information encoded within prokaryotic genomes.

3.4.5 Simulated Training Set Construction

Contaminations in GenBank genome assemblies are a documented problem. Contaminant sources, such as extraneous DNA or adapter sequences, must be identified and eliminated. On the other hand, horizontally acquired genomic regions are commonly present across prokaryotes and are integral parts of their genomes (Jani & Azad, 2019; Jani et al., 2016; Ochman, Lawrence, & Groisman, 2000). Not adequately accounting for these mobile elements in genomes could result in misclassification of a significant fraction of metagenomic reads.

In addition to horizontal gene transfer, genomic mosaicism may arise due to other evolutionary or biological factors (Jani & Azad, 2019). These compositionally disparate regions need to be accounted for in order to render a genome model that adequately represents the variability within a genome. For example, there must be distinct models representing horizontally acquired regions from distinct lineages and a model representing the vertically transmitted regions in a genome. Accounting for mosaic compositional structure of prokaryotic genomes is paramount to establishing a high-quality training dataset for regression. To address this, we used the Markovian Jensen-Shannon Divergence (MJSD) based segmentation and clustering method that has previously been applied to predict genomic islands in prokaryotic genomes (Azad & Li, 2013; Jani & Azad, 2019). This enabled isolation of compositionally distinct regions within each genome in our custom genome database. An optimized algorithm, based on the same methodology for segmentation and clustering as in IslandCafe (Jani & Azad, 2019) but designed to be computationally more efficient, allowed analysis of genomes at a rate capable of handling the entire RefSeq database on a single desktop computer within a reasonable time. For our test system based on a Ryzen 1600 CPU, our segmentation and clustering algorithm processes approximately 2 prokaryotic genomes per minute using all 6 physical cores. The new algorithm uses an optimized technique for computing entropies to estimate the divergence between DNA sequences through MJSD. The new algorithm uses a reverse-calculation step that allows rapid nucleotide-wise iteration across the entire genome (see below). This resulted in a 16-fold reduction of the average time for segmentation and clustering of a prokaryotic genome (average size ~5 Mbp), from over 41 minutes to approximately 2.5 minutes. For segmentation, we recursively iterated divergence computation at each position of the genome and segmented at the position with the highest MJSD between two resulting subsegments provided the associated p-value was less than 0.05. The significance threshold for clustering was set to 10^{-5} (readers should refer to Azad and Li, 2013 or Jani and Azad, 2019 for details).

Clusters less than 0.001% the size of the genome were discarded. The remaining clusters were queried for human, viral, and adapter sequence contamination using BLAST and those with significant similarity to these were also discarded. As segments within a cluster are compositionally similar, we expect these segments to generate more similar Markov model scores than the segments from different clusters. By using a random number generator, we generated fragments of random lengths between 30 and 500 bp from each cluster to generate labeled fragment sampling pools. Randomly sampling fragments from each cluster ensured representation of each compositionally distinct region in our training data. Multiple datasets of 250,000 reads were randomly sampled from these pools to generate 10 unique metagenomic training datasets for each taxon (phylum, class, order, family, genus, and species). By cycling through these datasets with a Bayesian optimization scheme (see below), we generated regression models that were used for taxonomic classification of reads as further discussed below.

3.4.6 Markovian Jensen-Shannon Divergence (MJSD) Based Segmentation and Clustering Algorithm

We were able to significantly reduce the runtime of genome segmentation and clustering algorithm, as implemented in IslandCafe (Jani & Azad, 2019), by introducing a reverse-calculation step during recursive segmentation. MJSD, entropy, and statistical significance were calculated as described in (Jani & Azad, 2019). Specifically, information content of a genome sequence, quantified by the entropy function for probability distribution *p*$_i$, is obtained as, $H^m(p_i) = -\sum_w P(w) \sum_{x\in\mathcal{A}} P(x|w) \log_2 P(x|w)$, where *P*(*x*|*w*) is the probability of nucleotide *x* given the preceding oligonucleotide *w* of length *m* (*m* defines the model order, is set to 2 in IslandCafe) and *P*(*w*) is the probability of oligonucleotide *w*. A genome is initially segmented by iterating the computation of entropy and thus MJSD at each position along the genome and identifying the location of highest MJSD of (user-defined) significance in the genome. This process is then iterated for the resulting genomic segments.

3.4.7 Augmenting Computational Efficiency of Segmentation and Clustering Algorithm

IslandCafe reduces the runtime by computing MJSD at every *l*/10000th position along the genome sequence of size *l* to be segmented, however, it computes afresh the probability parameters using the oligonucleotide counts for each MJSD computation. In contrast, we designed our new segmentation and clustering algorithm to iteratively computes MJSD at each nucleotide position in the genome. However, for each subsequent MJSD computation, rather than estimating the entropies afresh, the entropy values from the previous computation were adjusted based on only the oligonucleotides that have to be included and excluded in the current computation. This new approach is not only faster (over 16x faster segmentation and clustering of the same 4.7Mb *E. coli* genome), but also results in higher precision as MJSD computation is performed at each position, rather than *n*th position, in the genome.

3.4.8 Machine Learning Derived Score Models

To establish cutoffs based on probabilistic scores, we applied machine learning libraries to the raw SMM scores (Burks & Azad, 2020) of our sampled genomic fragments. We used Bayesian optimization to assign hyper-parameters for the logistic regression estimators of the scikit-learn library (Pedregosa et al., 2012) using the skopt BayesSearchCV module (https://scikit-optimize.github.io/stable/modules/generated/skopt.BayesSearchCV.html). We also tested SVM estimators with a linear kernel, however, the accuracy was on average lower than the logistic regression accuracy for all taxa, and the SVM estimators were found to be prone to overfitting.

Training data containing raw scores outputted by SMM, read lengths, and classification accuracy (True/False) from the top 50 scores for multiple 250,000 read simulated datasets were obtained following SMM analysis at 10th, 11th, and 12th order. We focused on the top 50 scores for each read of our simulated data, as this maintained a balance between the number of correct and incorrect classifications for our regression analysis. Model order and taxonomic rank specific training datasets were obtained, and individual regression models were optimized for phylum, class, order, family, genus, and species levels at 10th, 11th, and 12th model orders. Additional variables, such as read %GC, model %GC, and read entropy, were tested as potential training features, but they added unnecessary overhead with no appreciable gain in classification accuracy following the model training.

The scikit-optimize BayesSearchCV function allows for parameter optimization and model fitting using a “fit” and “score” method. A 3-fold cross validation was performed; the training data was randomly split into 3 groups during each optimization test. The first two sets were used for model training and validation respectively at various parameter combinations and the third set was used for testing the trained regression model. Performance was assessed by applying the trained regression models to the test data and determining the classification accuracy. Unlike grid search optimization, which tests all possible combinations of hyper-parameters, Bayesian optimization adjusts hyper-parameters based on prior performance results. Users are required to set static values or ranges for the model being optimized. We used the sklearn.linear_model.LogisticRegression module of scikit-learn as our model generator, and kept settings for dual formulation and 15,000 iterations constant for each training session. Otherwise with dual=false and lower iteration values, the logistic regression classifier may fail to converge. The inverse regularization parameter, referred to as the C parameter in scikit-learn, was sampled at values ranging from 1e-6 to 1e5. The tolerance value parameter was sampled with values ranging from 1e-7 to 1e-2. The L1 and L2 penalty norms, ‘liblinear’ and ‘saga’ solvers, and intercept fitting booleans were cross compared for various combinations by the BayesSearchCV function. Optimized parameters are included in (SUPPLEMENTARY TABLE 3.1), and guided final model building. All regression models were exported and stored in JSON format using the sklearn-json library.

While these models confer the ability to predict taxonomic identity, the predict_proba function of the final models provides probabilistic score (value in range 0-1) for thresholding, allowing users to prioritize precision over sensitivity at increasing stringencies.

3.4.9 Sensitivity, Precision, and Score Calculation

Sensitivity and precision were calculated as described in Kraken’s and CLARK’s benchmark studies (Ounit et al., 2015; Wood & Salzberg, 2014). In some cases, a genome may not have a taxonomic label for all ranks (species, genus, family, etc.); previous benchmarks have established taxon level accuracy, e.g. genus-level sensitivity is computed as *A*/*B* where *A* is the number of reads with the genera correctly assigned by a method and *B* is the total number of reads of known genera. Sensitivity was calculated similarly for all other taxonomic ranks.

Precision is also based on the definition established by prior benchmarks, wherein the genus-level precision is calculated as *X*/(*X*+*Z*), where *X* is the number of reads with genera correctly assigned by a method, and *Z* is the number of reads with an incorrect genus assignment by the method. As with sensitivity, precision was calculated independently for each taxonomic rank.

Kraken2's confidence thresholds were implemented using the --confidence parameter. Thresholds of 0.25, 0.50, and 0.75 were each tested. When the confidence threshold option is invoked, Kraken2 classifies a read to the lowest taxonomic rank satisfying that confidence score.

POSMM's scores are based on the predict_proba function of scikit's logistic regression models. The POSMM score for a read to be assigned to a taxon is therefore the probability that the read with the specified score, length, and model order would be assigned to that taxon based on the logistic regression model for that taxonomic rank. Each taxonomic rank and model order have their own regression models, and probabilistic scores are calculated independently for each.

3.4.10 POSMM Software Release

The underlying algorithm for POSMM is written in C++, with all user-interface and downstream processing written in Python. Source code for generating probabilistic scores using logistic regression models, written in JSON, as well as all other source codes, are available at https://www.github.com/djburks/POSMM. Simulated metagenomes are available at the Kraken2 website https://ccb.jhu.edu/software/kraken/dl/accuracy.tgz, while the mock and real metagenomes are available at the NCBI SRA (Leinonen et al., 2011).

A Python source distribution is also available at https://github.com/djburks/POSMM/blob/main/dist/POSMM-1.0.tar.gz, which handles all necessary dependencies for the end user when installed with pip. Genomes for modelling can be provided by the user with a custom lineage map, or downloaded using POSMM's internal RefSeq query system by using the --taxlist parameter and a list of GCF numbers.

CHAPTER 4

ASSESSMENT OF CLUSTERING APPROACHES FOR DECIPHERING BACTERIAL CHIMERISM*

4.1 Introduction

An Athenaeum of modern biology exists online, and amongst its digital corridors are wings devoted entirely to the blueprints of life (Leinonen et al., 2011; Pagani et al., 2012; Stoesser et al., 2002). Once hosting only sequences of DNA fragments from different life forms, publicly accessible nucleotide databases now boast thousands of fully sequenced genomes that can be electronically transmitted across the world within seconds, and the list of genome entries keeps growing daily (Land et al., 2015; Straiton, Free, Sawyer, & Martin, 2019). This scale of completely sequenced genomes has enabled large-scale genome-wide studies to understand relationships among organisms based on entire genetic content, reconstruction of ancestral genomes thus shining new light on evolution, and inference of genetic elements that interact to confer different phenotypes, to cite a few. In the context of prokaryotic evolution, this has provided novel insights into plasticity of prokaryotic genomes driven by the propensity to acquire and assimilate foreign genomic elements from different lineages and thus gain new traits to adapt to changes in the environment. Horizontal gene transfer thus renders mosaicism in genomes, with each such genome a collection of vertically transmitted and horizontally acquired genes. A number of mechanisms including conjugation, transformation, and transduction have been attributed to the emergence and evolution of chimeric genomes (Dagan, Artzy-Randrup, & Martin, 2008; Ochman et al., 2000). Deconstructing chimeric

*This chapter is reproduced from Burks, D. and Azad, R. K. (2020). Assessment of clustering approaches for deciphering bacterial chimerism. Submitted to PLOS Computational Biology. Authors retain copyright.

genomes and tracing the origins of their disparate segments will augment our understanding of prokaryotic evolution (Abby, Tannier, Gouy, & Daubin, 2012; Jani et al., 2016; Popa, Hazkani-Covo, Landan, Martin, & Dagan, 2011).

Deconstructing chimeric genomes and inferring the evolutionary histories of their distinct segments are central to understanding organismal evolution and relationships among organisms. One of the first steps in this direction is to identify evolutionarily distinct segments in genomes. Because these disparate segments represent different genomic contexts, i.e. the contexts of their source genomes, methods that invoke compositional disparity, e.g. biases in (oligo)nucleotide composition or codon usage, have been developed to delineate compositionally distinct segments in genomes (Rajeev K Azad & Lawrence, 2012; Karlin, 1998; Karlin, Mrázek, & Campbell, 1998; Waack et al., 2006; Zhang & Zhang, 2004) . Earlier attempts focused on finding “change points” in DNA sequences, i.e. the positions where there are transitions in certain properties, e.g. from low GC content to high GC content or vice versa (Braun & Muller, 1998). These inflection points were examined by moving a window over a sequence or performing top-down recursive segmentation of the sequence (Pham, 2007; Thakur, Azad, & Ramaswamy, 2007). This allowed finding segments that are compositionally distinct from the neighboring segments, but did not relate segments originating from the same source. Attempts to decipher “segment types” using hidden Markov model (HMM) in parallel with change point detection seemed promising (Nicolas et al., 2002), however, the requirement of assigning the number of hidden states (segment types) renders this approach not suitable in this context, as the number of segment types is not known *a priori*. A Bayesian approach was adapted to infer the number of segment types to be inputted into an HMM, however, this integrative approach was computationally prohibitive for long sequences (>50 Kbp) and therefore couldn't be applied to prokaryotic genomes that are typically comprised of millions of nucleotides (Boys & Henderson, 2004). A fully Bayesian model was also deployed (Keith, 2006) and additionally, other optimization methods were tested (Gionis & Mannila, 2003), but were not found efficient in deconstructing chimeric genomes in subsequent studies (Rajeev K. Azad & Li, 2013). A combination of recursive segmentation and agglomerative clustering was demonstrated to be most efficient among all assessed methods (Rajeev K. Azad & Li, 2013). A generalized information-entropy based measure, namely, Markovian Jensen-Shannon Divergence (MJSD), was used for assessing the compositional difference between DNA segments or clusters of DNA segments (Arvey, Azad, Raval, & Lawrence, 2009; Rajeev K. Azad & Li, 2013; Thakur et al., 2007). The MJSD based method combined a top-down approach (segmentation) with a bottom-up approach (clustering) to identify distinct segments and segment types in a given genome (Rajeev K. Azad & Li, 2013). First, a genome is recursively segmented into compositionally distinct, internally homogenous nucleotide fractions. Hyper-segmentation is allowed to identify boundaries between compositionally distinct segments with precision, however, this may generate splits in some otherwise homogeneous segments, which is remedied by a subsequent agglomerative clustering procedure to merge contiguous similar segments. Distinct segment types, representing potentially different sources, are identified via a recursive clustering procedure to group compositionally similar segments. This combination of segmentation and clustering was shown to be effective in addressing a host of interesting problems in biology, such as, identification of alien segments in bacterial genomes, detection of copy number variations, and alignment-free genome comparison (Rajeev K. Azad & Li, 2013).

Further advances focused on improvisations to identify horizontally acquired large structures, namely, the genomic islands (GIs), in prokaryotic genomes. This led to the conceptualization of tools such as GEMINI (Jani, Mathee and Azad 2016) and IslandCafe (Jani and Azad 2019). The method has also proven useful in identifying evolutionary strata on sex chromosomes, including those on the human and plant X chromosomes, where this approach was shown to decipher strata independent of X-Y chromosome comparison to infer serial recombination suppression events along the chromosomes (Pandey, Wilson Sayres, & Azad, 2013).

Once the segments are delineated via recursive segmentation and potential over-segmentation corrected via purge of boundaries between apparently similar segments, different segment types are recovered via an agglomerative clustering procedure. This step is critical for partitioning the genome into core (native, vertically inherited) and accessory (horizontally acquired) components, with accessory further partitioned into groups each representing a distinct donor source (Rajeev K. Azad & Li, 2013; Jani & Azad, 2019; Jani et al., 2016; Pandey et al., 2013; Thakur et al., 2007). The current approach prioritizes grouping of proximal segments or clusters of segments. This is based on the premise that the DNA segments from a donor source are likely to be localized to a region and therefore could be more robustly grouped via "proximal" clustering. Although this assumption may not always hold, this approach has been implemented with promising results (Rajeev K. Azad & Li, 2013; Jani & Azad, 2019). In the current implementation of proximal clustering, segments are assessed from 5'-end to 3'-end of a given genome sequence and compositionally similar segments are grouped recursively following this sequential order of the segments along the genome. One of the limitations of this approach is that once segments or clusters are merged, they are "frozen" i.e.

never changed even after the retrieval of more information to refine clustering in the latter rounds of clustering. Any erroneous mergers could have cascading effects of more erroneous groupings of segments or clusters as the clustering progresses. This could lead to sub-optimal clusters or a clustering configuration that does not adequately represent the inherent compositional structure of the genome. In an extensive benchmarking using a variety of test datasets, this approach still yielded better results than other methods including those based on optimization techniques (Rajeev K. Azad & Li, 2013; Jani & Azad, 2019). Although it compared well with other methods designed to perform similar tasks, comparisons were not performed with other clustering approaches and therefore whether the proximal clustering is the best approach in practice – so far as deconstructing chimeric genomes is concerned – is yet to be established.

In this study, we have attempted to bridge this knowledge gap in the usage of clustering approaches in reconstructing evolutionarily disparate components in chimeric genomes. We compared the performance of proximal clustering approach with hierarchical clustering, where grouping of two most similar segments or clusters takes precedence in a recursive process, regardless of segment locations within a genome, and with network clustering, where a network of segments (nodes representing the segments and edges signifying compositional disparity between the segments) is partitioned into clusters or modules. Affinity propagation, a popular clustering method often compared to network clustering (Frey & Dueck, 2007; Moschopoulos et al., 2011; Vlasblom & Wodak, 2009), was also tested. Assessment was performed using artificial chimeric genomes constructed within a generalized hidden Markov model framework as described in (Azad & Lawrence, 2005). In the absence of real genomes with evolutionary events known with certainty, artificial genomes serve as valid test platform for benchmarking different clustering approaches. Artificial chimeric genomes are comprised of segments with known evolutionary history and therefore have been used in the past to properly assess the performance of competing algorithms (Azad & Lawrence, 2005). As real genomes are often complex with varying levels of heterogeneity, we generated artificial chimeric genomes to mimic prokaryotic genomes with varying degrees of presumed horizontal transfer events. The comprehensive set of simulated genomes thus provided an ideal testing platform for assessing a gamut of clustering methods at different parameter settings.

In what follows, we describe clustering methods evaluated in this study and their performance on a set of artificially chimeric genomes of varying complexity. Each clustering method was tested for its ability to reconstruct segments originating from different sources in artificial chimeric genomes. The effects of parameterization were also investigated, as it is important to understand the sensitivity of a method at different parametric settings to genome heterogeneity given that the vertically inherited and horizontally acquired fractions may vary considerably across chimeric genomes.

4.2 Materials and Methods

4.2.1 Genome Segmentation and First Pass Clustering

Segmentation of genomes is based on the usage of a generalized information-entropic measure, Markovian Jensen-Shannon Divergence (MJSD), to recursively partition a sequence into compositional homogeneous segments as described in (Arvey et al., 2009; Thakur et al., 2007). MJSD of order *m*, $D^m(p_1,p_2)$, quantifies the difference between probability distributions p_1 and p_2 estimated from respective DNA sequences and is defined as:

$$D^m(p_1,p_2) = H^m(\pi_1 p_1 + \pi_2 p_2) - \pi_1 H^m(p_1) - \pi_2 H^m(p_2) \qquad \textbf{(Eq. 4.1)}$$

where $H^m(p_i) = -\sum_w P(w) \sum_{x \in \mathcal{A}} P(x|w) \log_2 P(x|w)$ is the Shannon entropy function for Markov model of order *m*, *x* denotes the nucleotide succeeding oligonucleotide *w* of length *m*, *P*(*x*/*w*) is the probability of *x* given the preceding oligonucleotide *w*, and *P*(*w*) is the probability of oligonucleotide w. $\mathcal{A} \equiv$ {A,T,C,G}. π_i is weight associated with p_i, $\pi_1 + \pi_2 = 1$.

Given a genomic sequence *S* of length *L* to be segmented into subsequences S_1 and S_2 of lengths l_1 and l_2 respectively, p_i in $D^m(p_1,p_2)$ represents the distributions *P*(*w*) and *P*(*x*/*w*) in S_i, which are estimated using the frequencies of oligonucleotides *w* and *wx* in S_i. When the weight π_i is equal to l_i/L, MJSD between S_1 and S_2 can be written as:

$$D^m(S_1 S_2) = H^m(S) - \left(\frac{l_1}{L} H^m(S_1) + \frac{l_2}{L} H^m(S_2)\right) \qquad \textbf{(Eq. 4.2)}$$

where $S = S_1 \oplus S_2$, $\oplus$ denotes concatenation. $H^m(S)$ or $H^m(S_i)$ is computed using probability distributions *P*(*w*) and *P*(*x*/*w*) in *S* or S_i as described above. The statistical significance of a value of this measure is assessed based on the probability distribution of this measure that was shown to approximate chi-square distribution $P(D^m \leq X) \approx \chi^2_{\upsilon}(2L(ln2)X)$ with υ degrees of freedom (Arvey et al., 2009). The P-value is thus $1 - P(D^m \leq X)$.

Given a sequence *S* of length *L*, the split point in *S* that results in maximum value of $D^m(S_1,S_2)$ between resulting subsequences S_1 and S_2 is identified. The position with maximum D^m value, denoted D^m_{max}, is determined by computing D^m for each possible split beginning with nucleotide position10 and through L-10. The statistical significance of D^m_{max} is inferred based on an analytic-numerical approximation of the probability distribution of D^m_{max} that was shown to follow chi-square distribution function with fitting parameters (Arvey et al., 2009; Thakur et al., 2007):

$$P(D_{max}^{m} \leq X) \approx \{\chi_{\upsilon}^{2}[2L(ln\,2)X\beta]\}^{N_{eff}} \qquad \textbf{(Eq. 4.3)}$$

where χ_{υ}^{2} is the chi-square distribution function with υ degrees of freedom, and β and N_{eff} are fitting parameters previously estimated using Monte Carlo simulations for model order up to 2 (Arvey et al., 2009; Thakur et al., 2007). The P-value for D_{max}^{m} is thus estimated as $1 - P(D_{max}^{m} \leq X)$.

If the P-value for D_{max}^{m} is below a pre-determined significance threshold, sequence *S* is segmented at the position of maximum divergence between S_1 and S_2. This process is recursively repeated for each resulting segment, until none of the segments can be segmented further, i.e. when P-value for D_{max}^{m} is above the significance threshold for each segment, or the segment is too short to be reliably fragmented (< 16 nucleotides).

Hyper-segmentation is allowed at a relaxed stringency in order to identify segment boundaries with precision. This generates additional splits of otherwise homogeneous segments, and therefore to remedy this, grouping of similar adjacent segments is performed within the same statistical hypothesis testing framework (Rajeev K. Azad & Li, 2013). If the P-value for MJSD between two adjacent segments is lower than an established significance threshold for contiguous segment merger, the segments are deemed significantly different, otherwise they are merged, i.e. the boundary between them is purged. This procedure is performed recursively until no two segments sharing a boundary are deemed similar (Rajeev K. Azad & Li, 2013). Note that oligonucleotide frequencies for a cluster with multiple segments were obtained by averaging the frequencies in these segments for each oligonucleotide, and the size was obtained as the mean of segment lengths. Both recursive segmentation and first-pass clustering were performed at the significance threshold of 0.05. This threshold was established based on our assessment on artificial chimeric genomes as described below.

4.2.2 Segment Order Based Clustering

Location-dependent clustering has been previously implemented in several MJSD-based genome segmentation-clustering programs (Rajeev K. Azad & Li, 2013; Jani & Azad, 2019; Jani et al., 2016). This approach is based on the premise that the horizontally acquired sequences from a source are often localized in a chimeric genome and therefore sequential clustering, based on the order of segments, could reconstruct such structure as this prioritizes grouping of proximal similar segments thus minimizing the effects of potential cross-clustering of apparently similar segments from different sources. Given *N* segments, this algorithm begins with *N* clusters containing 1 segment each. Beginning with the first cluster (at the 5'-end of the genome sequence), it is compared with the next cluster in order (5'-end to 3'-end) along the sequence and their compositional difference is assessed using MJSD within the same statistical framework as used for the segmentation and first-pass clustering. If the P-value for this difference is less than a preset significance threshold, the clusters are deemed compositionally distinct otherwise they are merged. This is performed recursively until no two clusters can be merged further, i.e. the P-value for MJSD between any two clusters is less than the significance threshold.

4.2.3 Hierarchical Clustering

Hierarchical clustering was performed to group *N* clusters containing a single segment each. In a pairwise manner, the compositional difference between clusters in each possible pair was assessed using MJSD. The pair of clusters with minimum MJSD was identified; if the P-value for MJSD was less than a preset significance threshold, then the clusters were deemed compositionally distinct, otherwise these clusters were merged into a single cluster resulting in total *N*-1 clusters. This procedure was followed recursively until no two clusters can be merged further. Hierarchical clustering had previously been implemented to perform clustering of genes based on codon usage patterns to localize putative alien genes in prokaryotic genomes (Azad & Lawrence, 2012).

4.2.4 MCL-Based Network Clustering

A network of segments was constructed, with nodes representing segments and edges signifying compositional disparity between segments. This followed segmentation of a genome and then first pass clustering at a significance level of 0.05 as described above. An edge connecting nodes was established for each node pair. To each edge was associated a weight based on P-value for MJSD between nodes (segments) connecting the edge. MJSD P-values for all node (segment) pairs were first computed and were then renormalized to represent weights on edges connecting the nodes. The weight π_{ij} for the edge connecting nodes *i* and *j* was obtained as:

$$\pi_{ij} = 1 - \left(\frac{P\left(D_{ij}^{m}\right) - P_{min}(D^{m})}{P_{max}(D^{m}) - P_{min}(D^{m})} \right) \qquad \textbf{(Eq. 4.4)}$$

where $P\left(D_{ij}^{m}\right)$ is the P-value for the order *m* MJSD between segments *i* and *j*, and $P_{max}(D^{m})$ and $P_{min}(D^{m})$ are the maximum and minimum P-values, respectively, among all segment pair MJSD P-values in the network. A graph clustering algorithm, namely, Markov clustering algorithm (MCL), was applied to partition this network into modules or clusters of segments (Dongen, 2000; van Dongen & Abreu-Goodger, 2012). MCL clustering simulates random walks within graph structures under the influence of repeated expansion and inflation. Expansion corresponds to the random walking within the graph, establishing transition probabilities between all nodes. Inflation exaggerates these probabilities by raising all probabilities to a specified value (inflation), followed by a scaling step to ensure that the resulting matrix remains stochastic, with the eventual goal of highlighting clusters (Dongen, 2000). The mcl version 14-137 suite of software was employed for this task (https://www.micans.org/mcl/index.html?sec_software). Tab-delimited (.abc) file containing each node pair and associated edge weight was inputted to the program that performed clustering with the ‘mcl’ command. Clustering granularity was controlled via the inflation (-I) parameter, with tested values ranging from 1.1 to 30 in 0.1 increments. As this value is increased, the number and the specificity of clusters also increase. The output was converted to the recommended native network format of mcl using the ‘mcxload’ and ‘mcxdump’ commands.

4.2.5 Affinity-Propagation Clustering

A matrix of similarity values based on the P-values for MJSD between segments was constructed for use with the affinity propagation clustering method (Bodenhofer, Kothmeier, & Hochreiter, 2011; Frey & Dueck, 2007). The similarity matrix contained the complement of P-values for MJSD between segments for all segment pairs, resulting from segmentation and first pass clustering at the significance level of 0.05. The similarity matrix was inputted to affinity propagation clustering program (‘apcluster’ package v1.4.8 through R v4.0.0) that generated clusters of similar segments (Bodenhofer et al., 2011). The performance of the apcluster function is controlled by two primary parameters, “p” and “q”, which control the likelihood of each data sample to become a cluster representative. By default, apcluster initializes exemplar preferences, determined by the diagonal of the matrix, for all data points. When the p parameter is unspecified, exemplar preferences are set to the quantile assigned by q, with q = 0.5 representing the median and therefore leading to default behavior (Bodenhofer et al., 2011). While unique preferences can be set for each data point by providing a vector to the p parameter, each segment was considered equally important and the p parameter thus remained unassigned for this benchmark. In all cases, the damping factor was set to 0.90, with the ‘maxits’ and ‘convits’ parameters set to 10,000 and 1,000, respectively.

4.2.6 Construction of Artificial Chimeric Genomes

Artificial genomes for assessing the performance of clustering methods were constructed using a generalized hidden Markov model framework as previously described (Rajeev K Azad & Lawrence, 2005). An artificial genome was modeled after a core genome comprised of a set of core (native) genes in the genome, which were extracted based on a model selection criterion algorithm at a conservative setting. To model genic variation within a core genome, the core gene set was partitioned into gene classes with distinct mutational biases using a *k*-means clustering algorithm with relative entropy as the distance metric. Gene models trained on distinct gene classes were incorporated within the framework of generalized hidden Markov model to generate an artificial genome modeled after the real core genome. Artificial chimeric genomes were constructed by simulating transfer of randomly sampled genes or clusters of genes from a pool of donor artificial genomes into recipient artificial genomes.

Complexity of chimeric genomes resulted from both the number of donors and proportion of horizontally acquired genes.

For this study, 35 artificial genomes modeled after the respective prokaryotic genomes were used (Supplementary Table 4.1, see tab “Genome Key”). The compositions of 11 artificial chimeric genomes constructed for assessment of clustering methods are also documented in Supplementary Table 4.1. The artificial chimeric genomes used in this study have been made available at https://github.com/djburks/SGM-Clustering-Programs.

4.2.7 Benchmarking of Clustering Methods

For each clustering method, the entire parameter space was explored to find the optimal parametric setting that yields the best performance of the method. For segment order based and hierarchical clustering, the significance threshold was varied from 0.999 to 10^{-15}. For MCL clustering, inflation parameter threshold was varied from 1.1 to 30 in 0.1 increment. For affinity propagation clustering, the q-parameter threshold was varied from 0.001 to 0.999 in 0.001 increment.

The accuracy metrics, namely, sensitivity (SN), precision (PR), and F_1-score (F_1) were computed at each threshold setting for each clustering method. Since the vertically inherited genes are most numerous in a prokaryotic genome, the native genome component (backbone) is identified by the largest cluster, whereas the alien component is identified by segments in the remaining smaller clusters. We established the optimal threshold settings for both, identifying native component and identifying alien component, based on F_1-score, which is a harmonic mean of SN and PR, computed as follows.

$$SN = \frac{TP}{TP+FN} \tag{Eq. 4.5}$$

$$PR = \frac{TP}{TP+FP} \qquad \textbf{(Eq. 4.6)}$$

$$F_1 = 2 \cdot \frac{SN \cdot PR}{SN+PR} \qquad \textbf{(Eq. 4.7)}$$

In terms of identifying the native component, TP, FN, and FP refer to number of native nucleotides correctly identified as native (true positives), number of native nucleotides incorrectly identified as alien (false negatives), and number of alien nucleotides incorrectly identified as native (false positives), respectively, by a method. In terms of identifying the alien component, TP, FN, and FP refer to number of alien nucleotides correctly identified as alien (true positives), number of alien nucleotides incorrectly identified as native (false negatives), and number of native nucleotides incorrectly identified as alien (false positives) respectively.

One aspect in deconstructing a chimeric genome is identification of native and alien components of the genome. Without multiple genome comparison, the segmentation coupled with clustering infer these components in a single genome. The success hinges on reconstructing the backbone, which has to be recovered in a single cluster by the method, and thus the native and alien components are inferred based on the cluster size– the largest being native and the rest being alien. Note that native segments often group into two or more distinct clusters and attempts to merge them by relaxing the stringency may result in undesirable mergers. We therefore assessed different clustering methods on their ability to merge native clusters and thus identify the native component robustly. At the algorithmic parameter setting where F_1 of a clustering method in identifying the native component was maximized, we further assessed the ability of the method in segregating segments from different sources (native and different donors). For each source, the clusters that contain the majority of their nucleotides from the source were identified; the nucleotides within these clusters were labeled positives, and rest all nucleotides negatives and TP, FP, and FN (with respect to the source) were computed to assess sensitivity, precision and overall accuracy (F_1) of a method in identifying the source. We performed source level assessment also at the algorithmic parameter setting where F_1 of a clustering method in identifying the alien component was maximized.

4.3 Results

4.3.1 Comparative Assessment of Clustering Methods in Native (Core) Genome Isolation

Identifying the native or core component of a genome is an important goal in evolutionary genomics as it enables reconstructing the tree of life based on vertically inherited genetic information, specifically the microbial clades where such signals may be obfuscated by frequent horizontal gene exchange (Chung, Munro, Tettelin, & Dunning Hotopp, n.d.; Daubin, 2002; Na et al., 2018; Segata & Huttenhower, 2011). Using compositional bias to identify the core genome may circumvent some limitations of comparative genomics approaches which rely on the sequenced genomes of close relatives to infer the core genome. Depending on the genomes and criteria considered, the core genome may vary significantly, and therefore using additional information such as composition in conjunction with genome comparison may help in robust extraction of core genome. For example, a native gene that displays sporadic presence in close relatives due to gene loss may gain a support for inclusion in the native group based on its compositional similarity with other (well-supported) native genes.

Since the evolutionary history of nucleotides is known with certainty in artificial chimeric genomes, we assessed different clustering methods for their ability to reconstruct the core components of artificial chimeric genomes. Markov model of order 2 (*m*=2) was used in all variants of the segmentation-clustering algorithm, as parameters for statistical hypothesis testing have already been determined for 2nd order model and previous studies have shown promising results with 2nd order model algorithms (Arvey et al., 2009; Rajeev K. Azad & Li, 2013; Thakur et al., 2007). The clustering threshold at which the overall accuracy (F_1 score) in identifying the core component of a genome was maximized was determined for each artificial chimeric genome, for each method. The values of SN, PR and F_1 in identifying core genomes at the optimal settings are provided in Table 4.1 and the values of SN, PR and F_1 in identifying each source (recipient and different donors) at these settings are provided in SUPPLEMENTARY TABLE 4.2. Note that the core genome was assembled from the segments resident in the largest cluster following segmentation and clustering (see Methods). The recursive segmentation and the first-pass clustering to group similar adjacent segments outputted by segmentation were both implemented at the significance level of 0.05. This threshold was established based on extensive experiments on artificial chimeric genomes. Significance threshold of 0.05 was observed to yield fragments that consistently match well with distinct segments in artificial chimeric genomes.

All clustering methods except affinity propagation clustering attained F_1 over 90% across all test genomes (Table 4.1). Segment order-based clustering and network clustering produced highest overall accuracy (F_1 averaged over 11 test genomes: 0.972); average F_1 values for hierarchical clustering and affinity propagation clustering were 0.969 and 0.609 respectively. In genome-wise assessment, segment order-based clustering was found to have maximum F_1 for 6 of 11 genomes, followed by networking clustering (3 genomes) and hierarchical clustering (2 genomes). As expected, we found the optimal performance varies with genome composition.

Table 4.1: Sensitivity (SN), Precision (PR), and F_1 score values in identifying native nucleotides in test genomes by different clustering methods at the genome-specific optimal parameter settings where F_1 scores are maximal

Test Genome	Segment-Order				Hierarchical				Network (MCL)				Affinity Propagation			
	Parameter (T3)	SN	PR	F1	Parameter (T3)	SN	PR	F1	Parameter (I)	SN	PR	F1	Parameter (q)	SN	PR	F1
Artificial Chimeric Genome 1	1.00E-08	0.999	0.996	0.997	1.00E-04	0.999	0.990	0.995	29.7	0.987	0.998	0.992	0.863	0.460	0.997	0.629
Artificial Chimeric Genome 2	0.019	0.997	0.962	0.980	0.034	0.996	0.966	0.981	27.4	0.987	0.971	0.979	0.712	0.507	0.973	0.667
Artificial Chimeric Genome 3	0.367	0.994	0.966	0.980	0.095	0.995	0.964	0.979	25.9	0.984	0.974	0.979	0.797	0.538	0.878	0.667
Artificial Chimeric Genome 4	0.449	0.974	0.959	0.966	0.012	0.994	0.951	0.972	27.2	0.994	0.951	0.972	0.887	0.406	0.962	0.571
Artificial Chimeric Genome 5	0.558	0.993	0.920	0.955	0.42	0.994	0.882	0.935	21.9	0.977	0.931	0.953	0.378	0.516	0.934	0.665
Artificial Chimeric Genome 6	0.62	0.987	0.922	0.953	0.863	0.987	0.916	0.950	21.2	0.970	0.934	0.952	0.769	0.420	0.916	0.576
Artificial Chimeric Genome 7	0.7	0.971	0.867	0.916	0.55	0.974	0.847	0.906	19.3	0.957	0.867	0.910	0.845	0.398	0.726	0.514
Artificial Chimeric Genome 8	0.313	0.988	0.981	0.985	0.018	0.995	0.971	0.983	23.6	0.987	0.982	0.984	0.83	0.458	0.980	0.624
Artificial Chimeric Genome 9	0.005	0.985	0.993	0.989	0.001	0.984	0.994	0.989	29	0.995	0.989	0.992	0.3	0.388	0.993	0.558
Artificial Chimeric Genome 10	0.029	0.985	0.997	0.991	1.00E-04	0.983	0.998	0.990	30	0.992	0.997	0.995	0.296	0.421	0.999	0.592
Artificial Chimeric Genome 11	1.00E-07	0.993	0.982	0.987	0.174	0.967	0.991	0.979	27.7	0.995	0.987	0.991	0.881	0.468	0.992	0.636

While all clustering methods, except affinity propagation clustering, did well on genomes with substantial contributions from each of fewer donors (e.g. Artificial Chimeric Genome 1 in SUPPLEMENTARY TABLE 4.1), the performance declined with increasing complexity, particularly with increasing number of donors with fewer gene contribution by each (e.g. Artificial Chimeric Genome 7 in SUPPLEMENTARY TABLE 4.1).

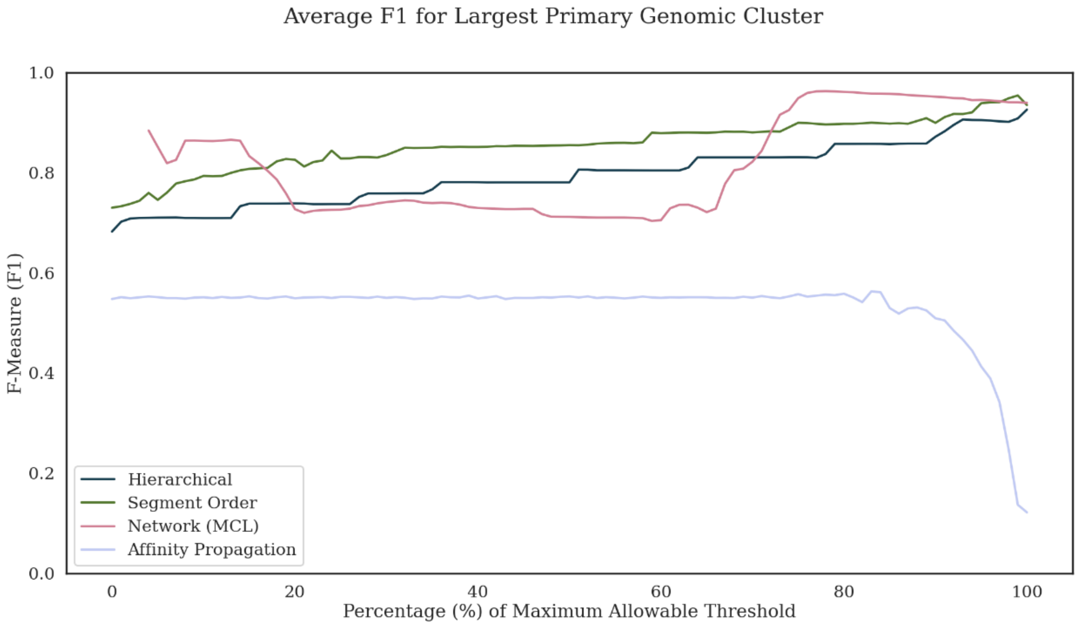

Figure 4.1: Overall accuracy (F1 score averaged over all test genomes) in identifying native (core) nucleotides for each clustering method at increasing thresholds. The network clustering threshold (MCL inflation parameter) is represented as the percentage of the maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented here as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level).

In our aforementioned assessment, we observed that performance was optimized at different thresholds for different genomes by each method. Thus, a default threshold that can work well across genomes of different compositions cannot be established. However, it could still be possible to determine a threshold, or a threshold range, where the overall performance is acceptable across genomes of different compositions. We therefore computed the performance metrics averaged over all test genomes at each threshold (varied from 10^{-15} to 0.999 for segment order based clustering and hierarchical clustering, from 1.1 to 30 (inflation) for network clustering, and 0.001 to 0.999 (q-value) for affinity propagation clustering). The overall accuracy, as assessed by F_1 measure, is shown as a function of threshold for each method (Figure 4.1). As expected, the best performance of both segment order based clustering and hierarchical clustering was attained when the significance threshold was close to 0 (towards the rightmost end of F_1 plots in Figure 4.1: 0.019 for the former and 0.002 for the latter; thresholds are represented here as a percentage of the maximum allowable threshold (complement of significance level for hierarchical and segment order clustering)). These are more relaxed stringencies for clustering, allowing merger of multiple native clusters into a single native cluster. This is apparent in the SN plot (SUPPLEMENTARY FIG. 4.1), which shows that relaxing stringency (lower P-values) continually increases sensitivity as the significance threshold approaches 0 (towards the rightmost end in SUPPLEMENTARY FIG. 4.1), however, the precision declines remarkably as alien clusters begin to coalesce with the native cluster as the stringency is successively relaxed (PR plot, SUPPLEMENTARY FIG. 4.2). The overall performance of hierarchical clustering was on average lower than that of segment order-based clustering across the entire threshold range (Figure 4.1), though this difference becomes much smaller at significance threshold 0.001 or lower (Figure 4.2). Notably, the F_1 varied in a stepwise manner with threshold for both these clustering approaches, more so with hierarchical clustering (Figure 4.1). F_1 varied in steps because of the fact that a significant increase in recall happened only when a threshold value was reached where small native cluster(s) merged with the largest cluster. The stepwise pattern is more pronounced with hierarchical clustering, where the largest (native) cluster remains invariant over longer threshold intervals until a smaller native cluster merges as the stringency is successively relaxed. The “punctuated” native cluster merger resulting in a ladder step pattern is less apparent with segment-order based clustering where proximal native segments or clusters are continually merged as encountered from 5’ to 3’ end, recursively (SUPPLEMENTARY FIG. 4.1).

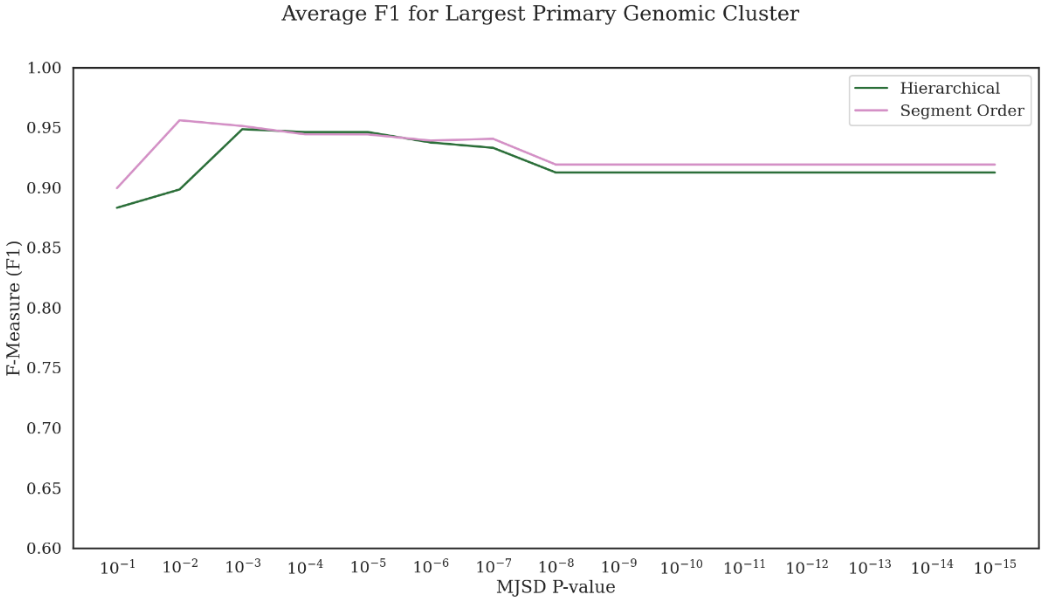

Figure 4.2: Overall accuracy (F1 score averaged over all test genomes) in identifying native (core) nucleotides for hierarchical and segment-order based clustering methods at MJSD significance level range of 10^{-1} to 10^{-15}.

In contrast, network clustering displayed a different pattern of F_1 variation with threshold. Unlike segment order-based or hierarchical clustering, network clustering employs MCL algorithm that uses an inflation parameter that ranges from 1.1 to 30. More granular clusters are produced as the value of inflation parameter is increased (van Dongen & Abreu-Goodger, 2012). Unlike agglomerative clustering, where clusters are built bottom-up beginning with single segment clusters, network clustering is a top-down approach that partitions the network of segments into modules of similar segments. F_1 varied differently with threshold in network clustering compared to agglomerative clustering (Figure 4.1). In contrast to gradual stepwise variation in F_1 with threshold in agglomerative clustering, the F_1 variation with inflation parameter is shaped like a bucket with rather abrupt transition from high to low or low to high F_1. The optimal performance (maximal F_1 score) was observed at inflation parameter values between 20 and 23 for artificial chimeric genomes. Unlike agglomerative clustering, where typically a single large native cluster is formed following successive rounds of cluster merger, analysis of cluster formation by network clustering revealed concurrent formation of at least two large native clusters which coalesce at the optimal threshold (typically between 20-23) yielding maximal F_1.

All methods, except for affinity propagation, attained high average F_1 (>0.9) in identifying native nucleotides, with network clustering with maximal average F_1 of 0.963 at threshold 23.3 slightly outperforming segment-order based clustering and hierarchical clustering that attained maximal average F_1 of 0.957 and 0.950 at significance thresholds 0.019 and 0.002 respectively (Figure 4.1, Table 4.2). The maximal average F_1 of affinity propagation was achieved with the q parameter set to 0.831, but was much lower than those of the other three methods at only 0.574. For agglomerative and network clustering, high performance in identifying native DNA was attained at threshold ranges that allowed merger of several native clusters into a single native cluster without incurring undesirable mergers (none or very few alien and native cluster mergers); these were inflation parameter range 20-23 for network clustering and significance level range 0.01-0.001 for agglomerative clustering. This was not the case for affinity propagation, where although the precision remained high across all tested thresholds (>0.9) (SUPPLEMENTAL FIGURE 4.2), the sensitivity was consistently low (< 0.54) and declined sharply after reaching a maximal average of 0.538, resulting in a similar trend observed with F_1 (SUPPLEMENTAL FIGURE 4.1). Notably, with network clustering, merger of two large native clusters happened within the inflation parameter range of 16-23 across all artificial chimeric genomes, whereas native cluster mergers that spiked the F_1 score happened across different threshold ranges for different artificial chimeric genomes with agglomerative clustering. Genome-wise performance shows that the overall accuracies of the methods declined (Table 4.2, SUPPLEMENTARY TABLE 4.3) compared to the performance at genomic-specific optimal settings (Table 4.1, SUPPLEMENTARY TABLE 4.2), as expected. Network clustering produced maximum F_1 for 8 out of 11 genomes, outperforming other methods (Table 4.2). Segment order-based clustering, hierarchical clustering, and affinity propagation clustering produced maximum F_1 for 2, 1, and 0 genome(s), respectively (Table 4.2). In SUPPLEMENTARY TABLE 4.3, we provide average SN, PR, and F_1 score in identifying sources (recipient and different donors) at the same threshold setting that yielded maximal average F_1 score in identifying native DNA for each method (Figure 4.1). Network clustering whose maximal average F_1 score in identifying native DNA was highest among all four methods identified sources more robustly than other methods (identified 71 sources with F_1 >0.7, whereas hierarchical, segment-order, and affinity propagation clustering methods identified 69, 68, and 59 sources respectively with F_1 >0.7). Notably, F_1 for native source was greater for the former compared to the latter three clustering approaches. This demonstrates that a better discrimination of native from alien is attained via robust grouping of segments from different sources, particularly by grouping robustly the native segments into a single large native cluster in this particular instance.

Table 4.2: Sensitivity (SN), Precision (PR), and F_1 score values in identifying native nucleotides in test genomes by different clustering methods, generated at the optimal parametric setting of each method where the F_1 score averaged over all test genomes was maximal.

Test Genome	Segment-Order				Hierarchical				Network (MCL)				Affinity Propagation			
	Parameter (T3)	SN	PR	F1	Parameter (T3)	SN	PR	F1	Parameter (I)	SN	PR	F1	Parameter (q)	SN	PR	F1
Artificial Chimeric Genome 1	0.019	0.987	0.996	0.992	0.002	0.999	0.990	0.995	23.3	0.975	0.998	0.986	0.831	0.450	0.997	0.620
Artificial Chimeric Genome 2	0.019	0.997	0.962	0.980	0.002	0.997	0.958	0.978	23.3	0.983	0.975	0.979	0.831	0.488	0.972	0.650
Artificial Chimeric Genome 3	0.019	0.998	0.946	0.971	0.002	0.997	0.948	0.972	23.3	0.979	0.975	0.977	0.831	0.534	0.877	0.664
Artificial Chimeric Genome 4	0.019	0.983	0.923	0.952	0.002	0.995	0.927	0.960	23.3	0.980	0.960	0.970	0.831	0.360	0.841	0.504
Artificial Chimeric Genome 5	0.019	0.984	0.885	0.932	0.002	0.996	0.875	0.931	23.3	0.983	0.906	0.943	0.831	0.504	0.933	0.655
Artificial Chimeric Genome 6	0.019	0.994	0.852	0.918	0.002	0.995	0.852	0.918	23.3	0.982	0.911	0.945	0.831	0.317	0.962	0.477
Artificial Chimeric Genome 7	0.019	0.992	0.753	0.856	0.002	0.995	0.655	0.790	23.3	0.994	0.734	0.844	0.831	0.376	0.719	0.494
Artificial Chimeric Genome 8	0.019	0.995	0.945	0.970	0.002	0.996	0.931	0.962	23.3	0.987	0.982	0.984	0.831	0.434	0.981	0.602
Artificial Chimeric Genome 9	0.019	0.984	0.991	0.988	0.002	0.984	0.994	0.989	23.3	0.984	0.994	0.989	0.831	0.320	0.994	0.484
Artificial Chimeric Genome 10	0.019	0.985	0.995	0.990	0.002	0.983	0.998	0.990	23.3	0.984	0.998	0.991	0.831	0.408	0.999	0.579
Artificial Chimeric Genome 11	0.019	0.980	0.989	0.984	0.002	0.996	0.943	0.968	23.3	0.979	0.990	0.985	0.831	0.412	0.992	0.582

4.3.2 Comparative Assessment for Alien Genome Identification

Identifying native components of genomes is an important goal and the alternative approach suggested here will complement the frequently used comparative genomics approach in establishing backbone genomes in different taxa. Our results showed that the overall accuracy (F_1 score) in identifying native DNA in an ensemble of artificial chimeric genomes could reach up to 0.972 with network clustering as the best overall performer. Genome-wise optimal performers for native DNA identification varied among agglomerative and network clustering methods. However, this does not necessarily imply that the approach and parameter setting that yielded maximal overall accuracy in native DNA identification represent the optimal framework for alien DNA identification as well. Quantifying alien components of prokaryotic genomes is central to understanding how prokaryotes evolve through horizontal gene transfer, and therefore a number of methods have been developed for alien DNA identification (Azad & Lawrence, 2012; Ravenhall, Škunca, Lassalle, & Dessimoz, 2015). Similar to native DNA identification, we compared the four clustering methods to determine the approach and algorithm parameter setting that yield the maximal overall accuracy in alien DNA identification. We approached this problem in two ways - first, at each threshold setting, the largest cluster was identified as native and the remaining clusters as alien (the native nucleotides being most numerous coalesce into a large cluster, while alien nucleotides coalesce into several small clusters each representing a distinct donor source, consistent with the composition of prokaryotic genomes reported previously). SN, PR, and F_1 for alien nucleotide detection were computed and the values for each artificial chimeric genome at the optimal setting were tabulated (Table 4.3).

Table 4.3: Sensitivity (SN), Precision (PR), and F_1 score values in identifying alien nucleotides in test genomes by different clustering methods at the genome-specific optimal parameter settings where F_1 scores are maximal.

Test Genome	Segment-Order				Hierarchical				Network (MCL)				Affinity Propagation			
	Parameter (T3)	SN	PR	F1	Parameter (T3)	SN	PR	F1	Parameter (I)	SN	PR	F1	Parameter (q)	SN	PR	F1
Artificial Chimeric Genome 1	1.00E-08	0.988	0.916	0.951	0.0001	0.984	0.821	0.895	29.7	0.799	0.959	0.872	0.863	0.088	0.975	0.161
Artificial Chimeric Genome 2	0.019	0.983	0.788	0.875	0.025	0.976	0.810	0.885	23.3	0.904	0.864	0.884	0.712	0.257	0.923	0.402
Artificial Chimeric Genome 3	0.367	0.973	0.864	0.915	0.095	0.979	0.856	0.913	25.9	0.934	0.897	0.915	0.551	0.328	0.954	0.489
Artificial Chimeric Genome 4	0.452	0.924	0.886	0.904	0.012	0.982	0.858	0.916	25	0.953	0.881	0.915	0.876	0.368	0.960	0.532
Artificial Chimeric Genome 5	0.722	0.979	0.809	0.886	0.275	0.983	0.706	0.821	21.4	0.939	0.842	0.888	0.378	0.461	0.919	0.614
Artificial Chimeric Genome 6	0.962	0.973	0.853	0.909	0.738	0.973	0.842	0.903	21.2	0.943	0.880	0.910	0.769	0.477	0.932	0.631
Artificial Chimeric Genome 7	0.997	0.970	0.863	0.914	0.937	0.973	0.838	0.900	19.3	0.957	0.865	0.908	0.968	0.562	0.980	0.714
Artificial Chimeric Genome 8	0.313	0.964	0.943	0.953	0.018	0.983	0.911	0.946	23.6	0.960	0.944	0.952	0.83	0.374	0.972	0.541
Artificial Chimeric Genome 9	0.005	0.958	0.978	0.968	0.001	0.954	0.982	0.968	29	0.986	0.967	0.977	0.003	0.355	0.992	0.523
Artificial Chimeric Genome 10	0.029	0.956	0.990	0.973	0.0001	0.952	0.993	0.972	30	0.978	0.991	0.984	0.296	0.368	0.999	0.538
Artificial Chimeric Genome 11	1.00E-07	0.977	0.946	0.961	0.043	0.906	0.973	0.938	27.7	0.984	0.961	0.973	0.881	0.378	0.988	0.547

Network clustering produced the highest F_1 score in identifying alien nucleotides for 5 of 11 artificial chimeric genomes, displaying a better performance than the other clustering approaches; the average F_1 score of 0.928 by segment-order based clustering was, however, ~0.003 higher than network clustering with highest alien nucleotide F_1 score for 4 of 11 artificial chimeric genomes. This is primarily due to the performance discrepancy between network clustering and segment-order clustering in detecting alien nucleotides in Artificial Chimeric Genome 1, the chimeric genome with the least amount of alien nucleotides (5%). Hierarchical clustering with average F_1 score of 0.914 generated highest alien nucleotide F_1 score for 2 of 11 artificial chimeric genomes; affinity propagation clustering with average F_1 score of 0.517 was outperformed by the other methods on both aggregate and individual genome tests. The values of these metrics in identifying nucleotides from each source (recipient and different donors) within the same optimized settings are provided in SUPPLEMENTARY TABLE 4.4.   The percentages (and relative percentages) of native and donor nucleotides in each cluster at these and other optimal settings are illustrated in SUPPLEMENTARY FIGS. 4.3-4.6.

Similar to the analysis for native nucleotide identification, we also assessed the F_1 score for alien DNA identification averaged over all test genomes at each threshold for each method (Figure 4.3). SN, PR, and F_1 for each genome at the setting where average F_1 for alien DNA identification was maximized is provided in Table 4.4 for each method, and the values of these metrics for each source in test genomes at the same setting are provided in SUPPLEMENTARY TABLE 4.5. At its optimal setting, network clustering generated F_1 (averaged over 11 genomes) of 0.903, outperforming other clustering methods (average F_1 for segment order based, hierarchical, and affinity propagation clustering were 0.881, 0.864, and 0.499, respectively;

Table 4.4). Although these values are less than those from genome-wise optimal setting (Table 4.2), as expected, they are still close and just within 3% for network clustering, highlighting its ability to provide a stable parametric setting for application to genomes of variable composition in identifying the alien nucleotides. Notably, network clustering was found to have maximum F_1 for 9 of 11 genomes.

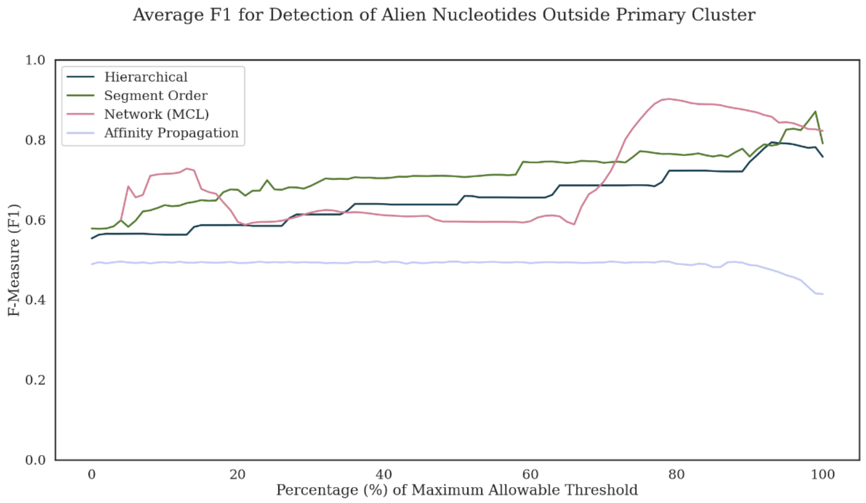

Figure 4.3: Overall accuracy (F1 score averaged over all test genomes) in identifying alien nucleotides for each clustering method at increasing thresholds. The network clustering threshold (MCL inflation parameter) is represented as the percentage of the maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level).

While identifying alien nucleotides could serve well the purposes of some studies, the goal of deciphering clusters arising from different donor sources may not be well served if the focus is on only native and alien discrimination. For example, a method may allow alien clusters from different sources to merge in order to merge native clusters to achieve maximal F_1 in alien nucleotide identification. Therefore, in our second way in assessment, we evaluated the ability of the clustering methods in inferring alien DNA sources by identifying clusters that harbor segments from each donor source.

Table 4.4: Sensitivity (SN), Precision (PR), and F_1 score values in identifying alien nucleotides in test genomes by different clustering methods, generated at the optimal parametric setting of each method where the F_1 score averaged over all test genomes was maximal.

Test Genome	Segment-Order				Hierarchical				Network (MCL)				Affinity Propagation			
	Parameter (T3)	SN	PR	F1	Parameter (T3)	SN	PR	F1	Parameter (I)	SN	PR	F1	Parameter (q)	SN	PR	F1
Artificial Chimeric Genome 1	0.019	0.934	0.793	0.858	0.002	0.821	0.983	0.895	23.7	0.960	0.782	0.862	0.753	0.976	0.078	0.145
Artificial Chimeric Genome 2	0.019	0.788	0.983	0.875	0.002	0.766	0.982	0.861	23.7	0.858	0.903	0.880	0.753	0.926	0.253	0.397
Artificial Chimeric Genome 3	0.019	0.779	0.988	0.871	0.002	0.787	0.986	0.875	23.7	0.902	0.925	0.913	0.753	0.947	0.322	0.480
Artificial Chimeric Genome 4	0.019	0.774	0.944	0.850	0.002	0.782	0.984	0.871	23.7	0.885	0.943	0.913	0.753	0.805	0.312	0.450
Artificial Chimeric Genome 5	0.019	0.716	0.953	0.818	0.002	0.684	0.987	0.808	23.7	0.761	0.958	0.848	0.753	0.916	0.454	0.607
Artificial Chimeric Genome 6	0.019	0.697	0.985	0.816	0.002	0.695	0.989	0.816	23.7	0.829	0.966	0.892	0.753	0.932	0.475	0.629
Artificial Chimeric Genome 7	0.019	0.700	0.990	0.820	0.002	0.516	0.991	0.679	23.7	0.632	0.992	0.772	0.753	0.891	0.551	0.681
Artificial Chimeric Genome 8	0.019	0.974	0.954	0.964	0.002	0.982	0.954	0.968	23.7	0.983	0.956	0.969	0.753	0.992	0.346	0.513
Artificial Chimeric Genome 9	0.019	0.984	0.956	0.970	0.002	0.993	0.952	0.972	23.7	0.993	0.954	0.973	0.753	0.999	0.365	0.535
Artificial Chimeric Genome 10	0.019	0.828	0.982	0.899	0.002	0.778	0.984	0.869	23.7	0.944	0.960	0.952	0.753	0.975	0.363	0.529
Artificial Chimeric Genome 11	0.019	0.967	0.939	0.953	0.002	0.815	0.984	0.892	23.7	0.970	0.948	0.959	0.753	0.989	0.359	0.527

If DNA sequences from a source organism were assigned to multiple clusters, only those clusters with a majority of resident nucleotides belonging to the source were deemed as representing the source organism. As artificial genomes are modeled after core genomes, we expect a method to generate as many clusters as sources contributing to an artificial chimeric genome. Therefore, we selected the largest cluster among clusters representing a source organism, and then computed SN, PR, and F_1 in identifying each donor source for each artificial chimeric genome. The values of these metrics, averaged over the donors, indicate the ability to identify alien DNA in a genome while preserving the identity of the donors in the process. These values at the optimal settings of clustering methods are provided for each genome in Table 4.5 and the respective values for each source in each genome are provided in SUPPLEMENTARY TABLE 4.6. Note that in the event that a donor was not identified as the majority representative of a single cluster, the F_1 score for that particular donor was considered zero. Segment order-based clustering produced highest F_1 scores for a majority of genomes (6 of 11 genomes), whereas hierarchical, network, and affinity clustering methods produced highest F_1 on 4, 1, and 0 genome(s) respectively. The values of F_1 averaged over 11 genomes were similar for segment order-based and hierarchical clustering methods (0.662 and 0.665 respectively), higher than those of network clustering (0.619) and affinity propagation clustering (0.609). Further, we computed SN, PR, and $F_{1,}$ averaged over all genomes, at different threshold settings for each method (Figure 4.4, SUPPLEMENTARY FIGS. 4.7 and 4.8). F_1 of all four methods varied within 5% for over 90% of their threshold range (Figure 4.4). The maximal average F_1 values were 0.623, 0.615, 0.576 and 0.563 for hierarchical clustering, segment-order clustering, network clustering, and affinity propagation clustering, respectively. Thus, overall, agglomerative clustering methods outperformed other clustering approaches in alien DNA identification while preserving donor identity in the genomes. Notably, the thresholds that yielded maximal F_1 in native and alien DNA detection differ noticeably for agglomerative clustering methods; in contrast, there was an overlap of the optimal threshold ranges for native and alien DNA detection by network clustering. Affinity propagation clustering, as with native genome isolation, performed worse than the other methods.

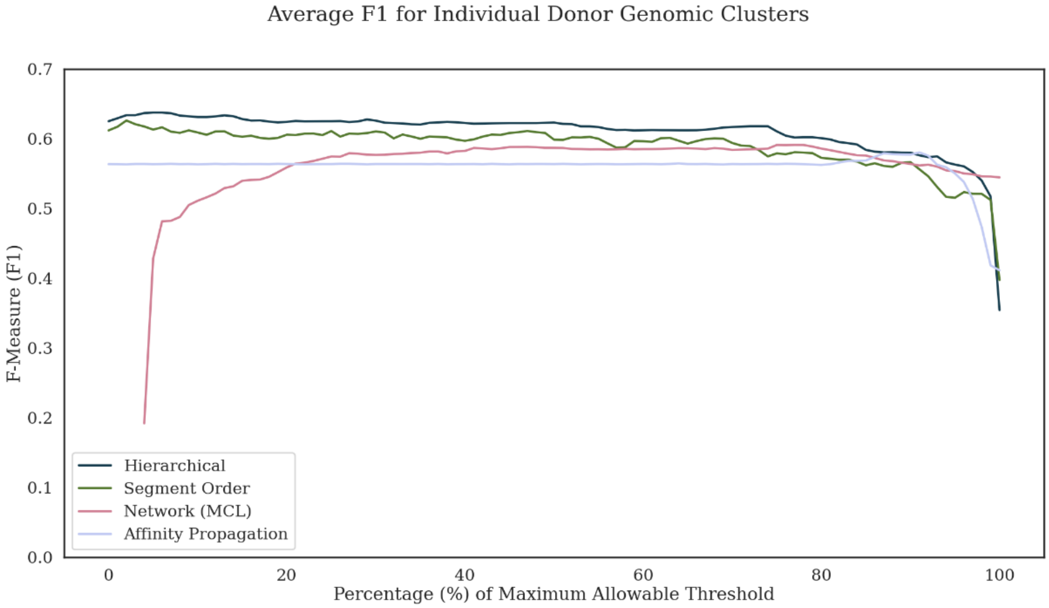

Figure 4.4: Overall accuracy (F1 score averaged over all test genomes) in identifying alien nucleotides while preserving donor identity in the genomes for each clustering method at increasing thresholds. The network clustering threshold (MCL inflation parameter) is represented as the percentage of the maximum allowable threshold of 30. The affinity propagation threshold is represented as a percentage of the maximum allowable threshold of 1. Agglomerative clustering (hierarchical and segment order) thresholds are represented as a percentage of the maximum allowable threshold of 1-10-15 (complement of significance level). The cluster representing each donor is the largest cluster harboring segments primarily of that donor. For donors that did not make up the majority of nucleotides in any cluster, the values of metrics SN, PR, and F1 were deemed zero.

Table 4.5: Sensitivity (SN), Precision (PR), and F_1 score values in identifying alien nucleotides while preserving donor identity in the test genomes by different clustering methods, generated at the optimal parametric setting of each method where the F_1 score averaged over all test genomes was maximal. The cluster representing each donor is the largest cluster harboring segments primarily of that donor.

Test Genome	Segment-Order				Hierarchical				Network (MCL)				Affinity Propagation			
	Parameter (T3)	SN	PR	F1	Parameter (T3)	SN	PR	F1	Parameter (I)	SN	PR	F1	Parameter (q)	SN	PR	F1
Artificial Chimeric Genome 1	0.453	0.587	0.981	0.730	0.332	0.596	0.978	0.738	1.4	0.664	0.974	0.778	0.776	0.590	0.980	0.734
Artificial Chimeric Genome 2	0.948	0.588	0.935	0.686	0.747	0.568	0.959	0.683	22.6	0.540	0.897	0.646	0.874	0.487	0.928	0.612
Artificial Chimeric Genome 3	0.561	0.615	0.850	0.696	0.987	0.609	0.851	0.695	27.8	0.490	0.779	0.571	0.869	0.495	0.798	0.578
Artificial Chimeric Genome 4	0.937	0.526	0.809	0.616	0.914	0.559	0.864	0.646	28.1	0.500	0.806	0.587	0.946	0.482	0.891	0.611
Artificial Chimeric Genome 5	0.999	0.481	0.847	0.572	0.999	0.498	0.912	0.612	22.5	0.479	0.749	0.563	0.914	0.469	0.832	0.572
Artificial Chimeric Genome 6	0.998	0.423	0.828	0.502	0.999	0.458	0.851	0.536	20.7	0.403	0.668	0.480	0.942	0.406	0.786	0.508
Artificial Chimeric Genome 7	0.991	0.418	0.826	0.462	0.998	0.401	0.874	0.455	18.7	0.365	0.582	0.392	0.955	0.353	0.746	0.448
Artificial Chimeric Genome 8	0.313	0.719	0.924	0.794	0.216	0.649	0.937	0.752	1.7	0.607	0.949	0.723	0.851	0.522	0.960	0.673
Artificial Chimeric Genome 9	0.754	0.684	0.925	0.761	0.173	0.692	0.920	0.764	2.2	0.564	0.964	0.700	0.753	0.507	0.982	0.666
Artificial Chimeric Genome 10	0.373	0.711	0.866	0.763	0.902	0.710	0.824	0.746	4.2	0.579	0.940	0.700	0.889	0.519	0.959	0.668
Artificial Chimeric Genome 11	0.796	0.603	0.922	0.703	0.043	0.604	0.905	0.689	1.4	0.605	0.891	0.678	0.873	0.476	0.982	0.640

4.4 Discussion

Our analysis highlights the promises and challenges of different clustering approaches in unraveling bacterial chimerism. While segment order-based clustering was introduced as a part of an integrative approach to segment and cluster bacterial DNA sequences, it is for the first time that hierarchical, network, and affinity propagation clustering techniques were assessed along-side segment order-based clustering in this context. Although hierarchical clustering was earlier employed for gene clustering to identify compositionally atypical genes (Rajeev K Azad & Lawrence, 2007), it was not tested on variable-length genomic fragments outputted by recursive segmentation program. Diametrically opposite to agglomerative clustering is network clustering; it partitions the network to identify clusters rather than builds bottom up as with agglomerative clustering. Network clustering brought a new perspective in addressing this problem, as this is conceptually different to often invoked agglomerative clustering. Top down approach that performs partitioning or segmentation of an organized dataset (e.g. a genome or a network) was earlier found to be more effective in localizing horizontally acquired regions in bacterial genomes in comparison to bottom up methods such as agglomerative clustering (Arvey et al., 2009). The horizontally acquired genomic segments were earlier identified by assessing the atypicality of each segment against the genome background (Arvey et al., 2009). Later studies showed that clustering following segmentation yields even more promising results, however, only agglomerative clustering approach was tested (Rajeev K. Azad & Li, 2013). The problem with bottom up (agglomerative) approach is that members of earlier formed clusters are not allowed to be reassigned at later rounds of clustering (except for merger of two clusters into a new cluster) when additional information emerges that may rectify any past misassignments. If clusters are not refined at successive steps, impurities in the clusters may only amplify as clustering proceeds. Both segment order based and hierarchical clustering approaches are prone to be affected by this. Whereas reassignment is itself a non-trivial problem, another issue that afflicts segment order based agglomerative clustering is that clustering configuration varies by how the grouping is initiated (i.e., from 5'-end or 3'-end of a genome). This is not unexpected, as earlier rounds of clustering dictate the later rounds and thus, the directionality impacts the clustering. Hierarchical clustering, however, proceeds independent of directionality.

Deciphering the underlying structure using agglomerative clustering has its pitfalls, however, this may still yield reasonably good outcomes. This was realized while probing genomic islands in bacterial genomes by previous studies that led to the development of a top down, recursive partitioning approach that allowed assessment of entire information content to successively generate compositionally homogeneous segments (Arvey et al., 2009). Here, we explored whether a top down rather than bottom up approach could again be invoked to more efficiently reconstruct the chimeric structure of bacterial genomes following segmentation. We posited that this could be feasible within a network framework that will allow partitioning the network of segments into modules or clusters of similar segments. Implementation of networking clustering and comparative assessment revealed its promising aspects, at the same time it also highlighted the complementarity of different approaches. Whereas agglomerative clustering performed well in discriminating between native and alien DNA at genome optimized thresholds, the optimal threshold settings varied among genomes. Considering that optimizing a clustering program on a just sequenced, anonymous genome may not be feasible, we assessed the clustering methods for their performance averaged over the test genomes across the entire threshold range for each method. Our intent was to identify a threshold range for each method where its maximal or close to maximal overall performance could be attained. We observed network clustering's overall performance maximizing within a certain inflation parameter range (averaging 25.7 ± 3.7 for native and 25.1 ± 3.7 for alien DNA identification, Figures 4.1, 4.3 and Tables 4.1, 4.2), reaching a maximal higher than by other methods for both native and alien identification. This demonstrates that top-down partitioning post segmentation indeed yields a better outcome. However, the accuracy is lower over a large range than agglomerative clustering before it begins to spike towards maximal that appeared more stable with native than alien DNA identification. Agglomerative clustering's performance is relatively less variable over the entire threshold range and reaches the maximal at low p-values, as expected (right end of hierarchical clustering and segment order clustering curves in Figures 4.1 and 4.3). These results highlight the strengths and weaknesses of different approaches and future studies could focus on exploiting the complementary strengths of different clustering approaches to develop an integrative method for native and alien DNA identification. For example– as agglomerative clustering's performance is less variable and performance optimal in low p-value range, one can first use agglomerative clustering to get a cue of cluster structure inherent to a genome of interest and then follow up with network clustering to "fine-tune" to optimal or nearly optimal cluster configuration. This will help exploit the benefits of network approach, while obviating low accuracy risk as could occur within certain inflation parameter ranges with network clustering (Figures 4.1 and 4.3).

In addition to estimating native and alien nucleotides in bacterial genomes, estimating donors is another problem where the power of clustering can be leveraged upon. Clustering methods may be expected to group segments from each donor within a unique cluster but a donor genome may be chimeric and therefore multiple clusters may also be expected for a donor. Here, as we used genomes modeled after the respective cores of the bacterial genomes, we expect methods to group segments from a donor into a single cluster. Note that the genomes were modeled on cores that were obtained by removing compositionally atypical genes; a core thus represents the mutational proclivity of the recipient genome with the composition shaped by the directional mutational pressure specific to that genome. Because of the superior performance of agglomerative clustering in identifying alien DNA while preserving donor identities in the major clusters of the donors, one can use this for more reliably estimating the donors. In summary, agglomerative and network clustering approaches possess complementary strengths that can exploited to understand different facets of bacterial chimerism, which may not be possible with any single method. Future efforts could focus on integrating different approaches towards the goal of still better interpretation of bacterial genomes.

CHAPTER 5

SUMMARY, DISCUSSION AND FUTURE DIRECTIONS

5.1 Metagenomic Taxonomic Labeling Assessment and Applications

In this work, we revisited the alignment-free approach to taxonomic classification of metagenomic datasets, assessed the strengths and weaknesses of different alignment-free and alignment-based methods, and leveraged the insights gained to develop a platform for robust taxonomic profiling of metagenomic sequences. This platform allows standalone use or use in concert with the existing methods. By using pseudo-count supplemented higher order Markov models, our SMM algorithm compares favorably with more sophisticated programs such as PhymmBL, both in terms of overall accuracy and computational efficiency. Overall, SMM also outperformed variants of interpolated Markov model, including IMM, ICM, and DIM in classifying metagenomic sequences. Simulated metagenomes allowed assessment of the performance of SMM relative to other methods that include alignment, demonstrating its utility in classifying reads ranging in size from 100 nt to 250 nt. Compared to ICM, SMM is less sensitive to sequencing errors, based on Illumina error models (Burks & Azad, 2020).

SMM assigns all reads to taxonomic lineages based on the best hit (highest scoring model) in the model database. However, this creates several problems. SMM assigns taxonomic identities to all reads, regardless of whether read originating taxa are represented in the database or not. Misclassification will amplify with the increasing number of reads from source taxa not represented in the database. Furthermore, the best hit may not have a score that could be a strong indicator of match to that taxon. False-positives are a problem inherent to Markov model classification (Wood & Salzberg, 2014). Raw scores generated by Markov models do not offer much beyond selecting the model that yields the highest score, however, this can lead to issues just mentioned above. Attempts have been made to establish score thresholds for classification purposes, but it requires a formulation that accounts for variables such as read length, model order, and taxonomic level. While this problem has been addressed for the ICM scores of PhymmBL using a 3D-curve fitting (Brady & Salzberg, 2011), we found that this approach does not work well with SMMs, and furthermore, their reliability has been called into question (Lan et al., 2012).

Machine-learning provides a viable framework for establishing thresholds for the raw scores of SMMs, but requires extensive forethought concerning the quality of training data. Such data must contain sets of reads of variable sizes, as observed in real metagenomic datasets. Taxonomic representation must be balanced. Even within the same genome, scores by a model for reads from different regions could be significantly different. This is expected as microbial genomes are known to be mosaic. Horizontally acquired sequences that have evolved in different genomic contexts prior to acquisition in a genome differ in composition from each other and from the vertically transmitted regions in the genome. Furthermore, the training data sourced from public repositories may be littered with contaminant sequences (O'Leary et al., 2016). Lacking manual curation, adapter sequences and extraneous DNA are so common that the developers of Kraken maintain their own *k*-mer databases and warn users of the potentially misleading conclusions that can be drawn from customized, uncurated databases (Wood et al., 2019; Wood & Salzberg, 2014).

To identify distinct clusters of DNA segments within a genome, we adapted the MJSD based segmentation and clustering algorithm implemented in the IslandCAFE program for genomic island prediction (Jani & Azad, 2019). Our C++ version of this program augments the computational efficiency, making possible analysis of thousands of genomes in a fraction of the time needed by the original version. Over 29,000 genomes, each representative of a unique species in RefSeq, were individually segmented and clustered using this algorithm, and reads were sampled from the clusters to construct metagenomic training datasets. These datasets were then used to train taxon and model order-specific logistic regression estimators, optimized within a Bayesian framework using a 6-fold cross-validation procedure. The regression model generated scores were used to perform taxonomic assignment after establishing thresholds for the assignment. Benchmarking of clustering algorithms assisted us in selecting the clustering algorithm for this purpose. A consistently stable performance by segment-order based clustering made it an appropriate choice for this task.

Combining logistic regression with SMM led to the development of POSMM. To our knowledge, POSMM is currently among the most accurate classifiers for WMS reads. POSMM's performance rivals that of the most popular alignment-free tools currently available, both in terms of the accuracy and computational speed. The multiprocessing library of Python allows for linear speedups by splitting model generation across all available CPU cores of the server. POSMM first implement SMM to generate raw scores for reads, which are then normalized using logistic regression allowing thresholding by users to yield high confidence classification, thus avoiding the pitfall of numerous misclassifications otherwise done by alignment-free classifiers. The ability of POSMM to build models on the fly and thus save time in loading the model data from a hard drive allows it to quickly accomplish the classification task, and when used in concert with Kraken, increase the sensitivity significantly. When using both POSMM and Kraken in concert, their complementary strengths are leveraged to achieve an overall accuracy (F1-score) higher than by either method alone.

5.2 Future Directions

5.2.1 Use of Cluster Models to Augment Taxonomic Profiling

Efforts so far have focused on building a single model for each genome, which represents unique signature of each genome. However, this ignores the fact that microbial genomes are often chimeric, that is, composed of genes of different ancestries. By performing segmentation and clustering for a genome, the mosaic compositional organization of the genome is revealed. We posit that models trained on clusters generated by our segmentation and clustering algorithm will adequately account for genomic variability and will aid in more robust classification of reads when queried against the database of ensembles of models for each genome. Reads from strain-specific regions may thus be classified correctly, which may not be possible using single genome models that represent the summary statistics over the genome. This will require re-addressing the model order, as clusters provide less training data, and are therefore more sensitive to the presence of pseudocounts, particularly when higher order models are used. Interpolation may help in this scenario; any progress in this direction will go a long way towards achieving strain-level classification, arguably among the most difficult tasks in metagenomics, yet possible with the advances in WMS sequencing and sequence analysis methods.

5.2.2 Metagenomic Binning via MJSD Clustering

Binning methods that group metagenomic reads into “bins” based on their similarity have been developed. These methods do not strive to assign taxonomic identities to reads or bins. Indeed, the clustering algorithms presented here can be used for metagenomic binning. In our preliminary studies, network clustering based on similarity scores (assessed in terms of significance values for MJSD) was able to bin reads to the species level. The computational and memory requirements of this analysis could be prohibitive and will require, at the very least, the parallel processing potential of GPUs. While GPU-based correlation programs such as Fast-GPU-PCC do exist (Eslami & Saeed, 2018), they are still bottlenecked by I/O and upstream CPU loading of data into memory. Novel developments such as Nvidia's new GPUDirect Storage interface could bypass these limitations, depending on how well these new technologies perform as well as their implementation in libraries such as CUDA (A. Li et al., 2020).

5.2.3 POSMM Regression Model Updates

The performance of regression model used in POSMM depends on the quality of training data used to estimate the parameters. We used all available species, in each case choosing the highest quality genome release, in the construction of our training data. However, we understand that in the future, genomes of more species will continue to populate RefSeq, and logistic regression estimators employed by POSMM could become even more reliable with expanding training data. To account for this, we developed all regression models in POSMM using the scikit-learn Python library, and everything is stored and loaded using JSON format. Users can easily exchange their own scikit-learn models as long as they are stored in JSON. This includes the logistic regression model, but can be extended to any machine learning classifier that uses the probability prediction function (logistic regression, SVM, random forests).

5.2.4 Architectural Independence of SMM

SMM is currently written in C++, and is entirely reliant upon standard libraries introduced with C++11. This includes a hash function that directs count and probability values to 2D vector indices. The program is also safely optimized beyond O3, producing exact matches at O2 through O4 optimization under the GCC compiler when tested across ~29,000 genomes and ~250,000 read combinations. Keeping the codebase simplified was intentional. While much of the core of SMM could have been more quickly integrated using non-standard data structures and hash-maps, our codebase lends itself to cross-architectural compilation with minimal adjustments necessary. With companies such as Apple fully transitioning their personal computing lines to ARM in the coming years, we believe POSMM will be one of the first ARM-compatible metagenomic classifiers available on the new MacOS platform. The Python aspects of POSMM were similarly kept as close to standard libraries as possible, with the exception of the scikit-learn libraries that are used for estimating the logistic regression parameters. Scikit-learn, however, does have an ARM port, and should also lend itself to easy cross-compilation with minimal code adjustment. Much like its namesake animal, POSMM is highly adaptable. Not only in regard to downstream taxonomic classification, POSMM is even adaptable to the systems on which it is to be executed.

5.2.5 Conclusions

Altogether, our work presents a next significant step in metagenomic taxonomic classification. As the WMS database continues to grow, alignment has become the de-facto method for metagenomic sequence classification by virtue of its unmatched throughput. Ultrafast taxonomic classifiers based on exact *k*-mer matches can classify millions of reads per minute. Their shortcomings, however, are apparent even in the benchmarks reported in their publications. Specifically, alignment-based methods fail to perform the same level of taxonomic abstraction as alignment-free methods, particularly when the closest available relative is several taxonomic ranks separated from the originating taxon of a read. While alignment-free methods have existed for years, their lack of throughput renders them not amenable to interrogation of metagenomic datasets. Furthermore, many of these methods that base their inference merely on highest scoring models are prone to large numbers of misclassification, and furthermore, because of the same reason, they lack an innate ability to leave reads, whose originating taxa are not represented in the database, unclassified. In the same way that WMS sequencing addresses the shortcomings of 16S amplicon sequencing, so does POSMM attempt to address the shortcomings of current taxonomic classifiers. Our platform, driven by the SMM algorithm and logistic regression, serves as a standalone tool or in concert with existing classifiers such as Kraken to assign taxonomic identities to metagenomic reads. Built with the future in mind, we envision POSMM becoming integral to metagenome profiling pipelines, and contribute towards better understanding of microbial communities dwelling different environments.

REFERENCES

Abby, S. S., Tannier, E., Gouy, M., & Daubin, V. (2012). Lateral gene transfer as a support for the tree of life. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(13), 4962–7. https://doi.org/10.1073/pnas.1116871109

Acinas, S. G., Marcelino, L. A., Klepac-Ceraj, V., & Polz, M. F. (2004). Divergence and redundancy of 16S rRNA sequences in genomes with multiple rrn operons. *Journal of Bacteriology*, *186*(9), 2629–35. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15090503

Ainsworth, D., Sternberg, M. J. E., Raczy, C., & Butcher, S. A. (2017). k-SLAM: accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets. *Nucleic Acids Research*, *45*(4), 1649–1656. https://doi.org/10.1093/nar/gkw1248

Almeida, A., Mitchell, A. L., Tarkowska, A., & Finn, R. D. (2018). Benchmarking taxonomic assignments based on 16S rRNA gene profiling of the microbiota from commonly sampled environments. *GigaScience*, *7*(5), 1–10. https://doi.org/10.1093/gigascience/giy054

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Anantharaman, K., Breier, J. A., & Dick, G. J. (2016). Metagenomic resolution of microbial functions in deep-sea hydrothermal plumes across the Eastern Lau Spreading Center. *The ISME Journal*, *10*(1), 225–239. https://doi.org/10.1038/ismej.2015.81

Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P., & Tyson, G. W. (2012). Grinder: A versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research*, *40*(12). https://doi.org/10.1093/nar/gks251

Arvey, A. J., Azad, R. K., Raval, A., & Lawrence, J. G. (2009). Detection of genomic islands via segmental genome heterogeneity. *Nucleic Acids Research*, *37*(16), 5255–66. https://doi.org/10.1093/nar/gkp576

Ashelford, K. E., Chuzhanova, N. A., Fry, J. C., Jones, A. J., & Weightman, A. J. (2005). At least 1 in 20 16S rRNA sequence records currently held in public repositories is estimated to contain substantial anomalies. *Applied and Environmental Microbiology*, *71*(12), 7724–36. https://doi.org/10.1128/AEM.71.12.7724-7736.2005

Azad, R. K., & Borodovsky, M. (2004). Effects of choice of DNA sequence model structure on gene identification accuracy. *Bioinformatics*, *20*(7), 993–1005. https://doi.org/10.1093/bioinformatics/bth028

Azad, R. K., & Lawrence, J. G. (2005). Use of artificial genomes in assessing methods for atypical gene detection. *PLoS Computational Biology*, *1*(6), e56. https://doi.org/10.1371/journal.pcbi.0010056

Azad, R. K., & Lawrence, J. G. (2007). Detecting laterally transferred genes: use of entropic clustering methods and genome position. *Nucleic Acids Research*, *35*(14), 4629–39. https://doi.org/10.1093/nar/gkm204

Azad, R. K., & Lawrence, J. G. (2012). Detecting laterally transferred genes. *Methods in Molecular Biology (Clifton, N.J.)*, *855*, 281–308. https://doi.org/10.1007/978-1-61779-582-4_10

Azad, R. K., & Li, J. (2013). Interpreting genomic data via entropic dissection. *Nucleic Acids Research*, *41*(1), e23. https://doi.org/10.1093/nar/gks917

Besemer, J. (2001). GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, *29*(12), 2607–2618. https://doi.org/10.1093/nar/29.12.2607

Bodenhofer, U., Kothmeier, A., & Hochreiter, S. (2011). APCluster: an R package for affinity propagation clustering. *Bioinformatics*, *27*(17), 2463–4. https://doi.org/10.1093/bioinformatics/btr406

Boto, L. (2010). Horizontal gene transfer in evolution: facts and challenges. *Proceedings of the Royal Society B: Biological Sciences*, *277*(1683), 819–827. https://doi.org/10.1098/rspb.2009.1679

Boys, R. J., & Henderson, D. A. (2004). A Bayesian approach to DNA sequence segmentation. *Biometrics*, *60*(3), 573-81–8. https://doi.org/10.1111/j.0006-341X.2004.00206.x

Brady, A., & Salzberg, S. (2011). PhymmBL expanded: confidence scores, custom databases, parallelization and more. *Nature Methods*, *8*(5), 367. https://doi.org/10.1038/nmeth0511-367

Brady, A., & Salzberg, S. L. (2009). Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods*, *6*(9), 673–676. https://doi.org/10.1038/nmeth.1358

Braun, J. V., & Muller, H.-G. (1998). Statistical methods for DNA sequence segmentation. *Statistical Science*, *13*(2), 142–162.

Břinda, K., Sykulski, M., & Kucherov, G. (2015). Spaced seeds improve k -mer-based metagenomic classification. *Bioinformatics*, *31*(22), 3584–3592. https://doi.org/10.1093/bioinformatics/btv419

Brooks, J. P., Edwards, D. J., Harwich, M. D., Rivera, M. C., Fettweis, J. M., Serrano, M. G., … Buck, G. A. (2015). The truth about metagenomics: quantifying and counteracting bias in 16S rRNA studies. *BMC Microbiology*, *15*(1), 66. https://doi.org/10.1186/s12866-015-0351-6

Buchfink, B., Xie, C., & Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, *12*(1), 59–60. https://doi.org/10.1038/nmeth.3176

Burks, D. J., & Azad, R. K. (2020). Higher-order Markov models for metagenomic sequence classification. *Bioinformatics (Oxford, England)*, *36*(14), 4130–4136. https://doi.org/10.1093/bioinformatics/btaa562

Burstein, D., Harrington, L. B., Strutt, S. C., Probst, A. J., Anantharaman, K., Thomas, B. C., … Banfield, J. F. (2016). New CRISPR–Cas systems from uncultivated microbes. *Nature*, *542*(7640), 237–241. https://doi.org/10.1038/nature21059

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics (Oxford, England)*, *34*(17), i884–i890. https://doi.org/10.1093/bioinformatics/bty560

Chen, Y., Ye, W., Zhang, Y., & Xu, Y. (2015). High speed BLASTN: an accelerated MegaBLAST search tool. *Nucleic Acids Research*, *43*(16), 7762–7768. https://doi.org/10.1093/nar/gkv784

Chung, M., Munro, J. B., Tettelin, H., & Dunning Hotopp, J. C. (n.d.). Using Core Genome Alignments To Assign Bacterial Species. *mSystems*, *3*(6). https://doi.org/10.1128/mSystems.00236-18

Corvelo, A., Clarke, W. E., Robine, N., & Zody, M. C. (2018). taxMaps: Comprehensive and highly accurate taxonomic classification of short-read data in reasonable time. *Genome Research*, *28*(5), 751–758. https://doi.org/10.1101/gr.225276.117

Dagan, T., Artzy-Randrup, Y., & Martin, W. (2008). Modular networks and cumulative impact of lateral transfer in prokaryote genome evolution. *Proceedings of the National Academy of Sciences*, *105*(29), 10039–10044. https://doi.org/10.1073/pnas.0800679105

Daubin, V. (2002). A Phylogenomic Approach to Bacterial Phylogeny: Evidence of a Core of Genes Sharing a Common History. *Genome Research*, *12*(7), 1080–1090. https://doi.org/10.1101/gr.187002

Delcher, A. (1999). Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, *27*(23), 4636–4641. https://doi.org/10.1093/nar/27.23.4636

Delcher, A. L., Bratke, K. A., Powers, E. C., & Salzberg, S. L. (2007). Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics (Oxford, England)*, *23*(6), 673–9. https://doi.org/10.1093/bioinformatics/btm009

DeSantis, T. Z., Hugenholtz, P., Larsen, N., Rojas, M., Brodie, E. L., Keller, K., … Andersen, G. L. (2006). Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. *Applied and Environmental Microbiology*, *72*(7), 5069–5072. https://doi.org/10.1128/AEM.03006-05

Dey, N., Wagner, V. E., Blanton, L. V, Cheng, J., Fontana, L., Haque, R., … Gordon, J. I. (2015). Regulators of gut motility revealed by a gnotobiotic model of diet-microbiome interactions related to travel. *Cell*, *163*(1), 95–107. https://doi.org/10.1016/j.cell.2015.08.059

Ding, X., Cheng, F., Cao, C., & Sun, X. (2015). DectICO: An alignment-free supervised metagenomic classification method based on feature extraction and dynamic selection. *BMC Bioinformatics*, *16*(1), 1–12. https://doi.org/10.1186/s12859-015-0753-3

Dongen, S. van. (2000). *Graph Clustering by Flow Simulation*. University of Utrecht. Retrieved from http://www.library.uu.nl/digiarchief/dip/diss/1895620/inhoud.htm

Eslami, T., & Saeed, F. (2018). Fast-GPU-PCC: A GPU-Based Technique to Compute Pairwise Pearson’s Correlation Coefficients for Time Series Data—fMRI Study. *High-Throughput*, *7*(2), 11. https://doi.org/10.3390/ht7020011

Essen, U., & Steinbiss, V. (1992). Cooccurrence smoothing for stochastic language modeling. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 161–164 vol.1). IEEE. https://doi.org/10.1109/ICASSP.1992.225947

Finn, R. D., Clements, J., & Eddy, S. R. (2011). HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, *39*(suppl), W29–W37. https://doi.org/10.1093/nar/gkr367

Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., … Bateman, A. (2015). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, *44*(December 2015), gkv1344. https://doi.org/10.1093/nar/gkv1344

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*(5814), 972–976. https://doi.org/10.1126/science.1136800

Gionis, A., & Mannila, H. (2003). Finding recurrent sources in sequences. In *Proceedings of the seventh annual international conference on Computational molecular biology - RECOMB ’03* (pp. 123–130). New York, New York, USA: ACM Press. https://doi.org/10.1145/640075.640091

Gregor, I., Dröge, J., Schirmer, M., Quince, C., & McHardy, A. C. (2016). PhyloPythiaS+ : a self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes. *PeerJ*, *4*, e1603. https://doi.org/10.7717/peerj.1603

Handelsman, J. (2004). Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and Molecular Biology Reviews : MMBR*, *68*(4), 669–85. https://doi.org/10.1128/MMBR.68.4.669-685.2004

Handelsman, J., Rondon, M. R., Brady, S. F., Clardy, J., & Goodman, R. M. (1998). Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & Biology*, *5*(10), R245–R249. https://doi.org/10.1016/S1074-5521(98)90108-9

Hillmann, B., Al-Ghalith, G. A., Shields-Cutler, R. R., Zhu, Q., Gohl, D. M., Beckman, K. B., … Knights, D. (2018). Evaluating the Information Content of Shallow Shotgun Metagenomics. *mSystems*, *3*(6). https://doi.org/10.1128/mSystems.00069-18

Hofer, U. (2018). The majority is uncultured. *Nature Reviews Microbiology*, *16*(12), 716–717. https://doi.org/10.1038/s41579-018-0097-x

Jani, M., & Azad, R. K. (2019). IslandCafe: Compositional Anomaly and Feature Enrichment Assessment for Delineation of Genomic Islands. *G3 (Bethesda, Md.)*, *9*(10), 3273–3285. https://doi.org/10.1534/g3.119.400562

Jani, M., Mathee, K., & Azad, R. K. (2016). Identification of Novel Genomic Islands in Liverpool Epidemic Strain of Pseudomonas aeruginosa Using Segmentation and Clustering. *Frontiers in Microbiology*, *7*, 1210. https://doi.org/10.3389/fmicb.2016.01210

Jovel, J., Patterson, J., Wang, W., Hotte, N., O'Keefe, S., Mitchel, T., … Wong, G. K.-S. K.-S. (2016). Characterization of the Gut Microbiome Using 16S or Shotgun Metagenomics. *Frontiers in Microbiology*, *7*, 459. https://doi.org/10.3389/fmicb.2016.00459

Juhas, M., van der Meer, J. R., Gaillard, M., Harding, R. M., Hood, D. W., & Crook, D. W. (2009). Genomic islands: tools of bacterial horizontal gene transfer and evolution. *FEMS Microbiology Reviews*, *33*(2), 376–393. https://doi.org/10.1111/j.1574-6976.2008.00136.x

Karlin, S. (1998). Global dinucleotide signatures and analysis of genomic heterogeneity. *Current Opinion in Microbiology*, *1*(5), 598–610. https://doi.org/10.1016/s1369-5274(98)80095-7

Karlin, S., Mrázek, J., & Campbell, A. M. (1998). Codon usages in different gene classes of the Escherichia coli genome. *Molecular Microbiology*, *29*(6), 1341–55. https://doi.org/10.1046/j.1365-2958.1998.01008.x

Keith, J. M. (2006). Segmenting eukaryotic genomes with the Generalized Gibbs Sampler. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, *13*(7), 1369–83. https://doi.org/10.1089/cmb.2006.13.1369

Kelley, D. R., Liu, B., Delcher, A. L., Pop, M., & Salzberg, S. L. (2012). Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, *40*(1), e9. https://doi.org/10.1093/nar/gkr1067

Koonin, E. V. (2016). Horizontal gene transfer: essentiality and evolvability in prokaryotes, and roles in evolutionary transitions. *F1000Research*, *5*. https://doi.org/10.12688/f1000research.8737.1

Korbel, J. O., Abyzov, A., Mu, X., Carriero, N., Cayting, P., Zhang, Z., … Gerstein, M. B. (2009). PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology*, *10*(2), R23. https://doi.org/10.1186/gb-2009-10-2-r23

Kuhn, R. (1988). Speech recognition and the frequency of recently used words. In *Proceedings of the 12th conference on Computational linguistics -* (Vol. 1, pp. 348–350). Morristown, NJ, USA: Association for Computational Linguistics. https://doi.org/10.3115/991635.991706

Ladunga, I. (2017). Finding homologs in amino acid sequences using network blast searches. *Current Protocols in Bioinformatics*, *2017*, 3.4.1-3.4.24. https://doi.org/10.1002/cpbi.34

Lan, Y., Wang, Q., Cole, J. R., & Rosen, G. L. (2012). Using the RDP classifier to predict taxonomic novelty and reduce the search space for finding novel organisms. *PloS One*, *7*(3), e32491. https://doi.org/10.1371/journal.pone.0032491

Land, M., Hauser, L., Jun, S.-R., Nookaew, I., Leuze, M. R., Ahn, T.-H., … Ussery, D. W. (2015). Insights from 20 years of bacterial genome sequencing. *Functional & Integrative Genomics*, *15*(2), 141–61. https://doi.org/10.1007/s10142-015-0433-4

Leinonen, R., Sugawara, H., Shumway, M., & International Nucleotide Sequence Database Collaboration. (2011). The sequence read archive. *Nucleic Acids Research*, *39*(Database issue), D19-21. https://doi.org/10.1093/nar/gkq1019

Li, A., Song, S. L., Chen, J., Li, J., Liu, X., Tallent, N. R., & Barker, K. J. (2020). Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect. *IEEE Transactions on Parallel and Distributed Systems*, *31*(1), 94–110. https://doi.org/10.1109/TPDS.2019.2928289

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, *25*(14), 1754–60. https://doi.org/10.1093/bioinformatics/btp324

Lloyd-Price, J., Abu-Ali, G., & Huttenhower, C. (2016). The healthy human microbiome. *Genome Medicine*, *8*(1), 51. https://doi.org/10.1186/s13073-016-0307-y

Lloyd, K. G., Steen, A. D., Ladau, J., Yin, J., & Crosby, L. (2018). Phylogenetically Novel Uncultured Microbial Cells Dominate Earth Microbiomes. *mSystems*, *3*(5). https://doi.org/10.1128/mSystems.00055-18

Lukashin, A. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research*, *26*(4), 1107–1115. https://doi.org/10.1093/nar/26.4.1107

Menzel, P., Ng, K. L., & Krogh, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, *7*, 1–9. https://doi.org/10.1038/ncomms11257

Metwally, A. A., Dai, Y., Finn, P. W., & Perkins, D. L. (2016). WEVOTE: Weighted Voting Taxonomic Identification Method of Microbial Sequences. *PLOS ONE*, *11*(9), e0163527. https://doi.org/10.1371/journal.pone.0163527

Mikheyev, A. S., & Tin, M. M. Y. (2014). A first look at the Oxford Nanopore MinION sequencer. *Molecular Ecology Resources*, *14*(6), 1097–102. https://doi.org/10.1111/1755-0998.12324

Mitchell, A. L., Almeida, A., Beracochea, M., Boland, M., Burgin, J., Cochrane, G., … Finn, R. D. (2020). MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*, *48*(D1), D570–D578. https://doi.org/10.1093/nar/gkz1035

Mitchell, A. L., Scheremetjew, M., Denise, H., Potter, S., Tarkowska, A., Qureshi, M., … Finn, R. D. (2018). EBI Metagenomics in 2017: Enriching the analysis of microbial communities, from sequence reads to assemblies. *Nucleic Acids Research*, *46*(D1), D726–D735. https://doi.org/10.1093/nar/gkx967

Moller, A. G., & Liang, C. (2017). MetaCRAST: reference-guided extraction of CRISPR spacers from unassembled metagenomes. *PeerJ*, *5*, e3788. https://doi.org/10.7717/peerj.3788

Moreno-Hagelsieb, G., & Latimer, K. (2008). Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics*, *24*(3), 319–324. https://doi.org/10.1093/bioinformatics/btm585

Moschopoulos, C. N., Pavlopoulos, G. A., Iacucci, E., Aerts, J., Likothanassis, S., Schneider, R., & Kossida, S. (2011). Which clustering algorithm is better for predicting protein complexes? *BMC Research Notes*, *4*(1), 549. https://doi.org/10.1186/1756-0500-4-549

Mukherjee, C., Beall, C. J., Griffen, A. L., & Leys, E. J. (2018). High-resolution ISR amplicon sequencing reveals personalized oral microbiome. *Microbiome*, *6*(1), 153. https://doi.org/10.1186/s40168-018-0535-z

Na, S.-I., Kim, Y. O., Yoon, S.-H., Ha, S., Baek, I., & Chun, J. (2018). UBCG: Up-to-date bacterial core gene set and pipeline for phylogenomic tree reconstruction. *Journal of Microbiology*, *56*(4), 280–285. https://doi.org/10.1007/s12275-018-8014-6

Nalbantoglu, O. U., Way, S. F., Hinrichs, S. H., & Sayood, K. (2011). RAIphy: Phylogenetic classification of metagenomics samples using iterative refinement of relative abundance index profiles. *BMC Bioinformatics*, *12*(1), 41. https://doi.org/10.1186/1471-2105-12-41

Ney, H., & Essen, U. (1991). On smoothing techniques for bigram-based natural language modelling. In *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing* (pp. 825–828 vol.2). IEEE. https://doi.org/10.1109/ICASSP.1991.150464

Nicolas, P., Bize, L., Muri, F., Hoebeke, M., Rodolphe, F., Ehrlich, S. D., … Bessières, P. (2002). Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models. *Nucleic Acids Research*, *30*(6), 1418–26. https://doi.org/10.1093/nar/30.6.1418

O'Leary, N. A., Wright, M. W., Brister, J. R., Ciufo, S., Haddad, D., McVeigh, R., … Pruitt, K. D. (2016). Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, *44*(D1), D733–D745. https://doi.org/10.1093/nar/gkv1189

Ochman, H., Lawrence, J. G., & Groisman, E. A. (2000). Lateral gene transfer and the nature of bacterial innovation. *Nature*, *405*(6784), 299–304. https://doi.org/10.1038/35012500

Ounit, R., Wanamaker, S., Close, T. J., & Lonardi, S. (2015). CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, *16*(1), 236. https://doi.org/10.1186/s12864-015-1419-2

Pagani, I., Liolios, K., Jansson, J., Chen, I.-M. A., Smirnova, T., Nosrat, B., … Kyrpides, N. C. (2012). The Genomes OnLine Database (GOLD) v.4: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, *40*(Database issue), D571-9. https://doi.org/10.1093/nar/gkr1100

Pandey, R. S., Wilson Sayres, M. A., & Azad, R. K. (2013). Detecting evolutionary strata on the human X chromosome in the absence of gametologous Y-linked sequences. *Genome Biology and Evolution*, *5*(10), 1863–1871. https://doi.org/10.1093/gbe/evt139

Patel, A., Belykh, E., Miller, E. J., George, L. L., Martirosyan, N. L., Byvaltsev, V. A., & Preul, M. C. (2018). MinION rapid sequencing: Review of potential applications in neurosurgery. *Surgical Neurology International*, *9*, 157. https://doi.org/10.4103/sni.sni_55_18

Patel, J. (2001). 16S rRNA gene sequencing for bacterial pathogen identification in the clinical laboratory. *Molecular Diagnosis*, *6*(4), 313–321. https://doi.org/10.1054/modi.2001.29158

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. https://doi.org/doi/10.5555/1953048.2078195

Pham, T. D. (2007). *Advanced computational methods for biocomputing and bioimaging*. (T. D. Pham, H. Yan, & D. Crane, Eds.). Nova Science Publishers.

Pham, T. D., & Zuegg, J. (2004). A probabilistic measure for alignment-free sequence comparison. *Bioinformatics*, *20*(18), 3455–3461. https://doi.org/10.1093/bioinformatics/bth426

Popa, O., Hazkani-Covo, E., Landan, G., Martin, W., & Dagan, T. (2011). Directed networks reveal genomic barriers and DNA repair bypasses to lateral gene transfer among prokaryotes. *Genome Research*, *21*(4), 599–609. https://doi.org/10.1101/gr.115592.110

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., … Glöckner, F. O. (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, *41*(D1), D590–D596. https://doi.org/10.1093/nar/gks1219

Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., & Lopez, R. (2005). InterProScan: protein domains identifier. *Nucleic Acids Research*, *33*(Web Server), W116–W120. https://doi.org/10.1093/nar/gki442

Quince, C., Walker, A. W., Simpson, J. T., Loman, N. J., & Segata, N. (2017). Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology*, *35*(9), 833–844. https://doi.org/10.1038/nbt.3935

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, *3*(1), 4–16. https://doi.org/10.1109/MASSP.1986.1165342

Ravenhall, M., Škunca, N., Lassalle, F., & Dessimoz, C. (2015). Inferring Horizontal Gene Transfer. *PLOS Computational Biology*, *11*(5), e1004095. https://doi.org/10.1371/journal.pcbi.1004095

Rosen, G., Garbarine, E., Caseiro, D., Polikar, R., & Sokhansanj, B. (2008). Metagenome Fragment Classification Using -Mer Frequency Profiles. *Advances in Bioinformatics*, *2008*(47), 1–12. https://doi.org/10.1155/2008/205969

Salzberg, S. L., Delcher, A. L., Kasif, S., & White, O. (1998). Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, *26*(2), 544–548. https://doi.org/10.1093/nar/26.2.544

Saul, L., & Pereira, F. (1997). Aggregate and mixed-order Markov models for statistical language processing. Retrieved from http://arxiv.org/abs/cmp-lg/9706007

Segata, N., & Huttenhower, C. (2011). Toward an Efficient Method of Identifying Core Genes for Evolutionary and Functional Microbial Phylogenies. *PLoS ONE*, *6*(9), e24704. https://doi.org/10.1371/journal.pone.0024704

Sevim, V., Lee, J., Egan, R., Clum, A., Hundley, H., Lee, J., … Woyke, T. (2019). Shotgun metagenome data of a defined mock community using Oxford Nanopore, PacBio and Illumina technologies. *Scientific Data*, *6*(1), 285. https://doi.org/10.1038/s41597-019-0287-z

Shah, N., Tang, H., Doak, T. G., & Ye, Y. (2011). Comparing bacterial communities inferred from 16S rRNA gene sequencing and shotgun metagenomics. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 165–76. https://doi.org/10.1142/9789814335058_0018

Skewes-Cox, P., Sharpton, T. J., Pollard, K. S., & DeRisi, J. L. (2014). Profile Hidden Markov Models for the Detection of Viruses within Metagenomic Sequence Data. *PLoS ONE*, *9*(8), e105067. https://doi.org/10.1371/journal.pone.0105067

Song, K., Ren, J., & Sun, F. (2019). Reads Binning Improves Alignment-Free Metagenome Comparison. *Frontiers in Genetics*, *10*. https://doi.org/10.3389/fgene.2019.01156

Stoesser, G., Baker, W., van den Broek, A., Camon, E., Garcia-Pastor, M., Kanz, C., … Vaughan, R. (2002). The EMBL Nucleotide Sequence Database. *Nucleic Acids Research*, *30*(1), 21–6. https://doi.org/10.1093/nar/30.1.21

Straiton, J., Free, T., Sawyer, A., & Martin, J. (2019). From Sanger sequencing to genome databases and beyond. *BioTechniques*, *66*(2), 60–63. https://doi.org/10.2144/btn-2019-0011

Tello-Ruiz, M. K., Stein, J., Wei, S., Preece, J., Olson, A., Naithani, S., … Ware, D. (2016). Gramene 2016: Comparative plant genomics and pathway resources. *Nucleic Acids Research*, *44*(D1), D1133–D1140. https://doi.org/10.1093/nar/gkv1179

Thakkar, J. R., Sabara, P. H., & Koringa, P. G. (2017). Exploring metagenomes using next-generation sequencing. In R. P. Singh, R. Kothari, P. G. Koringa, & S. P. Singh (Eds.), *Understanding Host-Microbiome Interactions - An Omics Approach: Omics of Host-Microbiome Association* (pp. 29–40). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-10-5050-3_3

Thakur, V., Azad, R. K., & Ramaswamy, R. (2007). Markov models of genome segmentation. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *75*(1). https://doi.org/10.1103/PhysRevE.75.011915

van Dongen, S., & Abreu-Goodger, C. (2012). Using MCL to extract clusters from networks. *Methods in Molecular Biology (Clifton, N.J.)*, *804*, 281–95. https://doi.org/10.1007/978-1-61779-361-5_15

Venter, J. C. (2004). Environmental Genome Shotgun Sequencing of the Sargasso Sea. *Science*, *304*(5667), 66–74. https://doi.org/10.1126/science.1093857

Větrovský, T., & Baldrian, P. (2013). The Variability of the 16S rRNA Gene in Bacterial Genomes and Its Consequences for Bacterial Community Analyses. *PLoS ONE*, *8*(2), e57923. https://doi.org/10.1371/journal.pone.0057923

Vinga, S., & Almeida, J. (2003). Alignment-free sequence comparison - A review. *Bioinformatics*, *19*(4), 513–523. https://doi.org/10.1093/bioinformatics/btg005

Vlasblom, J., & Wodak, S. J. (2009). Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics*, *10*(1), 99. https://doi.org/10.1186/1471-2105-10-99

Waack, S., Keller, O., Asper, R., Brodag, T., Damm, C., Fricke, W. F., … Merkl, R. (2006). Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models. *BMC Bioinformatics*, *7*, 142. https://doi.org/10.1186/1471-2105-7-142

Wang, Y., Hu, H., & Li, X. (2016). MBMC: An Effective Markov Chain Approach for Binning Metagenomic Reads from Environmental Shotgun Sequencing Projects. *Omics : A Journal of Integrative Biology*, *20*(8), 470–9. https://doi.org/10.1089/omi.2016.0081

Wheeler, T. J., & Eddy, S. R. (2013). nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, *29*(19), 2487–2489. https://doi.org/10.1093/bioinformatics/btt403

Wilson, M. C., Mori, T., Rückert, C., Uria, A. R., Helf, M. J., Takada, K., … Piel, J. (2014). An environmental bacterial taxon with a large and distinct metabolic repertoire. *Nature*, *506*(7486), 58–62. https://doi.org/10.1038/nature12959

Wilson, M. C., & Piel, J. (2013). Metagenomic Approaches for Exploiting Uncultivated Bacteria as a Resource for Novel Biosynthetic Enzymology. *Chemistry & Biology*, *20*(5), 636–647. https://doi.org/10.1016/j.chembiol.2013.04.011

Wood, D. E., Lu, J., & Langmead, B. (2019). Improved metagenomic analysis with Kraken 2. *Genome Biology*, *20*(1), 257. https://doi.org/10.1186/s13059-019-1891-0

Wood, D. E., & Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, *15*(3), R46. https://doi.org/10.1186/gb-2014-15-3-r46

Zhang, R., & Zhang, C.-T. (2004). A systematic method to identify genomic islands and its applications in analyzing the genomes of Corynebacterium glutamicum and Vibrio vulnificus CMCP6 chromosome I. *Bioinformatics (Oxford, England)*, *20*(5), 612–22. https://doi.org/10.1093/bioinformatics/btg453

Zielezinski, A., Vinga, S., Almeida, J., & Karlowski, W. M. (2017). Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, *18*(1), 1–17. https://doi.org/10.1186/s13059-017-1319-7