ABSTRACT

Title of Dissertation: DATA-DRIVEN ALGORITHMS FOR CHARACTERIZING MICROBIAL COMMUNITIES

Nidhi Shah
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Mihai Pop
Department of Computer Science

Complex microbial communities play a crucial role in environmental and human health. Traditionally, microbes have been studied by isolating and culturing them, missing organisms that cannot grow in standard laboratory conditions, and losing information about microbe-microbe interactions. With affordable high-throughput sequencing, a new field called metagenomics has emerged, that studies the genomic content of the microbial community as a whole. Metagenomics enables researchers to characterize complex microbial communities, however, many computational challenges remain with downstream analyses of large sequencing datasets. Here, we explore some fundamental problems in metagenomics and present simple algorithms and open-source software tools that implement these solutions.

In the first section, we focus on using a reference database of known organisms (and genomic segments within) to taxonomically classify sequences and estimate abundances of taxa in a metagenomic sample. We developed a "BLAST outlier detection" algorithm that identifies significant outliers within database search results.

We extended this method and developed ATLAS, which uses significant database hits to group sequences in the database into partitions. These partitions capture the extent of ambiguity within the classification of a sample. Besides taxonomically classifying sequences, we also explored the problem of taxonomic abundance profiling, i.e., estimating the abundance of different species in the community. We describe TIPP2, a marker gene-based abundance profiling method, which combines phylogenetic placement with statistical techniques to control classification accuracy. TIPP2 includes an updated set of reference packages and several algorithmic improvements over the original TIPP method.

Next, we explore how to reconstruct genomes from metagenomic samples. Despite advances in metagenome assembly algorithms, assembling reads into complete genomes is still a computationally challenging problem because of repeated sequences within and among genomes, uneven abundances of organisms, sequencing errors, and strain-level variation. To improve upon the fragmented assemblies, researchers use a strategy called binning, which involves clustering together genomic fragments that likely originate from an individual organism. We describe Binnacle, a tool that explicitly accounts for scaffold information in binning. We describe novel algorithms for estimating the scaffold-level depth of coverage information and show that variation-aware scaffolders help further improve the completeness and quality of the resulting metagenomic bins.

Finally, we explore how to organize enormous sets of sequence data generated through the surge of metagenomic studies. There have been recent efforts to organize and document genes found in microbial communities in "microbial gene catalogs".

Although gene catalogs are commonly used, they have not been critically evaluated for their effectiveness as a basis for metagenomic analyses. We investigated one such catalog and focus on both the approach used to construct this catalog and its effectiveness when used as a reference for microbiome studies. Our results highlight important limitations of the approach used to construct the catalog and call into question the broad usefulness of gene catalogs. We also recommend best practices for the construction and use of gene catalogs in microbiome studies and highlight opportunities for future research.

With the increasing data being generated in different metagenomic studies, we believe our ideas, algorithms, and software tools are well-timed with the need of the community.

DATA-DRIVEN ALGORITHMS FOR CHARACTERIZING MICROBIAL COMMUNITIES

by

Nidhi Shah

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Mihai Pop, Chair/Advisor
Professor Marine Carpuat,
Professor Robert Patro
Professor Brantley Hall
Professor Michael P. Cummings, Dean's Representative

Preface

The algorithms, tools, and results in this dissertation have been published in peer-reviewed journals. At the time of this writing, Chapters 2, 3, 4, 5, and 6 have already been published. I am extremely grateful to all my co-authors on these projects. Their dedication, knowledge, and encouragement has significantly improved the quality of my research work. Their expertise in the areas of computer science, statistics, and biology have resulted in much stronger scientific papers.

- **Chapter 2**

  - **Nidhi Shah**, Stephen F. Altschul, and Mihai Pop. "Outlier detection in blast hits." *Algorithms for Molecular Biology*, 13(1):1-9, 2018

  - **Nidhi Shah**, Stephen F. Altschul, and Mihai Pop. "Outlier detection in blast hits." In *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. My contributions include (1) design and implementation of the algorithm, (2) performing software evaluation, and (3) writing the manuscripts.

- **Chapter 3**

  - **Nidhi Shah**, Jacquelyn S. Meisel, and Mihai Pop. "Embracing ambiguity in the taxonomic classification of microbiome sequencing data." *Frontiers in*

genetics 10 (201 9): 1022. My contributions include (1) design and implementation of the algorithm, (2) performing software evaluation, and (3) writing the manuscript.

- **Chapter 4**

  - **Nidhi Shah**, Erin K. Molloy, Mihai Pop, and Tandy Warnow. "TIPP2: metagenomic taxonomic profiling using phylogenetic markers." *Bioinformatics* (2021). My contributions include (1) design of the database, (2) updating and maintaining software, and (3) writing the manuscript.

- **Chapter 5**

  - Harihara Subrahmaniam Muralidharan*, **Nidhi Shah***, Jacquelyn S. Meisel, and Mihai Pop. "Binnacle: using scaffolds to improve the contiguity and quality of metagenomic bins". *Frontiers in Microbiology* 12 (2021): 346. My contributions include (1) designing the algorithm (2) analyzing data, and (3) writing the manuscript.

- **Chapter 6**

  - Seth Commichaux*, **Nidhi Shah***, Jay Ghurye, Alexander Stoppel, Jessica A. Goodheart, Guillermo G. Luque, Michael P. Cummings, and Mihai Pop. "A critical assessment of gene catalogs for metagenomic analysis", *Bioinformatics* (2021). My contributions include (1) surveying the literature, (2) running experiments and analyzing data, and (3) writing the manuscript. In this paper, I contributed heavily in the clustering error and the abundance estimation using simulated data experiments.

Here is the list of other publications I have been involved with during my PhD.

- Albin, Dreycey, Dan Nasko, RA Leo Elworth, Jacob Lu, Advait Balaji, Christian Diaz, **Nidhi Shah** et al. "SeqScreen: a biocuration platform for robust taxonomic and biological process characterization of nucleic acid sequences of interest." In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1729-1736. IEEE, 2019.

- Olson, Nathan D., **Nidhi Shah**, Jayaram Kancherla, Justin Wagner, Joseph N. Paulson, and Hector Corrada Bravo. "metagenomeFeatures: An R package for working with 16S rRNA reference databases and marker-gene survey feature data." *Bioinformatics* 35, no. 19 (2019): 3870-3872.

- **Nidhi Shah**, Michael G. Nute, Tandy Warnow, and Mihai Pop. "Misunderstood parameter of NCBI BLAST impacts the correctness of bioinformatics workflows." *Bioinformatics* 35, no. 9 (2019): 1613-1614 (Letter to the Editor, not peer-reviewed).

  * Co-first authors

Dedication

To my family for their love and support!

To all the healthcare and frontline workers for they've sacrificed a lot to keep us safe in this pandemic!

Acknowledgments

I would like to thank all the people who have made this dissertation possible and made my graduate school experience enjoyable.

First, I would like to thank my advisor, Prof. Mihai Pop. He welcomed me to join his lab and kept my dream of pursuing PhD alive. He has been kind, forgiving, and just an amazing person to work with. I had no research experience before starting my PhD, and I am forever grateful for the time he spent patiently training and mentoring me. He always supported my exploration of new research directions and actively encouraged me to network and communicate my ideas at conference meetings. I cannot thank him enough for his guidance, care, and support through these years.

I would also like to thank my committee members - Prof. Michael Cummings, Prof. Marine Carpuat, Prof. Rob Patro, and Prof. Brantley Hall for their continued support of my research. Their ideas and feedback have been invaluable for shaping and improving this dissertation. I also deeply appreciate that they checked on me and my well-being along with keeping track of progress towards dissertation completion.

I am really lucky to have many mentors who always made time to help and support me. A very special thanks to Prof. Tandy Warnow for mentoring me and

introducing me to the field of phylogenetics. Her enthusiasm and passion for research were contagious, and I have learned a lot from her. I am grateful for her generous support and guidance, and her encouragement when I needed it the most. I would also like to thank many other professors that helped and supported me over the last many years, including Todd Treangen, Stephen Altschul, Michael Cummings, Siavash Mirarab, Aravind Srinivasan, Max Leiserson, and Héctor Corrada Bravo. Finally, I would like to thank Prof. Rajiv Gandhi - he has been my strongest supporter and without whom I would have never thought of pursuing a PhD.

I had a great pleasure collaborating with many researchers in the field. I thank them all for making this experience stimulating and fun. I have enjoyed working with all of them - Jacquelyn Meisel, Harihara Muralidharan Subramaniam, Seth Commichaux, Erin Molloy, Michael Nute, Dan Nasko, Brian Brubach, Alexander Stoppel, and Jessica Goodheart. I also thank the past, present, and honorary members of the Center for Bioinformatics and Computational Biology for enriching my graduate life. I am grateful to the UMIACS and CS staff, especially Barbara Lewis and Tom Hurst, who worked hard so that my time as a graduate student can be easier and more productive.

I would like to thank my friends in College Park - Mary Depascale, Pallabi Ghosh, Esther, Suraj Nair, Akriti Mittal, Sudha Rao, Manasij Venkatesh, Neha Joshi, Yogarshi Vyas, Karthik Abhinav, Jaideep Pathak. They gave me a sense of home away from home; navigating this PhD journey without them wouldn't have been fun. I thank my parents, Seema and Rajesh, and my siblings, Pooja, Prachi, and Aditya, for their constant source of love and support. Finally, I thank my

partner, Jay Ghurye, for everything! Thank you all!

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Microorganisms live in complex communities and play a vital role in human and environmental health [1, 2, 3]. These large and complex communities of bacteria, archaea, fungi, and viruses are collectively referred to as the microbiome. Although some microbes are pathogenic, most microbes assist and complement the functions of the host. In order to fully realize the therapeutic potential of the microbiome, it is crucial we understand how these microbes interact with each other and with their environment. Traditionally, the standard techniques for classifying microbes relied on microscopic observation of cell morphology and the use of enrichment cultures. However, most of the microbes cannot grow in standard laboratory environments and are considered "unculturable" [4, 5, 6]. With high throughput sequencing, it is not only possible to sequence the genome of a single organism but also to extract and sequence DNA from mixtures of organisms in the environment. This field of studying DNA sequencing data from mixtures of organisms is called Metagenomics [7]. It provides a way to study microbes in their environment and has the potential to advance our understanding of the microbial world.

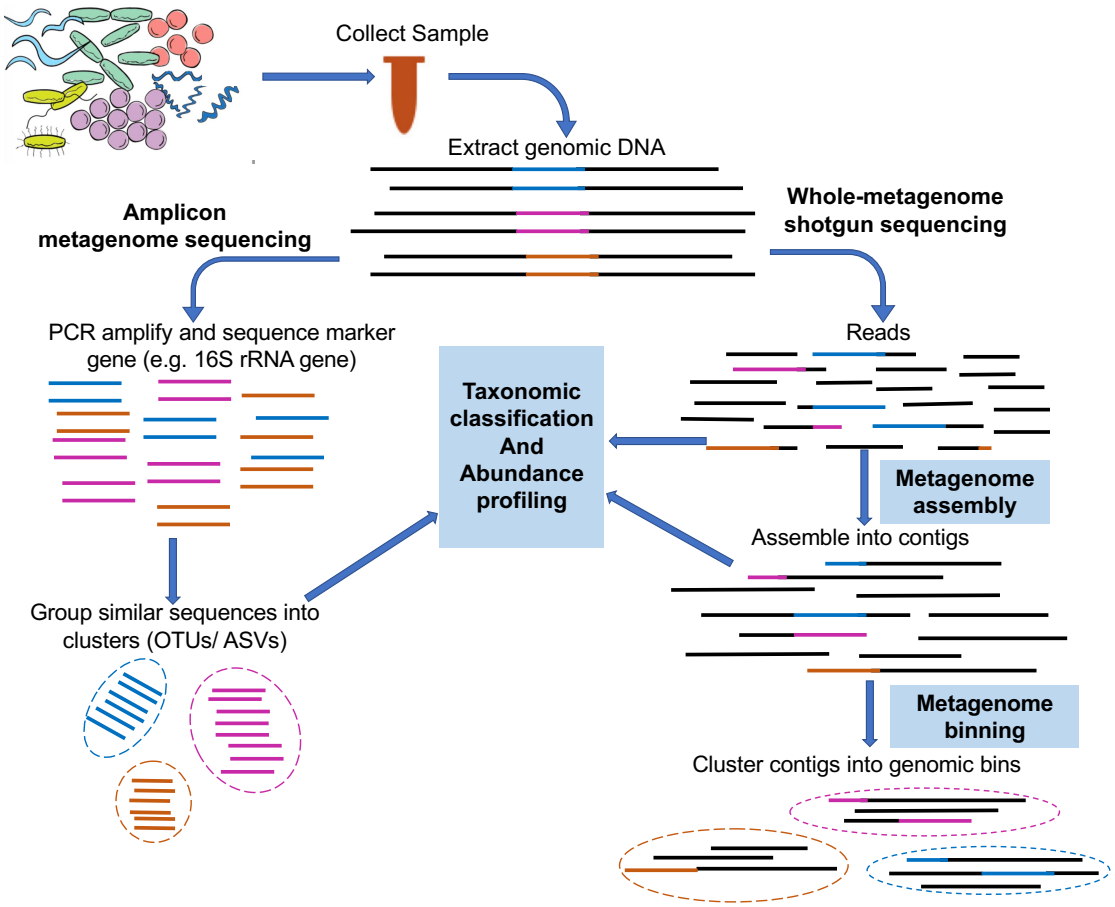

Figure 1.1: **Microbiome study workflow.** A typical metagenomic study involves collecting a sample, extracting DNA, and then sequencing. In amplicon sequencing, a gene, or a region within the gene, is targeted, amplified, and sequenced. These reads are clustered based on sequence similarity to obtain operational taxonomic units. In whole metagenomic sequencing, all DNA from sample is sequenced. Typically, reads are assembled into longer contiguous sequences (contigs). These contigs are further clustered (binned) to generate genome-level bins. Taxonomic classification annotates sequences with taxonomic labels. Abundance profiling uses reads to estimate species abundance. Note that these are just a few of the many types of analyses that can be performed on metagenomic data.

1.1 Metagenomic data

There are two widely used sequencing strategies in metagenomic studies, namely targeted amplicon sequencing and whole metagenome sequencing. Figure 1.1 shows a typical metagenomics study workflow. In targeted amplicon sequencing studies, a gene, or only a small region within the gene, is extracted and sequenced from the sample. Usually in such studies, genes that are universally found such as 16S rRNA in prokaryotes and ITS in fungi are used; such genes are called marker genes. The conserved regions of the gene allow binding to primers and the hypervariable regions help differentiate different organisms. Since we target and sequence only a small region of the genome, such data can provide information about taxonomic composition of the sample. Two of the major benefits of this approach are that it is cost effective and can leverage large repositories of already sequenced and characterized 16S rRNA gene sequences. However, there are many known problems with the approach, such as quantification errors introduced by copy-number variations [8], variable amplification efficiencies within different taxa [9, 10], different levels of resolution depending on which region of the gene is targeted [11] and generally low-resolution power of 16S rRNA sequences to differentiate between closely related species [12, 13].

In whole-metagenome sequencing data, DNA is extracted directly from a sample of microbial mixture and sequenced. This process generates millions of reads that are randomly sampled from the genomes present in the community. Whole-metagenome sequencing data resolves some biases of the targeted sequencing ap-

proach, and has the potential to capture strain-level resolution of bacterial, fungal, and viral communities. It can also help characterize the functional and metabolic potential of the sample. Compared to amplicon sequencing, it is relatively expensive to prepare, sequence, and analyze whole metagenomic sample.

1.2 Computational problems in Metagenomics

Metagenomics data analysis has many interesting computational challenges such as sequence clustering, sequence classification, abundance profiling, genome assembly, metagenome binning. Here, we describe some computational problems that are related to the dissertation work.

1.2.1 Taxonomic classification of sequences

One of the first steps in microbial characterization is taxonomic classification. Modern taxonomy was founded in the 1750s by Swedish botanist Carl Linnaeus, who worked to establish a hierarchical classification of organisms based on shared characteristics that were consistent and universally accepted. While the initial taxonomy was able to capture complex relationships between organisms, maintaining and expanding this taxonomy remains a challenge [14]. In particular, the microbial taxonomy has become significantly complex since the time of Linnaeus, most notably with the advent of next-generation sequencing technologies. In the current taxonomy, there are seven main taxonomic levels, namely Kingdom, Phylum, Class, Order, and Species.

Originally, phylogenetic approaches [15] were used to build trees to relate organisms based on how they evolved from each other. These trees were independent of taxonomic annotation and were instead generated directly from sequencing data via neighbor-joining [16], maximum parsimony [17, 18], maximum likelihood [19], or other methods. Because searching all sequences against all other sequences and establishing phylogenetic relationships for a large set of sequences is computationally expensive, we often perform taxonomic annotation by searching against a taxonomically characterized reference database instead.

1.2.2 Abundance profiling

Besides classifying sequences taxonomically, researchers are also often interested in estimating the abundance of different species in the community. This process is called taxonomic abundance profiling. Different strategies have been introduced for estimating the relative abundance of species in the sample from metagenomic data [20, 21, 22]. Note that, because sequencing is a sampling process, we can only estimate relative abundances of different taxa in the sample and not the absolute abundances. A common strategy is to classify reads by performing a homology search against taxonomically characterized reference genomes in public databases. The resulting read assignments normalized by the genome sizes can provide an estimate of relative abundance of individual species [20, 23]. An alternative strategy is to only use marker genes, which are genes that are clade-specific, unique and single-copy [21, 22, 24, 25, 26, 27]. When using marker genes, the resulting read

coverages can be used to estimate species abundance without having to normalize by genome size or copy number.

1.2.3 Metagenome binning

The data generated by the sequencing machine is often fragmented and usually contains sequencing errors. Thus, another important problem that needs to be addressed is how to best reconstruct genome sequences of organisms present in the sample. The process starts by assembling short metagenomic reads into longer contiguous sequences (contigs) based on sequence overlap. Paired-end read information can then link together and orient assembled contigs into scaffolds [28, 29, 30, 31]. However, constructing the genomes of organisms from a mixture (metagenomic assembly) is computationally challenging. The uneven abundance of organisms, repetitive sequences within and across genomes, sequencing errors, and strain-level variations within a single sample often contribute to incomplete and fragmented assemblies.

To improve upon the fragmented assemblies constructed by metagenomic assembly tools, researchers utilize a strategy called binning, which involves clustering together genomic fragments that likely originate from an individual organism. Several strategies have been proposed for metagenome binning. Classification-based approaches rely on assigning taxonomic labels to genomic contigs (as described earlier), then grouping together those contigs that share a taxonomic label [25, 32, 33, 34]. Because many of the microbes found in microbial communities are yet to be char-

acterized, classification-based approaches are limited to organisms (and genomic segments within) that are sufficiently related to known sequences. Clustering-based approaches focus on genomic features, such as GC content, oligonucleotide frequencies and contig abundance (coverage), to cluster together contigs that share similar properties [35, 36]. While such approaches are effective even when an organism shares no similarity to any known sequences, they often miss clustering genomic regions that have unusual DNA composition or that appear at higher depth of coverage than other segments of the organism of interest —situations that frequently occur in plasmids, mobile genetic elements, and highly conserved genomic segments (such as the 16S rRNA operon) [37].

1.2.4 Microbial gene catalogs

Today, researchers are interested in documenting data collected through metagenomic studies such that they are readily available to others in the community. One strategy focuses on genes found in metagenomic contigs and constructs "microbial gene catalogs". Microbial gene catalogs are data structures that organize genes found in microbial communities, providing a reference for standardized analysis of the microbes across samples and studies. Constructing a gene catalog generally involves collecting complete and fragmentary gene sequences from metagenomic samples, and then clustering them to reduce the redundancy. Typically, clustering tools that rely on heuristics, such as CD-HIT [38], are used to cluster genes at such a large scale. The first large scale gene catalog was constructed to study the diversity

of proteins found in the ocean [39]. The MetaHIT project [40] constructed a similar catalog in order to characterize the functional composition of the human gut microbiome. Following the MetaHIT catalog, gene catalogs have become ubiquitous in the analysis of metagenomic datasets, and have been created for the gut microbiota of multiple animals.

Motivated by these computational challenges in metagenomics, we present ideas, algorithms, and software to extract and interpret meaningful biological information from large datasets.

1.3 Contributions

In Chapter 2, we explore whether and when using top BLAST hits yields correct taxonomic classification. We developed a method to detect outliers among BLAST hits to separate the phylogenetically most closely related matches from matches to sequences from more distantly related organisms.

In Chapter 3, we present ATLAS, a novel strategy for taxonomic annotation that uses significant outliers within database search results to group sequences in the database into partitions. These partitions capture the extent of taxonomic ambiguity within the classification of a sample. These partitions provide better resolution than standard taxonomic levels, and improve our detection power in determining differential abundance in microbiome association studies.

In Chapter 4, we explore the problem of taxonomic abundance profiling. We present TIPP2, a marker gene-based abundance profiling method, which combines

phylogenetic placement with statistical techniques to control classification precision and recall. TIPP2 includes an updated set of reference packages and several algorithmic improvements over the original TIPP method. We find that TIPP2 provides comparable or better estimates of abundance than other profiling methods, and strictly dominates other methods when there are under-represented genomes present in the dataset.

In Chapter 5, we explore the problem of metagenome binning —clustering of contigs into genome-level bins. Existing binning algorithms often miss short contigs and contigs from regions with unusual coverage or DNA composition characteristics, such as mobile elements. We propose that information from assembly graphs can assist current strategies for metagenomic binning. We developed a tool, Binnacle, that extracts information from the assembly graphs and clusters scaffolds into comprehensive bins. We show that binning graph-based scaffolds, rather than contigs, improves the contiguity and quality of the resulting bins, and captures a broader set of the genes of the organisms being reconstructed.

In Chapter 6, we make an assessment of gene catalogs for metagenomic analyses. Although gene catalogs are commonly used, they have not been critically evaluated for their effectiveness as a basis for metagenomic analyses. As a case study, we investigate one such catalog, the Integrated Gene Catalog (IGC), however our observations apply broadly to most gene catalogs constructed to date. We focus on both the approach used to construct this catalog and, on its effectiveness, when used as a reference for microbiome studies. Our results highlight important limitations of the approach used to construct the IGC and calls into question the

broad usefulness of gene catalogs more generally. We also recommend best practices for the construction and use of gene catalogs in microbiome studies and highlight opportunities for future research.

Chapter 2: Finding the relevant database hits using outlier detection technique

This chapter contains material previously published in Outlier detection in BLAST hits [41, 42], which was a joint work with Stephen F. Altschul and Mihai Pop. NS, SFA, and MP designed the algorithm. NS developed the method and performed the experiments, with the help of MP. NS, SFA, and MP wrote the papers.

2.1 Introduction

One of the goals of metagenomic analyses is to characterize the biological diversity of microbial communities. This is usually achieved by targeted amplicon sequencing of the 16S rRNA gene, either as a whole gene or focused on a hypervariable region within the gene [43]. The 16S rRNA gene is commonly used for this purpose because it is universally found in bacteria and contains a combination of highly conserved and highly variable regions. Assigning accurate taxonomic labels to these reads is one of the critical steps for downstream analyses.

The most common approach for assigning taxonomic labels to reads involves comparing them to a database of sequences from known organisms. These similarity-based methods typically run rapidly and work well when organisms in the sample

are well represented in the database. However, a majority of microorganisms cannot be easily cultured in laboratories, and even if they are culturable, a smaller number have been sequenced. Thus, not all environmental organisms may be represented in the sequence database. This prevents the similarity-based methods from accurately characterizing organisms within a sample that are only distantly related to the sequences in the reference database. Phylogenetic-tree based methods can characterize novel organisms within a sample by statistically modeling the evolutionary processes that generated these sequences [25, 44]. However, such methods incur a high computational cost, limiting their applicability in the context of the large datasets generated in contemporary studies. Ideally, we would want to use similarity-based methods to assign labels to sequences from known organisms, and to use phylogenetic methods to assign labels to sequences from unknown organisms.

We propose a two-step method for taxonomy assignment where we use a rapid assignment method that can accurately assign labels to sequences that are well represented in the database, and then use more complex phylogenetic methods to classify only those sequences unclassified in the first step. In this work, we study whether and when a method can assign accurate taxonomic labels using a similarity search of a reference database. We employ BLAST because it is one of the most widely used similarity search methods [45]. However, it has been shown that the best BLAST hit may not always provide the correct taxonomic label [46]. Most taxonomic-assignment methods utilizing BLAST employ ad-hoc techniques such as recording the consensus label among the top five hits, or using a threshold based on E-value, percent identity, or bit score [47, 48, 49, 50]. Here we propose an alternative

approach for detecting whether and when the top BLAST hits yield correct taxonomic labels. We model the problem of separating phylogenetically correct matches from matches to sequences from similar but phylogenetically more distant organisms as a problem of outlier detection among BLAST hits. Our preliminary results involving simulated and real metagenomic datasets demonstrate the potential of employing our method as a filtering step before using phylogenetic methods.

2.1.1 Taxonomy assignment using BLAST

Several metagenomic analyses use BLAST to assign taxonomic labels to uncharacterized reads in a sample [47, 48, 49]. BLAST is a sequence similarity search tool, and it calculates an E-value and a bit score to assess the quality of each match. An E-value represents the number of hits of equal or greater score expected to arise by chance. A bit score can be understood as representing the size of the space one would need to search in order to find as strong a match by chance. However all 16S rRNA sequences are related, and therefore these scores, derived from a model of random sequences, do not provide simple information for separating sequences from different phylogenetic categories.

2.1.2 BILD scores for multiple sequence alignment

Multiple sequence alignments employ scoring functions to assess the quality of columns of aligned letters. Such functions have included Sum-of-the-Pairs (SP) scores [51], entropy scores [52], tree scores [53, 54] and the recently developed

Bayesian Integral Log-Odds (BILD) score [55, 56]. For local pairwise alignment, substitution scores are implicitly of log-odds form [57]. BILD scores extend the log-odds formalism to multiple sequence alignments. They may be used in numerous contexts such as the construction of hidden Markov model profiles, the automated selection of optimal motifs, and the selection of insertion and deletion locations, and they can inform the decision of whether to include a sequence in a multiple sequence alignment. BILD scores can also be used to classify related sequences into subclasses, as we describe below.

2.2 Methods

Broadly, our approach constructs a multiple alignment from all the top hits obtained by comparing a query sequence to a database. We use BILD scores to determine whether the multiple alignment can be split into two groups that model the data better than does a single group. In essence, we find a subset of the sequences that are more closely related to one another and to the query than to the rest of the sequences in the multiple alignment. When there is no such subset i.e. when the single alignment models the data better, we leave the query unclassified and such a query sequence is then classified in the second step by a phylogenetic method.

2.2.1 Processing query sequences

Let S be the set of sequences in the reference database, each with a taxonomic label, and Q be a set of uncharacterized reads (i.e. query sequences). We first

align each sequence in Q to sequences in S using BLAST. For each $q \in Q$, we construct the ordered set S_q that contains the segments yielding the top 100 bit scores, in decreasing order of their score. We discard all segments $l \in S_q$ where the BLAST alignment of q and l covers $\leq 90\%$ of q. We use the BLAST-generated local alignments involving q to impose a multiple alignment (M_q) on the sequences in $q \cup S_q$. We ignore all locations in the local alignment where there is an insertion in the BLAST hit sequence.

We base our score for a multiple alignment (M_q) on the Bayesian Integral Log-Odds (BILD) scores described in [55]. For each alignment column, we take the prior for the nucleotide probabilities to be a Dirichlet distribution with parameters α, and define $\alpha^* = \sum_{k=1}^{4} \alpha_k$. (Here, we always use Jeffreys' prior [58], for which all $\alpha_k = 0.5$, and $\alpha^* = 2$). For the j^{th} column M_j^q of the alignment and ignoring null characters, the log-probability of observing its particular vector of c_j^* nucleotides, with count vector c_j, is then given by

$$L(M_j^q) = \log \left[\frac{\Gamma(\alpha^*)}{\Gamma(\alpha^* + c_j^*)} \prod_{k=1}^{4} \frac{\Gamma(\alpha_k + c_{jk})}{\Gamma(\alpha_k)} \right]. \tag{2.1}$$

Here, Γ is a *gamma* function. As suggested in [55], the log-odds score for preferring a cut, at row i, of the column M_j^q into the two sub-columns X_{ji}^q and Y_{ji}^q, as illustrated in Figure 2.1, is given by

$$V_{ji}^q = L(X_{ji}^q) + L(Y_{ji}^q) - L(M_j^q). \tag{2.2}$$

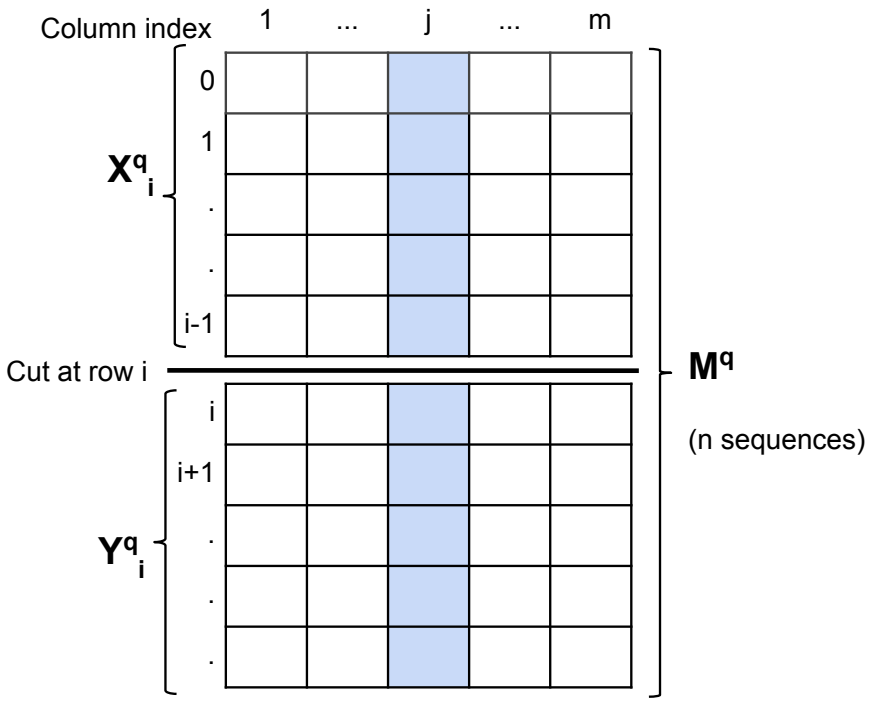

Figure 2.1: Schematic diagram of a multiple sequence alignment and how a cut divides it into two disjoint groups.

Taking all columns into account, the log-odds score for preferring a cut at row i is simply equation 2.2 summed over all columns. However, we have found it useful to give greater weight to columns with greater diversity. Thus we adopt the score V_i^q for a cut at row i given by the formula

$$V_i^q = \sum_{j=1}^{m} e_j^a V_{ji}^q, \tag{2.3}$$

where M^q has m columns, $e_j = -\sum_{k=1}^{4} (c_{jk}/c_j^*) \log_4(c_{jk}/c_j^*)$ is the entropy (base 4) of column j, and a is an arbitrary positive parameter. Note that, using this formula, perfectly conserved columns have entropy 0 and thus weight 0, whereas columns with uniform nucleotide usage have entropy 1 and thus weight 1. We have found, by experimentation, that a useful value for the parameter α is 2.7.

2.2.2 Outlier detection and taxonomy assignment

We are interested in finding the phylogenetically most closely related matches in the database to the query sequence q. We proceed by computing V_i^q for cuts with increasing i, from $i = 0$, and identify first i' for which $V_{i'}^q \geq 0$, $V_{i'}^q > V_{(i'-1)}^q$, and $V_{i'}^q > V_{(i'+1)}^q$. In other words, we find the first peak among those scores that imply the data are better explained by a split alignment. Scores below zero favor a single alignment. The first $i' - 1$ sequences from S_q we take as forming an outlier set $O_q = S_q[1 : i' - 1]$ for q. We extract the taxonomic labels of all sequences in O_q and assign the most recent common ancestor (MRCA), of these labels to q. In the case when scores favor a single alignment, we leave the query sequence unclassified.

The unclassified query sequences then should be classified, in step two of a two-step process, using a phylogenetic method.

2.3 Evaluation

2.3.1 Datasets

We used the RDP 16S rRNA gene $v16$ dataset (RTS), which has taxonomy annotated for each of its $13,212$ sequences [59], considering only the $12,320$ sequences that had taxonomic labels for all six levels - Kingdom, Phylum, Class, Order, Family, and Genus. These sequences belong to $2,320$ genera with, on average, 6 sequences per genus. To evaluate our outlier detection method, we compared taxonomic labels assigned to query sequences by our method to their true labels as given in RTS. First, we used V-Xtractor with default parameters to extract the V3, V4 and V3-V4 hypervariable regions of the sequences [60]. We then used these V3 (SIM-2), V4 (SIM-3), V3-V4 (SIM-4) and full (SIM-1) sequences as query datasets and RTS sequences as a reference database. We also used a real metagenomic dataset (Dataset-1) to study the effectiveness of our method in actual practice. Dataset-1 has 58,108 sequences from the V1-V2 hypervariable region.

2.3.2 Leave-one-out validation

In the RTS simulated dataset, we know true taxonomic labels for all query sequences. For each taxonomic level, we compare the taxonomic labels assigned by our method to the true labels to find the number of queries that are correctly

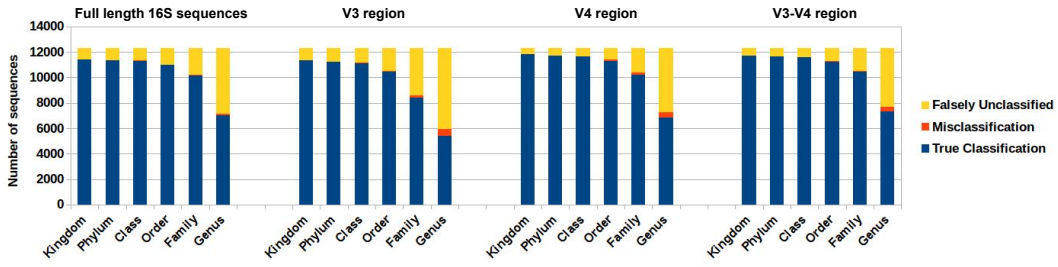

Figure 2.2: Leave-one-sequence-out validation using full-length, V3, V4, and V3-V4 region sequences from 16S rRNA dataset (RTS).

classified, misclassified or falsely unclassified. To identify correctly classified query sequences at each level, we compare, for all query sequences, the taxonomic labels assigned by our method to the true taxonomic label at that level. If the label assigned to a query by our method matches its true label, or if our method leaves the query sequence unassigned when there are no other sequences in the database with its particular label, we consider the query sequence as properly classified. For each taxonomic level, we consider misclassified those query sequences for which the assigned taxonomic label does not match the true label. We also consider falsely unclassified those sequences that were not assigned a taxonomic label at a particular level when the true label existed independently in the database. Figure 2.2 shows the number of correctly classified, misclassified and falsely unclassified sequences calculated by leave-one-out cross-validation, where we assign a taxonomic label to a query sequence (full or hypervariable region) after removing its associated sequence from the database. For all query datasets, our method rarely misclassified at all taxonomic levels, generally assigned correct labels at higher levels, but tended not to assign labels at lower levels. This may be because our method uses the MRCA of

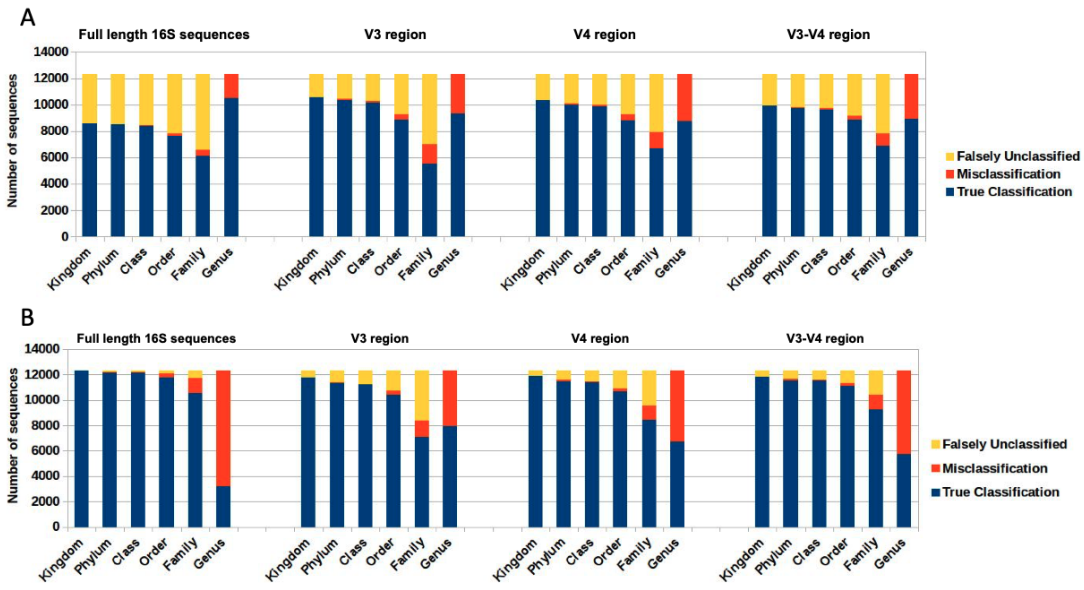

Figure 2.3: Leave-one-genus-out validation experiment results on 16S rRNA dataset (RTS). Performance of outlier detection method on full-length, V3, V4, and V3-V4 region sequences. (A). Performance of RDP classifier method on full-length, V3, V4, and V3-V4 region sequences (B).

taxonomic labels of outlier sequences. When there are closely related sequences in the database, our method chooses to be conservative by not assigning labels at lower taxonomic levels. To study the effectiveness of our method in classifying sequences with taxonomy unrepresented in the database, we performed genus-level leave-one-out cross-validation. Specifically, for each query, we removed all sequences from the database belonging to the same genus, and assigned taxonomic labels with our method and the RDP classifier [61]. We ran the RDP classifier using the QIIME [62] pipeline with the default confidence threshold of 80%. We calculated the number of queries that were correctly classified, misclassified and falsely unclassified as explained above. Figure 2.3A and B show results for our method and RDP respec-

tively. Because the genus to which a query sequence belongs is never present in the database, any label assigned at genus level will result in a misclassification error, and no assignment will result in correct classification. We observed that for higher taxonomic levels (down to Order) RDP and our method have comparable misclassification rates. However, at the Family and Genus levels, our method has a lower misclassification rate. For all datasets, RDP misclassified more query sequences at the Genus level than did our method. This is primarily because RDP aggressively tries to classify as many sequences as it can, whereas our method prefers to classify only when it can do so accurately, leaving other sequences to be dealt with later by a phylogenetic method. This experiment shows that even when sequences from the same genus as the query are absent from the database, our method has high precision and makes few mistakes.

2.3.3 Evaluation on a real 16S rRNA metagenomic dataset

To study the effectiveness of our outlier detection method in a realistic setting, we tested it on a real metagenomic dataset. Since we do not know the true taxonomic label for all query sequences, we compared our results with those produced by TIPP [25], a phylogenetic-tree based taxonomic assignment method. We used the RDP 2014 16S rRNA reference database for both methods [59]. In this dataset, there were 58,108 query sequences for which our method assigned 41,256 sequences at the Family level or below. Figure 2.4A shows that our method has a high precision for all taxonomic levels. Also, Figure 2.4B suggests that using our outlier method to

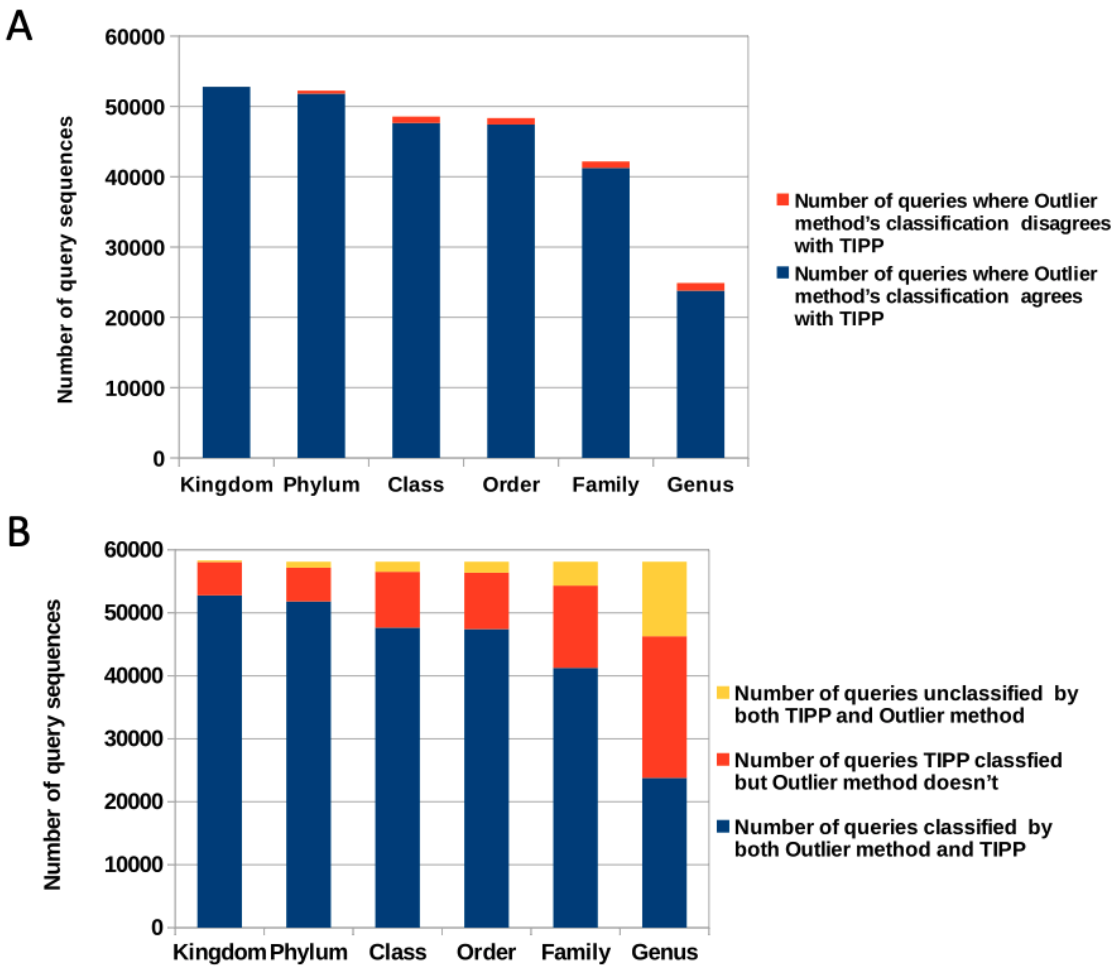

Figure 2.4: Evaluation of our outlier detection method along with TIPP on a real metagenomic dataset. Number of query sequences for which classification made by outlier detection method agrees with classification made by TIPP (A). Number of query sequences classified and unclassified by both outlier detection method and TIPP (B).

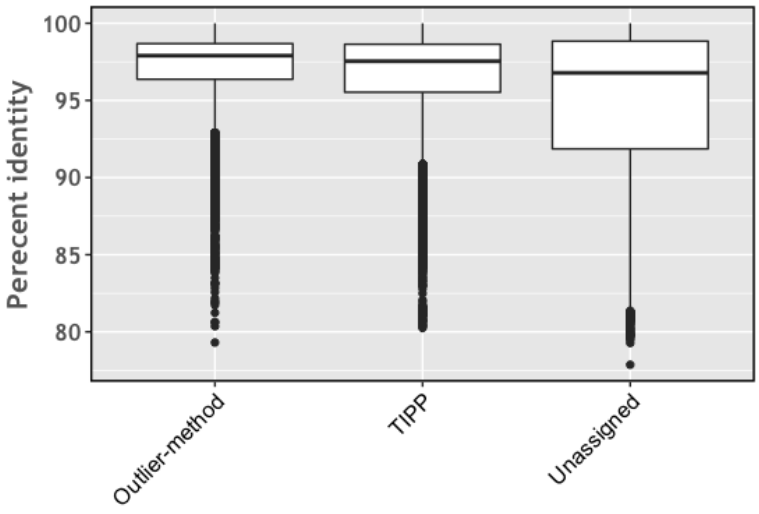

Figure 2.5: Box plot of percent identity of the best BLAST hit for all query sequences that were assigned label at genus level by our method and TIPP versus queries that remained unassigned by both methods.

make taxonomic assignments (at least down to the Family level) can significantly reduce the workload of a phylogenetic-tree based method like TIPP. A phylogenetic method can then search only in a subtree induced by database sequences in our outlier set as opposed to searching the whole tree for the best placement of the query sequence on the tree. About 11,000 sequences remained unclassified by both TIPP and our method, and we investigated whether the best BLAST hit's percent identity correlates with the ability of these programs to make classifications; see Figure 2.5. Unfortunately, there is no clear percent-identity cutoff one can employ to recognize sequences that will remain unassigned by both methods, although a large number of the unassigned sequences have low similarity to the nearest database sequence. We compared the running time of BLAST, BLAST+ outlier method, and TIPP on different input sizes. Figure 2.6 shows that both BLAST and our method

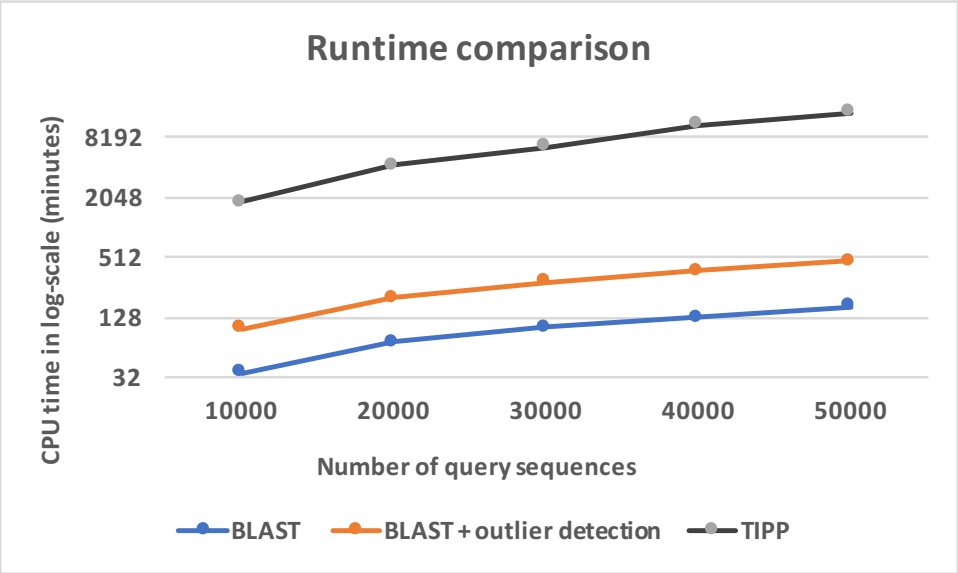

Figure 2.6: Runtime comparison of BLAST, BLAST+ outlier method and TIPP as a function of number of query sequences.

have running time growing linearly with the number of query sequences whereas the running time of TIPP increases rapidly with the increase in the number of sequences. This shows that our method can be used as a quick and accurate pre-processing step before using a phylogenetic method.

2.3.4 Distribution of outliers

Since prior approaches restrict the analysis to just a fixed number of top hits, we evaluated the number of outliers proposed by our method. As seen in Figure 2.7, the number of outliers has large variance, so a single cutoff (say, the best or top five BLAST hits) will not identify all phylogenetically related matches from the database. In this case, we relied on data for which the true taxonomic label is not known. To validate whether the set of outliers detected by our method is

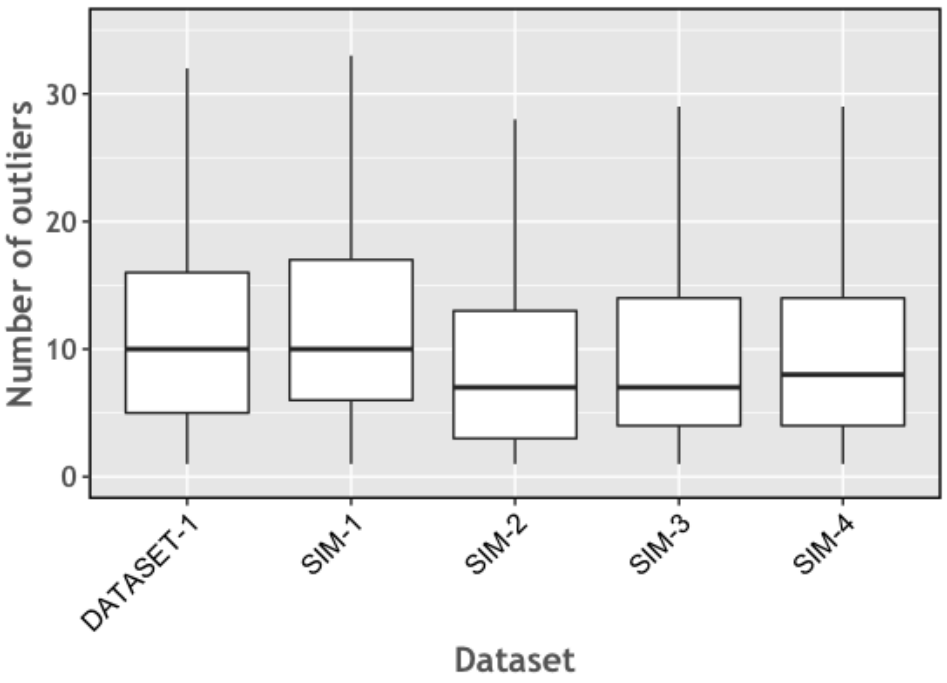

Figure 2.7: Box plot showing the variation in the number of outliers detected per query sequence in DATASET-1, SIM-1, SIM-2, SIM-3 and SIM-4

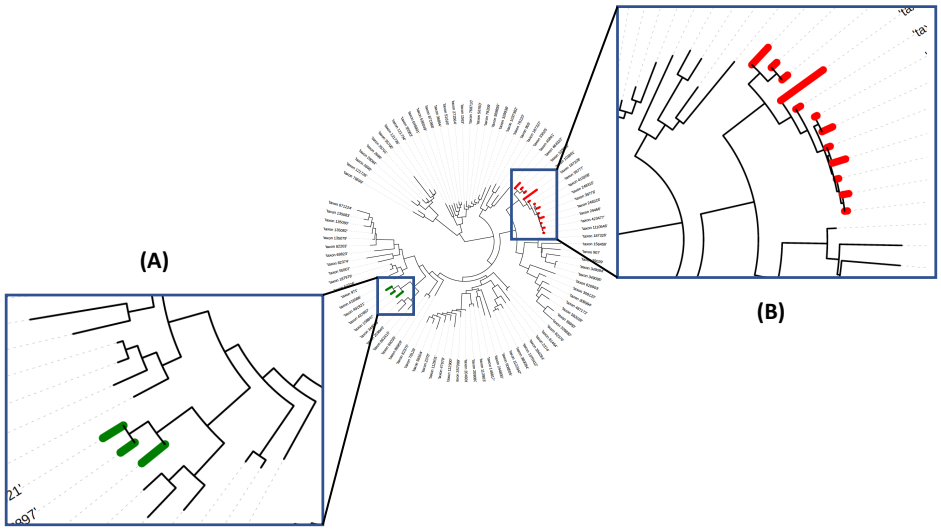

Figure 2.8: Phylogenetic tree showing outliers detected for two example query sequences. A subtree where the sequences identified as outliers are clustered closely to each other (A) and a subtree where the sequences identified as outliers cover a broader taxonomic range (B).

reasonable, and to better understand the performance of our approach, we evaluated the placement of the outlier sequences within a phylogenetic tree of the database. For this, we used the phylogenetic tree for the RDP 2014 database that was bundled in the TIPP reference package, and used the Interactive Tree Of Life web tool to visualize outliers [63]. In general, we noticed that the outliers are grouped close to each other in the phylogenetic tree (see examples in Figure 2.8), suggesting that our method produces reasonable results. This analysis also provided insights into the resolution level of the annotations produced by our method. When the outlier sequences cluster tightly within the phylogeny (Figure 2.8A), a reliable classification can be made at a low taxonomic level. When the outliers are distributed across a broader section of the tree (Figure 2.8B), the classification can only be made at a

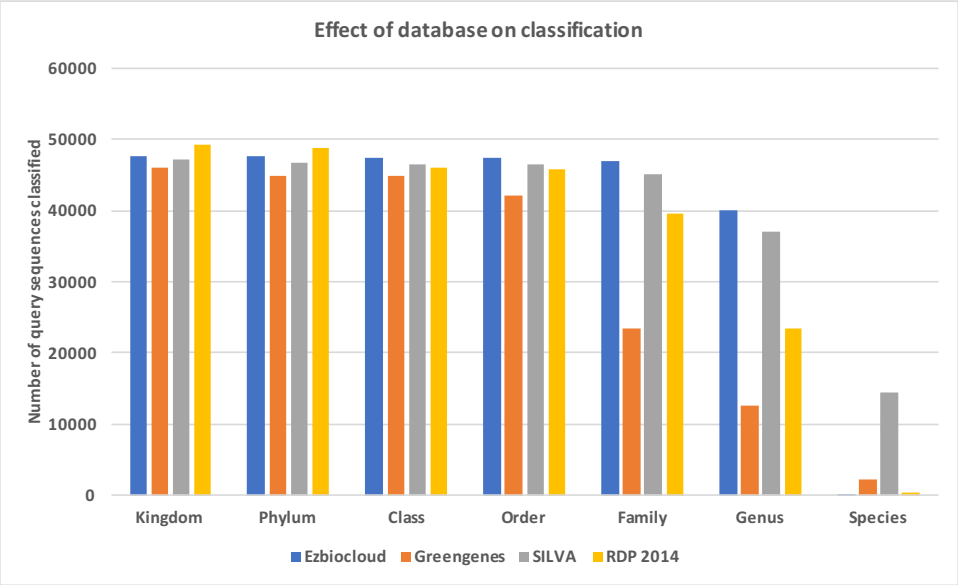

Figure 2.9: Number of query sequences classified by our method when using different databases in the BLAST search step.

higher taxonomic levels.

2.3.5 Effects of database and taxonomy

To understand the effect of the database on the final annotations provided by our method, we ran BLAST on four 16S rRNA gene databases —EzBiocloud, SILVA v.119, RDP 2014 and Greengenes on DATASET-1 [59, 64, 65, 66]. We used the Greengenes database from the QIIME package. It is known that these databases suffer from incorrect annotations. Mislabels can arise from the classification strategy used in curating the database or from errors in the current taxonomy, e.g. initial misidentification of species, or insufficient external sequence data for correctly arranging taxa [67]. Note also that these databases have different proportions of various taxa. Organisms that are well represented in a database will be classi-

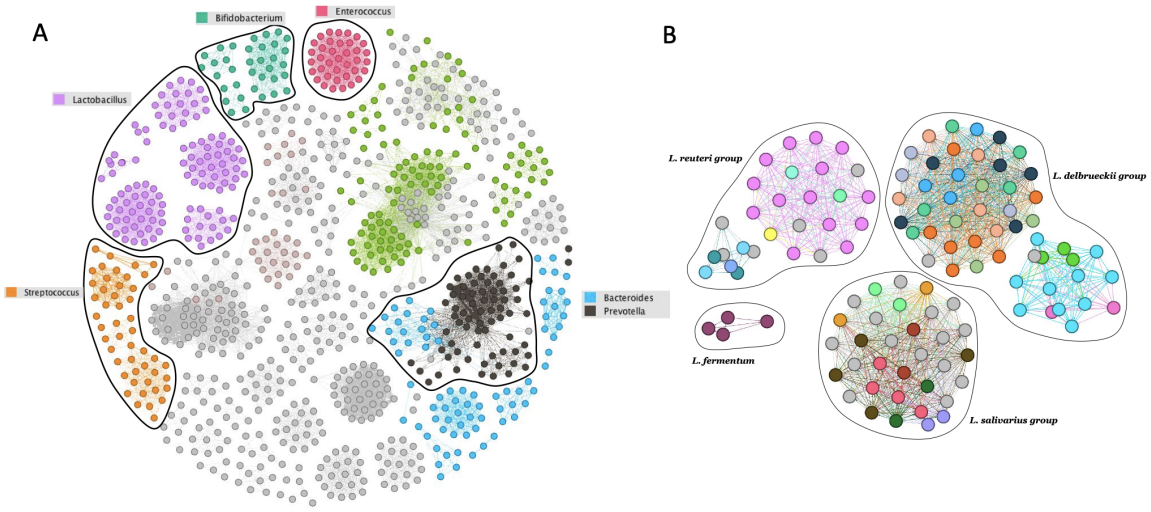

Figure 2.10: A graph where nodes are SILVA database sequences and edges between nodes are weighted by the number of query sequences from DATASET-1 for which the sequences of the two nodes are both present in the outlier set. We used the Gephi tool to visualize the graphs. The connected components, when edges of weight less than 20 are removed, and where nodes are colored by the Genus label of the sequence (A). The sub-graph showing only Lactobacillus species (B).

fied more precisely whereas under-represented organisms will have labels assigned only at higher taxonomic levels. Thus, the differences in the number of sequences annotated by our method for different databases, as shown in Figure 2.9, can be attributed primarily to the quality and the composition of the databases.

A current taxonomy may not be fully resolved and our outliers can suggest refinements. For illustration, we constructed a weighted graph whose nodes are the sequences in the SILVA database, and with edges between two nodes weighted by the number of times the nodes co-occur in an outlier set. For this analysis, we again used ∼58K query sequences from DATASET-1, keeping only the edges with weight at least 20. Figure 2.10A shows the connected components of the resulting

graph colored and grouped by Genus. We used the Gephi tool to visualize these graphs [68]. Most of the components contain nodes of same Genus. For example, there is one component for *Enterococcus*, six for *Lactobacillus*, four for *Streptococcus*, etc. However, there are some genera, such as *Bacteroides* and *Prevotella*, that have very similar regions in the V1-V2 segment of 16S rRNA gene. The query sequences matching these regions causes the edges in the graph and thus the two communities are not easily distinguished in our analysis. To analyze further the species distribution among these components, we examined the connected components for *Lactobacillus*, shown in Figure 2.10B. We found that, within each component, all species belonged to a *Lactobacillus* group as defined by Felis et al. [69] and Salvetti et al. [70]. This shows that the outliers detected by our method can provide insights for resolving and refining a taxonomy. Alternatively, a user with information on deeper taxonomic levels can infer more detailed annotation for the species found in an outlier set than is provided by our method.

2.4 Conclusion and Discussion

We propose a two-step approach for taxonomic assignment, in which we gain as much information as we reliably can from BLAST output before using computationally expensive phylogenetic-tree based methods on sequences that are difficult to classify. In this paper, we developed an outlier detection method for taxonomy assignment using BLAST hits that separates phylogenetically correct matches from matches to sequences from similar but phylogenetically more distant organ-

isms. This method can thus be used for step one of a two-step approach, to identify sequences that can be assigned accurate labels using just a BLAST search of a reference database.

Because all 16S rRNA sequences are related, statistics like E-value or bit score from BLAST do not provide ready information for separating sequences from different phylogenetic categories. Our experiments show also that a single cutoff cannot be used to select BLAST hits for correctly assigning taxonomic labels. We have experimented with finding outliers using bit score distributions, but found they provided insufficient information to detect phylogenetically correct matches (data not shown). Our experiments also show that although the percent identity of the best BLAST hit is correlated with whether a sequence is assigned a taxonomic label. However, there is not a single percent identity cutoff that can differentiate sequences that are classified from the sequences that are not classified. This has motivated our development of a BILD-score based method to identify when the top BLAST hits will yield accurate taxonomic labels.

Because our method is used as a filtering step, we seek to accurately classify as many query sequences as possible while making few misclassifications. The sequences that we leave unclassified are then to be handled by a phylogenetic method. Our results on simulated and real 16S rRNA metagenomic datasets show that our method has high precision at all taxonomic levels, assigning correct labels at higher levels to a majority of sequences, and that it is computationally efficient compared to phylogenetic-tree based taxonomic assignment methods. This demonstrates the promise of a two-step taxonomic assignment approach, using our method as a fil-

tering step.

In the future, we plan to study sequences that were classified correctly by phylogenetic methods but not by ours, to gain insight for possible improvements. We also plan to study the effectiveness of restricting phylogenetic-tree based methods to the subtree spanned by outliers. Finally, note that our method was developed for and tested on 16S rRNA data, and is not applicable as it stands to whole genome sequencing (WGS) datasets. However, the idea of using a two-step approach for taxonomy assignment in WGS datasets is an interesting avenue for research.

Chapter 3: ATLAS: embracing ambiguity in the taxonomic classification

This chapter contains material previously published in Embracing ambiguity in the taxonomic classification of microbiome sequence data [71], which was a joint work with Jacquelyn S. Meisel and Mihai Pop. NS and MP conceived the research project. NS designed and implemented the algorithm, with the help of JSM and MP. NS and JSM analyzed the data. NS, JSM, and MP wrote the manuscript.

3.1 Introduction

In Chapter 2, we presented a method to detect significant database hits, and use it as a taxonomic classification tool. Our results with both BLAST outlier detection and RDP classifier show that it is difficult to obtain species-level annotations. In this chapter, we address how we can improve the resolution of taxonomic classification results.

There are several limitations to database search based taxonomic classification approaches. First, it is often impossible to obtain confident genus- or even species-level classifications within samples due to the lack of discriminative power of the sequenced marker gene [72]. The 16S rRNA gene contains nine taxonomi-

cally discriminating hypervariable regions, however there is no single hypervariable region of the gene that can distinguish between all species. Additionally, reference databases are not always representative of a sample and are dominated by a small subset of easy to isolate organisms found at higher abundances [73]. Sequencing data in reference databases is largely biased towards pathogenic microbes and organisms commonly found in developed countries. The organisms found in many studies (e.g., in environmental communities or in developing countries) have no near neighbors in reference databases, making it difficult to assign to them accurate taxonomic labels.

Another problem with modern analysis of microbial communities is the relatively coarse-grained resolution obtained, which limits our ability to capture biologically relevant signals. This stems from the need to simplify computational workflows. Most commonly used bacterial taxonomies have been regularized to fit within a standard seven taxonomic levels. This problem is further compounded by errors and missing information in databases, as well as inherent ambiguities in the taxonomic assignment of some sequences. Current software tools frequently rely on "latest common ancestor" strategies to provide an annotation at the most general taxonomic level that encompasses all of the possible annotations of a sequence [50]. As a result, few methods ever make classifications below the genus level, and frequently sequences are only classified at the family, class, or even phylum level.

As the number and size of sequencing datasets continues to grow, taxonomic classification methods often make trade-offs between speed and accuracy. Different tools have been developed for taxonomic annotation, using either composition-based, sequence-similarity, or phylogenetic-placement methods [24, 25, 45, 74, 75].

Composition-based and sequence similarity-based approaches are fast and require less computational power, but only work well when the microorganisms in the sample have near neighbors in the database. On the other hand, phylogenetic-placement based methods statistically model the evolutionary processes that generate the query sequences and are computationally expensive, but allow classification even if only distant neighbors are found in databases.

Here we propose a novel strategy for taxonomic annotation that adequately captures and represents the complexity of the bacterial world, providing more specific and more interpretable characterizations of the composition of microbial communities, while also capturing the inherent ambiguity in the classification of sequences without near neighbors in public databases. This strategy builds upon our recent work on detecting significant outliers within database search results [41], allowing us to characterize, in a sample-specific manner, the extent of taxonomic ambiguity within the classification. This approach allows us to frequently make assignments at the species level, and even when such assignment is not possible, we are able to identify the few species within a genus that are the most likely origin of the fragment being analyzed. Such information is particularly relevant in clinical applications, allowing us to distinguish between the pathogenic and non-pathogenic members of the same genus even if the specific species cannot be uniquely identified.

Our method, called 'ATLAS-Ambiguous Taxonomy eLucidation by Apportionment of Sequences', is implemented in Python and released under the MIT license. We demonstrate that ATLAS yields similar results to phylogenetic methods, but with reduced computational requirements. We use ATLAS to re-examine

over two-thousand samples from the Human Microbiome Project (HMP) (The Human Microbiome Project Consortium, 2012) and interrogate almost one-thousand stool samples from the Global Enteric Multicenter Study (GEMS) of young children in low-income countries with moderate-to-severe diarrhea [49]. In these datasets, we identify partitions matching previously defined groupings of organisms, such as species within the *Bacillus* genus and the *Clostridia* class. We also demonstrate that the partitions identified by ATLAS increase the power of differential abundance analyses. Although our results specifically focus on data from 16S rRNA gene surveys, ATLAS can be used with any marker gene sequencing data to characterize the taxonomic composition of a microbial community and to determine microbiome associations with human and ecological health.

3.2 Materials and Methods

3.2.1 ATLAS algorithm overview

ATLAS groups sequences into biologically meaningful taxonomic partitions by querying them against a reference database and identifying and clustering significant database hits that capture the ambiguity in the assignment process. ATLAS has two phases (see Figure 3.1): (i) identifying significant database hits for query sequences and (ii) generating database partitions (clusters) that capture the ambiguity in the assignment process.

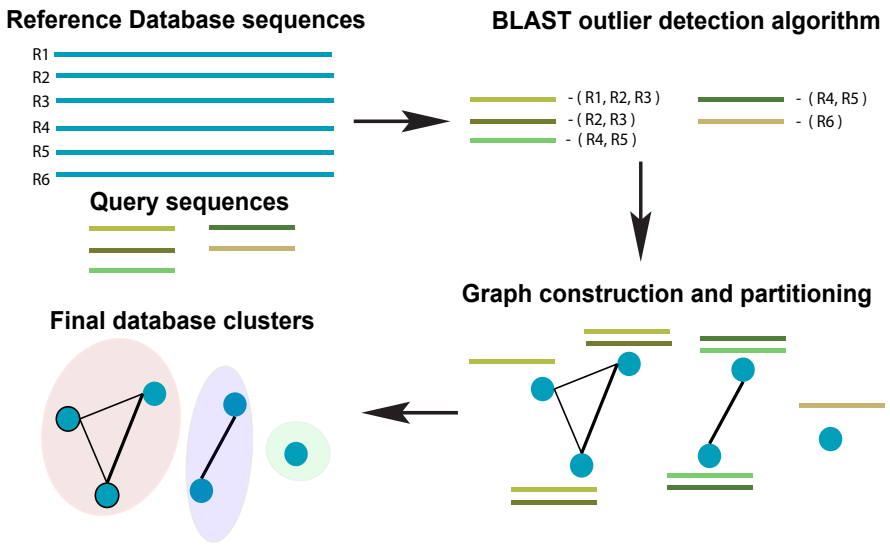

Figure 3.1: Schematic diagram of the ATLAS pipeline. ATLAS takes in query sequences from a marker gene and searches them against a reference database to identify outlier sequences. It then constructs a graph of database sequences and clusters those that are commonly identified together into partitions.

3.2.2 Aligning query sequences and identifying significant database hits

ATLAS uses BLAST [45] to align each sequence in an input set of uncharacterized query sequences, to sequences in a reference set (using parameters -outfmt "6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qseq sseq"). The previously published "BLAST outlier detection" algorithm is used to identify significant top BLAST hits for each query sequence [41]. We refer to these BLAST hits as outliers. In brief, the "BLAST outlier detection" algorithm constructs a multiple sequence alignment of the query sequence and the top BLAST hits from the BLAST-generated pairwise alignments. It then uses the Bayesian Integral Log Odds (BILD) score [55, 56] to determine whether the multiple alignment can be split into two groups that model the data better than a single group. This process identifies which BLAST hits are significantly associated with the query sequence, without resorting to ad hoc cut-offs on percent identity, bit score, and/or E-value.

3.2.3 Generating database partitions that capture the ambiguity in the assignment process

Ambiguity in the taxonomic assignment process occurs for two main reasons. First, the query sequence may not have any near-neighbors in the database, resulting in multiple equally-good hits (neighbors) (Figure 3.2). Second, the query sequence

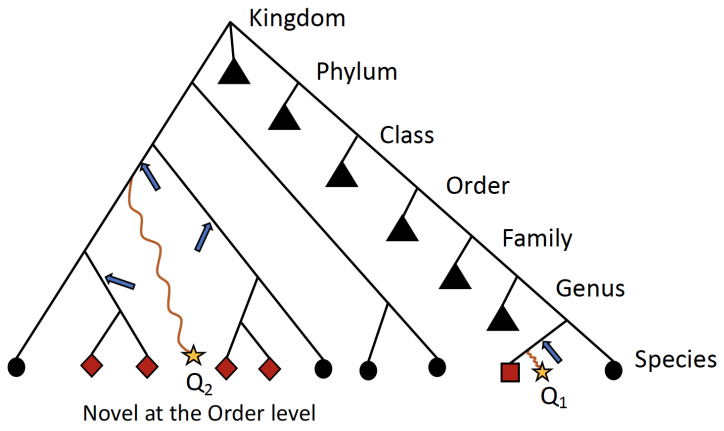

Figure 3.2: Schematic detailing when ATLAS will provide the greatest improvement to taxonomic annotation. Shown is a simple example of a phylogenetic tree with taxonomic information of reference sequences, where the leaves are actual sequences in the database. When a query sequence (yellow stars) has near neighbors in the reference, such as Q_1, most algorithms will be able to correctly classify the sequence. However, if a sequence, such as Q_2, does not have many near neighbors in the database, computationally expensive phylogenetic methods are required for accurate placement (blue arrows) and annotation. ATLAS captures groups (or partitions) of database sequences (red nodes) that are commonly confused during the annotation process and assigns them to the query sequence (square node for Q_1, and diamond nodes for Q_2,). Black triangles show collapsed portion of the tree. While this schematic is overly simplified and real phylogenies are much more complex, this is illustrating that ATLAS will provide additional information when query sequences do not have near neighbors in the database. This represents ideal cases, where 16S rRNA phylogeny and taxonomic annotations are congruent.

may align to a genomic region that is conserved across distantly related organisms. Our method characterizes this ambiguity in a sample-specific manner, identifying database sequences that are equivalent with respect to their similarity to the set of query sequences.

From all query sequences and their set of related database sequences (outlier set), we construct a confusion graph. The nodes in the graph represent sequences in the database, while the edges link nodes that are present together in the outlier set for at least one query sequence. The edges are weighted by the number of query sequences that share the same nodes (reference database sequences) within the outlier set. Tightly-knit sub-communities in the confusion graph indicate ambiguous database sequences that should be clustered together. To identify these sub-communities, we remove all the low-weight edges (below $mean - 2 * stddev$ of all edge weights) and identify strong communities in the network using the Louvain community detection algorithm, which optimizes the modularity of the network [76]. These sub-communities become the final database partitions (clusters).

3.2.4 Assigning query sequences to the partitions

A query sequence is assigned to a database partition if a certain percentage (user-defined, default 50%) of the database sequences in the outlier set belong to the partition. ATLAS does not classify the query sequence if no BLAST outliers can be detected, or the query sequence does not meet these thresholds. The goal of ATLAS is only to classify sequences when it has enough confidence in the taxonomic

assignment. Sequences that remain unclassified by ATLAS should be further examined with more sophisticated approaches, such as phylogenetic placement methods. For each query sequence, ATLAS provides a species list based on the reference database sequences included within the assigned partition. To provide a high-level summary of the data and simplify the comparison to other annotation methods, ATLAS also assigns to query sequences the MRCA of all sequences belonging to a partition. These partitions of database sequences attempt to capture the most accurate granularity of taxonomic assignment without relying solely on the main taxonomic levels.

3.2.5 Comparison to other taxonomic assignment methods

To benchmark ATLAS with other widely used taxonomic annotation methods, we downloaded TAXXI test and train datasets (sp_ten_16s_v35) from a recent study that benchmarked taxonomic methods for microbiome studies [77]. We compared ATLAS with RDP classifier [61], mothur [78], UCLUST [79], SortMeRNA [80], and the top BLAST hit. RDP classifier, mothur, and UCLUST were run with 80% confidence threshold. All methods except ATLAS were run via QIIME v. 1.9.1 [62], using the script assign_taxonomy.py. Metrics for method comparison were calculated as previously published [77].

We also compared ATLAS to the phylogenetic placement method, TIPP. We ran TIPP with the 16S rRNA reference package (rdp_bacteria.refpkg) provided by the authors (https://github.com/tandyw/tipp-reference/releases/download/v2.0.0/tipp.zip).

We used the alignment subset size of 100 and the placement subset size of $1,000$, and the default values for alignment and placement thresholds.

3.2.6 Analysis of samples from the Human Microbiome Project (HMP)

The OTU table and representative sequence FASTA files for the V1-V3 hypervariable region of the 16S rRNA gene sequenced as part of the Human Microbiome Project (The Human Microbiome Project Consortium, 2012) were downloaded from https://www.hmpdacc.org/HMQCP/. We used the 16S rRNA reference package from TIPP for ATLAS and ran it with default settings. The OTU table was filtered to retain OTUs with at least 20 reads and samples containing at least $1,000$ reads.

3.2.7 Analysis of samples from the GEMS study of diarrheal disease

A total of 992 samples were analyzed from a previously published study of diarrheal disease in children in low-income countries that sequenced the V1-V2 region of the 16S rRNA gene [49]. In this study, moderate-to-severe diarrhea cases were compared to age- and gender-matched healthy controls. Data was downloaded via Bioconductor, using the msd16s package. We used the 16S rRNA reference package from TIPP for ATLAS and ran it with default settings. The dataset was filtered to retain only OTUs with at least 20 reads total and found in at least 10% of case or 10% of control samples.

Significantly differentially abundant OTUs were identified between cases and controls using the R package metagenomeSeq [81], accounting for age in months,

country, and sample read counts as potential confounding factors. OTUs were also aggregated separately by genus and by partition. Significant findings were reported for features that had fold change or odds ratio exceeding 2 in either cases or controls and a significant statistical association ($P < 0.05$) after Benjamini-Hochberg correction for multiple testing.

3.2.8 Analysis of samples from Bangladeshi children with acute diarrhea

A total of 142 samples were analyzed from a previously published study of acute diarrhea in Bangladeshi children that sequenced the V3-V4 region of the 16S rRNA gene [82]. Fastq files were downloaded from BioProject SRP119744, using the SRA toolkit v. 2.8.2 and processed in QIIME v. 1.9.1. We used the 16S rRNA reference package from TIPP for ATLAS and ran it with default settings, identifying 77 partitions.

3.3 Results

3.3.1 ATLAS captures similar information as phylogenetic placement algorithms

We compared the taxonomic assignments generated by ATLAS for the HMP and GEMS datasets to the labels generated by TIPP [25]. Because TIPP relies on a phylogenetic approach for taxonomic annotation, it accounts for evolution-

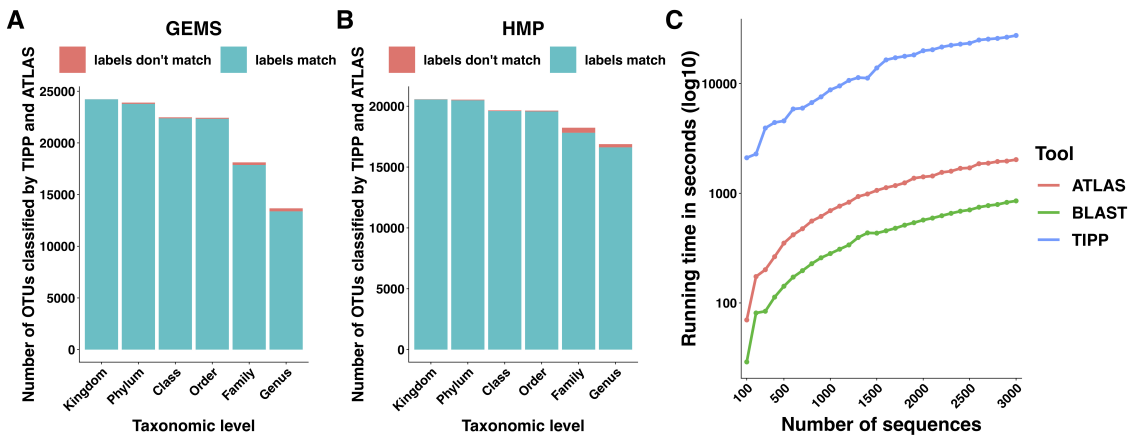

Figure 3.3: **ATLAS generates classifications similar to phylogenetic placement methods at an improved speed.** Taxonomic labels assigned by TIPP and ATLAS agree at all taxonomic levels for both (A) GEMS and (B) HMP datasets. (C) The ATLAS pipeline adds minimal post-processing time (in seconds) to standard BLAST analyses, but significantly outperforms TIPP.

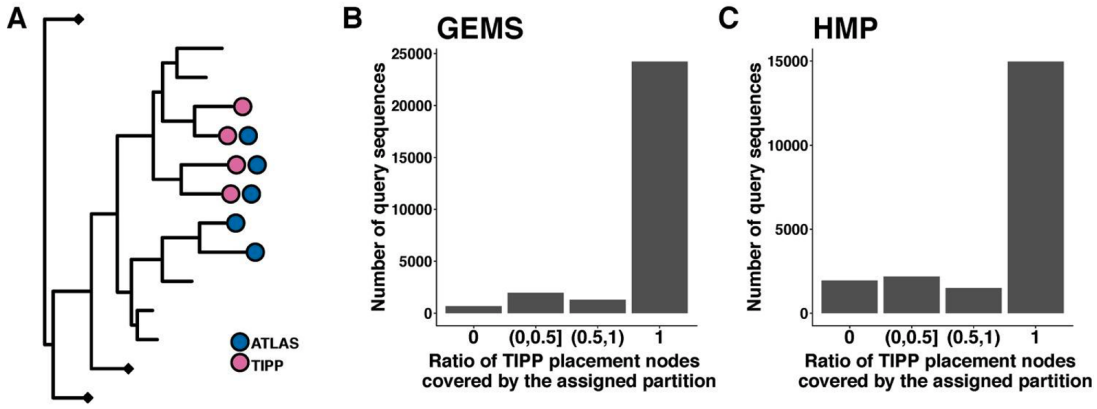

Figure 3.4: ATLAS partitions capture placement nodes identified by TIPP. (A) An example showing reference database sequences identified by TIPP placement and ATLAS' partition assignment for a query sequence. The ratio of TIPP placement nodes covered by the assigned partition for this query sequence is $3/4 = 0.75$. Partitions assigned by ATLAS contain a majority of reference database sequences identified by TIPP in the (B) GEMS and (C) HMP datasets.

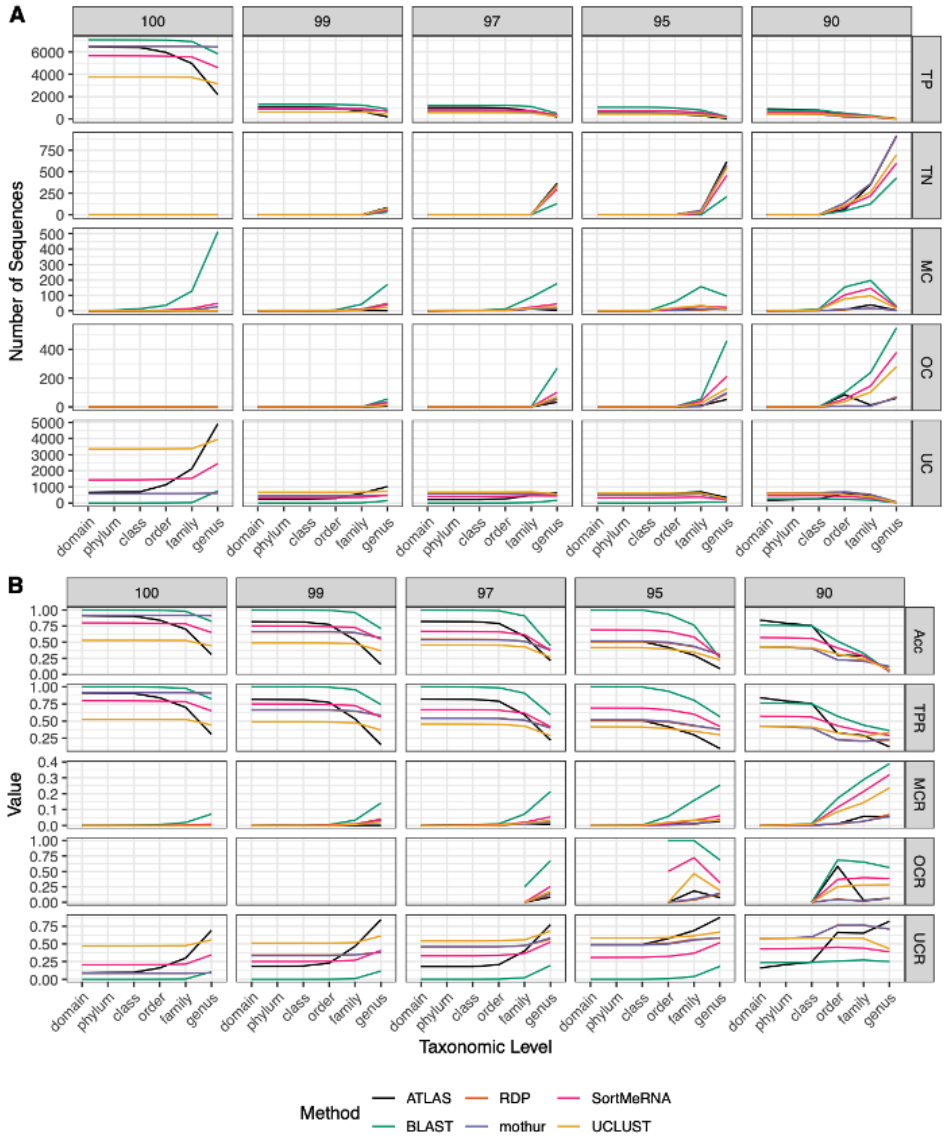

Figure 3.5: Comparison of ATLAS to other taxonomic annotation methods. Using a dataset where the ground truth is known, we characterized the performance of different classification methods by several metrics. Both the dataset (sp_ten_16s_v35) and metrics used are from Edgar, R. C. (2018) [77]. Sequences provided at the TAXXI website (https://drive5.com/taxxi/doc/index.html) were split into test and train dataset, such that for all test sequences, the most similar train sequence has given percent identity (horizontal facet for 100, 99, 97, 95). Reported here are (A) raw counts of true positives (TP), true negatives (TN), misclassified sequences (MC), over classified sequences (OC) and under classified sequences (UC). Also shown are (B) classification rates, including accuracy (Acc), true positive rate (TPR), misclassification rate (MCR), over classification rate (OCR), and under classification rate (UCR) for the same dataset.

		GEMS	HMP
A	Number of query sequences classified by TIPP at the species level	13050	10086
	Number of query sequences assigned to a partition that contained TIPP's species	12847	8999
B	Number of query sequences classified at the species level by ATLAS that match TIPP's labeling	29	128
	Number of query sequences classified at the species level by ATLAS that did not match TIPP's labeling	0	85
	Number of query sequences classified at species level by ATLAS but not by TIPP	18	36

Table 3.1: Comparison between our approach (ATLAS) and a phylogenetic method (TIPP) examining species level assignments. (A) For query sequences where ATLAS partitions do not have a species-level LCA, the assigned partition contains reference sequences that match TIPP's assigned species. (B) For query sequences where ATLAS partitions do have a species-level LCA, many of the assigned partitions match TIPP's classification.

ary divergence and, therefore, can more effectively analyze sequences without near neighbors in the database than non-phylogenetic methods. We assume here that the classifications provided by TIPP are most accurate because the ground-truth is not available for real datasets. The taxonomic assignments made by ATLAS and TIPP showed 97% and 98% agreement with TIPP assignments at the genus level for GEMS and HMP datasets, respectively (Figures 3.3A, B). Importantly, when TIPP could confidently assign a species level classification label to a query sequence, but ATLAS could not, the partition assigned by ATLAS for the majority of query sequences contained the species assigned by TIPP (Table 3.1). The algorithm used by TIPP identifies multiple putative placements of a sequence within the backbone tree representing the reference database. In the vast majority of cases, the partitions identified by ATLAS contained the database sequences selected by TIPP (Figure 3.4). Compared to TIPP, ATLAS had a lower run time and only added a small

overhead to the run time of BLAST (Figure 3.3C).

We also compared ATLAS to nonphylogenetic approaches (Figure 3.5) on the sp_ten_16s_v35 TAXXI benchmarking dataset where the ground truth is known [77]. Compared to other methods, ATLAS has similar or better overclassification and misclassification rates at all taxonomic levels. However, ATLAS often has a higher underclassification rate, particularly at lower taxonomic ranks. This behavior is intentional as ATLAS is meant to serve as a first-level analysis, followed by more sophisticated approaches (such as phylogenetic placement) for the sequences that cannot be confidently classified through sequence similarity searches.

3.3.2 Relationship between ATLAS partitions and standard taxonomic levels

	HMP		**GEMS**		
	OTU	**Partition**	**OTU**	**Genus**	**Partition**
Sequencing Technology	Illumina V1-V3		454 V1-V2		
Number of Samples Post Filtering	2711 180 gut, 1,553 oral, 719 skin, 259 vagina		992 508 Cases, 484 Controls		
Number of Features Pre-Filtering	43,140 OTUs	22,885 partitions (246 non-singleton partitions)	26,044 OTUs	172 genera	1,941 partitions (113 non-singleton partitions)
Number of Features Post-Filtering	36,560 OTUs	18,086 partitions (185 non-singleton partitions)	10,774 OTUs	149 genera	1,036 partitions (109 non-singleton partitions)

Table 3.2: Number of OTUs and partitions in the HMP and GEMS datasets pre- and post- filtering. Samples with > 1000 reads were retained for analysis. In the HMP data, features were retained if they had at least 20 total reads or were found in at least 5 samples. In the GEMS data, features were retained if they had at least 20 total reads or were found in at least 10% of case or control samples. Singleton partitions have a single OTU mapped to them.

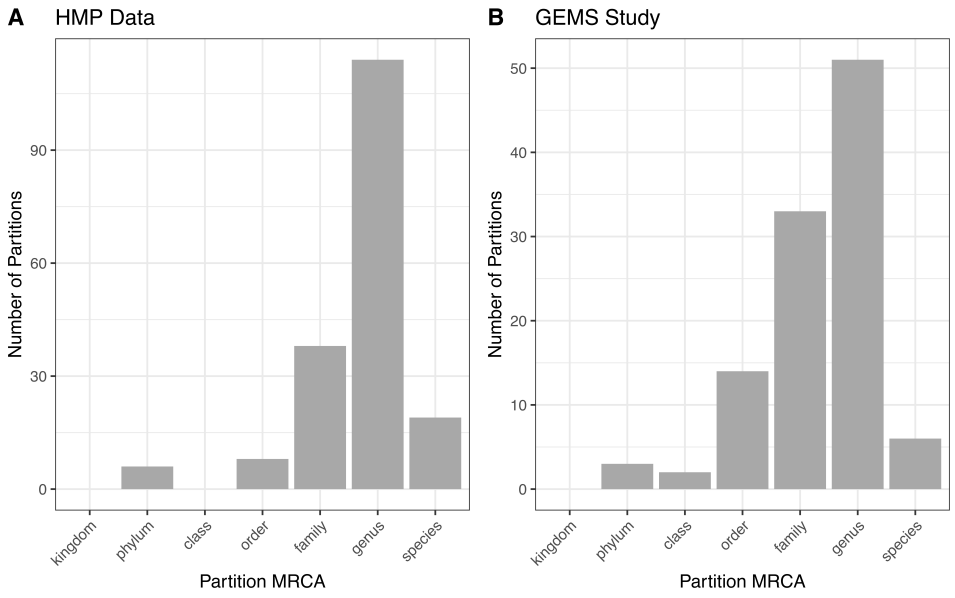

Figure 3.6: ATLAS partitions for HMP and GEMS data typically capture subgenera information. Most partitions have the most recent common ancestor at the genus level for both (A) HMP and (B) GEMS datasets

ATLAS grouped OTU representative sequences into 185 and 109 non-singleton partitions in the HMP and GEMS datasets, respectively (Table 3.2). A large number of these partitions each have an MRCA at the genus level, suggesting that they are capturing sub-genus information (Figure 3.6). Often, there is not enough information encoded in the short 16S rRNA gene sequence to offer species-level resolution. However, ATLAS is able to group similar species within a genus, providing resolution that is more specific than the genus level. For instance, in the HMP data, ATLAS identified seven partitions belonging to the genus *Bacillus* (Figure 3.7). Importantly, reference sequences in partition 156 capture members of the Bacillus cereus species group, including *B. cereus*, *B. thuringiensis*, *B. mycoides*, and *B. weihenstephanensis* [84]. These species have very high sequence similarity and have

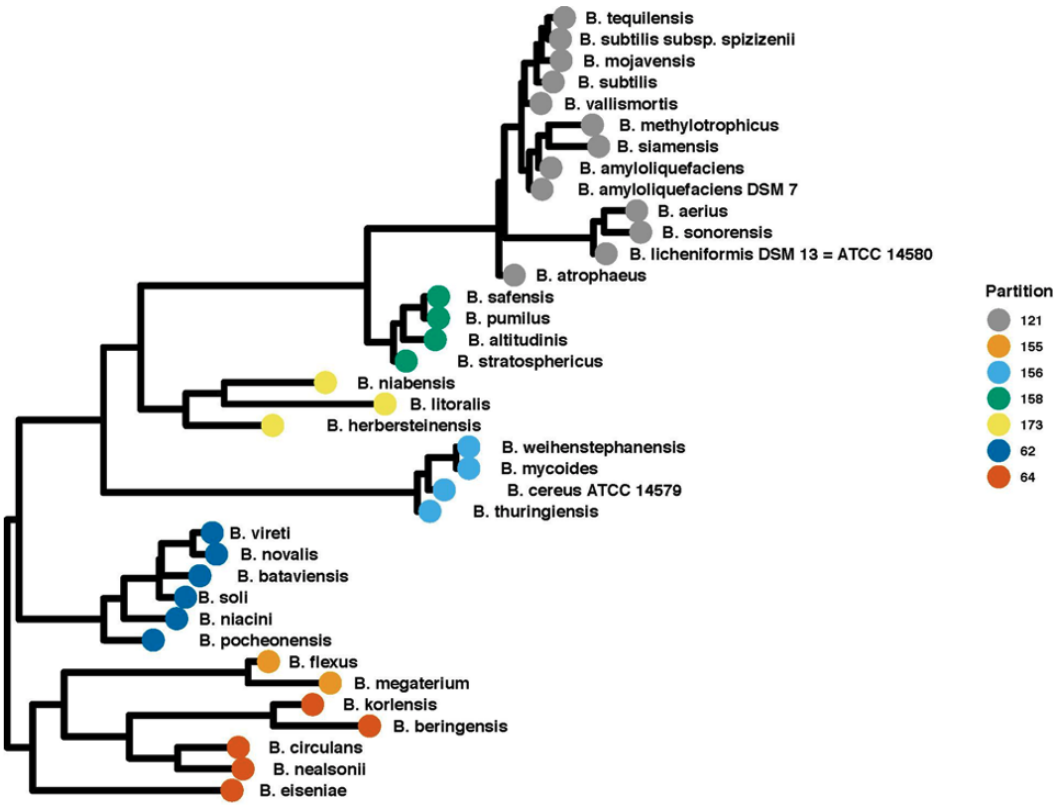

Figure 3.7: Reference database sequences in the sub-genera *Bacillus* partition in HMP samples. The TIPP reference tree was plotted using ggtree in R. Taxa included in partitions in the HMP dataset are indicated by dots, colored by partition. Branches not identified in our partitions were collapsed for visualization purposes.

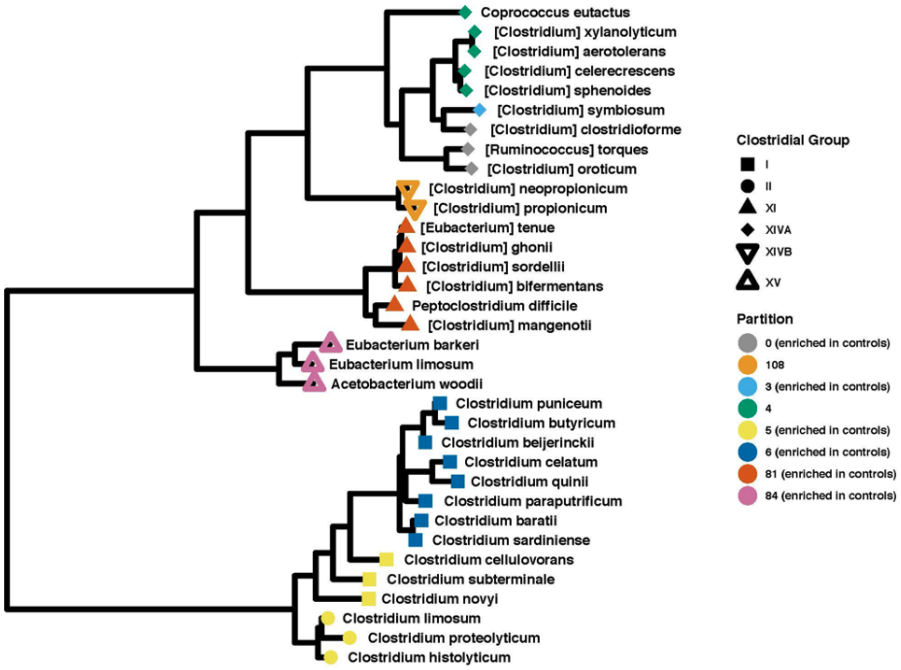

Figure 3.8: Reference database sequences in the Clostridial partition from the GEMS dataset. The TIPP reference tree was plotted using ggtree in R. Tree labels represent Clostridia grouped into partitions in the GEMS dataset, with the node colors representing the different partitions. The points on the tree nodes represent members of Clostridia groups identified in [83]. Branches not identified in our partitions were collapsed for visualization purposes, as were branches of Clostridia in the reference tree not grouped by Collins et al. 1994 [83].

been shown to play significant roles in human and environmental health [85]. ATLAS partition 121 corresponds to the *Bacillus subtilis* group, including species such as *B. subtilis*, *B. licheniformis*, and *B. amyloliquefaciens* [86]. Given the diverse function and pathogenic potential of species within this genus, the distinction of these two groups provides additional benefit to microbiome analyses.

It is important to note that ATLAS partitions are derived purely from sequence similarity; they do not take into consideration any taxonomic or phylogenetic information. Given our incomplete knowledge of microbial diversity and the inherent limitations of 16S rRNA sequences for taxonomic classification, these sub-genus partitions should be further examined and validated.

Other partitions with higher-level MRCA capture established phylogenetic groupings that span multiple genera. ATLAS was able to capture well-known phylogenetic groupings in the class Clostridia [83, 87]. In the GEMS data, ATLAS identified 15 partitions comprising sequences from the *Clostridia* class. Of particular note, partition 84 contains *Acetobacterium species* in *Clostridial group XV*, partition 81 contains members of *Clostridial group XI*, and *Clostridial group I* is represented in partitions 5 and 6 (Figure 3.8). Clostridial groups encompassed by partitions 0, 81, and 84 contained multiple genera, highlighting the utility of using partitions based on information from the sequences themselves rather than solely relying on modern taxonomic groupings. Interestingly, eight of these partitions were significantly differentially enriched in healthy control samples, supporting the role of *Clostridia* in the maintenance of gut homeostasis [88].

The percentage of query sequences assigned to partitions spanning multiple

genera was 8% for the HMP data and 39% for the GEMS data. Some of these higher-level partition groupings reflect limitations in the hypervariable region of the 16S rRNA gene sequenced. For instance, in both the HMP and GEMS data, ATLAS identified a single partition spanning the *Enterobacteriaceae* family. While it would be beneficial to distinguish between *Escherichia* and *Shigella* species in the GEMS dataset, the V1-V2 and V1-V3 hypervariable regions of the 16S rRNA marker gene are insufficient for discrimination [89].

3.3.3 ATLAS partitions improve the power of microbiome-disease association studies

	OTU	**Genus**	**Partition**
Number of Significant Features Increased in Case Samples	679 OTUs (415,257 sequences)	16 genera (892 OTUs, 342,960 sequences)	13 partitions and 71 non-partitioned OTUs (692 OTUs, 189,005 sequences)
Number of Significant Features Increased in Control Samples	1,112 OTUs (637,591 sequences)	22 genera (1,626 OTUs, 447,680 sequences)	17 partitions and 108 non-partitioned OTUs (4,917 OTUs, 1,300,544 sequences)
Number of Non-significant Features	8,983 OTUs (2,448,992 sequences)	105 genera (5,845 OTUs, 1,811,878 sequences)	77 partitions and 745 non-partitioned OTUs (5,165 OTUs, 2,012,291 sequences)

Table 3.3: Number of OTUs, genera, and ATLAS partitions that are statistically significantly different between moderate-to-severe diarrheal cases and healthy controls.

We explored whether ATLAS partitions could provide improved resolution over OTUs in differential abundance analyses. The original GEMS dataset contains 26,044 OTUs, many of which are not prevalent or abundant enough to provide

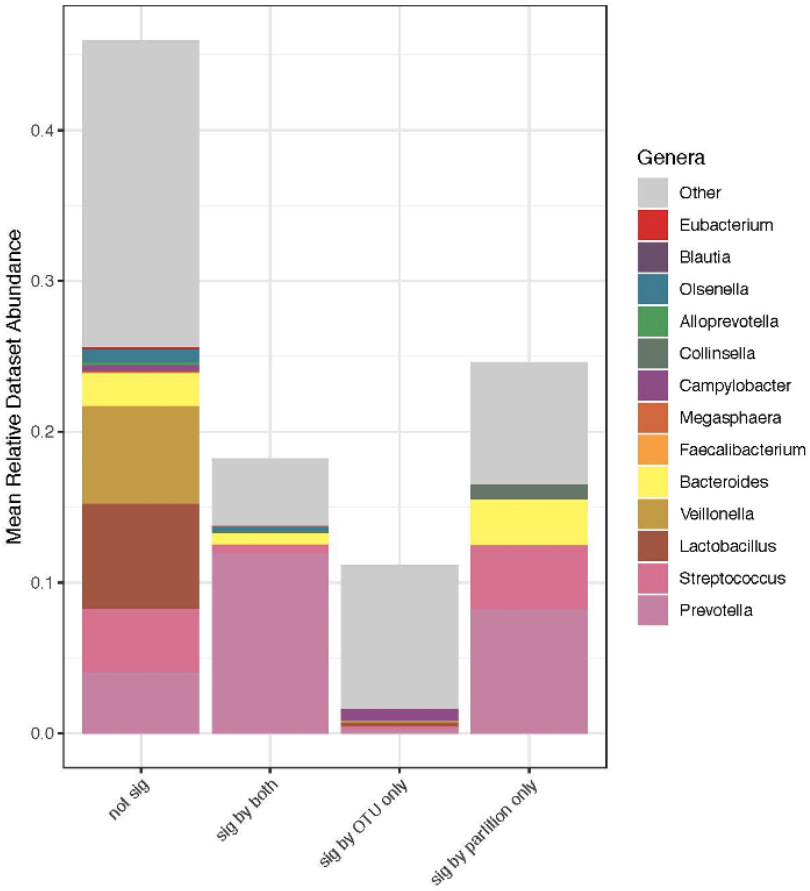

Figure 3.9: Differentially abundant OTUs in the GEMS dataset by genera. OTUs are grouped by whether they are not significant by either ATLAS partitions or individual OTUs, significant by both ATLAS partitions and individual OTUs, significant by individual OTUs only, or significant by ATLAS partitions only.

		OTUs	
		Not Significant	**Significant**
Partitions	**Not Significant**	4,557	608
	Significant	4,426	1,183

Table 3.4: Confusion matrix highlighting the number of shared/unshared statistically significant OTUs and ATLAS partitions.

statistical power for identifying associations between health and disease. Filtering OTUs and partitions according to their abundance and prevalence, we retained just those that contained at least 20 sequences and were found in at least 10% of the samples. Only 10,774 OTUs, comprising just 41% of the sequences in the dataset, were retained, whereas ATLAS partitions retained after filtering contained 25,135 total OTUs, comprising 97% of the sequences in the dataset (Table 3.2).

We identified statistically significantly different features between cases with diarrheal disease and healthy controls (Table 3.3). We performed this analysis separately on (i) OTUs, (ii) OTUs aggregated by genus-level assignments, and (iii) OTUs aggregated by ATLAS partitions. Compared to the OTU analysis, OTUs aggregated at the genus-level generally identified more significant OTUs, but fewer overall significant dataset sequences. This is potentially impacted by the fact that 2,411 OTUs and 899,322 sequences had no assignment at the genus level. OTUs aggregated by ATLAS partitions identified a greater number of significant OTUs and sequences enriched in the control samples. When looking at the 10,774 OTUs included in both the OTU-level and partition-based analyses, the majority agreed on differential abundance results (i.e., they were significant or not significant in both analyses) (Table 3.4). Forty-one percent were significant by the partition analysis, but not by OTU based methods. These OTUs were most likely lower abundant com-

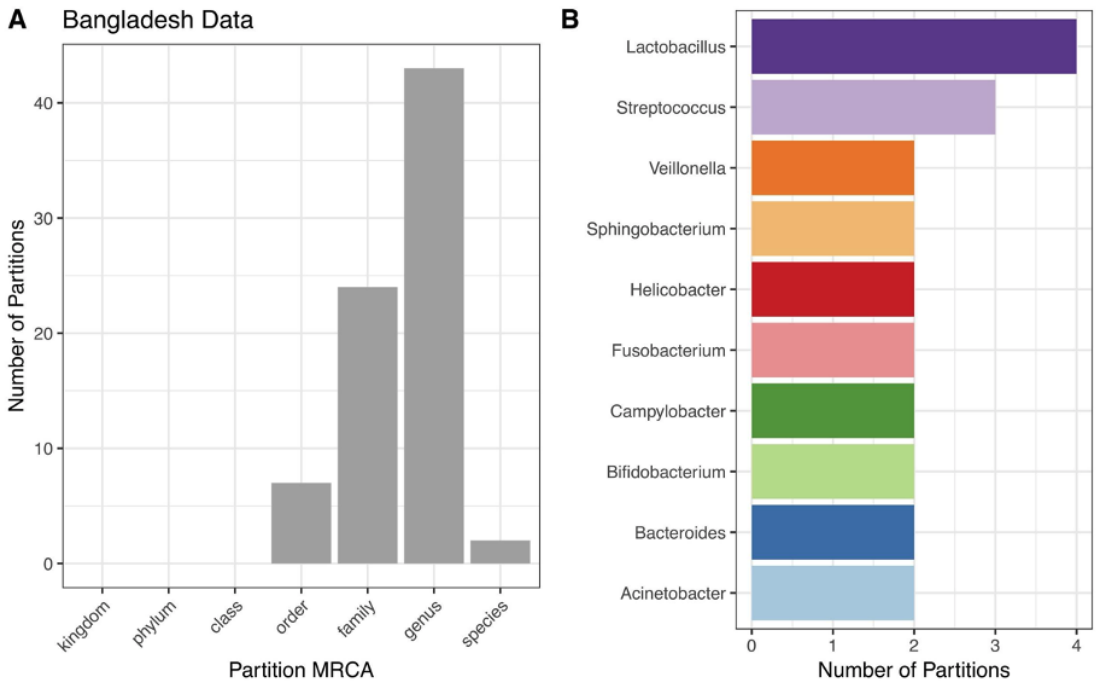

Figure 3.10: Partitions identified by ATLAS in acute diarrhea samples from Bangladesh. (A) Most partitions have the most recent common ancestor at the genus level for this dataset. (B) Number of partitions for the most common genera with sub-genus resolution in the Bangladesh dataset.

munity members that became significant as they were aggregated with similar, more abundant microbiota. The few remaining OTUs were significant at the OTU level but not in our partition-based analyses and generally belonged to low abundance genera (Figure 3.9).

We also applied ATLAS to a separate acute diarrhea dataset from children in Bangladesh [82], which used a different hypervariable region of the 16S rRNA gene, a different sequencing platform, and different downstream analyses. Within this dataset, we also identified sub-genus level partitions (Figure 3.10A). Many of the sub-genus level partitions in the Bangladesh dataset were in *Lactobacillus*, *Strep-*

tococcus, *Helicobacter*, and *Campylobacter*, genera which are commonly associated with diarrheal disease (Figure 3.10B).

3.4 Conclusion and Discussion

As DNA sequencing technologies become faster and cheaper, the number of microbiome studies are rapidly increasing. These studies are aimed at both developing a better understanding of the microbial communities inhabiting the world and at characterizing the association between microbiota and health. Accurate taxonomic assignment is a critical requirement for the interpretation of the data generated in such studies. Current approaches for taxonomic annotation fall at two extremes —computationally intensive phylogenetic inference methods that can accurately classify even sequences that are only distantly related to the reference database and fast approaches based on sequence alignment or k-mer analysis that are primarily effective in identifying already characterized sequences. Here, we have described an approach that bridges the two extremes. While it is based on sequence-similarity approach, ATLAS provides a similar level of accuracy as phylogenetic approaches while retaining computational efficiency.

ATLAS identifies the ambiguity in the classification of sequences in a sample-specific manner, thereby obviating the need for removing redundancy from the reference database (a computationally expensive process) and ensuring that the method effectively adapts to the specific parameters of the experiment (e.g., choice of hypervariable region in the 16S rRNA gene). While ATLAS is intended to replace

commonly-used "most recent common ancestor" (MRCA) approaches that are unnecessarily conservative, it can also improve on such techniques. The ATLAS partitions are constructed after examining all the query sequences, and after removing spurious connections between database sequences, thereby eliminating many of the errors that can reduce the taxonomic resolution of the MRCA approach.

We have shown that ATLAS is effective in analyzing real microbiome datasets, where it is able to automatically discover taxonomic groupings that are relevant to the interpretation of the data but that do not match predefined taxonomic levels. Examples include subdivisions of the *Bacillus* genus and *Clostridial* class homology groups. Our paper describes results generated from 16S rRNA gene sequencing data, however, the approach is applicable to any other marker gene dataset. Because ATLAS relies on marker gene data, it can only provide a level of resolution matching that of the maker gene itself.

Our analysis of the HMP and GEMS datasets reveals a difference in the level of ambiguity identified by ATLAS; our method was able to better resolve the taxonomy of sequences from the HMP project than that of sequences from the GEMS dataset. This finding is likely due to the relationship between the sequences from the two studies and the data found in the reference database. The GEMS study contains data from children from sub-Saharan Africa and Southeast Asia, sequences that are only distantly related to the reference sequences primarily characterized within Western populations. Our findings support the idea that the choice of database plays a huge role in classification accuracy [90]. To ensure an accurate taxonomic annotation, a custom environment-specific database is desirable, and the accuracy

of the database sequences and their annotation must be ensured. Studies must also carefully consider and document the choice of database.

The GEMS dataset was generated several years ago using 454 sequencing technology with high-insertion-deletion error rates. This can provide useful information for future applications to current long read sequencing datasets, which also have higher insertion-deletion error rates compared to short-read technologies. Despite differences between the GEMS and Bangladesh datasets, ATLAS identified subgenus partitions in important taxa previously associated with diarrhea. This improved resolution will provide greater insight into potentially harmful or beneficial organisms.

An opportunity for future research is the integration of the approach embodied in ATLAS with phylogenetic algorithms. Phylogenetic approaches can use the partitions identified by ATLAS to prune the reference tree before attempting to place query sequences on the tree, resulting in higher accuracy with lower computational overhead. In the future, we also plan to identify and investigate cases where ATLAS assignments and phylogenetic classifications disagree in order to identify opportunities for improvements to either alignment-based or phylogenetic approaches. As the wealth of microbiome data increases, greater emphasis is being placed on more accurate taxonomic annotations that currently cannot be obtained using fast, sequence similarity-based methods. ATLAS is the first step in this direction.

Chapter 4: TIPP2: metagenomic taxonomic profiling using phylogenetic markers

This chapter contains material previously published in TIPP2: metagenomic taxonomic profiling using phylogenetic markers [91], which was joint work with Erin K. Molloy, Mihai Pop, and Tandy Warnow. NS developed the database for the TIPP2 package under the guidance of MP and TW. NS and EKM changed TIPP software and maintain it. EKM helped in initial exploration experiments. NS designed and performed all experiments with the help of EKM, MP, and TW. NS, EKM, MP, and TW wrote the paper.

4.1 Introduction

In Chapters 2 and 3, we explored taxonomic classification problems in metagenomics. In this chapter, we explore a related problem of taxonomic abundance profiling. The goal is to estimate the relative proportions of different species in the sample.

Several different strategies have been introduced for estimating the relative abundance of species in the sample from metagenomic data [20, 21, 22]. One approach is to classify reads by searching against known sequences database, and then

normalize the read assignments by genome sizes to provide an estimate of relative abundance of each species [20, 23]. An alternative strategy involves the use of marker genes. Marker genes are generally clade-specific or universal, unique and single-copy [21, 22, 24, 25, 26, 27]. Because marker genes are unique and single-copy, the resulting read coverages can be readily used to estimate species abundance without normalization by genome size or copy number.

However, methods that just rely on aligning against sequences in reference databases (pairwise alignment) are likely to miss species in the sample that are not well-represented in the database, and fail to account for the abundance of these species. Good reference collections are missing for many understudied environments, such as soil or ocean [92, 93, 94, 95]. Phylogenetic approaches are designed to be able to detect distant homology, enabling the characterization of previously unseen organisms. However, with the ever-growing number of sequences in databases, scaling up phylogenetic approaches creates new challenges.

In 2014, we developed TIPP [25], a method that uses phylogenetic placement and a database of marker genes for abundance profiling, along with a method that uses an ensemble of Hidden Markov models (eHMMs; [96]) for improving classification accuracy. TIPP performed especially well in reads with high insertion and deletion (indel) sequencing errors, and in reads from unrepresented genomes (novel genomes). Here, we present TIPP2, an updated version of TIPP (henceforth referred to as TIPP1). Our experiments show that TIPP2 substantially improves on TIPP1 with respect to accuracy in abundance profiling. A comparison between TIPP2 and the leading current methods for abundance profiling reveals the following trends:

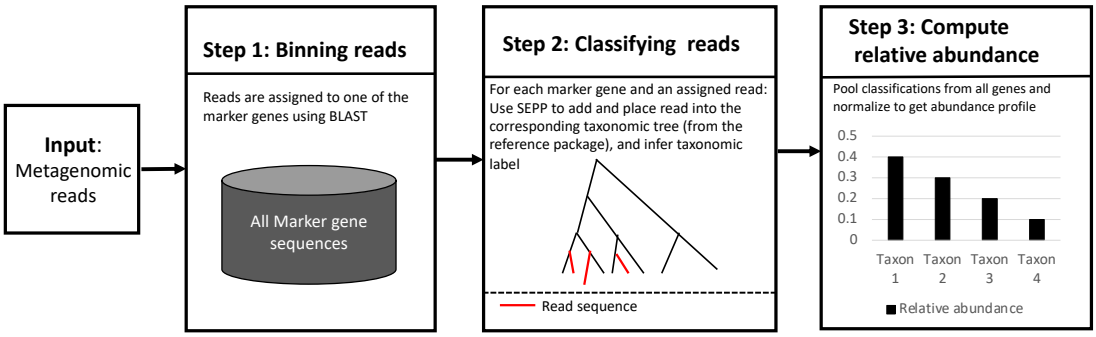

Figure 4.1: **Schematic of TIPP1 and TIPP2 pipelines.** TIPP1 has a database of 30 marker genes with ∼1300 sequences each, and TIPP2 has a database of 39 marker genes with ∼4300 sequences each; each also has a reference package of taxonomies and sequence alignments for each marker gene in their database. In Step 1, TIPP1 and TIPP2 assign metagenomic reads to marker genes using BLAST (and so some reads are not assigned to any marker gene, and are discarded), but differ in the specific technique used with BLAST (see text).

when the reads are drawn from genomes that are well-represented in the reference databases (i.e. the "known genomes" condition), then TIPP2 matches the accuracy of the leading alternative methods, while when the reads are drawn from genomes that are not present in the reference database (i.e. the "novel genomes" condition), then TIPP2 dominates the other methods in terms of accuracy. Hence, TIPP2 is a new method for abundance profiling that provides superior accuracy.

4.2 Approach

TIPP2 builds upon TIPP1, but uses a larger set of marker genes for the reference database and has slightly modified algorithmic steps (Figure 4.1). Here, we explain briefly the TIPP1 algorithm and highlight the novel contributions in TIPP2.

4.2.1 TIPP1 algorithm overview

As a preprocessing step for each marker gene, TIPP1 generates a multiple sequence alignment and a maximum likelihood tree constrained by the NCBI taxonomy; the collection of alignment-tree pairs for all marker genes is collectively referred to as the 'reference package'. TIPP1 has three key steps when analyzing reads from a metagenomic sample.

1. **Binning reads:** First, all reads are assigned to one of the marker gene sequences using BLAST [45]. Specifically, each read is assigned to the marker gene to which it has the best alignment. Reads that do not have a good alignment to any of the marker genes are excluded from analysis.

2. **Classifying reads:** Each read is added to the multiple sequence alignment of the marker gene and then placed into the corresponding taxonomic tree. This alignment/placement step is performed with SEPP [97], a method that uses an ensemble of hidden Markov models (eHMMs) designed to provide high taxonomic placement accuracy for large alignment/tree pairs. Importantly, each read is placed at the lowest internal node N so that placement support values summed across the placements on all edges of the subtree rooted at that node N surpasses the user-specified threshold (0.95 by default).

   Because we are working with a taxonomic tree, placing the read onto any internal branch gives a taxonomic label to the read. Note that placement on a terminal branch yields a species or strain-level classification, but placements

at internal branches may yield only higher-level classifications. For example, a read placed on an internal branch may only be classified at the family level and higher, in which case the read is unclassified at the genus and species levels.

3. **Computing relative abundance:** Once all reads are taxonomically classified, TIPP1 pools together the information from all marker genes, and computes relative abundances for each taxon. The relative abundance of a taxon is computed as the total number of reads classified within the taxon divided by the total number of reads classified by TIPP1.

4.2.2 Improvements to the reference package

In TIPP1, we used a set of bacterial marker genes that had been previously used in MetaPhyler [24] for the database: this contains 30 marker genes, with 1300 sequences per gene. In TIPP2, we changed to a set of 39 bacterial and archaeal marker genes, which were previously used for prokaryotic species delineation by [98]. The RpoB gene was not included in the TIPP2 database, as it had lower precision than all the other genes (discussed later). We downloaded approximately 170,000 bacterial and archaeal genomes from the NCBI RefSeq database. We then performed a sequence of analyses that were designed to identify those genomes that had a large number of marker genes retrievable using fetchMG tool (http://vm-lux.embl.de/mende/fetchMG/about.html). Finally, from that reduced set of genomes, we selected either one or two genomes per genus. This resulted in a

Marker	COG ID	Number of sequences	Median gene length	Max p-distance	Average p-distance
ArgS	COG0018	4411	11484	0.78	0.56
CysS	COG0215	4358	9252	0.72	0.5
Ffh	COG0541	4493	8367	0.69	0.46
FtsY	COG0552	4507	31581	0.68	0.48
Gtp1	COG0012	4268	2367	0.7	0.43
HisS	COG0124	4082	7680	0.77	0.55
LeuS	COG0495	4497	24744	0.74	0.48
PheS	COG0016	4321	5289	0.71	0.47
RplA	COG0081	4315	1275	0.72	0.44
RplB	COG0090	4343	1500	0.65	0.42
RplC	COG0087	4110	3120	0.71	0.47
RplD	COG0088	4269	2286	0.74	0.52
RplE	COG0094	4185	1155	0.74	0.42
RplF	COG0097	4137	1053	0.74	0.47
RplK	COG0080	4265	1041	0.73	0.4
RplM	COG0102	4322	1101	0.81	0.45
RplN	COG0093	4345	534	0.64	0.37
RplO	COG0200	4306	1743	0.79	0.5
RplP	COG0197	4309	717	0.66	0.41
RplR	COG0256	4329	1050	0.75	0.48
RplV	COG0091	4525	3471	0.78	0.5
RpoA	COG0202	4308	3867	0.74	0.48
RpoB	COG0085	4416	35103	0.78	0.42
RpsB	COG0052	4089	4941	0.74	0.46
RpsC	COG0092	4325	5550	0.73	0.45
RpsD	COG0522	4360	1404	0.78	0.48
RpsE	COG0098	4328	2907	0.73	0.46
RpsG	COG0049	4333	903	0.69	0.42
RpsH	COG0096	4516	651	0.71	0.47
RpsI	COG0103	4170	1737	0.74	0.46
RpsK	COG0100	4497	1083	0.69	0.41
RpsL	COG0048	4496	843	0.69	0.36
RpsM	COG0099	4343	591	0.71	0.41
RpsO	COG0184	4354	606	0.71	0.44
RpsQ	COG0186	4505	1692	0.74	0.48
RpsS	COG0185	4338	606	0.64	0.39
SecY	COG0201	4341	6153	0.73	0.49
SerS	COG0172	4371	4686	0.73	0.49
TsaD	COG0533	4364	8376	0.71	0.5
ValS	COG0525	4425	27369	0.75	0.49

Table 4.1: Statistics for the 40 marker genes.

set of approximately 4300 genomes per marker gene.

To identify whether a marker gene was retrievable in a given genome, we performed the following sequence of analyses. For each marker gene, we extracted sequences from each of the 170,000 genomes using the HMMs in the fetchMG tool, and computed the median length of these sequences. We noticed that many gene sequences recruited by the fetchMG HMMs had lengths that were far from the median, thus suggesting that these were false positives (i.e., not truly homologous). Subsequently, we used a length-based filtering approach to remove from consideration gene sequences that were far from the median length for that gene (median $\pm$

3 × standard deviation).

For each marker gene, a multiple sequence alignment was built on the full length sequences using PASTA [99]. We used RAxML [100] to generate a phylogenetic tree constrained to the NCBI taxonomy for each marker gene. Table 4.1 provides the list of marker genes and corresponding statistics of the reference multiple sequence alignments. The number of sequences per marker gene ranges from 4082 to 4525, with an average of 4339 sequences per marker gene.

4.2.3 Improvements to the TIPP1 algorithm

In TIPP2, we changed step 1 (i.e., the way reads are assigned to marker genes). TIPP1 used the top BLAST hit [45] to identify the marker gene for each read. Due to the way in which BLAST handles gapped alignments, the top hit may not always represent the correct marker gene [46, 101]. Therefore, TIPP2 requires the BLAST hit to cover at least a certain length (user-defined parameter, default 50bp), and it selects the marker gene based on the hit that has maximum-length alignment to the read. TIPP2 uses BLAST instead of HMMER [96] to determine the orientation of the reads with respect to the gene sequences in the reference database. When the reads align across the end of a marker gene sequence, TIPP2 trims the read to just the aligned region, a feature that also enables the use of the tool on long read data or assembled contigs.

4.3 Experimental study design

4.3.1 Overview

We set out to evaluate the performance of the improvements made in TIPP2, both with respect to the performance of TIPP1 and with respect to commonly used taxonomic profiling tools. We also evaluated the impact of the set of marker genes used as part of the reference package for TIPP2. We performed three experiments:

- Experiment 1: Testing whether abundances can be accurately estimated using a small subset of the marker genes, enabling a fast variant of TIPP2.
- Experiment 2: Comparing TIPP2 to TIPP1 on datasets with both known and novel genomes.
- Experiment 3: Comparing TIPP2, TIPP2-fast, and other existing methods for abundance profiling.

In Experiment 1, we worked with training datasets where the query sequences are from 'novel' genomes (i.e., species that are not present in TIPP2 databases). We used the training datasets only in the design phase, to select the subset of marker genes used for TIPP2-fast. In Experiments 2 and 3, we evaluated the performance of TIPP2 and TIPP2-fast using test datasets which contain both known and novel genomes, and which are simulated with different sequencing technologies and read lengths.

4.3.2 Metagenomic abundance profiling methods used for benchmarking

We compared the performance of TIPP2 with TIPP1, MetaPhlAn2 [27], mOTUsv2 [21], Bracken [20], and BLAST [45]. Except for BLAST, all of these methods are specifically designed for estimating taxonomic relative abundances in the metagenomic data, and except for Bracken, all these methods are marker-based. MetaPhlAn2 uses ∼1 million clade-specific markers to detect and estimate the relative abundance of organisms. TIPP2 uses a set of 39 marker genes from [98] and mOTUsv2 uses a subset of 10 marker genes described in the same study. Bracken [20] is an extension of Kraken [75] that reassigns the unclassified portion of reads based on probabilistic estimates of the true abundance profile from the Kraken output. We used BLAST as a baseline to compare TIPP2 performance, assigning each query sequence the taxonomic label of the hit selected during the read binning phase of TIPP2 (therefore the BLAST analysis utilizes the same marker gene database as TIPP2). We then calculate the abundance profile as relative abundance of reads from each taxon.

4.3.3 Simulated metagenomic datasets

We simulated metagenomic datasets with different characteristics, such as the average read length (range 100–250 bp), the number of genomes in the sample, sequencing technology profile, and whether the datasets contain known or novel

Dataset	Test/Train dataset	Known/Novel genomes	Number of genomes	Number of reads	Sequencing technology	Median read length
Known-51 454 Roche	Test	Known	51	2,863,285	454 Roche	229
Known-51 Illumina 100bp	Test	Known	51	8,130,295	Illumina	100
Known-51 Illumina 250bp	Test	Known	51	3,252,030	Illumina	250
Novel-100 454 Roche	Test	Novel	100	48,196,018	454 Roche	173
Novel-100 Illumina 150bp	Test	Novel	100	55,730,636	Illumina	150
Novel-100 Illumina 250bp	Test	Novel	100	33,397,306	Illumina	250
Novel-33 454 Roche	Train	Novel	33	17,314,764	454 Roche	173
Novel-33 Illumina 150bp	Train	Novel	33	20,052,474	Illumina	150
Novel-33 Illumina 250 bp	Train	Novel	33	12,031,210	Illumina	250

Table 4.2: Properties of simulated datasets.

genomes. A dataset is called 'novel' if none of the genomes in the dataset are present in the reference databases of any of the methods being tested. A dataset is called 'known' if it contains genomes that were used in reference databases of at least one of the methods tested. We simulated three groups of datasets. The first group of datasets contained reads from known genomes. We used the ART sequence simulator [102] to generate three datasets from a mixture of 51 genomes. We simulated one dataset with the 454 sequencing profile and the other two with the Illumina profile and different read lengths (100bp and 250bp). Table 4.2 provides the overview of all datasets, and a more detailed description of how these datasets were constructed can be found in supplementary section S4.

To find genomes that are 'novel' to the methods studied, we downloaded all complete genomes from the NCBI GenBank database and identified species that are not represented in the reference database of any method. There were 133 such genomes. We selected a set of 100 genomes and created three metagenomic datasets using the ART simulator [102]. We simulated one dataset with the 454 sequencing profile and the other two with Illumina profile and two different read lengths (150bp, and 250bp). We used the set of remaining 33 novel genomes to create a third group

of datasets, which were used just for training and optimizing the TIPP2 pipeline.

4.3.4 Accuracy evaluation

Because we know the true abundance profile for each dataset, we can compute the error in abundance estimation. To evaluate error in the estimated abundance profile, we compute the Hellinger distance [103] between the estimated abundance profile and the true abundance profile. Briefly, the Hellinger distance is

$$H_l = \frac{\sqrt{\sum_{x \epsilon C_l} (\sqrt{T_x} - \sqrt{E_x})^2}}{\sqrt{2}},$$

where C_l is the set of clades in the true and estimated profiles for taxonomic level l, T_x is the abundance of the clade x in the true profile, and E_x is the abundance in the estimated profile. H_l ranges from 0 (if there is a perfect match between the profiles) and 1 (if the two profiles are fully disjoint). Note that the reads that are unclassified at a given taxonomic level are not included in the H_l calculation for that level.

4.3.5 Running time study

We generated five replicate datasets with 2,000,000 sequences from novel genomes with two sequencing technologies—454 Roche and Illumina 250bp. Each method was run on a Blue Waters machine with 16 CPUs and 32 GB of total memory. We report the average wall clock running time for all methods.

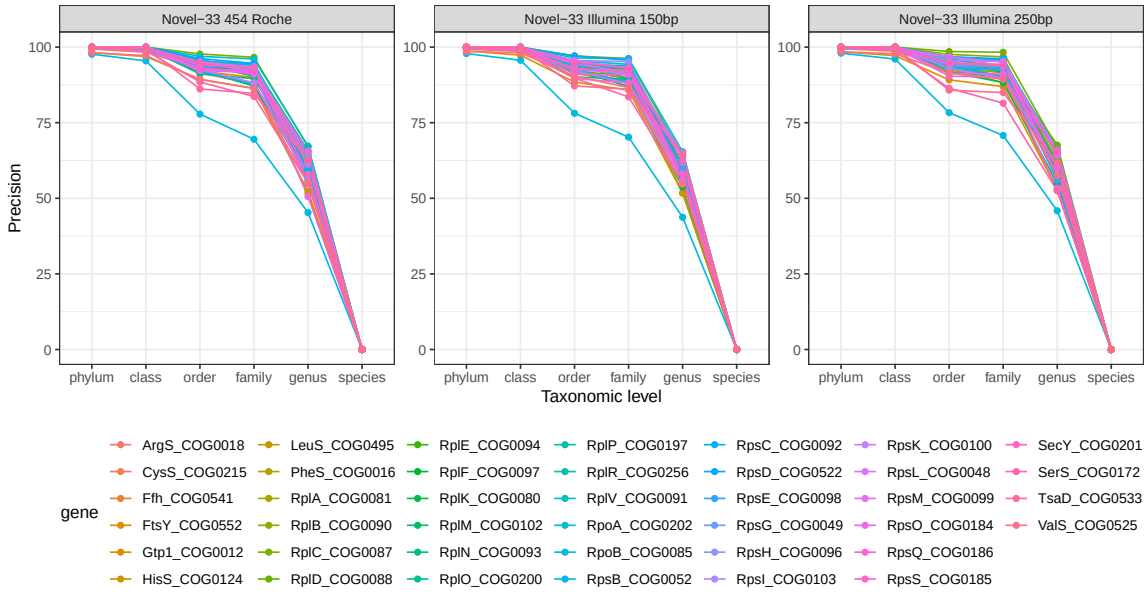

Figure 4.2: Experiment 1: Precision of reads classified by each marker gene on the training datasets (Novel-33 datasets). For a given taxonomic level and a marker gene, the precision (also called positive predictive value) is calculated as the ratio of the number of reads correctly classified (True positives) to the total number of reads classified (True positives + True negatives).

4.4 Results

4.4.1 Experiment 1: Testing whether fewer marker genes can correctly estimate abundances

We wanted to test whether we need the complete set of marker genes in the TIPP2 pipeline. Using training datasets (Novel-33 454 Roche, Novel-33 Illumina 100bp, Novel-33 Illumina 250bp), we explored the accuracy of abundance estimation for each gene separately.

We computed precision on a per-gene basis using the accuracy of taxonomic

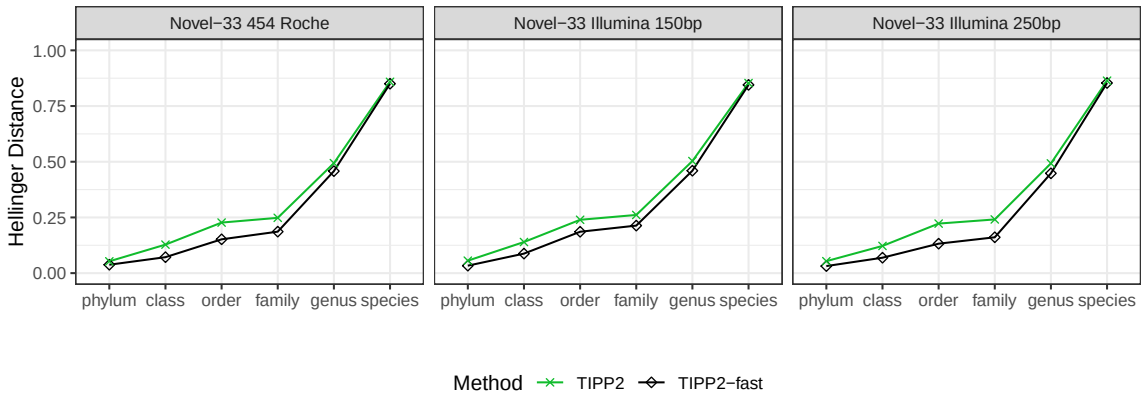

Figure 4.3: Experiment 1: Error in abundance profile estimates on training datasets. The training datasets are simulated metagenomic datasets from novel genomes with different sequencing technology and read lengths. We show Hellinger distance for TIPP2 using all marker genes, and TIPP2-fast, which uses just three genes (RpsL, RpsK, and RplO).

identification of reads using TIPP2. For a given taxonomic level, the precision (also called positive predictive value) is calculated as the ratio of the number of reads correctly classified (True positives) to the total number of reads classified (True positives + True negatives). We found that the RpoB (COG0085) gene consistently had lower precision than all other genes (see Figure 4.2). Hence, we removed the RpoB gene from the TIPP2 reference package. Moreover, we found that TIPP2, when run with just three genes—RpsL (COG0048), RpsK (COG0100), and RplO (COG0200)—provided better abundance estimates compared to the version that used the full set of genes (see Figure 4.3). These three genes were the top three high precision genes when the genes were ranked based on average precision across taxonomic ranks and datasets. We call this version TIPP2-fast because using fewer genes for classification reduces the running time of the pipeline (see Table 4.3). We

found that TIPP2-fast had lower error than TIPP2 at all taxonomic levels above the species level, and matched TIPP2's performance at the species level, showing that using fewer marker genes does not negatively affect the overall accuracy of the pipeline, and can actually improve accuracy.

4.4.2 Experiment 2: Comparing TIPP2 to TIPP1

In Experiment 2, we compared TIPP2 and TIPP2-fast with TIPP1. First, we compared TIPP1 and TIPP2 with the same reference package; these results (see Figures 4.4 and 4.5) show that the changes to the algorithmic design have minimal impact. Hence, our subsequent comparisons are between TIPP2 (and TIPP2-fast) using the updated reference package and TIPP1 using the 2014 reference package.

Figure 4.6 shows the average Hellinger distance for TIPP2-fast, TIPP2, and TIPP1 for known and novel genomes datasets. For both known and novel genomes, TIPP2 improves on TIPP1, but the improvement is much larger for the known genome condition, where TIPP2 has a consistent and large improvement over TIPP1. TIPP2 improves on TIPP2-fast for the known genome condition at all taxonomic levels, but just matches TIPP2-fast for the novel genome condition. This suggests that when samples contain well-characterized genomes, using a comprehensive set of genes (as contained in TIPP2) can provide better abundance estimates, and that the advantage in using the larger reference package is reduced for novel genomes.

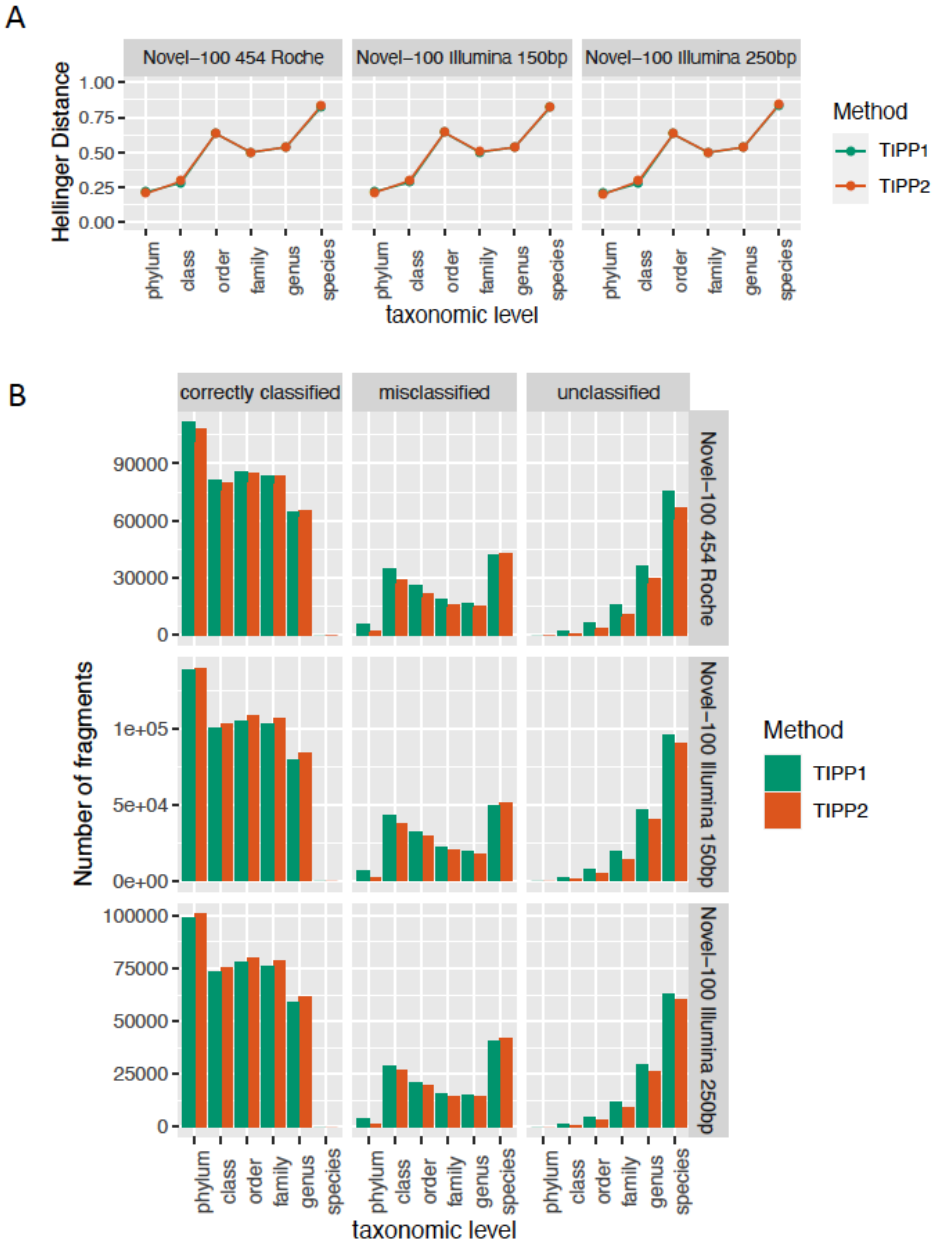

Figure 4.4: Experiment 2: Comparing TIPP2 and TIPP1, both using the reference package from 2014 (TIPP1 reference package) on the novel genome datasets. (A) The Hellinger distances to the true abundance profile (i.e., error) for TIPP2 and TIPP1 on the novel genome datasets. (B) Number of fragments (reads) correctly classified, misclassified, and unclassified by TIPP2 and TIPP1 for the novel genome datasets.

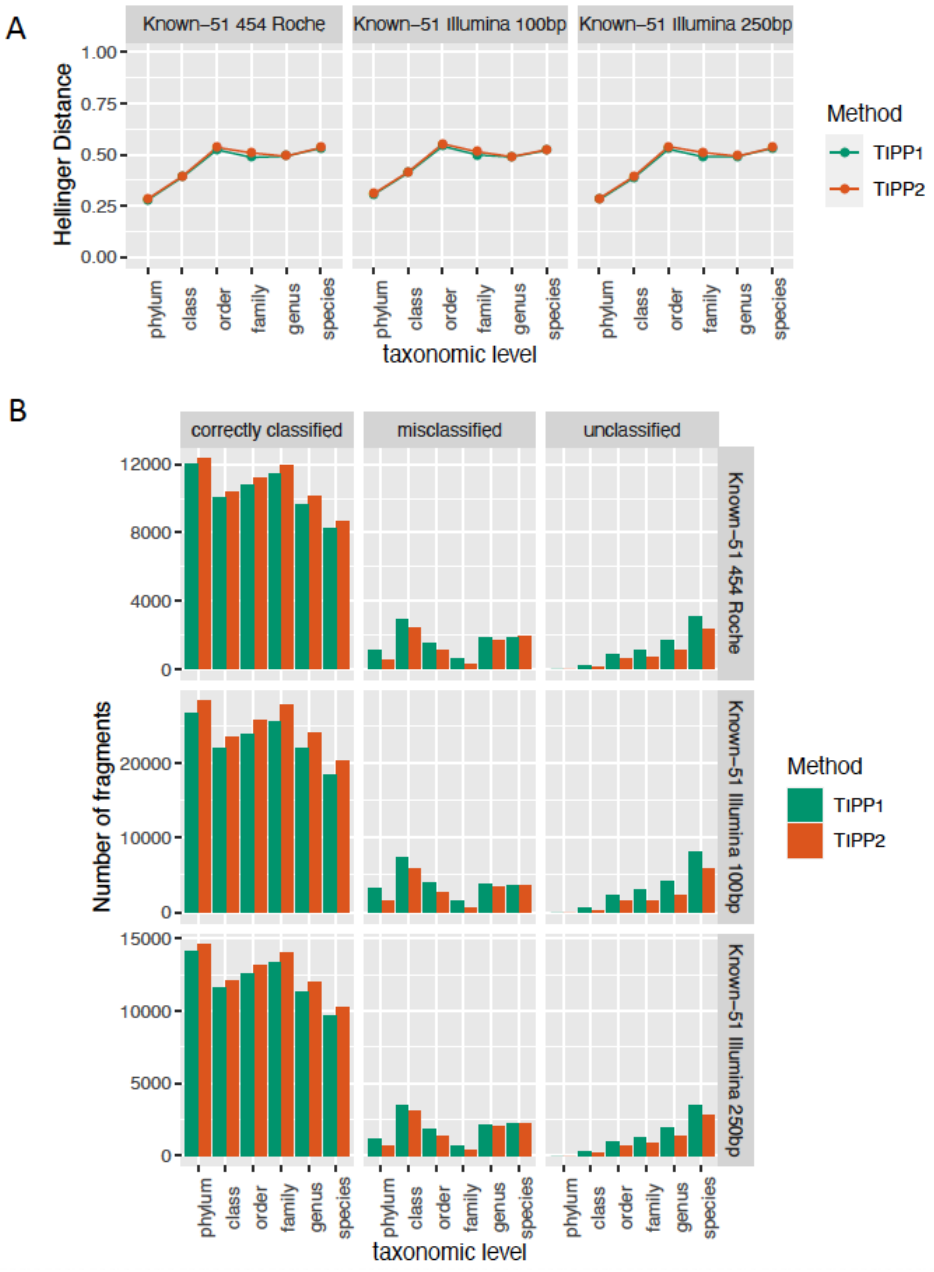

Figure 4.5: Comparing TIPP2 and TIPP1 using the reference package from 2014 (TIPP1 reference package) on the known genome datasets. (A) The Hellinger distances to the true abundance profile (i.e., error) for TIPP2 and TIPP1 on the known genome datasets. (B) Number of fragments (reads) correctly classified, misclassified, and unclassified by TIPP2 and TIPP1 in known genome datasets.

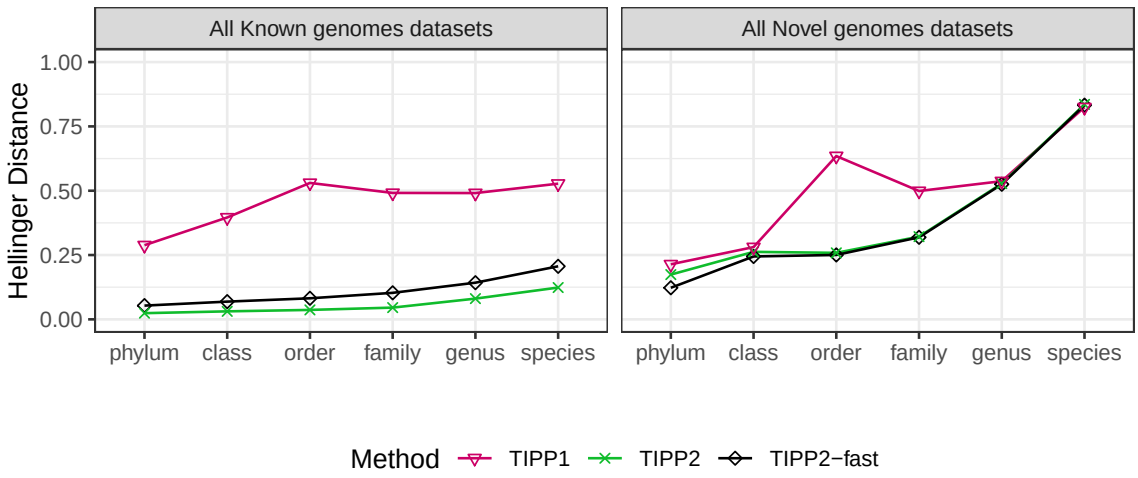

Figure 4.6: Experiment 2: Evaluating TIPP2 to TIPP. We show error in abundance estimates on simulated metagenomic datasets from known and novel genomes, with different sequencing technology and read lengths. We show average Hellinger distance for TIPP2 using all marker genes, TIPP2-fast using three marker genes, and TIPP1.

4.4.3 Experiment 3: Comparing TIPP2 with other methods

In Experiment 3, we used all testing datasets with known and novel genomes to compare the performance of different abundance profiling methods. Figure 4.7 shows the Hellinger distances for three known genomes datasets (top row), and the three novel genomes datasets (bottom row). Across datasets with different read length and sequencing technologies, we observe very minute differences in Hellinger distances. The trends are consistent and robust regardless of the read length and technology. As expected, we observe higher Hellinger distances for all methods when working with datasets with novel genomes than datasets with known genomes.

For the known genomes dataset, MetaPhlAn2 consistently has higher error

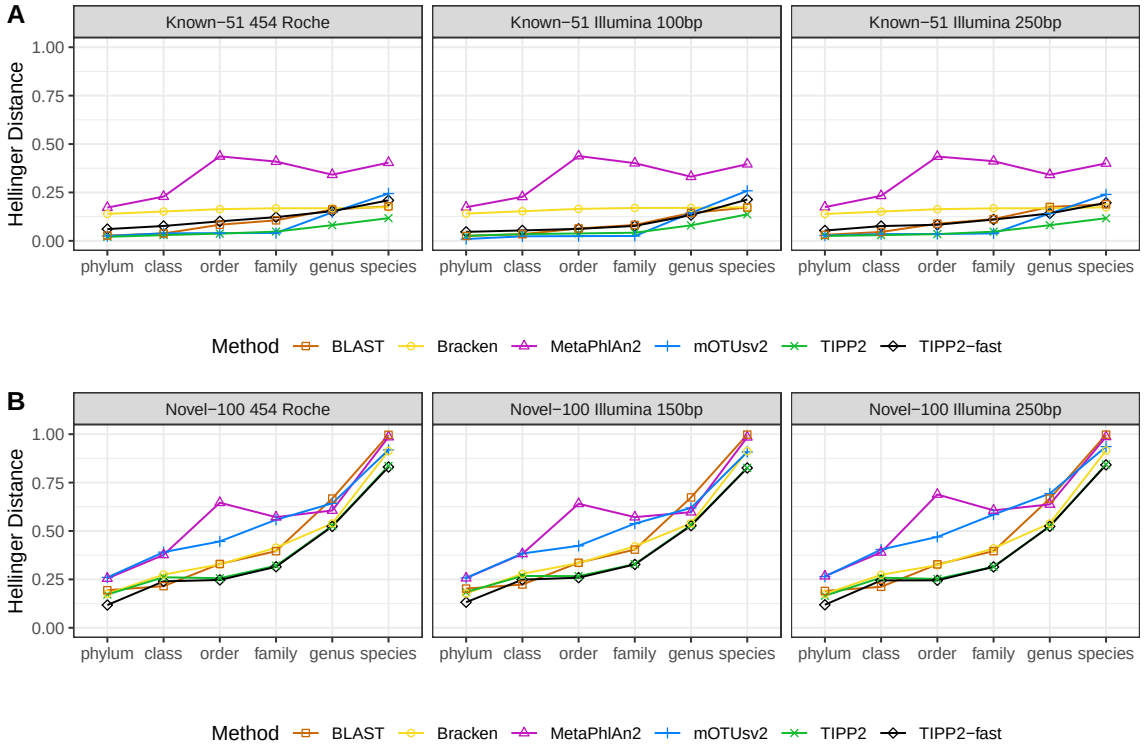

Figure 4.7: Experiment 3: Error in abundance estimates of simulated metagenomic datasets containing known genomes (A) and novel genomes (B). For each dataset, we show Hellinger distance between estimated profile and true abundance profile for TIPP2, TIPP2-fast, and other metagenomic abundance profiling methods.

compared to other methods. Bracken has the second highest error, after MetaPhlAn2, at all taxonomic levels except at the species level where it performs similar to BLAST, mOTUsv2, and TIPP2-fast. BLAST performs very similar to TIPP2-fast in these datasets. At the genus and species levels, mOTUsv2, BLAST, TIPP2-fast, and Bracken have very similar performance. mOTUsv2 and TIPP2 have the best performance at the higher taxonomic levels (phylum to family), but at the genus and species levels, TIPP2 outperforms mOTUsv2 and all other methods.

In the novel genomes datasets, TIPP2-fast and TIPP2 have similar performance except at the phylum level, where TIPP2-fast has lower error than TIPP2. At all taxonomic levels, TIPP2-fast has comparable or better performance than all other methods. BLAST has similar performance as TIPP2 at the phylum and class levels, but incurs higher errors at the order, family, genus, and species levels. MetaPhlAn2, followed by mOTUsv2, have larger Hellinger distances than all other methods at higher taxonomic levels of phylum, class, order, and family. At the genus level, BLAST has the worst performance, closely followed by mOTUsv2 and then MetaPhlAn2; and at the species level, BLAST and MetaPhlAn2 have the worst performance, closely followed by mOTUsv2 and Bracken.

4.4.4 Running time

We generated five replicates of 2,000,000 reads from the novel-100 genomes datasets with 454 and Illumina sequencing technologies. Table 4.3 shows the average wall clock time to run these methods on a computer with 16 CPUs and 32 GB

2,000,000 reads from novel-Illumina datasets	
Method	Average wall clock running time (hrs)
TIPP2-fast	1.0
TIPP2	5.0
MetaPhlAn2	0.3
mOTUsv2	0.3
Bracken	<0.1

2,000,000 reads from novel-454 datasets	
Method	Average wall clock running time (hrs)
TIPP2-fast	0.8
TIPP2	5.2
MetaPhlAn2	0.2
mOTUsv2	0.2
Bracken	<0.1

Table 4.3: Average wall clock running time, in hours, analyzing five replicates of 2,000,000 reads for each sequencing technologies. Each method was run on a dedicated node with 16 CPUs and 32 GB of memory. All methods have multi-threading implementation, so took advantage of all available CPUs.

memory. All methods are multi-threaded and were able to exploit parallelism by using multiple cores. Bracken was the fastest of all methods, finishing in less than 6 minutes. Both mOTUsv2 and MetaPhlAn2 performed similarly and finished in under an hour. Even though TIPP2 performs complex alignment and phylogenetic placement steps, it is able to complete within five hours. TIPP2-fast, which uses fewer markers, is significantly faster, and completes within an hour.

To evaluate the running time and peak memory usage for TIPP2, we generated five datasets, varying the total number of reads from 1M to 10M, where 1% of reads align to the TIPP2 marker genes and the rest do not align to marker genes. We generated four replicates for each dataset to document variation in measurements. For all datasets, TIPP2 was run with 8 CPUs and 36 GB of memory allocation. In the TIPP2 pipeline, there is a constant cost for creating the reference databases (specifically the HMMs) for each run on the fly. In our experiments, this step took 27 CPU minutes and 0.8 GB of memory; this is a small overhead cost in comparison to the rest of the pipeline usage, and its impact on total CPU time and peak memory decreases with the increase in number of reads. We also observe that the read binning phase takes significantly less time than the placement and

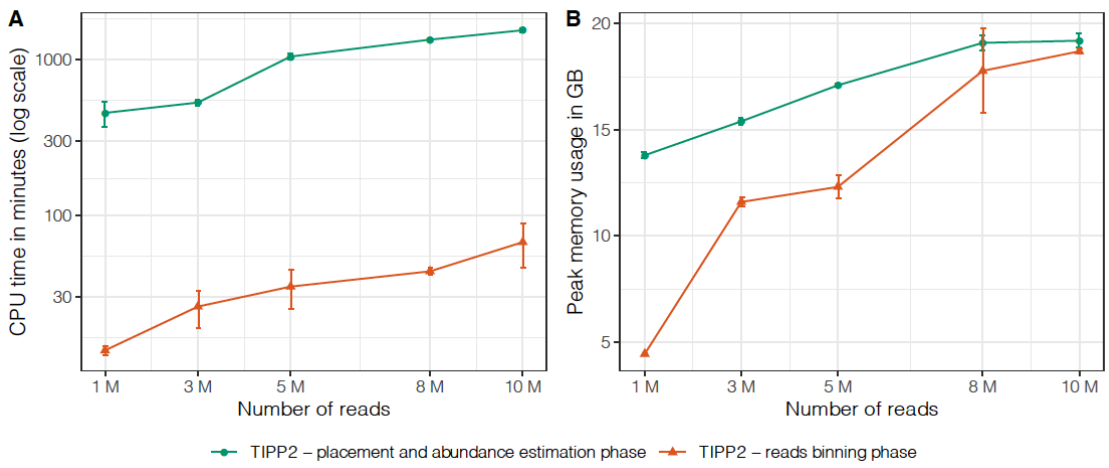

Figure 4.8: CPU time (A) and peak memory usage (B) for five datasets with varying number of reads. Note that all reads are processed through the first read binning phase of TIPP2; however, only the reads that align to marker genes are processed through the placement and abundance estimation phases of the TIPP2 pipeline (this is 1% of the total set of reads).

abundance estimation phase of TIPP2 pipeline (Figure 4.8). Thus, even in large metagenomic datasets, the running time and memory usage will be mainly used for analyzing the reads aligning to marker genes. For example, if the dataset has ten million (10,000,000) reads and 1% of the reads align to marker genes, then this will be approximately 1500 CPU minutes.

4.5 Discussion

Our prior work showed that methods based on marker genes provide better abundance estimates than the composition-based methods [25]. In this study, we compared methods for metagenomic abundance profiling, including several based on marker genes and one (Bracken) that is composition-based. Sequencing technology

did not have a significant impact on the accuracy of any of the tools tested, and sequence length had only a small impact. However, methods performed differently when novel genomes are analyzed compared to known genomes, and differences between methods were also larger. In datasets that contained sequences from genomes closely related to those within the reference packages used by the various tools, most methods performed well, with TIPP2 and mOTUsv2 largely outperforming the other tools. When working with novel genomes, the performance of all methods degraded. TIPP2-fast and TIPP2 had the best performance in all datasets at almost all taxonomic levels. We conjecture that TIPP2 performs well with novel genomes due to its ability to detect distant homology through its use of sequence alignment (enhanced by the use of an ensemble of profile HMMs) and phylogenetic placement, which has shown improved recall compared to other methods (including single HMMs) in prior studies [104, 105].

Our study shows that TIPP2 improves on TIPP1, a consequence mainly of using a new reference package which contains a denser reference taxonomy and multiple sequence alignment for each marker gene. This result shows the importance of the choice of database in taxonomic classification and abundance estimation tools [90]. To ensure accurate annotation and abundance estimation, a custom, environment-specific database is desirable; however, many marker-genes based methods are released with fixed databases which make it nearly impossible to customize the tool for specific applications, or even to upgrade the database as more data become available. Consequently, within the TIPP2 package, we release code and documentation for creating new references to enable end-users to create custom databases.

One of the interesting observations from our study is that we can accurately estimate abundances using just a carefully selected small set of marker genes rather than working with a comprehensive set of marker genes. This is consistent with mOTUsv2 and mOTU, which also worked with a subset of ten marker genes from the set of forty genes known to be single-copy and universal [21, 26].

4.6 Conclusions

TIPP, introduced in 2014 (here referred to as TIPP1), provided high accuracy for abundance profiling of metagenomic reads. Here we introduced TIPP2, an updated version of TIPP1. TIPP2 not only provides more accurate abundance profiling than TIPP1, but also outperforms commonly used taxonomic profiling tools —especially when datasets contain genomes that are not closely related to the reference sequences used by these packages. These improvements will enable a more precise characterization of microbial communities, particularly those that contain species that are not well characterized in public databases. Moreover, the biodiversity on Earth remains under-explored, and tools like TIPP2 are critical for characterizing the composition of microbial communities, many of which are expected to include currently uncharacterized genomes.

Our work indicates several directions for future research. The main improvement of TIPP2 over TIPP1 was obtained through the modification to the reference package. While TIPP1 had 30 marker genes each with about 1300 sequences, our new reference package had 39 marker genes, each with about 4339 sequences; hence,

the revised reference package contains more marker genes and each marker gene sequence collection is more densely sampled. The improvement of TIPP2 over TIPP1 indicates the value added in increased taxon sampling, and suggests further improvement might be obtained by maintaining and improving the reference package used by TIPP2. For example, one of the major challenges for taxonomic annotation and abundance profiling tools is keeping up with constant re-arrangements, renaming, and changes in microbial taxonomy, spurred in part by metagenomic studies. As a result, taxonomic profiling tools need to be based on the most recent databases, since these should provide the most accurate annotations. In our experiments, we found that many species had changed their order-level labels after MetaPhlAn2 databases were released, and that those changes led to higher error in our evaluations (Experiment 3). Beyond constant updates to reference packages, there is a need for developing taxonomy-agnostic annotation approaches that rely on sequence characteristics rather than man-made taxonomic labels (which can have errors, as this history indicates). We also observed that increasing the taxon sampling of the reference database improves accuracy. However scaling up accurate phylogenetic placement to the number of publicly available sequences remains a challenge. Our study also suggests that additional investigation into the selection of a subset of the marker genes could be helpful in improving accuracy and reducing running time.

Chapter 5: Binnacle: using graph scaffolds improves the quality of metagenomic bins

This chapter contains material previously published in Binnacle: Using Scaffolds to Improve the Contiguity and Quality of Metagenomic Bins [106], which was joint work with Harihara Subrahmaniam Muralidharan, Jacquelyn S. Meisel, and Mihai Pop. HSM, NS, JSM, and MP conceived the research project. HSM and NS designed and implemented the algorithm, with the help of JSM and MP. HSM, NS, and JSM analyzed the data. HSM, NS, JSM, and MP wrote the manuscript.

5.1 Introduction

In chapters 2–4, we explore techniques to characterize the composition of the sample using a reference database of genomes available in public repositories. However, many microbes are still not studied, sequenced, or characterized in the databases. In this chapter, we focus on how to best reconstruct genomes from a metagenomic sample.

Whole metagenomic shotgun sequencing allows for a comprehensive analysis of microbial DNA from a sample. It has been instrumental in expanding our understanding of the functional potential and genetic composition of different microorgan-

isms that have not been previously cultured. An important step in characterizing organisms that have not been isolated is the reconstruction of their complete genome sequence [107, 108]. This process involves assembling short metagenomic reads into longer contiguous sequences (contigs) based on sequence overlap. Paired-end read information can then be used to link together and orient assembled contigs into scaffolds [28, 29, 30, 31]. However, constructing the genomes of organisms from a mixture (metagenomic assembly) is computationally challenging. The uneven abundance of organisms, repetitive sequences within and across genomes, sequencing errors, and strain-level variations within a single sample often contribute to incomplete and fragmented assemblies.

In order to improve upon the fragmented assemblies constructed by metagenomic assembly tools, researchers utilize a strategy called binning, which involves clustering together genomic fragments that likely originate from an individual organism. Several strategies have been proposed for metagenome binning. Classification-based approaches rely on assigning taxonomic labels to genomic contigs, then grouping together those contigs that share a taxonomic label [25, 32, 33, 34]. Classification-based approaches are limited to organisms (and genomic segments within) that are sufficiently related to known sequences, and will miss microbes that are yet to be characterized. Clustering-based approaches focus instead on genomic features, such as GC content, oligonucleotide frequencies and contig abundance (coverage), to cluster together contigs that share similar properties [35, 36]. While such approaches are effective even when an organism shares no similarity to any known sequences, they are stymied by genomic regions that have unusual DNA composition or that

appear at higher depth of coverage than other segments of the organism of interest such as plasmids and mobile genetic elements [37].

Clustering/binning has also been applied to genes rather than contigs [109]. The resulting clusters were termed co-abundance gene groups (CAGs). CAGs that contained a large number of genes, roughly equivalent to the expected number of genes in a bacterial genome were referred to as metagenome species (MGS). More recently, in metagenome binning, when a cluster of contigs represents a complete, or close to complete, genome, it is referred to as a "metagenome-assembled genome" (MAG). While it is possible to recover MAGs from automated metagenome binning algorithms, many of the clusters obtained are incomplete or contaminated, and manual "finishing steps" are required to recover MAGs. In this paper, because we work with clusters obtained directly from binning algorithms, we refer to them as metagenomic bins rather than MAGs unless, referring to high quality bins.

While scaffolding and binning are both approaches for grouping together contigs that belong to an individual organism, they are often applied independently of each other, with some exceptions. MaxBin [110], for example, uses genomic scaffolds as a substrate for binning, however, they appear to be handled as if they were linear contigs. A newer version of this tool, MaxBin 2.0 [111], focuses solely on contigs. COCACOLA [112] incorporates paired-end information as another source of linkage information during the binning process, and does not explicitly construct or leverage scaffold information. GraphBin2 [113] independently bins contigs then refines the bins in the context of an assembly graph, by correcting bin assignments and propagating labels to unbinned nodes in the graph.

Here, we demonstrate the effectiveness of explicitly accounting for scaffold information in binning. We describe novel algorithms for estimating scaffold-level depth of coverage information that are effective even for non-linear (graph) scaffolds, and show that variation-aware scaffolders, which detect and explicitly model ambiguity in the assembly graph, help further improve the completeness and quality of the resulting metagenomic bins. We present a new software tool, Binnacle that accurately computes coverage of graph scaffolds and seamlessly integrates with leading binning methods. We show that using graph scaffolds for binning improves the contiguity and quality of metagenomic bins and captures a broader set of the accessory elements of the reconstructed genomes. Binnacle is implemented in Python 3 and released open source on GitHub at https://github.com/marbl/binnacle.

5.2 Materials and Methods

Binnacle operates as an add-on to existing binning tools. It relies on MetaCarvel [31] to construct genomic scaffolds, then uses a new algorithm for estimating the depth of coverage/abundance of scaffolds from read-mapping data, taking into account genomic variation as well as potential mis-assemblies and other artifacts. The resulting abundance information across one or more samples is then provided to a binning algorithm in order to generate scaffold-level bins (Figure 5.1). Each step in this pipeline is described in more detail below.

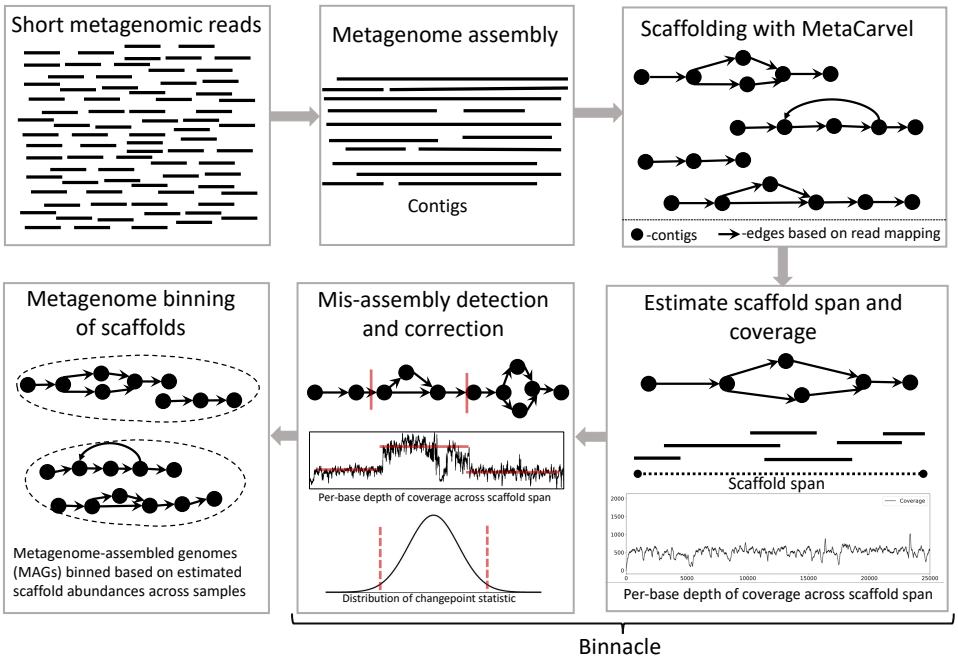

Figure 5.1: Short reads are assembled into contigs with a metagenome assembly tool. These contigs are oriented and ordered to generate graph scaffolds. For each scaffold, based on the length, orientation, and gap estimates, each contig in a scaffold is assigned global start and end coordinates; and the span of the scaffold is computed. Scaffold coverage is the per-base depth of coverage across the scaffold span. In the mis-assembly detection and correction routine, scaffolds are broken up if there are discontinuities in coverage signals. The final set of scaffolds and corresponding coverage information are used as input to binning methods to generate metagenomic bins.

5.2.1 Metagenome assembly

Like other binning approaches, Binnacle relies on the output of a metagenomic assembler. Any metagenomic assembler can be used to assemble the data, with the caveat that assembly errors can have a significant negative impact on binning. The results presented in this paper were generated by assembling each sample separately (i.e., avoiding a possibly expensive co-assembly step), and details about the tools and parameters used are presented below.

5.2.2 Scaffolding with MetaCarvel

Sequencing reads are mapped back to the assembled contigs, and the paired-end read information is used to scaffold the contigs using MetaCarvel [31]. This process results in a scaffold graph, where nodes are contigs and edges represent contig adjacencies inferred from paired-end read information. The scaffold graphs constructed by MetaCarvel are non-linear and preserve complex graph patterns, such as bubbles, which manifest when contigs diverge into one or more paths before converging again. Such patterns typically correspond to sequence variants between closely related organisms within a community, such as insertion/deletion (indel) events. Binnacle specifically works with the MetaCarvel scaffolder because of its unique ability to preserve variation in scaffolds.

5.2.3 Estimating scaffold span and coverage

One of the key features used by binning algorithms is information about the abundance/depth of coverage of genomic contigs, either within a single sample, or across multiple samples. To our knowledge, coverage estimation of scaffolds within metagenomic data sets has not been critically explored. Most current approaches rely on raw read counts averaged across the contigs or scaffolds being binned, similar to the "reads per kilo-basepair per million" (RPKM) measure used in RNA-seq analysis. A number of artifacts impact coverage estimation from scaffolds using such an approach, including potential overlaps between contigs (particularly relevant within regions of genomic variation), and assembly or scaffolding errors.

In non-linear "graph" scaffolds, such as those generated by MetaCarvel, the genomic extent covered by the scaffold cannot be directly inferred from the size of the contigs that are scaffolded together. To estimate the scaffold span —total effective length of the scaffold, i.e., the distance from the starting contig to the maximal rightmost coordinate of contigs contained in the scaffold —we rely on the following algorithm. For every graph scaffold, we identify a node with in-degree 0 which is assigned coordinate 0. If a scaffold contains no nodes with in-degree 0, we break the cycle using an approximation of the minimum feedback arc set problem. This problem is known to be NP-complete [114, 115] and hence we use an approximate solution: delinking the incoming edges of a vertex with the lowest in-degree. Coordinates for the other contigs in the scaffold are assigned in a breadth-first manner taking into account the length of the contig, the length of overlap

Algorithm 1: Pseudo-code to Assign Coordinates to Contigs in a Scaffold

```
Input  : Scaffold Subgraph G
Output: coordinates
if (The minimum in-degree is 0) then
    source ← Node with in-degree 0
else
    Delink the predecessors of node with lowest in-degree.
    source ← Node with in-degree 0
end
q ← Queue()
q.enqueue(source)
coordinates ← {}
if source.orientation ==“Forward” then
    start, end ← 0, source.Length
else
    start, end ← source.Length, 0
end
while q ≠ φ do
    v ← q.dequeue()
    Mark v as visited.
    start_v, end_v ← coordinates[v]
    for n ∈ G.neighbors(v) do
        overlap ← G.edge[(v,n)].overlap
        if v.orientation == “Forward” then
            if n.orientation == “Forward” then
                end ← end_v + overlap
                start ← end + n.Length
            else if n.orientation == “Reverse” then
                start ← end_v+overlap
                end ← start + n.Length
            end
        else if v.orientation == “Reverse” then
            if n.orientation == “Forward” then
                start ← start_v+overlap
                end ← start + n.Length
            else if n.orientation == “Reverse” then
                end ← start_v+overlap
                start ← end + n.Length
            end
        end
        if coordinates[n].start < start then
            coordinates[n] ← (start,end)
        end
        if n NOT visited then
            q.enqueue(n)
        end
    end
end
return coordinates
```

Figure 5.2: Assigns start and end coordinates to contigs in a scaffold. The lowest start coordinate and the highest end coordinate determine the scaffold span.

between contigs, and the relative orientation of the contigs (Figure 5.2). If there are multiple possible coordinate assignments for a contig (vertex), we retain the one with the largest possible value. We use this heuristic because choosing any other strategy to break ties might lead to an artificial increase in depth of coverage and negatively impact coverage computation. The span of the scaffold is then assigned to the distance between the right-most and left-most ends of the scaffolded contigs, based on the inferred contig coordinates.

Once the coordinates are available, we map reads to the contigs using Bowtie 2 (version 2.3.0) [116] and estimate per-base contig coverage using the genomecov program in the bedtools (version 2.26.0) [117] suite with the options -bga and -split. The per-base coverage of the scaffold is computed by adding up the coverage information of the contigs that overlap at each position in the scaffold span.

5.2.4 Detection and correction of mis-assemblies

When building graph scaffolds, MetaCarvel uses mapping of paired-end reads to contigs to infer adjacency information, however, this approach can sometimes falsely link together contigs. To detect such events, we rely on discontinuities in the depth of coverage signal as follows.

Ignoring sequencing biases, we expect each genomic position within a scaffold span to be covered equally well (uniformly). Hence, we assume that the per-base coverage of each organism (scaffold) follows a Poisson distribution and can be approximated by a Gaussian distribution with a mean, μ and a variance, σ^2. To break

up scaffolds containing contigs possibly originating from multiple species, we rely on a change point detection algorithm [118, 119] that operates on the per-base coverage signals.

To identify change points, we slide a window w of size $|w|$ along the coverage signal, computing the empirical means and variances. The user can select any value of w, but by default, we set $|w| = 1500$ bp. For scaffolds shorter than 3000 bp, we recursively set $|w| = |w|/5$ until the scaffold length is at least $2w$. For each position i along the scaffold span, we note the mean μ_{i-1} and variance σ^2_{i-1} of the window w_{i-1} defining the coverages from the coordinates $i - |w|$ to i and the mean μ_i and variance σ^2_i of the window w_i defining the coverage along the positions from i to $i + |w|$. We identify the windows w_{i-1} and w_i with respect to the position i as predecessor and successor windows, respectively. Given the coverage distribution for the two windows, we compare these distributions using the two-sample Z-statistic given by,

$$Z = \frac{\mu_{i-1} - \mu_i}{\sqrt{\sigma^2_{i-1} + \sigma^2_i}} \tag{5.1}$$

The empirical distribution of the Z-statistic such derived forms a Gaussian distribution, and we select the points within the tails of the Z-statistic distribution as candidates for change points (by default, we set $\alpha = 1$ percentile). To reduce the potential for false-positives, we next check if the change points coincide with the start or end of a contig within the scaffold, which suggest that the identified contig is incorrectly linked into the scaffold. Therefore, we delink the contig from its predecessors if the change point coincides with its start and delink from its successor

Algorithm 2: Pseudo-code Describing the Changepoint Detection Algorithm for Identifying Outliers in Graph Scaffold Coverages

Input : coverage- Perbase coverage of the scaffold
coordinates- Coordinates of the contigs in the scaffold computed by Algorithm 1
$|w|$- Size of the sliding window
α- The threshold for identifying outliers
β- The cutoff parameter to delinking contigs

$Z_{stat} \leftarrow []$
for $i \leftarrow |w|$ *to coverage.Length-*$|w|$ **do**
  $w_{i-1} \leftarrow$ coverage[i-$|w|$,i]
  $w_i \leftarrow$ coverage[i,i+$|w|$]
  $\mu_{i-1}, \sigma_{i-1} \leftarrow$ mean(w_{i-1}), SD(w_{i-1})
  $\mu_i, \sigma_i \leftarrow$ mean(w_i), SD(w_i)
  $Z_{stat}[i] \leftarrow \frac{\mu_{i-1}-\mu_i}{\sqrt{\sigma_{i-1}^2+\sigma_i^2}}$
end
$Z_{low} \leftarrow$ Percentile(Z_{stat}, α)
$Z_{high} \leftarrow$ Percentile($Z_{stat}, 100-\alpha$)
$Z_{outliers} \leftarrow Z_{stat}[Z_{stat} > Z_{high} | Z_{stat} < Z_{low}]$
outliers $\leftarrow$ Index($Z_{outliers}$)
for $o \in$ *outliers* **do**
  for *contig* $\in$ *coordinates* **do**
    start,end $\leftarrow$ coordinates[contig]
    if $|o - start| \leq \beta$ **then**
      Delink the predecessors of contig
    if $|o - end| \leq \beta$ **then**
      Delink the successors of contig
    end
  end
end

Figure 5.3: Pseudocode describing the change-point detection algorithm. The algorithm takes in two parameters α and β denoting the threshold for identifying outliers and the cutoff parameter to delink contigs, respectively.

if the change point coincides with its end β = read length). We also note that there are a few change points identified by our algorithm that do not coincide with the start or end of a contig. These could be due to either statistical artifacts or errors introduced by the assembler, but we do not currently address these in Binnacle.

This change point detection algorithm can work with both contig and scaffold coverages. We note that 40% of the time, a change point coincides with the beginning or end of a contig. When this happens, we delink the contig in the scaffold (i.e., remove the connections between the contig and its neighbors, resulting in multiple scaffolds). The remaining 60% of change points either occur too close to a previously

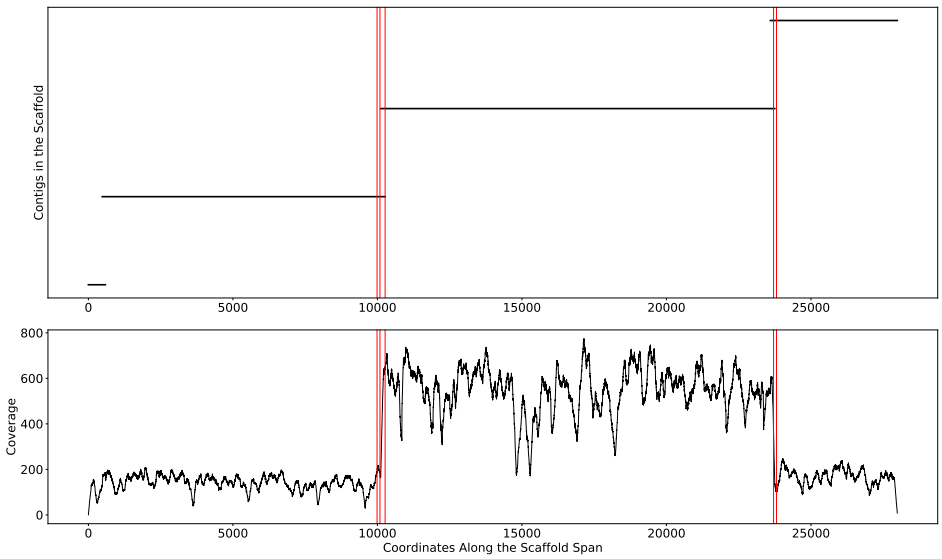

Figure 5.4: The mis-assembly detection algorithm in Binnacle. This is a scaffold from HMP sample SRS012902. The plot on the top shows the position of contigs along the scaffold span. The plot at the bottom shows the per-base depth of coverage across the scaffold span. The locations detected by the change point detection algorithm are highlighted by vertical red lines.

delinked contig or occur in the middle of contigs, revealing potential assembly errors. The handling of such situations requires further research that goes beyond the scope of this manuscript. The algorithm is described in detail in Figure 5.3. An example of the algorithm applied to a scaffold in the HMP dataset is shown in Figure 5.4. In the HMP dataset, an average of 4% of all the scaffolds were broken by change point detection.

After correcting potential scaffolding errors, Binnacle generates files reporting the per-base coverage for all scaffolds, describing the global coordinate information and describing the mean and standard deviation in coverage for all the scaffolds. In addition, we also provide a FASTA file of the final set of scaffolds after the misassembly detection routine. The abundance file and the scaffolds file provided by Binnacle can be readily used by existing binning algorithms. We currently provide interfaces to MetaBAT2 [120], MaxBin 2.0 [111], and CONCOCT [121].

5.2.5 Estimating scaffold coverage across multiple samples

The procedure described above is used when estimating scaffold coverage within the sample from which the scaffold is derived. If multiple samples are available, binning algorithms can leverage coverage information from all the samples to identify contigs/scaffolds that co-vary in abundance. When using multiple samples, the reads from each sample are mapped to the contigs/scaffolds of all of the samples and the mean abundance of each contig/scaffold is reported on a per sample basis. This approach produced fewer high contamination bins than binning with-

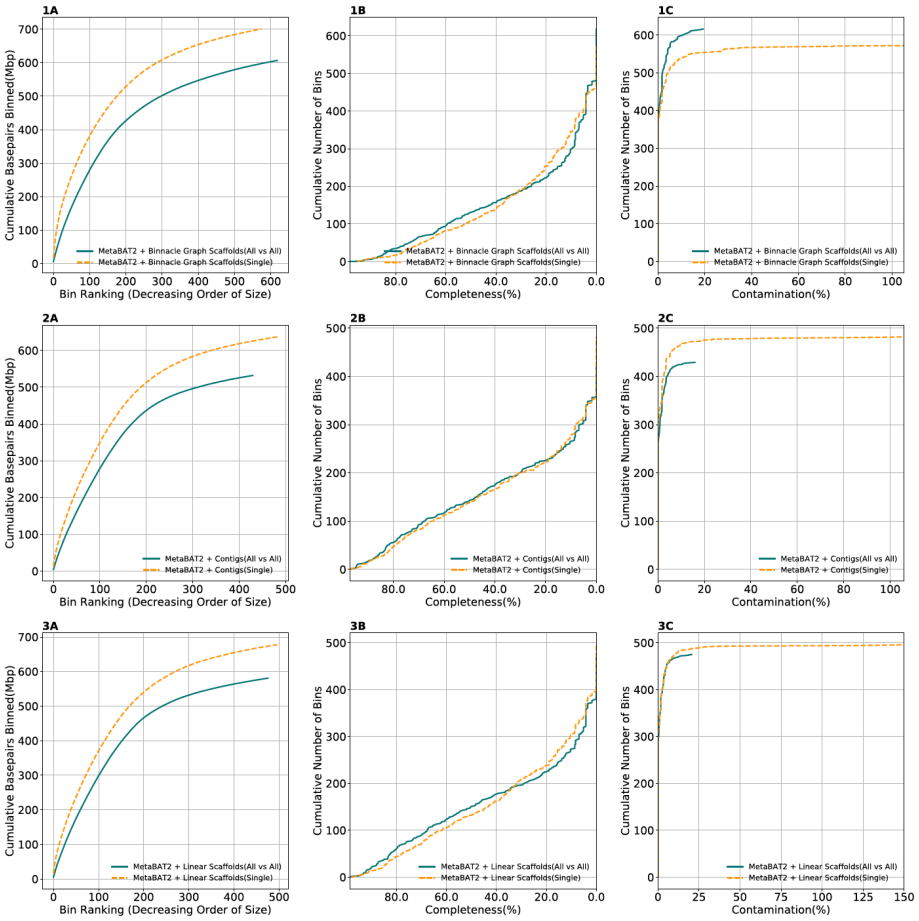

Figure 5.5: Binning using coverage information from all samples produces fewer high contamination bins for the HMP dataset. Comparing bins generated by MetaBAT2 with graph scaffolds (1), contigs (2), and linear scaffolds (3) for the HMP dataset. A) Cumulative base pairs binned when using coverage from the single sample (yellow dotted line), and when using coverage information from all samples (blue solid line). Bins are ordered in decreasing order of their size. The upper curve corresponds to higher contiguity for the same number of bins. B) Bins are ordered in decreasing order of their completeness value from CheckM evaluation. The upper curve indicates that more bins are contained at the same or higher level of completeness. C) Bins are ordered in the increasing order of their contamination value from CheckM evaluation. The higher curve indicates that more bins are contained at the same or lower level of contamination.

out combining coverage information from multiple samples (Figure 5.5). Identifying and comparing contigs across samples is challenging. Determining how to best use abundances estimated from multiple samples remains an active area of research.

5.2.6 Analysis of metagenomic datasets

To benchmark Binnacle, we first relied on a known-composition mock dataset described in [122], which is referred to as "simulated data" in the remainder of this paper. The corresponding data were obtained from the GigaDB database (http://dx.doi.org/10.5524/100719). We also evaluated our method on three real metagenomic datasets: (1) a time series of 18 fecal samples from a single premature infant (infant 31) from Sharon et al. study [123] referred to as the "infant gut data" in the remainder of this paper, (2) 20 complex stool samples from the Human Microbiome Project [124] referred to as the "HMP gut data", and (3) a time series of 12 samples from subject HV12 in a skin microbiome study [125] referred to as the "skin longitudinal data". All three datasets are complex, human-associated microbiomes. The infant gut data was selected because there is good understanding of the underlying community structure and the study assembled and published several reference genomes1 of organisms identified within these samples. For the three real metagenomic datasets, we downloaded reads from the NCBI read archive.

For the HMP gut dataset, we used IDBA-UD assemblies provided by the HMP consortium. For all other datasets, we assembled the reads into contigs using MEGAHIT (version 1.1.2) [126]. For all datasets, we generated scaffolds using

MetaCarvel [31]. Both MetaCarvel and MEGAHIT were run with default parameters. MetaCarvel outputs both variation-aware graph scaffolds and optimized linear sequences as linear scaffolds. Through Binnacle, a mis-assembly detection and correction routine was used to break up any mis-joined scaffolds, and then scaffold coverages were estimated. We refer to scaffolds obtained through the Binnacle step as "graph scaffolds" and linear sequences from MetaCarvel as "linear scaffolds".

To assess the quality of binning, in the simulated data set we relied upon the known genome sequences from which this dataset was constructed. Similarly, the publication describing the infant gut dataset identified a set of 33 reference genomes that were present in these samples, which we use as a reference for validation. In both datasets, we aligned the binned contigs to the reference genomes using minimap2 (version 2.1) [127]. Each bin was assigned to the genome to which the majority of base pairs aligned. We compute completeness as the percentage of the assigned genome represented in the bin, and contamination as the percentage of base pairs in the bin that did not align to the assigned genome. For the HMP gut data and the skin longitudinal data, where reference genomes were not available, we used CheckM (version 1.0.11) [128] to compute the completeness and contamination of the bins.

In the simulated dataset, we tested three binning methods —MaxBin 2.0 (version 2.2.5) [111], COCACOLA [112], and MetaBAT2 (version 2.12.1) [120] focusing on three features: contigs, linear scaffolds, and graph scaffolds. All methods employ a different threshold on the length of contigs used for binning. To make comparisons across binning methods fair, we ran MaxBin 2.0, COCACOLA, and MetaBAT2 with the same contig threshold ($>$ 2500 bp). COCACOLA can use paired-end informa-

tion to assist binning. To assess the effectiveness of this feature we ran COCACOLA in paired-end mode on the assembled contigs. When applied to graph scaffolds and linear scaffolds, we disabled COCACOLA's paired-end processing.

MetaBAT2 generated bins with lower contamination than both MaxBin 2.0 and COCACOLA (discussed later in results). Hence, for the three real metagenomic datasets, we only show results obtained with MetaBAT2 [120] (default parameters). MetaBAT2 uses the abundances and sequence composition information to bin genomic sequences. We estimated the coordinates, span, and abundance of scaffolds using Binnacle for each sample with its own set of reads. We then estimated abundances for each scaffold along the scaffold span using the reads of all other samples in the dataset as additional features. Similarly, while binning with contigs and binning with linear scaffolds, we computed mean and variance of coverages from all samples.

To examine bins in the skin longitudinal dataset, we focused on bins that belonged to the *Cutibacterium* (*Propionibacterium*) genus, as identified by CheckM [128]. We extracted the contigs within each bin and aligned them to the *Cutibacterium acnes* KPA171202 reference genome (GCA_000008345.1) using MetaQUAST [129]. Contigs within linear and graph scaffolds were used (instead of the scaffold sequences) to prevent misalignment of structural variant features. For pangenome analyses, a total of 27 complete *C. acnes* reference genomes were downloaded from NCBI. Genes were predicted from these references using Prokka [130] and the pangenome was calculated using Roary [131]. Genes found in all 27 references were considered "core" genes and those found in at least 2 samples were considered

"accessory." Genes were predicted in the MAGs using Prodigal [132] with the "-p meta" option and were aligned using BLAST [45] against the pangenome reference sequences (E-value 1e-3, percent identity 75). BLAST hits with a query and subject coverage of at least 50% were retained and annotated as either "core" or "accessory" genes. Genes with multiple hits were assigned to the hit with the greatest alignment length and percent identity. Genes identified in the metagenomic assemblies but not found in the reference genomes were flagged as "putative-accessory" genes. CRISPR/Cas elements were detected within the bins using CRISPRCasFinder on the web [133]. Contigs in MET0773 were annotated using Prokka v 1.12 [130] and visualized with the R package genoPlotR [134].

5.3 Results

To determine whether graph scaffolds can improve binning quality, we analyzed one simulated dataset and three sets of real metagenomic samples: infant gut samples, HMP gut samples, and skin longitudinal samples, described further in Methods. For samples from each of these datasets, we assembled and binned contigs and scaffolds with Binnacle and MetaBAT2.

5.3.1 Impact of accurate estimation of scaffold coverage/abundance

Depth of coverage information is one of the key features used by binning algorithms. Correctly estimating this information is difficult, particularly in metagenomic datasets where genomic variants and highly conserved regions confound the

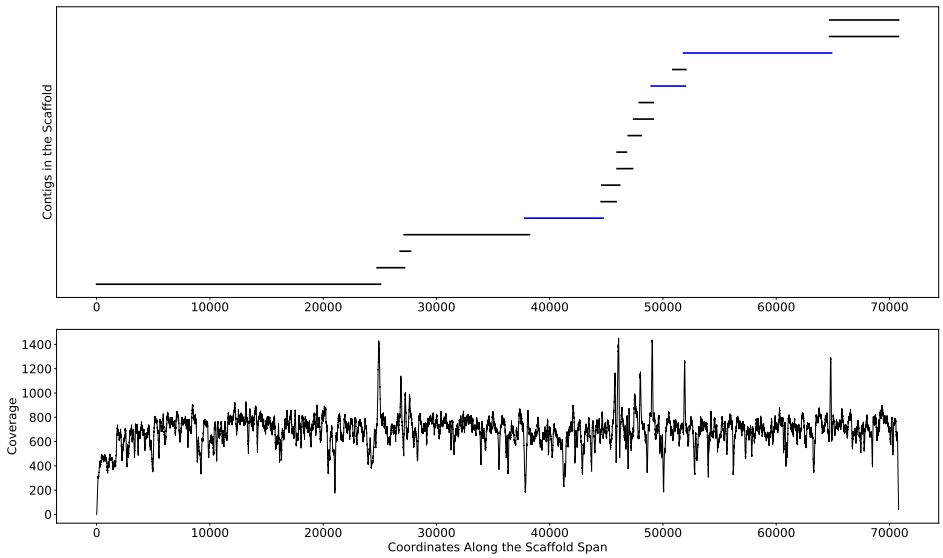

Figure 5.6: An example scaffold with coverage estimated with Binnacle. The plot at the top shows the position of contigs along the scaffold span. Contigs within the red dotted box are part of a bubble (signature of strain variation) detected by MetaCarvel. Only three contigs (highlighted in blue color) were binned by MetaBAT2 when contigs rather than scaffolds were provided as input. The plot at the bottom shows the cumulative per-base depth of coverage across the scaffold span as estimated by Binnacle.

signal. As described in Methods, Binnacle leverages information about the relative placement of contigs inside of a scaffold to better estimate abundance. As seen in Figure 5.6, the coverage signal estimated by Binnacle across the scaffold span of a single scaffold from the HMP stool sample SRS023829 is fairly uniform. This signal takes into account the overlap between multiple contigs, aggregating the coverage information within the overlapping region. The contigs from this scaffold can be assigned to organisms from the *Bacteroides* genus through a BLAST [45] search against the nt database. When using contigs alone for binning, only three of these contigs were binned (highlighted in blue color in Figure 5.6). Some of the unbinned contigs may have been excluded due to their size as, by default, MetaBAT2 only bins contigs greater than 2,500 base pairs. However, there were also several long contigs that remained unbinned despite having strong paired-end read connections to the rest of the contigs.

5.3.2 Binnacle improves contiguity, completeness, and contamination of bins

To assess the effectiveness of different types of information in binning, we provided binning algorithms with three sources of data: (i) contigs (the most common usage); (ii) linear scaffolds; and (iii) graph scaffolds that preserve the ambiguity introduced in the assembly graph by genomic variation. The comparison between linear scaffolds and graph scaffolds allows us to determine whether any improvement in binning effectiveness is due to the longer sequences provided to binning

algorithms, or if there is a real benefit in accounting for the structure of the graph in regions of genomic variation.

We compared results from three binning methods, MaxBin 2.0, COCACOLA, and MetaBAT2 each supplied with contigs, linear scaffolds, or graph scaffolds. For all three methods, bins generated with graph scaffolds comprised more base pairs, and had higher completeness and lower contamination than bins generated with contigs or with linear scaffolds (Figure 5.7). The simulated dataset contained 100 genomes. We aligned contigs from each bin to the known reference genomes and taxonomically annotated bins with the genome for which the majority of base pairs aligned. To ensure only one bin per reference genome, we only considered bins that were at least 50% complete. Graph scaffolds, linear scaffolds, and contigs recovered 40, 38, and 21 putative genomes on average, respectively. In the case of COCACOLA, a tool that can leverage paired-end information natively, we observed that its handling of this information was less effective than that provided by scaffolding approaches such as MetaCarvel (the basis for the scaffolds used in Binnacle) (second row in Figure 5.7). Moreover, when using paired-end information, contiguity and completeness were comparable; only contamination of the bins was improved. Irrespective of the binning method employed, graph scaffolds improved the contiguity, completeness, and contamination of the resulting bins. However, we used MetaBAT2 as the binning method for the remaining analyses in this paper.

We assessed both the completeness and level of contamination of the resulting bins from all three real metagenomic datasets. For the infant gut dataset, we computed completeness and contamination of the bins based on a set of 33 reference

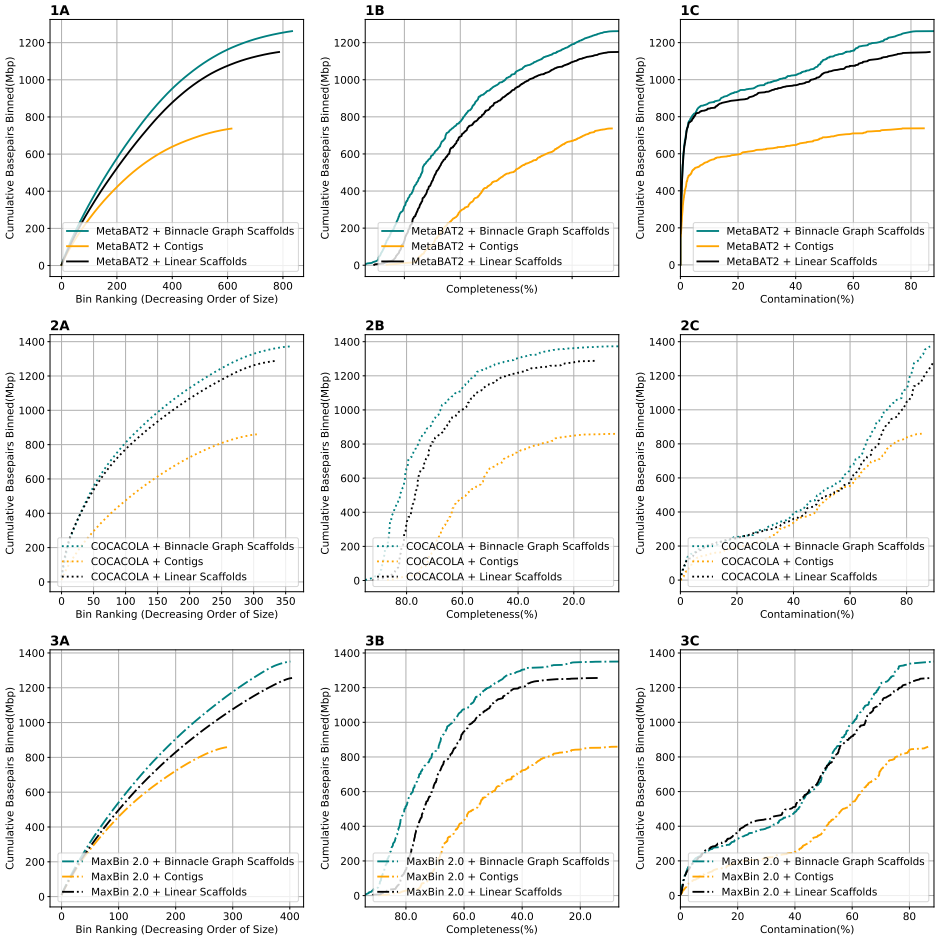

Figure 5.7: Binning with graph scaffolds improves contiguity, completeness, and contamination in genome bins from the simulated dataset. Comparing bins generated by MetaBAT2 (solid lines) (1), COCACOLA (dotted lines) (2), and MaxBin 2.0 (dashed-dotted lines) (3) using contigs (yellow), linear scaffolds (black), and graph scaffolds (blue) for the simulated dataset. COCACOLA contigs were binned both with and without paired end information. (A) Cumulative base pairs binned with contigs, linear scaffolds, and graph scaffolds. Bins are ordered in decreasing order of their size. The upper curve corresponds to higher contiguity for the same number of bins. (B) Completeness is defined as the percentage of the assigned genome represented in the bin. Bins are ordered in decreasing order of their completeness value. The upper curve indicates that more base pairs are binned by graph scaffolds at the same or higher level of completeness. (C) Contamination of a bin is defined as the percentage of base pairs that did not align to the assigned genome. Bins are ordered in the increasing order of their contamination value. The higher curve indicates that more base pairs are binned by graph scaffolds at the same or lower level of contamination.

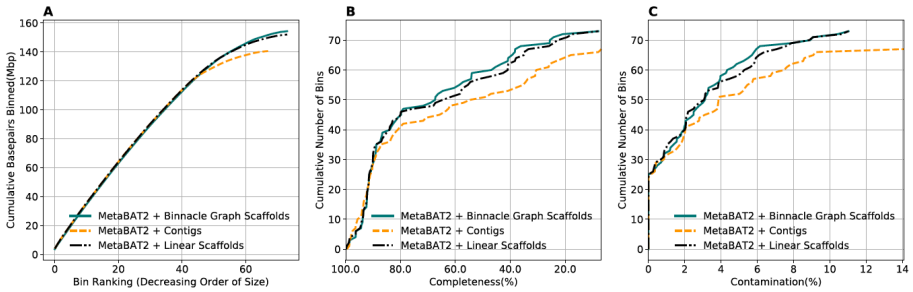

Figure 5.8: Binning with graph scaffolds improves contiguity, completeness, and contamination in genome bins from the infant gut dataset. Comparing bins generated by MetaBAT 2 using contigs, linear scaffolds, and graph scaffolds for the infant gut dataset. A) Cumulative base pairs binned with contigs, linear scaffolds, and graph scaffolds. Bins are ordered in decreasing order of their size. The upper curve corresponds to higher contiguity for the same number of bins. B) Completeness is defined as the percentage of the assigned genome represented in the bin. Bins are ordered in decreasing order of their completeness value. The upper curve indicates that more bins are contained in graph scaffolds at the same or higher level of completeness. C) Contamination of a bin is defined as the percentage of base pairs that did not align to the assigned genome. Bins are ordered in the increasing order of their contamination value. The higher curve indicates that more bins are contained in graph scaffolds at the same or lower level of contamination.

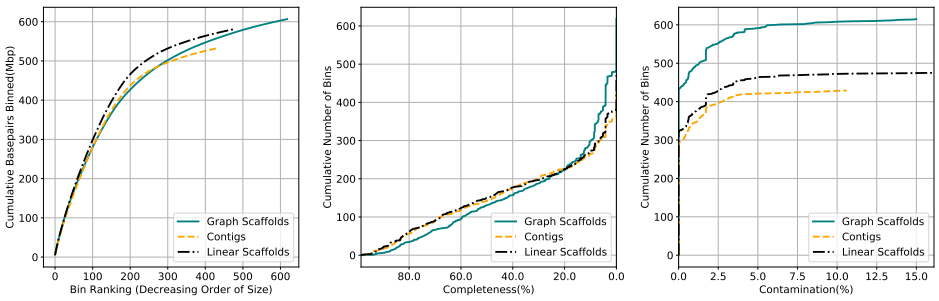

Figure 5.9: Graph scaffolds bin more contigs and reduce bin contamination in the HMP gut dataset. Comparing bins generated by MetaBAT2 using contigs, linear scaffolds, and graph scaffolds for the HMP gut dataset. The completeness and contamination of bins were evaluated with CheckM. (A) Cumulative base pairs binned with contigs, linear scaffolds, and graph scaffolds. Bins are ordered in decreasing order of their size. The upper curve corresponds to higher contiguity for the same number of bins. (B) Bins are ordered in decreasing order of their completeness value from CheckM evaluation. The upper curve indicates that more bins are at the same or higher level of completeness. (C) Bins are ordered in the increasing order of their contamination value from CheckM evaluation. The higher curve indicates that more bins are at the same or lower level of contamination.

genomes that were identified to be present in these samples (See section "Materials and Methods"). Similar to the performance on simulated data, bins generated with graph scaffolds contained more base pairs than bins generated with contigs and linear scaffolds (Figure 5.8). Moreover, bins from graph scaffolds had higher completeness and lower contamination than bins generated with contigs and linear scaffolds.

We next analyzed complex metagenomic samples from the HMP gut study. We did not have prior information about the community structure and genomes present, so we used CheckM [128] to evaluate the bins. CheckM uses sets of highly prevalent single-copy genes to assess the overall quality of genomes or genome bins, including their completeness, contamination, and strain heterogeneity. Bins gener-

ated from linear scaffolds grouped more base pairs than bins generated with contigs (Figure 5.9A). They also had comparable completeness and generally lower contamination (Figures 5.9B,C). When using graph scaffolds that include potential strain variants, the contiguity of the resulting bins improved, and a majority of bins have low contamination level (Figure 5.9, solid blue line).

Samples in the HMP gut dataset contained an average of 70 million reads. Binnacle took an average of 7.75 min to run (min = 2.7, max = 96.75, SD = 31.75 min) and had a peak memory usage of less than 3GB on average (min = 1.6, max = 10, SD = 2.57 GB). We ran these samples on a Linux computing cluster specifying a memory limit of 36 GB using a single processor. Given that these jobs took less than 10 GB of memory to run, they should run efficiently on most modern computing hardware.

5.3.3 Binnacle recovers *Cutibacterium acnes* bins from sebaceous skin samples

To further evaluate Binnacle's performance, we used it to bin the skin longitudinal dataset with multiple samples from two sebaceous, or oily, skin sites —the back of the head (occiput) and the external auditory canal of the ear —as well as two moist body sites —the toe web and plantar heel —all from the same healthy volunteer. Within these samples, there were similar improvements in bin contiguity, completeness, and level of contamination when binning graph scaffolds compared to when binning contigs and linear scaffolds (Figure 5.10).

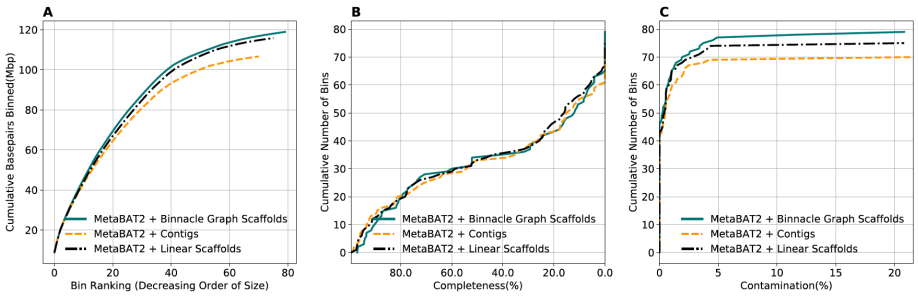

Figure 5.10: Binning with graph scaffolds improves contiguity, completeness, and contamination in genome bins from the skin longitudinal study dataset. Comparing bins generated by MetaBAT 2 using contigs, linear scaffolds, and graph scaffolds for the skin longitudinal study dataset. A) Cumulative base pairs binned with contigs, linear scaffolds, and graph scaffolds. Bins are ordered in decreasing order of their size. The upper curve corresponds to higher contiguity for the same number of bins. B) Bins are ordered in decreasing order of their completeness value from CheckM evaluation. The upper curve indicates that more bins are at the same or higher level of completeness. C) Bins are ordered in the increasing order of their contamination value from CheckM evaluation. The higher curve indicates that more bins are at the same or lower level of contamination.

Body Site	Timepoint	Sample	Method	# contigs	# contigs (>1,000 bp)	Total length	Total aligned length	Genome fraction (%)	# of bubble contigs
External auditory canal (Ea)	1	MET0308	contig	367	367	2,290,385	2,136,612	80.191	12
			linear scaffold	591	493	2,475,611	2,390,694	87.702	43
			graph scaffold	669	520	2,606,365	2,475,094	88.404	52
	2	MET0749	contig	237	237	2,601,507	2,452,477	93.375	4
			linear scaffold	288	256	2,662,358	2,502,296	94.439	7
			graph scaffold	305	260	2,680,429	2,514,146	94.495	7
	3	MET0768	contig	136	136	2,548,346	2,444,275	94.219	2
			linear scaffold	120	108	2,506,265	2,447,111	94.6	3
			graph scaffold	160	137	2,370,487	2,262,214	86.668	4
Occiput (Oc)	2	MET0754	linear scaffold	1059	671	1,711,485	1,559,541	57.717	0
			graph scaffold	972	625	1,606,432	1,463,457	54.226	0
	3	MET0773	contig	365	365	1,850,617	1,782,219	67.091	5
			linear scaffold	742	546	2,342,529	2,183,639	77.716	41
			graph scaffold	966	677	2,777,243	2,460,422	81.625	73

Table 5.1: *Cutibacterium* bins detected in the skin longitudinal samples.

Cutibacterium acnes, formerly referred to as *Propionibacterium acnes*, is a known prominent bacterial community member at sebaceous skin sites because it utilizes the fatty acids in the sebum (the oily substance produced by sebaceous glands) for energy. Different strains of the commensal *C. acnes* have been associated with acne vulgaris [135]. Because of its prominence on the skin and its implications for skin health, we searched for this organism in the skin longitudinal dataset; we were able to recover bins belonging to the *Cutibacterium* genus from five of the six sebaceous samples (Table 5.1). These bins contained contigs belonging to *C. acnes*. We mapped the *Cutibacterium* bins to the reference genome for *C. acnes* and found that bins generated with graph scaffolds generally covered a greater proportion of the reference genome than bins generated with contigs and linear scaffolds. Furthermore, both linear and graph scaffolds were able to recover a *Cutibacterium* bin from sample MET0754 that was not identified when binning with contigs alone.

A common concern with binning algorithms is that they largely capture the core genome of organisms, omitting potentially relevant accessory genes. We classified *C. acnes* genes into core, accessory, and putative-accessory genes as described in Methods. As seen in Figure 5.11, bins constructed from graph scaffolds captured a

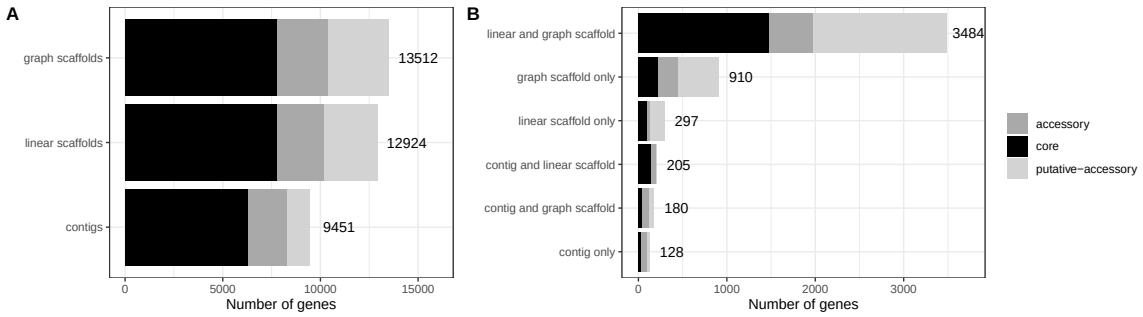

Figure 5.11: *Cutibacterium* bins generated by graph scaffolds capture more auxiliary genome elements. Genes predicted from *C. acnes* bins were mapped to genes from the *C. acnes* pangenome and characterized as core, accessory, or putative-accessory. The x-axis denotes the number of genes in all of the *C. acnes* bins and the y-axis denotes the method by which each gene was binned. The label denotes the total number of genes in each bar. In (A) all genes binned by each method are included in the bars, while in (B) they are separated by how they are shared across binning methods.

larger fraction of accessory and putative-accessory genes, while bins constructed from contigs (the most commonly used approach) contained mostly core genes. Among the accessory and putative accessory genes identified in the metagenomic assemblies, 86.9% were binned within graph scaffold bins (10.5% were uniquely binned by graph scaffolds and no other methods).

5.3.4 Binnacle captures structural genomic variation

By using scaffolds that include structural variants, we intended to capture genes and genomic elements that are typically missed by contig-based analyses. As shown in Table 5.1, many contigs identified within variant regions by MetaCarvel appeared only in bins constructed from these scaffolds, i.e., the information typically used by binning algorithms was not able to associate these contigs with the *C. acnes*

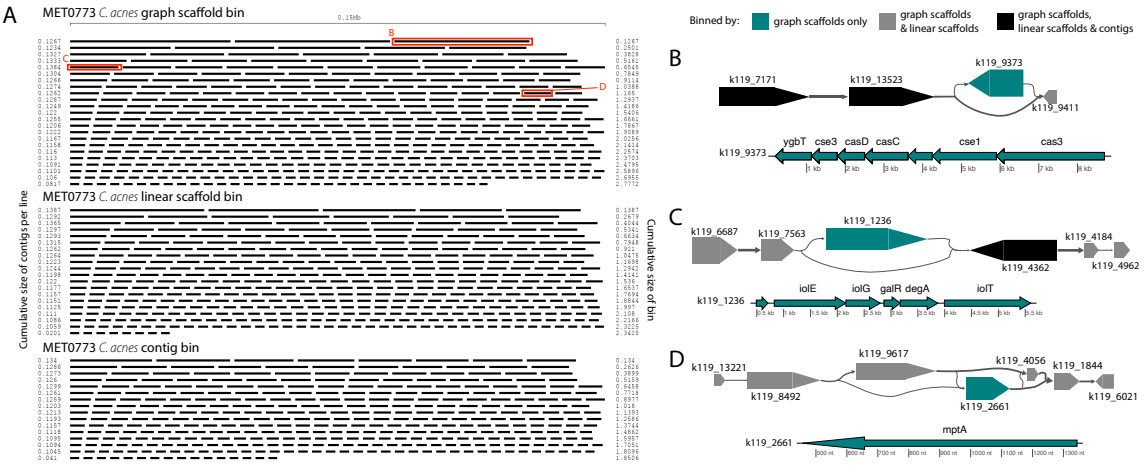

Figure 5.12: *Cutibacterium* bins in sample MET0773. (A) Ordered lengths of graph scaffolds (top), linear scaffolds (middle) and contigs (bottom) included in *C. acnes* bins, highlighting the greater fragmentation in the bin generated using contigs. Red boxes highlight graph scaffolds depicted in parts (B-D). In (B-D), the large arrows represent contigs in a single graph scaffold. Lines connecting contigs denote paired-end read support. Contigs are colored to indicate the methods that include them in the C. acnes bins. Scaffold plots were generated by MetagenomeScope [136] but updated and modified to improve visualization in Illustrator. Genes in contigs uniquely binned by graph scaffolds are depicted below the scaffold as thin arrows. Genes were predicted and annotated by Prokka [130] and visualized with the R package genoPlotR [134].

genome.

In sample MET0773, all three scaffolding methods detected a *C. acnes* bin (Table 5.1), however, the *C. acnes* bin generated using graph scaffolds was more contiguous and had less fragmentation than the bin generated using contigs (Figure 5.12A). Furthermore, a total of 32 variant contigs (2 indels, 20 simple strain variants, and 10 complex strain variants) were uniquely identified in the *C. acnes* bin generated using graph scaffolds. One such variant contained elements of the subtype I-E CRISPR-Cas system (Figure 5.12B) that has previously been characterized in *C. acnes* [137]. Within this same sample, a contig that was not in a structural variant but was uniquely binned using graph scaffolds contained a CRISPR array with five spacers, one of which had close similarity to the *Cutibacterium phage PAVL21* genome (Supplementary Table 3). Another indel that was only binned by graph scaffolds contains genes involved in the degradation of myo-inositol into acetyl-CoA (Figure 5.12C). In *Corynebacterium glutamicum*, genes involved in this pathway allow the bacterium to use myo-inositol as a carbon and energy source [138]. This indel also contains genes encoding two HTH-type transcriptional regulators (galR and degA). A contig uniquely binned by graph scaffolds in a complex strain variant contains a gene annotated as mptA (Figure 5.12D); in *Mycobacterium tuberculosis* and *C. glutamicum*, this gene is involved in the biosynthesis of cell-wall associated lipomannan that has several immunomodulatory properties *mishra2011lipoarabinomannan*.

5.4 Discussion

Binning (based on sequence composition and depth of coverage) and scaffolding (based on paired-end information) provide complementary approaches for grouping together contigs from metagenomic samples that likely originate from the same organism. At the outset of our study, we hypothesized that combining the two approaches would yield improvements in the contiguity and quality of the resulting bins. While others have used paired-end read or scaffold information to augment binning, we identified a major overlooked factor —the computation of depth of coverage at a scaffold level, computation that can be impacted by scaffolding errors and strain variation. To our knowledge this contribution is novel, and as we have shown, providing binning algorithms with depth of coverage information derived from linear and non-linear (graph) scaffolds improves the quality of the bins over what can be achieved by binning contigs alone.

We attribute the improvements we have demonstrated to three factors. The first is, as already mentioned, a more accurate estimation of scaffold depth of coverage, information used by the binning algorithm to determine which contigs or scaffolds should be grouped together. The second is simply the longer-range information available in scaffolds as opposed to individual contigs. A third factor is the use of variation-aware scaffolds which were referred to as "graph scaffolds" in the manuscript.

Binning algorithms rely on depth of coverage and sequence composition information, and accurately estimating this information requires long genomic segments.

As a result, small contigs get excluded from binning either by design or because of incorrect estimates of coverage or sequence composition. The longer genomic context of scaffolds provides an opportunity for binning algorithms to more accurately estimate the information necessary for binning. Furthermore, certain genomic regions, such as mobile elements, usually have a different sequence composition from the rest of the genome (this is in fact one of the signals used to detect such regions) and may, therefore be missed. Paired-end information, however, can link together contigs irrespective of length and sequence composition, thereby capturing a larger fraction of the sequence from the assembly. These links are generally accurate; in the simulated dataset over 99% of the paired-end reads linked contigs belonging to the same species.

Typically, metagenome assemblers and scaffolders attempt to construct a single linear sequence representing a segment from the chromosome of an organism in the sample. In many cases, however, such a linear representation ignores the presence in the sample of multiple variants of an organism, not unlike the presence of multiple isoforms of genes in eukaryotic transcriptomes. By explicitly modeling this variation, Binnacle is able to more accurately estimate the depth of coverage of scaffolds, thereby improving the efficacy of the binning process. When considering only a linear representation of a contig or scaffold, conserved genomic regions would appear to have higher depth of coverage than the variant regions. We examined the distribution of coverage across contigs, linear scaffolds, and graph scaffolds. In the human metagenomic datasets analyzed here, the median coverage of contigs binned was 4.2 (Sharon), 19.5 (skin), and 23 (HMP). We found that graph scaffolds are not

biased toward contigs that are more highly abundant. In fact, graph scaffolds have the ability to bin variants that are usually lower coverage, simply because variants are linked to higher coverage neighbors.

We observed that binning results varied widely across samples. When samples had great strain diversity, like the mock community that contains over 100 different taxa, using graph scaffolds significantly improved the contiguity and quality of the bins. However, when samples were less diverse, like those in the Sharon dataset, all binning approaches produced similar results. The complexity and strain diversity of a sample have a significant impact on the effectiveness of binning, and on the improvement that can be obtained by leveraging variation-aware scaffolds.

Another advantage of working with variation-aware scaffolds in Binnacle is that the resulting bins contain a better representation of the genic content of the organisms from the sample. In our investigation of *C. acnes* in the skin microbiome, bins constructed from graph scaffolds contain a larger number of accessory genes than bins constructed from linear scaffolds or contigs. Furthermore, graph scaffold bins uniquely identified contigs in structural variants that were related to the CRISPR-Cas system, catabolic processes, transcriptional regulation, and cell wall biosynthesis; traditional binning approaches missed the association of these variants with this genome. We hope that this observation will further strengthen the case for the development and use of tools that explicitly model strain variation when analyzing metagenomic data sets.

It is important to note that while read-based binning approaches exist [122, 139], many metagenome binning methods, including Binnacle, can only work with

assembled sequences from the sample. It has been shown that assembled sequences improve taxonomic classification [140]. Generally, reads from rare species and low-coverage regions do not assemble well. Thus, binning methods may not be effective for low abundance species. Another important but often overlooked point is the variable resolution of bins obtained. Even though one would like to obtain all bins as species-level metagenome assembled genomes, this goal is rarely achieved in practice. First, it is important to note that the concept of a bacterial species is not well defined. Second, the level of sequence divergence between closely related organisms varies widely across the bacterial taxonomy and even across the length of genomes. This may explain the somewhat surprising observation that Binnacle maintains low bin contamination even when using graph scaffolds that include sequence variation. CheckM relies on the number of multicopy marker genes to compute contamination, and these genes are more likely to be conserved among the strains forming the pangenome represented by Binnacle bins. In mock communities, we were able to compute contamination more precisely by mapping contigs to the relevant reference genome sequences. Even in this setting, the use of graph scaffolds did not result in higher contamination levels. As we have noted earlier, the paired end information we used accurately linked together contigs from the same organism, i.e., the underlying scaffold information itself has a low level of contamination. We hypothesize that the longer context provided by scaffolds allows binning algorithms to more accurately detect relationships between sequences derived from a same organism, thereby leading to lower levels of contamination than when using contigs as a substrate for binning.

In its current implementation, Binnacle does not attempt to resolve the multiple strains/haplotypes represented in its bins. A number of algorithms developed for haplotype phasing [141, 142], viral quasi-species estimation [143, 144, 145], and species estimation in metagenomics [146] can be applied here to estimate the number of species in a bin, and to split bins into multiple MAGs. We intend to pursue this line of research in future iterations of our tool.

We would also like to argue for the importance of effective visualization tools that can provide researchers with more information about the relative placement of contigs within a bin along a chromosome as well as variation information. Tools for visualizing assembly graphs, such as Bandage [147] and MetagenomeScope [136] are a first step in this direction, but these tools are still cumbersome to use in large data sets. Further opportunities for future research include new approaches for estimating depth of coverage, particularly when using data from multiple samples. While substantial progress has been made in the field of RNA-seq quantification [e.g., Salmon [148]], metagenomic approaches still rely on fairly simplistic assumptions.

We believe that Binnacle represents a first step toward the development of effective metagenomic analysis tools that can leverage all the information contained in one or more samples to reconstruct nearly complete genomic sequences, approaching the goal of automated reconstruction of MAGs.

Chapter 6: A critical assessment of gene catalogs for metagenomic analysis

This chapter contains material previously published in A critical assessment of gene catalogs for metagenomic analysis [149]. This project was joint work with Seth Commichaux, Jay Ghurye, Alexander Stoppel, Jessica A. Goodheart, Guillermo G. Luque, Michael P. Cummings, and Mihai Pop. This project was conceived and initiated at the 2017 Bioinformatics Exchange for Student and Teachers (BEST) summer school in Heiligkreuztal, Germany. MP conceived this project. All authors helped initiate the project. SC, NS, and MP were involved in the design and execution of all experiments. JG, AS, and JAG contributed to initial data analysis. SC, NS, and MP wrote the manuscript with contributions from all authors.

6.1 Introduction

Increasingly, studies of microbial communities rely on metagenomics —the sequencing of DNA extracted directly from a microbial mixture. Assembling metagenomic reads into longer contiguous sequences (contigs) is still a computationally challenging problem, because of repeated sequences within and among genomes, uneven abundances of organisms, sequencing errors, and strain-level variation. Due to

these challenges, and to limitations of sequencing technology, reconstructing complete and accurate genomes for all organisms in a single, complex metagenomic sample is still challenging. Given enough samples, metagenome assembled genomes can be reconstructed for many, but often not all, of the species comprising a microbiome. Regardless, metagenomic assemblies typically comprise many small contigs of unknown taxonomic origin.

The fragmented nature of metagenomic assemblies complicates data analysis, both because it is difficult to associate genomic fragments with individual taxa, and because it is difficult to identify related genomic fragments across samples. For these reasons, the earliest metagenomic studies focused on genes (and their inferred functions) found within assembled fragments, ignoring their precise taxonomic origin. Even in fragmented data, genes can be fairly effectively identified [150]. A gene-centric approach was used in the first large scale metagenomic study of ocean bacteria [39]. To prevent overcounting due to sequencing and assembly errors, or due to small differences in gene sequences within closely related organisms, Yooseph et al. [39] clustered the protein sequences based on similarity and focused their analysis on the representative sequence of each cluster. This gene "catalog" revealed the tremendous diversity of bacterial functions in the ocean, with the newly predicted protein sequences doubling the number of known proteins. The MetaHIT project [40] constructed a similar catalog in order to characterize the functional composition of the human gut microbiome. Qin et al. [151] leveraged a gene catalog as the basis for a microbiome association study in type 2 diabetes, and introduced the concept of metagenomic linkage groups —groups of genes that co-vary in abundance across

samples. The gene catalog thus represents the basis for grouping together genes that likely originate from a single organism, an idea further extended by Nielsen et al. [109] to help reconstruct partial genome sequences from metagenomic data.

Following these initial studies, gene catalogs have become ubiquitous in the analysis of metagenomic datasets, and have been created for the gut microbiota of multiple animals (e.g., mouse [152], rat [153], pig [154, 155], dog [156], cow [157], macaque [158], chicken [159], lion, leopard and tiger [160], ocean bacteria [161], soil bacteria [162], and the human vagina [163] and respiratory tract [164]. Gene catalogs are commonly used to: i) reduce redundancy in the data, thereby improving estimates of diversity [39]; ii) act as a common frame of reference across samples and studies; iii) serve as a basis for metagenomic-wide association studies [165]; and iv) guide the binning of metagenomic contigs into organism-specific groups [109, 166].

Such analyses may be confounded by the specific properties of the catalog being used. Yet, to our knowledge, the structure and construction of gene catalogs have not been critically evaluated. Because the processes for constructing and using gene catalogs are broadly the same across studies, generalizable observations can be obtained from the analysis of any of the catalogs referenced above. We focus here on the Integrated Gene Catalog (IGC) [167], which seeks to provide a nearly comprehensive collection of the gene sequences identified in the human gut microbiome. We chose the IGC because it provides all the supporting metadata and intermediate files necessary to conduct a critical analysis of the structure of the resulting clusters.

6.1.1 The construction and use of gene catalogs

Catalog construction starts by identifying genes within metagenomic data. The gene sequences are then clustered together based on similarity in order to remove trivial differences between sequences due to fragmentary data (e.g., genes that miss the start or stop codons), sequencing errors, or small, strain-level variations. The clustering can be performed at the DNA level (e.g., the IGC [167]), or at the amino acid level (e.g., the Global Ocean Survey [39]). Analysis at the DNA level provides greater resolution for taxonomic classification, whereas the amino acid level is better suited for functional analysis and is more able to group together distantly-related but functionally-similar sequences. The implied, but often unstated, goal of the clustering process is to reproducibly group together sequences that have the same function and/or taxonomic origin, thereby defining the gene from which the sequences are derived in a way that is consistent across samples. Each cluster is typically represented by one sequence, either a representative selected from the sequences clustered together, or a sequence that represents the consensus of the clustered sequences. Beyond the obvious use of these sequences in a broad range of sequence-based analyses (e.g., database searches, function or structure prediction), the cluster representatives can also be used to estimate the relative abundance of the corresponding genes within microbiome samples.

6.1.2 Historical context

Clustering of biological sequences that share a common function or taxonomic origin has been at the core of biological research long before the first metagenomic experiment. Databases such as the Clusters of Orthologous Groups (COG) [168] and Pfam [169] date back to the late 1990s and were developed to organize the rapidly accumulating protein sequence information. To define the boundary of clusters, these databases used reciprocal best hit links (COG), or hidden Markov models built upon multiple alignments of related proteins (Pfam), approaches that rely on statistical significance measures instead of arbitrary thresholds based on sequence similarity. At the same time, taxonomic analyses based on housekeeping genes relied on careful phylogenetic analyses to define species boundaries [170].

In the early 2000s, metagenomic studies yielded much larger data sets than previously seen. The challenge of effectively scaling analyses to cope with increasingly larger data sets led to the development of new approaches that emphasized speed over the accuracy or comprehensiveness of the analysis. CD-HIT [38], for example, a greedy clustering approach we briefly describe below, was developed to address the challenges encountered when analyzing the data from the Global Ocean Survey. Although CD-HIT and some other clustering tools developed [79, 171] relied on fixed thresholds to determine the boundaries of clusters, it was already recognized that such thresholds were not consistent with biologically relevant entities [41, 172]; for a given threshold, some clusters contained sequences from multiple species, whereas other species were represented in multiple clusters.

The estimation of abundances from sequencing reads is a relatively new development in metagenomic studies but has been used extensively in the study of gene expression in eukaryotes. A number of factors have been identified that confound abundance estimation including multi-mapped reads, uneven depth of coverage, and sequence composition biases. Computational and statistical approaches have been developed to address such challenges [148, 173, 174, 175].

6.1.3 Overview of the Integrated Gene Catalog

The Integrated Gene Catalog comprises 9,879,896 annotated gene clusters that were constructed from a combination of 511 prokaryotic reference genomes from species known to occur in the human gut, and 1,267 gut metagenome data sets from Chinese, American, and European cohorts. The IGC has been used to discover correlations between gut microbiome composition and resistance to immune checkpoint inhibitors in cancer patients [176], to observe that microbiome composition is modulated to a greater degree by environmental factors than by human genetics [177], to correlate glycemic response after meals with microbiome composition [178], and to identify signs of human fecal contamination in a river with sewage input [179].

The IGC was created through a multistep clustering process [167]. First, separate gene catalogs were created from the metagenomic data derived from each cohort: American (AGC), Chinese (CGC), and European (EGC), and for the sequenced prokaryotic reference genomes collection (SPGC). The three cohort-specific gene catalogs were then clustered together into a larger gene catalog called the 3CGC,

which was then clustered with the SPGC catalog to create the IGC. Gene clustering was performed with CD-HIT [38]. As employed in the construction of the IGC, this tool operates in an iterative fashion, processing the gene sequences in decreasing order of length. The longest gene sequence is selected to be the representative of the first cluster. The next longest sequence is then assigned to the cluster if it matches the representative sequence with $\geq 95\%$ sequence identity over $\geq 90\%$ of the length of the query sequence, or becomes the representative of a new cluster. In the following iterations, query sequences either become representatives of new clusters or are added to an existing cluster if they match the corresponding representative sequence sufficiently well. For most applications, only the set of representative sequences is used, however the IGC project also provides the full assignment of individual genes to clusters. Each representative gene sequence in the IGC is assigned, if possible, taxonomic and functional labels, however, only 16.3% of the sequences are assigned a genus-level annotation and only 60.4% have functional annotations.

6.2 Results

6.2.1 Inconsistent fidelity of clustering

That a 95% sequence identity cut-off is used throughout the multiple rounds of clustering in the construction of the IGC appears to imply that the final clusters are consistent with this threshold. However, the multiple rounds of clustering used to construct the IGC may yield clusters with a (much) lower identity than the intended threshold. We call this methodological artifact transitive clustering error (Figure

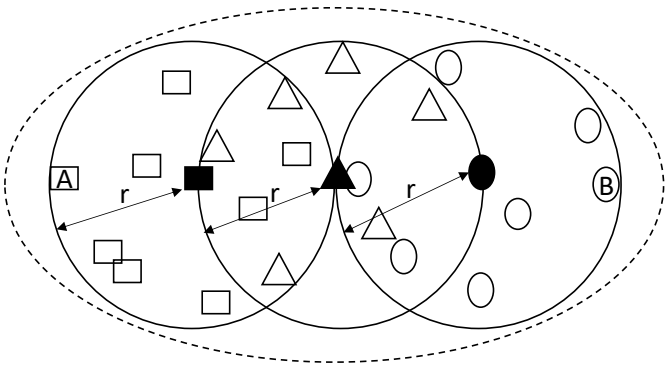

Figure 6.1: The circles represent three clusters from three distinct catalogs. Within each catalog, the sequences within a cluster (represented by points of different shapes) are guaranteed to be within a distance r (5% divergence in the case of the IGC) from the corresponding cluster representative (solid shape). When merging multiple catalogs, only the representative sequences are clustered together (also within the same tolerance r), while the sequences contained within each cluster are implicitly assigned to the same cluster as the corresponding representative. In this figure, after clustering the representatives in one round, the triangle cluster representative is the representative of a meta-cluster (dashed line) that includes the representative sequences of the square and circle clusters. Within this cluster, the maximum distance between two sequences (marked with A and B in the figure), may be as high as 4r, or 20% sequence divergence in the case of the parameters used in the IGC. The distance between a sequence and its corresponding cluster representative may be as high as 2r, or 10% sequence divergence.

6.1), which occurs when different gene catalogs are sequentially clustered. Although each clustering step guarantees the 95% threshold for the sequences being clustered, this threshold does not constrain the similarity between sequences that were clustered in prior iterations. The result of transitive clustering error is an unintended increase in the effective radius of the new cluster with respect to the representative sequence (see Appendix for a detailed explanation of transitive clustering error). When the three cohorts were clustered into the 3CGC, individual gene sequences could potentially share as low as 90% identity to the new representative gene sequences, while two sequences within a cluster may share as little as 80% sequence identity. The final clustering of the SPGC and the 3CGC, could potentially have clustered sequences with only 85% identity to the representative sequence and as low as 70% identity between sequences assigned to the same cluster.

To evaluate the actual impact of transitive clustering error within the IGC, we focused on the 255,191 IGC gene clusters that contained at least 100 sequences each. Among these clusters, 29.6% contained sequences that differed from the cluster representative by more than the intended 95% identity cut-off (Figure 6.2A). Furthermore, 8.2% of the clusters contained sequences that are different by 50% or more from the corresponding cluster representative. This difference is much higher than the expected error due to transitive clustering. An explanation is that the construction of the IGC did not require full length alignments to each cluster representative, but rather allowed matches that cover as little as 90% of the clustered sequence. In the worst case, after two or more rounds of clustering, sequences within an IGC cluster may not overlap with the selected representative sequence at

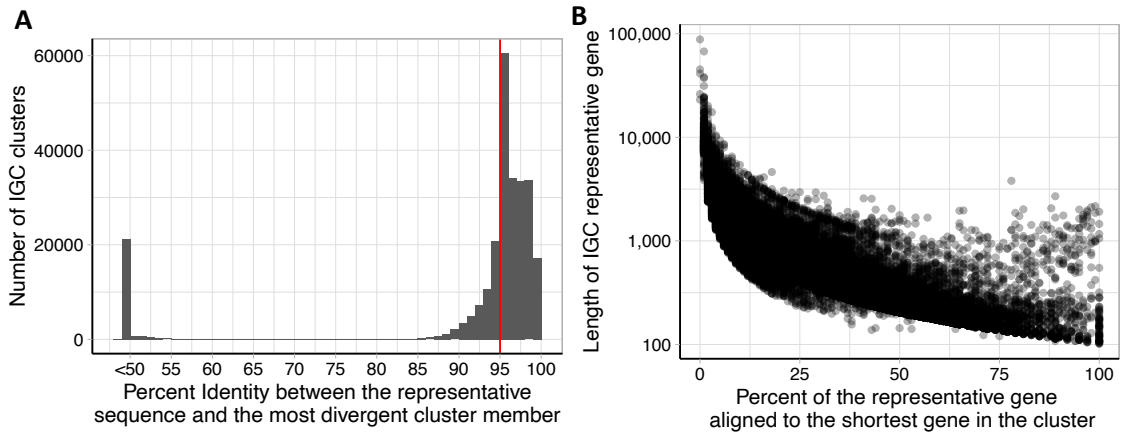

Figure 6.2: Data shown refer to the 255,191 IGC clusters that contain 100 sequences or more. A) The distribution of CD-HIT percent identity between the representative and the most divergent cluster member. The vertical red line indicates the 95% identity clustering threshold used to create the IGC. Note that many sequences are below the target threshold of 95%. B) Relationship of percent of the representative gene aligned to the shortest cluster member and the length of the representative gene.

all (Figure 6.3).

The process used to construct the IGC does not constrain the fraction of the representative sequence that needs to match the sequences within the cluster. This choice makes it possible for two sequences to both align to the cluster representative perfectly without sharing any sequence with each other. As an example, cluster 303 contains four sequences of different lengths –16,111 nt (representative), 7,122 nt, 3,012 nt, and 2,982 nt. All of these genes are complete, spanning from start codon to stop codon, and originate from the SPGC (genes found in nearly-complete reference genomes). The alignment between the three genes to the cluster representative (Figure 6.4) demonstrates the lack of overlap between the individual sequences, suggesting that they align to distinct domains of the representative sequence, rather

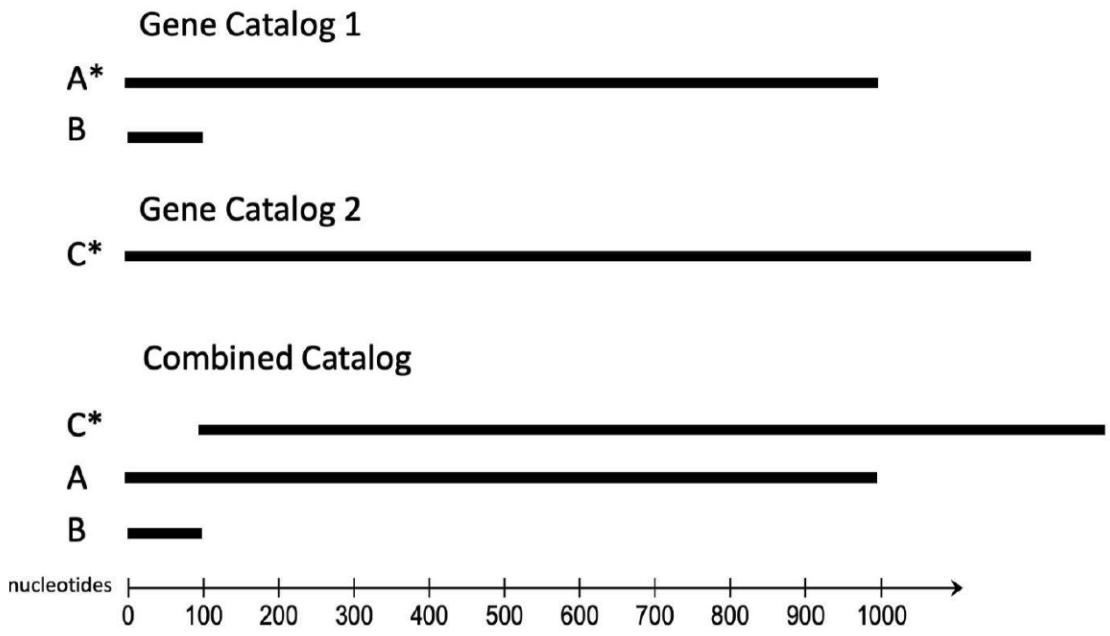

Figure 6.3: A schematic example of how, in a worst-case scenario, clustering separate gene catalogs with CD-HIT can recruit sequences that do not overlap with the representative sequence given the IGC clustering parameters. The sequences within each gene catalog are aligned. Here * denotes the representative sequence of the catalog. Gene A and Gene B were clustered together to create Gene Catalog 1. Gene A is the representative sequence because it is the longest sequence (default of CD-HIT). In this case 100% of the length of Gene B aligns to 10% the length of Gene A with 100% identity. Gene C is a representative sequence in Catalog 2 with no clustered sequences. Gene A and Gene C were clustered to create the Combined Catalog. Gene C becomes the new representative, because it is longer than Gene A, and Gene A and Gene B become cluster members. In the Combined Catalog, 90% of the length of Gene A aligns to Gene C with 100% identity and Gene B has no overlap with Gene C at all.

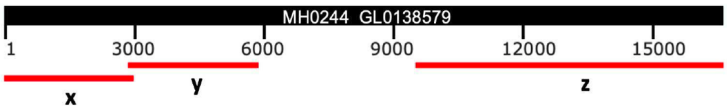

Figure 6.4: BLASTN alignment of the IGC Cluster 303 representative sequence, MH0244_GL0138579 (16,611 nt), and the three cluster members x (469585.HMPREF9007_02027, 2,982 nt), y (469585.HMPREF9007_02028, 3,012 nt), and z (469585.HMPREF9007_02029, 7,122 nt). All were predicted as complete genes (from start to stop codon), yet each cluster member only partially aligns to the representative with a small overlap between x and y and no overlap between y and z.

than representing variants of this gene. This artifact may be wide-spread within the IGC - within the 255,191 IGC clusters with a minimum of 100 members, the mean difference between longest and shortest gene length is 590 nt, representing an average of 14.4% of the length of the cluster representative (Figure 6.2B).

6.2.2 Taxonomic inconsistency of clusters

The 95% identity threshold selected by the IGC was intended to create clusters with taxonomic homogeneity at the species level [167]. Taxonomic homogeneity is desirable for analyses with the IGC, however, as we briefly described above, it has long been recognized that no specific threshold can universally and accurately capture biologically meaningful boundaries [41, 172].

This can be demonstrated by clustering the genes of genera like *Bacteroides* and *Lactobacillus* comprising multiple species within which many strains have been sequenced. We separately clustered the RefSeq genes from 167 *Bacteroides* (5,355,696 genes) and 166 *Lactobacillus* species (1,876,284 genes) using the IGC clustering pa-

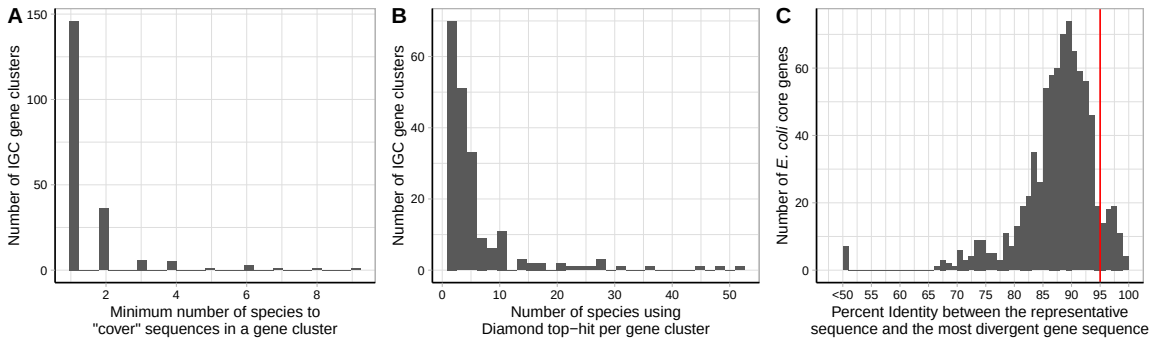

Figure 6.5: A and B show the taxonomic heterogeneity of 200 IGC clusters (the 100 largest clusters and 100 randomly chosen clusters with at least 100 sequences). A) The minimum number of species such that each sequence in a cluster had at least one significant Diamond hit to one of these species B) Number of species per cluster if each sequence is assigned the label of the top Diamond result. C) The distribution of CD-HIT percent identity between the representative and the most divergent gene sequence for the 818 core genes of *Escherichia coli* identified from 86,830 assemblies. The red vertical line denotes 95% identity, the IGC clustering threshold.

rameters. Of the resulting 438,106 *Bacteroides* and 256,949 *Lactobacillus* clusters, 32% and 24% were composed of multiple species, respectively.

The taxonomic homogeneity of the IGC clusters can be most readily assessed within the SPGC because this gene catalog has well-defined taxonomic labels; however, we note that the SPGC only contains 200 species with sparse representation per species (a mean of 2.6 reference genomes). Still, we found that 42,208 (6.4%) of all clusters in the SPGC grouped together sequences from multiple distinct species, with a maximum of 21 species in a single cluster.

To estimate the number of species within the IGC clusters derived from sequences with unknown taxonomic origin (namely, the three country-specific catalogs), we focused on a subset of 200 IGC clusters: the 100 largest clusters and 100

randomly chosen clusters from those with at least 100 sequences each. We aligned each sequence within an IGC cluster to the NCBI nr database (version 5) using Diamond [180] (version 0.9.29). We used the same alignment thresholds as those used by the IGC, requiring at least 95% sequence identity and 90% query coverage. We retained all database entries that matched each query sequence within these thresholds. We conservatively inferred the number of species per gene cluster using a minimum set cover approach. Specifically, we identified the smallest number of species such that each sequence had at least one hit to a database sequence from one of these species. As seen in Figure 6.5A, 73% of clusters (57% of the largest and 89% of the randomly-selected clusters) are covered by a single species. If we used just the top database hit for each sequence, the most commonly-used approach in practice, only 20.5% of clusters (5% of the largest and 36% of the randomly-selected clusters) were composed of a single species (Figure 6.5B).

To explore the converse, the possibility that variants of a gene from a single species may be distributed across multiple clusters, we analyzed a collection of 86,830 *Escherichia coli* genomes obtained from the GenomeTrakr database [181]. When focusing on just the 818 core genes of the *E. coli* pan-genome (genes found in all of the genomes), the mean sequence identity between the representative and the most divergent clustered sequence was 87.7% which is lower than the 95% threshold used by the IGC. In fact, only 63 core genes met or exceeded the 95% threshold and would have been clustered properly by the IGC (Figure 6.5C).

6.2.3 Hidden species within the IGC

A direct consequence of multi-species clusters is the possibility that genes from an individual species may be "hidden" by representative sequences belonging to a different species. A species for which no gene is selected as a representative for a cluster in the catalog becomes effectively undetectable in the samples being analyzed.

To explore the extent of this problem, we focused on just the SPGC (genes from complete and near complete genomes) because these genes have well defined taxonomic labels. Within the SPGC, the number of representative genes per species ranged from 139 (*Escherichia sp. 1_1_43*) to 28,404 (*Escherichia coli*). We simulated reads from 507 genomes from the same species (or strain, if known) as the SPGC reference genomes, and mapped these reads to the SPGC using Bowtie2 [116]. As expected, the rate of assigning reads to a species was correlated with the number of representative genes for the species (Figure 6.6). A possible confounding factor might be the fraction of reads that map ambiguously to multiple species, however the median fraction of multi-mapped reads was only 3% across species. Only 129 of the 201 species in the SPGC had an assignment rate of 90% or higher, i.e., 90% of the reads originating from these genomes would be assigned a correct species-level taxonomic label. At one extreme, *Escherichia sp. 1_1_43*, had the lowest number of representative genes and the lowest assignment rate at 2%. Despite having a large number of representative genes, *E. coli* only had an assignment rate of 83%, because of the large number of closely related species in the SPGC. All four *Shigella*

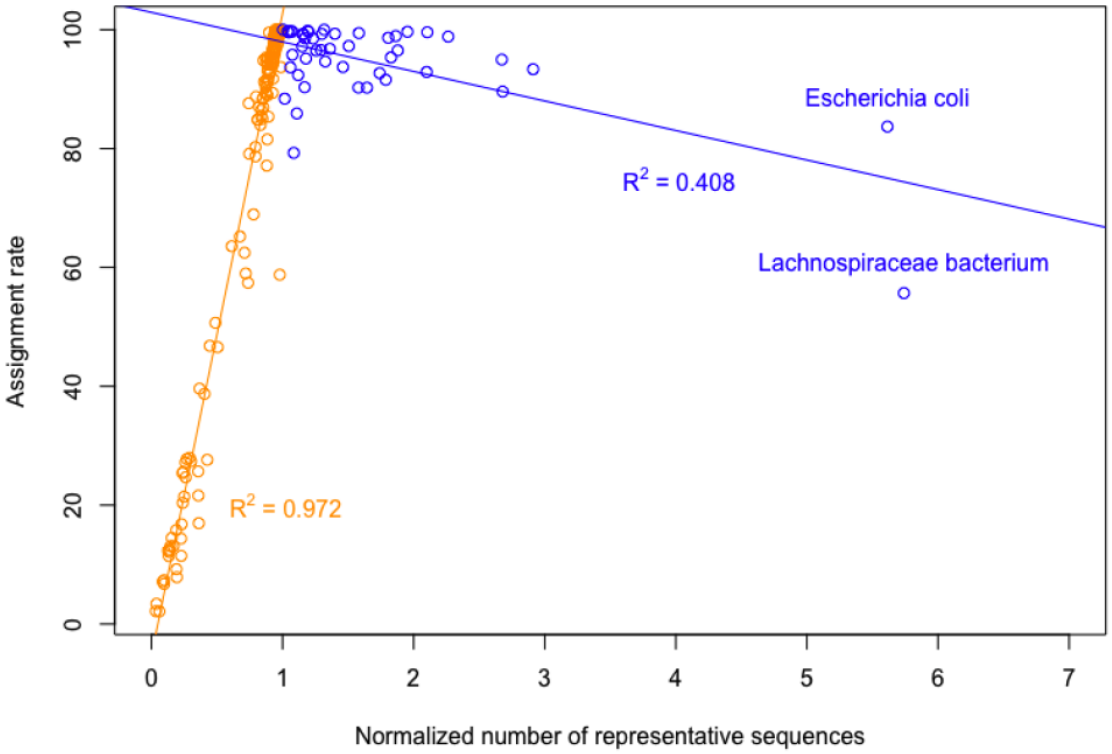

Figure 6.6: The relationship between the number of representative genes (normalized by the mean number of genes per genome) per species and their assignment rate in a simulated metagenomic dataset of the SPGC genomes. The assignment rate is the percent of simulated reads from a species that map to the corresponding representative sequences for that species in the SPGC. For most species in the SPGC, the number of representative genes (normalized by the mean number of genes per genome) is 1 or less (orange). The assignment rate for these species has a positive correlation (orange least squares line) with the number of representatives. For some species, however, the number of representative genes normalized by the mean number of genes per genome can be greater than 1 (blue). These species have genes from multiple genomes and are effectively represented as a pangenome in the SPGC. For example, *E. coli* has 28,404 representative genes and 124 genomes in the SPGC. For these species there is a weak negative correlation (blue least squares line) between the assignment rate and the number of representatives.

Toxin/Virulence factor	Genus of most similar gene in IGC	Percent identity of the top BLAST hit in IGC
ShiA	*Shigella*	100.00
ShiB	*Shigella*	94.41
ShiC	*Shigella*	100.00
ShiD	*Shigella*	100.00
ShiE	*Shigella*	99.43
ShiF	*Shigella*	99.75
ShiG	*Escherichia*	84.44
IucA	*Escherichia*	99.83
IucB	*Escherichia*	99.37
IucC	*Escherichia*	96.38
IucD	*Shigella*	99.78
IutA	*Escherichia*	99.45
Pic	*Shigella*	99.64
GtrA	*Shigella*	99.34
GtrB	*Shigella*	98.15
SigA	*Shigella*	97.67
set1A	Not found	NA
set1B	Not found	NA
Stx1A	*Escherichia*	100.00
Stx1B	*Escherichia*	100.00

Table 6.1: Taxonomic annotation of twenty virulence/toxin genes of *Shigella sonnei* when aligned to the SPGC catalog.

sp. within the SPGC had low assignment rates: 17%, 11%, 8% and 7% for *S. flexneri*, *S. dysenteriae*, *S. boydii*, *S. sonnei*, respectively. This is because the reads from *Shigella sp.* often map to clusters with an *E. coli* representative sequence.

Due to the importance of *Shigella sp.* for human health, we further analyzed 20 known virulence/toxin genes of *S. sonnei* [42-44] (Table 6.1). Only 11 of the 20 genes were taxonomically labelled as *Shigella*, seven were labelled as *Escherichia*, and two, set1A and set1B, were not found at all. Notably, Shiga toxins Stx1A and Stx1B are labelled as *Escherichia*, even though they are part of a mobile prophage genome, which has been horizontally transferred among many *Enterobacteriaceae* [182], highlighting the difficulty of annotating a mobilome.

Read Datasets	BLASTN	Bowtie2	BWA-MEM
Illumina 100 nt	74.31	86.44	96.22
Illumina 250 nt	43.98	76.49	98.97
454 Roche 225 nt (mean)	64.48	77.82	98.18

Table 6.2: The percent of simulated Illumina and 454 Roche reads, from 507 prokaryotic reference genomes, that map to the IGC with BWA-MEM, Bowtie2, and BLASTN. For BLASTN, only those alignments with $\geq$ 95% identity and $\geq$ 90% read coverage are considered. BWA-MEM and Bowtie 2 were run with default parameters requiring full length matches.

6.2.4 Using the IGC as a reference for metagenomic analyses –simulated data

The primary strategy for using the IGC as a reference when analyzing metagenomic data sets involves mapping sequencing reads to the representative sequences of the clusters. Although a seemingly straightforward bioinformatics task, the selection of mapping tools, parameters of the mapping process, and characteristics of the reads themselves (e.g., read length) may have a significant impact on the results. To evaluate the effects of such features on the use of the IGC for metagenomic analysis, we simulated three metagenomic samples composed of the species in the SPGC. Two samples simulated Illumina reads (100 nt, 250 nt), and the other simulated 454/IonTorrent reads (225 nt). We compared mapping statistics for tools that are widely used in metagenomic analyses, BWA-MEM [183] and Bowtie2 [116] with default parameters, and BLASTN [45] with thresholds of 95% identity, 90% read coverage, and default values for all other parameters (Table 6.2).

The fraction of reads mapped by different tools, and across different read lengths, varied substantially (Table 6.3). BLASTN consistently mapped fewer reads

Mapping tool	Dataset	Unmapped reads	Reads mapped exactly once	Multi-mapped reads	Total reads
BLASTN	454 Roche 225 nt	12,046,662 (35.52%)	20,607,264 (60.76%)	1,259,937 (3.72%)	33,913,863
Bowtie2	454 Roche 225 nt	7,523,531 (22.18%)	12,727,602 (37.53%)	13,662,730 (40.29%)	33,913,863
BWA-MEM	454 Roche 225 nt	615,080 (1.81%)	17,789,182 (52.45%)	15,509,601 (45.73%)	33,913,863
BLASTN	Illumina 100 nt	24,590,586 (25.69%)	63,782,504 (66.64%)	7,339,930 (7.67%)	95,713,020
Bowtie2	Illumina 100 nt	12,977,730 (13.56%)	42,142,225 (44.03%)	40,593,065 (42.41%)	95,713,020
BWA-MEM	Illumina 100 nt	3,618,165 (3.78%)	49,637,777 (51.86%)	42,457,078 (44.36%)	95,713,020
BLASTN	Illumina 250 nt	21,407,600 (56.03%)	16,112,369 (42.17%)	690,019 (1.81%)	38,209,988
Bowtie2	Illumina 250 nt	8,984,244 (23.51%)	14,631,373 (38.29%)	14,594,371 (38.20%)	38,209,988
BWA-MEM	Illumina 250 nt	392,069 (1.03%)	20,811,950 (54.47%)	17,005,969 (44.50%)	38,209,988

Table 6.3: Read mapping statistics for different tools (BLASTN, Bowtie2, BWA-MEM) for the reads simulated by ART simulator for 454 Roche technology and Illumina (100 nt, 250 nt) technology. For BLASTN, only those alignments that have $\geq 95\%$ identity and $\geq 90\%$ read coverage are considered.

Read Length	BWA-MEM vs Bowtie2	BWA-MEM vs BLASTN	Bowtie2 vs BLASTN
Illumina 100 nt	2.68×10^{-251}	1.12×10^{-39}	2.45×10^{-25}
Illumina 250 nt	3.27×10^{-07}	0.0	3.84×10^{-123}

Table 6.4: P-values from Mann Whitney U Test comparing the gene abundance profiles generated by different mapping tools when mapping simulated reads, of varying lengths, to the IGC.

Read Length	Percent of reads mapped to IGC	Percent of reads mapped to correct genus
Illumina 100 nt	86.4	81.7
Illumina 250 nt	76.5	82.1

Table 6.5: Read mapping statistics for testing the taxonomic classification performance of the IGC on data simulated from genomes with the same taxonomy as the SPGC reference genomes.

than the other tools. The gene abundance profiles estimated from these mappings differed significantly across different mapping tools (Mann Whitney U test, p-value < 0.001) at every read length, suggesting the choice of mapping tool may confound abundance estimates and, therefore, the associations derived from the data (Table 6.4). Furthermore, nearly half of the reads multi-mapped, i.e., mapped equally well to multiple IGC clusters. Multi-mapped reads can confound taxonomic classification and estimates of abundance, as previously highlighted in RNA-seq studies [184]. Our results suggest the need for abundance estimation algorithms that can account for mapping ambiguity [148, 174, 175], which are rarely used in metagenomic studies.

Together, multi-mapped reads and the poor visibility of some species within the catalog, led to 20% of the reads mapping to gene clusters classified as a different genus than that from which the reads originated (Table 6.5). This raises concerns about the accuracy of taxonomic profiles derived from real metagenomic data given that these reads were generated from the genomes used in the construction of the IGC.

6.2.5 Using the IGC as a reference for metagenomic analyses –real data

In addition to the read mapping artifacts discussed previously, genes that are not represented in the IGC but are present in a sample can confound the analysis of metagenomic data. Prior studies have demonstrated the IGC is not a comprehensive representation of the diversity of the human gut microbiome, lacking many genes found in the gut of infants [50], patients suffering from various diseases such as gout [185] or diabetes [186], adults from India (only 61% of their gene catalog mapped to the IGC) [187], and even adult twins from the UK (in which a putative 1.5 million genes were not present in the IGC) [188].

To investigate how read mapping artifacts and genes not represented in the catalog impact analyses based on the IGC, we used a human gut sample from a 61 year-old Cameroonian male with a hunter gatherer diet (SRA accession ERR2619707) [189]. We assembled the data with MEGAHIT [126] and predicted genes using Prokka [130]. Only 66.6% of the predicted genes from this sample clustered to an IGC gene representative, genes to which we refer as the clustered predicted genes. The other genes predicted from the sample could not be confidently assigned to IGC clusters (and thus are likely not represented in the IGC), and we refer to these genes as the unclustered predicted genes.

We separately mapped the reads from the Cameroon data set with Bowtie2 to the two sets of genes predicted from the sample and the IGC clusters, respectively (Figure 6.7). The percent of reads mapping to the predicted genes and the IGC was

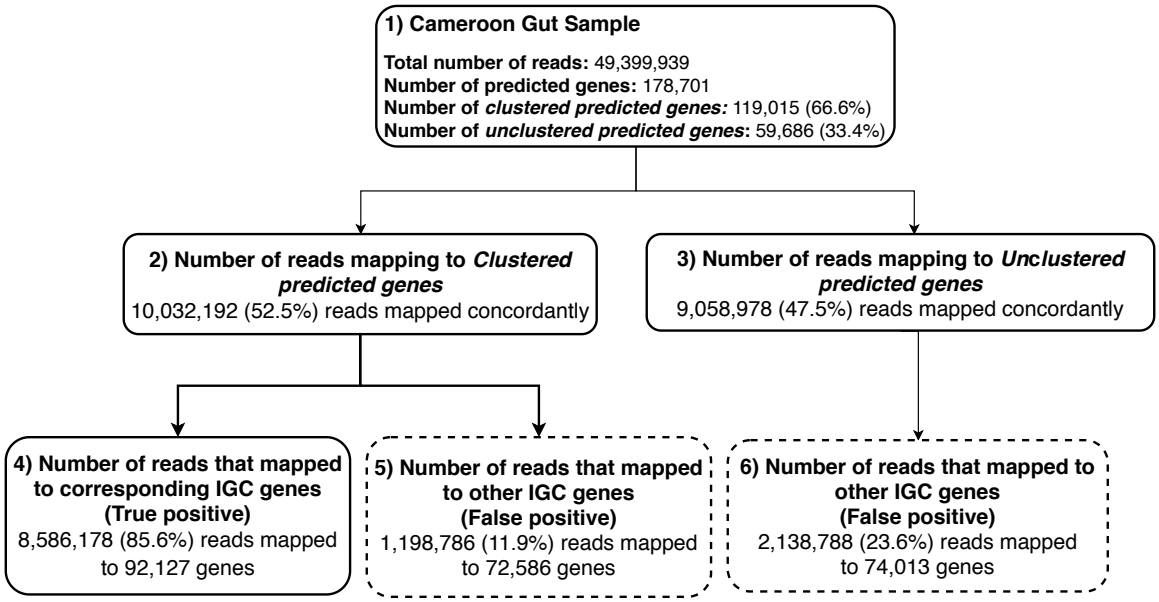

Figure 6.7: Analysis of reads from Cameroonian human gut metagenome sample. Box 1 shows the general statistics of the sample. 66.6% of the predicted genes could be assigned to IGC gene clusters *clustered predicted genes*. 33.4% of the predicted genes could not be confidently mapped to the IGC clusters *unclustered predicted genes*. 19,091,170 reads mapped concordantly to the predicted genes (52.5% to the clustered predicted genes and 47.5% to the unclustered predicted genes). Among the 10,032,192 reads that mapped concordantly to the clustered predicted genes (Box 2), 11.9% mapped to a different IGC gene than expected (false positives denoted by dashed line Box 5). Of the 9,058,978 reads that mapped concordantly to the unclustered predicted genes (Box 3), 23.6% mapped to IGC genes (false positives denoted by dashed line Box 6).

similar (59.0% to the predicted genes and 55.3% to the IGC), but the percent of multi-mapped reads was much higher for the IGC (24.1%) compared to the predicted genes (3.8%). The reads also mapped to an order of magnitude more IGC clusters (1,369,981) than predicted genes (177,745). Together this suggests a high false positive rate, i.e., that reads from unclustered predicted genes are mapping to IGC clusters representing potentially unrelated genomic sequences and/or functions.

To determine the IGC clusters to which the reads from the clustered predicted genes and the unclustered predicted genes were aligned, we focused our analysis on the read pairs that mapped concordantly to both the predicted genes and to the IGC clusters (24.1% of all reads). A read pair is considered concordantly mapped when the forward and reverse reads of the pair map to a gene with the correct insert size and orientation. Such concordant mappings are less likely to represent mapping artifacts. Given that each clustered predicted gene has a corresponding IGC cluster, we would expect the reads mappings to also be shared between the gene and the cluster to which it is related. Among the 10,032,192 reads that concordantly mapped to clustered predicted genes, 11.9% mapped to a different IGC gene than expected. Conversely, we would expect few read pairs which map to the unclustered predicted genes to map to any IGC clusters given that these genes do not share sufficient similarity with any IGC cluster. Of the 9,058,978 reads that concordantly mapped to the unclustered predicted genes, 23.6% mapped to IGC genes (Figure 6.7).

Gene catalog	Year published	Transitive clustering error	Clusters sequences of highly different lengths	Taxonomic inconsistency	Hidden species	Clustering criteria
Human gut (cirrhosis study) [190]	2014	Yes	Yes	Yes	Yes	- Pairwise comparison of all genes with BLAT: > 95% identity and > 90% of the shorter gene length. - Merged genes from three catalogs using the same clustering technique.
Mouse gut [152]	2015	No	Unclear	Yes	Yes	- Pairwise comparison of all genes with BLAT: > 95% identity and overlap > 90%
Human gut (infants) [185]	2015	No	Yes	Yes	Yes	-Pairwise comparison of all genes with BLAT: > 95% identity and > 90% of the shorter gene length.
Human gut (diabetes) [186]	2015	No	Yes	Yes	Yes	- Predicted protein-coding genes with a minimum length of 100 bp were clustered at 95% sequence identity using CD-HIT with parameters set to: -c 0.95, -G 0 -aS 0.9, -g 1, -r 1.
Pig gut [154]	2016	No	Unclear	Yes	Yes	- Pairwise comparison of all genes with BLAT: > 95% identity and overlap > 90%
Human gut (gout) [191]	2016	No	Yes	Yes	Yes	- Genes were clustered with CD-HIT using a sequence identity cut-off of 0.95 and a minimum coverage cut-off of 0.9 for the shorter sequences.
Human gut (diabetes) [188]	2016	Yes	Yes	Yes	Yes	- Genes were clustered using CD-HIT of the MOCAT pipeline (95% identity, 90% overlap). - Merged the gene set with the IGC catalog using CD-HIT.
Chicken gut [159]	2018	Unclear	Yes	Yes	Yes	- Gene catalog was constructed using CD-HIT-EST with parameters set to: -c 0.95 -n 10 -G 0 -aS 0.9
Rat gut [153]	2018	No	Yes	Yes	Yes	- Gene ORFs were clustered using CD-HIT with a criterion of 95% identity > 90% of the shorter ORF length with default parameter except "-G 0 -n 8 -aS 0.9 -c 0.95 -d 0 -g 1".
Dog gut [156]	2018	No	Unclear	Yes	Yes	- Genes were clustered at 95% identity using CD-HIT.
Macaque gut [158]	2018	No	Unclear	Yes	Yes	- Pairwise comparison of all genes using CD-HIT with identity of > 95% and overlap of > 90%
Human gut (children) [192]	2018	Yes	Yes	Yes	Yes	- Clustered gene based on sequence similarity at 95% identity and 90% coverage of the shorter sequence using CD-HIT. - Merged with the IGC using the same CD-HIT clustering technique to form a comprehensive catalog.
South China soil [162]	2019	No	Unclear	Yes	Yes	- Nucleic acids longer than 100bp were translated into amino acid sequences. Pairwise comparison of all genes using CD-HIT with parameters > 95% identity and > 90% overlap.
Pig gut [193]	2019	Yes	Unclear	Yes	Yes	- Predicted genes were clustered at the nucleotide level using CD-HIT with > 95% identity and > 90% overlap. - Combined the catalog with an earlier Pig gut catalog to create a comprehensive catalog.
Human lung [164]	2019	No	Yes	Yes	Yes	- Genes with a length ≥ 100 bp and without Ns (unidentified nucleotides) were selected to construct non-redundant gene sets using CD-HIT with criteria of > 95% identity and > 90% alignment of shorter sequence (-c 0.95 -aS 0.9).
Human gut (Indian cohort) [187]	2019	Yes	Unclear	Yes	Yes	- Pair-wise alignment of genes using BLAT and the genes that had an identity > 95% and alignment coverage > 90% were clustered into a single set of non-redundant genes. - The gene catalog constructed from Indian samples was combined with the IGC to construct a non-redundant gene catalog (using identity ≥ 95% and alignment coverage ≥ 90%).
Rat gut [194]	2019	No	Yes	Yes	Yes	- Predicted ORFs were clustered using CD-HIT with criteria of > 95% identity and > 90% alignment of shorter ORF. (-c 0.95, -G 0, -aS 0.9, -g 1, -d 0).
Panthera gut [160]	2020	No	Yes	Yes	Yes	- Predicted genes from contigs and from the top abundant microbial species were clustered using CD-HIT using a sequence identity cutoff of 0.95 and minimum coverage cutoff of 0.9 for shorter sequences.
Cow gut [157]	2020	No	Yes	Yes	Yes	- Predicted genes were clustered using CD-HIT with ≥ 95% identity and ≥ 90% overlap of the shorter sequence (-n 8 -d 0 -g 1 -T 6 -G 0 -aS 0.9 -c 0.95).
Mouse gut [195]	2020	No	No	Unclear	No	- Predicted ORFs were taxonomically annotated at different levels, i.e., ORF, contig, and bin.
Human dental caries [196]	2020	No	No	Yes	Yes	- Predicted ORFs were clustered using CD-HIT default parameters.
Human vagina [163]	2020	No	Yes	Yes	Yes	- Genes and gene fragments that were at least 99 bp long, with greater than 95% identity over 90% of the shorter gene length were clustered together by CD-HIT-EST.
Rhizosphere soil [197]	2020	No	Yes	Yes	Yes	- Predicted genes were clustered using CD-HIT with > 95% sequence identity.
Sheep rumen [198]	2020	No	Yes	Yes	Yes	- CD-HIT tool with the similarity threshold of 95% was used to remove redundant genes

Table 6.6: A non-exhaustive list of microbial gene catalogs created in the last few years, and the issues —identified in our analysis of the IGC —that likely affect them based upon a review of the written methods. The columns of the table list: 1) the gene catalog; 2) the year it was published; 3) if the clusters are affected by transitive clustering error; 4) if the clusters contain sequences of highly different lengths; 5) if the clusters contain sequences from different species; 6) if species have genes hidden by the genes of other species; 7) the clustering criteria employed to create the catalog.

6.2.6 Analysis of other gene catalogs

A survey of 24 gene catalog studies from the last few years highlights that many were created using a similar clustering algorithm as the IGC and thus likely share many of the same issues as those identified above (Table 6.6). While none of these catalogs provided all the necessary metadata and intermediate files to perform the same analyses as done for the IGC, we were able to predict which issues likely affect the catalogs based upon the description of the methods used to construct these catalogs. We note that 5 of the 24 catalogs were affected by transitive clustering error. Additionally at least 15 catalogs allowed genes of highly divergent lengths to be clustered together. Further, taxonomic inconsistency and hidden species also likely affect 23 of the catalogs.

6.3 Discussion

Gene catalogs help organize the vast volumes of data generated in metagenomic experiments. If carefully constructed, they provide a valuable resource for the analysis of metagenomic samples. Through our analysis of the IGC —one of the largest gene catalogs available to scientists today —we have highlighted how the design and construction of a gene catalog can affect downstream analyses in unintended ways. These issues affected a large percent of the gene catalogs we found in the literature because many were constructed using similar methods as the IGC.

Perhaps the most prevalent and important source of error for gene catalogs is caused by clustering gene sequences with a fixed threshold, creating clusters com-

posed of sequences with variable levels of taxonomic relatedness. Our observation recapitulates the finding that no specific sequence similarity threshold can be used to consistently capture a particular taxonomic level or functional category. This finding has been well documented previously in the context of 16S rRNA sequencing [41, 172]. Clustering in this manner effectively hides the taxonomic origin of all but the gene sequences selected as cluster representatives. As a result, each species in a catalog might have a different proportion of genes that are not represented (that are hidden by the genes of other species), genes that are represented once and genes that are represented in multiple copies. This can introduce bias in downstream analyses that aim to explore the presence or abundance of taxa across samples, a bias already noted in the community [199]. For example, if a catalog contains multiple variants of a gene from a species, metagenomic reads from that gene and species might map to multiple variants in the catalog either uniquely or by multimapping. Through our analysis of the hidden species of the SPGC and the *E. coli* core genes, we have shown that this effect is non-uniform across taxonomic groups and can result in the biased recruitment of reads across taxa.

Another common source of error for gene catalog construction is the clustering of genes of widely different lengths. This can result in clusters where there is little or no overlap between cluster members. While it is not currently possible to confirm the functional consistency of all clusters in a gene catalog, if cluster members share little sequence similarity with the representative (which is treated as the functional homolog of all cluster members) it is likely that they do not share the same function. Furthermore, assessing the relationship between sequence and functional similarity

is non-trivial [200] even in the absence of the confounding information introduced by the co-clustering of sequences with widely-divergent lengths.

The iterative clustering of catalogs can further exacerbate all of the previously mentioned issues by amplifying the differences between sequences assigned to a cluster. Among the gene catalogs we have explored (Table 6.6), the use of a multi-step clustering process is typically used for two purposes: to mitigate computational costs, and/or to update an old catalog by merging it with a newer one. However, none of the studies we analyzed took into account the amount of error introduced by iterative clustering. It is certainly desirable to develop computationally-efficient catalog construction methods as data sets increase in size, as well as to efficiently incorporate new data into existing catalogs. Our analysis, however, suggests that it is important to ensure that the fidelity of the clusters is not impacted by computational convenience, and highlights the need for additional research in this field.

Coupled with the issues arising from the structure of the clusters themselves, we have shown that the use of the IGC to analyze a real metagenomic sample induces many analytical artifacts, including a high false positive rate —IGC clusters that are not actually found in a sample, but which "recruit" many reads nonetheless. Conversely, as the number of species and the number of their gene variants represented in the catalog increases, so will the number of reads that map ambiguously [90]. As a result, using gene catalogs that are constructed similarly to the IGC for metagenomic studies will likely introduce analytical artifacts that outweigh the benefit of the common frame of reference these catalogs provide.

While raising these concerns, we agree with the authors of the IGC that prop-

erly constructed gene catalogs can be an effective reference for metagenomic studies. However, to maximize their usefulness, gene catalogs should either be created directly from the samples being analyzed or from closely related samples. Our findings indicate that the goal of tracking individual clusters across studies is not met by the IGC and other similarly constructed catalogs. We believe that universal taxonomic identifiers and gene ontologies represent a better approach for relating findings across gene catalogs and metagenomic studies. For gene catalogs to be used as global resources for metagenomic data analysis, new methods for updating catalogs and accounting for biases introduced by read mapping tools need to be researched. For now, we believe the best use case for gene catalogs is within the narrow context of the samples used to create them.

Our results highlight pitfalls that need to be avoided when constructing such catalogs and reveal several best practices:

1. The iterative integration of clusters should be avoided as it amplifies the errors inherent to the clustering process. A multi-step clustering process may be necessary to mitigate computational costs, however we recommend limiting the number of rounds and accounting for the growth in cluster diameter that is due to the multi-round process.

2. Arbitrary similarity thresholds should be avoided, and instead researchers should use approaches that are able to dynamically tune clustering parameters [41, 201, 202, 203, 204].

3. The clustering procedure should ensure all sequences within a cluster are of

similar length.

4. The construction of gene catalogs should not exclusively rely on data from metagenomic experiments, but rather should be augmented with genomic sequences from organisms that are commonly found at low abundance in the samples of interest (including eukaryotes and viruses), as such organisms are unlikely to be assembled sufficiently well within the metagenomic data.

5. The alignment of sequences to the catalog, as well as estimation of gene abundances from the alignments, should be conducted in a way that adequately addresses non-specific mapping. Several approaches have been developed for RNA-seq analysis that effectively handle multi-mappings in an alignment-free manner [148, 174, 175], though it remains to be seen whether these are sufficiently effective in metagenomic settings or whether the underlying algorithms need to be adapted.

During the preparation of our manuscript, a new catalog was published [205], which partly addresses some of the issues we have highlighted above. The underlying data being clustered were derived from cultured genome sequences and metagenome-assembled sequences, potentially ensuring a higher quality protein catalog (the Unified Human Gastrointestinal Protein catalog). Gene-level clustering was performed at the protein level in one round of clustering, thereby avoiding transitive clustering error. Notably, the authors of this new study re-clustered the genes from the IGC and appear to be unaware of the blow-up in divergence caused by the iterative process used by the IGC: "We clustered the IGC only at 90% and

50% protein identity, as it was originally de-replicated at 95% nucleotide identity" [205]. The Unified Human Gastrointestinal Protein catalog was provided as multiple catalogs constructed with different similarity thresholds, acknowledging that no threshold is appropriate for all analyses. Some of the pitfalls identified above, however, still apply to the new catalog. When clustering protein sequences, Almeida et al. only control the fraction of the clustered sequence that needs to match the cluster representative (80% in this case), raising the possibility of artifacts such as that highlighted in Figure 6.4. Furthermore, the new catalog includes the Unified Human Gastrointestinal Genome catalog which is constructed in a two-step process to address the computational cost of clustering. The paper does not indicate that the authors are aware of the additional sequence divergence introduced by this process.

A full-fledged analysis of the new catalog, similar to what we have described above, is beyond the scope of this manuscript. However, as discussed here, it is apparent that issues such as those we have described are not widely appreciated in our community. We hope that our manuscript provides readers with an appreciation for the complexity of sequence clustering, particularly as it relates to metagenomic sequence analysis, and leads to a more thoughtful consideration of the pitfalls we have identified when using gene catalogs as a reference for data analysis.

6.4 Appendix

6.4.1 Transitive Clustering error

Clustering a large number of sequences can require impractical amounts of computing time and memory. One technique for addressing the computational cost of clustering uses a divide and conquer paradigm: disjoint subsets of the sequences are clustered separately, then the cluster representatives are clustered together in one or more additional rounds of clustering. When representative sequences from different subsets are clustered together, all members of the corresponding clusters implicitly become part of the resulting cluster. In the IGC, this process was conducted in three rounds. Sequences from each of three distinct geographic regions (American, Chinese, and European) were clustered separately, as were sequences extracted from isolate genomes within the NCBI and EMBL databases, resulting in four distinct catalogs: AGC, CGC, EGC, and SPGC, respectively. In a second round, the three geographically defined catalogs were clustered together, yielding a new catalog, 3CGC, which were then clustered together with the SPGC in a third round of clustering. Each round of clustering used the same cut-offs for the percent identity between a sequence and the cluster representatives, and for the fraction of the sequence that needs to align to the cluster representative in order for it to be assigned to a cluster.

In the following, we discuss the implications of using such an iterative clustering process on the size of the resulting clusters. We focus on two measures of

the "tightness" of clusters: the radius (maximum distance between a sequence and the cluster representative); and the diameter (maximum distance between two sequences within a cluster). For the purpose of this discussion, we ignore the impact of partial alignments between a sequence and the cluster representative, and focus exclusively on percent identity as a measure of distance between sequences.

The percent identity cut-off provided to CD-HIT controls the radius of the clusters. After a single round of clustering, the maximum effective radius, R, of the clusters is exactly the same as the cut-off, r, that was given as a parameter to CD-HIT. The maximum effective diameter, D, is exactly 2r. Below, we will show that, with each round of clustering, both the maximum effective radius and diameter of the resulting clusters increases despite using the same cut-off, r, when clustering the representative sequences of clusters generated in a prior round. We call this unintended increase in the effective radius and diameter of clusters transitive clustering error.

6.4.1.1 A general formulation for transitive clustering error

We will start by assuming a set of clusters already constructed, $C_1, C_2, \cdots, C_n$, which have the effective radii $R_1, R_2, \cdots, R_n$. We explore here the impact of clustering together the representative sequences of the clusters contained in $C_1, C_2, \cdots, C_n$, as defined by the effective radius and diameter of the resulting clusters. Before we proceed, it is important to note that our analysis focuses on the worst-case scenario, i.e., we show that it is possible that at least one of the resulting clusterings can have

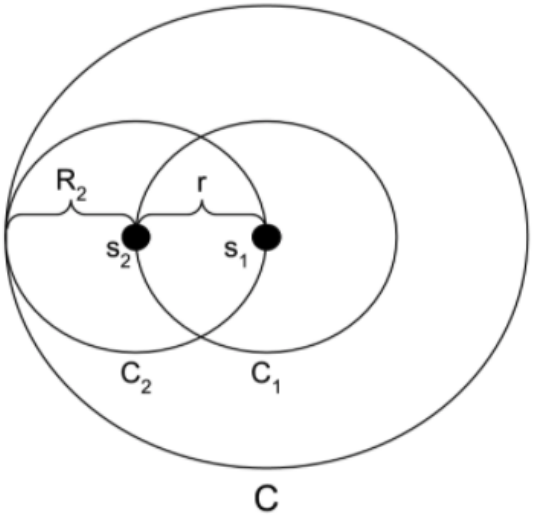

Figure 6.8: An example cluster, C, with the maximum possible effective radius when clustering the representatives from $C_1, C_2, \cdots, C_n$ with tolerance r.

the values for R and D as defined below. Whether such a worst-case situation may occur depends on the characteristics of the data.

Lemma 1. *The maximum effective radius, R, of the resulting clusters is,*

$$R = max(R_1, R_2, \cdots, R_n) + r$$

Proof. Refer to figure 6.8. Without loss of generality, we can assume one resulting cluster, C. By definition, the representative sequence of this cluster must be the representative sequence of one of the clusters in $C_1, C_2, \cdots, C_n$. Without loss of generality we assume that this is the same as the representative sequence s_1 of the cluster C_1. Further, assume that there is another cluster whose representative sequence was clustered together with s_1. This cluster cannot come from the same catalog as C_1 since its representative sequence is within distance r of s_1 and thus would have been clustered with s_1 already, and therefore could not have seeded

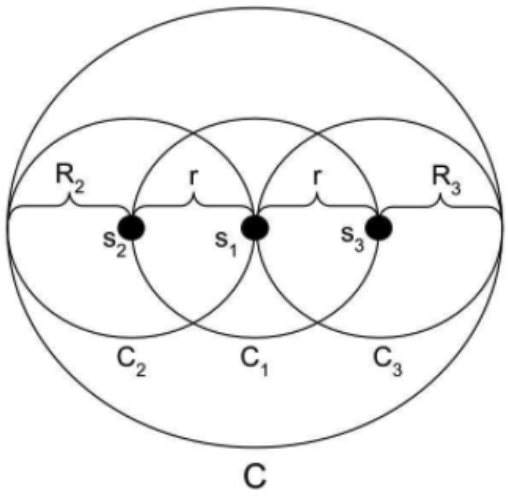

Figure 6.9: An example cluster, C, with the maximum possible effective diameter when clustering the representatives from $C_1, C_2, \cdots, C_n$ with tolerance r.

its own cluster. Without loss of generality, we can assume that the second cluster is C_2 and that its representative sequence is s_2. To define the radius R of the cluster C we need to compute the maximum distance between a sequence within the selected cluster and its representative s_1. Without loss of generality, let us assume that $R_2 = max(R_1, \cdots, R_n)$. Given the above, the maximum distance between a sequence within the cluster defined by s_1 and s_1 is the sum of r, the maximum distance between s_1 and s_2, and R_2, the maximum distance between a sequence within the cluster defined by s_2 and its cluster representative, thereby proving the lemma. □

Lemma 2. *The maximum effective diameter D, is,*

$D = 2 * max(R_1, R_2, \cdots, R_n) + 2r = 2R$

Proof. Refer to figure 6.9. The proof follows the same template as that for the radius,

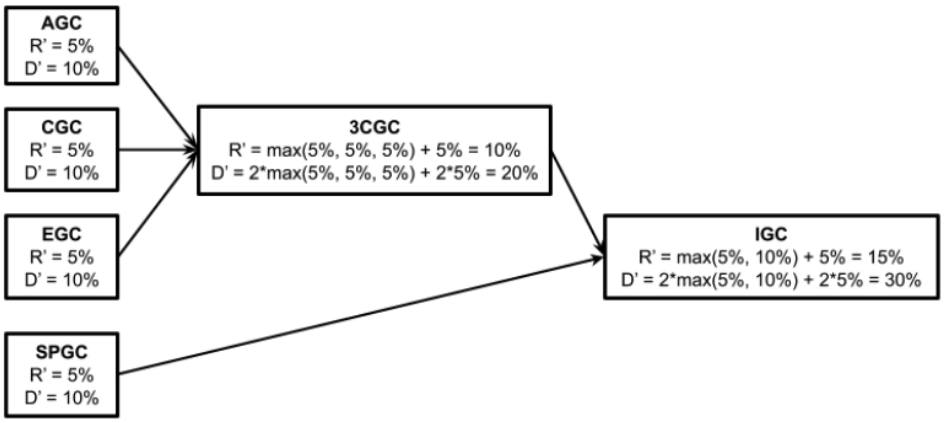

Figure 6.10: Clustering strategy used in creating the IGC. Each block represents catalogs created in the process, and shows the worst-case radius and diameter of clusters in the catalog.

except that the selected cluster, C, is assumed to be clustered with an additional cluster, C_3, where $R_3 = R_2 = max(R_1, \cdots, R_n)$ and the representative of C_3, s_3, is $2r$ divergent from s_2.

Given the two lemmas, we can now explore the impact on R and D of the number of iterative clustering steps. □

6.4.1.2 Transitive clustering error in the IGC

The diagram in figure 6.10 highlights the clustering strategy used by the IGC. At each stage, the clustering cut-off, r, was set to 5% divergence (95% identity). Using the formulas derived above, we demonstrate the increase in effective radius and diameter that occurs at each clustering stage, reaching a maximum radius of 15% for the IGC. In other words, within the IGC it is theoretically possible that a sequence may share as little as 85% identity with the corresponding cluster representative,

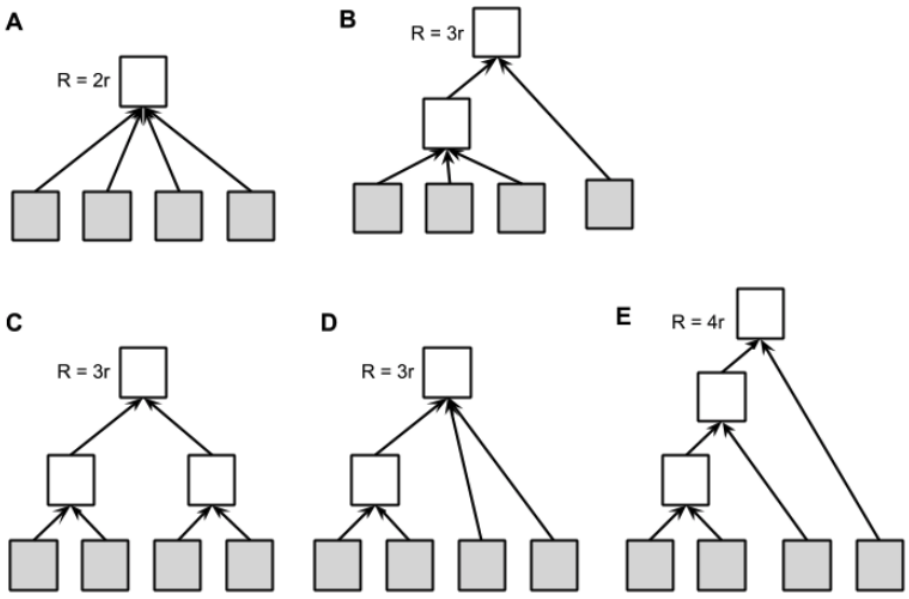

Figure 6.11: Example topologies to combine sequences from four catalogs (grey squares) into one final catalog. The order of combining catalogs can impact the effective radius and diameter of the clusters in the final catalog. For each topology, the effective radius of the final catalog is displayed.

and two sequences that are co-clustered may share as little as 70% identity with each other. These values exceed the nucleotide identity assumed to define a species for the IGC i.e. 95% identity.

6.4.1.3 Impact of clustering strategy on cluster radius

In figure 6.11 we compare five different approaches for constructing a catalog from four different catalogs. These range from a single round of clustering that joins all catalogs together in one round (A), to a four-round process that iteratively adds an additional catalog to the previously clustered ones (E). The process used by the IGC is in panel (B). As can be seen in the figure, the final effective radius ranges

from 2r (A) to 4r (E), demonstrating how different clustering strategies impact the effective radius of clusters in the final catalog.

Chapter 7: Conclusions

In this dissertation, we presented new methods for large-scale metagenomic data analysis. The tools and algorithms will help extract meaningful information from these datasets. We have demonstrated the efficiency and capability of our tools on several real metagenomic datasets.

For taxonomic classification tasks, with BLAST outlier detection and ATLAS, we provide a similar level of accuracy as phylogenetic approaches while retaining computational efficiency. Even though we use the BLAST outlier detection algorithm to identify the top database hits when working with DNA sequences, it can be easily extended to other biomolecular sequences such as RNA and amino acid sequences. ATLAS deviates from traditional classification methods that use "most recent common ancestor" (MRCA) to encompass all the possible annotations of a sequence. We have shown that ATLAS is able to automatically discover taxonomic groupings that are relevant to the interpretation of the data but that do not match pre-defined taxonomic levels. The majority of partitions identified by ATLAS are at the subgenus level, replacing higher-level annotations with specific groups of species. These more precise partitions improve our detection power in determining differential abundance in microbiome association studies. In the abundance estimation

task, we presented TIPP2, an updated version of the original method. TIPP2 outperforms commonly used taxonomic profiling tools, especially when datasets contain genomes that are not closely related to the reference sequences used by these packages. These improvements will enable a more precise characterization of microbial communities, particularly those that contain species that are not well characterized in public databases.

For reference database-dependent tasks such as taxonomic and abundance profiling, it is important to recognize the importance of the choice of the database [90]. Oftentimes, biologists work with custom, environment-specific databases to improve the accuracy of results. Thus, it is important to release code or protocol that can help users create their own databases or update databases when more data is available. Consequently, we have made database construction steps available for both ATLAS and TIPP.

One of the major benefits of metagenomics is that it can sequence previously unknown organisms. Thus, reference independent methods that help recover genomes of understudied organisms from metagenome samples are extremely important. We developed Binnacle to explore how to best reconstruct genomes from the sample. We show that combining scaffolding and binning steps together improves the contiguity and quality of the resulting bins. Moreover, our experiments show that by using variation-aware scaffolds for binning, the resulting bins contain a better representation of the genic content of the organisms.

It is important to note that the bins (clusters) generated by Binnacle can be of variable resolution. Even though one would like to obtain all bins as species-level

metagenome assembled genomes, this goal is rarely achieved in practice. An interesting future direction would be to resolve the multiple strains/haplotypes represented in the bins. Ideas from haplotype phasing [141, 142], viral quasi-species estimation [143, 144, 145], and species estimation in metagenomics [146] can be applied here to estimate the number of species in a bin, and to split (refine) bins into multiple metagenome assembled genomes (MAGs).

We would also like to highlight the importance of effective visualization tools that can provide researchers with more information about the relative placement of contigs within a bin along a chromosome as well as variation information. This is particularly important for identifying mis-assemblies and polishing MAGs. Few tools such as Bandage [147] and MetagenomeScope [136] exist, but there are still opportunities for future research to create tools that combine contig placements, coverage, and annotation information together and can scale to large metagenomic datasets.

Last, our assessment of the Integrated Gene Catalog reveals several pitfalls that need to be avoided when constructing such catalogs. We recommend best practices should one need to construct such a catalog. Current catalogs do not meet the goal of tracking individual clusters across studies. We believe that universal taxonomic identifiers and gene ontologies represent a better approach for relating findings across gene catalogs and metagenomic studies. For gene catalogs to be used as global resources for metagenomic data analysis, new methods for updating catalogs and accounting for biases introduced by read mapping tools need to be researched. Until then, the best use case for gene catalogs may be to analyze the

samples used to create them.

However, we would like to argue that organizing metagenomic data in a way that can provide useful information to others would be extremely beneficial. Right now, we work with genes because they are easy to identify based on their well-defined boundaries, but it would be interesting to see whether non-geneic regions contain important functions and how to effectively identify such regions. Moreover, clustering large sets of sequences efficiently is still an open computational problem and an important future research direction.

With rapidly growing metagenome sequencing studies, we believe our ideas and methods will be a useful resource for the metagenomics community.

Bibliography

[1] Jack A Gilbert, Folker Meyer, Janet Jansson, Jeff Gordon, Norman Pace, James Tiedje, Ruth Ley, Noah Fierer, Dawn Field, Nikos Kyrpides, et al. The Earth Microbiome Project: Meeting report of the "1 st EMP meeting on sample selection and acquisition" at Argonne National Laboratory October 6 2010. *Standards in Genomic Sciences*, 3(3):249–253, 2010.

[2] Jason Lloyd-Price, Anup Mahurkar, Gholamali Rahnavard, Jonathan Crabtree, Joshua Orvis, A Brantley Hall, Arthur Brady, Heather H Creasy, Carrie McCracken, Michelle G Giglio, et al. Strains, functions and dynamics in the expanded Human Microbiome Project. *Nature*, 550(7674):61–66, 2017.

[3] David Zeevi, Tal Korem, Anastasia Godneva, Noam Bar, Alexander Kurilshikov, Maya Lotan-Pompan, Adina Weinberger, Jingyuan Fu, Cisca Wijmenga, Alexandra Zhernakova, et al. Structural variation in the gut microbiome associates with host health. *Nature*, 568(7750):43–48, 2019.

[4] Susannah Green Tringe and Edward M Rubin. Metagenomics: DNA sequencing of environmental samples. *Nature Reviews Genetics*, 6(11):805–814, 2005.

[5] James T Staley and Allan Konopka. Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats. *Annual Review of Microbiology*, 39(1):321–346, 1985.

[6] Hilary P Browne, Samuel C Forster, Blessing O Anonye, Nitin Kumar, B Anne Neville, Mark D Stares, David Goulding, and Trevor D Lawley. Culturing of 'unculturable' human microbiota reveals novel taxa and extensive sporulation. *Nature*, 533(7604):543–546, 2016.

[7] Jo Handelsman. Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and Molecular Biology Reviews*, 68(4):669–685, 2004.

[8] Joel A Klappenbach, Paul R Saxman, James R Cole, and Thomas M Schmidt. rrndb: the Ribosomal RNA Operon Copy Number Database. *Nucleic Acids Research*, 29(1):181–184, 2001.

[9] Anna Engelbrektson, Victor Kunin, Kelly C Wrighton, Natasha Zvenigorodsky, Feng Chen, Howard Ochman, and Philip Hugenholtz. Experimental factors affecting PCR-based estimates of microbial species richness and evenness. *The ISME journal*, 4(5):642–647, 2010.

[10] Rashmi Sinha, Galeb Abu-Ali, Emily Vogtmann, Anthony A Fodor, Boyu Ren, Amnon Amir, Emma Schwager, Jonathan Crabtree, Siyuan Ma, Christian C Abnet, et al. Assessment of variation in microbial community amplicon sequencing by the Microbiome Quality Control (MBQC) project consortium. *Nature Biotechnology*, 35(11):1077, 2017.

[11] Marcus J Claesson, Qiong Wang, Orla O'Sullivan, Rachel Greene-Diniz, James R Cole, R Paul Ross, and Paul W O'Toole. Comparison of two next-generation sequencing technologies for resolving highly complex microbiota composition using tandem variable 16S rRNA gene regions. *Nucleic Acids Research*, 38(22):e200–e200, 2010.

[12] Dirk Gevers, Frederick M Cohan, Jeffrey G Lawrence, Brian G Spratt, Tom Coenye, Edward J Feil, Erko Stackebrandt, Yves Van de Peer, Peter Vandamme, Fabiano L Thompson, et al. Re-evaluating prokaryotic species. *Nature Reviews Microbiology*, 3(9):733–739, 2005.

[13] Susan M Huse, Les Dethlefsen, Julie A Huber, David Mark Welch, David A Relman, and Mitchell L Sogin. Exploring Microbial Diversity and Taxonomy Using SSU rRNA Hypervariable Tag Sequencing. *PLOS Genetics*, 4(11): e1000255, 2008.

[14] H C Godfray. Towards taxonomy's 'glorious revolution'. *Nature*, 420(6915): 461–461, 2002.

[15] Ziheng Yang and Bruce Rannala. Molecular phylogenetics: principles and practice. *Nature Reviews Genetics*, 13(5):303–314, 2012.

[16] Wei Zhang and Zhirong Sun. Random local neighbor joining: a new method for reconstructing phylogenetic trees. *Molecular Phylogenetics and Evolution*, 47(1):117–128, 2008.

[17] Walter M Fitch. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*, 20(4):406–416, 1971.

[18] Koichiro Tamura, Daniel Peterson, Nicholas Peterson, Glen Stecher, Masatoshi Nei, and Sudhir Kumar. MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Molecular Biology and Evolution*, 28(10):2731–2739, 2011.

[19] Alexandros Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[20] Jennifer Lu, Florian P Breitwieser, Peter Thielen, and Steven L Salzberg. Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science*, 3:e104, 2017.

[21] Alessio Milanese, Daniel R Mende, Lucas Paoli, Guillem Salazar, Hans-Joachim Ruscheweyh, Miguelangel Cuenca, Pascal Hingamp, Renato Alves, Paul I Costea, Luis Pedro Coelho, et al. Microbial abundance, activity and population genomic profiling with mOTUs2. *Nature Communications*, 10(1): 1–11, 2019.

[22] Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, 9(8): 811–814, 2012.

[23] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R Mende, Gabriel R Fernandes, Julien Tap, Thomas Bruls, Jean-Michel Batto, et al. Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180, 2011.

[24] Bo Liu, Theodore Gibbons, Mohammad Ghodsi, Todd Treangen, and Mihai Pop. Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. *Genome Biology*, 12(1):1–27, 2011.

[25] Nam-phuong Nguyen, Siavash Mirarab, Bo Liu, Mihai Pop, and Tandy Warnow. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics*, 30(24):3548–3555, 2014.

[26] Shinichi Sunagawa, Daniel R Mende, Georg Zeller, Fernando Izquierdo-Carrasco, Simon A Berger, Jens Roat Kultima, Luis Pedro Coelho, Manimozhiyan Arumugam, Julien Tap, Henrik Bjørn Nielsen, et al. Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods*, 10(12):1196–1199, 2013.

[27] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12(10):902–903, 2015.

[28] Song Gao, Wing-Kin Sung, and Niranjan Nagarajan. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. *Journal of Computational Biology*, 18(11):1681–1691, 2011.

[29] Sergey Koren, Todd J Treangen, and Mihai Pop. Bambus 2: scaffolding metagenomes. *Bioinformatics*, 27(21):2964–2971, 2011.

[30] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome Research*, 27 (5):824–834, 2017.

[31] Jay Ghurye, Todd Treangen, Marcus Fedarko, W Judson Hervey, and Mihai Pop. MetaCarvel: linking assembly graph motifs to biological variants. *Genome Biology*, 20(1):1–14, 2019.

[32] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, 7(1):1–9, 2016.

[33] FA Bastiaan von Meijenfeldt, Ksenia Arkhipova, Diego D Cambuy, Felipe H Coutinho, and Bas E Dutilh. Robust taxonomic classification of uncharted microbial sequences and bins with CAT and BAT. *Genome Biology*, 20(1): 1–14, 2019.

[34] Derrick E Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with Kraken 2. *Genome Biology*, 20(1):1–13, 2019.

[35] Gene W Tyson, Jarrod Chapman, Philip Hugenholtz, Eric E Allen, Rachna J Ram, Paul M Richardson, Victor V Solovyev, Edward M Rubin, Daniel S Rokhsar, and Jillian F Banfield. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, 428(6978):37–43, 2004.

[36] Mads Albertsen, Philip Hugenholtz, Adam Skarshewski, Kåre L Nielsen, Gene W Tyson, and Per H Nielsen. Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nature Biotechnology*, 31(6):533–538, 2013.

[37] Sergio Arredondo-Alonso, Rob J Willems, Willem Van Schaik, and Anita C Schürch. On the (im) possibility of reconstructing plasmids from whole-genome short-read sequencing data. *Microbial Genomics*, 3(10), 2017.

[38] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22 (13):1658–1659, 2006.

[39] Shibu Yooseph, Granger Sutton, Douglas B Rusch, Aaron L Halpern, Shannon J Williamson, Karin Remington, Jonathan A Eisen, Karla B Heidelberg, Gerard Manning, Weizhong Li, et al. The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families. *PLOS Biology*, 5(3): e16, 2007.

[40] Junjie Qin, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, 2010.

[41] Nidhi Shah, Stephen F Altschul, and Mihai Pop. Outlier detection in BLAST hits. *Algorithms for Molecular Biology*, 13(1):1–9, 2018.

[42] Nidhi Shah, Stephen Altschul, and Mihai Pop. Outlier detection in BLAST hits. In *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[43] Susannah G Tringe and Philip Hugenholtz. A renaissance for the pioneering 16S rRNA gene. *Current Opinion in Microbiology*, 11(5):442–446, 2008.

[44] Frederick A Matsen, Robin B Kodner, and E Virginia Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, 11(1):1–16, 2010.

[45] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215 (3):403–410, 1990.

[46] Liisa B Koski and G Brian Golding. The closest BLAST hit is often not the nearest neighbor. *Journal of Molecular Evolution*, 52(6):540–542, 2001.

[47] Raúl Y Tito, Simone Macmil, Graham Wiley, Fares Najar, Lauren Cleeland, Chunmei Qu, Ping Wang, Frederic Romagne, Sylvain Leonard, Agustín Jiménez Ruiz, et al. Phylotyping and functional analysis of two ancient human microbiomes. *PLOS ONE*, 3(11):e3703, 2008.

[48] Susannah Green Tringe, Christian von Mering, Arthur Kobayashi, Asaf A. Salamov, Kevin Chen, Hwai W. Chang, Mircea Podar, Jay M. Short, Eric J. Mathur, John C. Detter, Peer Bork, Philip Hugenholtz, and Edward M. Rubin. Comparative metagenomics of microbial communities. *Science*, 308(5721): 554–557, 2005. ISSN 0036-8075. doi: 10.1126/science.1107851.

[49] Mihai Pop, Alan W Walker, Joseph Paulson, Brianna Lindsay, Martin Antonio, M Anowar Hossain, Joseph Oundo, Boubou Tamboura, Volker Mai, Irina Astrovskaya, et al. Diarrhea in young children from low-income countries leads to large-scale alterations in intestinal microbiota composition. *Genome Biology*, 15(6):1–12, 2014.

[50] Daniel H Huson, Alexander F Auch, Ji Qi, and Stephan C Schuster. MEGAN analysis of metagenomic data. *Genome Research*, 17(3):377–386, 2007.

[51] Michio Murata, Jane S Richardson, and Joel L Sussman. Simultaneous comparison of three protein sequences. *Proceedings of the National Academy of Sciences*, 82(10):3073–3077, 1985.

[52] Thomas D Schneider, Gary D Stormo, Larry Gold, and Andrzej Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188(3):415–431, 1986.

[53] David Sankoff. Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28(1):35–42, 1975.

[54] David Sankoff and Robert J Cedergren. Simultaneous comparison of three or more sequences related by a tree. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison/edited by David Sankoff and Joseph B. Krustal*, 1983.

[55] Stephen F Altschul, John C Wootton, Elena Zaslavsky, and Yi-Kuo Yu. The construction and use of log-odds substitution scores for multiple sequence alignment. *PLoS Comput Biol*, 6(7):e1000852, 2010.

[56] Michael Brown, Richard Hughey, Anders Krogh, I Saira Mian, Kimmen Sjölander, and David Haussler. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In *International Conference on Intelligent Systems for Molecular Biology*, volume 1, pages 47–55, 1993.

[57] Samuel Karlin and Stephen F Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264–2268, 1990.

[58] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[59] James R Cole, Qiong Wang, Jordan A Fish, Benli Chai, Donna M McGarrell, Yanni Sun, C Titus Brown, Andrea Porras-Alfaro, Cheryl R Kuske, and James M Tiedje. Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic acids research*, 42(D1):D633–D642, 2014.

[60] Martin Hartmann, Charles G Howes, Kessy Abarenkov, William W Mohn, and R Henrik Nilsson. V-Xtractor: an open-source, high-throughput software tool to identify and extract hypervariable regions of small subunit (16 S/18 S) ribosomal RNA gene sequences. *Journal of Microbiological Methods*, 83(2): 250–253, 2010.

[61] Qiong Wang, George M Garrity, James M Tiedje, and James R Cole. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, 73(16):5261–5267, 2007.

[62] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Pena, Julia K Goodrich, Jeffrey I Gordon, et al. QIIME allows analysis of high-throughput community sequencing data. *Nature methods*, 7(5):335–336, 2010.

[63] Ivica Letunic and Peer Bork. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. *Bioinformatics*, 23(1):127–128, 2007.

[64] Seok-Hwan Yoon, Sung-Min Ha, Soonjae Kwon, Jeongmin Lim, Yeseul Kim, Hyungseok Seo, and Jongsik Chun. Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies. *International Journal of Systematic and Evolutionary Microbiology*, 67 (5):1613, 2017.

[65] Christian Quast, Elmar Pruesse, Pelin Yilmaz, Jan Gerken, Timmy Schweer, Pablo Yarza, Jörg Peplies, and Frank Oliver Glöckner. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1):D590–D596, 2012.

[66] Todd Z DeSantis, Philip Hugenholtz, Neils Larsen, Mark Rojas, Eoin L Brodie, Keith Keller, Thomas Huber, Daniel Dalevi, Ping Hu, and Gary L Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

[67] Alexey M Kozlov, Jiajie Zhang, Pelin Yilmaz, Frank Oliver Glöckner, and Alexandros Stamatakis. Phylogeny-aware identification and correction of taxonomically mislabeled sequences. *Nucleic Acids Research*, 44(11):5022–5033, 2016.

[68] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, 2009.

[69] Giovanna E Felis and Franco Dellaglio. Taxonomy of lactobacilli and bifidobacteria. *Current Issues in Intestinal Microbiology*, 8(2):44, 2007.

[70] Elisa Salvetti, Sandra Torriani, and Giovanna E Felis. The genus Lactobacillus: a taxonomic update. *Probiotics and Antimicrobial Proteins*, 4(4):217–226, 2012.

[71] Nidhi Shah, Jacquelyn S Meisel, and Mihai Pop. Embracing ambiguity in the taxonomic classification of microbiome sequencing data. *Frontiers in Genetics*, 10:1022, 2019.

[72] Jennifer J Barb, Andrew J Oler, Hyung-Suk Kim, Natalia Chalmers, Gwenyth R Wallen, Ann Cashion, Peter J Munson, and Nancy J Ames. Development of an analysis pipeline characterizing multiple hypervariable regions of 16S rRNA using mock samples. *PLOS ONE*, 11(2):e0148047, 2016.

[73] Alan W Walker, Sylvia H Duncan, Petra Louis, and Harry J Flint. Phylogeny, culturing, and metagenomics of the human gut microbiota. *Trends in Microbiology*, 22(5):267–274, 2014.

[74] Rachid Ounit, Steve Wanamaker, Timothy J Close, and Stefano Lonardi. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, 16(1):1–13, 2015.

[75] Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):1–12, 2014.

[76] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[77] Robert C Edgar. Accuracy of taxonomy prediction for 16S rRNA and fungal ITS sequences. *PeerJ*, 6:e4652, 2018.

[78] Patrick D Schloss, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, Brian B Oakley, Donovan H Parks, Courtney J Robinson, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75 (23):7537–7541, 2009.

[79] Robert C Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[80] Evguenia Kopylova, Laurent Noé, and Hélène Touzet. SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics*, 28(24):3211–3217, 2012.

[81] Joseph N Paulson, O Colin Stine, Héctor Corrada Bravo, and Mihai Pop. Differential abundance analysis for microbial marker-gene surveys. *Nature Methods*, 10(12):1200–1202, 2013.

[82] Silas Kieser, Shafiqul A Sarker, Olga Sakwinska, Francis Foata, Shamima Sultana, Zeenat Khan, Shoheb Islam, Nadine Porta, Séverine Combremont, Bertrand Betrisey, et al. Bangladeshi children with acute diarrhoea show faecal microbiomes with increased Streptococcus abundance, irrespective of diarrhoea aetiology. *Environmental Microbiology*, 20(6):2256–2269, 2018.

[83] MD Collins, PA Lawson, A Willems, JJ Cordoba, J Fernandez-Garayzabal, P Garcia, J Cai, H Hippe, and JAE Farrow. The phylogeny of the genus Clostridium: proposal of five new genera and eleven new species combinations. *International Journal of Systematic and Evolutionary Microbiology*, 44 (4):812–826, 1994.

[84] Yang Liu, Qiliang Lai, Markus Göker, Jan P Meier-Kolthoff, Meng Wang, Yamin Sun, Lei Wang, and Zongze Shao. Genomic insights into the taxonomic status of the Bacillus cereus group. *Scientific Reports*, 5(1):1–11, 2015.

[85] David A Rasko, Michael R Altherr, Cliff S Han, and Jacques Ravel. Genomics of the Bacillus cereus group of organisms. *FEMS Microbiology Reviews*, 29(2): 303–329, 2005.

[86] Vaibhav Bhandari, Nadia Z Ahmod, Haroun N Shah, and Radhey S Gupta. Molecular signatures for Bacillus species: demarcation of the Bacillus subtilis and Bacillus cereus clades in molecular terms and proposal to limit the placement of new species into the genus Bacillus. *International Journal of Systematic and Evolutionary Microbiology*, 63(7):2712–2726, 2013.

[87] J L Johnson and Barbara S Francis. Taxonomy of the Clostridia: Ribosomal Ribonucleic Acid Homologies among the Species. *Microbiology*, 88(2):229–244, 1975.

[88] Loris R Lopetuso, Franco Scaldaferri, Valentina Petito, and Antonio Gasbarrini. Commensal Clostridia: leading players in the maintenance of gut homeostasis. *Gut Pathogens*, 5(1):1–8, 2013.

[89] Soumitesh Chakravorty, Danica Helb, Michele Burday, Nancy Connell, and David Alland. A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria. *Journal of Microbiological Methods*, 69 (2):330–339, 2007.

[90] Daniel J Nasko, Sergey Koren, Adam M Phillippy, and Todd J Treangen. RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. *Genome Biology*, 19(1):1–10, 2018.

[91] Nidhi Shah, Erin K Molloy, Mihai Pop, and Tandy Warnow. TIPP2: metagenomic taxonomic profiling using phylogenetic markers. *Bioinformatics*, 2021.

[92] Rolf Daniel. The metagenomics of soil. *Nature Reviews Microbiology*, 3(6): 470–478, 2005.

[93] Matthias Hess, Alexander Sczyrba, Rob Egan, Tae-Wan Kim, Harshal Chokhawala, Gary Schroth, Shujun Luo, Douglas S Clark, Feng Chen, Tao Zhang, et al. Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. *Science*, 331(6016):463–467, 2011.

[94] Rachel Mackelprang, Mark P Waldrop, Kristen M DeAngelis, Maude M David, Krystle L Chavarria, Steven J Blazewicz, Edward M Rubin, and Janet K Jansson. Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature*, 480(7377):368–371, 2011.

[95] Kenneth H Nealson and J Craig Venter. Metagenomics and the global ocean survey: what's in it for us, and why should we care? *The ISME journal*, 1(3): 185–187, 2007.

[96] Sean R. Eddy. Profile hidden Markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763, 1998.

[97] Siavash Mirarab, Nam Nguyen, and Tandy Warnow. SEPP: SATé-enabled phylogenetic placement. In *Biocomputing 2012*, pages 247–258. World Scientific, 2012.

[98] Daniel R Mende, Shinichi Sunagawa, Georg Zeller, and Peer Bork. Accurate and universal delineation of prokaryotic species. *Nature Methods*, 10(9):881–884, 2013.

[99] Siavash Mirarab, Nam Nguyen, Sheng Guo, Li-San Wang, Junhyong Kim, and Tandy Warnow. PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *Journal of Computational Biology*, 22(5): 377–386, 2015.

[100] Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

[101] Nidhi Shah, Michael G Nute, Tandy Warnow, and Mihai Pop. Misunderstood parameter of NCBI BLAST impacts the correctness of bioinformatics workflows. *Bioinformatics*, 35(9):1613–1614, 2019.

[102] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[103] C Radhakrishna Rao. A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance. *Qüestiió: quaderns d'estadística i investigació operativa*, 1995.

[104] Nam-phuong Nguyen, Michael Nute, Siavash Mirarab, and Tandy Warnow. HIPPI: highly accurate protein family classification with ensembles of HMMs. *BMC Genomics*, 17(10):89–100, 2016.

[105] D Nguyen Nam-phuong, Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. Ultra-large alignments using phylogeny-aware profiles. *Genome Biology*, 16(1):1–15, 2015.

[106] Harihara Subrahmaniam Muralidharan, Nidhi Shah, Jacquelyn S Meisel, and Mihai Pop. Binnacle: Using Scaffolds to Improve the Contiguity and Quality of Metagenomic Bins. *Frontiers in Microbiology*, 12:346, 2021.

[107] Gherman Uritskiy and Jocelyne DiRuggiero. Applying genome-resolved metagenomics to deconvolute the halophilic microbiome. *Genes*, 10(3):220, 2019.

[108] Andre Mu, Brian C Thomas, Jillian F Banfield, and John W Moreau. Subsurface carbon monoxide oxidation capacity revealed through genome-resolved metagenomics of a carboxydotroph. *Environmental Microbiology Reports*, 12 (5):525–533, 2020.

[109] H Bjørn Nielsen, Mathieu Almeida, Agnieszka Sierakowska Juncker, Simon Rasmussen, Junhua Li, Shinichi Sunagawa, Damian R Plichta, Laurent Gautier, Anders G Pedersen, Emmanuelle Le Chatelier, et al. Identification and

assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nature Biotechnology*, 32(8):822–828, 2014.

[110] Yu-Wei Wu, Yung-Hsu Tang, Susannah G Tringe, Blake A Simmons, and Steven W Singer. MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome*, 2(1):1–18, 2014.

[111] Yu-Wei Wu, Blake A Simmons, and Steven W Singer. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, 32(4):605–607, 2016.

[112] Yang Young Lu, Ting Chen, Jed A Fuhrman, and Fengzhu Sun. COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. *Bioinformatics*, 33(6):791–798, 2017.

[113] Vijini Mallawaarachchi, Anuradha Wickramarachchi, and Yu Lin. GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics*, 36(11):3307–3313, 2020.

[114] Bonnie Berger and Peter W Shor. Approximation alogorithms for the maximum acyclic subgraph problem. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 236–243, 1990.

[115] Guy Even, J Seffi Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2): 151–174, 1998.

[116] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357, 2012.

[117] Aaron R Quinlan. BEDTools: the Swiss-army tool for genome feature analysis. *Current Protocols in Bioinformatics*, 47(1):11–12, 2014.

[118] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. arXiv. *arXiv preprint arXiv:0710.3742*, 2007.

[119] Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2): 339–367, 2017.

[120] Dongwan D Kang, Feng Li, Edward Kirton, Ashleigh Thomas, Rob Egan, Hong An, and Zhong Wang. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, 7:e7359, 2019.

[121] Johannes Alneberg, Brynjar Smári Bjarnason, Ino De Bruijn, Melanie Schirmer, Joshua Quick, Umer Z Ijaz, Leo Lahti, Nicholas J Loman, Anders F Andersson, and Christopher Quince. Binning metagenomic contigs by coverage and composition. *Nature Methods*, 11(11):1144–1146, 2014.

[122] Olexiy Kyrgyzov, Vincent Prost, Stéphane Gazut, Bruno Farcy, and Thomas Brüls. Binning unassembled short reads based on k-mer abundance covariance using sparse coding. *GigaScience*, 9(4):giaa028, 2020.

[123] Itai Sharon, Michael J Morowitz, Brian C Thomas, Elizabeth K Costello, David A Relman, and Jillian F Banfield. Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome Research*, 23(1):111–120, 2013.

[124] Curtis Huttenhower, Dirk Gevers, Rob Knight, Sahar Abubucker, Jonathan H Badger, Asif T Chinwalla, Heather H Creasy, Ashlee M Earl, Michael G FitzGerald, Robert S Fulton, et al. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207, 2012.

[125] Julia Oh, Allyson L Byrd, Morgan Park, Heidi H Kong, Julia A Segre, NISC Comparative Sequencing Program, et al. Temporal stability of the human skin microbiome. *Cell*, 165(4):854–866, 2016.

[126] Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10): 1674–1676, 2015.

[127] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[128] Donovan H Parks, Michael Imelfort, Connor T Skennerton, Philip Hugenholtz, and Gene W Tyson. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, 25 (7):1043–1055, 2015.

[129] Alla Mikheenko, Vladislav Saveliev, and Alexey Gurevich. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, 32(7):1088–1090, 2016.

[130] Torsten Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014.

[131] Andrew J Page, Carla A Cummins, Martin Hunt, Vanessa K Wong, Sandra Reuter, Matthew TG Holden, Maria Fookes, Daniel Falush, Jacqueline A Keane, and Julian Parkhill. Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*, 31(22):3691–3693, 2015.

[132] Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):1–11, 2010.

[133] David Couvin, Aude Bernheim, Claire Toffano-Nioche, Marie Touchon, Juraj Michalik, Bertrand Néron, Eduardo PC Rocha, Gilles Vergnaud, Daniel Gautheret, and Christine Pourcel. CRISPRCasFinder, an update of CRISRFinder, includes a portable version, enhanced performance and integrates search for Cas proteins. *Nucleic Acids Research*, 46(W1):W246–W251, 2018.

[134] Lionel Guy, Jens Roat Kultima, and Siv GE Andersson. genoPlotR: comparative gene and genome visualization in R. *Bioinformatics*, 26(18):2334–2335, 2010.

[135] Sorel Fitz-Gibbon, Shuta Tomida, Bor-Han Chiu, Lin Nguyen, Christine Du, Minghsun Liu, David Elashoff, Marie C Erfe, Anya Loncaric, Jenny Kim, et al. Propionibacterium acnes strain populations in the human skin microbiome associated with acne. *Journal of Investigative Dermatology*, 133(9):2152–2160, 2013.

[136] Marcus Fedarko, Jay Ghurye, Todd Treagen, and Mihai Pop. MetagenomeScope: Web-Based Hierarchical Visualization of Metagenome Assembly Graphs. 2017.

[137] Holger Brüggemann, Hans B Lomholt, Hervé Tettelin, and Mogens Kilian. CRISPR/cas loci of type II Propionibacterium acnes confer immunity against acquisition of mobile elements present in type I P. acnes. *PLOS ONE*, 7(3): e34171, 2012.

[138] Eva Krings, Karin Krumbach, Brigitte Bathe, Ralf Kelle, Volker F Wendisch, Hermann Sahm, and Lothar Eggeling. Characterization of myo-inositol utilization by Corynebacterium glutamicum: the stimulon, identification of transporters, and influence on L-lysine formation. *Journal of Bacteriology*, 188(23): 8054–8061, 2006.

[139] Brian Cleary, Ilana Lauren Brito, Katherine Huang, Dirk Gevers, Terrance Shea, Sarah Young, and Eric J Alm. Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. *Nature Biotechnology*, 33(10):1053–1060, 2015.

[140] Quang Tran and Vinhthuy Phan. Assembling reads improves taxonomic classification of species. *Genes*, 11(8):946, 2020.

[141] Wai Yee Low, Rick Tearle, Ruijie Liu, Sergey Koren, Arang Rhie, Derek M Bickhart, Benjamin D Rosen, Zev N Kronenberg, Sarah B Kingan, Elizabeth Tseng, et al. Haplotype-resolved genomes provide insights into structural

variation and gene content in Angus and Brahman cattle. *Nature Communications*, 11(1):1–14, 2020.

[142] Arang Rhie, Brian P Walenz, Sergey Koren, and Adam M Phillippy. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biology*, 21(1):1–27, 2020.

[143] Nicholas Eriksson, Lior Pachter, Yumi Mitsuya, Soo-Yon Rhee, Chunlin Wang, Baback Gharizadeh, Mostafa Ronaghi, Robert W Shafer, and Niko Beerenwinkel. Viral population estimation using pyrosequencing. *PLOS Computational Biology*, 4(5):e1000074, 2008.

[144] Osvaldo Zagordi, Rolf Klein, Martin Däumer, and Niko Beerenwinkel. Error correction of next-generation sequencing data and reliable estimation of HIV quasispecies. *Nucleic Acids Research*, 38(21):7400–7409, 2010.

[145] Irina Astrovskaya, Bassam Tork, Serghei Mangul, Kelly Westbrooks, Ion Măndoiu, Peter Balfe, and Alex Zelikovsky. Inferring viral quasispecies spectra from 454 pyrosequencing reads. In *BMC Bioinformatics*, volume 12, pages 1–10. BioMed Central, 2011.

[146] Christopher Quince, Tom O Delmont, Sébastien Raguideau, Johannes Alneberg, Aaron E Darling, Gavin Collins, and A Murat Eren. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biology*, 18(1):1–22, 2017.

[147] Ryan R Wick, Mark B Schultz, Justin Zobel, and Kathryn E Holt. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, 31 (20):3350–3352, 2015.

[148] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, 14(4):417–419, 2017.

[149] Seth Commichaux, Nidhi Shah, Jay Ghurye, Alexander Stoppel, Jessica A Goodheart, Guillermo G Luque, Michael P Cummings, and Mihai Pop. A critical assessment of gene catalogs for metagenomic analysis. *Bioinformatics*, 2021.

[150] Mina Rho, Haixu Tang, and Yuzhen Ye. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Research*, 38(20):e191–e191, 2010.

[151] Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, et al. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, 2012.

[152] Liang Xiao, Qiang Feng, Suisha Liang, Si Brask Sonne, Zhongkui Xia, Xinmin Qiu, Xiaoping Li, Hua Long, Jianfeng Zhang, Dongya Zhang, et al. A catalog of the mouse gut metagenome. *Nature Biotechnology*, 33(10):1103–1108, 2015.

[153] Hudan Pan, Ruijin Guo, Jie Zhu, Qi Wang, Yanmei Ju, Ying Xie, Yanfang Zheng, Zhifeng Wang, Ting Li, Zhongqiu Liu, et al. A gene catalogue of the Sprague-Dawley rat gut metagenome. *GigaScience*, 7(5):giy055, 2018.

[154] Liang Xiao, Jordi Estelle, Pia Kiilerich, Yuliaxis Ramayo-Caldas, Zhongkui Xia, Qiang Feng, Suisha Liang, Anni Øyan Pedersen, Niels Jørgen Kjeldsen, Chuan Liu, et al. A reference gene catalogue of the pig gut microbiome. *Nature Microbiology*, 1(12):1–6, 2016.

[155] L Xiao, J Estellé, P Kiilerich, Y Ramayo-Caldas, Z Xia, Q Feng, AØ Pedersen, NJ Kjeldsen, E Maguin, J Doré, et al. P1016 The pig's other genome: A reference gene catalog of the gut microbiome as a new resource for deep studies of the interplay between the host and its microbiome. *Journal of Animal Science*, 94(suppl_4):22–22, 2016.

[156] Luis Pedro Coelho, Jens Roat Kultima, Paul Igor Costea, Coralie Fournier, Yuanlong Pan, Gail Czarnecki-Maulden, Matthew Robert Hayward, Sofia K Forslund, Thomas Sebastian Benedikt Schmidt, Patrick Descombes, et al. Similarity of the dog and human gut microbiomes in gene content and response to diet. *Microbiome*, 6(1):1–11, 2018.

[157] Junhua Li, Huanzi Zhong, Yuliaxis Ramayo-Caldas, Nicolas Terrapon, Vincent Lombard, Gabrielle Potocki-Veronese, Jordi Estellé, Milka Popova, Ziyi Yang, Hui Zhang, et al. A catalog of microbial genes from the bovine rumen unveils a specialized and diverse biomass-degrading environment. *GigaScience*, 9(6): giaa057, 2020.

[158] Xiaoping Li, Suisha Liang, Zhongkui Xia, Jing Qu, Huan Liu, Chuan Liu, Huanming Yang, Jian Wang, Lise Madsen, Yong Hou, et al. Establishment of a Macaca fascicularis gut microbiome gene catalog and comparison with the human, pig, and mouse gut microbiomes. *GigaScience*, 7(9):giy100, 2018.

[159] Peng Huang, Yan Zhang, Kangpeng Xiao, Fan Jiang, Hengchao Wang, Dazhi Tang, Dan Liu, Bo Liu, Yisong Liu, Xi He, et al. The chicken gut metagenome and the modulatory effects of plant-derived benzylisoquinoline alkaloids. *Microbiome*, 6(1):1–17, 2018.

[160] Parul Mittal, Rituja Saxena, Atul Gupta, Shruti Mahajan, and Vineet K Sharma. The Gene Catalog and Comparative Analysis of Gut Microbiome of Big Cats Provide New Insights on Panthera Species. *Frontiers in Microbiology*, 11:1012, 2020.

[161] Shinichi Sunagawa, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, Georg Zeller, Daniel R

Mende, Adriana Alberti, et al. Structure and function of the global ocean microbiome. *Science*, 348(6237), 2015.

[162] Jiali Lou, Min Liu, Jiali Gu, Qinghai Liu, Li Zhao, Yushu Ma, and Dongzhi Wei. Metagenomic sequencing reveals microbial gene catalogue of phosphinothricin-utilized soils in South China. *Gene*, 711:143942, 2019.

[163] Bing Ma, Michael T France, Jonathan Crabtree, Johanna B Holm, Michael S Humphrys, Rebecca M Brotman, and Jacques Ravel. A comprehensive non-redundant gene catalog reveals extensive within-community intraspecies diversity in the human vagina. *Nature Communications*, 11(1):1–13, 2020.

[164] Wenkui Dai, Heping Wang, Qian Zhou, Dongfang Li, Xin Feng, Zhenyu Yang, Wenjian Wang, Chuangzhao Qiu, Zhiwei Lu, Ximing Xu, et al. An integrated respiratory microbial gene catalogue to better understand the microbial aetiology of Mycoplasma pneumoniae pneumonia. *GigaScience*, 8(8):giz093, 2019.

[165] Jun Wang and Huijue Jia. Metagenome-wide association studies: fine-mining the microbiome. *Nature Reviews Microbiology*, 14(8):508–522, 2016.

[166] Florian Plaza Oñate, Emmanuelle Le Chatelier, Mathieu Almeida, Alessandra CL Cervino, Franck Gauthier, Frédéric Magoulès, S Dusko Ehrlich, and Matthieu Pichaud. MSPminer: abundance-based reconstitution of microbial pan-genomes from shotgun metagenomic data. *Bioinformatics*, 35(9):1544–1552, 2019.

[167] Junhua Li, Huijue Jia, Xianghang Cai, Huanzi Zhong, Qiang Feng, Shinichi Sunagawa, Manimozhiyan Arumugam, Jens Roat Kultima, Edi Prifti, Trine Nielsen, et al. An integrated catalog of reference genes in the human gut microbiome. *Nature Biotechnology*, 32(8):834–841, 2014.

[168] Roman L Tatusov, Michael Y Galperin, Darren A Natale, and Eugene V Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33–36, 2000.

[169] Erik LL Sonnhammer, Sean R Eddy, and Richard Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function, and Bioinformatics*, 28(3):405–420, 1997.

[170] Ruiting Lan and Peter R Reeves. When does a clone deserve a name? A perspective on bacterial species based on population genetics. *Trends in Microbiology*, 9(9):419–424, 2001.

[171] Mohammadreza Ghodsi, Bo Liu, and Mihai Pop. DNACLUST: accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics*, 12(1): 1–11, 2011.

[172] Nam-Phuong Nguyen, Tandy Warnow, Mihai Pop, and Bryan White. A perspective on 16S rRNA operational taxonomic unit clustering using sequence similarity. *npj Biofilms and Microbiomes*, 2(1):1–8, 2016.

[173] Bo Li and Colin N Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12 (1):1–16, 2011.

[174] Rob Patro, Stephen M Mount, and Carl Kingsford. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology*, 32(5):462–464, 2014.

[175] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527, 2016.

[176] Bertrand Routy, Emmanuelle Le Chatelier, Lisa Derosa, Connie PM Duong, Maryam Tidjani Alou, Romain Daillère, Aurélie Fluckiger, Meriem Messaoudene, Conrad Rauber, Maria P Roberti, et al. Gut microbiome influences efficacy of PD-1–based immunotherapy against epithelial tumors. *Science*, 359 (6371):91–97, 2018.

[177] Daphna Rothschild, Omer Weissbrod, Elad Barkan, Alexander Kurilshikov, Tal Korem, David Zeevi, Paul I Costea, Anastasia Godneva, Iris N Kalka, Noam Bar, et al. Environment dominates over host genetics in shaping human gut microbiota. *Nature*, 555(7695):210–215, 2018.

[178] David Zeevi, Tal Korem, Niv Zmora, David Israeli, Daphna Rothschild, Adina Weinberger, Orly Ben-Yacov, Dar Lador, Tali Avnit-Sagi, Maya Lotan-Pompan, et al. Personalized nutrition by prediction of glycemic responses. *Cell*, 163(5):1079–1094, 2015.

[179] Alexandra Meziti, Despina Tsementzi, Konstantinos Ar. Kormas, Hera Karayanni, and Konstantinos T Konstantinidis. Anthropogenic effects on bacterial diversity and function along a river-to-estuary gradient in Northwest Greece revealed by metagenomics. *Environmental Microbiology*, 18(12): 4640–4652, 2016.

[180] Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, 2015.

[181] Marc W Allard, Errol Strain, David Melka, Kelly Bunning, Steven M Musser, Eric W Brown, and Ruth Timme. Practical value of food pathogen traceability through building a whole-genome sequencing network and database. *Journal of Clinical Microbiology*, 54(8):1975–1983, 2016.

[182] Mario Juhas. Horizontal gene transfer in human pathogens. *Critical Reviews in Microbiology*, 41(1):101–108, 2015.

[183] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[184] Bo Li, Victor Ruotti, Ron M Stewart, James A Thomson, and Colin N Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

[185] Fredrik Bäckhed, Josefine Roswall, Yangqing Peng, Qiang Feng, Huijue Jia, Petia Kovatcheva-Datchary, Yin Li, Yan Xia, Hailiang Xie, Huanzi Zhong, et al. Dynamics and stabilization of the human gut microbiome during the first year of life. *Cell Host & Microbe*, 17(5):690–703, 2015.

[186] Kristoffer Forslund, Falk Hildebrand, Trine Nielsen, Gwen Falony, Emmanuelle Le Chatelier, Shinichi Sunagawa, Edi Prifti, Sara Vieira-Silva, Valborg Gudmundsdottir, Helle Krogh Pedersen, et al. Disentangling type 2 diabetes and metformin treatment signatures in the human gut microbiota. *Nature*, 528(7581):262–266, 2015.

[187] DB Dhakan, A Maji, AK Sharma, R Saxena, J Pulikkan, T Grace, A Gomez, J Scaria, KR Amato, and VK Sharma. The unique composition of Indian gut microbiome, gene catalogue, and associated fecal metabolome deciphered using multi-omics approaches. *GigaScience*, 8(3):giz004, 2019.

[188] Hailiang Xie, Ruijin Guo, Huanzi Zhong, Qiang Feng, Zhou Lan, Bingcai Qin, Kirsten J Ward, Matthew A Jackson, Yan Xia, Xu Chen, et al. Shotgun metagenomics of 250 adult twins reveals genetic and environmental impacts on the gut microbiome. *Cell Systems*, 3(6):572–584, 2016.

[189] Ana Lokmer, Amandine Cian, Alain Froment, Nausicaa Gantois, Eric Viscogliosi, Magali Chabé, and Laure Ségurel. Use of shotgun metagenomics for the identification of protozoa in the gut microbiota of healthy individuals from worldwide populations with various industrialization levels. *PLOS ONE*, 14(2):e0211139, 2019.

[190] Nan Qin, Fengling Yang, Ang Li, Edi Prifti, Yanfei Chen, Li Shao, Jing Guo, Emmanuelle Le Chatelier, Jian Yao, Lingjiao Wu, et al. Alterations of the human gut microbiome in liver cirrhosis. *Nature*, 513(7516):59–64, 2014.

[191] Zhuang Guo, Jiachao Zhang, Zhanli Wang, Kay Ying Ang, Shi Huang, Qiangchuan Hou, Xiaoquan Su, Jianmin Qiao, Yi Zheng, Lifeng Wang, et al. Intestinal microbiota distinguish gout patients from healthy humans. *Scientific Reports*, 6(1):1–10, 2016.

[192] Tommi Vatanen, Damian R Plichta, Juhi Somani, Philipp C Münch, Timothy D Arthur, Andrew Brantley Hall, Sabine Rudolf, Edward J Oakeley, Xiaobo Ke, Rachel A Young, et al. Genomic variation and strain-specific functional adaptation in the human gut microbiome during early life. *Nature Microbiology*, 4(3):470–479, 2019.

[193] Chunlai Wang, Peng Li, Qiulong Yan, Liping Chen, Tiantian Li, Wanjiang Zhang, He Li, Changming Chen, Xiuyan Han, Siyi Zhang, et al. Characterization of the pig gut microbiome and antibiotic resistome in industrialized feedlots in China. *mSystems*, 4(6), 2019.

[194] Ancai Zheng, Hong Yi, Fan Li, Lu Han, Jianhua Yu, Xiaoshu Cheng, Hai Su, Kui Hong, and Juxiang Li. Changes in gut microbiome structure and function of rats with isoproterenol-induced heart failure. *International Heart Journal*, 60(5):1176–1183, 2019.

[195] Till R Lesker, Abilash C Durairaj, Eric JC Gálvez, Ilias Lagkouvardos, John F Baines, Thomas Clavel, Alexander Sczyrba, Alice C McHardy, and Till Strowig. An integrated metagenome catalog reveals new insights into the murine gut microbiome. *Cell Reports*, 30(9):2909–2922, 2020.

[196] G Liu, C Wu, WR Abrams, and Y Li. Structural and Functional Characteristics of the Microbiome in Deep-Dentin Caries. *Journal of Dental Research*, 99(6):713–720, 2020.

[197] Yi Zhou, David R Coventry, Vadakattu VSR Gupta, David Fuentes, Andrew Merchant, Brent N Kaiser, Jishun Li, Yanli Wei, Huan Liu, Yayu Wang, et al. The preceding root system drives the composition and function of the rhizosphere microbiome. *Genome Biology*, 21:1–19, 2020.

[198] Leila Ghanbari Maman, Fahimeh Palizban, Fereshteh Fallah Atanaki, Naser Elmi Ghiasi, Shohreh Ariaeenejad, Mohammad Reza Ghaffari, Ghasem Hosseini Salekdeh, and Kaveh Kavousi. Co-abundance analysis reveals hidden players associated with high methane yield phenotype in sheep rumen microbiome. *Scientific Reports*, 10(1):1–12, 2020.

[199] Michael R McLaren, Amy D Willis, and Benjamin J Callahan. Consistent and correctable bias in metagenomic sequencing experiments. *Elife*, 8:e46923, 2019.

[200] Kenneth W Ellens, Nils Christian, Charandeep Singh, Venkata P Satagopam, Patrick May, and Carole L Linster. Confronting the catalytic dark matter encoded by sequenced genomes. *Nucleic Acids Research*, 45(20):11495–11514, 2017.

[201] Saket Navlakha, James White, Niranjan Nagarajan, Mihai Pop, and Carl Kingsford. Finding biologically accurate clusterings in hierarchical tree decompositions using the variation of information. In *Annual International Conference on Research in Computational Molecular Biology*, pages 400–417. Springer, 2009.

[202] James R White, Saket Navlakha, Niranjan Nagarajan, Mohammad-Reza Ghodsi, Carl Kingsford, and Mihai Pop. Alignment and clustering of phylogenetic

markers-implications for microbial diversity studies. *BMC Bioinformatics*, 11 (1):1–10, 2010.

[203] Xiaolin Hao, Rui Jiang, and Ting Chen. Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics*, 27 (5):611–618, 2011.

[204] Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. DADA2: high-resolution sample inference from Illumina amplicon data. *Nature Methods*, 13(7):581–583, 2016.

[205] Alexandre Almeida, Stephen Nayfach, Miguel Boland, Francesco Strozzi, Martin Beracochea, Zhou Jason Shi, Katherine S Pollard, Ekaterina Sakharova, Donovan H Parks, Philip Hugenholtz, et al. A unified catalog of 204,938 reference genomes from the human gut microbiome. *Nature Biotechnology*, 39 (1):105–114, 2021.