Computational Analyses of Metagenomic Data

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Xi Chen
aus Xinyang, China

Tübingen
2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	18.12.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Daniel H. Huson
2. Berichterstatterin:	Prof. Dr. Ruth E. Ley

Erklärung

Ich erkläre hiermit, dass ich die zur Promotion eingereichte Arbeit selbständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich erkläre, dass die Richtlinien zur Sicherung guter wissenschaftlicher Praxis der Universität Tübingen (Beschluss des Senats vom 25.5.2000) beachtet wurden. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

Unterschrift Xi Chen:

Abstract

Metagenomics studies the collective microbial genomes extracted from a particular environment without requiring the culturing or isolation of individual genomes, addressing questions revolving around the composition, functionality, and dynamics of microbial communities. The intrinsic complexity of metagenomic data and the diversity of applications call for efficient and accurate computational methods in data handling. In this thesis, I present three primary projects that collectively focus on the computational analysis of metagenomic data, each addressing a distinct topic.

In the first project, I designed and implemented an algorithm named Mapbin for reference-free genomic binning of metagenomic assemblies. Binning aims to group a mixture of genomic fragments based on their genome origin. Mapbin enhances binning results by building a multilayer network that combines the initial binning, assembly graph, and read-pairing information from paired-end sequencing data. The network is further partitioned by the community-detection algorithm, Infomap, to yield a new binning result. Mapbin was tested on multiple simulated and real datasets. The results indicated an overall improvement in the common binning quality metrics.

The second and third projects are both derived from ImMiGeNe, a collaborative and multidisciplinary study investigating the interplay between gut microbiota, host genetics, and immunity in stem-cell transplantation (SCT) patients. In the second project, I conducted microbiome analyses for the metagenomic data. The workflow included the removal of contaminant reads and multiple taxonomic and functional profiling. The results revealed that the SCT recipients' samples yielded significantly fewer reads with heavy contamination of the host DNA, and their microbiomes displayed evident signs of dysbiosis. Finally, I discussed several inherent challenges posed by extremely low levels of target DNA and high levels of contamination in the recipient samples, which cannot be rectified solely through bioinformatics approaches.

The primary goal of the third project is to design a set of primers that can be used to cover bacterial flagellin genes present in the human gut microbiota. Considering the notable diversity of flagellins, I incorporated a method to select representative bacterial flagellin gene sequences, a heuristic approach based on established primer design methods to generate a degenerate primer set, and a selection method to filter genes unlikely to occur in the human gut microbiome. As a result, I successfully curated a reduced yet representative set of primers that would be practical for experimental implementation.

Zusammenfassung

Die Metagenomik untersucht kollektive mikrobielle Genome, die aus einer bestimmten Umgebung extrahiert werden, ohne dass Kultivierung der Mikroben oder Isolierung einzelner Genome notwendig sind. Durch diese Methodik werden Fragen über die Zusammensetzung, Funktionalität und Dynamik mikrobieller Gemeinschaften behandelt. Die inhärente Komplexität metagenomischer Daten und die Vielfalt der Anwendungen erfordern effiziente und präzise computergestützte Methoden zur Datenverarbeitung. In dieser Arbeit präsentiere ich drei Hauptprojekte, die sich mit verschiedenen Themen befassen, allerdings alle der Methodik der computergestützten Analyse metagenomischer Daten bedienen. Im ersten Projekt habe ich einen Algorithmus namens Mapbin für das referenzfreie genomische Binning von metagenomischen Assemblies entworfen und implementiert. Binning zielt darauf ab, eine Mischung genomischer Fragmente basierend auf ihrer genetischen Herkunft zu gruppieren. Mapbin verbessert die Binning-Ergebnisse, indem es ein Multilayer-Netzwerk aufbaut, welches das ursprüngliche Binning, den Assemblierungsgraphen und die Informationen zur Paar-End-Sequenzierung kombiniert. Das Netzwerk wird mit Hilfe des Community-Detection-Algorithmus Infomap weiter aufgeteilt, um ein neues Binning-Ergebnis zu erzielen. Mapbin wurde anhand mehrerer simulierter und realer Datensätze getestet. Die Ergebnisse zeigten insgesamt eine Verbesserung der gängigen Qualitätsmetriken für das Binning. Das zweite und dritte Projekt sind abgeleitet von ImMiGeNe, einer kollaborativen und interdisziplinären Studie, die das Zusammenspiel zwischen Darmmikrobiota, Wirt-Genetik und Immunität bei Stammzelltransplantationspatienten (SCT) untersucht. Im zweiten Projekt führte ich Mikrobiomanalysen für die metagenomischen Daten durch, wobei der Workflow Entfernung von kontaminierten Reads sowie die taxonomische und funktionelle Profilerstellung umfasste. Die Ergebnisse zeigten, dass die Proben der Transplantationsempfänger signifikant weniger Reads mit einer starken Kontamination durch die Wirt-DNA aufwiesen und ihre Mikrobiome deutliche Anzeichen von Dysbiose zeigten. Schließlich diskutierte ich mehrere inhärente Herausforderungen, die durch extrem niedrige Konzentrationen von Ziel-DNA und hohe Kontaminationsraten in den Proben der Empfänger entstehen und die nicht allein durch bioinformatische Ansätze behoben werden können. Das Hauptziel des dritten Projekts besteht darin, einen Satz von Primern zu entwerfen, mit denen die bakteriellen Flagellin-Gene des menschlichen Darmmikrobioms abgedeckt werden können. Angesichts der bemerkenswerten Vielfalt der Flagelline habe ich eine Methode integriert, um repräsentative, bakterielle Flagellin-Gensequenzen auszuwählen, einen heuristischen Ansatz basierend auf etablierten Methoden zur Primerentwicklung verwendet, um einen degenerierten Primer-Satz zu generieren, und eine Auswahlmethode angewendet, um Gene auszufiltern, die unwahrscheinlich im menschlichen Darmmikrobiom vorkommen. Als Ergebnis konnte ich einen reduzierten, aber repräsentativen Satz von Primern erstellen, der für experimentelle Anwendungen geeignet ist.

“The moral I draw is that the writer should seek his reward in the pleasure
of his work and in release from the burden of his thoughts”
- The Moon and Sixpence

Acknowledgements

Throughout my PhD life, I have been mentored, helped, inspired, listened to, and tolerated, by many people, to whom I would like to express my gratitude. I have an extraordinary supervisor and mentor, Prof. Dr. Daniel H. Huson, who gave me the precious opportunity to do a PhD in his lab. He has steered my work further down a path beyond my own imagination. And his deep understanding of his field and astonishing enthusiasm for science has been a constant inspiration for me. My thesis advisory committee people, Prof. Dr. Ruth Ley and Dr. Estienne Swart, have been giving me timely and insightful advice for my research and guiding me when I was lost.

I would also like to thank Dr. Frank Chan and Dr. Nicholas Youngblut for bringing me to the haplotagging project, and Dr. Alex Weber for giving me the ImMiGeNe project, and all the amazing people who have offered me precious input for my projects. These interactions have taught me a valuable lesson about communication in academia and furthered me professionally. Special thanks to Andrea, for offering me the flagellin database, and for being a lovely friend.

I shall thank bwForCluster BinAC clusters, which is a part of the Baden-Württemberg state-wide IT infrastructure, for the computational resources. I would like to thank dear Martin and Andreas, for all the IT support in our department.

The young scientists of the Huson lab have my gratitude and appreciation. The pleasure is all mine to have worked side by side with Anupam, Wenhuan, Timo, Banu and Monika. They are my company and my source of faith in my work during the pandemic time. And very special thanks to Bettina, Marine, Sibylle, Sarah, the RST, and the international office. They all helped me tremendously over the years, with kindness and patience few could match.

I was very fortunate to have a few friends and fantastic Huson lab alumni who guided me in research work and also helped me get oriented well in the early days to a culture so different from what I was born into. It was a great pleasure to know Sascha, a generous man who always offers his help and a cheerful friend who lights you up. I am grateful to have learned a lot also from Caner. He is an astute bioinformatician and a close friend with whom I could talk about everything. I'd always remember Ania and Sina, who welcomed me to their parties with open friendliness, bringing me the hidden vitality of Tübingen.

I cherish all the memories of my periodically-heavy-drinking (PhD) group of four. It is a blessing to have in one's life a caring, supportive and generous true friend like Anasuya, a friend to share music taste and go to gigs with like Joe, and a like-minded friend to test mutual insights with like Mikel. They have been there for me through thick and thin during my PhD. I am the luckiest to be friends with Jihyung Kim. There is comforting familiarity and closeness when I am with her that dissipate my anxiety. I truly appreciate Hadeer for always going above and beyond to make a friend feel empowered and cared for. I miss Alexandros very much and memories of the time we spent together still put a smile on my face. And I could not fully express my gratitude to all my friends, Saskia, Amit, Klaudia, and many others. It is impossible to name everyone who was with me in this journey, but I have them all in my heart.

I would like to thank my parents, who gave me everything they could offer, even when they get barely anything in return except for perhaps a printed version of this obscure piece of work for home decor. And I would not have made it without Aleks. He gives me the privilege to be my sane and insane self around him. In all the highs and lows, he makes me feel there is a cozy place to be.

Clarification on pronoun usage in the thesis

In accordance with the conventions of scientific writing, the pronoun "we" is frequently used to refer to myself and other contributors involved in the research work discussed. It is important to note that I, Xi Chen, the author of this thesis, have undertaken all the primary work included herein and written the content independently. The use of "we" is intended to align with the collaborative and inclusive nature of scientific research and to objectively present the findings and conclusions while avoiding a personal perspective in the narratives.

Contents

List of Figures

List of Tables

Prologue

Early microbiologists already noticed that the vast majority of microorganisms did not show up on the Petri dishes. After decades of culturing, scientists came to realize that a more effective way is to study all microbes from a certain environment as a bulk rather than as individual isolates. Earlier work using PCR amplification of phylogenetic marker genes, like 16S ribosomal RNA (rRNA) gene, has led to an era of culture-independent microbial studies and revolutionized our understanding of microbial diversity. However, with these methods, it is difficult to deduce information about the microbial genomes apart from their phylogenetic relationships.

In 1998, the term “metagenome” made its debut, coined by Jo Handelsman and her colleagues to describe the “collective genomes” of soil microbes [1]. Metagenomics as a field has come a long way since then. Extracting and sequencing large DNA fragments and producing data of high throughput used to be the biggest technical obstacle in the early days, but now it has been effectively overcome by the advances in sequencing technology. Today’s metagenomics is marked by an immense amount of data generated from the widespread use of next-generation sequencing (NGS) and long-read sequencing, covering a spectrum of ecosystems. The great legacy of metagenomic data accumulated so far, and the readiness to generate more, have been the ever-lasting source of inspiration for new methods and applications.

We are able to keep up with the exploding data volume thanks to groundbreaking advances in both computer hardware and software. Today, hardware accelerators such as graphical processing units (GPUs) have been frequently used in large-scale biology projects. The mainstream algorithms are highly efficient, optimized, and automated. The computation can be well distributed, and data effectively compressed. A plethora of new methods have been developed to flexibly transform metagenomic data to address various biological or medical questions. Classic computational problems for metagenomics, such as genome assembly, binning, and taxonomic profiling, may arguably serve as the gateway to all explorations of metagenomic data. Their fundamental importance may have been indicated by the subtopics of the Critical Assessment of Metagenome Interpretation (CAMI) challenge [2]. And analyses such as functional annotation, correlation analysis, and gene screening bring the sequencing data to the context of other fields, such as biochemistry and ecology.

Metagenomics has become an indispensable tool for navigating the microbial realm. It has enriched genome databases with numerous high-quality metagenome-assembled genomes (MAGs), unveiling the genetic makeup of many previously uncharacterized microbial species. It also unlocks microbial ecological and evolutionary studies at a scale that would be impossible in the past. For example, the NIH Human Microbiome Project (HMP1), as well as its second phase, Integrative HMP (iHMP) (https://www.hmpdacc.org/), is a specimen of today's multi-omics studies of microbiota. Samples were collected at multiple time points from hundreds of individuals, generating more than 40 terabytes of sequencing data in total. The data have led to a wealth of MAGs, which served as a great source to establish references for human microbiome studies [3, 4]. Linking the microbiome data to a variety of factors sheds light on the role of the microbiome in human health and disease by characterizing the human microbiome dynamics under various host health conditions [4, 5].

This thesis will delve into both software development and applications of metagenomics. Three topics will be presented. The first one focuses on the metagenomic binning problem, coming up with a software named Mapbin. The second and third ones were derived from the same multi-disciplinary study of the human gut microbiome, which is named ImMiGeNe. We will dig into the processing and interpretation of the metagenomic data in the second topic and the primer design problem in the third.

Chapter 1

Background

1.1 Metagenomics brought about a paradigm shift in microbial research

Our modern view of microbes is the result of a series of greatest technological revolutions. For centuries, microbiological research was dominated by culture-based methods, and we understood microbial diversity from what could be grown in the laboratories. However, compared to that of animals and plants, the morphological descriptions of microbes were limited and intrinsically different. Adopting the then-popularized criteria in plant and animal genealogy to microbial evolution was met with incredible difficulty. As a result, for a long time, the genealogy of microbes stayed vague to microbiologists [6]. With the advent of nucleotide sequence, in 1977, Carl R. Woese published with George E. Fox a pioneering work that had the phylogenetic taxonomy based on the 16S ribosomal RNA (rRNA) genes [7]. Using this molecular marker, Archaea was also added as a domain for the first time onto the tree of life. Since then, sequencing technology has paved the way for decades of thriving culture-independent microbiological research. Today's sequencing depth and bioinformatics advances allow a deep look into a complex microbial sample with mixed populations without *a priori* knowledge about their composition. A growing number of studies incorporate multi-omics methods, including metagenomics, metatranscriptomics, metaproteomics, epigenomics, and metabolomics, to study known and unknown microbes in a variety of environments. Essentially, the meta-omics terms all share the notion of analyzing the collection of microbial contents from environmental samples. The beginning of such a notion is marked by the advent of metagenomics.

Metagenomics is a field that analyzes genetic material from all microbial organisms in a bulk sample retrieved directly from the environment they inhabit, without requiring the separation of individual genomes or species [1, 8]. In ecology, the collection of microbial organisms coexisting in a specific ecosystem is referred to as a microbial community. Metagenomics provides a comprehensive overview of the composition and diversity of the microbial communities and allows for extrapolation of their metabolic capacities. In the following section, we briefly review the technology backbone of metagenomics, and highlight some key achievements that transformed our perception of microbial research.

1.1.1 Metagenomic studies powered by advances in sequencing technology

Ever since the early days, metagenomics has relied on shotgun sequencing to generate data [9]. Shotgun sequencing refers to the DNA library preparation method which breaks DNA molecules into smaller fragments. The fragments will subsequently be sequenced and yield *reads*. Different sequencing platforms require different sample preparation and produce reads of varying lengths.

Next-generation sequencing (NGS)

In the early 2000s, following the completion of human genome assembly, a new generation of sequencers became commercially available. They include the now-discontinued pyrosequencing and sequencing by ligation, the Ion Torrent sequencing, and the widely used Illumina sequencing. They vary in technical details and performances but, in general, produce short reads (under 400 bp per read) in a massively parallel manner, which differentiates them from first-generation sequencing technologies such as Sanger sequencing[10]. The sequencing usually involves tethering the DNA templates to a surface with adapter hybridization, and amplifying them into a cluster. The sequencing is done by measuring the fluorescent signal released when fluorescently tagged nucleotides are added onto the single-stranded templates one position at a time with DNA polymerases. These technologies are commonly called next-generation sequencing (NGS). Their advantages are marked by the high throughput, high accuracy (99.9% and above at the base level) and low cost[10].

Long read sequencing

In the 2010s, a third generation of sequencing began. Dominated by PacBio Single Molecule, Real-Time (SMRT) sequencing, and Oxford Nanopore sequencing, these technologies typically produce reads of kilobases or even megabases, which are commonly referred to as *long reads*. Long reads are not uniform in length like short reads. PacBio HiFi reads usually range from 10 to 25 kb in lengths[11]. Nanopore long reads are usually within 10 to 100 kb, and ultra-long reads 100 to 300 kb. The new technologies have also made groundbreaking achievements in the throughput. Illumina NovaSeq sequencing platform achieved a new level of throughput in 2017, yielding 3000 Gb reads in a single run[10]. This record is now surpassed by the third-generation sequencers. For instance, Nanopore PromethION sequencers could yield up to 290 Gb per flow cell, and generate terabases in total [12, 13]. The two technologies are completely different in sequencing mechanisms. PacBio SMRT sequences the DNA as a part of a single-stranded circular molecule shaped like a bell (SMRTbell). The sequencing is performed in a well by measuring the fluorescent signals[14, 11]. Nanopore uses ion current to drive a single-stranded DNA through a nanoscale protein pore, and measures the current change to determine the base that passes. Initially, long reads are significantly more error-prone in terms of the base level accuracy compared to short reads, but the gap has been diminished gradually with the technical upgrades of both PacBio and Nanopore[15, 16]. The most recent technologies achieve an accuracy of 99.5% for PacBio and 99% for nanopore[16, 17, 11, 13].

Besides short and long read sequencing, some alternatives, such as Hi-C [18, 19] and read-cloud sequencing [20], have also been applied to metagenomic studies. With powerful modern sequencing technologies as the engine, metagenomics has revolutionized our perception of microbiology in many ways in the last decade.

1.1.2 Genome-resolved era of metagenomics

Retrieving the constituent genomes from read data has been a goal for metagenomics since the very beginning of the field. Initially, this could only be achieved with desirable output in low-diversity microbial communities, such as those in the acid mine drainage (AMD), an extreme environment with a handful of detectable microbes [9]. Nowadays, the high volume of data and the rapidly upgraded computational hardware and software have extended this application to complex systems, such as soil microbiomes.

Resolving individual genomes from metagenomic data is commonly done by assembling the reads into contigs. Modern metagenomic assembly algorithms like short read assemblers metaSPAdes [21] and MEGAHIT [22], and long read assemblers metaFlye [23], hifiasm-meta [24], Canu [25], HiCanu [26] and Shasta [27] have been reported in a wide range of application cases with metagenomic data. With short reads, usually, all the contigs in an assembly are fragments of complete microbial genomes. Long reads can achieve remarkably greater sequence contiguity. The performance is further enhanced when assembly is coupled with contig polishing, using tools such as Racon [28], Medaka (for Nanopore assembly, `https://github.com/nanoporetech/medaka`) and Pilon [29]. This could lead to a few complete, closed genomes [30, 31]. Recently, it has been reported that the newest technology can lead to more excellent performance without the polishing step [17].

Most metagenome-assembled contigs are still far from being a complete genome. A binning step can be applied to further separate the mixture of contigs based on their source taxa, resulting in lineage-deconvoluted clusters of contigs. These clusters are commonly referred to as metagenome-assembled genomes (MAGs). We will discuss the computation of contig bins in more detail in Chapter 1.2. Due to the computational difficulties in separating related genomes during both assembly and binning, the resultant MAGs vary in quality and many consist of contigs from multiple species or strains. Among the bins, the most valuable are those with low contamination and encompassing all or nearly all the content of the microbial genome. These bins can be documented as a microbial genome in a public database because, despite perhaps the fragmented nature, they are able to provide most of the genomic information of a microbial lineage. Genomes of many unculturable microbes are now available in the format of MAGs [32]. MAGs have become indispensable to modern genome databases and profoundly expanded the tree of life. Although alternative methods, such as single-cell genomics, can also retrieve microbial genomes culture-free, metagenomic approaches have made the most prominent contribution due to their cost-effectiveness in obtaining and mining the data. In the last decade, hundreds of thousands of near-complete MAGs have been produced from large-scale metagenomic projects, such as Human Microbiome Project [5, 4], the Earth Microbiome Project (https://earthmicrobiome.org/ [33], and research efforts by Parks *et al.* [32], Almeida *et al.* [34] Pasolli *et al.* [35], and Nayfach *et al.* [36].

We used to be informed of the existence of a microbe via its activities, such as manifestations of its metabolic processes, or the presence of its molecular markers, such as its 16S rRNA gene. Now we are right in an era in which the entire genomes of the unknown can be made available, and their metabolic potential predicted upon the confirmation of their existence. And the inferences of their existence are highly automated, statistics-based, and batch produced. Genomes of various unknown or understudied microbes can be generated all at once by automated pipeline from one or more metagenomic datasets [37, 34, 36]. The novelty of their lineages can be verified via genome comparison despite the lack of description or evidence regarding their physiology [32]. With the genome-resolving power, metagenomics revolutionized the way we study microbial lives.

1.1.3 Linking functions to microbial genomic sequences

Gaining a large number of genomes is not the ultimate goal. The genomes are the most meaningful to be read as a recipe for various cellular functions. Our view of evolution is also based on the comparison of the biological functions encoded by the genomes. A primary feature that distinguishes metagenomics from other cultivation-independent genomics methods (e.g., 16S rRNA sequencing) is that it is able to directly reveal the metabolic capacity of the targeted microbiome. We highlight two important aspects regarding the functional analysis: Genome annotation and functional profiling of microbial communities.

Annotating genomes or genomic fragments is a common crucial task for genomic studies. Compared to the eukaryotes, such a task is less complicated with prokaryotes, because both bacterial and archaeal genomes are densely packed with protein-encoding genes in an almost non-overlapping manner [38]. Tools like Prodigal [39] and GeneMarkS [40] allow for efficient gene prediction of microbial genomes. Because 70 to 80% of protein-encoding genes on prokaryotic genomes are conserved even over a long evolutionary distance [41, 38], the functions of the predicted genes can be annotated based on the multiple sequence alignment to known genes. Another basis is that many prokaryotic genes are organized in the genetic unit of an operon, and the function of the unknown could also be deduced from the known genes which are consistently neighboring them [42, 38]. With these, we have discovered from the newly reconstructed MAGs a substantial number of putative or hypothetical genes and even new homologs [43], which further broaden our view of microbial gene orthology.

The enriched gene ortholog databases greatly facilitate the functional profiling of microbial communities. The goal of functional profiling is to quantify the metabolic content of a community. It is commonly performed by assigning the reads to a nucleotide or protein reference which has already been functionally annotated [44, 45]. Functional profiling reveals the metabolic state of a community, and effectively highlights changes in key processes, which can serve as a roadmap for designing subsequent biochemical or cellular experiments [46, 47].

1.1.4 The expansion and alteration of the microbial tree of life

Metagenomics has also made a profound impact on our understanding of microbial phylogenies. The first and most obvious contribution is the discovery of new lineages. It is estimated that for both bacteria and archaea, culture-free methods expanded the domains by a scale of around five folds [43], a majority contributed by metagenomic data. Single-cell methods are also able to resolve genomes, but they have been shown to have lower throughput and produce assembled genomes that are generally not as complete as MAGs [48, 43, 49]. Large-scale metagenomic studies are also much more frequently reported than single-cell studies.

One famous recent example is the discovery of the archaea superphylum, the Asgard group. Initially, 16S rRNA gene-based study on deep marine sediments hinted at the existence of certain uncharacterized new species [50]. This study was quickly followed by metagenomic sequencing of the same sample, leading to the discovery of a subgroup *Lokiarchaeota* (now *Lokiarchaeia*) with a 92% complete genome successfully reconstructed [51]. Two years later, the Asgard superphylum was reported with additional groups of uncultivated archaea, adding a brand new clade to the archaeal phylogenetic tree [52].

Some taxonomic groups were only identified with few representatives at the time of their discovery, and therefore could only be tentatively placed on the tree of life. Their lineages and phylogenetic relationships with others were later refined with a flood of newly reconstructed genomes, many of which were MAGs. Besides Asgard, the new discoveries of the bacterial candidate phyla radiation (CPR) and the archaeal radiation DPANN (an acronym for its five major subgroups *Diapherotrites*, *Parvarchaeota*, *Aenigmarchaeota*, *Nanoarchaeota* and *Nanohaloarchaeota*) have led to topological changes to the tree of life. In the late 1990s, Dr. Norman Pace Jr. and colleagues uncovered a candidate "division" named OP11 [53]. This division was later found to encompass several groups at the phylum level, including the *Microgenomates* and *Parcubacteria*. Later, based on the analysis of rRNA gene and protein sequences derived from metagenomic data, these were further clustered as CPR [54, 55]. Analogous to CPR, DPANN were also initially a few groups of nanoorganisms recovered by different technologies, otherwise poorly identified in the pre-metagenomics era. The availability of new genomes allows for genome comparison, based on which the supergroup DPANN was proposed [54]. So far, thousands of high-quality genomes are available for Asgard, CPR, and DPANN, enabling the delineation of many new candidate phyla within them.

More importantly, phylogenetic placement of Asgard, CPR and DPANN showed their unique position on the tree of life. A considerable proportion of the proteins predicted from Asgard MAGs have homologs only or mainly belonging to eukaryotes. This includes the cell structural protein actins and many other actin-related proteins, as well as small GTPases that are necessary for the phagocytosis process in cellular organisms [51, 47]. These findings support an intriguing hypothesis that eukaryotes might have evolved from certain archaea. Asgard archaea are now widely acknowledged as the closest to eukaryotes among known prokaryotes [52, 47]. Both CPR and DPANN are organisms with very small genomes (between 0.5 to 1 Megabases) and cell sizes. The functional annotation of their genomes revealed a lack of key biosynthesis-related genes, leading to the common inference that they are mostly symbionts that rely on other community members to survive [56, 38, 43]. Both groups contain an astonishing level of diversity, the extent of which is still an unsettled question. Analyses derived from the annotation of the genomes suggested that the diversity within CPR can be a quarter of what is observed in all other bacteria combined [32]. CPR and DPANN both display as major radiations, and the presence and absence of genes in each radiation are highly similar in patterns, raising questions about their evolutionary history. They may have resulted from rapid evolution, but it is also likely that they are descendants of very ancient microorganisms with small genomes, and the long branches within the radiations may result from undersampling [43, 38]. This question can only be addressed through deeper genomic comparison with additional distinct genomes.

1.1.5 Genome taxonomy database (GTDB): a conceptual shift in microbial taxonomy

Another major conceptual shift lies in the advent of GTDB. Taxonomy is a scientific classification of organisms into hierarchical groups, ideally based on and reflecting their evolutionary relationships [57]. Before GTDB came out, the mainstream taxonomy of organisms, such as the NCBI taxonomy, was essentially a product of successive historical updates, bearing inconsistency and conflicts within their classification systems. Many taxonomic groups are classified by their phenotypes, which are hard to standardize, and the taxonomic ranks reflect the evolutionary distances inconsistently [57, 58]. Although the notion of sequence comparison-based phylogenies has been circulating for a few decades [6], a genome-based taxonomy could not have been successfully constructed without a sufficient number of microbial genomes. In 2018, Parks *et al.* reported the success in creating a standardized bacterial taxonomy, using genomes from RefSeq as well as a considerable number of MAGs from an extensive metagenomic assembly project that they conducted earlier [57, 32]. The taxonomy was based on an underlying genome tree constructed from a concatenated alignment of ubiquitous single-copy proteins and initially annotated with NCBI taxonomy. Polyphyletic groups were removed, and the threshold of taxonomic ranks was consistently redefined based on the relative evolutionary divergence (RED), or for the species clusters, average nucleotide identity (ANI) [57, 59]. GTDB has gone through several updates, which integrated the archaea domain and introduced impressive expansions of the tree (over 270% increase in the number of genomes integrated, and a 200% increase in the number of species clusters compared to the first release) [60, 61, 59]. As a phylogenetically consistent, comprehensive, and up-to-date taxonomy, GTDB has been widely applied to microbial studies since its release [37, 36, 62]. By mentioning this, we do not mean to make the unjustified argument that GTDB prevails over traditional taxonomic systems in all scenarios [63], but rather to emphasize the conceptual innovation it has brought about. Unlike its predecessors, it does not strive for a delicate balance between multiple aspects such as the historical labeling, microbial morphology, physiology, and genomics, in order to define prokaryotic taxa [63, 58, 59].

It instead focuses sorely on features derived from the genomes themselves. This method also allows for better inclusion of uncultivated microbes and provides a more comprehensive framework for integrating lineages derived from metagenomic samples.

1.1.6 From metagenomics to pangenomics

The lavish wealth of assembled microbial genomes means not just the inclusiveness of previously undiscovered lineages but also a high number of genomes available for each species on average. This advancement firstly highlights the question of strain variations. There is a growing trend among the bioinformatics community towards developing strain-aware tools and adopting the strain-level resolution as a performance evaluation metric [64, 65, 66, 2]. Furthermore, the copious availability of genomes allows us to compare genomes within or across phylogenetic groups. This leads us to the field of pangenomics. Pangenome is a concept first introduced by a study in 2005 focusing on multiple strains of the species *Streptococcus agalactiae* [67, 68]. They described the set of genes present in all strains as the "core genome" and the set of genes occasionally absent in some trains as the "dispensable genome". Genes in the dispensable genome can be further categorized as "unique", if they are exclusive to one strain, or "accessory", if they are shared in multiple strains but not all [68]. Arguably, the pangenome concept addresses a challenge to the traditional definition of a genome. In the earlier time, due to the scarcity of fully sequenced genomes, the genome of a species usually refers to a reference genome. Nowadays, with tens or even thousands of complete genomes available for one species, we are able to inspect the diversity within.

Even in Carl Woese's time, the potential versatility of the genomic content has been well noted [6]. Preceding the advent of sequence analysis-driven microbial studies, researchers made many attempts to use physiological traits to classify bacteria. However, this hit a bottleneck because many physiological traits that could be used to cluster a group of organisms were not always found in their close relatives, which led to a fundamental problem of defining a bacterial species [6]. This issue was not resolved until Carl Woese proposed the use of a phylogenetic marker, 16S rRNA gene, as the basis to define lineages [7, 6]. Today, the overabundance of assembled microbial genomes brings a conceptual reminiscence of the previous problem. In a sense, the phylogenetic marker approaches unified the taxonomic classification system of bacteria and archaea, and such classification eventually guided us to return to the previously perplexing physiological diversity within a lineage.

Within one species, while many metabolic activities are common to all members, some strains may exhibit unique traits such as drug resistance and pathogenicity. Pangenomic approaches are immensely helpful in linking these metabolic observations to the genomic mechanisms. Recently, Pöntinen *et al.* uncovered the adaptation mechanism of the hospital pathogen, *Enterococcus faecalis*, by analyzing its pangenome derived from genomes collected from 1936 to 2018 [69]. Pangenomic analyses have also led to the identification of virulent genes in species like *Helicobacter pylori* and *Escherichia coli* [68]. Pangenomic analyses are especially useful for mining MAGs recovered from longitudinal or spatial studies, providing valuable insights into the ecological mechanisms of a given species by examining patterns in its gene content [70]. For instance, the gene content of each metagenomic sample could be predicted by annotating the MAGs. Generally speaking, in these studies, each sample comes from a specific ecological setup. By performing the meta-analysis of the genes in all metagenomic samples, we can address the question of whether a gene is a generalist, common to all samples due to shared ecological factors, or a specialist, found only in one or a subset of samples because of certain ecological factors specific to these samples. Tierney *et al.* conducted a pangenomic analysis of multi-body site human microbiomes, and built a gene catalog based on the comparison of gene content. These results provided a comprehensive overview of the metabolic landscape of the human microbiome [71].

In summary, metagenomics is the primary means for reconstructing the genomes and phylogenies of the uncultivated vast majority of microbes. Metagenomics has taken microbial research to a new era in which the genomic blueprint of the research targets is readily obtained and utilized as a guide for future studies. The constant influx of MAGs has not only extended and refined the tree of life, but also expanded the diversity of gene orthologs, which has tremendously facilitated the functional analyses of microbial communities. The explosive growth of metagenomic data is leading us to new and deeper questions about microbial evolutionary history. The overall abundance of metagenome-derived genomic data further brought in the concept of pangenome, broadening our perspectives of microbial genomic studies.

1.2 Metagenomic contig binning

A highly informative end of metagenomic sequencing data analysis is to retrieve individual genomes. This has brought us unprecedentedly close to the uncultivated majority of microbes. As of today, our understanding of the microbial world has been greatly enriched by the tens of thousands of reference-quality microbial genomes reconstructed from metagenomic samples. Commonly, this is done by means of *de novo* metagenomic assembly.

Recent assembly methods are able to produce highly accurate genomic fragments (termed contigs) of members in a microbial community. But to further complete them to full-length genomes is a difficult task. Many microbial communities contain closely related members, and their genome-level similarity prohibits assembly algorithms from correctly putting their fragments together [72, 73]. Such limitation could be compensated subsequently by binning. Metagenomic binning is a computational step to cluster biological sequences by their organism of origin. By a broad definition, the sequences could be genomic fragments of any kind, such as sequencing reads or assembled contigs or scaffolds. The goal is to deconvolute the sequence mixture and place sequences into taxonomic units of a certain level.

Binning could be computed with or without references [74]. In reference-based binning, or supervised binning, bins are inferred from the sequence comparison between input and reference [2]. As sequence comparison is able to produce meaningful results for sequences as short as 100bp, this binning approach has the advantage of low requirement for input sequence lengths. However, the taxonomy of the resultant bins is derived from that of the reference genomes. Therefore, the performance of this approach heavily depends on the inclusiveness of the reference. It works best when the source genomes of the input sequences are either included or closely related to those in the reference. Benefiting from the high efficiency of current sequence comparison algorithms, many reference-based binning approaches are efficient, achieving satisfactory sensitivity and specificity. But they usually perform poorly for sequences from unknown or unclassified organisms [74].

On the other hand, reference-free binning, or unsupervised binning infers the bins by analyzing the intrinsic characteristics of the input sequences and therefore requires no prior knowledge of the constituent genomes [74]. It typically works with longer sequences. It suits well contigs from metagenomic assemblies, as they typically contain longer pieces that might originate from undocumented genomes [2].

Unsupervised binning in general consists of two key steps, to quantify the input sequence properties, and based on that, to cluster the sequences. It is a lively field with a constant flux of new algorithms. We may navigate their novelties from the said two aspects, which we will briefly summarize here.

1.2.1 Genomic features used for unsupervised binning

Oligonucleotide composition and abundance patterns are two sequence properties that are conventionally used to infer sequence clusters. DNA fragments from the same genome are believed to have a similar short oligonucleotide (or k-mer) usage pattern. These oligonucleotides are usually 2-5 bases, typically tetranucleotide (4-mer) [75, 76]. Abundance or coverage-based methods assume that the abundance or coverage of subsequences from the same organism shall be strongly correlated in samples. For a single sample, this means the read coverage of contigs from the same chromosome follows the Lander-Waterman statistics [77], or in other words, their abundance profiles shall be similar. For multiple samples, contigs from the same genome shall be co-abundant, or in other words, the covariance of their abundances across the samples shall be high [78, 79, 80].

Both sequence properties have been adopted by the field of metagenomics for over a decade. In 2009, Dick *et al.* studied microbial communities from acidophilic biofilms. Using tetranucleotide frequency (TNF) patterns as a genome-specific signature, they successfully partitioned contigs and revealed a list of previously unknown organisms [76]. Iverson *et al.* were the first to unveil a closed genome from the marine group II *Euryarchaeota* in 2012, with the help of TNF to cluster the scaffolds [81]. Abundance statistics-assisted binning can be dated back to 2013, when Albertsen *et al.* segregated some dominant species from wastewater samples [79]. In 2014, both Canopy [80] and CONCOCT [82] came out, bringing this method further by introducing co-abundance across multiple samples, typically coupled with co-assembly of metagenomic samples. CONCOCT was one of the first programs to adopt both features, followed by several other widely-used binning tools, such as MetaBAT 2.0 [83], MaxBin 2.0 [84] and GroopM [85].

Other sequence features have also been employed to deduce bins. Conventional binning methods all work with assembled contigs. With paired-end sequencing, read pairs could provide extra linkage between contigs. COCACOLA leverages this information to improve its performance [86]. Binnacle pipeline scaffolds the contigs first, and then computes and evaluates read coverage, and detects the misassemblies, before passing them to an existent binning algorithm [87].

Very recently, some attention was drawn to the contiguity information in the assembly graph. Mallawaarachchi *et al.* implemented GraphBin, a binning refinement program that reevaluates the bins by checking in the assembly graph the linkage of their constituent contigs [88]. Later, Lamurias *et al.* released GraphMB, a deep learning (DL)-based binning algorithm that encodes into embeddings not only contig statistical features but also the assembly graph. Compared to GraphBin, it integrates the assembly graph into the binning process instead of post-processing. It also addresses the lack of focus on long-read assemblies in currently available tools [89].

We shall point out that the innovations in binning methods go beyond what was discussed above. DNA sequencing is a fast-moving field and new types of sequencing data always come out. This means, while the data clustering part of the binning problem may remain a classic mathematical question, what to be clustered gets updated rapidly with the sequencing technology. Alternative genomic features can always be incorporated as part of the clustering data. For instance, several new methods came out recently for high-throughput chromosome conformation capture (Hi-C) metagenomic data [90, 19, 91, 92],. Hi-C technique captures topological proximity of DNA segments *in vivo*. It was initially invented for the human genome, and later found its application in metagenomics. With the assistance of the crosslinking proteins bound to them, spatially adjacent DNA fragments get ligated, and a library for Illumina shotgun sequencing is prepared. With Hi-C data, a contact map between assembled DNA segments can be generated [93, 94]. Hi-C binners typically interpret and integrate sequence connectivity information offered by the contact map to perform clustering [90, 19, 91]. These methods altogether have demonstrated that a high-quality characterization of the differences between constituent genomes in the data is crucial for binning. The more distinct genomic signatures used as the clustering basis, the better extraction of individual genomes from metagenomic data.

1.2.2 Clustering methods used for unsupervised binning

Binning is a clustering problem by nature. The descriptions "supervised" and "unsupervised" in fact are terminology from machine learning. All binners, after preparing the data, need to fit the data to a model to perform clustering. There are myriad clustering algorithms. Here we sort them out into 3 broad categories: classic, network-based, and DL-based.

Classic algorithms here refer to some basic methods that do not involve a network analysis or DL. It includes centroid-based models such as K-means and K-medoids, distribution-based ones such as Gaussian-mixture models, and density-based ones such as density-based spatial clustering of applications with noise (DBSCAN). From the input data, the similarity between data points is quantified and passed on to the clustering algorithm. Early binning tools used to rely on these algorithms. One primary concern with these approaches is their request of the number of clusters to be predetermined. MaxBin [95] and MaxBin 2.0 [84] both rely on expectation maximization (EM). They first work out the pairwise sequence similarities and then use that as a basis to compute the probability of a sequence coming from a certain genome. And the number of bins is estimated from the analysis of single-copy marker genes. MetaBAT kneads the TNF and coverages into one sequence similarity score, and in its original version, it uses a K-medoids model to cluster [96]. CONCOCT, after using a combined vector to represent the genomic features of contigs and applying principal component analysis (PCA) for dimensionality reduction, uses a Gaussian mixture model to cluster [82].

Network-based methods use a graph structure to represent the relationships between the objects, i.e., contigs for the binning problem. Our notion here excludes neural network-related approaches, and refers only to methods like Markov clustering, label propagation, and community detection algorithms such as the Louvain algorithm and the more recent Leiden algorithm. MetaBAT 2.0 has turned to modified label propagation clustering instead of K-medoids [83]. The binning refinement tool GraphBin also uses a label propagation algorithm [88]. As for community-detection algorithms, both Louvain and Leiden algorithms partition the network by computing the modularity and trying to optimize it [97]. These methods are particularly useful for complex, dense graphs, which could be the case in binning depending on how edges are established between the contig nodes. Community detection algorithms have been adopted by some Hi-C binners. Bin3C uses a Louvain algorithm-derived method Infomap to perform the binning [19], and HiCBin a Leiden algorithm [91]. Our binning method, as will be explained in full detail later, also adopts the Infomap algorithm [98]. So far as we know, binners for regular metagenomic assemblies using community-detection network clustering have not yet been published.

And finally, DL-based methods are those that perform the clustering using neural networks. DL has only made its debut in metagenomic binning quite recently, in VAMB, GraphMB, CLMB and etc., yet has already been reported to be of high performance [99, 89, 100]. VAMB uses variational autoencoders (VAE), a DL method, to autoencode the TNF and abundance features to perform the clustering [99]. CLMB uses the same features, but before training, it augments some simulated statistical noises to the features, then uses the VAMB framework to get the bins [100]. GraphMB adds the assembly graph into its feature learning process. It uses graphic neural networks (GNNs), an application of deep neural networks to graph data structure [89]. As developers of these methods pointed out, the advantage of DL is that it is able to encode latent features, and thus has great potential for data whose underlying statistic model is challenging to figure out. Another positive factor is that embeddings are low-dimensioned compared to the original features.

Table 1.1: Summary of open-access metagenomic binning tools. TNF: tetranucleotide frequency; ABD: abundance statistics (or coverage).

Name	Recent update	Features	clustering method	Input data optimized for	Ref.
Canopy	2014	ABD (genes)	Self-defined	Short reads	[80]
CONCOCT	2019	TNF, ABD	Gaussian mixture models	Short reads	[82]
GroopM	2014	TNF, ABD	Self-defined	Short reads	[85]
COCACOLA	2017	TNF, ABD	K-means	Short reads (paired-end)	[86]
MetaBAT 2.0	2019	TNF, ABD	Label propagation	Short reads	[83]
MaxBin 2.0	2020	TNF, ABD	Expectation maximization	Short reads	[84]
VAMB	2022	TNF, ABD	Variational autoencoder	Short and long reads	[99]
GraphBin2	2020	Assembly graph	Label propagation	Short and long reads	[88]
CLMB	2022	TNF, ABD	Deep learning based on VAMB	Short and long reads	[100]

GraphMB	2022	TNF, ABD, assembly graph	Graphic neural networks	Short and long reads	[89]
Bin3C	2019	Hi-C read-pairs	Infomap	Hi-C reads	[19]
HiCBin	2022	Hi-C read-pairs	Leiden algorithm	Hi-C reads	[91]

1.2.3 From binning to high-quality MAGs

As introduced in Chapter 1.1, binning is the key step to retrieve MAGs from metagenomic assemblies, but the quality of resultant MAGs varies between bins. Studies have widely acknowledged the use of completeness and contamination as the two key metrics of MAG quality. Completeness refers to the inclusiveness of all genomic components of a given genome, and contamination means the presence of contigs from other organisms or sources other than the target. To be inclusive of the undocumented genomes, the assessment of a bin is usually based on the inference of the gene content rather than aligning the bin directly against existing genomes. CheckM [101] is one of the most popular tools to assess the MAGs from unsupervised binning. CheckM uses a reference genome tree that has been annotated for lineage-specific marker genes. An inquiry putative genome is first predicted for its marker gene content. Based on this, it is placed to a specific lineage on the reference tree. Completeness and contamination are estimated based on the inclusiveness of the marker gene content in the lineage [101].

Most genome databases have established quality standards for registering a MAG, and genomic research communities have also put forward a few general criteria for reporting microbial genomes. The Genomic Standards Consortium (GSC) developed the minimum information about a metagenome-assembled genome (MIMAG) standard for MAGs. A MAG is deemed as of high quality if it has completeness over 90% and contamination below 5%, contains 16S and 23S rRNA genes, and a minimum of 18 tRNA genes [102]. But a technical problem with implementing such a standard is that the rRNA gene regions are highly similar among different lineages, and common *de novo* assembly tools are not optimized to reconstruct these regions well [103, 35, 34]. Therefore, many studies report near-complete or high-quality MAGs with no or relaxed criteria for the rRNA regions [34, 99, 89].

1.3 The gut microbiota and host immunity

The human body comprises at least half, if not more, microbial cells - bacteria, archaea, viruses, fungi and other microorganisms [104, 5, 105]. In the biomedical field, the most extensively studied human-associated microbiome is the gut microbiome. Above 90% of microorganisms found in the human body reside in the intestine [105], and the microbial community is colloquially referred to as the gut microbiome.

Strictly speaking, the gut microbiome refers to the community living in close proximity to and interacting with the intestinal mucosa. In practice, the "human gut microbiome" in many studies refers to the microbial communities in the large intestine or the colon [106]. As direct sampling of the mucosal microbes is not feasible for human subjects due to its invasiveness, fecal samples are the most commonly used proxies for the mucosal microbiota. Alternative sources of samples, such as cecal samples, could also be collected, but usually only in animal studies (e.g., with chicken and mice), and may require the sacrifice of the animals [107, 108].

Although gut microbiota composition varies between individuals, at a higher phylogenetic level, for healthy adults, certain similarities can be observed. Generally speaking, the human gut is a highly anaerobic environment, and the microbiota mainly consists of anaerobic bacteria and archaea [109]. In healthy adults, the gut microbiota typically comprises some hundreds to a thousand bacterial species, with a dominant proportion of the phyla *Bacteroides* and *Firmicutes*, a small fraction of *Actinobacteria* and *Proteobacteria*, alongside a variety of other organisms of lower abundances [110, 106, 111]. A vast majority of the gut microbes are beneficial or at least harmless to the host [110]. For over a decade, intensive research efforts have been devoted to uncovering the taxonomic composition of the gut microbiome, and now we have gained near-complete knowledge of the microbial lineages present in the regular human gut, at least at the genus and species level and higher. In other words, the gut microbiota is essentially comprised of species that have been characterized [112].

The significance of the gut microbiome to human health is becoming increasingly clear, providing biomedical researchers an excellent opportunity to explore the potential of therapeutic interventions targeting the gut microbiome [113, 114]. A thorough understanding of the microbial interactions with the host is crucial in this regard. Studies often describe the relationship between the gut microbial community and the human host as an "interplay", as the host environment helps shape the microbial community and, in turn, the community can impact the health conditions of the host [115, 104]. While certain diseases, such as cholera, salmonellosis and gastroenteritis, can be attributed to specific infectious agents, many other human health issues are associated with changes in the composition and metabolic functions of the microbial community [112, 104]. The widespread use of sequencing technologies has led to the discovery of connections between gut microbial communities and a wide range of health issues, including gastrointestinal diseases [116, 37], obesity [117, 118], immune disorders [115, 119, 120], cancer [121], and mental disorders [122]. These studies indicated that the impact of host-gut microbiota relationships could be far-reaching rather than confined to the gut. The connections are established through the comparison of sequencing data collected over time (longitudinally) or across different geographical locations (spatially) [112]. However, these findings are observational in nature. At the moment, the field is gradually shifting toward revealing the underlying metabolic or ecological mechanisms.

Perhaps the first layer to probe into is the interplay between the gut microbiota and the immune system, as the immune system is directly responsible for the regulation of the microbiota [123, 124]. Following the discoveries from association studies, the underlying metabolic dynamics could be directly addressed from multiple angles. These include using animal models such as germ-free (presence of microbes eliminated) or gnotobiotic (gut microbiota configuration pre-defined) mice to look into specific metabolic pathways, or integrating the host immune profiles to identify the causality [115]. These mechanistic studies help bridge the gap between statistical correlations and their potential therapeutical applications.

In this thesis, we focus on linking the gut microbiota to host immunity, a theme that is well-addressed by the project ImMiGeNe. Here, we will introduce the project and present some foundational knowledge in this area.

1.3.1 The ImMiGeNe project

ImMiGeNe is a project aiming to study the relationship between the gut microbiota, host immunity and genetics in stem cell transplantation (SCT) patients. This is a longitudinal study integrating multi-omics and clinical data from 20 patients and their stem cell donors. Feces, urine, and blood samples were collected, and used to obtain the fecal microbiota sequencing data, host physiological measures, whole-exome sequencing (WES), and transcriptomic data. A detailed sample collection agenda and data analysis will be presented in Chapters 3 and 4. Overall, the project focuses on three main aspects:

1. Impact that the transferred donor immune system may have on the recipients' gut microbiota

2. Linking the alterations in the post-SCT microbiome to the immunogenicity of gut microbes.

3. Matching of donor-recipient biomarkers apart from human leukocyte antigens (HLA). While donor-recipient HLA "matching" is commonly known as the prerequisite of SCT, it is unlikely to be the sole factor determining the compatibility. The project aims to explore other factors in the host genetics that have an impact on the matching.

ImMiGeNe brings together a multi-disciplinary collaboration. Our role in the project mainly revolves around questions (1) and (2).

SCT patients typically undergo intensive chemotherapy, long-term medication, hospitalization, and broad-spectrum antibiotic treatment, besides the SCT itself. Subject to these strong perturbations to their body systems, the patients are usually not in a state of homeostasis throughout the treatment process. Their immune systems have been compromised from the outset and further destroyed by chemotherapy. The unstable host physiological conditions lead to dysbiotic microbiota, which is further depleted by the heavy use of antibiotics. During the engraftment phase, both the host immune system and the gut microbiota struggle to restore balance. Tracking the development of the microbiota and examining it jointly with the host genetics and physiology can provide valuable insights into the mechanism of host-microbiome co-regulation.

1.3.2 The human immune system

The immune system is comprised of all the cells, substances, and metabolic processes that protect the human body from foreign or potentially harmful agents, such as microbes, toxins, damaged cells, and cancer cells. In vertebrates, it consists of both the innate and adaptive immune systems. The innate immune system is characterized by its quick and non-specific response, while the adaptive immune system is highly specific to antigens [125]. Upon the recognition of potentially harmful agents, innate immunity tries to eliminate invading entities by initializing physiological changes such as fever and lower pH, inflammation, and phagocytosis which neutralizes pathogens with macrophages and neutrophils [126]. Adaptive immunity, on the other hand, relies on the activation and differentiation of antigen-specific T and B cells to contain and neutralize infectious agents, and to keep an immunological memory for long-term protection against future infections of the same agent [126].

The innate immune receptors recognize a relatively fixed set of molecules, some of which are commonly found in microbes as a result of evolutionary conservation [125, 127]. It has evolved to detect microbial intruders by recognizing the shared patterns of these molecules, commonly referred to as microbe- or pathogen-associated molecular patterns (MAMPs or PAMPs), through receptor proteins known as pattern recognition receptors (PRRs). The most well-studied PRRs in mammals include the Toll-like receptors (TLRs) and nucleotide-binding and oligomerization domain (NOD)-like receptors (NLRs) [128, 125, 127]. Generally speaking, innate immune receptors are highly similar within the same species.

By contrast, the adaptive immune system comprises specialized immune cells, the T and B lymphocytes, as well as antibodies [125]. Antigen-presenting cells process the antigen and present them to be recognized by T and B cells, by the specific receptors they express (T- and B- cell receptor, TCR and BCR). The recognition leads to the activation and proliferation of the T and B cells and the production of molecules such as antibodies [129]. The specificity of receptors to antigens is generated via site-specific DNA recombination [125, 130]. These TCRs or BCRs form a repertoire to enhance the chances of detecting any antigens that the host has encountered. These immune repertoires are individual-specific, decided by factors like host genetics and the set of antigens they encounter through their lifetime [125, 130].

1.3.3 An overview of the host-gut microbiome interactions

Having co-evolved for eons, the gut microbiota and the human hosts are in a complex symbiotic (commensalistic, mutualistic, or parasitic) relationship. The gut microbiota starts to develop at birth, and continues until it reaches a stable, resilient state of a climax community in late childhood or adolescence, during which time the host immune system also develops and matures [131]. Throughout our lifetime, our gut microbiome and host immune system are closely interconnected through a network of intertwined metabolic pathways. They regulate and educate each other, and their overall stability is maintained interdependently. As a result, perturbations in one can result in significant changes to the other [115, 132].

A long-standing prevalent view was that only pathogens elicit immune responses from the host, as a defense mechanism. But later, it was proved that commensal microbes are also recognized by the immune system[133]. This transformed our understanding about the interconnections between the two entities, demonstrating that the host pattern recognition of the microbial components can also act as an integral part of daily physiological processes, and a crucial mechanism to the mutual regulation of each other and the defense against pathogens[134, 135].

Both our innate and adaptive immune systems are being molded by the gut microbiota since birth, through their metabolic communications. A number of studies have uncovered how certain microbial lineages or metabolic pathways mediate immune maturation and help shape immune cell dynamics. Mazmanian *et al.* proved in a murine model that *Bacteroides fragilis*, a pioneer species in neonates' gut, produces a polysaccharide for dendritic cells, the antigen presentation cells for mammals. This leads to the expansion of T cells and a correction of imbalanced T helper (Th) 1/Th2 cells (both differentiated from naïve CD4$^+$ T cells) [136]. Ivanov *et al.* showed that segmented filamentous bacteria (SFB) introduced to the small intestines of mice induces the differentiation of Th17 cells and enhances the antimicrobial or inflammatory immune responses, leading to increased immunity against the gut pathogen *Citrobacter rodentium*. Such effect is not observed in other lineages even if they are closely related to SFB [137]. Research has indicated that some immunoregulatory effects are not exclusive to specific microbial strains, but rather are shared by microbes that possess certain metabolite-producing abilities. For instance, short-chain fatty acid (SCFAs) is produced by a range of gut commensals via fermentation processes. Regulatory T cells (Tregs) are pivotal to immune homeostasis. Smith *et al.* demonstrated that SCFAs expand Tregs in mice and protect against experimentally-induced colitis [138]. Chang *et al.* reported that N-butyrate, a type of SCFAs, is able to mediate the macrophage activities by down-regulating the production of pro-inflammatory cytokines [139]. Wampach *et al.* revealed that compared to the gut microbiota of caesarean section delivered neonates, that of the vaginally delivered neonates are enriched with the biosynthetic pathways of lipopolysaccharide (LPS), a ligand to the TLR4. The recognition stimulates the production of interleukin 18 (IL-18) and tumor necrosis factor (TNF-α), which are pro-inflammatory mediators [140]. Such cross-talk could potentially contribute to the development of the innate immune system.

In a nutshell, the microbial community as a whole can influence the host's immune state through their fundamental metabolic activities. Meanwhile, certain microbial lineages may exert a more pronounced effect, due to their extra immunogenicity, unique metabolic capabilities, or close proximity to host mucosal cells. In ImMiGeNe, with the metagenomic sequencing data, we aim to investigate the metabolic landscape of the gut microbial communities, and also explore the lineage-specific immunogenicity by examining the flagellins produced by different microbial species.

1.3.4 Gut microbial energy metabolism landscape under host homeostasis and inflammation

The intestinal epithelial cells are responsible for nutrient uptake and immunomodulation. Gut microbes live in a race to compete for limited resources, and it is crucial to their fitness to be able to utilize what is allowed to be available by the host cells [141].

Healthy human gut is maintained as an anaerobic environment that promotes the fitness of obligate anaerobic microbes. These microbes can convert complex dietary carbohydrates into energy by fermentation, and at the same time, generate metabolites that are beneficial for the host [141]. These microbes encode a variety of glycoside hydrolase genes, which allow them to catabolize a wide range of polysaccharides for energy and therefore adapt well to the anoxic gut environment [109]. Facultative anaerobic microbes, in comparison, are able to survive, but their energy metabolism is not optimally configured for living in an environment lacking oxygen [141, 109].

Alongside fermentation, gut microbes all have certain mechanisms to respire, because compared to fermentation, respiration is much more efficient in producing energy. One main difference between the two is, respiration involves the electron transport chain (ETC), while fermentation does not. An oxidizing agent is used as the terminal acceptor in ETC, such as oxygen in aerobic and nitrate in anaerobic respiration. Strict anaerobes like *Clostridium spp.* and *Roseburia spp.* evolve to have a simple ETC that uses fumarate as the terminal electron acceptor, reducing it to succinate [142]. Fumarate is made available by gut microbes as a fermentation product. Facultative anaerobes like *Gammaproteobacteria* do not commonly have fumarate as a terminal electron acceptor; instead, they encode several families of oxidoreductases to efficiently make use of exogenous electron acceptors that are rarely available under homeostatic conditions [142, 123].

Under inflammatory conditions, changes in nutrient availability reshuffle the fitness of microbes. During inflammation, the host immune system produces a variety of antimicrobial radicals like reactive oxygen species (ROS) (such as superoxide and hydrogen peroxide), and reactive nitrogen species (RNS) (such as peroxynitrite and nitric oxide). Despite being non-toxic to the host themselves, these radicals will further form by-products that can serve as terminal electron acceptors in anaerobic respiration. They include nitrate, nitrite, and trimethylamine N-oxide (TMAO), and dimethyl S-oxide (DMSO). Using these chemicals to respire requires terminal oxidoreductases, which can be found in many facultative anaerobes but rarely in obligate anaerobes [143, 46]. And compared to fumarate respiration, through which many obligate anaerobes can respire, these are stronger oxidizing agents that release more energy. Therefore, these exogenous electron acceptors greatly enhance the fitness of the facultative anaerobes, allowing for their significant expansion and the depletion of their obligate anaerobic counterparts. Studies have also suggested that these electron acceptors allow facultative anaerobes to efficiently utilize alternative carbon sources, such as succinate, ethanolamine, and L-lactate, which are common metabolites produced by the host or obligate anaerobes [144, 145, 132]. This further reinforces the notion that facultative anaerobes can efficiently exploit the physiological changes for their own growth.

Another key factor in immune homeostasis is the low oxygen levels. The intestinal epithelial cells actively maintain the anoxic gut environment through their metabolism. One pathway involves butyrate, an SCFA that is produced exclusively through anaerobic fermentation by gut commensals such as *Clostridia* [132, 123]. The colonic epithelial cells, colonocytes, use butyrate as an energy source, which consumes a considerable amount of oxygen. This helps maintain the low-oxygen level in the colonic lumen, favoring the dominance of obligate anaerobes [132, 123]. The presence of butyrate also leads to a significant decrease in host-derived nitrate, through the suppression of the inducible nitric oxide synthase (iNOS) synthesis by the host. By providing the host with butyrate, obligate anaerobes restrict the access of facultative anaerobes to two potential electron acceptors, oxygen and nitrate, thereby limiting their expansion [123, 146].

Inflammation or antibiotic use can lead to the depletion of butyrate-producing anaerobes, thus cutting off the butyrate supply to the host. In the absence of butyrate, the epithelial cells switch to glycolysis and lactate fermentation instead, which do not consume oxygen. This leads to elevated oxygen levels and further intensifies the shifts in the metabolic landscape of the microbial communities [123]. Evidently, oxygen inhibits the growth of strictly anaerobic commensals, and promotes the expansion of facultative anaerobes like *Enterobacteriaceae*, whose most optimal way to generate energy is through aerobic respiration using the superior electron acceptor, oxygen [46, 147]. Furthermore, it has been reported in a mouse model that colitis could be alleviated using a microbiota-engineering method that inhibits certain respiratory pathways of *Enterobacteriaceae*. This implies that some facultative anaerobes may not only benefit from inflammation, but also plays a role in amplifying its effect [148].

It has been established that gut microbiome can display significant differences in composition between healthy and diseased host conditions [149, 150]. The host-microbiome correlations could be largely driven by the metabolic interactions, as demonstrated above. Investigating fundamental metabolic processes like energy production at the community level can facilitate our understanding of the mechanisms behind the correlations.

1.3.5 Bacterial flagellin and host innate immune recognition

As previously mentioned, the intestinal innate immune system is able to recognize both commensal and pathogenic gut microbes via MAMP-PRR signaling, which is a basis for the symbiotic relationship between the host and gut microbiome. But recognition of pathogens commonly leads to pro-inflammatory responses, while recognition of commensals does not. It is therefore intriguing to explore the various immune responses different microbes can elicit, i.e., their immunogenicity. One of the most well-studied MAMP-PRR interactions is the recognition of microbial flagellin by TLR5.

Flagellins are the basic units of flagella. A flagellum is a hollow lash-shaped organelle that enables the organism's motility. TLRs are transmembrane PRR proteins capable of binding to various components on microbial cell surfaces. Among them, TLR5, a protein commonly found in epithelial cells of mucosal barriers, recognizes microbial flagellins by binding to a phylogenetically conserved domain, D1, of the protein.

Both commensal and pathogenic microbes could be flagellated, but the host immune responses they stimulate could differ vastly in intensity. This can be attributed to the structural variations of the flagellin proteins, as well as the versatility of gene expression. Several studies have indicated that some types of flagellins could evade TLR5 recognition. For instance, it has been demonstrated that pathogens from alpha- and epsilon-proteobacteria have significant amino acid changes in their flagellin, which preserve the microbes' motility while evading TLR5 activation [151]. A recent study tested different types of flagellins that are commonly found in the human gut [152]. Previous studies based on the well-characterized FliC model reported that although TLR5 recognition site is at domain D1, D0 is necessary for the activation [153]. Consistent with these findings, this study pinpointed an additional binding site on the D0 domain that is necessary for the activation of TLR5. Commensal bacteria such as *Roseburia hominis* lack such site and bind sorely to the epitope instead of the full-length TLR5, resulting in weakly activated TLR5. This mechanism is referred to as "silent recognition"[152].

Some microbes carry and express only one flagellin gene, e.g., *Escherichia coli* encodes only FliC [154]. Others can carry multiple genes which may not be expressed all at the same time. For example, *Salmonella* in general has *fliC* and *fljB* that they expressed in two different phases of their life cycle [154].

The TLR5 recognition of different flagellins is an interesting illustration of how different microbes have varying abilities in influencing host immunity. The gut microbiome is potentially a reservoir of enteropathogens to the host [132]. Analyzing the diversity of flagellin can help us unravel the contributions of individual microbial taxa to the host immune responses.

1.3.6 Looking into human gut microbiome through the lens of metagenomics

Like other microbiomes, the most relevant questions related to a gut microbiome include (1) the community's taxonomic composition; (2) the functional activities of the community members; (3) the ecological factors that shape the community. A majority of human gut microbiome studies rely on sample comparison to highlight group differences, or establish the roles of certain host or environmental factors.

The fundamental questions can be effectively addressed by sequence-based methods, primarily 16S rRNA and metagenomic sequencing. While 16S rRNA sequencing remains a very popular, cost-effective, and scalable way to examine the taxonomic composition of microbial communities, it bears the intrinsic limitation of containing only the information of bacterial and archaeal 16S rRNA genes. And conventional short read 16S rRNA methods only have a decent resolution at the genus level and above [155]. Metagenomic sequencing is able to overcome these issues. Commonly, metagenomic data could reveal species-level composition, and more recently, there has been a trend of pursuing the strain-level resolution [156, 157]. Such improvement is especially relevant for human gut microbiome studies. Metagenomics enhances the accuracy and specificity of the microbiome comparison and functional analyses. For instance, Costea *et al.* identified subspecies based on the single-nucleotide variation (SNVs) analyses in a large-scale human gut microbiome dataset. The gene content analysis further demonstrated that some subspecies-specific genes, such as certain pro-inflammatory flagellum operons in two of three subspecies of *Eubacterium rectale*, may be linked to some key factors like microbiome diversity and host BMI [158, 112]. This suggested that higher resolution provided by metagenomic data can shed light on some overlooked microbiome functions and host-microbiome interactions that happen at more refined taxonomic levels.

With metagenomic data, one could also reconstruct the constituent genomes. MAGs are particularly useful for genomic comparison and gene screening. MAGs from large projects like MetaHIT [159] and Human Microbiome Project [5, 4] has contributed considerably to our current understanding of the baseline compositions and functions of the gut microbiome [112].

Metagenomic data allows for in-depth functional analyses, as they are meant to target all gene contents of a sample. The functional potentials of a microbiome can be estimated by gene prediction and annotation. Long sequences such as assembly sequences and certain kinds of long reads may contain multiple genes. Gene prediction is used to find out the coordinates of each possible gene [40, 39]. Annotation assigns functional information to a given sequence. A query sequence can be annotated by mapping it to functionally-annotated reference sequences [160]. Alternatively, genes can also be directly assembled from reads [161]. Functional analyses boosted comparative analyses between metagenomes, providing valuable guidance for inferring the mechanisms behind their compositional differences. For instance, Chng *et al.* compared gut microbiomes that either did or failed to recover from antibiotic treatments. They identified key species crucial for microbiome recovery. By combining this finding with the functional analyses, they hypothesized that the recoveries were related to the enrichment of carbohydrate metabolism, which was supported by subsequent experiments using a mouse model [162].

While metagenomics provides a comprehensive genomic landscape of given microbiomes, its functional readouts only indicate the potential existence of a set of genes, and no information regarding their expression levels or their roles in biological processes. Furthermore, despite our relatively complete taxonomic knowledge of the human gut microbial species, a significant proportion of the genes predicted in the metagenomes are still uncharacterized [112]. Correlations drawn from metagenomic analyses are also only indicative and serve as a guide for generating hypothesis, but do not establish causation nor imply directionality of the impact [112].

In recent years, human gut microbiome studies are increasingly compensating these limitations by employing metagenomics in conjunction with other meta-omics methods, such as metatranscriptomics, metaproteomics, and metabolomics, as well as perturbation or intervention experiments in vitro and in vivo [37, 163, 164, 165]. These methods are able to provide more direct profiles of gene expression, proteins, and metabolites. Hypotheses based on the metagenomic analyses can be tested using animal models or human interventional experiments [46, 163, 115].

1.4 Outline of the projects

Three main projects are involved in this thesis. Chapter 2 focuses on a software development project, in which we developed a tool named Mapbin, for metagenomic binning refinement. The chapter first demonstrates the core algorithm behind Mapbin, the Python implementation, and the performance on synthetic and real-world datasets. Chapter 3 is about the microbiome data analysis in the ImMiGeNe project. In this chapter, we will explain the data processing and taxonomic and functional analyses first, linking them to the biological theme of ImMiGeNe. We will then evaluate the methodology of the data analysis, including its practicality and potential limitations. In Chapter 4, we will design a list of PCR primers suitable for batch amplification of bacterial flagellin in the microbiome samples from ImMiGeNe. We will elucidate computational challenges associated with primer design, present various algorithms and techniques to solve them, and demonstrate the application of these methods in our specific case. Altogether, this thesis covers a wide range of topics in metagenomics, including theories, algorithms, and real-world applications.

Chapter 2

Mapbin: versatile refinement of metagenomic binning using multilayer networks

2.1 Introduction

2.1.1 Community detection algorithm Infomap

Networks are instrumental in representing large-scale data of complex systems. A network is a form of graph composed of edges and nodes. To schematize the organization structure of a complex system, we often use the nodes (or vertices) to represent the objects in the system and the edges (or links) to represent their connections or interactions. By analyzing the network, the structure of the system can be revealed. In the binning problem, we could model the sequences and the connections between them as a network. The binning problem is then solved on top of a network structure, i.e. network clustering. The goal is to find communities in the network. A community is a subset of nodes (which makes a subgraph) that are more close-knit, with edges connecting them internally significantly denser than the external ones. They are also called modules or clusters. By detecting the communities in a network, the network gets divided into subgraphs, and therefore it is also termed network partitioning[166, 167].

A network can be partitioned in countless ways, with some depicting the underlying structure better than others. Generally speaking, community detection algorithms search for possible partitioning and evaluate which is the best. The evaluation is usually via an objective function, and the search process aims to optimize it. The search and evaluate methods define an algorithm[166, 167, 168]. Infomap is one community detection method that searches for partitioning by tracing the information flow, using an objective function which is called the map equation[98, 169, 170].

Network flow is a concept from information theory. The structure of the network can be depicted by tracing the information flows through the network, which is driven by the interactions between nodes[171, 168]. Infomap captures the flow by recording the path of a random walk across the network. In short, the walk is initiated at a random node, and the probability of its possible next move is determined by its edges and their weights. The random walker can also teleport, which means at each step, it can jump to a random node on the network with a predefined probability. This is to ensure the walker does not get stuck due to node disconnection so that it can visit all nodes in a finite number of steps. The path of the walk is encoded in bit strings. In a given partition, the encoding scheme, or a codebook, is based upon the community structure. Each node is given an address as if on a map. The moves between nodes in the same community can be described with fewer bits compared to those between communities. Because the interactions inside a community are supposed to be denser than the outside, a good partitioning will end up using shorter strings, as it is more faithful to the true community structure. In other words, it is able to achieve better compression of the path description. The cost function of Infomap, the map equation, measures the efficiency of the coding length for a given partitioning[170, 169]. Infomap is agglomerative. It is initiated with every node as its own cluster, then proposes a move by clustering a node to its neighbor. The proposal is accepted if the encoding cost is reduced and rejected if not. Each accepted proposal rebuilds the network, with the newly accepted cluster making a node of one level higher, in replacement for the nodes constituting it. The clustering stops when no moves can reduce the coding lengths. The resultant clustering is hierarchical[98, 169]. The general conceptual framework is shown in Figure 2.1 (a).

Infomap has the backbone of Louvain algorithm, the famous community detection algorithm using a random walk[169, 168]. The algorithmic modification was meant to circumvent the clustering of poorly linked nodes, which is a major disadvantage of Louvain algorithm. As the communities are generated in a dynamic manner, a clustering proposal that seemed optimal in an early stage might turn out to be disadvantageous later on[97, 169]. In Louvain algorithm, the clustering is barely reversible, and Infomap algorithm tries to allow some changes in the clustering to alleviate such a problem. The cost function is also part of Infomap's originality[98]. By comparison, Louvain's objective function measures modularity, a concept that can be generally interpreted as the contrast between intra- and inter-community edge density[168].

2.1.2 Multilayer network analysis with Infomap

The model above, which is sometimes referred to as the first-order network, depicts only dyadic relations between each pair of nodes[172, 173]. For complex systems, this can be insufficient. Two nodes can have multiple links of different natures, or links that are present at different times. In the binning problem, as discussed earlier, two nodes of contigs can be linked due to their similarity in sequence compositional patterns and abundance profiles, or they have an overlap in the assembly graph. One possible way to model these links of different types is to construct a multilayer network. Multilayer network is a genre of high-order networks that can come in several different shapes[173, 174]. It can model multiple types of interactions between exactly the same set of nodes (multiplex networks), or interactions that happen at different time points between the same set of nodes (multi-slice networks), and also that between different sets of nodes (network of networks)[175, 176]. In Infomap, all these multilayer networks are modeled with a uniform approach. Infomap introduces the notion of physical and state nodes for objects in the complex system. The physical nodes represent the objects themselves, and the state nodes are used to model the interactions. An object can have one physical node and multiple state nodes if its interactions with others are from different data sources. Each layer represents one data source[173].

Just like in the first-order networks, the random walker chooses its next node to visit based on its edges. Note that the walker travels between state nodes. The edges of a state node can be intra- or inter-layer. Let α_i be the state node of a physical node i that is on layer α, $\alpha_i\beta_j$ the edge between state node α_i and β_j. The move via $\alpha_i\beta_j$ is inter-layer when $\alpha \neq \beta$, and intra-layer when $\alpha = \beta$. But because in reality a lot of data come without inter-layer links, the model also allows the walker to move between layers via state nodes of the same physical node at a certain relax rate r. In other words, in such a network where the inter-layer links are missing, the walker moves within the same layer at probability $1 - r$, following the edges of state nodes. It jumps to a different layer at probability r, following the edges of any state node of the current physical node. And finally, just like in the single-layer network, the walker can teleport at a rate, so that every state node can be visited[172, 173].

The algorithm still tries to minimize the code length needed to describe the random walker's path. The objective function is still the map equation at its core, but the path directly involves the state nodes instead of the physical ones. Or, to put it simply, the goal is to cluster the state nodes. And as a result, the output clusters can overlap, assigning the state nodes of the same physical nodes into different clusters[172, 173]. Figure 2.1 (b) illustrates the conceptual framework of the Infomap multilayer network partitioning.

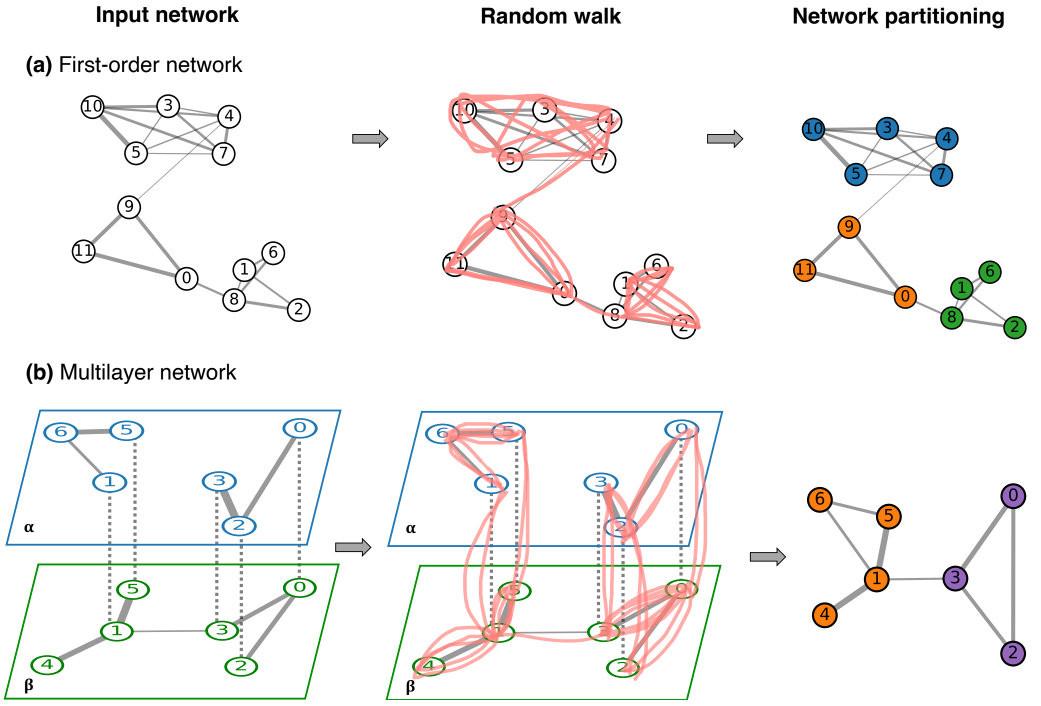


Figure 2.1: Framework of the Infomap algorithm. (a) A basic first-order network partitioning, adapted from [98] (b) Multilayer network partitioning, adapted from [172]. There are 11 and 7 physical nodes in (a) and (b), respectively. In (b), except for Nodes 4 and 6, all nodes have two state nodes.

2.1.3 Using Infomap multilayer network to cluster metagenomic contigs

Here we present a binning algorithm, Mapbin, that uses the Infomap multilayer network partitioning algorithm to refine metagenomic binning. Mapbin models the metagenomic contigs as the nodes in the network. Transforming a user-provided binning result into a base layer, Mapbin constructs additional layers of the network with either the assembly graph, the read pairing from paired-end sequencing data, or both, and calls the Infomap algorithm to obtain new clustering of the input contigs. In this chapter, we will first demonstrate the algorithm of Mapbin in detail, and then showcase the performance of Mapbin using several datasets of various data volumes.

2.2 The algorithmic framework of Mapbin

A majority of binning programs use TNF and abundance features of the contigs. Mapbin is designed to refine their binning results by taking into consideration two additional sources of information regarding the relationships between contigs, namely the assembly graph and read pairing provided paired-end sequencing data of the same metagenome exist. Mapbin transforms such features of the contigs into a multilayer network and relies on the Infomap algorithm to partition the nodes in the network. It follows three key steps: (1) parse the input contig and binning data and construct the multilayer network, with nodes representing contigs; (2) call Infomap algorithm and cluster the nodes; (3) create new bins based on the partitioning result.

In the first step, Mapbin handles at maximum three types of associations between contigs, the binning result, the assembly graph, and the read pairing information. Mapbin tries to model the contig connectivity as a layer. For the binning result, Mapbin connects contigs in each bin as a grid subgraph. For the assembly graph, Mapbin reformats the graph using contigs as the nodes, and the overlaps between contigs as edges. Assembly graphs produced by certain assembly tools, such as SPAdes, feature segments as the basic unit, and contigs are generated from paths of segments. In this case, the reformatting computes the connectivity of contigs based on their paths. For the read pairs, Mapbin requires as input the alignment of paired-end reads to the contigs and creates an edge between two contigs when a pair of reads are aligned to them.

Mapbin builds the network layer by layer. Users are free to choose to use any combination of the three features. Given a binning result together with paired-end read-to-contig alignment, the assembly graph, or both, Mapbin will use the latter to refine the former. This shall be the most regular usage of Mapbin. Other usages are not recommended for the common binning practice, although they may assist certain sequence analyses or fulfill certain testing purposes. Given only a binning result, Mapbin constructs a first-order network that returns exactly the same clustering as the input. Given only the alignment or assembly graph, Mapbin outputs the clustering based on only the sequence connectivity, which may work for simple communities with unrelated organisms.

A summary of the overall algorithm is shown in Figure 2.2. A detailed description of the network-building algorithm is illustrated in the following sections.

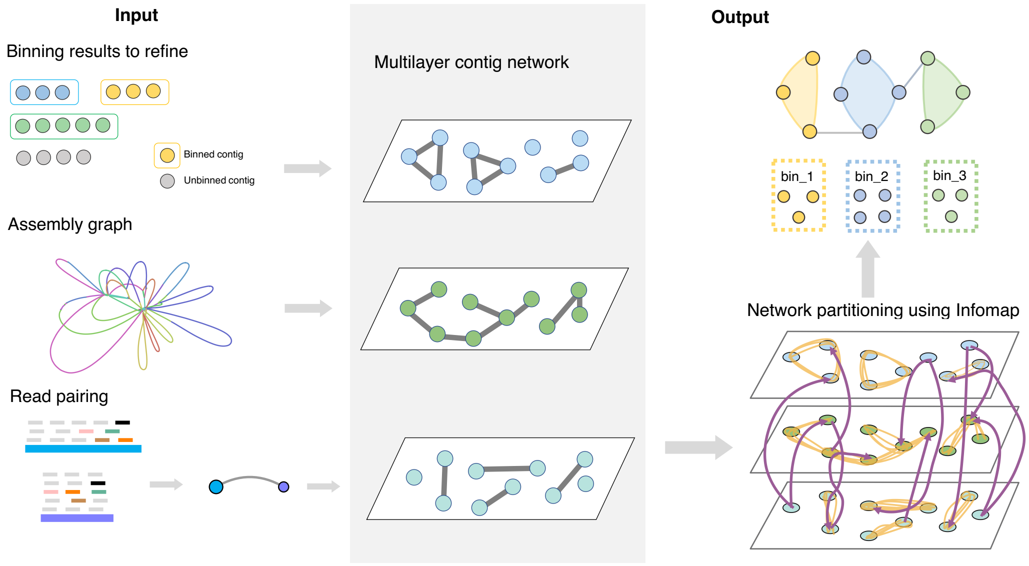

Input
Binning results to refine
Binned contig
Unbinned contig
Assembly graph
Read pairing
Multilayer contig network
Output
bin_1
bin_2
bin_3
Network partitioning using Infomap

Figure 2.2: The conceptual framework of Mapbin.

2.2.1 Mapbin network construction: the binning layer

The basic algorithm to construct the binning, assembly, and read-pairing layers is presented in Figure 2.3. To refine a binning result, Mapbin requires a directory of bins as input. Contigs of the same bin shall be concatenated as one fasta file. For each bin, Mapbin makes a connected subgraph that is close to a two-dimensional lattice graph (see Figure 2.3 (a)).

Given a bin of n contigs, Mapbin makes a subgraph consisting of n nodes representing these contigs:

- if $n = 1$, only add a single node;
- if $n = 2$, add the two nodes and pairwise directed links;
- if $2 < n \leq 5$, add the nodes and create a Hamiltonian cycle using all nodes;
- if $n > 5$, add the nodes and make a subgraph close to a triangular lattice graph, using all nodes. In this case, the number of rows is calculated as $n_{row} = \lceil \sqrt{N} \rceil$, and the number of columns $n_{col} = \lceil n/n_{row} \rceil$. All rows shall have n_{col} nodes but the last one, which may have less.

With this method, we could represent an existing binning as a set of disconnected subgraphs, all of which are lattice-graph-like. When given only the binning result, with such network topology, Infomap is able to produce identical clustering results in a time as short as under a second. This means the binning result can be accurately integrated into the new analysis. We argue that this topology is practical because, firstly, each subgraph is or is close to being regular, so every member in the bin is impartially connected. Secondly, the cost of construction of such a graph scales linearly with the number of nodes. This is important for the binning problem, because the number of contigs in a bin can range from one to thousands. And finally, the entire graph remains sparse, so the clustering algorithm could work out a solution fast even with a large number of contigs or bins.

Some input bins may contain contamination, and linking the falsely-placed contig to other contigs may reinforce the contamination. To alleviate this problem, we constructed the lattice subgraphs with the edge weight and node ordering not randomly assigned. The nodes are sorted by their estimated coverages, and edges are only added between two neighboring nodes if their coverages are relatively close (coverage of the higher no more than twice that of the lower), as contigs with disparate coverages are unlikely to belong to the same genome. Two edges of opposite directions are added between each pair of neighboring nodes.

It is widely agreed that in the binning problem, shorter contigs are less likely to be able to generate strong and reliable sequence features[95, 83, 99].

Therefore the edge weights are assigned proportionally to the contig length of the source node.

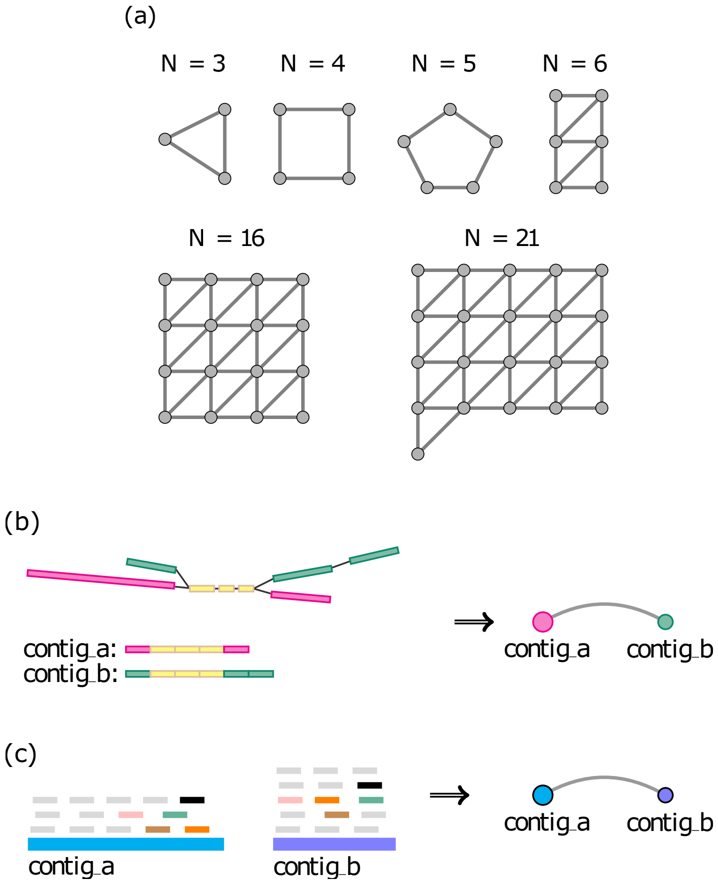


Figure 2.3: Construction of network layers in Mapbin. (a) Representation of the initial bins in the binning layer. The nodes represent contigs. Mapbin constructs a Hamiltonian cycle for a bin when it contains no more than 5 contigs, otherwise a lattice-graph lookalike. (b) Creating edges based on the assembly graph. The contigs are presented as a path of segments in the assembly graph. The magenta and green segments are unique to the two contigs, and the yellow ones are shared. An edge can be created between two contigs when their paths intersect or overlap. (c) Adding edges in the read pairing layer. Thick long lines on the left represent contigs and short dashes of the reads. Highlighted reads of the same color belong to the same pair but are found on different contigs.

2.2.2 Mapbin network construction: the assembly layer

The assembly layer network is constructed by parsing the assembly graph. Mapbin first computes all the possible linkages between a pair of contigs by comparing their paths. Paths of contigs consist of sequence segments. Between a pair of contigs, an edge can be established either from the segments they share or from the segment overlaps. For the segment overlaps, Mapbin considers only those that happen at the ends of the two contigs and ignores internal segment overlaps. Internal segment overlaps occur at a high frequency, and most of them are a result of sequence similarity at short local regions. Compared to the end overlaps, they lack specificity for the purpose of our network. This strategy is visualized in Figure 2.4.

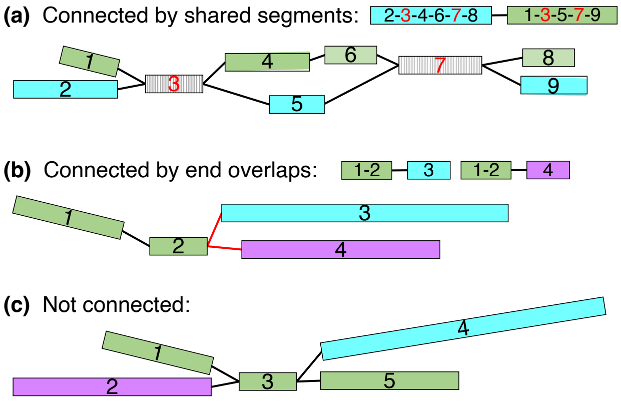


Figure 2.4: Assembly parsing strategy of Mapbin. The bars are segments of contigs, and lines are overlaps between segments. Segments from the same contig are shown in the same color. (a) Contigs are linked by shared segments in their paths. The striped bars, 3 and 7, are shared by the two contigs. (b) Two links are established between three contigs by the overlaps at their ends (highlighted in red). (c) The overlaps between contigs only happen at Segment 3, which is internal for the green contig, and no links are established.

When parsing the graph, Mapbin rejects links between contigs that seem to differ significantly in coverage. The coverage is estimated from the segment coverage recorded in the assembly graph. Given a pair of contigs, C_a and C_b, consisting of segment sets $S_i = \{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ and $S_j = \{s_{j1}, s_{j2}, \ldots, s_{jk}\}$, the contig coverages cov_a and cov_b are estimated as the median coverages of S_i and S_j, respectively. The suggested link between C_a and C_b will be rejected if cov_a and cov_b differ more than a user-defined threshold (default: 30), or $\left|\frac{cov_a - cov_b}{cov_a + cov_b}\right| > \frac{1}{5}$.

Once the assembly graph is parsed into edges between contigs, Mapbin performs a round of filtering to get rid of spurious edges. The filtering strategy is explained in a later section.

2.2.3 Mapbin network construction: the read-pairing layer

In paired-end sequencing, reads in a pair are sequenced from the same genomic fragment. Therefore two different contigs can be potentially bridged if a pair of reads are found to be on them respectively. Similar to scaffolding, the connection can be established between contigs even though the exact fragment bridging them is unknown. Mapbin uses contig-to-contig alignments to locate reads on the contigs, then computes the links between contigs by analyzing read pairs that fall on different contigs. And unlike it is in scaffolding, there is no need for computing the relative orientation of the paired contigs in the binning problem.

The read-pairing layer is established by parsing the alignment file in SAM or BAM format. The weight of a contig link reflects the length of the interval where the bridging read pairs occur. That is, given a pair of contigs, C_a and C_b, let a read pair that bridge them be r_i and r'_i, and denote their start and end positions on the contigs as (s_i, e_i) and (s'_i, e'_i); C_a and C_b are linked by a set of read pairs $\{(r_1, r'_1), (r_2, r'_2), \ldots, (r_k, r'_k)\}$, which is sorted by their alignment start positions s_i, and the linked interval between them is calculated as $l_{ab} = e_k - s_1$ and $l_{ba} = e'_k - s'_1$.

For each pair of potentially linked contigs, Mapbin compiles the set of read pairs they share and decide whether to add an edge between them. A pair of contigs sharing less than 3 read pairs are not considered significant enough to be linked. Edges will also not be added if the read-pairing is likely a result of sequence similarity. The read-pairing network is established under the assumption that the two reads in a pair come from the same source genome. However, as local similarity can be frequently found even between very remotely related genomes, reads in a pair may not necessarily be aligned to the same genome. To avoid establishing edges due to secondary alignments, two contigs are not to be linked if the linkage can be established from read-pairing at multiple non-consecutive regions of the contigs, or if it does not happen at the ends of the contigs. This strategy is illustrated in Figure 2.5.

Similar to the generation of the assembly layer, Mapbin performs for the read-pairing links a round of filtering to remove spurious links. The details are explained in the following section.

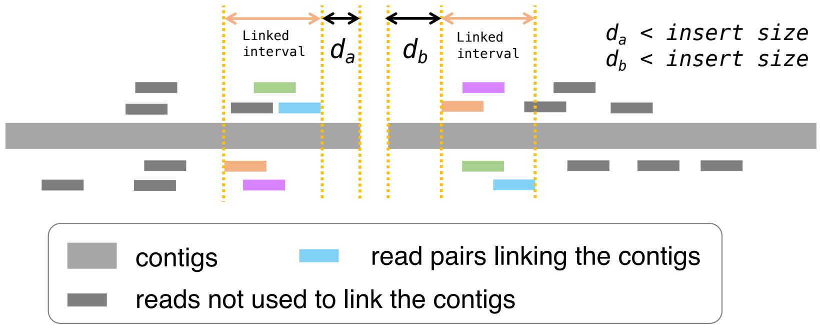


Figure 2.5: Mapbin's strategy to establish contig-contig links from read alignment. Mapbin selects for each contig a consecutive interval located near the contig ends that is covered by reads. The distances of the intervals to the ends, d_a and d_b shall not exceed the insert size. Note that the contigs may overlap.

2.2.4 Network edge filtering before clustering

The construction of the assembly or read-pairing layers is followed by a quality control step to filter out spurious edges and correct the bin assignment of some contigs. The construction of the binning layer follows subsequently. Quality control of edges is implemented for two main reasons. First, the contig linkages suggested by assembly graph and read-pairing come from a small fraction of the contigs and they do not reflect the full-length characteristics of contigs like TNF and sequence abundance. Both of the two layers alone have a limited capacity to resolve closely-related genomes. Second, compared to the binning layer, both the assembly and read-pairing layers are much sparser. In sparse networks, spurious edges have a more pronounced impact on the network topology compared to dense networks. Consequently, the presence of these edges can strongly mislead the final clustering output.

Aware of these issues, Mapbin preprocesses the potential edges by leveraging the provided binning result and other sequence attributes. Mapbin performs two rounds of filtering, at the contig level and at the bin level, respectively, with the goal of removing hubs from the network. A hub refers to a node in a network that has a number of edges that is significantly greater than the average. Hubs are a natural topological structure of networks, but in the context of genome binning, they arise mostly due to repeats or regions of high local similarity rather than authentic sequence connectivity.

In an ideal scenario, a repeat-free contig can have at most two edges with other contig nodes. A complete, contamination-free genome bin has only internal edges between its constituent contigs, and a semi-complete bin associates only with bins of the same genome. The number of external edges a bin has roughly correlates positively with the number of contigs it contains. In reality, binning programs often produce a certain number of contaminated bins. For them, the more sources of contamination, the more number of other bins it can be connected to, and the more likely it is to form a hub. A heavily contaminated bin with contigs from all bacterial genomes has the potential to be linked to all the bacterial bins at present.

A hub, at either the contig or bin level, enhances network connectivity and facilitates the integration of clusters within the network. However, in the genome binning problem, they increase the risk of bridging bins of different genomes. The Infomap algorithm is based on the information flow of the network, and a hub will naturally attract flow and induces the formation of a supercluster between loosely connected clusters in highly sparse networks like the assembly and read-pairing layers. An example of a hub bin bringing together multiple disconnected bins is shown in Figure 2.6.

Apart from the hub, Infomap may also cluster weakly connected small clusters into one top-level cluster in the presence of a large cluster. This is a prominent issue in metagenomic binning. Sizes of bins from the same dataset can vary drastically, from small bins of less than 10 contigs to large bins of thousands of contigs, leading to the occurrence of such topology. An example of this problem is shown in Figure 2.7.

Based on this, Mapbin implements a filtering strategy to circumvent the formation of misleading network topology, tailored specifically to work with the Infomap algorithm. Mapbin detects and removes edges of a hub firstly at the contig level. Mapbin computes the degree of each node, i.e., the number of edges linked to the node, and treats nodes with a degree greater than five as a hub. Contig hubs happen frequently in the assembly network due to

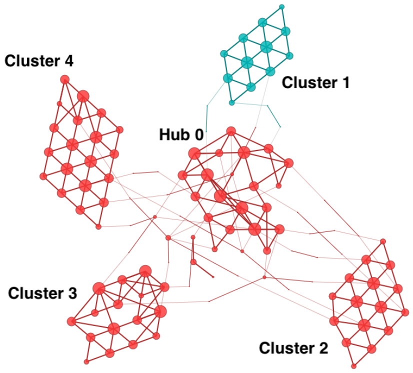


Figure 2.6: Example of Infomap algorithm applied to a 2-layer subgraph involving a hub. Nodes and edges in the two layers are flattened onto one plane. The contigs are shown in small circles with the size reflecting the flow. Each color denotes a top-level cluster in Infomap's solution. Clusters 2, 3, and 4 all interact with the hub but share very few edges with each other. With the presence of the hub, Infomap clustered 0, 2, 3, and 4 into one large cluster, which will likely end up as a combined and heavily contaminated bin. The figure is generated by [177].

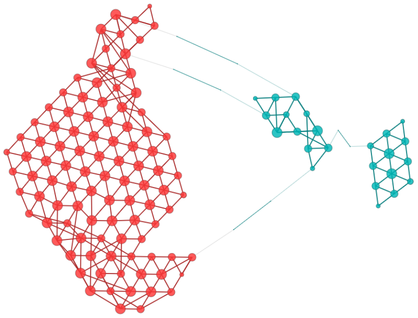


Figure 2.7: Example of Infomap algorithm applied to a 2-layer subgraph involving two loosely connected small clusters and one large cluster. Nodes and edges in the two layers are flattened onto one plane. The contigs are shown in small circles with the size reflecting the flow. Each color denotes a top-level cluster in Infomap's solution. The two small clusters in green only share one edge. Due to the presence of the large bin in red, they are assigned by Infomap to the same cluster, forming a combined and potentially-contaminated bin. The figure is generated by [177].

local regions of similarity. All edges linking to a hub node will be removed. Moreover, a contig that was assigned to a certain bin, B_x, will be reassigned to Bin B_y, if it has more than three edges and all its edges are formed with contigs from B_y.

Next, the pre-clustered bins are treated as high-level nodes, and have their linkages inspected by Mapbin to detect hub bins.

We consider a bin B_x to be significantly connected to another bin B_y if the number of external edges of B_x to B_y is above $0.05 \cdot n_x$, with n_x denoting the number of contigs in B_x. A bin is considered a hub if it is linked to more than 3 bins (regardless of being significant or not), or significantly linked to more than one bin. We argue that a highly incomplete but pure bin is unlikely to be falsely flagged as a hub, because they also contain less genomic content, which heavily limits the number of both internal and external edges they may have. It is also proven in the benchmarking step, which we will elaborate on later on, that the hub bins Mapbin detects are mostly truly contaminated.

Hub bins are detected by checking the external edges of each bin. Upon detection, the hub bin will be resolved by either removing the contig from the bin, or by removing their edges. If a hub B_h has signigicant edges with B_x, all contigs linked to B_x will be removed from B_h to form a new bin B_{hx}. In plain words, in this case, the subgroup which are all linked to the same other bin will be severed from the initial bin and the subdivision will participate in the final network clustering. Mapbin requires a minimum number of external edges to be present for the removal of contigs from the hub to prevent rigorous removal at the cost of bin completeness.

Further, for non-hub bins, if two bins are not significantly connected, the edges between them will also be deleted.

2.2.5 Inspecting the contig length distribution

Binners often face the challenge of clustering highly fragmented genomes that consist of hundreds or even thousands of contigs. This problem commonly arises when complex genomes are sequenced with insufficient coverage. Due to the limited sequence contiguity throughout the genome, the majority of the assembled contigs are relatively small (e.g., under 10kbp). Generally speaking, misclustering rate is higher among small contigs due to their less distinctive, stable, and reliable sequence features [96]. Additionally, during the clustering step, a large cluster formed by these small contigs can falsely recruit longer contigs, further compromising the quality of the resulting bin.

We implemented in Mapbin a step to inspect the contig length distribution in bins with more than 50 contigs. This will detect contigs whose lengths are extreme high outliers in the bin. Our rationale is that the great majority being short contigs suggests a fragmented assembly of the underlying genome. In this case, it is highly improbable to successfully assemble an ultra-long region of the genome while failing at all the remaining parts. Therefore, the ultra-long contigs are more likely derived from other source genomes.

To determine the contig length upper bound for a bin with n contigs (where $n > 5$), we calculate the lower q-th percentile Q_l and higher $(100-q)$-th percentile Q_h, in which $q = min\{0.05, 50/n\} \cdot 100$. The upper bound for contig lengths l_{max} is then set as $l_{max} = Q_h + 50 \cdot (Q_h - Q_l)$. Contigs whose lengths are above the upper bound are identified as high outliers and will be removed from their assigned bin. Note that this step specifically targets contigs with extreme deviation in length within bins that are evidently from highly fragmented underlying genomes. Longer contigs within a reasonable length range are not affected. This step is designed to address the contamination caused by ultra-long contigs, which can lower the bin quality greatly due to their substantial contribution to the bin's base count. We do not rely on the previously described network-based contamination detection step to identify the misplacement of these contigs, because the networks only involve a small fraction of contigs and may not include them.

Mapbin also extracts ultra-long contigs from the set of unbinned contigs after the network partitioning step, and outputs each of them as an individual bin. The default minimum length to form a standalone bin is 1 Mbp.

2.2.6 Infomap network partitioning and the generation of bins

Once the network is formed, the core Infomap algorithm is called to perform the partitioning. In short, nodes will be assigned to modules by Infomap. Each module forms a bin.

Overall, Mapbin's method to increase bin completeness is by targeting genomes that are divided into multiple bins. Mapbin is not designed to target each individual contig. Mapbin rearranges the bin assignment of contigs by moving them in clusters rather than individually. By nature, both the assembly and read-pairing networks are only able to involve a minority of all contigs. The rest usually have no overlaps with any other contig and therefore have no edges with others in these two layers. If a contig is isolated in the two layers, its placement usually follows its original bin assignment.

2.3 Mapbin implementation

Mapbin is an open-source software implemented in Python. The git repository is available on GitHub: `https://github.com/u-xixi/mapbin`. A brief user manual is included in the repository.

Mapbin requires Infomap Python module with Version 2.0 or above, and the Pysam module with a recent version (e.g., 0.18.0 and above). Users need to provide the input contig sequences and output directory. A pre-computed binning result shall be provided to perform refinement using assembly graphs or read pairing. For assembly graph-enhanced binning, users need to provide the assembly graph. Currently, Mapbin is designed to work with short-read assembly generated by SPAdes [21], requiring the graph in GFA format. SPAdes only output the GFA formatted graph for scaffolds. If the input sequences are contigs, the contig path file shall also be provided. For read pairing-enhanced binning, the user shall provide the coordinate-sorted, indexed alignment file in SAM or BAM format. Both single and multiple alignment files are acceptable by Mapbin.

It is also important to clarify a few trivial details regarding the implementation of Mapbin.

2.3.1 Overlapping bin output of Mapbin

Unlike many conventional binning algorithms, bins produced by Mapbin could be overlapping. Metagenomic assemblies usually contain a number of short contigs, e.g., under 3000 bp. Fragments of such length could possibly be shared among multiple genomes. Mapbin enables overlapping output to allow certain fragments to be shared by different genomes.

The overlapping bin output is enabled by the algorithm of Infomap. As introduced previously, in the multilayer network, there are physical and state nodes. State nodes are generated based on the physical nodes, and the network partitioning is based on the state nodes only. The overlapping happens when state nodes of the same physical node are assigned to different modules.

Non-overlapping output can also be produced by randomly choosing a bin for a shared contig or completely removing it from the binning result.

2.3.2 Other details

Infomap gives hierarchical clustering results, and Mapbin takes only the top modules as the basis of bins.

Infomap could detect trivial modules with few nodes. But in the genome binning problem, a bin with few very short contigs may not be of interest. Mapbin allows users to set a lower limit of total bases contained in a bin, below which the bin will not appear in the final output.

2.4 Benchmarking Mapbin and other binners

2.4.1 Benchmarking dataset

We ran performance tests on two datasets simulated by ourselves, five synthetic human microbiome datasets from the CAMI challenge, and four realistic human gut microbiome datasets. Five other publicly available tools are included in the benchmarking: MetaBAT2, CONCOCT, VAMB, MaxBin2, and GraphBin2.

Simulated Datasets ***Random*** and ***Half-random***

The datasets *Random* and *Half-random* were generated by simulating paired-end reads from a set of complete genomes available on NCBI, using InSilicoSeq [178]. The abundances of the chromosomes of the genomes were randomly sampled from a log-normal distribution. If plasmids are present, their abundances are k times that of the chromosome, with k randomly drawn from a uniform distribution between 1.5 and 2.5. The reads were assembled by SPAdes v3.15.5 [21]. The contigs were used for binning, and ground truth bin assignment was made by mapping the contigs to the source genomes.

Both the *Random* and *Half-random* datasets contain 50 microbial genomes, among which there are 32 bacteria, ten archaea, five viruses, and three fungi. Each dataset contains 100 million reads. The source genomes of the *Random* dataset were randomly chosen from a set of reference-quality microbial genome assemblies in the NCBI database. All these genome assemblies are complete, and most are either listed as a reference genome by NCBI or were assembled from type strains. The Half-random dataset contains 25 genomes from predefined microbial species, and the other half was made by random selection. The predefined 25 genomes are from the bacterial species *Escherichia coli*, *P. aeruginosa*, the bacterial genus *Nostoc* and *Rhizobium*, as well as the archaeal genus *Methanobacterium*, five genomes each taxon.

A detailed list of genomes used to simulate the two datasets are listed in Supplementary Tables A.2 and A.3.

CAMI II toy human microbiome datasets

We used the five synthetic datasets from the second Critical Assessment of Metagenomic Interpretation (CAMI). They are toy microbiome of 5 human body sites: gastrointestinal (GI), airways (air), oral, skin, and urogenital (urog). All data are available at `https://data.cami-challenge.org/participate`. We used the gold standard assembly as the input contigs. As the gold standard assemblies were provided without assembly graphs, they were used to demonstrate Mapbin's read pairing refinement. A gold standard bin assignment is available for each of these datasets.

Four real datasets

The four real-world datasets are human gut microbiome DNA samples prepared and sequenced on Illumina platforms by our collaborators. *Mock* is a mock community of 15 commercially available bacterial and archaeal strains commonly found in human gut microbiomes. To highlight the challenge of closely-related genomes, *Mock* was designed to have ten strains from the same genus, *Bifidobacterium*. The strains used in *Mock* are listed in Supplementary Table A.4.

Sample2, *Sample3* and *Sample4* are samples from the TwinsUK project (see `https://twinsuk.ac.uk/`). *Sample2* and *Sample3* were originally the same sample but differed in that *Sample2* was selected for fragment size between 8 to 20 kb, while no size selection was performed for *Sample3*. *Sample4* is a different community with a size selection between 8 to 20 kb. For them, the ground truth binning is not available.

2.4.2 Assembly statistics of the benchmarking datasets

The basic statistics of all benchmarking datasets are summarized in Table 2.1. Contigs with lengths above 1500 bp were used as the input sequences for the benchmarking. In the calculations, L50 and N50 are measures for the assembly contiguity. Given a set of genomic assembly, L50 is the minimum number of contigs from this set required to cover up half of the total assembly size. And N50 refers to the size of the smallest contig included in the minimal set of contigs that make up half of the total assembly size, or in other words, the length of the contig that separates the assembly into the larger and smaller halves. For example, in the original airways dataset, 77,837 largest contigs are needed to cover up half of the total size, which is 1,938,277,766 bp. And the smallest among the 77,837 has a length of 2555 bp. Therefore, its N50 is 2555, and its L50 is 77,837.

Table 2.1: Basic statistics of the assemblies from the benchmarking datasets

	No. samples	No. contigs (K)	Total bp (10^6)	L50	N50	Longest (Mbp)	Avg. GC
Original Random and Half-random							
random	1	26.4	273	301	177,663	3.34	0.52
half-random	1	35.3	265	258	179,434	3.49	0.50
Filtered Random and Half-random (> 1500 bp)							
random	1	10.5	265	278	189,514	3.34	0.52
half-random	1	7.6	252	223	195,969	3.49	0.50
Original CAMI							
air	10	1,971	1,938	77,837	2,555	6.19	0.48
GI	10	211	933	120	1,949,862	6.53	0.48
oral	10	1,287	1,670	15,593	7,227	5.50	0.44
skin	10	753	1,527	13,037	9,045	5.64	0.46
urog	9	207	666	143	660,741	7.12	0.49
Filtered CAMI (> 1500 bp)							
airways	10	162	1,131	3,093	26,695	6.19	0.47
GI	10	38.6	848	102	2,807,504	6.53	0.48
oral	10	123	1,128	609	120,855	5.50	0.43
skin	10	156	1,228	2,292	30,599	5,64	0.45
urogenital	9	38.2	577	104	1,667,350	7.12	0.48

Original real datasets							
Mock	1	44	71	245	33,813	1.22	0.57
Sample 2	1	744	660	43,465	1,758	0.92	0.49
Sample 3	1	739	646	44,926	1,673	1.10	0.49
Sample 4	1	760	672	49,775	1,733	0.68	0.49
Filtered real datasets (>1500 bp)							
Mock	1	3.8	54	109	113,096	1.22	0.57
Sample 2	1	53	346	3,375	15,721	0.92	0.48
Sample 3	1	52	334	3,365	15,371	1.10	0.48
Sample 4	1	60	352	4,545	11,593	0.68	0.48

2.4.3 Benchmarking tools and metrics

For datasets Random, Half-random, and CAMI datasets, since the ground truth binning is available, we used AMBER 2.0.3 to assess the qualities of the binning results[179]. AMBER is a binning evaluation toolkit originating from the CAMI challenge. It provides the calculation of common metrics such as accuracy, precision, completeness, and contamination, as well as data visualization. For the real-world datasets, due to the lack of a gold standard, we used CheckM 1.2.2 to estimate the quality of the binning results. CheckM [101] is a toolkit for the evaluation of genomes reconstructed from all kinds of genomic sequencing data. The estimation is based on single-copy marker genes that are specific to and ubiquitous within a phylogenetic clade. The clade can be of different taxonomic ranks.

The binning quality evaluation is based on the genome or taxonomic assignment of each bin. In AMBER, first, a source genome is assigned to each contig, then each bin gets assigned to the genome that contains the largest number of its contigs. CheckM evaluation is similar, but as the ground truth is unavailable, the actual source genome is unknown, and thus the taxon placement is used instead. With genome or taxonomic information as a reference, every contig falls under the four classes: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Assume a bin from genome or taxon X and a contig which the bin contains, if the binner does assign the contig to X, then it is a TP for the contig, otherwise a FN. Assume a bin of X and a contig from another genome Y. If the contig gets assigned to X, then it is an FP, otherwise, a TN. For the samples with gold standard binning available, We used the following measures for the assessment, following AMBER's definition[179, 74]. In the formulas, G_X denotes the number of X in the scope of a genome, and B_X the number of X in the scope of a bin, $X \in \{TP, TN, FP, FN\}$.

Completeness (or recall)

$$completeness = recall = \frac{G_{TP}}{G_{TP} + G_{FN}}$$

Purity (or precision)

$$purity = precision = \frac{\sum_{i=1}^{n} B_{TPi}}{\sum_{i=1}^{n} N_i}$$
$$comtamination = 1 - purity$$

Note that *purity* is calculated for each bin and does not account for the unbinned contigs. AMBER computes *purity* and CheckM computes *contamination*.

Both *completeness* and *purity* are computed for each genome or bin. AMBER can calculate the metrics based on the number of sequences or base counts. Unless specified, the metrics used in the context of this chapter are all based on the latter. As CheckM uses no ground truth, it computes the two metrics from the number of single-copy genes.

For the simulated datasets, AMBER estimates two types of overall statistics, the sample-wide statistics based on the total set of underlying genomes and the average statistics based on the computed bins. For realistic datasets evaluated by CheckM, due to the difficulty of unambiguously defining the set of underlying genomes, and the various taxonomic levels at which the bins are identified, no overall statistics are calculated. Only the number of high-quality bins will be reported.

Number of high-quality (HQ) bins

The number of HQ bins are bins with over 90% *completeness* and over 95% *purity*.

Average completeness

$$avg.\ completeness = \sum_{i}^{n} \frac{N_i}{N} \cdot completeness_i$$

Average purity

$$avg.\ purity = \sum_{i}^{n} \frac{N_i}{N} \cdot purity_i$$

Here n is the number of bins, N is the total number of bases, and N_i denotes the number of bases contained in the i-th bin.

Accuracy

$$accuracy = \frac{\sum_{i=1}^{n} B_{TPi}}{N}$$

B_{TPi} represents the TP in the i-th bin. Note that this metric is based on the computed bins rather than the source genomes.

F1-score

$$F1 = 2 \times \frac{purity \times completeness}{purity + completeness}$$

Adjusted Rand Index (ARI)

The use of ARI in the binning evaluation was popularized by Alneberg *et al.* in their work demonstrating CONCOCT[82]. We denote the binning result as a $K \times S$ matrix, in which K represents the number of bins and S is the number of species. N_{ij} represents the number of contigs clustered to i-th bin and j-th species.

$$ARI = \frac{\sum_{i,j} \binom{N_{ij}}{2} - E_3}{\frac{1}{2}(E_1 + E_2) - E_3}$$

with $E_1 = \sum_i^K \binom{\sum_j^S N_{ij}}{2}$, $E_2 = \sum_j^S \binom{\sum_i^K N_{ij}}{2}$, $E_3 = \frac{E_1 E_2}{\binom{N}{2}}$.

The Rand Index quantifies how close a clustering result is compared to the ground truth. ARI adjusts the value by the expectation of the Rand Index of random clustering.

2.4.4 Performance on *Random* and *Half-random*

The metagenomic assemblies of *Random* and *Half-random* recovered 48 and 45 out of the original 50 genomes, respectively. We used binning results from

CONCOCT, MetaBAT2, and MaxBin2, and tested Mapbin in three ways: with the assembly graphs, with the read-pairing, or with both. The binning refinement tool GraphBin is also tested with the three original binning results.

Table 2.2 shows the number of HQ bins for each run, and the overall bin quality statistics are visualized in Figure 2.8. We compare the performance of Mapbin as well as GraphBin to that of the original binners. Mapbin effectively addressed the low bin purity problem with MaxBin2 output and successfully reduced the contamination rate. MetaBAT2 and CONCOCT both achieved a good balance between completeness and purity. For these two, Mapbin enhanced the completeness, F1-score, and ARI of the resulting bins, although the extent of improvement was not as pronounced. Notably, Mapbin tends to work well with MetaBAT2 results, as it was able to combine bins that are incomplete but pure and boosted the number of HQ bins. Compared to the other binning refiner, GraphBin, Mapbin was able to retain the advantages of the original tools. In most of the runs, GraphBin obtained fewer HQ bins compared to the original tool, and the decrease was especially prominent in the case of CONCOCT. And it worsened the other metrics in all the runs with MetaBAT2 and CONCOCT, although it did significantly improve the results from MaxBin2 more than Mapbin.

Table 2.2: Numbers of high-quality bins obtained with Mapbin, GraphBin, and the original tools from datasets *Random* and *Half-random*. Mapbin-a, -p, and -ap refer to using Mapbin with the assembly graph, the read-pairing, and both features, respectively.

		MetaBAT2	**CONCOCT**	**MaxBin2**
Random	*Original*	*35*	*33*	***22***
	+ GraphBin	-1	-9	-2
	+ Mapbin-a	0.	0.	0.
	+ Mapbin-p	+1	**+1**	0.
	+ Mapbin-ap	**+2**	**+1**	-1
Half-random	*Original*	*19*	***26***	***6***
	+ GraphBin	-3	-10	0.
	+ Mapbin-a	**+3**	-2	0.
	+ Mapbin-p	+1	-1	-1
	+ Mapbin-ap	**+3**	-2	-1

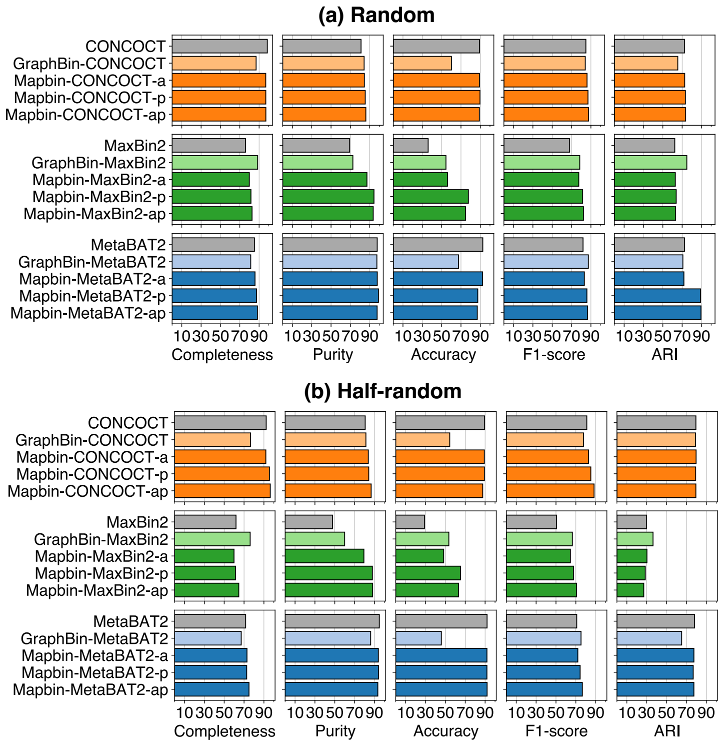


Figure 2.8: Bin quality metrics on the simulated short-read datasets *Random* and *Half-random*.

Given a typical single-sample short-read assembly dataset, Mapbin usually refines the overall bin quality but achieves only modest success in increasing new HQ bins. This is primarily due to the limited presence of supportive edges for merging bins of the same genome. When contigs belonging to the same genome are initially distributed across multiple bins, edges between these bins must exist for Mapbin to merge them as one. However, the networks typically only involve a small fraction of contigs. This is inherent to metagenomic data, where less abundant genomes may exhibit missing regions in short-read assemblies. As a result, the occurrence of bin-bridging edges is not always guaranteed. An example can be found with CONCOCT runs. Eight out of the total 42 genomes in the original CONCOCT output are contained in more than one bin. But only three out of the eight have edges between their member bins, among which two have insufficient numbers (below 5% of the number of contigs in the bin) of edges to support the merging of the bins. Consequently, Mapbin did not drastically increase the number of HQ bins, but mainly corrected the misplacement of a few contigs.

Among Mapbin's three modes, Mapbin-a (assembly only), Mapbin-p (read-pairing only), and Mapbin-ap (assembly and read-pairing), the latter two achieved higher overall bin quality. This can be attributed to the denser nature of the read-pairing networks compared to the assembly graphs. However, Mapbin-ap sometimes may show inferior performance compared to Mapbin-p. This discrepancy is mainly because Mapbin-ap goes through a round of quality control at each network layer, which can sometimes lead to excessive correction of initial bin assignments and filtering of edges.

Investigating the case of MaxBin2 runs, we noticed that MaxBin2 was the only original tool that performed poorly with several large genomes in both datasets. Large genomes have a greater influence on the overall performance statistics due to their larger contribution of bases and contigs. MaxBin2 achieved very limited success with all fungal genomes, and in dataset *Half-random*, it binned only a small portion of closely related genomes from genera *Rhizobium* and *Nostoc* with heavy contamination. Mapbin and GraphBin were then responsible for clustering the contigs from these genomes from scratch. Mapbin generated a few small yet pure bins, while GraphBin produced large bins with higher contamination. This result aligns with Mapbin's algorithm, which avoids creating large new bins from scratch to prevent merging closely related genomes. In contrast, from our understanding, GraphBin is more inclusive of edges that are likely a result of genome relatedness, allowing the formation of larger new bins from unbinned contigs. This approach led to bins with completeness above 50% or even 90%. The generation of large new bins boosted GraphBin's performance scores more significantly than the production of trivial small bins did for Mapbin. However, it's worth noting that GraphBin's large new bins typically have contamination levels above 10% or even 50%, as the algorithm tended to confuse unrelated contigs which have certain sequence similarities.

2.4.5 Performance on the CAMI human microbiome datasets

The gold standard assembly sequences used in our analysis cover 703, 242, 561, 710, and 233 genomes in Air, GI, Skin, Oral and UG samples, respectively. As we intended to use the gold standard assemblies, we skipped the *de novo* assembly step, and as a result, no assembly graphs are available. We used Mapbin only in the read-pairing mode, and replaced GraphBin with VAMB as the former also requires an assembly graph. Note that VAMB was not used in the single-sample datasets because it is optimized for multi-sample datasets with large numbers of contigs (e.g., above 20,000), as pointed out by the developers [99]. Our single-sampled datasets all have much fewer contigs which are expected to cause an overfitting problem for VAMB.

The performances of Mapbin and original binners are presented in Figure 2.9. Mapbin added a remarkable number of new HQ bins to all original binning results. It also improved the purity and accuracy of all four original binners in all samples. As for sample-wide completeness, Mapbin achieved the most notable improvement with MaxBin2 bins, and it also significantly enhanced that of VAMB bins in dataset *GI* and *UG*. The completeness of MetaBAT2 and CONCOCT bins was not further improved. However, the F1-score and ARI indicated that Mapbin successfully achieved a more favorable balance between completeness and purity compared to the original tools, delivering binning results that are closer to the ground truth.

The enhancement of initial bins by Mapbin was found to be more pronounced on the CAMI datasets compared to the *Random* and *Half-random* datasets. Such difference is likely because the CAMI datasets have more contigs that are close to full-length genomes. As explained in Section 2.2.5, Mapbin analyzes the contig length distribution within each middle-to-large sized bin and removes contigs whose lengths are extreme outliers. This led to the separation of a number of near-complete contigs from their initial bins, resulting in more single-contig bins with high completeness and purity. In contrast, short read assemblies like the *Random* and *Half-random* datasets commonly contain very few near-complete contigs, and deviations in contig length distribution were non-existent for almost all the initial bins.

The CAMI datasets are much more complex than *Random* and *Half-random*, but their lower read coverage led to sparser networks. As a result, Mapbin was unable to bin a significant number of initially unbinned contigs. However, it was effective in identifying bin contamination using the networks and correcting the bin assignment of already binned contigs, which made key contributions to the enhanced bin quality.

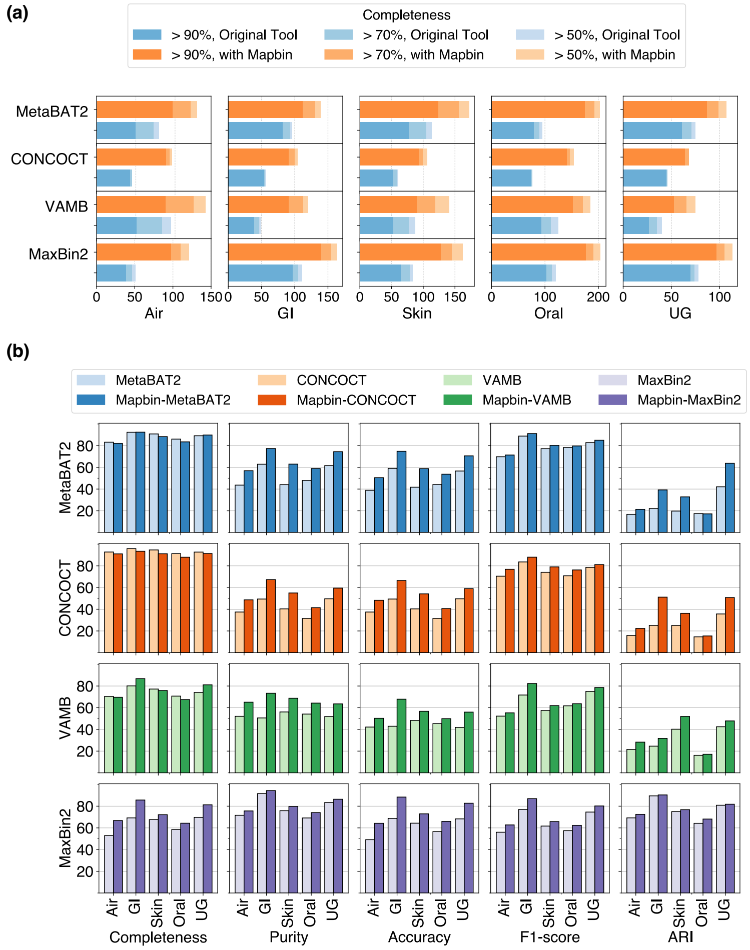


Figure 2.9: Performance of Mapbin and other binners on the five CAMI II toy human microbiome datasets.

It is worth noting that Mapbin employs a multi-step refinement process, wherein the effectiveness of each step varies depending on the specific limitations of the original binners. Of the four original binners we used, MetaBAT2 and CONCOCT clustered a majority of contigs and rarely left out long contigs. However, the benchmarking results showed that their large clusters seemingly have a strong affinity for attracting ultra-long contigs, causing near-complete contigs to be mixed into these clusters. Mapbin was able to resolve the contamination of their large clusters by analyzing its networks and creating bins of single, long contigs. VAMB, on the other hand, left a small fraction of long contigs unbinned. In datasets *GI* and *UG*, which had denser networks than the rest, Mapbin successfully clustered a number of contigs initially unbinned by VAMB. Together with its ability to decontaminate bins, Mapbin eventually raised the completeness of these two datasets for VAMB. Similar to its performance on datasets *Random* and *Half-random*, MaxBin2 left the highest number of contigs unbinned, many of which are in fact ultra-long. This aspect in fact significantly contributed to its higher purity compared to MetaBAT2, VAMB, and CONCOCT. In the case of MaxBin2, Mapbin was able to integrate a notable fraction of its unbinned contigs into the network, leading to a substantial enhancement in completeness.

2.4.6 Performance on the real-world datasets

As explained in Section 2.4.3, for the four experimentally generated datasets, we will mainly report the number of HQ bins. A summary is presented in Table 2.3. Since bins of low contamination and relatively high completeness ($> 50\%$) are generally considered desirable, we also provide a detailed report on the number of bins with less than 5% contamination and 90%, 70%, and 50% completeness in Supplementary Table A.5. Overall, Mapbin's results on these datasets are in line with its performance on *Random* and *Half-random* datasets, which is unsurprising as all are short-read assemblies. The results indicated that Mapbin was more effective in producing new HQ bins for MetaBAT2 output than other tools. For the *Mock* dataset, some genomes were split into multiple bins by MetaBAT2, and they were successfully merged by Mapbin, resulting in four new bins with enhanced completeness (see Table 2.4). Given the fact that this dataset has only 15 source genomes, the improvement was substantial. Mapbin had limited success with *Sample2* and *Sample3*, but managed to increase the number of MetaBAT2's HQ bins in *Sample4* by 28%. In contrast, Mapbin did not generate any additional high-quality (HQ) bins when used with CONCOCT on all samples. Similarly, when applied to MaxBin2's results, only small increases in the number of medium- to high-quality bins can be observed.

In the Mapbin algorithm, the input binning results do not affect the assembly and read-pairing network topology by design, but different binning solutions result in different landscapes of intra- and inter-bin edges. With binning results of acceptable quality, most edges are between contigs from the same bin. But the refinement process is mostly driven by inter-bin edges.

Table 2.3: Numbers of high-quality bins obtained from four real-world datasets using different binners.

	Mock	Sample2	Sample3	Sample4
MetaBAT2	6	38	32	32
+ GraphBin	-3	-28	-24	-18
+ Mapbin-a	+1	0.	0.	+4
+ Mapbin-p	**+2**	**+1**	**+1**	**+8**
+ Mapbin-ap	**+2**	0.	**+1**	**+8**
CONCOCT	**7**	**39**	**39**	**40**
+ GraphBin	-2	-28	-21	-26
+ Mapbin-a	0.	-1	-1	0.
+ Mapbin-p	0.	-1	-2	-1
+ Mapbin-ap	0.	-2	-2	0.
MaxBin2	**4**	17	13	15
+ GraphBin	-2	-9	-7	-10
+ Mapbin-a	0.	**+1**	**+2**	**+1**
+ Mapbin-p	0.	-1	+1	0.
+ Mapbin-ap	0.	0.	+1	0.

Examining the internal data generated from the network filtering step, we observed that CONCOCT results have the least proportion of between-bin edges in the assembly and read-pairing layers. This indicated that CONCOCT clustering solutions were often the most agreeable with the sequence-connectivity data. This may explain the lack of major bin merging or structural changes in the new clustering solution. We shall point out that this may not necessarily serve as proof of CONCOCT bins' perfect quality, but rather indicates that the two additional networks did not provide much new information to challenge CONCOCT's initial result.

All the benchmarking tools displayed consistent performance across different short-read assembly datasets. CONCOCT tended to bin the largest proportion of input contigs. MetaBAT2, on the other hand, occasionally fragmented one genome in multiple yet pure bins. MaxBin2 demonstrated more limitations in handling closely-related genomes, producing less number of bins with higher contamination on average. Consequently, Mapbin showed a higher proficiency in merging broken bins from MetaBAT2, and mitigating bin contamination for MaxBin2 results.

We again observed a critical loss of HQ bins in the output of GraphBin. We use the small dataset *Mock* as an example to break down the performance

Table 2.4: Bins Mapbin-ap merged from original MetaBAT2 bins in dataset *Mock*. Comp.: Completeness, Cont.: Contamination. Both metrics are evaluated by CheckM.

Mapbin-ap taxon	Comp. (%)	Cont. (%)	Comp. (%)	Cont. (%)	MetaBAT2 taxon
o__Burkholderiales	90.81	0.47	89.53	0.47	o__Burkholderiales
			0	0	root
g__Prevotella	98.31	1.01	33.95	0.51	g__Prevotella
			56.9	0	k__Bacteria
f__Bifidobacteriaceae	77.59	0.88	42.4	0.46	f__Bifidobacteriaceae
			22.41	0	k__Bacteria
f__Bifidobacteriaceae	62.25	0	68.44	1.44	o__Actinomycetales
			0	0	root

difference between Mapbin and GraphBin (see Figure 2.10). The diagrams summarize the whereabouts of contigs in different binning results, and the heights of bins are proportional to their total base counts. Despite the noticeable differences between their output, Mapbin and GraphBin did make a few similar modifications to the original binning result. Both made fewer contig shuffling on CONCOCT results compared to that from MetaBAT2. And the two appeared to reach a certain agreement regarding bin merging. For instance, MetaBAT2 Bin 1 and 5 were merged by both methods. However, it is plain to see that GraphBin removed the bin assignments of a considerable fraction of contigs (which are colored gray in the original bins). This was most likely due to the entanglement of shared regions in closely related source genomes, given that a majority of genomes in this dataset are from the same genus. GraphBin labels contigs in the assembly graph with bin IDs, and this issue possibly led to conflicts in contig labels that were unresolved by the algorithm. Moreover, GraphBin falsely exchanged a few contigs between MetaBAT2 bins while, as a matter of fact, no bins produced by MetaBAT2 suffered from a high level of contamination. This issue suggests that it is essential to remove dubious edges arising from sequence similarity. Indeed, when designing Mapbin, we checked the authenticity of all edges in the raw network using the ground truth in simulated datasets, and our edge filtering rules were based on the findings. The benchmarking performance further verified the adequacy of these rules.

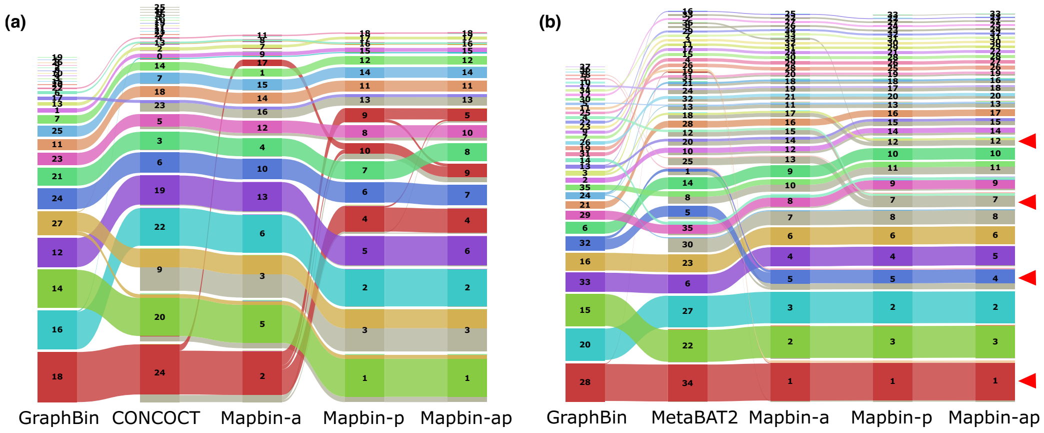


Figure 2.10: Alluvial diagram of binning results on the *Mock* Dataset. (a) Bins generated by CONCOCT, and by Mapbin and GraphBin based on CONCOCT bins. (b) Bins generated by MetaBAT2, and by Mapbin and GraphBin based on MetaBAT2 bins. In (b), Mapbin bins that are merged from multiple original bins are labeled with red triangles at the right end.

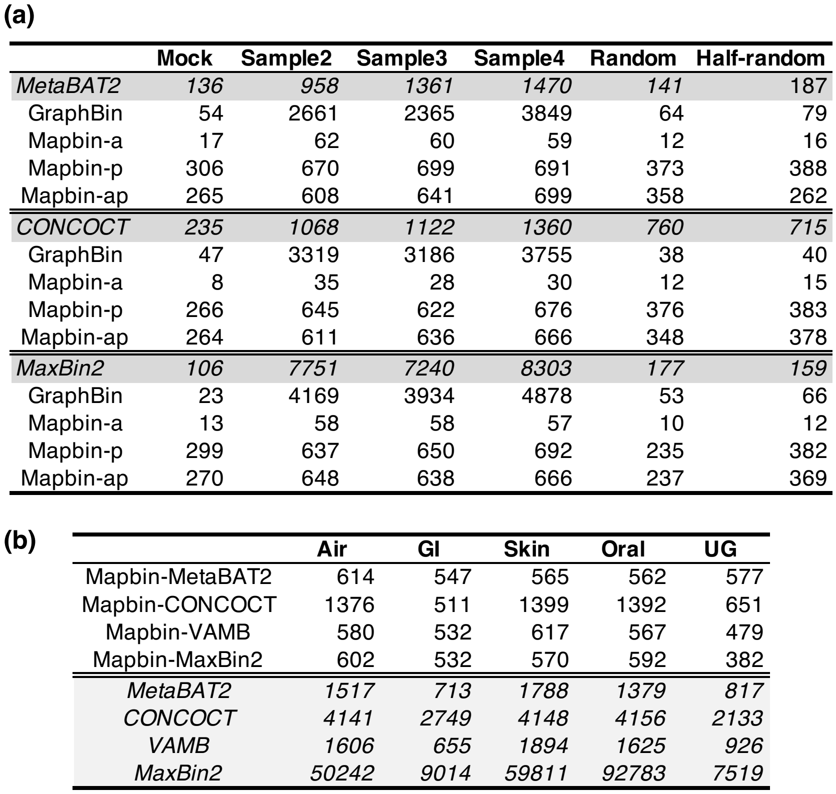
2.4.7 Runtime and memory usage

A summary of the running time and the peak RAM usage of all benchmarking runs is shown in Table 2.5 and 2.6, respectively. We tested Mapbin using eight threads on the CAMI datasets and a single thread on the other datasets. It is worth mentioning that Infomap may use additional threads it detected, but this is an internal process and the parameter is not adjustable from its Python API wrapper.

Overall, Mapbin's runtime and RAM usage scale linearly with the number of input contigs and sizes of the original. The two factors directly decide the size and density of the multilayer network which, in turn, determine Infomap's time and space usage. For the read-pairing mode, parsing reads-to-contigs alignment is the most computationally demanding step, and the size of the alignment file is a key factor for resource usage.

Among all the benchmarking datasets we used, *Random*, *Half-random* and *Mock* are the smallest and the CAMI datasets the largest. The other three experimentally generated datasets, *Sample2* to *Sample4*, are of medium complexity, but their reads-to-contigs alignment files are the largest in size. As expected, the CAMI runs were computationally the most intensive. Mapbin finished within ten minutes with all binning results from MetaBAT2, VAMB, and MaxBin2, as well as the CONCOCT bins for the *GI* and *UG* datasets, but the elapsed times for the other CONCOCT runs were between 22 to 23 minutes. This may be explained by the fact that CONCOCT binned nearly all the contigs and its binning layer networks were always the most complex among the original binning results, requiring more resources for the network clustering step. The peak RAM usage in these runs showed a roughly linear correlation with the maximum total size of alignment files being processed at the same time. Among the CAMI datasets, *UG*'s alignments were the smallest, and the peak memory usage for *UG* runs ranged from 29 to 37 GB. *Air* runs required the most RAM, ranging from 71 to 78 GB.

Table 2.5: Elapsed times (in seconds) of benchmarking runs.



(a)

	Mock	Sample2	Sample3	Sample4	Random	Half-random
MetaBAT2	*136*	*958*	*1361*	*1470*	*141*	187
GraphBin	54	2661	2365	3849	64	79
Mapbin-a	17	62	60	59	12	16
Mapbin-p	306	670	699	691	373	388
Mapbin-ap	265	608	641	699	358	262
CONCOCT	*235*	*1068*	*1122*	*1360*	*760*	*715*
GraphBin	47	3319	3186	3755	38	40
Mapbin-a	8	35	28	30	12	15
Mapbin-p	266	645	622	676	376	383
Mapbin-ap	264	611	636	666	348	378
MaxBin2	*106*	*7751*	*7240*	*8303*	*177*	*159*
GraphBin	23	4169	3934	4878	53	66
Mapbin-a	13	58	58	57	10	12
Mapbin-p	299	637	650	692	235	382
Mapbin-ap	270	648	638	666	237	369

(b)

	Air	GI	Skin	Oral	UG
Mapbin-MetaBAT2	614	547	565	562	577
Mapbin-CONCOCT	1376	511	1399	1392	651
Mapbin-VAMB	580	532	617	567	479
Mapbin-MaxBin2	602	532	570	592	382
MetaBAT2	*1517*	*713*	*1788*	*1379*	*817*
CONCOCT	*4141*	*2749*	*4148*	*4156*	*2133*
VAMB	*1606*	*655*	*1894*	*1625*	*926*
MaxBin2	*50242*	*9014*	*59811*	*92783*	*7519*

All the runs on *Random*, *Half-random* and *Mock* were finished within 6.5 minutes using less than 1.5 GB RAM. *Mock* runs took similar time but used less than 0.5 GB RAM. The assembly-only mode of Mapbin, Mapbin-a, was the most time and space efficient among the three modes. Compared

Table 2.6: Peak RAM usage (in MB) of benchmarking runs

(a)

	Mock	Sample2	Sample3	Sample4	Random	Half-random
MetaBAT2						
GraphBin	149	2205	2110	3005	158	215
Mapbin-a	177	908	891	950	518	493
Mapbin-p	538	1847	1822	2172	1760	1446
Mapbin-ap	539	1847	1822	2168	1760	1430
CONCOCT						
GraphBin	150	2950	2912	3231	141	173
Mapbin-a	187	914	896	958	515	513
Mapbin-p	536	1860	1838	2182	1760	1441
Mapbin-ap	534	1864	1838	2183	1760	1438
MaxBin2						
GraphBin	141	2419	2159	2428	157	203
Mapbin-a	205	912	894	953	688	657
Mapbin-p	536	1860	1831	2184	1760	1437
Mapbin-ap	540	1861	1831	2183	1759	1447

(b)

	Air	GI	Skin	Oral	UG
Mapbin-MetaBAT2	73713	45191	72757	68263	37996
Mapbin-CONCOCT	78623	46865	74384	70888	29937
Mapbin-VAMB	77470	43335	71800	68587	35987
Mapbin-MaxBin2	76863	43098	69849	68603	38373

to the other assembly graph-assisted bin refiner, GraphBin, Mapbin-a ran significantly faster. However, for these three datasets, most of the Mapbin-a runs required more memory. The peak space usage for Mapbin-a appeared at the internally parallelized Infomap clustering step.

For *Sample2* to *Sample4*, Mapbin-a finished within one minute for all original binning results, requiring 920 MB RAM on average. Mapbin-p and Mapbin-ap runs took around 10 to 12 minutes, using around 2 GB RAM. Mapbin is able to process multiple alignment files in parallel, but the computation on a single alignment file is limited to a single thread. Consequently, some Mapbin runs on these datasets took even longer time than the more complex CAMI datasets. Compared to GraphBin, all three Mapbin modes showed markedly higher time and space efficiency, indicating that Mapbin scales better with the increased data volume.

2.5 Discussion

Both assembly graphs and read-pairing networks extract sequence-connectivity information from the data that is conventionally underused. Our benchmarking results demonstrate the power of the multilayer network data structure in proposing the merging of bins, detecting bin contamination, and adjusting the placement of contigs accordingly. The core concept of Mapbin is to minimize the disagreements between the original contig clustering solution and the contig linkage derived from assembly graphs and read alignments. By transforming all input data into a uniform network structure, the disagreement can be efficiently pinpointed. Preprocessing the network partially solves some disagreements and is potentially solved by merging bins, moving contigs between clusters, or removing dubious edges. And by using the fast Infomap algorithm, a new clustering solution can be produced quickly.

Assembly graphs and reads-to-contigs alignments can indicate both genuine connections between contigs from the same genomes in close proximity and sequence similarities between different genomes. We designed Mapbin to be aware of this caveat. We implemented a few precautions against relying solely on the assembly and read-pairing networks for the clustering, as it can lead to merged bins of contigs from closely related genomes. As a result, non-trivial bins produced by Mapbin are expected to be a modification of original bins rather than clustered from scratch based on the assembly and read-pairing networks alone. In this way, Mapbin is able to avoid major drops in bin quality caused by sequence similarity-induced edges. The necessity and effectiveness of our measures were indicated in the benchmarking performance. Unlike GraphBin, Mapbin did not display a tendency to lose a critical number of HQ bins or erroneously remove the original bin assignment of many contigs. However, it should be acknowledged that while our algorithm is indeed effective in selecting trustworthy edges between contigs, it may have a minor side effect of preventing unbinned contigs from forming large new clusters. Therefore, it may not be the optimal choice when working with original binners that leave out a notable number of long contigs.

As explained previously, an inherent limitation of our method is that the assembly graphs and read-pairing network layers only involve a small fraction of contigs by nature, but Mapbin's algorithm cannot work on contigs represented by isolated nodes. This issue becomes more pronounced in cases of more fragmented assemblies and lower read coverage. We observed that bins with noticeable contamination can be easily detected even at a lower network density, as seen in the CAMI cases. But merging bins and clustering initially unbinned contigs were more dependent on the network density. For datasets consisting of contigs with low sequence contiguity and insufficient coverage, Mapbin may primarily improve the bin purity but have a limited impact on completeness.

Although we did not test Mapbin on conventional long-read assemblies, the comparison between Mapbin's performance on the CAMI datasets and the short-read datasets suggests that Mapbin may have a more substantial effect on long-read assemblies as they are statistically similar to the CAMI datasets. Mapbin's read-pairing mode is compatible with both short- and long-read assemblies as long as paired-end sequencing data of the same DNA sample is available. But the current version of Mapbin supports only short-read assembly graphs. We could further broaden Mapbin's scope of applications by adding modules for handling the common formats of long-read assembly graphs.

While Mapbin was demonstrated to scale well with the input data volume, we suggest the alignment parsing step may be further optimized. In the current version of Mapbin, each record in the alignment is inspected to extract the read pairs aligned to different contigs. Parallelization of this step may significantly improve the runtime of Mapbin-p and Mapbin-ap.

2.6 Conclusion

We designed Mapbin, an algorithm that refines genomic binning results by using assembly graphs and read-pairing information from paired-end sequencing data. Mapbin constructs a network individually from the original binning result, the assembly graph, and the reads-to-contigs alignments. These networks are then integrated into a multilayer network with nodes representing contigs. By applying the community-detection algorithm, Infomap, Mapbin clusters the network to produce a refined binning result. We demonstrated Mapbin's performance on multiple simulated and real-world datasets of varying complexity. In summary, Mapbin is a versatile tool suitable for both short and long-read assemblies. It is proven effective in detecting and mitigating contamination in the given binning results, correcting contig misclustering, and enhancing overall bin quality. In our benchmarking, Mapbin's performance in boosting the number of HQ bins was generally limited on short-read assemblies but quite pronounced with assemblies of higher sequence contiguity, such as gold standard assemblies of the CAMI datasets. Mapbin scales efficiently with the growing complexity of input data volume. The resource usage is dependent mainly on the number of contigs, the number of bins, and the reads-to-contigs alignment file sizes. These results demonstrated the value of sequence connectivity information in improving metagenomic binning quality and the power of multilayer networks in seamlessly integrating them into the binning process.

Chapter 3

Metagenomic analysis of human gut microbiome during stem cell transplantation

3.1 Introduction and dataset description

This chapter reviews our microbiome analysis for the project ImMiGeNe, a longitudinal multi-omics study investigating the associations between human gut microbiota and the host immune systems in stem cell transplantation (SCT) patients. SCT is a strong intervention that drastically reshapes the immune system and gut microbiota of the patients. It is of great interest to understand how the microbiome changes over the period of treatment in response to a variety of factors.

To this end, feces samples from the subjects were collected over the entire treatment period and subjected to shotgun metagenomic sequencing. We analyzed the data to characterize and identify patterns of their gut microbiome at different therapeutic stages, in the hope to delineate their developmental trajectory. It should be noted that this chapter focuses on documenting and demonstrating data-related computational work, and a detailed description of the research design and setup, such as sample collection methods, medical treatment and biomedical characteristics of the patients, laboratory protocols, etc. will be excluded here.

Twenty stem cell donor-recipient pairs were recruited in ImMiGeNe and the sample collection spanned the entire time window of the treatment. A

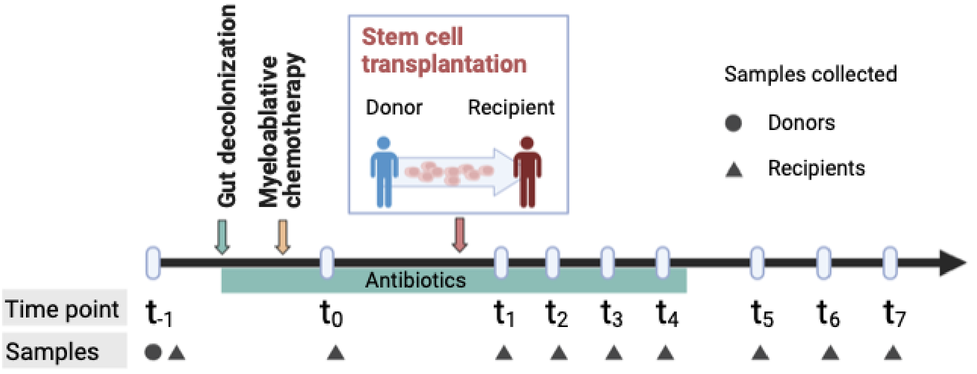


Figure adapted from Prof. Dr. Alexander N.R. Weber, Interfaculty Institute for Cell Biology, University of Tübingen, 07.04.2020

Figure 3.1: Stool sample collection schedule in the ImMiGeNe project.

summary of the sample collection plan is shown in Figure 3.1. The patients first underwent enteric decolonization with antibiotics, a standard preoperative prophylaxis to help reduce the risks of potential infection. The procedure uses broad-spectrum antibiotics and usually results in the radical removal of existing gut microbes. Next, they received high-dose chemotherapy which was to destroy cells in the bone marrow to prepare a clean slate for receiving the donor's stem cells. At this time point (myeloablative chemotherapy in Figure 3.1), a severely compromised immune system was expected. The next procedure was stem cell transplantation. Antibiotics treatment continued for some time after the SCT. Our sample collection started at one time point shortly before the administration of antibiotics, at which both the donor and their recipients' stool samples were collected (t_{-1}). This was the only time point with donors' samples. For the recipients, there was a round of stool sample collection between their chemotherapy and SCT (t_0). Samples collected after the SCT started from t_1, with four time points during the antibiotic treatment and three more afterwards.

Due to their medical condition, many patients in this study were unable to offer samples consistently. Almost for each patient and at each time point, some missing samples could be expected. After the removal of missing or low-quality samples, all of the donors', and 17 out of the 20 recipients' data were used for the microbiome analyses.

3.2 Raw data processing

Initially, we obtained a dataset of 146 paired-end shotgun sequencing samples, 8 of which are blank control samples. The goal of raw data processing was to remove reads that are of low quality, duplicates, or of human origin. It is not only crucial as a standard procedure for quality control, but also key to keeping patient-sensitive data out of the analysis.

We used Trimmomatic[180] to remove low-quality reads. For host read removal, we mapped reads against human genome reference GRCh38 (hg38, RefSeq ID GCF_000001405.39) using Bowtie 2[181]. Reads that were unmapped were kept. At this step, we found human reads prevalent in the recipient samples (see Figure 3.2. In many samples, human reads constituted more than half of the total. This was mainly due to the low target DNA concentration in these samples. We noticed that this problem is also frequently combined with other issues such as low complexity reads (e.g. AAAAAAAAAAAAAAAAAAAAAAAAT). Due to their high base quality, these reads were not removed by Trimmomatic, and due to their low complexity, they were unaligned to the human genome. We performed one round of data preprocessing and found that, out of all 138 samples, 36 had less than 1 million reads, and 62 had less than 10 million.

Compared to large-scale gut microbiome analysis such as The Integrative Human Microbiome Project[182], the read yields of these samples were moderate to low. To retain more microbial reads, we re-processed the data, and treated the paired-end reads as single-end for the alignment against the human genome, and re-paired them afterwards. This yielded a higher proportion of unpaired reads, but left more reads for downstream analysis. In some samples, duplicates were also prominent. This is a known problem with samples that are too fragmented or have low target DNA concentration[183].

The overall statistics of the preprocessing are shown in Figure 3.2. Samples from the healthy donors turned out to be of better quality, with few human genome contamination and duplicates, and also showed much less individual differences. For the SCT patients, the samples collected before the treatment (first bar in each cell in the figure) were of good quality on average, but those collected since the antibiotic treatment varied a lot. Many of them were low in read count or heavily contaminated by human reads.

As none of the blank samples yielded a sufficient number of reads (more than 1000), nor contained any taxa detectable by our taxonomic analysis, we subsequently excluded them in the findings section.

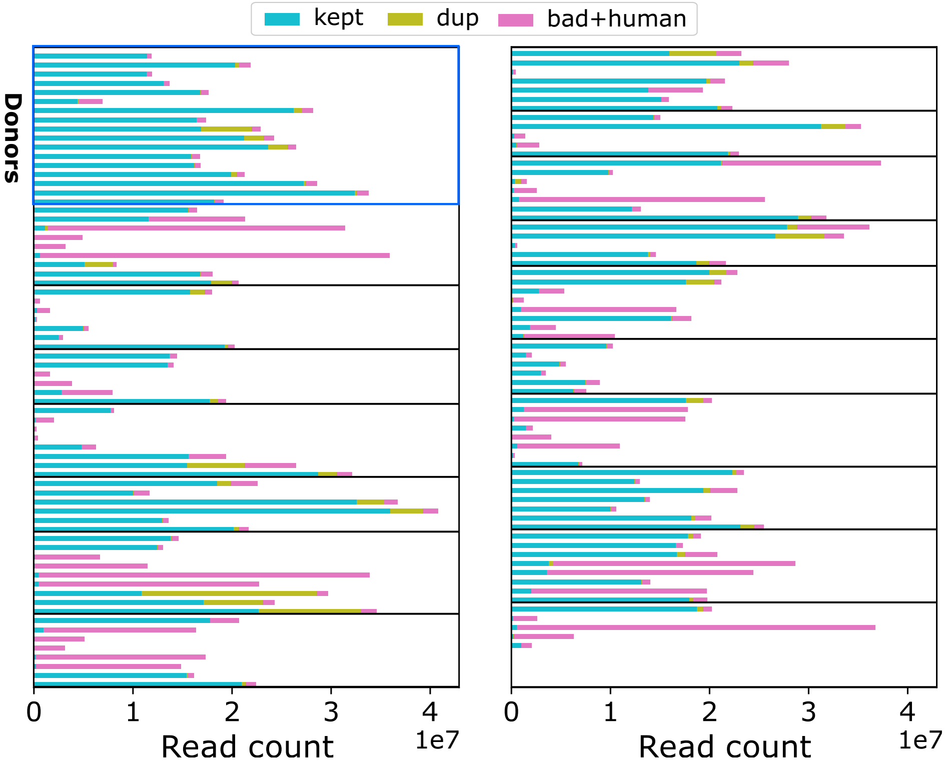


Figure 3.2: Statistical summary of ImMiGeNe raw metagenomic data processing. Each stacked bar is a metagenomic sample. The horizontal lines separate the samples by individuals. Each individual's samples are sorted by the sampling time points. *Kept*: reads that can be used for downstream analysis; *dup*: discarded duplicates; *bad*: discarded low quality reads; *human*: discarded human reads. The samples from donors are on the top left, highlighted with a blue frame. All the rest are from the recipients.

3.3 Taxonomic and functional profiling

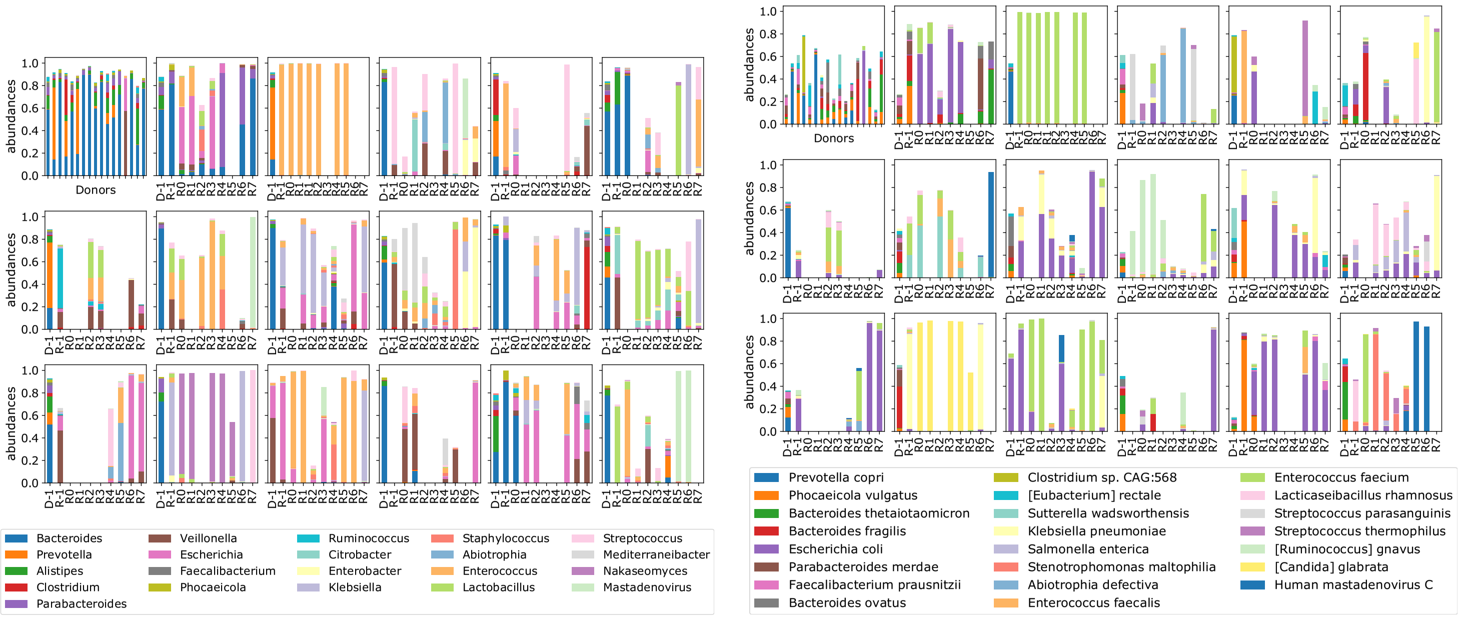
We characterized the gut microbiome of the donors and recipients with DIAMOND+MEGAN pipeline[184]. We first aligned the filtered reads against NCBI-nr database (retrieved via NCBI ftp in December 2020) using DIAMOND 2.0.11[185]. Then we performed taxonomic and functional profiling using MEGAN 6 Ultimate Edition[186].

3.3.1 Compositional instability of gut microbiota among the recipients

The taxonomic profiling was based on both NCBI Taxonomy and the Genome Taxonomy Database (GTDB[59]). The result showed a high level of similarity in the gut microbiome of the healthy donors, and great instability in that of the recipients. We examined the composition both intra- and inter-individually. For each recipient (see Figure 3.3), we compared their gut microbiome over time, and to that of their donor. The results revealed significant deviation from that of the healthy donors and erratic fluctuations over time. All but one donor's gut microbiome were alike. However, for the majority of the recipient cohort, there was little similarity, and abrupt changes in the taxonomic composition. A lot of samples appeared to be dominated by very few taxa, which appeared to be present by chance. And the succession of dominant taxa occurred frequently and dramatically. They also had little consensus when compared between individuals within the same time group (see Figure 3.4).

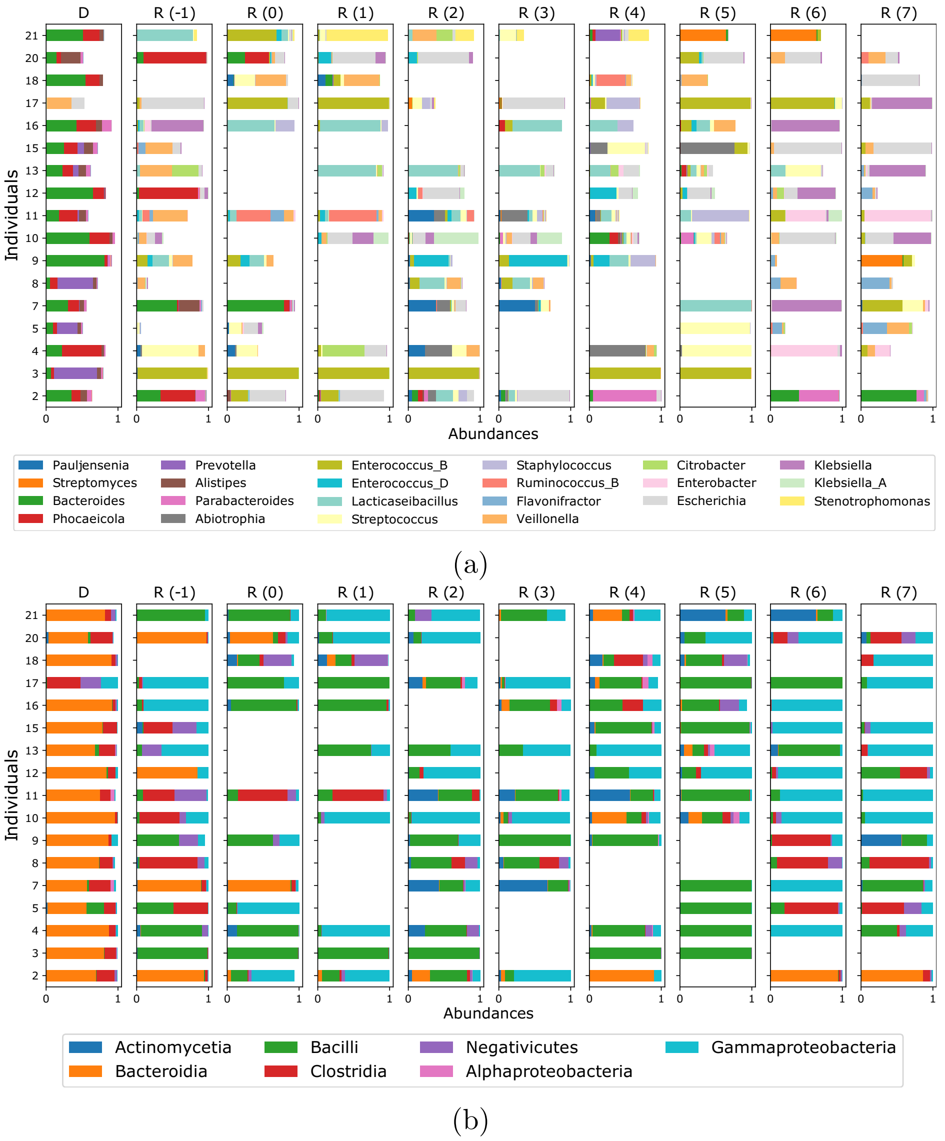
Additionally, the composition at the species level (see Figure 3.3 (B)) revealed that most of the dominant taxa in the recipients' samples were either multidrug-resistant species, or pathogens known to be associated with inflammation in the gastrointestinal tract. This includes and is not limited to *Escherichia coli*, *Klebsiella pneumoniae*, *Enterobacter spp*, *Enterococcus faecium*, *Salmonella enterica*, *Ruminococcus gnavus*, *Streptococcus parasanguinis*, etc[187, 188, 189]. It implies that the precarious physical conditions of the host and the heavy use of antibiotics had disrupted the colonization of beneficial commensal gut microbes.

One important signature of gut microbiota dysbiosis is the decreased abundance of strict anaerobes. The majority of their samples are dominated by the classes *Gammaproteobacteria* and *Bacilli*, while that of the donors *Bacteroidia*, followed by *Clostridia*. The major metabolic difference between the two groups is that the former is facultative anaerobic, while the latter is obligate anaerobic. Previous studies have indicated that such difference in the microbial nutrient metabolic landscape is closely correlated with changes in the available nutrients in the colon during inflammation[109, 46, 143]. This inspired us to investigate the related metabolic pathways in the functional analysis.



(a) (b)

Figure 3.3: Significant instability in the gut microbiota composition among the recipients. The taxonomy is at the NCBI genus level in (a) and NCBI species level in (b). The top 10 abundant taxa from donor's samples, and otherwise taxa that make up over 1% of all communities, are included. The top left cell in the blue frame shows all donors, and each of the others shows a donor-recipient pair sorted by time. D represents the donor, and R the recipient. The i in Ri is the time point.



(a)

(b)

Figure 3.4: Gut microbiome composition over time. (a) disparate post-SCT microbiota development of recipients. The taxa are at the GTDB genus level; (b) Enrichment of *Gammaproteobacteria* and *Bacilli* in the recipients. The taxa are at the GTDB class level. The leftmost columns show all donors, and each row is one donor-recipient pair, with their IDs on the y-axis. Each of the other columns are samples from the same time point.

Fungi and viruses can only be detected using the NCBI taxonomy. While

GTDB does provide a taxonomic hierarchy that is phylogenetically more consistent than the NCBI taxonomy, it is a database only for bacteria and archaea. One non-trivial caveat is that some viruses are mislabelled as their host bacteria and end up in the GTDB database too. Their presence may introduce computational inflation of the abundance of the host. We recommend using both NCBI and GTDB taxonomies in the context of inflammation- and disease-related gut microbiome study for a more comprehensive analysis.

Using the NCBI taxonomy, we found a considerable amount of Mastadenoviruses, from the family *Adenoviridae*, as well as fungi like *Candida* spp.. Mastadenoviruses have been frequently reported to be associated with gastrointestinal infection in children[190, 191]. *Candida* spp., on the other hand, is a prevalent component of human microbiome[131].

3.3.2 Marked contrast of microbial diversity between donors and recipients

We further computed the alpha and beta diversity of the samples. The alpha diversity measures the diversity within one sample, while the beta diversity measures the differences between different samples. We used the Shannon-Weaver index (or Shannon index) as the measurement of alpha diversity, which is computed as

$$H' = -\sum_{i=1}^{n} p(i) \cdot \log(p(i))$$

where H' is the Shannon index, $p(i)$ is the relative abundance of the i-th species in the microbial community (sample).

The Bray-Curtis dissimilarity is used to measure the beta diversity, the formula is:

$$D = \frac{\sum |A(i) - B(i)|}{\sum (A(i) + B(i))}$$

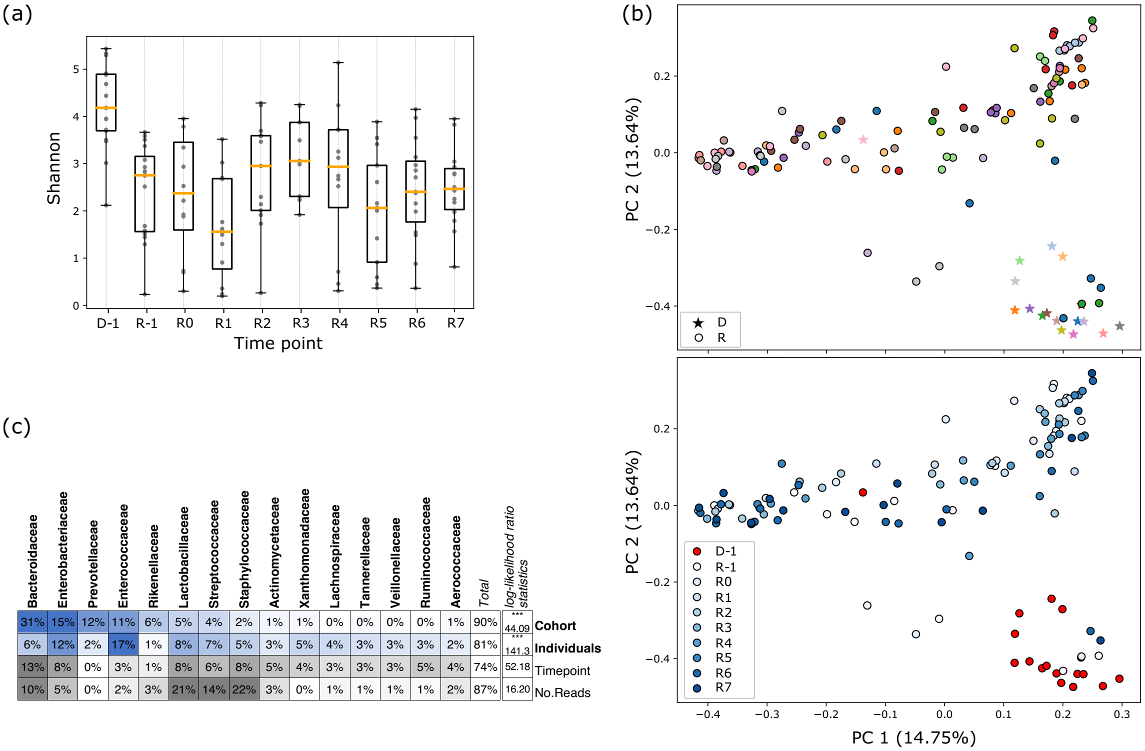
It computes the dissimilarity between a pair of samples A and B. $A(i)$ and $B(i)$ are the relative abundances of the i-th species, respectively. All samples are subsampled (rarified) to the read count of the smallest sample 1000 times.

Additionally, we sought to explain the variations between samples with the following factors: cohort (that is, whether the sample comes from a donor or a recipient), individual, sample collection time, and read counts of the samples. We performed a multivariate analysis using a generalized linear model. The number of reads was included in the analysis because it varied significantly between samples, and the recipient samples typically had fewer reads compared to the donors'. This might have contributed to their low diversity. The analysis was done with `mvabund` package in R, tested with log-likelihood ratio statistics.

The results, shown in Figure 3.5 confirmed the loss of microbial diversity, and the erratic nature and lack of consistency in the development trajectories of the recipients' gut microbiome. Overall, all except for one sample in the donor cohort showed a great level of similarity. In contrast, the recipient cohort displayed great individual differences and deviated greatly from the donors. The read count differences did not significantly correlate with the microbial diversity.

3.3.3 Key genes in respiratory pathways enriched in the recipient cohort

Since the taxonomic profiles indicated that facultative anaerobes overtook their fastidiously anaerobic competitors following antibiotic use, in our functional analysis, we focused on the genes that may give the facultative anaerobes the upper hand. As introduced in Chapter 1.3.4, their metabolic versatility is manifested in their ability to utilize energetically valuable terminal electron acceptors that emerge with the onset of inflammation. These include nitrate, nitrite, DMSO, TMAO, which can be used in anaerobic respiration, and oxygen which enables aerobic respiration. We selected the key enzyme-encoding genes of interest based on a study conducted by Hughes and colleagues[46], and categorized them roughly into two groups, one related to aerobic and the other to anaerobic electron transport chain. Most of the genes encoding the key enzymes in these respiratory pathways are not commonly found in the genomes of obligate anaerobes. Note that we put formate dehydrogenases under aerobic respiratory pathways. This is because, under a microaerobic environment, commensal *Enterobacteriaceae* tends to couple formate oxidation with aerobic respiration, using formate as electron donor and oxygen as the terminal electron acceptor[46].



Bacteroidaceae	Enterobacteriaceae	Prevotellaceae	Enterococcaceae	Rikenellaceae	Lactobacillaceae	Streptococcaceae	Staphylococcaceae	Actinomycetaceae	Xanthomonadaceae	Lachnospiraceae	Tannerellaceae	Veillonellaceae	Ruminococcaceae	Aerococcaceae	*Total*	*log-likelihood ratio statistics*	
31%	15%	12%	11%	6%	5%	4%	2%	1%	1%	0%	0%	0%	0%	1%	90%	*** 44.09	**Cohort**
6%	12%	2%	17%	1%	8%	7%	5%	3%	5%	4%	3%	3%	3%	2%	81%	*** 141.3	**Individuals**
13%	8%	0%	3%	1%	8%	6%	8%	5%	4%	3%	3%	3%	5%	4%	74%	52.18	Timepoint
10%	5%	0%	2%	3%	21%	14%	22%	3%	0%	1%	1%	1%	1%	2%	87%	16.20	No.Reads

Figure 3.5: Statistical analysis demonstrates the largest variations are between different cohorts and individuals. (a) Shannon diversity of the microbial profiles at the species level. (b) Principal coordinates analysis (PCoA) based on Bray-Curtis dissimilarity at the species level. In the top figure, samples are colored by the individuals and each donor-recipient pair has the same coloring. In the bottom one, donors are all in red and recipients in blue. The shades of blue distinguish the sampling time point. (c) Multivariate analysis, at the family level. The numbers are the percentage of variance explained by different factors. For instance, the cell at the top left corner shows 31% of the variance of *Bacteroidaceae* abundances across samples can be explained by the cohort. $*** : P \leq 0.001$.

Metabolic functions of the gut microbiome are generated based on the orthologous groups in two databases: *Kyoto Encyclopedia of Genes and Genomes* (KEGG)[192] and *evolutionary genealogy of genes: Non-supervised Orthologous Groups* (eggNOG)[193]. KEGG hierarchy is based on sequence similarities. EggNOG on the other hand, on top of that, also takes into consideration the phylogenetic relationships of the source organisms of the genes. We compiled a list of KEGG orthologs (KOs) and Clusters of Orthologous Groups (COGs) by searching each enzyme by name. The enzymes, KOs and COGs are listed in Table 3.1 A.6, and A.7, respectively. The functional profiling was performed by MEGAN 6 based on the reads-to-nr alignments. One important difference is that the KOs can be more fine-grained than the COGs in terms of gene functions. Some COGs include multiple genes in the list, such as *dmsB* and *nrfC* in *E. coli* are both under COG0437, but in KEGG they are under K07307 and K04104 respectively.

Table 3.1: Enzymes in the respiratory electron transport chain.

Symbol	**Enzyme**
Anaerobic respiratory enzymes	
Nap	periplasmic nitrate reductase
Nar	respiratory nitrate reductase
DmsABC	anaerobic dimethyl sulfoxide reductase
YnfFGH	Tat-targeted selenate reductase
TOR	trimethylamine-N-oxide reductase
NrfABCD	nitrite reductase
FrdABCD	succinate dehydrogenase
Aerobic respiratory enzymes	
CyoABCD	cytochrome o ubiquinol oxidase
CydAB	cytochrome bd ubiquinol oxidase
FDN	formate dehydrogenase-N
FDO	formate dehydrogenase-O
HydABC	NAD(H)-dependent [FeFe]-hydrogenase

The results, shown in Figure 3.6, once again indicated the dysbiotic nature of the gut microbiome among the recipients. Using the median of the donors' measurements as a reference, we found genes encoding the key enzymes in respiratory pathways were highly abundant. Both the formate dehydrogenase and cytochrome oxidase, which are associated with oxygen respiration, were found to be enriched by 1000 fold in many recipient samples. As for anaerobic respiratory potentials, the level of genes encoding the nitrate, nitrite, TMAO, and DMSO respiration enzymes was also significantly augmented.

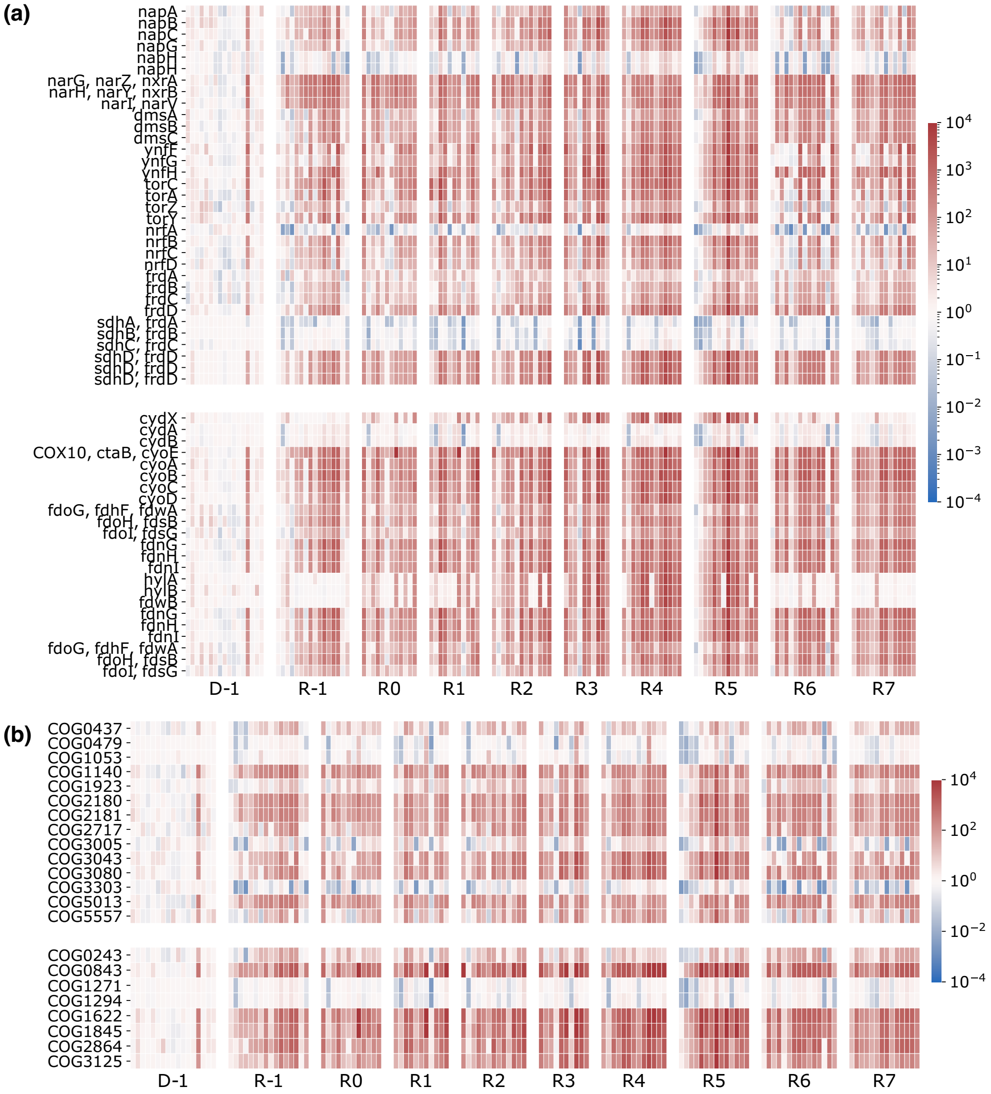


Figure 3.6: Functional analysis of the respiratory pathway-related oxidoreductase genes. The orthologs are from (a) KEGG, and (b)eggNOG. Their abundances are normalized against the total read count, and the values indicate their abundances compared to the median of the values in the donor group. The samples are sorted according to the time point. The top and bottom halves are anaerobic and aerobic respiration-related, respectively.

Linking the taxonomic to the functional analysis provides insights into the functional similarities of taxonomically different groups. As shown in Figure 3.4 (b), taxonomic dominance seemed to oscillate between the classes *Bacilli* and *Gammaproteobacteria* in the recipient cohort (see Individual 4, 5, 7, 11, 12, 13, 15, 16, 17, 18, 20 and 21). But in the functional analysis, elevated abundances of oxygen, nitrate, nitrite, TMAO, and DMSO respiration-related genes were observed in almost all recipient samples. This hints that the functional similarity of the two dominant groups may help them colonize the strongly disrupted new environment.

3.3.4 The donor and recipient gut microbiome encode different flagellins

We also performed an analysis of the flagellin content. As introduced in Chapter 1.3.5, a wide range of gut microbes are flagellated, but they produce structurally different flagella, which elicit distinct immune responses from the host. Since FliC encompasses all bacterial flagellins, and they are under the same ortholog in both KEGG and eggNOG, we used a customized database to capture the flagellin content. The database was provided as a courtesy by Andrea Borbon from Max Planck Institute for Developmental Biology. It was curated by compiling all 33051 bacterial and archaeal flagellins available on NCBI at that time. Dereplicated sequences were predicted for domains with InterProScan (`https://www.ebi.ac.uk/interpro/`) using Pfam (hosted now at `https://www.ebi.ac.uk/interpro/`) and PANTHER`http://www.pantherdb.org/` databases, and those predicted to be bacterial with both C- and N-termini or archaeal flagellin were kept.

The results are shown in Figure 3.7. In the donor cohort, the flagellin are mostly coming from the family *Lachnospiraceae*. Despite having moderate abundance, they are highly prevalent among the samples. The flagellin profiles echo the taxonomic profile of the flagellated microbes. For the recipients, *Enterobacteriaceae* flagellins are the most abundant, followed by those of *Enterococcus* origin.

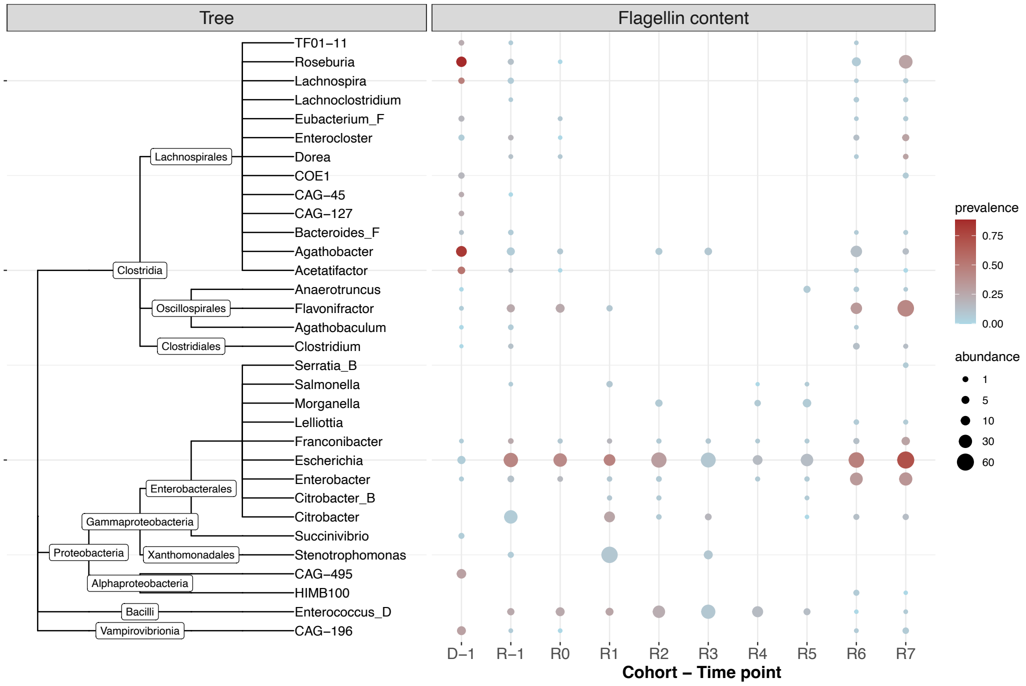


Figure 3.7: Flagellin content of the samples. The tree is based on GTDB taxonomy, reflecting the source genome of the flagellins. Prevalence measures how frequently a taxon occurs across samples, whereas abundance describes the average count of a taxon among samples where it occurs.

3.4 Ecological interpretations based on the metagenomic analysis

The prolonged microbiota dysbiosis may be linked to a plethora of factors, including:

Compromised host immune system which has a profound impact on the gut microbiome developmental trajectory, possibly since as early as the primary succession process at their birth;

Long-term use of medications and therapeutic interventions which constantly cause major environmental stress to the gut microbes, leading to chronic instability of the community;

Exposure to nosocomial pathogens which can prolong the inflammation and interrupt the re-colonization of beneficial commensals.

Each of these factors may have been extensively studied individually, but rarely combined. The gut microbiome under the impact of SCT is a rather intriguing subject to study: the microbial communities may be much less complex than common human gut microbiomes, but the ecological mechanism behind the dysbiosis could be fairly complicated to infer.

In this study, in the patient cohort, SCT treatment combined with antibiotic use destroyed their initial gut microbial community, creating a typical secondary succession scenario[131, 162]. Secondary succession is the process of re-colonization of a community of organisms in an ecosystem after a disturbance event. Conceptually it is in contrast to the primary succession which happens at birth when a few pioneer microbes establish the gut microbiome in the newborn. The disturbance that led to the secondary succession in our context was the radically invasive treatment which removed most of the existing microbes and drove the gut microbiota to start re-colonizing on a clean slate. The secondary succession can be expected to happen in roughly three stages. First, a stochastic process will happen for a certain time, in which the first settlers arrive and grow not as a result of their defined ecological fitness, but due to a chance event. The trajectory is pushed towards a more deterministic second stage when a few "keystone species" arrive. Like pioneer species in primary succession, they are able to establish symbiosis with the host by utilizing host-derived nutrients, and produce metabolites that favor the growth of obligate anaerobes. The return of obligate anaerobes marks the third stage, culminating in the ecological recovery of the microbial community[194, 131].

Our data showcased the relative stability of gut microbiota in healthy adults, and pronounced volatility of that in the SCT patients. The metagenomic analysis implied that the process of re-establishing ecological equilibrium in the recipients is most likely longer than the span of our study. A majority of the recipients started off at t_{-1} with an already imbalanced microbiome, characterized by the dissimilarity between individuals, lack of overall diversity and strict anaerobes' dominance, and overgrowth of known pathogens. After the major disturbance at t_0, and until t_7, what we witnessed the most was the blooming facultative anaerobes. Occasionally, there are signs of emerging fibre degraders, but they barely dominated, and usually got replaced right at the next time of sampling. Chng *et al.* conducted an inter-study analysis regarding human gut microbiome recovery after antibiotic treatment, and suggested a list of keystone species that are consistently positively associated with rapid community rebound. These species turned out to be specializing in metabolizing complex polysaccharides[162, 194]. In our recipient samples, none of the 21 taxa were frequently detected with noticeable abundances. Instead, we found our samples displaying typical features of the "non-recoverers", which are the community whose microbial diversity did not recover even three months after the antibiotic treatment in the other study[162]. At the final sampling time, all patients still seemed to be in the very first phase, where the primary species, i.e., the opportunistic pathogens occupied their gut. The secondary succession and keystone species hypotheses may help us understand the slow recovery of microbiota in these patients, providing valuable perspectives for medical intervention-aided microbiome restoration.

3.5 Limitations and caveats of the data analysis

In this discussion, we address the limitations stemming from the dataset itself as well as the chosen microbiome analysis methods.

3.5.1 Contamination issue with low-biomass samples

The gut dysfunction of the SCT patients leads to feces samples of low biomass, or a limited quantity of endogenous microbial DNA that is our sequencing target. This was a common problem among the patient group, which was also manifested in the low yield of reads[195]. The most prominent issue with low-biomass samples is the contaminants, or off-target DNA material, which can potentially undermine microbiome analysis. Contamination can come from the sample itself, and also the laboratory environment. It happens not simply due to the negligence of the experimenters. Contaminants are commonly present in most microbial samples, but their impact is minimal in regular samples due to the much stronger signal from target sequences. However, in low-biomass samples, even a trivial amount of contaminants could outnumber the targets, and introduce a considerable amount of noise to the data.

Host DNA contamination

We have addressed the host DNA contamination problem previously in the analysis. Due to the lack of microbial content, many samples were full of human DNA. The impact of host DNA contamination on metagenomic analysis has been elucidated previously. In short, it leads to distorted relative abundances of different taxa and reduced capability to capture low-abundance species. This is demonstrated in case studies using synthetic samples in two ways: by gradually increasing[196] or depleting host DNA[197, 198] in the samples. One possible explanation is that the overwhelming presence of host DNA diminishes the chances for microbial genomes to be sequenced, resulting in insufficient coverage of microbial genomes in general[196]. The strong presence of host DNA adds considerable uncertainty to the data interpretation. And computational removal of host reads could not remedy the data distortion it caused. Pereira-Marques *et al.*[196] demonstrated with their study case that, with high levels of host contamination, species with below 1% relative abundance (based on 16S as well as genome copies) already become occasionally undetectable. For metagenomics, this level is typically considered to be still quantitatively relevant.

One key observation about our recipient samples was that many of them seemed to be of extremely low diversity, consisting of several pathogens. However, it is possible that the pathogen content was overrepresented, and other community members were undetected, due to the high ratio of host to microbial genomic content. Or promisingly, the dominance of the pathogens may not be as strong as it appeared, and the keystone fiber-degraders (see Section 3.4) may still be present, but poorly represented by the sample.

Laboratory contamination

Besides the host DNA, some other contaminants might be introduced by the laboratory practice, such as the reagents, DNA extraction procedure, the researcher, and lab facilities[199]. Our study involves a few control samples ("blanks"). However, we warn that they are insufficient proof to conclude that the samples are free from contaminants. The completely empty profile could be a consequence of limited detection sensibility. It is also not clear how the blank samples were made and what types of DNA contamination issues they intended to address.

Weyrich *et al.*[195] illustrated that a regular laboratory environment is expected to introduce a substantial amount of contamination to delicate samples of low-biomass. Moreover, different types of "blanks" vary a lot in their detection capabilities. In the study, they introduced two types of control samples, extraction blank controls (EBCs) and no-template amplification controls (NTCs). EBCs involve the use of empty tubes during the DNA extraction step, and NTCs are amplified without the DNA input. NTCs alone are unable to tell contaminants introduced from DNA extraction, which can constitute half of the total contamination. Another study, conducted by Salter *et al.*, addressed the impact of reagent contamination[200]. Their target DNA dilution series demonstrated the presence of reagent contaminants is comparable to or able to overpower that of the target species. Further, the impact of reaction kit contamination could mislead microbiota studies with low-biomass samples, such as those taken from nasopharyngeal swabs, as they may explain the variations even better than the other ecological factors considered. Interestingly, the common contaminants identified in this study as well as earlier studies are also common members of human gut microbiota[200, 195].

Moreover, contaminant taxa may also have certain common metabolic pathways. One example is nitrogen fixation pathways, which are likely enriched because nitrogen gas is widely used in the industrial production of ultrapure water[201]. This is especially relevant to our case, because the low-biomass problem was limited to the recipient samples. The contrast of metabolic pathways in the two groups might be attributed to the contamination rather than the target microbiota.

There is no basis for computational depletion of environmental contaminants in the lack of proper control samples as reference, as the background "noise" may have a lot in common with the target microbiota in terms of the taxonomic makeup. We stress that the purpose of using control samples is not to deny the presence of contaminants, but rather to accurately detect them. Failure to detect DNA material of low concentration is not a proper means of prevention against contaminants, because it also leads to skewed microbiota profile with poor resolution.

3.5.2 Partial coverage of genomic regions in low-biomass samples

Even for the most abundant species, the recipient samples did not accurately capture all parts of their genomes. It happened frequently that, even for taxa with a large number of assigned reads, a significant proportion of their genomic elements were still missing from the data. This implies that the genomic material in the library was likely quite fragmented, and a high abundance of a taxon was probably generated by sequencing several fragments from its genome over and over again. In other words, the samples took only a small and biased subset of all genomic components in the community. Clearly, this further confounds diversity analysis, leading to the overestimation of dominant taxa.

The likely incomplete genomic material further limited the information we could extract from the data. We have also attempted a few other analyses apart from what was mentioned above. Metagenomic assembly was only successful for the donor group. For the recipients, many of the read libraries, even if their read counts were not too low, were characterized by high coverage over some short windows on the genomes of abundant species, such as *E.coli*. They could not come together to an informative length.

We also noticed that functional annotation worked out poorly when performed at a finer grain. Looking back on the oxidoreductase and flagellin analyses, the former gathers content from orthologs of non-trivial gene families, while the latter looks for genes with a specific source species. Interestingly, for recipient samples, the amount of functional content scales with the read counts of relevant species, only at the ortholog, but not more specific level. Take the flagellin content analysis as an example (see Figure 3.8). *Roseburia* spp. and *Escherichia* spp. were the most prevalent flagellated bacteria in the donor and the recipient samples, respectively. In the donor samples, there was a positive correlation between the taxon-level abundance and the flagellin content detected in the sample, as we expected. In the recipient samples, however, the taxonomic profiling indicated quite a few samples had as many as above 10^6 reads assigned to the genus *Escherichia*, but few to none of them were identified as flagellin-encoding. Some other samples contained a few tens of thousands of reads, but above hundreds of them came from flagellin genes. A similar pattern for *Roseburia* was observed in the samples where it was present. Based on this, we chose not to pursue additional analysis that would suggest high specificity, such as antibiotic-resistance gene analysis, as the comparison between samples would not be made with great confidence.

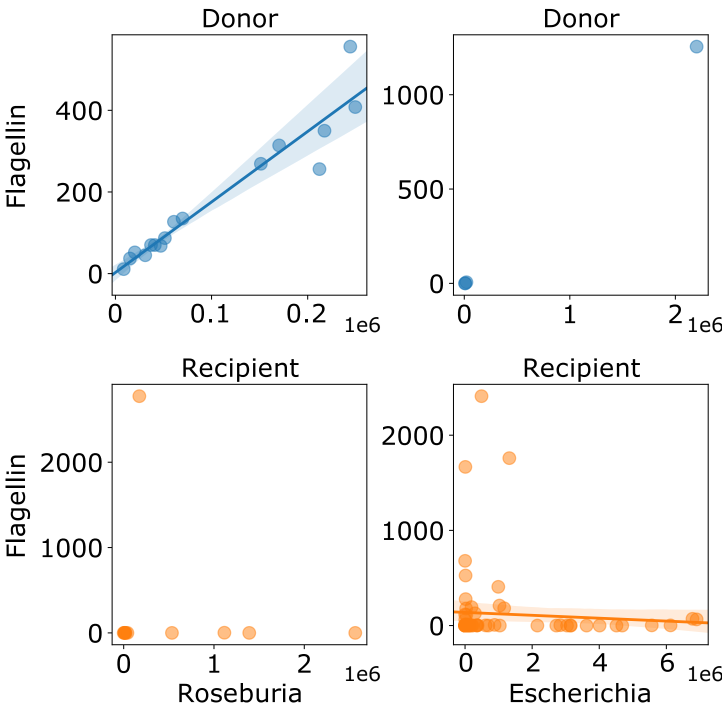


Figure 3.8: Number of reads assigned to genera *Roseburia* and *Escherichia* (x-axis) and their flagellins (y-axis). The numbers are absolute counts. A linear regression line is plotted for *Roseburia* in donor, and *Escherichia* in recipient samples.

3.5.3 Potential biases in the sequence abundance estimation

Sun *et al.* have recently called attention to the differentiation between sequence and taxonomic abundance estimation in microbial profiling. Both types estimate the abundance from reads-to-reference alignment. Sequence abundance derives from the alignment to a whole set of microbial elements, while taxonomic abundance is based on the alignment to markers[202]. In other words, the former calculates how much genomic content comes from an individual taxon, while the latter estimates the number of organisms. Tools like Kraken2[203], DIAMOND[185], and Kaiju[204], which are based on reads to reference genomes or proteins, estimate sequence abundances, while others like MetaPhlAn[205] and mOTUs[206] perform profiling based on the marker genes. Sun et al performed benchmarking, with which they concluded that microbiome diversity analysis based on sequence abundance has the tendency to overestimate the microbes of larger genome size and the other way around for the smaller genomes.

Our taxonomic profiling was based on the sequence abundance. MEGAN+ DIAMOND pipeline was able to capture taxonomic information to an extent, because using the entire nr as a reference compensates for the impact of missing genomic regions. However, we warn that the lack of proper representation of the metagenome potentially added a great cost to the accuracy of abundance estimation, especially for the recipient samples. And for our dataset, the bias was not only introduced by the intrinsic size differences between individual genomes. In the low-biomass samples, genomes with larger fragments included in the sample are likely to be overestimated than the rest. As for taxonomic abundance, to our knowledge, many microbial single-copy marker genes frequently appear in close proximity to each other. Given the high possibility that many genomic regions may be absent in our recipient samples, the calculation of taxonomic abundance can be even more prone to bias than that of sequence abundance. While the sequence abundance-based analysis provided evidence for the significant differences between the two groups, it shall not be taken as a conclusive or definitive ground for any quantitative interpretations.

3.6 Conclusion

We performed the metagenomic analysis for ImMiGeNe, a project that follows the gut microbiota development of patients who underwent SCT treatment, as well as their stem cell donors. The ImMiGeNe consortium focuses on the dynamics between the host immune system and the gut microbial community. To address this goal, we first removed contaminating components from the raw data, then performed a series of taxonomic and functional analyses. An evident contrast was revealed between the donor and recipient groups. Compared to the raw data of donor samples, that of the recipients typically yielded fewer reads, and many suffered from a high level of host DNA contamination and duplicates. We identified that the donor gut microbiomes were marked by a highly similar abundance of obligate anaerobic microbes, and that of the recipient had a highly dissimilar dominance of facultative anaerobes, among which many were known pathogens. Further, in the functional analysis, we focused on the respiratory pathways, as they have been widely reported to be strongly associated with colonic inflammation. The result showed an enrichment of key genes in both anaerobic and aerobic electron transport chain, in the recipient group. This suggested that the recipient gut microbiota were likely undergoing a secondary succession process. Even till the end of the sample collection, a functional, balanced ecological equilibrium had not been achieved. This implies a long span of gut microbiota development under a compromised host immune system. We also addressed the major challenges and limitations to the metagenomic analysis, many of which came from the fact that the recipient samples contained extremely low levels of DNA of the targeted gut microbial community (low-biomass). These samples are susceptible to both host and laboratory contamination, and tend to represent the metagenome only partially. Despite the computational precautions, this likely has led to the distortion of microbial profiles, hindered the feasibility of other regular microbial analyses, potentially narrowing the scope and undermining the credibility of the taxonomic, functional, and ecological interpretations of the dataset. Altogether, we provided a detailed description of the metagenomic makeup of gut microbiota inhabiting a challenging and unique physiological environment. We also offered insights into the metabolic landscape, linking it to the hidden ecological processes at play. These findings point to some promising new directions for further research of the ImMiGeNe consortium. Additionally, this work highlights the unconventionality of our samples and the importance of careful sample preparation and data generation.

Chapter 4

Designing primers for diverse flagellins in human gut

Different types of flagellins exhibit distinct levels of immunogenicity. One possible way to obtain various flagellins is to extract their genes by PCR, clone and express them *in vivo*. Bacterial flagellins make up an immensely diverse family. A quick browse through UniProt database (https://www.uniprot.org/) shows 47,568 entries, at the time of the chapter being written. Generally speaking, structurally, the two terminal domains of flagellins, D0 and D1, form a helical core and are relatively conserved. They are connected by a hypervariable region consisting of a few other domains. The amplification must be able to cover all the domains in order to have a fully functional protein in the end. The challenging tasks are (1) to compile a coherent list of flagellin genes; (2) to design a reasonable set of primers that could efficiently amplify as many of the genes as possible. We use FliC proteins and their encoding genes as a start, but the workflow could also serve as a protocol in general for a lab design problem of this kind.

4.1 The primer design problems

For almost four decades, PCR has been widely used to amplify specific regions of nucleic acids from meager to desirable amounts. PCR could fulfill the goal only if the primers are properly designed. The configuration of the primer design problem depends on what the PCR is applied to. It could be to detect certain genetic elements or variations, identify gene expression or epigenetic changes, prepare transcripts for gene cloning and etc. The goal may be extracting one specific sequence from the sample, retrieving a few loci from one genome at the same time, or capturing a family of genes simultaneously from some DNA mixture. In this chapter, we mainly focus on the primer design problems that deal with genetic variations. We first have a look at the computational problems, reviewing a few variants roughly in the order of their complexity.

4.1.1 Basic primer design problem

We shall first clarify a few terms to avoid confusion. A *target site* is the region on the template that we want the PCR product to flank. The *amplification site* is the region that is actually amplified into PCR products. The two are not exactly the same. The amplification site shall surround the target site. The *annealing site* is the region that anneals to the primer.

A DNA segment can be amplified with a pair of primers that have complementary sequences to its two ends. But sequence complementarity itself does not ensure the success of PCR. Usually, more factors shall be carefully considered too. We list a few most common aspects as follows:

1. specificity of the primers. It is crucial to make sure the primers have little chance to bind to anything else but the target DNA. This means the annealing sites shall be found only at the target region. To ensure this, first, primer lengths shall be sufficient to avoid random occurrence. Conventionally, they are 18 to 30 bp in length. Second, ideally, the annealing sites shall have no other copies at unintended genomic locations.

2. melting temperature of the primers (T_m). This is the temperature at which half of the primers would bind to the template. The T_m difference between the forward and reverse primers should also be within 5 °C;

3. the GC content of the primers, which not only affects T_m, but also decides the stability of the primer during its synthesis;

4. avoiding the formation of primer-dimers, in which the forward and reverse primers form a dimer, and the formation of a hairpin, in which one primer with palindrome subsequences forms a hairpin secondary structure on its own.

5. GC clamps, which are the presence of guanine (G) or cytosine (C) bases within the last 5 bases of the 3' end of a primer. This is because a GC clamp enhances primer binding specificity.

Designing eligible primer pairs for one specific DNA template is what we consider a basic type of primer design problem. The template sequence is given. Usually, the goal is to have PCR products containing certain target regions. The product sizes are limited to not exceed the template length, nor what was allowed by the sequencing platform to be used next.

Such a problem can be solved in two steps: first, come up with candidate pairs of primers, and second, check the eligibility of the candidate primers. In the first step, exhaustively find out all possible amplification sites, then for each of them, come up with a pair of oligos that are complementary to its two ends. They are candidate forward and reverse primer sequences. In the second step, candidates that do not fit the constraints are eliminated. Those that are closer to the optimum are prioritized. The constraints and optimums are usually set in light of the aforementioned aspects. For instance, the minimum and maximum product sizes are constraints, and at the same time, sizes within a certain range could be the most ideal. The same goes for T_m.

One widely used toolkit that addresses this problem is Primer3[207]. It contains a group of programs that are capable of handling a variety of PCR primer design scenarios, such as proposing primers for amplifying a DNA segment, evaluating given primers for given templates, or designing hybridization probes. Specific needs of the application case are described in the form of constraints and optimums for the PCR reaction, and the algorithm looks for suitable solutions by obeying the constraints and optimizing the objective functions. We adapted an example in the Primer3 manual to demonstrate a use case (Table 4.1). We input a 108bp template, asking the program to design a primer pair that is able to flank a CA repeat, along with other specified parameters. The program outputs a primer pair for amplifying an 88bp segment, with characteristics such as T_m and GC content just as specified. (See Table 4.2, original program output in Supplementary Fig A.3.)

This basic form of the problem serves as the building bricks of some more complex questions. For instance, primers for a few tens or hundreds of template sequences can be designed by repeatedly using the method that designs primers for a single template. However, with a large number of template sequences, this clearly can soon become fastidious. Besides, it offers

Table 4.1: An example of picking PCR primers for amplifying a target sequence using Primer3.

Content to set	**Primer3 Boulder-IO line**
Basic settings	
Template sequence	SEQUENCE_TEMPLATE=GTAGTCAGTAGACNATGACNACTGACGATGCAGACNACACAACACACACAGCACACAGGTATTAGTGGGCCATTCGATCCCGACCCAAATCGATAGCTACGATGACG
Target site located at the 37th base with a length of 21	SEQUENCE_TARGET=37,21

Pick left and right primers, not internal primers	PRIMER_PICK_LEFT_PRIMER=1
	PRIMER_PICK_INTERNAL_OLIGO=0
	PRIMER_PICK_RIGHT_PRIMER=1

Design a primer pair for sequence detection (not for other purposes like evaluating primers)	PRIMER_TASK=generic
Constraints	
Minimum primer size	PRIMER_MIN_SIZE=15
Maximum primer size	PRIMER_MAX_SIZE=21
PCR product size range	PRIMER_PRODUCT_SIZE_RANGE= 75-100
Maximum T_m of the primers	PRIMER_MAX_TM=45.0
Minimum T_m of the primers	PRIMER_MIN_TM=75.0
Maximum T_m difference between forward and reverse primers	PRIMER_PAIR_MAX_DIFF_TM=5.0
Number of consecutive Gs and Cs at the 3' end of both forward and reverse primers (not required)	PRIMER_GC_CLAMP=0
Allowed Ns in primer	PRIMER_MAX_NS_ACCEPTED=1
Optimums	
Optimal primer length	PRIMER_OPT_SIZE=18
Optimal GC content	PRIMER_OPT_GC_PERCENT=50.0
Optimal T_m	PRIMER_OPT_TM=60.0

Table 4.2: Primer3 output for the example input in Table 4.1. The primer pair is selected from 4,124 candidates.

```
 1 GTAGTCAGTAGACNATGACNACTGACGATGCAGACNACACACACACACACAGCACACAGG
61 TATTAGTGGGCCATTCGATCCCGACCCAAATCGATAGCTACGATGACG
```

∗ annealing sites in blue; target site in brown. The numbers are base indices.

	Sequence	Start	length	T_m	GC%
Forward	ATGACNACTGACGAT	15	18	51.58	47.06
Reverse	CGTAGCTATCGATTTG	102	20	55.85	50.00

no condensation of the number of primers yielded. If there is a lack of consensus between the proposed primers for each sequence, the number of primers needed can grow unrealistic for experiment implementations. In the case of a large number of target sequences, the top priority will be to efficiently come up with a set of primers that is as concise as possible to cover as many targets as possible.

4.1.2 Primer degeneracy

Incorporating positional ambiguity within the primers proves advantageous in effectively increasing primer coverage. This approach is especially applicable in scenarios where template sequences are closely related and thus have variations at only a few sites while the rest are identical. Another use case is when the target is a protein, and only its amino acid (AA) sequences are known. To design the primer, the encoding gene is deduced from all the possible combinations of the codons of each AA, which will result in a few sites with multiple possible bases. The uncertain sites can be designed with ambiguous bases to cover all possibilities.

A primer is *degenerate* if some of its positions are made of a mixture of possible bases. Such primer represents the whole set of all possible base combinations. Degenerate primers are as commercially available as non-degenerate ones. As more wobble positions are allowed, more targets could be covered by the primer, but at the same time, the specificity of the primer decreases. Highly degenerate primers would lead to a lot of off-target sequence patterns that are not present in the input sequence set and as a result, create unwanted amplicons. Therefore, the tradeoff is between the degeneracy of primers and the target coverage.

The *degeneracy* of a primer for a set of targets is calculated as the sequence product of the number of possible bases at every position. Or precisely, for a degenerate primer p, let p_i be the set of all possible bases at the ith position in p ($p = p_1p_2 \dots p_k$). The degeneracy d of p is $\prod_{i=1}^{k} |p_i|$. For instance, $p = \{A, T\}\{C\}\{A, T, G\}$ has degeneracy of 6. A primer is considered to be able to cover a target if the target sequence is one of its combinations.

Introducing degeneracy to the primers could be particularly useful when dealing with multiple template sequences, although it does add to the complexity of the computational problem. In the following context, we sort the primer design problem for a set of templates into two types, based on whether the primers are degenerate or not.

4.1.3 Classic non-degenerate primer design for multiple template sequences

The goal is to pick a minimum set of primers for a given set of target sequences. This problem can be applied to studies of a gene family of interest. Proper primer design based on a few known members of a gene family can lead to the discovery of undocumented novel genes. In this case, although designing a distinct pair of primers for every given target sequence can be carried out as demonstrated in Section 4.1.1, it will likely end in a large number of primers. Moreover, the high specificity of the primers usually hinders the discovery of undocumented genes. Therefore, we instead seek a method to reduce the candidate primers to a succinct set that is able to take advantage of the sequence similarity and capture as many related sequences as possible.

A few notions are different in this context compared to the basic primer design problem. First, the template sequences are reduced to the regions where the genes occur, and the rest of the target sequences are ignored. Second, forward and reverse primers are no longer designed simultaneously. Here, one end is processed at a time, and the algorithm is repeated for the other end.

This is an NP-complete, decision problem[208, 209] that could come in a variety of forms. For better understanding, we first introduce a classic variant that addresses the goal of gene detection, which means the primers only need to capture the distinction of the templates even if the resulting product only contains a partial gene sequence.

In plain words, the problem is to minimize the number of non-degenerate primers required to amplify a set of DNA sequences. To describe it mathematically, a DNA sequence is a string. For a set of template sequences, a set of primers is a *cover* for the templates if it allows all targets to be amplified. The lengths of the primers do not have to be uniform in practice, but we set a specified length for all primers for the sake of simplicity. In this way, the basic form of the problem can be formulated mathematically as follows[208]:

(Non-degenerate) optimal primer cover problem (OPC). Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k, find a minimum set P_{nd} of strings of length k that covers S.

This could be generalized to a *set cover problem*[208], a classic combinatorial problem (See Supplementary A.1.1), which is proven to be NP-complete[210]. As described in [208], the OPC problem could be deemed as a special case of the set cover problem, and therefore is NP-complete as well. As the latter is proven to be NP-complete[210], OPC problem is also NP-complete. But there are a few established heuristics for this classic combinatorial problem that can be adopted to solve OPC problem[208, 211].

4.1.4 Classic degenerate primer design for multiple template sequences

When the template sequences are not in a broad consensus, degeneracy can be introduced to primers to increase the coverage. Similar to its non-degenerate counterpart, in this problem, the forward and reverse primers are designed separately. The objective is to maximize the coverage among targets while minimizing the primer degeneracy. This is crucial as higher degeneracy increases the likelihood of amplifying off-target regions.

The very basic form of the degenerate primer design problem is, given a set of templates, design a primer with a defined length and degeneracy of no more than a threshold that could amplify as many templates as possible. Mathematically, it is:

Degenerate primer design problem (DPD). Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k and d, come up with a degenerate primer p_d of length k and degeneracy at most d that maximizes the number of covered strings c in S.

To simplify the problem, we assume that the correspondent locations of the primer have already been extracted from each template. In this way, the problem is reduced to finding a consensus string for a list of strings. The strings in S are assumed to be of the same length as the primer. Locating in each template string the substring that matches the primer string is not included in this definition. In reality, it is fulfilled by picking a gap-free window as the primer annealing region from the multiple sequence alignment (MSA) of templates.

The primer length, k, is always bounded (to 18-30bp, for instance). To come up with a solution, we also bound the degeneracy d to an upper limit while trying to optimize the coverage c. Note that it is impractical the other way around, seeking a minimal d needed for achieving full coverage, as in most real-world cases, the resultant d would be too high to be practical.

DPD problem alone is NP-complete. Linhart and Shamir, authors of HYDEN, proved so by reduction from the clique decision problem[209], which is another classic complexity problem[212]. The detailed explanations can be found in Supplementary A.1.2.

A few heuristics can be used for the problem. Two example heuristics are *contraction* and *expansion*[209]. They both make approximations by turning the coverage maximization problem into a mismatch minimization problem, that is, to minimize the number of sequences that the primer fails to cover. The first step is to calculate the base distribution matrix from a block of MSA. The matrix takes note of the counts of A, T, G, and C in each column of the MSA. Then we flatten the matrix into an array and sort it by the base counts. Next, for *contraction*, we start with a primer of full degeneracy, i.e., a primer that encompasses all possible bases at each column. This primer covers all the sequences in the MSA, but its degeneracy is usually higher than the required upper limit. If it is the case, we remove a base of the smallest count because they would cost the least coverage loss for the primer. Then we recalculate the degeneracy and repeat this step until the degeneracy meets the requirement.

Expansion algorithm works the other way around. We start with a primer of 0 or very low degeneracy, so the starting point meets the degeneracy requirement, but at the same time, in most cases, leaves quite a few input sequences not covered by the primer. Then in order to increase the coverage, we gradually add alternative bases to the primer until the degeneracy rises to the upper limit.

For our own use case, we have implemented a heuristic based on the *con-*

traction algorithm, whose details are demonstrated later on in Section 4.5.2.

4.1.5 Degenerate primer design problem with mismatches

More intricate variants of the DPD problem can be built by including additional factors. We will first introduce the variant involving mismatches between the primer and template.

In the previous discussion, a degenerate primer is considered to be able to cover a template sequence only when one of its combinations is an exact match to the template. In reality, a primer could still anneal with a few mismatching positions involved, especially when they are closer to the 5' ends. We add the parameter e as the maximum number of mismatches allowed for the primer to anneal to a template, and a degenerate primer *covers* a template when at least one of its combinations matches the template with at most e mismatches. This problem is inspired by the MD-EDPD problem in [209].

DPD with mismatches. Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k, d and e, come up with a degenerate primer p_d of length k and degeneracy at most d that maximizes the number of covered strings c in S with up to e mismatches.

4.1.6 Degenerate primer set design problem

The basic DPD seeks to find one primer for a set of templates. We extend the objective to finding a limited number of primers. This could be generalized to the following two DPD variants that aim to find a set of degenerate primers, which are similar to the MP-DPD problem in [209].

Minimum primer set size DPD Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k and d, find a minimum set P_d of strings of length k that covers all of S, with degeneracy up to d.

Maximum primer set coverage DPD. Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k and d, find a set P_d of up to m strings, each string of length k, that maximizes the number of covered strings c in S, with degeneracy up to d.

The first looks for optimization of the primer set size that achieves full coverage, and the second looks for optimization of the coverage while constraining the size of the primer set.

4.2 Software that designs primers for multiple template sequences

There are a variety of primer design programs, both open-source and commercial, motivated by the evolving specialization of PCR applications. As illustrated in Section 4.1.1, designing primers for one or a handful of template sequences is, in general, a basic task that can be solved with a quick, straightforward use of Primer3. Many online tools or simple programs also serve this goal. But with a larger number of templates, some different computational concerns arise. Theoretically, Primer3 is an all-encompassing toolkit that could be tailored to address such new concerns. However, to customize it properly, it takes much meticulous manual configuration and parameter manipulation[213].

We will briefly discuss some more efficient and automated methods. We aim to target the application to large gene families and therefore focus on degenerate primer designs.

4.2.1 Computational framework to design a set of degenerate primer

We outline a 3-step general framework as follows, including not only the aforementioned DPD problems but also certain necessary pre- and post-processing:

1. Compute MSA of the template sequences. As mentioned earlier, the DPD problem operates on a *window* of MSA. The *windows* shall be some locally aligned regions of the template sequences that have zero or very few gaps involved. In this step, we first make MSA for the template sequences, then locate the regions where it is suitable for the forward and reverse primers to be designed at, and pass them on to the next step.

2. Solve the DPD problems. In this step, for each *window* given by the last step, solve the DPD problem and output a primer or a primer set, which will be treated as the candidate degenerate primers for the next step. Note that the forward and reverse primers are designed separately here.

3. Pair up the forward and reverse primers and check their eligibility. First, the separately designed two ends shall be combined as a pair. Next, calcu-

late their chemical properties as discussed in Section 4.1.1, and filter out those that fail to satisfy the reaction conditions.

4.2.2 HYDEN: a computer program for degenerate primer design

We will use HYDEN as an example for the workflow above. HYDEN is a software for degenerate primer designs. It could handle hundreds or thousands of DNA sequences, with extremely high degeneracy upper limit (e.g., 10^7). Its algorithm consists of 3 major steps, written in C++: (1) Find ungapped, conserved windows in the alignment. The conservation of a window is quantified with an entropy score. (2) Solve the DPD problem using adapted *contraction* and *expansion* algorithms. (See Section 4.1.4) (3) Perform a greedy hill-climbing to improve the fitness of the primer solution, then output a primer that gives the highest coverage. The algorithm solves one DPD problem case at a time, but after designing one primer pair, the procedure could be repeated on the target sequences that were not covered by the first solution. In the end, the program is able to come up with a set of primers that covers the target sequence set well[209].

4.2.3 Other available software for degenerate primer design

Although a variety of primer design tools has been released since PCR technology emerged in the 1980s, as it has become a routine task in biological laboratories, primer design has seen fewer software implementations and updates recently. Here we will briefly name a few more software besides HYDEN that are capable of designing degenerate primers for multiple target sequences, automated, free, non-commercial, and still runnable as of today. PriFi takes the alignments of a group of phylogenetically related sequences, finds highly conserved regions, and proposes primer pairs which are scored based on the fitness of their properties. It is available as a web server[214]. GeneFisher is another web-based tool that proposes degenerate primers from the MSA of input sequences. It could take multiple sequences and perform the alignment itself. Besides DNA sequences, GeneFisher also works on protein sequences, for which it will perform a back-translation to DNA sequences. GeneFisher was released in 1998 and had a few upgrades over the next decade to improve its user-friendliness[215]. A more recent tool is Gemi, which comes with a graphic user interface. It works with MSA as input and also proposes primers with a report about their chemical properties[216]. We noticed a dearth of such programs in recent years, except for a few niche products that are made for specific application cases.

4.3 Designing primers for bacterial flagellin genes

In the ImMiGeNe project, our goal was to characterize microbial flagellins in the human gut comprehensively. To achieve this, we encountered the challenge of designing primers that would enable us to extract the target genes from stool samples accurately.

4.3.1 Amplifying a set of genes from environmental samples

Amplifying a group of related genes from mixed-population samples such as environmental samples (eDNA) is not a new nor recent practice. Since the beginning of the NGS era, researchers have been profiling taxonomic groups in the samples by amplifying and sequencing phylogenetic marker genes. Amplicon sequencing targeting 16S rRNA genes for prokaryotes and 18S rRNA genes or ITS regions for fungi are widely used methods to investigate microbial diversity. In animal studies, mitochondrial cytochrome *c* oxidase subunit I gene (*COI* or *cox-1*) is used as the genetic barcode[217]. But note that an important factor that ensures the success with marker genes is that they have certain well-conserved regions so that they could be covered with a handful of universal primers. In fact, many other widespread gene families, despite conservation in their function, may display greater divergence in the sequences, which places a major challenge for designing universal primers or probes. And being environmental samples is already a factor contributing to problem complexity, as it means no prior knowledge about the source genomes of target genes. As a result, studies that work with eDNA tend to work with a set of target genes whose diversity is at a manageable level. A good example of non-marker gene is that Razavi and colleagues amplified class I integron gene cassettes from river sediments, with only 3 primer pairs[218]. And quite a few studies that involve large, divergent gene families choose to limit the target to a subset, for instance, members from certain taxonomic groups.

4.3.2 Scope of the project

The purpose of this project is to design a set of primers that are able to amplify at least a great majority of bacterial *fliC* genes in the metagenomic samples. The ImMiGeNe project involves human stool samples, which were collected for the characterization of the gut microbiome. As cloned genes are expected to be further expressed into fully functioning proteins, the amplicons must include all regions that encode all essential domains of the FliC proteins.

Studies involving amplifying *fliC* genes mostly use a single species or strain as the DNA material, and their primers are designed for a single bacterial genome. To our knowledge, no universal primers targeting the entirety of the flagellin diversity have been publicly available. Few studies have tackled the complexity of environmental samples and the whole *fliC* gene family at the same time. In fact, very rarely has a study reported the effort of simultaneously capturing a large group of divergent genes from eDNA.

A few key details shall be highlighted. Firstly, for the FliC protein product to contain all essential domains, the primer pair must be designed to include start and stop codons. Therefore there is no freedom to consider other conservative regions but the two ends of the genes. Secondly, the goal of our work is to design only the annealing part of the primer. A full-length primer has its very 3' end to anneal to the target sequence, and the 5' tail could contain additional sequences such as restriction sites, linker sequences, etc. The design of the 5' tail is not included in this work. The *primers* in this chapter generally refer to the segments that anneal to the target sequences. Despite their availability, the aforementioned degenerate primer design software will not be adopted for our project, for a few reasons. Most prominently, it is due to the sheer volume of our target sequence set, as we will explain in the next section (see Section 4.4). Most of these tools were designed for the volume of sequencing data a decade ago, and would not expect as many as tens of thousands of target sequences. Another key reason is that a common core feature of these programs is the selection of conservative regions, which is unnecessary for our project, as we have fixed primer sites. Finally, we will not make a real MSA for the sequences, which we will also explain later (see Section 4.4). Therefore, we could not offer the input those programs require.

4.4 Collection of the target *fliC* genes

4.4.1 Acquiring FliC protein and gene sequences from AnnoTree database

To analyze the diversity of FliCs and design primers, both their protein and encoding gene sequences are needed. Popular protein databases such as NCBI (`https://www.ncbi.nlm.nih.gov/`), UniProt (`https://www.uniprot.org/`), Pfam (`https://pfam.xfam.org/`), TigrFam (`http://tigrfams.jcvi.org`) and InterPro (`http://www.ebi.ac.uk/interpro`) could be used to navigate reported flagellins by their similarity. But despite their crosslinks to nucleotide databases, they are usually protein-oriented, and tracing back proteins to their source genomes can be challenging. Therefore, we opted for an alternative approach using AnnoTree, which allows us to obtain CDS and their corresponding proteins. AnnoTree leverages prokaryotic genomes from GTDB to generate a functionally-annotated, interactive tree of life for bacteria and archaea[219]. It ensures a standardized and consistent dataset for our analysis.

We used KEGG ortholog (KO) ID K02406 to search for FliC, with percent identity 70%, E-value 1e−5, percent subject alignment 70%, and percent query alignment 70%. This yields 18,667 hits, with their protein sequences, source genome, and taxonomic information recorded. AnnoTree visualization of their distribution on the bacterial tree of life is shown in Figure 4.1(a). The detailed taxonomic statistics can be found in Supplementary Table A.8. They are the basis for all the subsequent sequence analyses.

4.4.2 Sequence diversity analysis

First, we evaluate the sequence variability of the targets. Out of the 18,667 hits, 392 are duplicates. Overall, the hits are from 9379 species and 55 phyla (suffixed names are counted independently, not as one). Over 11,000 hits come from Proteobacteria, and over 3,000 are from Firmicutes (including Firmicutes, Firmicutes_A, Firmicutes_B, . . . , Firmicutes_G). Other main clades are Spirochaetota, Campylobacterota, and Actinobacteriota. (See Figure 4.1 (a) and (b)).

The lengths of the flagellin protein sequences range from below 100 to above 1000 AA, 349 AA on average (see Figure 4.1 (c)). Following the primer length conventions, we set to design primers of lengths between 18 and 30bp.

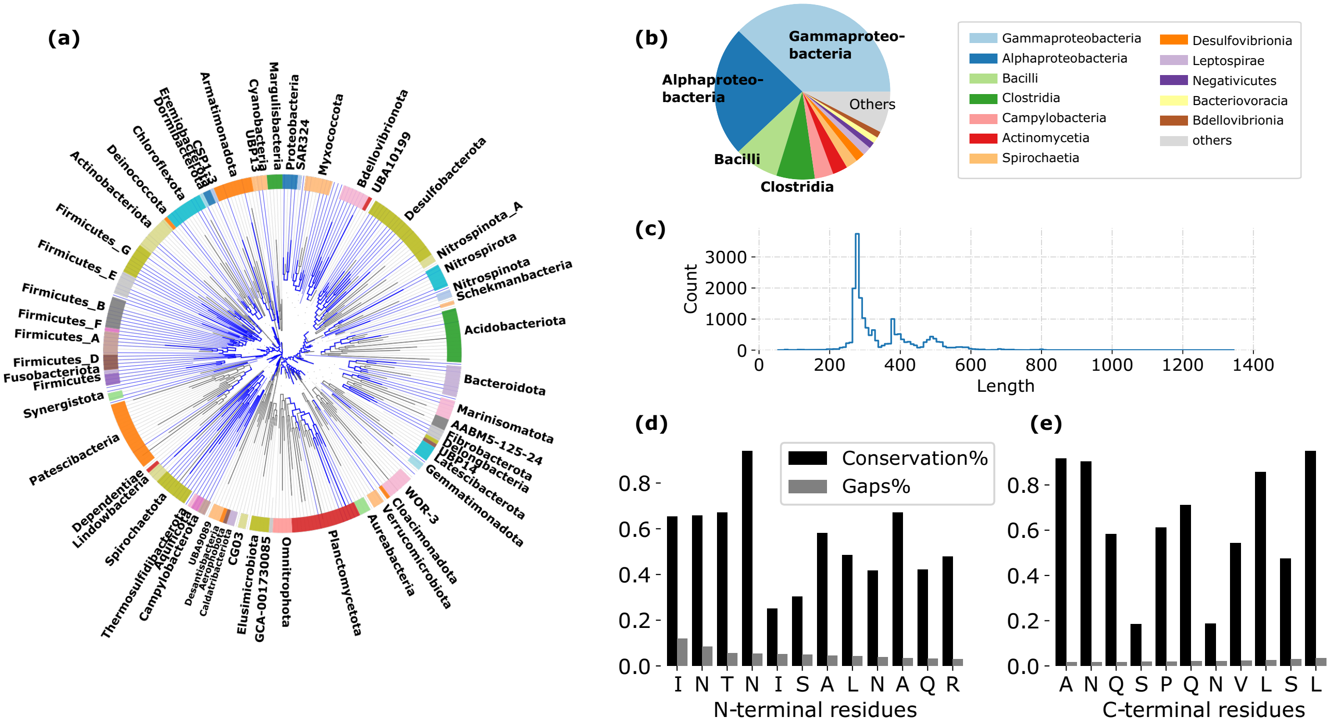


Figure 4.1: Summary of AnnoTree FliC (KO K02406) hits. (a) AnnoTree visualization of the hits in the bacterial phylogenetic tree. The tree uses GTDB taxonomy[59]. Tree leaves and labels are at the class and phylum levels, respectively. Clades in blue are hits. (b) Taxonomic summary of the hits at the class level. (c) Flagellin protein sequence length distribution. (d) and (e) conservation and gap bar charts of each position in the first 12 residues in the HMM alignment of the N-terminus and the last 12 in that of the C-terminus. X-axis directions are both from N- to C- terminus.

Our focus is on the two ends, N- and C- termini, while the variability of the sequences in between is irrelevant. To check the conservation of the AA residues, we aligned the two terminal HMM profiles to the AnnoTree flagellin sequences respectively (Pfam Flagellin_N and Flagellin_C, PF00769 and PF00700), using `hmmsearch` from HMMER [220] package (E-value 1e$-$5, other parameters as default). With the alignment, we capture the location of terminal domains in each protein sequence and, correspondently, in each nucleotide sequence. The forward primer will be designed at the start of the N-terminal codons and the reverse at the end of C-terminal codons. Although the primer locations are anchored, their lengths could still be adjusted.

The terminal domain alignments are low in gaps, which is a positive factor for primer design. But among the very first residues at N-terminus and the very last at C-terminus, many sites are not highly conserved (see Figure 4.1(d) and (e)). The nucleotides at those sites show even higher sequence diversity (see Figure 4.2 (a) and (b)). On the DNA template, at both ends, especially 3', many sites have below 60% conservation.

4.5 Primer design for target *fliC* genes

4.5.1 Non-degenerate primer design

To design primers for the targets, we first formulate an OPC problem (see Section 4.1.3), as this could quickly give us an impression about the computational scale. The question in a biological context is, for a given set of target DNA sequences, design a minimum set of non-degenerate primer pairs that cover them all. We add mismatches e as a parameter. That is, for a primer to anneal to the target, it could have at most e mismatches to the target. Compared to the classic OPC illustrated in Section 4.1.3 (or [208]), our problem is much simplified because our primer site is strictly limited. There is no need to decide the locations of the match. We simply approximated our problem as a clustering problem. For the forward primer design, we take the k bases from the 5' end of the N-terminal nucleotide MSA, and for the reverse, from the 3' end of that of the C-terminus. Note that in reality, mismatch location matters. It is more problematic when it happens near 3' end. But to simplify, we will leave out this factor.

To cluster these fragments of length k, we used CD-HIT-est[221], taking them as paired-end at a sequence identity of $(k-e)/k$. With $k = 21$ and $e =$

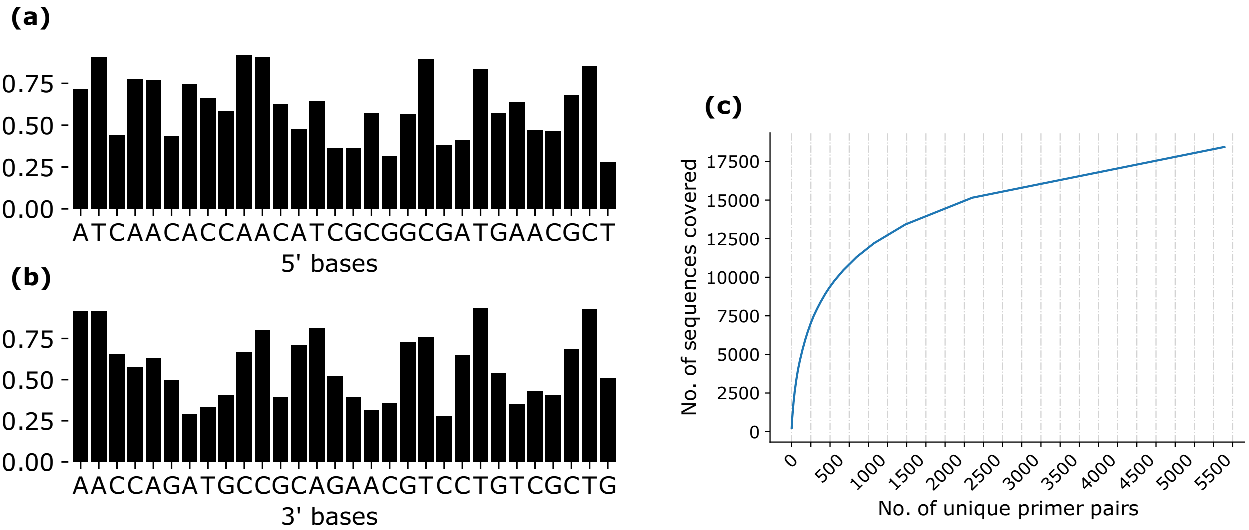


Figure 4.2: Statistical summary of DNA templates and primer coverages. (a) and (b) nucleotide sequence conservation rate per position, 30bp from the start of the HMM pattern of the N-terminus and 30bp to that of the end of the C-terminus. Both x-axes are from 5' to 3'. (c) The number of pairs of primers needed to cover all AnnoTree flagellins. Primers are sorted by the number of sequences (n_{seq}) they could cover, then counted. In other words, an x-axis value of N means the top N primer pairs sorted by their n_{seq}, and its y-axis value is the total number of sequences they could cover.

3, 5643 clusters are yielded. Each cluster can be covered by a representative, which leads to one non-degenerate primer pair that could cover the whole cluster. This gives a solution to the problem that requires 5643 pairs of primers to cover all targets.

Here, the trade-off is between the number of primers and that of covered targets. Alternatively, and more practically, we seek for a minimum primer set that achieves certain coverage. As shown in Figure 4.2 (c), 500 primer pairs could selectively cover more than half of the targets. But as the other half of targets exist only in small clusters or exist only as a singleton, they largely extended P_{nd}. We could not expect to cover a vast majority of the input target sequences without employing thousands of non-degenerate primers.

4.5.2 Degenerate primer design

As more than five thousand primers are far from being feasible, we now explore the possibility of using degenerate primers. The problem is formulated as a DPD problem (see Section 4.1.4), and more precisely, a combination of the DPD problem with mismatches (see Section 4.1.5, and minimum primer set size DPD (see Section 4.1.6). We define it as, for a given set of target DNA sequences, finding a minimum set of primers with a fixed length, an upper limit for their degeneracy, and an upper limit for their mismatches to the targets. Or mathematically:

Minimum primer set design with mismatches. Given a set of n strings $S = \{s_1, s_2, ...s_n\}$ over alphabet $\Sigma = \{A, T, G, C\}$, integers k, d and e, find a minimum set P_d of strings of length k that covers all of S, with degeneracy up to d, and mismatches up to e.

As explained in Section 4.1.4, the forward and reverse sequence sets were processed separately and combined only at the end. We took a few approximation methods in order to find a solution. First, we took the same clustering result in Section 4.5.2 to remove the mismatch factor, so we have a set of 5643 representative sequences for each end, which we set S as. Next, We took two steps to approximate the set design problem: (1) turning it into a collection of subproblems of basic DPD; (2) using heuristics to solve the basic DPD.

We partitioned the input string set into subgroups based on sequence similarity. Then, for each subgroup, the goal is to design one degenerate primer, given its length k and degeneracy upper limit d, which is a basic DPD problem by nature. Since CD-HIT-est only clusters at a high sequence identity, we used affinity propagation clustering[222] with a pairwise hamming distance matrix of S, and retrieved 409 and 393 clusters, respectively for 5' and 3' ends. Note that this only partitions the input data, but gives no guarantee that an ideal degenerate primer must be found for each cluster. The clustering manifests low similarity between 3'-end sequences, as the convergence is more difficult to achieve at the same damping factor.

For a cluster $S_i \subseteq S$, we use the *contraction* algorithm to find a degenerate primer. This is a simple approximation method similar to the H-CONTRACTION algorithm used by HYDEN [209]. Let matrix of D be the base distribution for S_i, $D(char, loc)$ denotes the number of character *char* at the position *loc* for the MSA of S_i. An example is shown below, in which $D(G, 2) = 2$ means there are 2 Gs in Column 2 of the alignment.

(a) Sequence strings (S_i)

A	T	C	C
A	G	G	C
T	G	T	G

(b) D

	1	2	3	4
A	2	0	0	0
T	1	1	1	0
G	0	2	1	1
C	0	0	1	2

The algorithm first makes a primer that encompass all possible base combinations, and therefore with full degeneracy. Then it remove low-frequency bases one by one, until the degeneracy is reduced to or below required level d. The pseudocode is shown in Algorithm 1.

We used this approach to find an optimal primer for each cluster and computed its coverage within the cluster. The result shows that a good trade-off between primer degeneracy and sequence coverage is difficult to achieve due to the high sequence diversity. For our metagenomic samples, we do not expect high degeneracy d could achieve great specificity. Therefore, we computed only for degeneracy no more than 10^5. Setting d to 128 and k as small as 18, the average coverage of primer is only 0.62 and 0.50 for 5' and 3' ends, even with 1 more mismatch allowed between the primer and the target (Figure 4.3). Even with d as high as 2048, the average coverage of the primers is still below 0.8. This means, even with a few hundred degenerate primers, the full landscape of bacterial flagellin diversity is unlikely to be well captured. Furthermore, pairing up the forward and reverse primers ended in 4156 unique pairs. This is unsurprising, as similarity in one terminus does not indicate that in the other. As a snippet of the resultant primers, Table 4.4 lists those that cover a subgroup of more than 100 targets at coverage above 0.8.

Surely our approximation method could be further optimized, but the chance of significant improvement is slim. Practically, it is difficult to avoid using a large number of primers to counter the effect of high sequence diversity without compromising coverage or annealing specificity. Our method divides the input sequences into small similarity groups of less than 50 sequences and assigns one primer for each group. But even with a highly elevated level of degeneracy, the number of primers needed in this scenario is intensive to be implemented in regular wet lab experiments, and the coverage is far from satisfactory. Until this point, it is clear that if we target full-range bacterial flagellins for amplification, it is very unlikely we could find a realistic balance between the amplification specificity, number of primers needed, and target coverage.

Algorithm 1 Contraction

```
Flatten $D$ as an array $A$
$A = \{(char_1, loc_1), (char_2, loc_2), \ldots, (char_k, loc_k)\}$
sort $A$, so that for $A_i \in A$, $D(A_1) < D(A_2) < \ldots < D(A_k))$
$p = p_1 p_2 \ldots p_k$
for $loc \leftarrow 1$ to $k$ do
    $p_i \leftarrow \emptyset$
    for $char \in \Sigma$ do
        if $D(char, loc) > 0$ then
            add $char$ to $p_i$                    ▷ initialize $p$ at full degeneracy

$d_p \leftarrow CalculateDegeneracy(p)$
while $d_p > d$ do
    $(char, loc) \leftarrow$ pop $A_1$                                  ▷ update $A$
    if $|p_{loc}| > 1$ then                  ▷ no action if there is only 1 base left
        remove $char$ from $p_{loc}$                               ▷ update $p$
        $d_p \leftarrow CalculateDegeneracy(p)$
return $p$
```

Table 4.4: Universal forward and reverse primers designed for all bacterial *fliC* genes. The primers are degenerate, with ambiguous bases following the IUPAC nucleotide code. Every primer listed here represents a subgroup of more than 100 targets and its coverage of said group is above 0.8. d: degeneracy; c: coverage; n:size of the subgroup. The forward and reverse sequences are not presented as pairs because most pairs are only able to represent a very small ($n < 10$) subgroup. Unconventional bases: M: A or C; R: A or G; W: A or T; S: C or G; Y: C or T; K: G or T; V: A, C or G; H: A, C or T; D: A, G or T; B: C, G or T; N: A, C, G or T.

Forward primer	d	c	n	**Forward primer**	d	c	n
`ATTAACMACAAYATYKCD`	48	0.89	199	`GTAAAYACAAAYTAHRGB`	72	0.85	40
`ATMAATCAYAAYATGRRH`	96	0.95	186	`ATYAAYWCCAAYACWTYA`	64	0.89	40

ATYAATACHAACRSYCTS	96	0.93	177	ATTAAYRCCAAYMYKGCY	128	0.85	40
ATYAATCACAAYVYKAGY	96	0.92	172	ATTWMHCACAATMTTART	48	0.9	38
ATMAAYCACAACATMAVY	48	0.91	166	ATMAACACYAAYKYGKSC	128	0.87	38
ATYAACCAGAACATSRNS	64	0.92	163	GTSMACACSAATCAGSYN	128	0.83	37
ATCAAYAMCAACSWSTCS	64	0.81	150	ATCMMGMMCAACRTBGAG	96	0.82	33
GTAAATACHAACGTBDCN	108	0.86	145	ATTAACACDAACDTMSCS	72	0.82	32
ATCAAYMACAACMTSARY	64	0.89	134	ATHAATCATAACWTRRRY	96	0.86	31
RTMAAYACCAACGTBTCD	72	0.84	131	ATYKYGACSAAYGTGKCK	128	0.89	30
ATCAATMAYAACHWDATG	72	0.86	129	ATTCTKACRAACTCNRGY	64	0.86	28
ATCAACWCCAAYRTYMAR	64	0.81	123	GTVAAYACCAATGTSRGY	48	0.92	27
GTYAAYACAAATGYNARY	128	0.89	116	ATYAACKYCAAYKCCRGC	64	0.89	26
ATCAACACSAACKTKNMM	128	1.0	84	ATCAAYAMMAAYSTGCMR	128	0.85	26
ATCCTCACSAAYSWYGCB	96	0.9	82	ATTAWBACGAACACBBCK	108	0.85	26
GTACARCAYAATTTAWMV	48	0.81	81	GTYARCACSAAYGYGTCB	96	0.92	24
ATYAAYCACAACWTBTCN	96	0.87	75	ATTCAATCHAAYRYRGCK	96	1.0	22
GTSAACACSAACWMVGGY	96	0.85	73	ATTTCRACWAACGTDKCW	48	0.85	22
ATTAATACSAAYAWHTYM	96	0.93	70	ATCCTGACGAACNCDKCN	96	0.9	21
ATTAAYCATAAYATYYMD	96	0.82	65	ATTTAYMACAACATTBMN	96	0.89	20
ATYAAYMAYAAYAYSATG	128	0.85	64	ATTGGTACGAMTRTYWYR	64	0.83	20
ATTCARCACAACMTNKCH	96	0.88	62	GTSAACACGAAYYMWGGN	128	1.0	18
ATCAACACVAACAHNTCB	108	0.84	60	CACACTAACYMCRCMHAS	96	1.0	17
ATMAAKACRAAYGTYSCS	128	0.89	57	ATCGGAACAAAYRYRKCR	64	0.88	17
GTHAAYACTAACRTKAVC	72	0.81	54	GTCAATAVYAATCHGKCD	108	1.0	16
ATYAACACAAACGTMMHN	96	1.0	52	RTCMABACAAATAYSGGR	96	1.0	15
ATWYTKACHAACAATGGY	48	1.0	52	ATYAACACAAACACDSMR	48	0.88	14
ATTCTKACRAATWYNTCS	128	1.0	51	ATAYTWACWAAYAGVTCY	96	1.0	13
ATTAACMATAATATBAVY	36	0.86	50	RTCAACCAYAACSWVATG	48	1.0	13
GCGCTTYAYRTVYTGCGY	96	1.0	46	ATTAAYACSAATMTRHTM	96	0.88	11
ATMAATCACAATCDHWTG	36	1.0	45	AKTTYTYTAAWYCAAACT	32	1.0	11
RTCAAYACCAATAYTKCN	64	0.92	43	GTRAAYACMAACRKMGGW	128	1.0	11
Reverse primer	*d*	*c*	*n*	**Reverse primer**	*d*	*c*	*n*
YAAYTGYAAHACRCYTTG	96	0.84	269	YWRAGCATAATCWRCRTC	64	0.89	39
GAGMGAVARGATRBTCTG	72	0.86	264	YARRCYNAGWACTCCCTG	128	0.86	38
RAKDGAMAGWACWCCCTG	96	0.83	255	SAGMGACAKGRCCRKCTG	64	0.92	31
NARCTKCAKSACRCCCTG	128	0.86	89	MAGWBYCAWAACACCYTG	96	1.0	26
GAGCTGSAGRAYGSYYTS	128	0.86	71	YARGCTCAKRSCRAGCTG	64	0.83	17
HAGHGATARBGCTACYTG	108	0.9	61	HARYTGTAARGCSAGYTG	96	0.86	16
YWMGGYATRATCWACGTC	64	1.0	42	RTTGCGCAGBGTGTCBGG	18	1.0	11
CAGGCKCRDCRCCGYTTG	48	0.82	40	YARYTTGAGRATHGCKTC	96	1.0	11
BACBGTRTCGTABGTSGC	108	1.0	39	TRSYTGWGAAACCATRCT	32	1.0	11

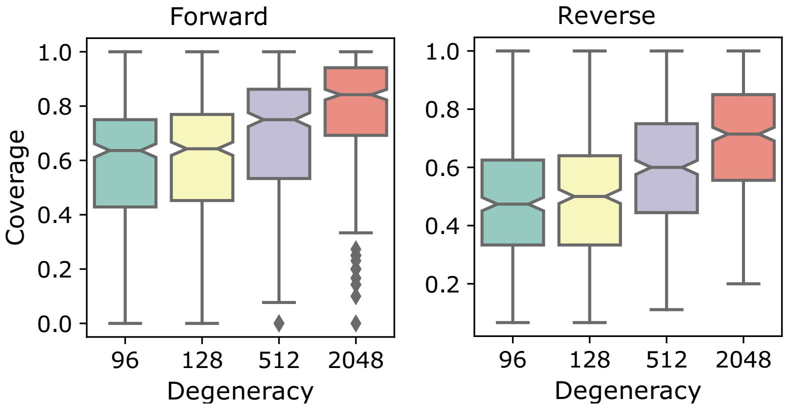


Figure 4.3: Coverages of primers at different upper limits of primer degeneracy, using the full set of bacterial flagellins as the target. Forward and reverse primers are considered separately, not as a pair. Each primer is expected to cover a cluster of similar sequences.

4.5.3 Simplifying primer design by reducing targets

The difficulties mainly come from the high sequence diversity of our targets. One practical way is to directly shrink the targets to reduce the sequence diversity. This could be considered from a few aspects. First, we could remove irrelevant targets and focus on those that could possibly occur in our environment of interest, the human gut. Second, depending on the study's aim and feature, the priorities might fall into only a handful of clades. Therefore, we turned to inspect the occurrences of all flagellins, to offer the biologists a statistical roadmap to evaluate the relevance of each flagellin to their study interest, allowing them to tailor the range of their experimental subjects.

To estimate the prevalence of the AnnoTree flagellins in the human gut, we used a dataset produced in an earlier phase of the ImMiGeNe project as a reference. The dataset contains shotgun sequencing data from 51 non-blank, non-redundant gut microbiome samples of healthy participants. We aligned each sample to the AnnoTree flagellins using DIAMOND 2.0.11 [185]. The results indicate that many of the flagellin sequences are rarely found in the metagenomic data. 5485 flagellins has no reads aligned in any of the samples. And if the infrequently occurred are to be excluded, 6770 has no more than 10 hits. The total number of sequences could be reduced to 2781 when the minimum number of reads is set to 50. We further document the phylogenomic landscape of the protein family. Figure 4.4 summarizes by the level of phylum the occurrence of the 2781 non-redundant flagellins that have at least 50 hits in all samples. Overall, it shows when the amplification target is limited to those that are likely to occur in the human gut, the target sequence diversity could be drastically reduced to a level that is much more feasible. Prioritizing the more prevalent ones could further condense the list of targets. We repeated the degenerate primer design steps in Section 4.5.2 for the aforementioned 2781 sequences. CD-HIT-est clustering was skipped, as the set here could be easily handled by affinity propagation. The forward and reverse template sequences ended up as 153 and 176 clusters, at the same *damping factor* and *preference*. We computed a primer for each cluster. On average, these primers are with degeneracy below 32 and coverage of over 0.9, when 3 mismatches or less are allowed (Figure 4.5). A detailed summary of all primers designed is listed in Supplementary Table A.9. It is clear that out of all the efforts demonstrated above, subsetting the target is by far the most effective way to tackle the overly large sequence diversity. Since their phylogenomic origin is tightly linked to sequence diversity, we also propose that one could make a shortlist by picking representatives for each taxon that occurs or is of interest.

4.6 Discussion

As a part of the ImMiGeNe project, the goal of this project is to study the diversity of microbial flagellins, which will serve as a crucial step towards understanding their immunogenicity. We devised and carried out a feasible workflow to design PCR primers for capturing a comprehensive collection of flagellins that are able to represent the diversity of the protein family. First, a non-redundant but all-encompassing set of gene sequences was compiled. The primer design was based on sequence analysis of the genes. The greatest challenge was to achieve a reasonable tradeoff between a realistic number of primers and good coverage over the target set. Through our effort to design PCR primers for bacterial *fliC* gene, we first demonstrated using AnnoTree database to obtain the genes systematically, then illustrated a series of approximation approaches for finding an optimal middle ground for the primer coverage tradeoff. Although the analysis is done on *fliC* genes only, the addressed issues are of generic nature, and the workflow could be applied to other protein-encoding gene families. To design primers for obtaining a protein family from our metagenomic microbial samples, a computational

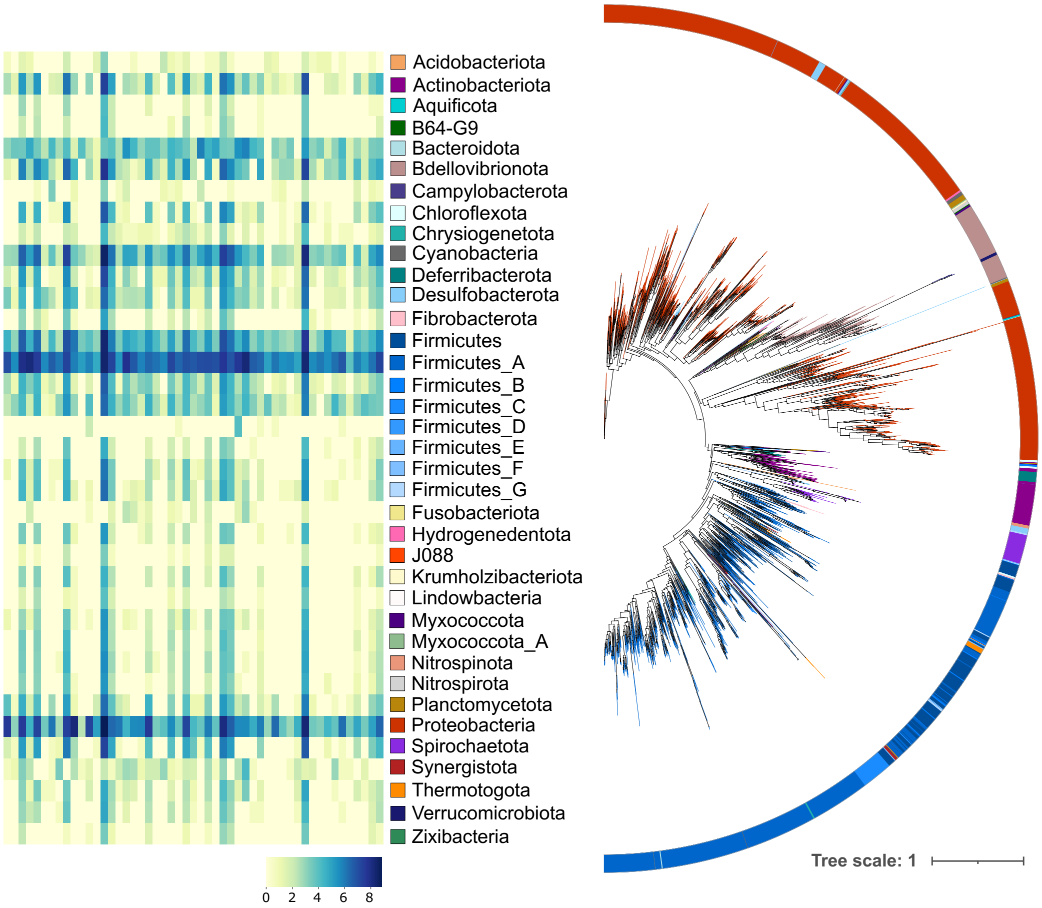


Figure 4.4: Phylogeny and frequencies of 2781 flagellins that are likely to occur in human gut. The phylogeny on the right is based on their protein sequences. Tree branches and color strip around the tree are colored according to the phylum of the source genome. Its legend is in the middle. The heatmap on the left shows the number of reads (log-transformed) the flagellins get aligned to. Each row is a sample, and each column is a phylum which aligns with the phylum name in the middle. The numbers are counted non-redundantly, that is, when a read is only counted once when aligned to multiple taxa in one phylum.

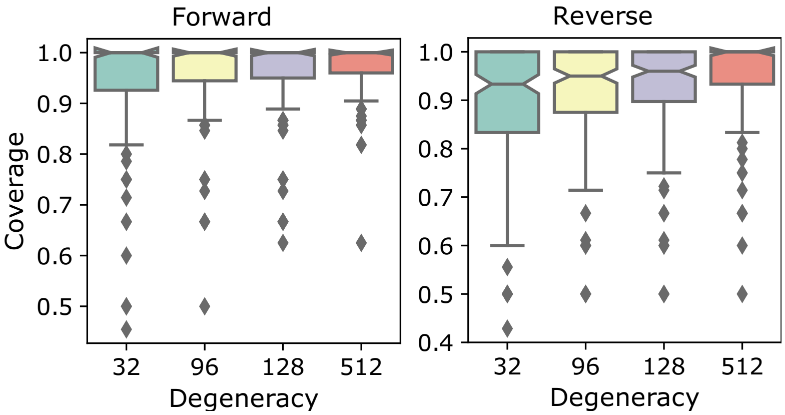


Figure 4.5: Coverages of primers at different upper limits of primer degeneracy, using as target the 2781 flagellins that are likely to occur in human gut. The mismatch allowed between the primer and the target is 3.

framework is proposed as follows:

Compile target sequences. We recommend using AnnoTree for bacterial and archaeal gene families to obtain the protein and their corresponding genomic sequences.

Make HMM alignments. This step seeks to first look for a pair of conserved domains and then use HMM search to locate them in the AnnoTree sequences. This sets the range of amplification templates. The primer length should be determined, and the primer annealing subsequences can be captured from the chosen conserved domains of the target sequences. Note that conventional multiple sequence alignments could easily lead to an MSA full of gaps in every region, which hinders the selection of possible primer sites. In comparison, picking a predetermined conservative domain and using HMM alignment can help bypass the hypervariable regions and generate nearly gap-free alignments suitable for finding the primer annealing locations.

Analyze sequence diversity and select target. This analysis focuses on evaluating the conservation and gap rate of the captured subsequences. Our example shows that a large number of sequences with overly high diversity already precludes a feasible primer set design. Therefore, we argue that in a similar scenario, it is most effective to directly cut down the number of target sequences. The target could be narrowed down to genes or proteins that are likely to appear in the samples or are from specific taxa of interest. We point out that this has more advantages over using a small database to get fewer target sequences, as our method allows more flexibility in tailoring the target based on the research interest.

Partition target sequences. The subsequences should be clustered based on their pairwise similarity (hamming distance). The number of clusters shall be consistent with the number of primers planned.

Design a degenerate primer for each cluster. In our analysis, we implemented a simple contraction algorithm (Algorithm 1). One could also adopt existing degenerate primer design programs such as HYDEN[209]. These programs are generally unable to deal with a large number of sequences, and therefore sequence partitioning is necessary.

Coverage estimation Lastly, the coverage of each primer is evaluated to estimate the overall primer coverage.

There are some concerns or limitations with our method. First, a key step in our method is sequence clustering, for which we introduced two algorithms, CD-HIT and affinity propagation. CD-HIT, just like many other sequence clustering methods, is not designed for clustering extra short sequences at a lower similarity level, like it is in our project. While general unsupervised clustering algorithms that work on distance matrix could solve the problem in theory, with more sequences, the distance matrix grows large, and the computation becomes consuming. Second, there is no quantified evaluation of the chances of off-target amplification. Our method is based on the general rule of thumb that the higher the primer degeneracy, the shorter the primers, the more the allowed mismatches, and the more off-targets the primers could potentially bring.

Primer design is an everyday problem in molecular biology labs, but extending it to a large scale of targets brings about brand new issues as feasibility becomes the top concern. In our work, we addressed the practicalities of the problem in a real-world scenario, dissected it, and defined the subproblems. We outlined our strategy and illustrated ways of approximations to get to a realistic solution. Our work highlights the complexity of large-scale primer design problems and provides a reference or guidance to primer design problems of similar nature.

Bibliography

[1] J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy, and R. M. Goodman, “Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products,” *Chemistry & biology*, vol. 5, no. 10, pp. R245–R249, 1998.

[2] F. Meyer, A. Fritz, Z.-L. Deng, D. Koslicki, T. R. Lesker, A. Gurevich, G. Robertson, M. Alser, D. Antipov, F. Beghini, *et al.*, “Critical Assessment of Metagenome Interpretation: the second round of challenges,” *Nature methods*, vol. 19, no. 4, pp. 429–440, 2022.

[3] H. M. J. R. S. Consortium, K. E. Nelson, G. M. Weinstock, S. K. Highlander, K. C. Worley, H. H. Creasy, J. R. Wortman, D. B. Rusch, M. Mitreva, E. Sodergren, *et al.*, “A catalog of reference genomes from the human microbiome,” *Science*, vol. 328, no. 5981, pp. 994–999, 2010.

[4] T. I. H. (iHMP) Research Network Consortium, “The integrative human microbiome project,” *Nature*, vol. 569, no. 7758, pp. 641–648, 2019.

[5] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon, “The human microbiome project,” *Nature*, vol. 449, no. 7164, pp. 804–810, 2007.

[6] C. R. Woese, “Bacterial evolution,” *Microbiological reviews*, vol. 51, no. 2, pp. 221–271, 1987.

[7] C. R. Woese and G. E. Fox, “Phylogenetic structure of the prokaryotic domain: the primary kingdoms,” *Proceedings of the National Academy of Sciences*, vol. 74, no. 11, pp. 5088–5090, 1977.

[8] J. C. Venter, K. Remington, J. F. Heidelberg, A. L. Halpern, D. Rusch, J. A. Eisen, D. Wu, I. Paulsen, K. E. Nelson, W. Nelson, *et al.*, "Environmental genome shotgun sequencing of the sargasso sea," *science*, 2004.

[9] G. W. Tyson, J. Chapman, P. Hugenholtz, E. E. Allen, R. J. Ram, P. M. Richardson, V. V. Solovyev, E. M. Rubin, D. S. Rokhsar, and J. F. Banfield, "Community structure and metabolism through reconstruction of microbial genomes from the environment," *Nature*, vol. 428, no. 6978, pp. 37–43, 2004.

[10] A. M. Giani, G. R. Gallo, L. Gianfranceschi, and G. Formenti, "Long walk to genomics: History and current approaches to genome sequencing and assembly," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 9–19, 2020.

[11] H. Xie, C. Yang, Y. Sun, Y. Igarashi, T. Jin, and F. Luo, "Pacbio long reads improve metagenomic assemblies, gene catalogs, and genome binning," *Frontiers in Genetics*, vol. 11, p. 516269, 2020.

[12] S. M. Nicholls, J. C. Quick, S. Tang, and N. J. Loman, "Ultra-deep, long-read nanopore sequencing of mock microbial community standards," *Gigascience*, vol. 8, no. 5, p. giz043, 2019.

[13] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, *et al.*, "The complete sequence of a human genome," *Science*, vol. 376, no. 6588, pp. 44–53, 2022.

[14] M. J. Levene, J. Korlach, S. W. Turner, M. Foquet, H. G. Craighead, and W. W. Webb, "Zero-mode waveguides for single-molecule analysis at high concentrations," *science*, vol. 299, no. 5607, pp. 682–686, 2003.

[15] S. Huang, J. He, S. Chang, P. Zhang, F. Liang, S. Li, M. Tuchband, A. Fuhrmann, R. Ros, and S. Lindsay, "Identifying single bases in a dna oligomer with electron tunnelling," *Nature nanotechnology*, vol. 5, no. 12, pp. 868–873, 2010.

[16] Y. Wang, Y. Zhao, A. Bollas, Y. Wang, and K. F. Au, "Nanopore sequencing technology, bioinformatics and applications," *Nature biotechnology*, vol. 39, no. 11, pp. 1348–1365, 2021.

[17] M. Sereika, R. H. Kirkegaard, S. M. Karst, T. Y. Michaelsen, E. A. Sørensen, R. D. Wollenberg, and M. Albertsen, "Oxford nanopore r10. 4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing," *Nature methods*, vol. 19, no. 7, pp. 823–826, 2022.

[18] D. M. Bickhart, M. Kolmogorov, E. Tseng, D. M. Portik, A. Korobeynikov, I. Tolstoganov, G. Uritskiy, I. Liachko, S. T. Sullivan, S. B. Shin, *et al.*, "Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities," *Nature biotechnology*, vol. 40, no. 5, pp. 711–719, 2022.

[19] M. Z. DeMaere and A. E. Darling, "bin3C: exploiting hi-c sequencing data to accurately resolve metagenome-assembled genomes," *Genome biology*, vol. 20, no. 1, pp. 1–16, 2019.

[20] A. Bishara, E. L. Moss, M. Kolmogorov, A. E. Parada, Z. Weng, A. Sidow, A. E. Dekas, S. Batzoglou, and A. S. Bhatt, "High-quality genome sequences of uncultured microbes by assembly of read clouds," *Nature biotechnology*, vol. 36, no. 11, pp. 1067–1075, 2018.

[21] A. Prjibelski, D. Antipov, D. Meleshko, A. Lapidus, and A. Korobeynikov, "Using spades de novo assembler," *Current protocols in bioinformatics*, vol. 70, no. 1, p. e102, 2020.

[22] D. Li, C.-M. Liu, R. Luo, K. Sadakane, and T.-W. Lam, "Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph," *Bioinformatics*, vol. 31, no. 10, pp. 1674–1676, 2015.

[23] M. Kolmogorov, D. M. Bickhart, B. Behsaz, A. Gurevich, M. Rayko, S. B. Shin, K. Kuhn, J. Yuan, E. Polevikov, T. P. Smith, *et al.*, "metaFlye: scalable long-read metagenome assembly using repeat graphs," *Nature Methods*, vol. 17, no. 11, pp. 1103–1110, 2020.

[24] X. Feng, H. Cheng, D. Portik, and H. Li, "Metagenome assembly of high-fidelity long reads with hifiasm-meta," *Nature Methods*, vol. 19, no. 6, pp. 671–674, 2022.

[25] S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman, and A. M. Phillippy, "Canu: scalable and accurate long-read assembly via

adaptive k-mer weighting and repeat separation," *Genome research*, vol. 27, no. 5, pp. 722–736, 2017.

[26] S. Nurk, B. P. Walenz, A. Rhie, M. R. Vollger, G. A. Logsdon, R. Grothe, K. H. Miga, E. E. Eichler, A. M. Phillippy, and S. Koren, "HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads," *Genome research*, vol. 30, no. 9, pp. 1291–1305, 2020.

[27] K. Shafin, T. Pesout, R. Lorig-Roach, M. Haukness, H. E. Olsen, C. Bosworth, J. Armstrong, K. Tigyi, N. Maurer, S. Koren, *et al.*, "Nanopore sequencing and the shasta toolkit enable efficient de novo assembly of eleven human genomes," *Nature biotechnology*, vol. 38, no. 9, pp. 1044–1053, 2020.

[28] R. Vaser, I. Sović, N. Nagarajan, and M. Šikić, "Fast and accurate de novo genome assembly from long uncorrected reads," *Genome research*, vol. 27, no. 5, pp. 737–746, 2017.

[29] B. J. Walker, T. Abeel, T. Shea, M. Priest, A. Abouelliel, S. Sakthikumar, C. A. Cuomo, Q. Zeng, J. Wortman, S. K. Young, *et al.*, "Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement," *PloS one*, vol. 9, no. 11, p. e112963, 2014.

[30] K. Arumugam, C. Bağcı, I. Bessarab, S. Beier, B. Buchfink, A. Górska, G. Qiu, D. H. Huson, and R. B. Williams, "Annotated bacterial chromosomes from frame-shift-corrected long-read metagenomic data," *Microbiome*, vol. 7, no. 1, pp. 1–13, 2019.

[31] E. L. Moss, D. G. Maghini, and A. S. Bhatt, "Complete, closed bacterial genomes from microbiomes using nanopore sequencing," *Nature biotechnology*, vol. 38, no. 6, pp. 701–707, 2020.

[32] D. H. Parks, C. Rinke, M. Chuvochina, P.-A. Chaumeil, B. J. Woodcroft, P. N. Evans, P. Hugenholtz, and G. W. Tyson, "Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life," *Nature microbiology*, vol. 2, no. 11, pp. 1533–1542, 2017.

[33] L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, *et al.*,

"A communal catalogue reveals Earth's multiscale microbial diversity," *Nature*, vol. 551, no. 7681, pp. 457–463, 2017.

[34] A. Almeida, A. L. Mitchell, M. Boland, S. C. Forster, G. B. Gloor, A. Tarkowska, T. D. Lawley, and R. D. Finn, "A new genomic blueprint of the human gut microbiota," *Nature*, vol. 568, no. 7753, pp. 499–504, 2019.

[35] E. Pasolli, F. Asnicar, S. Manara, M. Zolfo, N. Karcher, F. Armanini, F. Beghini, P. Manghi, A. Tett, P. Ghensi, *et al.*, "Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle," *Cell*, vol. 176, no. 3, pp. 649–662, 2019.

[36] S. Nayfach, S. Roux, R. Seshadri, D. Udwary, N. Varghese, F. Schulz, D. Wu, D. Paez-Espino, I.-M. Chen, M. Huntemann, *et al.*, "A genomic catalog of earth's microbiomes," *Nature biotechnology*, vol. 39, no. 4, pp. 499–509, 2021.

[37] J. Lloyd-Price, C. Arze, A. N. Ananthakrishnan, M. Schirmer, J. Avila-Pacheco, T. W. Poon, E. Andrews, N. J. Ajami, K. S. Bonham, C. J. Brislawn, *et al.*, "Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases," *Nature*, vol. 569, no. 7758, pp. 655–662, 2019.

[38] E. V. Koonin, K. S. Makarova, and Y. I. Wolf, "Evolution of microbial genomics: conceptual shifts over a quarter century," *Trends in microbiology*, vol. 29, no. 7, pp. 582–592, 2021.

[39] D. Hyatt, G.-L. Chen, P. F. LoCascio, M. L. Land, F. W. Larimer, and L. J. Hauser, "Prodigal: prokaryotic gene recognition and translation initiation site identification," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–11, 2010.

[40] J. Besemer, A. Lomsadze, and M. Borodovsky, "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions," *Nucleic acids research*, vol. 29, no. 12, pp. 2607–2618, 2001.

[41] R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, no. 5338, pp. 631–637, 1997.

[42] L. Aravind, "Guilt by association: contextual information in genome analysis," *Genome Research*, vol. 10, no. 8, pp. 1074–1077, 2000.

[43] C. J. Castelle and J. F. Banfield, "Major new microbial groups expand diversity and alter our understanding of the tree of life," *Cell*, vol. 172, no. 6, pp. 1181–1197, 2018.

[44] E. A. Franzosa, L. J. McIver, G. Rahnavard, L. R. Thompson, M. Schirmer, G. Weingart, K. S. Lipson, R. Knight, J. G. Caporaso, N. Segata, *et al.*, "Species-level functional profiling of metagenomes and metatranscriptomes," *Nature methods*, vol. 15, no. 11, pp. 962–968, 2018.

[45] D. H. Huson, S. Beier, I. Flade, A. Górska, M. El-Hadidi, S. Mitra, H.-J. Ruscheweyh, and R. Tappu, "Megan community edition-interactive exploration and analysis of large-scale microbiome sequencing data," *PLoS computational biology*, vol. 12, no. 6, p. e1004957, 2016.

[46] E. R. Hughes, M. G. Winter, B. A. Duerkop, L. Spiga, T. F. de Carvalho, W. Zhu, C. C. Gillis, L. Büttner, M. P. Smoot, C. L. Behrendt, *et al.*, "Microbial respiration and formate oxidation as metabolic signatures of inflammation-associated dysbiosis," *Cell host & microbe*, vol. 21, no. 2, pp. 208–219, 2017.

[47] T. Rodrigues-Oliveira, F. Wollweber, R. I. Ponce-Toledo, J. Xu, S. K.-M. Rittmann, A. Klingl, M. Pilhofer, and C. Schleper, "Actin cytoskeleton and complex cell architecture in an asgard archaeon," *Nature*, pp. 1–8, 2022.

[48] A. J. Probst, B. Ladd, J. K. Jarett, D. E. Geller-McGrath, C. M. Sieber, J. B. Emerson, K. Anantharaman, B. C. Thomas, R. R. Malmstrom, M. Stieglmeier, *et al.*, "Differential depth distribution of microbial function and putative symbionts through sediment-hosted aquifers in the deep terrestrial subsurface," *Nature microbiology*, vol. 3, no. 3, pp. 328–336, 2018.

[49] C. He, R. Keren, M. L. Whittaker, I. F. Farag, J. A. Doudna, J. H. Cate, and J. F. Banfield, "Genome-resolved metagenomics reveals site-specific diversity of episymbiotic CPR bacteria and DPANN archaea in

groundwater ecosystems," *Nature microbiology*, vol. 6, no. 3, pp. 354–365, 2021.

[50] S. L. Jørgensen, I. H. Thorseth, R. B. Pedersen, T. Baumberger, and C. Schleper, "Quantitative and phylogenetic study of the deep sea archaeal group in sediments of the arctic mid-ocean spreading ridge," *Frontiers in microbiology*, vol. 4, p. 299, 2013.

[51] A. Spang, J. H. Saw, S. L. Jørgensen, K. Zaremba-Niedzwiedzka, J. Martijn, A. E. Lind, R. Van Eijk, C. Schleper, L. Guy, and T. J. Ettema, "Complex archaea that bridge the gap between prokaryotes and eukaryotes," *Nature*, vol. 521, no. 7551, pp. 173–179, 2015.

[52] K. Zaremba-Niedzwiedzka, E. F. Caceres, J. H. Saw, D. Bäckström, L. Juzokaite, E. Vancaester, K. W. Seitz, K. Anantharaman, P. Starnawski, K. U. Kjeldsen, *et al.*, "Asgard archaea illuminate the origin of eukaryotic cellular complexity," *Nature*, vol. 541, no. 7637, pp. 353–358, 2017.

[53] P. Hugenholtz, C. Pitulle, K. L. Hershberger, and N. R. Pace, "Novel division level bacterial diversity in a yellowstone hot spring," *Journal of bacteriology*, vol. 180, no. 2, pp. 366–376, 1998.

[54] C. Rinke, P. Schwientek, A. Sczyrba, N. N. Ivanova, I. J. Anderson, J.-F. Cheng, A. Darling, S. Malfatti, B. K. Swan, E. A. Gies, *et al.*, "Insights into the phylogeny and coding potential of microbial dark matter," *Nature*, vol. 499, no. 7459, pp. 431–437, 2013.

[55] C. T. Brown, L. A. Hug, B. C. Thomas, I. Sharon, C. J. Castelle, A. Singh, M. J. Wilkins, K. C. Wrighton, K. H. Williams, and J. F. Banfield, "Unusual biology across a group comprising more than 15% of domain bacteria," *Nature*, vol. 523, no. 7559, pp. 208–211, 2015.

[56] C. J. Castelle, R. Méheust, A. L. Jaffe, K. Seitz, X. Gong, B. J. Baker, and J. F. Banfield, "Protein family content uncovers lineage relationships and bacterial pathway maintenance mechanisms in DPANN archaea," *Frontiers in Microbiology*, vol. 12, p. 660052, 2021.

[57] D. H. Parks, M. Chuvochina, D. W. Waite, C. Rinke, A. Skarshewski, P.-A. Chaumeil, and P. Hugenholtz, "A standardized bacterial taxon-

omy based on genome phylogeny substantially revises the tree of life," *Nature biotechnology*, vol. 36, no. 10, pp. 996–1004, 2018.

[58] C. L. Schoch, S. Ciufo, M. Domrachev, C. L. Hotton, S. Kannan, R. Khovanskaya, D. Leipe, R. Mcveigh, K. O'Neill, B. Robbertse, *et al.*, "NCBI taxonomy: a comprehensive update on curation, resources and tools," *Database*, vol. 2020, 2020.

[59] D. H. Parks, M. Chuvochina, C. Rinke, A. J. Mussig, P.-A. Chaumeil, and P. Hugenholtz, "GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy," *Nucleic acids research*, vol. 50, no. D1, pp. D785–D794, 2022.

[60] D. H. Parks, M. Chuvochina, P.-A. Chaumeil, C. Rinke, A. J. Mussig, and P. Hugenholtz, "A complete domain-to-species taxonomy for Bacteria and Archaea," *Nature biotechnology*, vol. 38, no. 9, pp. 1079–1086, 2020.

[61] C. Rinke, M. Chuvochina, A. J. Mussig, P.-A. Chaumeil, A. A. Davín, D. W. Waite, W. B. Whitman, D. H. Parks, and P. Hugenholtz, "A standardized archaeal taxonomy for the Genome Taxonomy Database," *Nature Microbiology*, vol. 6, no. 7, pp. 946–959, 2021.

[62] D. Tamarit, E. F. Caceres, M. Krupovic, R. Nijland, L. Eme, N. P. Robinson, and T. J. Ettema, "A closed candidatus odinarchaeum chromosome exposes Asgard archaeal viruses," *Nature microbiology*, vol. 7, no. 7, pp. 948–952, 2022.

[63] K. G. Lloyd and G. Tahon, "Science depends on nomenclature, but nomenclature is not science," *Nature Reviews Microbiology*, vol. 20, no. 3, pp. 123–124, 2022.

[64] R. Vicedomini, C. Quince, A. E. Darling, and R. Chikhi, "Strainberry: automated strain separation in low-complexity metagenomes using long reads," *Nature Communications*, vol. 12, no. 1, p. 4485, 2021.

[65] F. Beghini, L. J. McIver, A. Blanco-Míguez, L. Dubois, F. Asnicar, S. Maharjan, A. Mailyan, P. Manghi, M. Scholz, A. M. Thomas, *et al.*, "Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with biobakery 3," *elife*, vol. 10, p. e65088, 2021.

[66] X. Kang, X. Luo, and A. Schönhuth, "Strainxpress: strain aware metagenome assembly from short reads," *Nucleic Acids Research*, vol. 50, no. 17, pp. e101–e101, 2022.

[67] H. Tettelin, V. Masignani, M. J. Cieslewicz, C. Donati, D. Medini, N. L. Ward, S. V. Angiuoli, J. Crabtree, A. L. Jones, A. S. Durkin, *et al.*, "Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: implications for the microbial "pan-genome"," *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, pp. 13950–13955, 2005.

[68] R. M. Sherman and S. L. Salzberg, "Pan-genomics in the human genome era," *Nature Reviews Genetics*, vol. 21, no. 4, pp. 243–254, 2020.

[69] A. K. Pöntinen, J. Top, S. Arredondo-Alonso, G. Tonkin-Hill, A. R. Freitas, C. Novais, R. A. Gladstone, M. Pesonen, R. Meneses, H. Pesonen, *et al.*, "Apparent nosocomial adaptation of enterococcus faecalis predates the modern hospital era," *Nature communications*, vol. 12, no. 1, p. 1523, 2021.

[70] T. Li and Y. Yin, "Critical assessment of pan-genomic analysis of metagenome-assembled genomes," *Briefings in Bioinformatics*, vol. 23, no. 6, p. bbac413, 2022.

[71] B. T. Tierney, Z. Yang, J. M. Luber, M. Beaudin, M. C. Wibowo, C. Baek, E. Mehlenbacher, C. J. Patel, and A. D. Kostic, "The landscape of genetic content in the gut and oral human microbiome," *Cell host & microbe*, vol. 26, no. 2, pp. 283–295, 2019.

[72] O. Mineeva, M. Rojas-Carulla, R. E. Ley, B. Schölkopf, and N. D. Youngblut, "DeepMAsED: evaluating the quality of metagenomic assemblies," *Bioinformatics*, vol. 36, no. 10, pp. 3011–3017, 2020.

[73] L.-X. Chen, K. Anantharaman, A. Shaiber, A. M. Eren, and J. F. Banfield, "Accurate and complete genomes from metagenomes," *Genome Research*, vol. 30, no. 3, pp. 315–333, 2020.

[74] Y. Yue, H. Huang, Z. Qi, H.-M. Dou, X.-Y. Liu, T.-F. Han, Y. Chen, X.-J. Song, Y.-H. Zhang, and J. Tu, "Evaluating metagenomics

tools for genome binning with real metagenomic datasets and CAMI datasets," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–15, 2020.

[75] S. Karlin, J. Mrazek, and A. M. Campbell, "Compositional biases of bacterial genomes and evolutionary implications," *Journal of bacteriology*, vol. 179, no. 12, pp. 3899–3913, 1997.

[76] G. J. Dick, A. F. Andersson, B. J. Baker, S. L. Simmons, B. C. Thomas, A. P. Yelton, and J. F. Banfield, "Community-wide analysis of microbial genome sequence signatures," *Genome biology*, vol. 10, no. 8, pp. 1–16, 2009.

[77] E. S. Lander and M. S. Waterman, "Genomic mapping by fingerprinting random clones: a mathematical analysis," *Genomics*, vol. 2, no. 3, pp. 231–239, 1988.

[78] S. N. Evans, V. Hower, and L. Pachter, "Coverage statistics for sequence census methods," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–10, 2010.

[79] M. Albertsen, P. Hugenholtz, A. Skarshewski, K. L. Nielsen, G. W. Tyson, and P. H. Nielsen, "Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes," *Nature biotechnology*, vol. 31, no. 6, pp. 533–538, 2013.

[80] H. B. Nielsen, M. Almeida, A. S. Juncker, S. Rasmussen, J. Li, S. Sunagawa, D. R. Plichta, L. Gautier, A. G. Pedersen, E. Le Chatelier, *et al.*, "Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes," *Nature biotechnology*, vol. 32, no. 8, pp. 822–828, 2014.

[81] V. Iverson, R. M. Morris, C. D. Frazar, C. T. Berthiaume, R. L. Morales, and E. V. Armbrust, "Untangling genomes from metagenomes: revealing an uncultured class of marine Euryarchaeota," *Science*, vol. 335, no. 6068, pp. 587–590, 2012.

[82] J. Alneberg, B. S. Bjarnason, I. De Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson, and C. Quince, "Binning metagenomic contigs by coverage and composition," *Nature methods*, vol. 11, no. 11, pp. 1144–1146, 2014.

[83] D. D. Kang, F. Li, E. Kirton, A. Thomas, R. Egan, H. An, and Z. Wang, "MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies," *PeerJ*, vol. 7, p. e7359, 2019.

[84] Y.-W. Wu, B. A. Simmons, and S. W. Singer, "MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets," *Bioinformatics*, vol. 32, no. 4, pp. 605–607, 2016.

[85] M. Imelfort, D. Parks, B. J. Woodcroft, P. Dennis, P. Hugenholtz, and G. W. Tyson, "GroopM: an automated tool for the recovery of population genomes from related metagenomes," *PeerJ*, vol. 2, p. e603, 2014.

[86] Y. Y. Lu, T. Chen, J. A. Fuhrman, and F. Sun, "COCACOLA: binning metagenomic contigs using sequence composition, read coverage, coalignment and paired-end read linkage," *Bioinformatics*, vol. 33, no. 6, pp. 791–798, 2017.

[87] H. S. Muralidharan, N. Shah, J. S. Meisel, and M. Pop, "Binnacle: Using scaffolds to improve the contiguity and quality of metagenomic bins," *Frontiers in microbiology*, vol. 12, p. 638561, 2021.

[88] V. Mallawaarachchi, A. Wickramarachchi, and Y. Lin, "GraphBin: refined binning of metagenomic contigs using assembly graphs," *Bioinformatics*, vol. 36, no. 11, pp. 3307–3313, 2020.

[89] A. Lamurias, M. Sereika, M. Albertsen, K. Hose, and T. D. Nielsen, "Metagenomic binning with assembly graph embeddings," *Bioinformatics*, vol. 38, pp. 4481–4487, 08 2022.

[90] M. O. Press, A. H. Wiser, Z. N. Kronenberg, K. W. Langford, M. Shakya, C.-C. Lo, K. A. Mueller, S. T. Sullivan, P. S. Chain, and I. Liachko, "Hi-C deconvolution of a human gut microbiome yields high-quality draft genomes and reveals plasmid-genome interactions," *biorxiv*, p. 198713, 2017.

[91] Y. Du and F. Sun, "HiCBin: Binning metagenomic contigs and recovering metagenome-assembled genomes using hi-c contact maps," *Genome biology*, vol. 23, no. 1, pp. 1–21, 2022.

[92] Y. Du and F. Sun, "HiFine: integrating hi-c-based and shotgun-based methods to refine binning of metagenomic contigs," *Bioinformatics*, vol. 38, no. 11, pp. 2973–2979, 2022.

[93] E. Lieberman-Aiden, N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *science*, vol. 326, no. 5950, pp. 289–293, 2009.

[94] J. Han, Z. Zhang, and K. Wang, "3C and 3C-based techniques: the powerful tools for spatial genome organization deciphering," *Molecular Cytogenetics*, vol. 11, no. 1, pp. 1–10, 2018.

[95] Y.-W. Wu, Y.-H. Tang, S. G. Tringe, B. A. Simmons, and S. W. Singer, "MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm," *Microbiome*, vol. 2, no. 1, pp. 1–18, 2014.

[96] D. D. Kang, J. Froula, R. Egan, and Z. Wang, "MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities," *PeerJ*, vol. 3, p. e1165, 2015.

[97] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[98] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[99] J. N. Nissen, J. Johansen, R. L. Allesøe, C. K. Sønderby, J. J. A. Armenteros, C. H. Grønbech, L. J. Jensen, H. B. Nielsen, T. N. Petersen, O. Winther, *et al.*, "Improved metagenome binning and assembly using deep variational autoencoders," *Nature biotechnology*, vol. 39, no. 5, pp. 555–560, 2021.

[100] P. Zhang, Z. Jiang, Y. Wang, and Y. Li, "CLMB: deep contrastive learning for robust metagenomic binning," in *International Conference on Research in Computational Molecular Biology*, pp. 326–348, Springer, 2022.

[101] D. H. Parks, M. Imelfort, C. T. Skennerton, P. Hugenholtz, and G. W. Tyson, "CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes," *Genome research*, vol. 25, no. 7, pp. 1043–1055, 2015.

[102] R. M. Bowers, N. C. Kyrpides, R. Stepanauskas, M. Harmon-Smith, D. Doud, T. Reddy, F. Schulz, J. Jarett, A. R. Rivers, E. A. Eloe-Fadrosh, *et al.*, "Minimum information about a single amplified genome (misag) and a metagenome-assembled genome (mimag) of bacteria and archaea," *Nature biotechnology*, vol. 35, no. 8, pp. 725–731, 2017.

[103] C. Yuan, J. Lei, J. Cole, and Y. Sun, "Reconstructing 16s rrna genes in metagenomic data," *Bioinformatics*, vol. 31, no. 12, pp. i35–i43, 2015.

[104] A. B. Shreiner, J. Y. Kao, and V. B. Young, "The gut microbiome in health and in disease," *Current opinion in gastroenterology*, vol. 31, no. 1, p. 69, 2015.

[105] R. Sender, S. Fuchs, and R. Milo, "Revised estimates for the number of human and bacteria cells in the body," *PLoS biology*, vol. 14, no. 8, p. e1002533, 2016.

[106] E. R. Davenport, J. G. Sanders, S. J. Song, K. R. Amato, A. G. Clark, and R. Knight, "The human microbiome in evolution," *BMC biology*, vol. 15, no. 1, pp. 1–12, 2017.

[107] Q. Tang, G. Jin, G. Wang, T. Liu, X. Liu, B. Wang, and H. Cao, "Current sampling methods for gut microbiota: a call for more precise devices," *Frontiers in cellular and infection microbiology*, p. 151, 2020.

[108] D. Stanley, M. S. Geier, H. Chen, R. J. Hughes, and R. J. Moore, "Comparison of fecal and cecal microbiotas reveals qualitative similarities but quantitative differences," *BMC microbiology*, vol. 15, no. 1, pp. 1–11, 2015.

[109] M. A. Mahowald, F. E. Rey, H. Seedorf, P. J. Turnbaugh, R. S. Fulton, A. Wollam, N. Shah, C. Wang, V. Magrini, R. K. Wilson, *et al.*, "Characterizing a model human gut microbiota composed of members of its two dominant bacterial phyla," *Proceedings of the National Academy of Sciences*, vol. 106, no. 14, pp. 5859–5864, 2009.

[110] C. A. Lozupone, J. I. Stombaugh, J. I. Gordon, J. K. Jansson, and R. Knight, "Diversity, stability and resilience of the human gut microbiota," *Nature*, vol. 489, no. 7415, pp. 220–230, 2012.

[111] J. Lloyd-Price, G. Abu-Ali, and C. Huttenhower, "The healthy human microbiome," *Genome medicine*, vol. 8, no. 1, pp. 1–11, 2016.

[112] T. S. Schmidt, J. Raes, and P. Bork, "The human gut microbiome: from association to modulation," *Cell*, vol. 172, no. 6, pp. 1198–1215, 2018.

[113] L. V. Blanton, M. J. Barratt, M. R. Charbonneau, T. Ahmed, and J. I. Gordon, "Childhood undernutrition, the gut microbiota, and microbiota-directed therapeutics," *Science*, vol. 352, no. 6293, pp. 1533–1533, 2016.

[114] M. J. FitzGerald and E. J. Spek, "Microbiome therapeutics and patent protection," *Nature Biotechnology*, vol. 38, no. 7, pp. 806–810, 2020.

[115] S. P. Spencer, G. K. Fragiadakis, and J. L. Sonnenburg, "Pursuing human-relevant gut microbiota-immune interactions," *Immunity*, vol. 51, no. 2, pp. 225–239, 2019.

[116] J. Halfvarson, C. J. Brislawn, R. Lamendella, Y. Vázquez-Baeza, W. A. Walters, L. M. Bramer, M. D'amato, F. Bonfiglio, D. McDonald, A. Gonzalez, *et al.*, "Dynamics of the human gut microbiome in inflammatory bowel disease," *Nature microbiology*, vol. 2, no. 5, pp. 1–7, 2017.

[117] P. J. Turnbaugh, R. E. Ley, M. A. Mahowald, V. Magrini, E. R. Mardis, and J. I. Gordon, "An obesity-associated gut microbiome with increased capacity for energy harvest," *nature*, vol. 444, no. 7122, pp. 1027–1031, 2006.

[118] R. E. Ley, P. J. Turnbaugh, S. Klein, and J. I. Gordon, "Human gut microbes associated with obesity," *nature*, vol. 444, no. 7122, pp. 1022–1023, 2006.

[119] H.-J. Wu, I. I. Ivanov, J. Darce, K. Hattori, T. Shima, Y. Umesaki, D. R. Littman, C. Benoist, and D. Mathis, "Gut-residing segmented

filamentous bacteria drive autoimmune arthritis via t helper 17 cells," *Immunity*, vol. 32, no. 6, pp. 815–827, 2010.

[120] S. Manfredo Vieira, M. Hiltensperger, V. Kumar, D. Zegarra-Ruiz, C. Dehner, N. Khan, F. Costa, E. Tiniakou, T. Greiling, W. Ruff, *et al.*, "Translocation of a gut pathobiont drives autoimmunity in mice and humans," *Science*, vol. 359, no. 6380, pp. 1156–1161, 2018.

[121] G. D. Sepich-Poore, L. Zitvogel, R. Straussman, J. Hasty, J. A. Wargo, and R. Knight, "The microbiome and human cancer," *Science*, vol. 371, no. 6536, p. eabc4552, 2021.

[122] M. Valles-Colomer, G. Falony, Y. Darzi, E. F. Tigchelaar, J. Wang, R. Y. Tito, C. Schiweck, A. Kurilshikov, M. Joossens, C. Wijmenga, *et al.*, "The neuroactive potential of the human gut microbiota in quality of life and depression," *Nature microbiology*, vol. 4, no. 4, pp. 623–632, 2019.

[123] M. X. Byndloss, E. E. Olsan, F. Rivera-Chávez, C. R. Tiffany, S. A. Cevallos, K. L. Lokken, T. P. Torres, A. J. Byndloss, F. Faber, Y. Gao, *et al.*, "Microbiota-activated PPAR-γ signaling inhibits dysbiotic enterobacteriaceae expansion," *Science*, vol. 357, no. 6351, pp. 570–575, 2017.

[124] A. I. Lim, T. McFadden, V. M. Link, S.-J. Han, R.-M. Karlsson, A. Stacy, T. K. Farley, D. S. Lima-Junior, O. J. Harrison, J. V. Desai, *et al.*, "Prenatal maternal infection promotes tissue-specific immunity and inflammation in offspring," *Science*, vol. 373, no. 6558, p. eabf3002, 2021.

[125] E. Vivier and B. Malissen, "Innate and adaptive immunity: specificities and signaling hierarchies revisited," *Nature immunology*, vol. 6, no. 1, pp. 17–21, 2005.

[126] J. S. Marshall, R. Warrington, W. Watson, and H. L. Kim, "An introduction to immunology and immunopathology," *Allergy, Asthma & Clinical Immunology*, vol. 14, no. 2, pp. 1–10, 2018.

[127] S. Akira, S. Uematsu, and O. Takeuchi, "Pathogen recognition and innate immunity," *Cell*, vol. 124, no. 4, pp. 783–801, 2006.

[128] K. J. Ishii, S. Koyama, A. Nakagawa, C. Coban, and S. Akira, “Host innate immune receptors and beyond: making sense of microbial infections,” *Cell host & microbe*, vol. 3, no. 6, pp. 352–363, 2008.

[129] M. D. Cooper and M. N. Alder, “The evolution of adaptive immune systems,” *Cell*, vol. 124, no. 4, pp. 815–822, 2006.

[130] J. J. Calis and B. R. Rosenberg, “Characterizing immune repertoires by high throughput sequencing: strategies and applications,” *Trends in immunology*, vol. 35, no. 12, pp. 581–590, 2014.

[131] C. Martino, A. H. Dilmore, Z. M. Burcham, J. L. Metcalf, D. Jeste, and R. Knight, “Microbiota succession throughout life from the cradle to the grave,” *Nature Reviews Microbiology*, pp. 1–14, 2022.

[132] L. Spiga and S. E. Winter, “Using enteric pathogens to probe the gut microbiota,” *Trends in microbiology*, vol. 27, no. 3, pp. 243–253, 2019.

[133] S. Rakoff-Nahoum, J. Paglino, F. Eslami-Varzaneh, S. Edberg, and R. Medzhitov, “Recognition of commensal microflora by toll-like receptors is required for intestinal homeostasis,” *Cell*, vol. 118, no. 2, pp. 229–241, 2004.

[134] C. O’Mahony, P. Scully, D. O’Mahony, S. Murphy, F. O’Brien, A. Lyons, G. Sherlock, J. MacSharry, B. Kiely, F. Shanahan, *et al.*, “Commensal-induced regulatory t cells mediate protection against pathogen-stimulated nf-κb activation,” *PLoS pathogens*, vol. 4, no. 8, p. e1000112, 2008.

[135] S. K. Mazmanian, J. L. Round, and D. L. Kasper, “A microbial symbiosis factor prevents intestinal inflammatory disease,” *Nature*, vol. 453, no. 7195, pp. 620–625, 2008.

[136] S. K. Mazmanian, J. L. Round, and D. L. Kasper, “A microbial symbiosis factor prevents intestinal inflammatory disease,” *Nature*, vol. 453, no. 7195, pp. 620–625, 2008.

[137] I. I. Ivanov, K. Atarashi, N. Manel, E. L. Brodie, T. Shima, U. Karaoz, D. Wei, K. C. Goldfarb, C. A. Santee, S. V. Lynch, *et al.*, “Induction of intestinal th17 cells by segmented filamentous bacteria,” *Cell*, vol. 139, no. 3, pp. 485–498, 2009.

[138] P. M. Smith, M. R. Howitt, N. Panikov, M. Michaud, C. A. Gallini, M. Bohlooly-y, J. N. Glickman, and W. S. Garrett, "The microbial metabolites, short-chain fatty acids, regulate colonic treg cell homeostasis," *Science*, vol. 341, no. 6145, pp. 569–573, 2013.

[139] P. V. Chang, L. Hao, S. Offermanns, and R. Medzhitov, "The microbial metabolite butyrate regulates intestinal macrophage function via histone deacetylase inhibition," *Proceedings of the National Academy of Sciences*, vol. 111, no. 6, pp. 2247–2252, 2014.

[140] L. Wampach, A. Heintz-Buschart, J. V. Fritz, J. Ramiro-Garcia, J. Habier, M. Herold, S. Narayanasamy, A. Kaysen, A. H. Hogan, L. Bindl, *et al.*, "Birth mode is associated with earliest strain-conferred gut microbiome functions and immunostimulatory potential," *Nature communications*, vol. 9, no. 1, p. 5091, 2018.

[141] G. Den Besten, K. Van Eunen, A. K. Groen, K. Venema, D.-J. Reijngoud, and B. M. Bakker, "The role of short-chain fatty acids in the interplay between diet, gut microbiota, and host energy metabolism," *Journal of lipid research*, vol. 54, no. 9, pp. 2325–2340, 2013.

[142] M. A. Fischbach and J. L. Sonnenburg, "Eating for two: how metabolism establishes interspecies interactions in the gut," *Cell host & microbe*, vol. 10, no. 4, pp. 336–347, 2011.

[143] S. E. Winter, C. A. Lopez, and A. J. Bäumler, "The dynamics of gut-associated microbial communities during inflammation," *EMBO reports*, vol. 14, no. 4, pp. 319–327, 2013.

[144] L. Spiga, M. G. Winter, T. F. de Carvalho, W. Zhu, E. R. Hughes, C. C. Gillis, C. L. Behrendt, J. Kim, D. Chessa, H. L. Andrews-Polymenis, *et al.*, "An oxidative central metabolism enables salmonella to utilize microbiota-derived succinate," *Cell host & microbe*, vol. 22, no. 3, pp. 291–301, 2017.

[145] P. Thiennimitr, S. E. Winter, M. G. Winter, M. N. Xavier, V. Tolstikov, D. L. Huseby, T. Sterzenbach, R. M. Tsolis, J. R. Roth, and A. J. Bäumler, "Intestinal inflammation allows salmonella to use ethanolamine to compete with the microbiota," *Proceedings of the National Academy of Sciences*, vol. 108, no. 42, pp. 17480–17485, 2011.

[146] Y. Litvak, M. X. Byndloss, and A. J. Bäumler, "Colonocyte metabolism shapes the gut microbiota," *Science*, vol. 362, no. 6418, p. eaat9076, 2018.

[147] C. A. Lopez, B. M. Miller, F. Rivera-Chávez, E. M. Velazquez, M. X. Byndloss, A. Chávez-Arroyo, K. L. Lokken, R. M. Tsolis, S. E. Winter, and A. J. Bäumler, "Virulence factors enhance citrobacter rodentium expansion through aerobic respiration," *Science*, vol. 353, no. 6305, pp. 1249–1253, 2016.

[148] W. Zhu, M. G. Winter, M. X. Byndloss, L. Spiga, B. A. Duerkop, E. R. Hughes, L. Büttner, E. de Lima Romão, C. L. Behrendt, C. A. Lopez, *et al.*, "Precision editing of the gut microbiota ameliorates colitis," *Nature*, vol. 553, no. 7687, pp. 208–211, 2018.

[149] Y. Litvak, M. X. Byndloss, R. M. Tsolis, and A. J. Bäumler, "Dysbiotic proteobacteria expansion: a microbial signature of epithelial dysfunction," *Current opinion in microbiology*, vol. 39, pp. 1–6, 2017.

[150] N.-R. Shin, T. W. Whon, and J.-W. Bae, "Proteobacteria: microbial signature of dysbiosis in gut microbiota," *Trends in biotechnology*, vol. 33, no. 9, pp. 496–503, 2015.

[151] E. Andersen-Nissen, K. D. Smith, K. L. Strobe, S. L. R. Barrett, B. T. Cookson, S. M. Logan, and A. Aderem, "Evasion of toll-like receptor 5 by flagellated bacteria," *Proceedings of the National Academy of Sciences*, vol. 102, no. 26, pp. 9247–9252, 2005.

[152] S. J. Clasen, M. E. Bell, D. Lee, Z. Henseler, A. Borbon, J. de la Cuesta-Zuluaga, K. Parys, J. Zou, N. D. Youngblut, A. T. Gewirtz, *et al.*, "Silent recognition of flagellins from human gut commensal bacteria by toll-like receptor 5," *bioRxiv*, 2022.

[153] M. Nedeljković, D. E. Sastre, and E. J. Sundberg, "Bacterial flagellar filament: a supramolecular multifunctional nanostructure," *International Journal of Molecular Sciences*, vol. 22, no. 14, p. 7521, 2021.

[154] R. M. Macnab, "Genetics and biogenesis of bacterial flagella," *Annual review of genetics*, vol. 26, no. 1, pp. 131–158, 1992.

[155] J. Kuczynski, C. L. Lauber, W. A. Walters, L. W. Parfrey, J. C. Clemente, D. Gevers, and R. Knight, "Experimental and analytical tools for studying the human microbiome," *Nature Reviews Genetics*, vol. 13, no. 1, pp. 47–58, 2012.

[156] M. Yassour, E. Jason, L. J. Hogstrom, T. D. Arthur, S. Tripathi, H. Siljander, J. Selvenius, S. Oikarinen, H. Hyöty, S. M. Virtanen, *et al.*, "Strain-level analysis of mother-to-child bacterial transmission during the first few months of life," *Cell host & microbe*, vol. 24, no. 1, pp. 146–154, 2018.

[157] F. Asnicar, S. Manara, M. Zolfo, D. T. Truong, M. Scholz, F. Armanini, P. Ferretti, V. Gorfer, A. Pedrotti, A. Tett, *et al.*, "Studying vertical microbiome transmission from mothers to infants by strain-level metagenomic profiling," *MSystems*, vol. 2, no. 1, pp. e00164–16, 2017.

[158] P. I. Costea, L. P. Coelho, S. Sunagawa, R. Munch, J. Huerta-Cepas, K. Forslund, F. Hildebrand, A. Kushugulova, G. Zeller, and P. Bork, "Subspecies in the global human gut microbiome," *Molecular systems biology*, vol. 13, no. 12, p. 960, 2017.

[159] J. Qin, R. Li, J. Raes, M. Arumugam, K. S. Burgdorf, C. Manichanh, T. Nielsen, N. Pons, F. Levenez, T. Yamada, *et al.*, "A human gut microbial gene catalogue established by metagenomic sequencing," *nature*, vol. 464, no. 7285, pp. 59–65, 2010.

[160] T. Seemann, "Prokka: rapid prokaryotic genome annotation," *Bioinformatics*, vol. 30, no. 14, pp. 2068–2069, 2014.

[161] D. H. Huson, R. Tappu, A. L. Bazinet, C. Xie, M. P. Cummings, K. Nieselt, and R. Williams, "Fast and simple protein-alignment-guided assembly of orthologous gene families from microbiome sequencing reads," *Microbiome*, vol. 5, pp. 1–10, 2017.

[162] K. R. Chng, T. S. Ghosh, Y. H. Tan, T. Nandi, I. R. Lee, A. H. Q. Ng, C. Li, A. Ravikrishnan, K. M. Lim, D. Lye, *et al.*, "Metagenome-wide association analysis identifies microbial determinants of post-antibiotic ecological recovery in the gut," *Nature ecology & evolution*, vol. 4, no. 9, pp. 1256–1267, 2020.

[163] R. H. Mills, P. S. Dulai, Y. Vázquez-Baeza, C. Sauceda, N. Daniel, R. R. Gerner, L. E. Batachari, M. Malfavon, Q. Zhu, K. Weldon, *et al.*, "Multi-omics analyses of the ulcerative colitis gut microbiome link bacteroides vulgatus proteases with disease severity," *Nature microbiology*, vol. 7, no. 2, pp. 262–276, 2022.

[164] C. J. Worby, H. L. Schreiber IV, T. J. Straub, L. R. van Dijk, R. A. Bronson, B. S. Olson, J. S. Pinkner, C. L. Obernuefemann, V. L. Muñoz, A. E. Paharik, *et al.*, "Longitudinal multi-omics analyses link gut microbiome dysbiosis with recurrent urinary tract infections in women," *Nature microbiology*, vol. 7, no. 5, pp. 630–639, 2022.

[165] R. A. Mars, Y. Yang, T. Ward, M. Houtti, S. Priya, H. R. Lekatz, X. Tang, Z. Sun, K. R. Kalari, T. Korem, *et al.*, "Longitudinal multi-omics reveals subset-specific mechanisms underlying irritable bowel syndrome," *Cell*, vol. 182, no. 6, pp. 1460–1473, 2020.

[166] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[167] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[168] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[169] A. E. D. Edler and M. Rosvall, "The MapEquation software package," 2022. `mapequation.org`.

[170] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[171] S. P. Borgatti, "Centrality and network flow," *Social networks*, vol. 27, no. 1, pp. 55–71, 2005.

[172] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, "Identifying modular flows on multilayer networks reveals highly overlapping

organization in interconnected systems," *Physical Review X*, vol. 5, no. 1, p. 011027, 2015.

[173] D. Edler, L. Bohlin, and M. Rosvall, "Mapping higher-order network flows in memory and multilayer networks with infomap," *Algorithms*, vol. 10, no. 4, p. 112, 2017.

[174] A. Eriksson, D. Edler, A. Rojas, M. de Domenico, and M. Rosvall, "How choosing random-walk model and network representation matters for flow-based community detection in hypergraphs," *Communications Physics*, vol. 4, no. 1, pp. 1–12, 2021.

[175] G. Bianconi, *Multilayer networks: structure and function*. Oxford university press, 2018.

[176] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.

[177] D. Edler, A. Holmgren, and M. Rosvall, "The MapEquation software package." `https://mapequation.org`, 2023.

[178] H. Gourlé, O. Karlsson-Lindsjö, J. Hayer, and E. Bongcam-Rudloff, "Simulating illumina metagenomic data with insilicoseq," *Bioinformatics*, vol. 35, no. 3, pp. 521–522, 2019.

[179] F. Meyer, P. Hofmann, P. Belmann, R. Garrido-Oter, A. Fritz, A. Sczyrba, and A. C. McHardy, "AMBER: assessment of metagenome BinnERs," *GigaScience*, vol. 7, no. 6, p. giy069, 2018.

[180] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: a flexible trimmer for illumina sequence data," *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, 2014.

[181] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature methods*, vol. 9, no. 4, pp. 357–359, 2012.

[182] J. Lloyd-Price, C. Arze, A. N. Ananthakrishnan, M. Schirmer, J. Avila-Pacheco, T. W. Poon, E. Andrews, N. J. Ajami, K. S. Bonham, C. J. Brislawn, *et al.*, "Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases," *Nature*, vol. 569, no. 7758, pp. 655–662, 2019.

[183] C. Rinke, S. Low, B. J. Woodcroft, J.-B. Raina, A. Skarshewski, X. H. Le, M. K. Butler, R. Stocker, J. Seymour, G. W. Tyson, *et al.*, "Validation of picogram-and femtogram-input dna libraries for microscale metagenomics," *PeerJ*, vol. 4, p. e2486, 2016.

[184] C. Bağcı, S. Patz, and D. H. Huson, "DIAMOND+MEGAN: fast and easy taxonomic and functional analysis of short and long microbiome sequences," *Current protocols*, vol. 1, no. 3, p. e59, 2021.

[185] B. Buchfink, K. Reuter, and H.-G. Drost, "Sensitive protein alignments at tree-of-life scale using DIAMOND," *Nature methods*, vol. 18, no. 4, pp. 366–368, 2021.

[186] D. H. Huson, B. Albrecht, C. Bağcı, I. Bessarab, A. Gorska, D. Jolic, and R. B. Williams, "Megan-lr: new algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs," *Biology direct*, vol. 13, no. 1, pp. 1–17, 2018.

[187] M. S. Mulani, E. E. Kamble, S. N. Kumkar, M. S. Tawre, and K. R. Pardesi, "Emerging strategies to combat ESKAPE pathogens in the era of antimicrobial resistance: a review," *Frontiers in microbiology*, vol. 10, p. 539, 2019.

[188] B. Stecher, R. Robbiani, A. W. Walker, A. M. Westendorf, M. Barthel, M. Kremer, S. Chaffron, A. J. Macpherson, J. Buer, J. Parkhill, *et al.*, "Salmonella enterica serovar typhimurium exploits inflammation to compete with the intestinal microbiota," *PLoS biology*, vol. 5, no. 10, p. e244, 2007.

[189] M. T. Henke, D. J. Kenny, C. D. Cassilly, H. Vlamakis, R. J. Xavier, and J. Clardy, "Ruminococcus gnavus, a member of the human gut microbiome associated with crohn's disease, produces an inflammatory polysaccharide," *Proceedings of the National Academy of Sciences*, vol. 116, no. 26, pp. 12672–12677, 2019.

[190] K. Kosulin, "Intestinal HAdV infection: tissue specificity, persistence, and implications for antiviral therapy," *Viruses*, vol. 11, no. 9, p. 804, 2019.

[191] K. Takahashi, G. Gonzalez, M. Kobayashi, N. Hanaoka, M. J. Carr, M. Konagaya, N. Nojiri, M. Ogi, and T. Fujimoto, "Pediatric infections by human mastadenovirus c types 2, 89, and a recombinant type detected in japan between 2011 and 2018," *Viruses*, vol. 11, no. 12, p. 1131, 2019.

[192] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, *et al.*, "KEGG for linking genomes to life and the environment," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D480–D484, 2007.

[193] J. Huerta-Cepas, D. Szklarczyk, D. Heller, A. Hernández-Plaza, S. K. Forslund, H. Cook, D. R. Mende, I. Letunic, T. Rattei, L. J. Jensen, *et al.*, "eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses," *Nucleic acids research*, vol. 47, no. D1, pp. D309–D314, 2019.

[194] S. M. Gibbons, "Keystone taxa indispensable for microbiome recovery," *Nature microbiology*, vol. 5, no. 9, pp. 1067–1068, 2020.

[195] L. S. Weyrich, A. G. Farrer, R. Eisenhofer, L. A. Arriola, J. Young, C. A. Selway, M. Handsley-Davis, C. J. Adler, J. Breen, and A. Cooper, "Laboratory contamination over time during low-biomass sample analysis," *Molecular ecology resources*, vol. 19, no. 4, pp. 982–996, 2019.

[196] J. Pereira-Marques, A. Hout, R. M. Ferreira, M. Weber, I. Pinto-Ribeiro, L.-J. Van Doorn, C. W. Knetsch, and C. Figueiredo, "Impact of host DNA and sequencing depth on the taxonomic resolution of whole metagenome sequencing for microbiome analysis," *Frontiers in microbiology*, vol. 10, p. 1277, 2019.

[197] M. R. Hasan, A. Rawat, P. Tang, P. V. Jithesh, E. Thomas, R. Tan, and P. Tilley, "Depletion of human DNA in spiked clinical specimens for improvement of sensitivity of pathogen detection by next-generation sequencing," *Journal of clinical microbiology*, vol. 54, no. 4, pp. 919–927, 2016.

[198] M. T. Nelson, C. E. Pope, R. L. Marsh, D. J. Wolter, E. J. Weiss, K. R. Hager, A. T. Vo, M. J. Brittnacher, M. C. Radey, H. S. Hayden, *et al.*, "Human and extracellular dna depletion for metagenomic analysis of

complex clinical infection samples yields optimized viable microbiome profiles," *Cell reports*, vol. 26, no. 8, pp. 2227–2240, 2019.

[199] R. Eisenhofer, J. J. Minich, C. Marotz, A. Cooper, R. Knight, and L. S. Weyrich, "Contamination in low microbial biomass microbiome studies: issues and recommendations," *Trends in microbiology*, vol. 27, no. 2, pp. 105–117, 2019.

[200] S. J. Salter, M. J. Cox, E. M. Turek, S. T. Calus, W. O. Cookson, M. F. Moffatt, P. Turner, J. Parkhill, N. J. Loman, and A. W. Walker, "Reagent and laboratory contamination can critically impact sequence-based microbiome analyses," *BMC biology*, vol. 12, no. 1, pp. 1–12, 2014.

[201] L. A. Kulakov, M. B. McAlister, K. L. Ogden, M. J. Larkin, and J. F. O'hanlon, "Analysis of bacteria contaminating ultrapure water in industrial systems," *Applied and environmental microbiology*, vol. 68, no. 4, pp. 1548–1555, 2002.

[202] Z. Sun, S. Huang, M. Zhang, Q. Zhu, N. Haiminen, A. P. Carrieri, Y. Vázquez-Baeza, L. Parida, H.-C. Kim, R. Knight, *et al.*, "Challenges in benchmarking metagenomic profilers," *Nature methods*, vol. 18, no. 6, pp. 618–626, 2021.

[203] D. E. Wood, J. Lu, and B. Langmead, "Improved metagenomic analysis with Kraken 2," *Genome biology*, vol. 20, no. 1, pp. 1–13, 2019.

[204] P. Menzel, K. L. Ng, and A. Krogh, "Fast and sensitive taxonomic classification for metagenomics with Kaiju," *Nature communications*, vol. 7, no. 1, pp. 1–9, 2016.

[205] D. T. Truong, E. A. Franzosa, T. L. Tickle, M. Scholz, G. Weingart, E. Pasolli, A. Tett, C. Huttenhower, and N. Segata, "MetaPhlAn2 for enhanced metagenomic taxonomic profiling," *Nature methods*, vol. 12, no. 10, pp. 902–903, 2015.

[206] A. Milanese, D. R. Mende, L. Paoli, G. Salazar, H.-J. Ruscheweyh, M. Cuenca, P. Hingamp, R. Alves, P. I. Costea, L. P. Coelho, *et al.*, "Microbial abundance, activity and population genomic profiling with mOTUs2," *Nature communications*, vol. 10, no. 1, pp. 1–11, 2019.

[207] A. Untergasser, I. Cutcutache, T. Koressaar, J. Ye, B. C. Faircloth, M. Remm, and S. G. Rozen, “Primer3—new capabilities and interfaces,” *Nucleic acids research*, vol. 40, no. 15, pp. e115–e115, 2012.

[208] W. R. Pearson, G. Robins, D. E. Wrege, and T. Zhang, “On the primer selection problem in polymerase chain reaction experiments,” *Discrete Applied Mathematics*, vol. 71, no. 1-3, pp. 231–246, 1996.

[209] C. Linhart and R. Shamir, “The degenerate primer design problem: theory and applications,” *Journal of Computational Biology*, vol. 12, no. 4, pp. 431–456, 2005.

[210] T. Grossman and A. Wool, “Computational experience with approximation algorithms for the set covering problem,” *European journal of operational research*, vol. 101, no. 1, pp. 81–92, 1997.

[211] Y.-C. Huang, C.-F. Chang, C.-h. Chan, T.-J. Yeh, Y.-C. Chang, C.-C. Chen, and C.-Y. Kao, “Integrated minimum-set primers and unique probe design algorithms for differential detection on symptom-related pathogens,” *Bioinformatics*, vol. 21, no. 24, pp. 4330–4337, 2005.

[212] R. M. Karp, *Reducibility among Combinatorial Problems*, pp. 85–103. Boston, MA: Springer US, 1972.

[213] J. Guo, D. Starr, and H. Guo, “Classification and review of free PCR primer design software,” *Bioinformatics*, vol. 36, no. 22-23, pp. 5263–5268, 2020.

[214] J. Fredslund, L. Schauser, L. H. Madsen, N. Sandal, and J. Stougaard, “PriFi: using a multiple alignment of related sequences to find primers for amplification of homologs,” *Nucleic acids research*, vol. 33, no. suppl_2, pp. W516–W520, 2005.

[215] A.-L. Lamprecht, T. Margaria, B. Steffen, A. Sczyrba, S. Hartmeier, and R. Giegerich, “GeneFisher-P: variations of genefisher as processes in bio-jeti,” *BMC bioinformatics*, vol. 9, no. 4, pp. 1–15, 2008.

[216] H. Sobhy and P. Colson, “Gemi: PCR primers prediction from multiple alignments,” *Comparative and functional genomics*, vol. 2012, 2012.

[217] P. D. Hebert, A. Cywinska, S. L. Ball, and J. R. DeWaard, "Biological identifications through dna barcodes," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. 1512, pp. 313–321, 2003.

[218] M. Razavi, N. P. Marathe, M. R. Gillings, C.-F. Flach, E. Kristiansson, and D. Joakim Larsson, "Discovery of the fourth mobile sulfonamide resistance gene," *Microbiome*, vol. 5, no. 1, pp. 1–12, 2017.

[219] K. Mendler, H. Chen, D. H. Parks, B. Lobb, L. A. Hug, and A. C. Doxey, "AnnoTree: visualization and exploration of a functionally annotated microbial tree of life," *Nucleic acids research*, vol. 47, no. 9, pp. 4442–4448, 2019.

[220] S. R. Eddy and the HMMER development team, "HMMER: biosequence analysis using profile hidden markov models," 2022. `http://hmmer.org/`.

[221] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012.

[222] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Appendix A

Appendix

A.1 Additional content

A.1.1 Reduction of optimal primer cover problem (OPC) from the set cover problem

Set cover problem. Given a set of elements U (termed *universe*), and a collection of sets $M = \{m_1, m_2, ...m_n\}$, each member of which is a subset of U, and their union covers U ($m_i \in M$, $m_i \subset U$, $\cup m_i = U$), is there a minimum sub-collection of sets $X \subset U$ that covers U?

If we represent each input string $s_i \in S$ in the OPC problem as a series of k-mers, S is the universe to be covered, then we could solve the OPC problem by solving the set cover problem. A toy example is shown below (Table A.1).

Table A.1: Formulate the OPC problem as a set cover problem: a toy example for picking a primer cover of length 4, with $k = 4$ and $|S| = 200$. There are $4^4 = 256$ possible 4-mers.

	4-mer_1	4-mer_2	...	4-mer_{256}
s_1	0	1	...	0
s_2	0	0	...	0
...	...	...	...	...
s_{200}	1	0	...	1

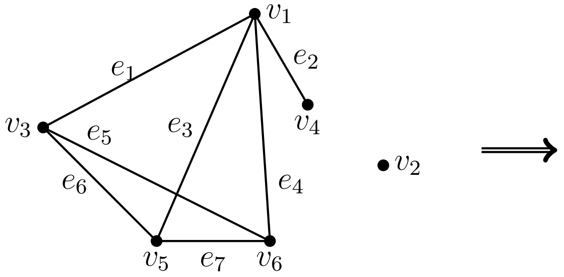
A.1.2 Reduction of the degenerate primer design problem (DPD) from the clique problem

In [209], the complexity of DPD problem (MC-DPD in the cited literature) is demonstrated by way of reduction from the clique problem. In an undirected graph, a subset of vertices is called a *clique* if any two of them are adjacent to each other. The clique problem is formulated as follows:

Maximum clique problem. Given a graph $\mathcal{G} = (V, E)$, which consists of a set of nodes V and a set of edges E, integer c, find a clique of size c in $\mathcal{G}$.

Let the number of $|V| = k$, $|E| = n$, the clique of size c is the number of input strings covered, i.e., coverage. Let integer $a \leq k, b \leq k$, $v_a, v_b \in V$. The problem can be reduced to a DPD problem with binary alphabet ($\Sigma = \{0, 1\}$) in Section 4.1.4: encode each edge $e_i = \{v_a, v_b\} \in E$ as a string $s_i = t_1 t_2 \ldots t_k$ of length k, with the character t_x being 1 if $x \in \{a, b\}$. For instance, in the example in A.1, $e_1 = \{v_1, v_3\}$, it can be encoded to string 101000. The degeneracy of the primer is given by 2^c, which is 8.

This problem is NP-complete. For DNA primers, $|\Sigma| = 4$, and it is not hard to see, the complexity will not be any less.



s_1: `101000`
s_2: `100100`
s_3: `100010`
s_4: `100001`
s_5: `001001`
s_6: `001010`
s_7: `000011`

Clique: $\{v_1, v_3, v_5, v_6\}$

Output primer: $*0*0**$

Figure A.1: An example of reducing a clique problem to a DPD problem. Adapted from [209]. The graph has 6 nodes and 7 edges, which is correspondent to the 7 sequences of length 6. A 4-node clique is found in the graph, resulting in the 4 degenerate positions in the DPD solution.

A.2 Supplementary tables

Table A.2: Source genomes of the *Random* simulated dataset

RefSeq ID	Name
GCF_013085055.1	Fusarium oxysporum Fo47
GCF_019915245.1	Fusarium musae
GCF_000011425.1	Aspergillus nidulans FGSC A4
GCF_000021265.1	Ureaplasma urealyticum serovar 10 str. ATCC 33699
GCF_001190745.1	Campylobacter gracilis
GCF_027626975.1	Streptomyces nigrescens
GCF_020736405.1	Odoribacter splanchnicus DSM 20712
GCF_002240355.1	Prauserella marina
GCF_020162295.1	Alysiella filiformis DSM 16848
GCF_011046555.1	Sphingobacterium lactis
GCF_001747425.1	Actinoalloteichus hymeniacidonis
GCF_002211785.1	Francisella halioticida
GCF_000242635.2	Solitalea canadensis DSM 3403
GCF_001025195.1	Bifidobacterium catenulatum DSM 16992 = JCM 1194 = LMG 11043
GCF_024181585.1	Treponema socranskii subsp. buccale
GCF_000055785.1	Chromohalobacter salexigens DSM 3043
GCF_008704495.1	Streptomyces kanamyceticus
GCF_000504085.1	Pseudothermotoga elfii DSM 9442 = NBRC 107921
GCF_000196135.1	Wolinella succinogenes DSM 1740
GCF_026651605.1	Alicyclobacillus dauci
GCF_015476235.1	Caldimonas thermodepolymerans
GCF_000196535.1	Mobiluncus curtisii ATCC 43063
GCF_003012915.1	Staphylococcus felis
GCF_000235605.1	Desulfosporosinus orientis DSM 765
GCF_000024085.1	Kangiella koreensis DSM 16069
GCF_900105065.1	Friedmanniella luteola
GCF_000024225.1	Lancefieldella parvula DSM 20469
GCF_000166055.1	Rhodomicrobium vannielii ATCC 17100
GCF_019443985.1	Liquorilactobacillus hordei DSM 19519
GCF_008806995.1	Neisseria animalis
GCF_000442645.1	Corynebacterium maris DSM 45190
GCF_000968375.1	Clostridium scatologenes
GCF_000024205.1	Desulfofarcimen acetoxidans DSM 771
GCF_001542625.1	Streptomyces griseochromogenes
GCF_024347055.1	Vibrio porteresiae DSM 19223
GCF_000025505.1	Ferroglobus placidus DSM 10642
GCF_000172995.2	Halogeometricum borinquense DSM 11551
GCF_000015145.1	Hyperthermus butylicus DSM 5456
GCF_000243255.1	Methanoplanus limicola DSM 2279

GCF_000166095.1	Methanothermus fervidus DSM 2088
GCF_000016385.1	Pyrobaculum arsenaticum DSM 13514
GCF_018200015.1	Haloarcula sinaiiensis ATCC 33800
GCF_000091665.1	Methanocaldococcus jannaschii DSM 2661
GCF_000147875.1	Methanolacinia petrolearia DSM 11571
GCF_000016525.1	Methanobrevibacter smithii ATCC 35061
GCF_000861165.1	Enterovirus C
GCF_000837225.1	Escherichia phage Mu
GCF_000846805.1	Human mastadenovirus A
GCF_003094435.1	Escherichia phage T2
GCF_000880515.1	Human mastadenovirus B

Table A.3: Source genomes of the *Half-random* simulated dataset

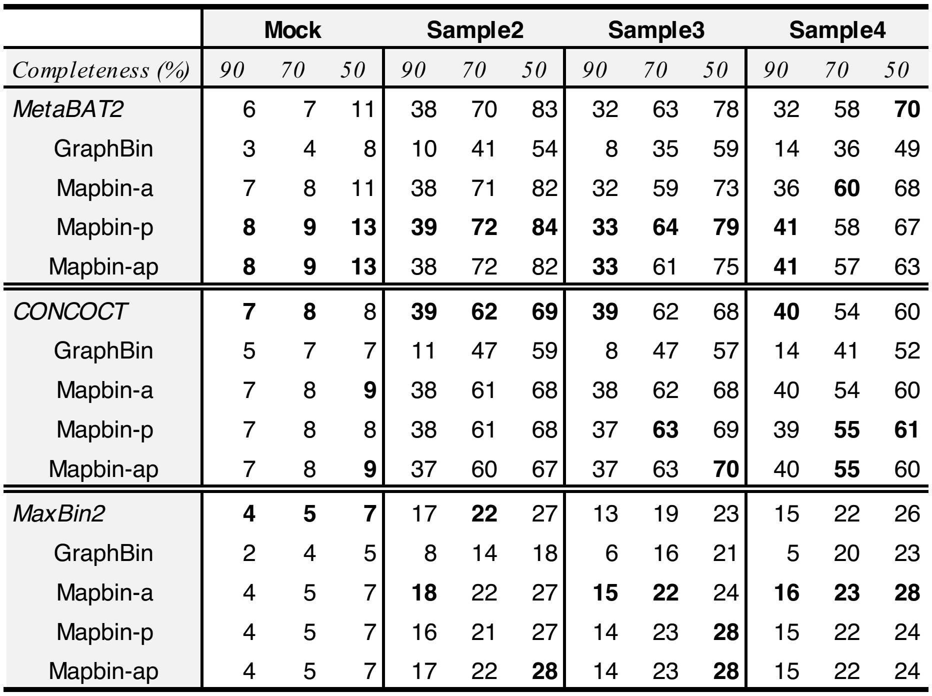
RefSeq ID	**Name**
GCF_000743255.1	Escherichia coli ATCC 25922
GCF_000019385.1	Escherichia coli ATCC 8739
GCF_003697165.2	Escherichia coli DSM 30083 = JCM 1649 = ATCC 11775
GCF_000464955.2	Escherichia coli O104:H21 str. CFSAN002236
GCF_007922655.1	Escherichia coli O157:H7
GCF_018141185.1	Methanobacterium alkalithermotolerans
GCF_000762265.1	Methanobacterium formicicum
GCF_000191585.1	Methanobacterium lacus
GCF_000214725.1	Methanobacterium paludis
GCF_002813695.1	Methanobacterium subterraneum
GCF_014023275.1	Nostoc edaphicum CCNP1411
GCF_002813575.1	Nostoc flagelliforme CCNUN1
GCF_001298445.1	Nostoc piscinale CENA21
GCF_000020025.1	Nostoc punctiforme PCC 73102
GCF_003443655.1	Nostoc sphaeroides
GCF_001687285.1	Pseudomonas aeruginosa
GCF_022699485.1	Pseudomonas aeruginosa
GCF_022699505.1	Pseudomonas aeruginosa
GCF_024507955.1	Pseudomonas aeruginosa
GCF_001045685.1	Pseudomonas aeruginosa DSM 50071
GCF_002531755.2	Rhizobium acidisoli
GCF_001664265.1	Rhizobium esperanzae
GCF_017352135.1	Rhizobium lentis
GCF_010669145.1	Rhizobium oryzihabitans
GCF_011046895.1	Rhizobium rhizoryzae
GCF_016861865.1	Aspergillus puulaauensis
GCF_026873545.1	Fusarium falciforme
GCF_019915245.1	Fusarium musae
GCF_000013325.1	Novosphingobium aromaticivorans DSM 12444

GCF_900637025.1	Streptococcus oralis ATCC 35037
GCF_000012485.1	Pelodictyon luteolum DSM 273
GCF_000517365.1	Spiroplasma mirum ATCC 29335
GCF_900090285.1	Micromonospora inositola
GCF_000512915.1	Barnesiella viscericola DSM 18177
GCF_000815065.1	Mesomycoplasma flocculare ATCC 27399
GCF_000233715.2	Desulfoscipio gibsoniae DSM 7213
GCF_000022325.1	Caldicellulosiruptor bescii DSM 6725
GCF_001275365.1	Francisella persica ATCC VR-331
GCF_000590925.1	Roseibacterium elongatum DSM 19469
GCF_000235405.2	Fervidobacterium pennivorans DSM 9078
GCF_000186365.1	Desulfurococcus mucosus DSM 2162
GCF_000147875.1	Methanolacinia petrolearia DSM 11571
GCF_946463545.1	Methanothermococcus thermolithotrophicus DSM 2095
GCF_000016525.1	Methanobrevibacter smithii ATCC 35061
GCF_026684035.1	Methanogenium organophilum
GCF_000871845.1	Dengue virus type 2
GCF_000894695.1	Hippeastrum mosaic virus
GCF_000836805.1	Chlamydia phage CPG1
GCF_000860865.1	Equine arteritis virus
GCF_000849665.1	Chlamydia phage 2

Table A.4: Intended composition of *Mock* benchmarking dataset

RefSeq ID	Name
GCF_000771585.1	Bifidobacterium actinocoloniiforme DSM22766
GCF_000771685.1	Bifidobacterium reuteri DSM23975
GCF_003697165.1	Escherichia coli DSM30083
GCF_000016525.1	Methanobrevibacter smithii DSM861
GCF_000157935.1	Prevotella copri DSM18205
GCF_003047065.1	Lactobacillus acidophilus DSM20079
GCF_000010425.1	Bifidobacterium adolescentis DSM20083
GCF_000020425.1	Bifidobacterium longum subsp. infantis DSM20088
GCF_000771225.1	Bifidobacterium pseudolongum subsp. pseudolongum DSM20099
GCF_000771285.1	Bifidobacterium longum subsp. suis DSM20211
GCF_001311295.1	Bifidobacterium breve DSM20213
GCF_900104835.1	Bifidobacterium longum subsp. longum DSM20219
GCF_001042595.1	Bifidobacterium dentium DSM20436
GCF_001025135.1	Bifidobacterium bifidum DSM20456
GCF_900099625.1	Lactococcus lactis subsp. lactis DSM20481

Table A.5: Numbers of high- to medium-quality bins generated by different methods from four real-world datasets. All bins counted here have contamination lower than 5%. Minimum completeness for bins of high quality: 90%, moderate-high quality: 70%, medium quality: 50%.



	Mock			**Sample2**			**Sample3**			**Sample4**		
Completeness (%)	*90*	*70*	*50*	*90*	*70*	*50*	*90*	*70*	*50*	*90*	*70*	*50*
MetaBAT2	6	7	11	38	70	83	32	63	78	32	58	**70**
GraphBin	3	4	8	10	41	54	8	35	59	14	36	49
Mapbin-a	7	8	11	38	71	82	32	59	73	36	**60**	68
Mapbin-p	**8**	**9**	**13**	**39**	**72**	**84**	**33**	**64**	**79**	**41**	58	67
Mapbin-ap	**8**	**9**	**13**	38	72	82	**33**	61	75	**41**	57	63
CONCOCT	**7**	**8**	8	**39**	**62**	**69**	**39**	62	68	**40**	54	60
GraphBin	5	7	7	11	47	59	8	47	57	14	41	52
Mapbin-a	7	8	**9**	38	61	68	38	62	68	40	54	60
Mapbin-p	7	8	8	38	61	68	37	**63**	69	39	**55**	**61**
Mapbin-ap	7	8	**9**	37	60	67	37	63	**70**	40	**55**	60
MaxBin2	**4**	**5**	**7**	17	**22**	27	13	19	23	15	22	26
GraphBin	2	4	5	8	14	18	6	16	21	5	20	23
Mapbin-a	4	5	7	**18**	22	27	**15**	**22**	24	**16**	**23**	**28**
Mapbin-p	4	5	7	16	21	27	14	23	**28**	15	22	24
Mapbin-ap	4	5	7	17	22	**28**	14	23	**28**	15	22	24

Table A.6: KOs of microbial genes encoding key respiratory oxidoreductases.

Orthologs	Function description in KEGG
Anaerobic respiratory enzyme-encoding genes	
K02567	napA; nitrate reductase (cytochrome) [EC:1.9.6.1]
K02568	napB; nitrate reductase (cytochrome), electron transfer subunit
K02569	napC; cytochrome c-type protein NapC
K02573	napG; ferredoxin-type protein NapG
K02574	napH; ferredoxin-type protein NapH
K12636	napH; N-acetylpuromycin N-acetylhydrolase
K00370	narG, narZ, nxrA; nitrate reductase / nitrite oxidoreductase, alpha subunit [EC:1.7.5.1 1.7.99.-]
K00371	narH, narY, nxrB; nitrate reductase / nitrite oxidoreductase, beta subunit [EC:1.7.5.1 1.7.99.-]
K00374	narI, narV; nitrate reductase gamma subunit [EC:1.7.5.1 1.7.99.-]
K07306	dmsA; anaerobic dimethyl sulfoxide reductase subunit A [EC:1.8.5.3]
K07307	dmsB; anaerobic dimethyl sulfoxide reductase subunit B
K07308	dmsC; anaerobic dimethyl sulfoxide reductase subunit C
K07310	ynfF; Tat-targeted selenate reductase subunit YnfF [EC:1.97.1.9]
K07311	ynfG; Tat-targeted selenate reductase subunit YnfG
K07312	ynfH; Tat-targeted selenate reductase subunit YnfH
K03532	torC; trimethylamine-N-oxide reductase (cytochrome c), cytochrome c-type subunit TorC
K07811	torA; trimethylamine-N-oxide reductase (cytochrome c) [EC:1.7.2.3]
K07812	torZ; trimethylamine-N-oxide reductase (cytochrome c) [EC:1.7.2.3]
K07821	torY; trimethylamine-N-oxide reductase (cytochrome c), cytochrome c-type subunit TorY
K03385	nrfA; nitrite reductase (cytochrome c-552) [EC:1.7.2.2]
K04013	nrfB; cytochrome c-type protein NrfB
K04014	nrfC; protein NrfC
K04015	nrfD; protein NrfD
K00244	frdA; succinate dehydrogenase flavoprotein subunit [EC:1.3.5.1]
K00245	frdB; succinate dehydrogenase iron-sulfur subunit [EC:1.3.5.1]
K00246	frdC; succinate dehydrogenase subunit C
K00247	frdD; succinate dehydrogenase subunit D
K00239	sdhA, frdA; succinate dehydrogenase flavoprotein subunit [EC:1.3.5.1]
K00240	sdhB, frdB; succinate dehydrogenase iron-sulfur subunit [EC:1.3.5.1]
K00241	sdhC, frdC; succinate dehydrogenase cytochrome b subunit
K00242	sdhD, frdD; succinate dehydrogenase membrane anchor subunit
K18859	sdhD, frdD; succinate dehydrogenase subunit D
K18860	sdhD, frdD; succinate dehydrogenase subunit D
K25995	frdB, fdrB; succinate dehydrogenase iron-sulfur subunit [EC:1.3.5.1 7.1.1.12]
K25996	frdC, fdrC; succinate dehydrogenase subunit C
Aerobic respiratory enzyme-encoding genes	

K00424	cydX; cytochrome bd-I ubiquinol oxidase subunit X [EC:7.1.1.7]
K00425	cydA; cytochrome bd ubiquinol oxidase subunit I [EC:7.1.1.7]
K00426	cydB; cytochrome bd ubiquinol oxidase subunit II [EC:7.1.1.7]
K02257	COX10, ctaB, cyoE; heme o synthase [EC:2.5.1.141]
K02297	cyoA; cytochrome o ubiquinol oxidase subunit II [EC:7.1.1.3]
K02298	cyoB; cytochrome o ubiquinol oxidase subunit I [EC:7.1.1.3]
K02299	cyoC; cytochrome o ubiquinol oxidase subunit III
K02300	cyoD; cytochrome o ubiquinol oxidase subunit IV
K00122	FDH; formate dehydrogenase [EC:1.17.1.9]
K00123	fdoG, fdhF, fdwA; formate dehydrogenase major subunit [EC:1.17.1.9]
K00124	fdoH, fdsB; formate dehydrogenase iron-sulfur subunit
K00126	fdsD; formate dehydrogenase subunit delta [EC:1.17.1.9]
K00127	fdoI, fdsG; formate dehydrogenase subunit gamma
K08348	fdnG; formate dehydrogenase-N, alpha subunit [EC:1.17.5.3]
K08349	fdnH; formate dehydrogenase-N, beta subunit
K08350	fdnI; formate dehydrogenase-N, gamma subunit
K22338	hylA; formate dehydrogenase (NAD+, ferredoxin) subunit A [EC:1.17.1.11]
K22339	hylB; formate dehydrogenase (NAD+, ferredoxin) subunit B [EC:1.17.1.11]
K22340	hylC; formate dehydrogenase (NAD+, ferredoxin) subunit C [EC:1.17.1.11]
K22515	fdwB; formate dehydrogenase beta subunit [EC:1.17.1.9]
K08348	fdnG; formate dehydrogenase-N, alpha subunit [EC:1.17.5.3]
K08349	fdnH; formate dehydrogenase-N, beta subunit
K08350	fdnI; formate dehydrogenase-N, gamma subunit
K00123	fdoG, fdhF, fdwA; formate dehydrogenase major subunit [EC:1.17.1.9]
K00124	fdoH, fdsB; formate dehydrogenase iron-sulfur subunit
K00127	fdoI, fdsG; formate dehydrogenase subunit gamma

Table A.7: COGs of microbial genes encoding key respiratory oxidoreductases.

Orthologs	Function description in eggNOG
Anaerobic respiratory enzyme-encoding genes	
COG3043	anaerobic respiration
COG3005	denitrification pathway
COG0437	Nadh dehydrogenase
COG5013	belongs to the prokaryotic molybdopterin-containing oxidoreductase family
COG1140	nitrate reductase beta subunit
COG2181	nitrate reductase activity
COG2180	chaperone-mediated protein complex assembly
COG3303	Catalyzes the reduction of nitrite to ammonia
COG2717	repairs oxidized periplasmic proteins containing methionine sulfoxide residues
COG1053	succinate dehydrogenase
COG0479	belongs to the succinate dehydrogenase fumarate reductase iron-sulfur protein family
COG3080	seems to be involved in the anchoring of the catalytic components of the fumarate reductase complex to the cytoplasmic membrane
COG0437	4 iron, 4 sulfur cluster binding
COG5557	Polysulphide reductase, NrfD
COG1923	positive regulation of translation, ncRNA-mediated
COG3303	Catalyzes the reduction of nitrite to ammonia
Aerobic respiratory enzyme-encoding genes	
COG1622	Cytochrome c oxidase subunit
COG0843	heme-copper terminal oxidase activity
COG1845	cytochrome c oxidase, subunit III
COG3125	oxidoreductase activity, acting on diphenols and related substances as donors, oxygen as acceptor
COG1271	aerobic electron transport chain
COG1294	oxidative phosphorylation
COG2864	formate dehydrogenase
COG0243	molybdopterin cofactor binding

Table A.8: Summary of AnnoTree FliC (KO K02406) hits.

	Count	**Sequence length**		
		min	max	mean
Proteobacteria	**11648**			
Rhizobiaceae	1784	83	659	328.4
Burkholderiaceae	1379	126	989	384.1
Vibrionaceae	930	199	499	369.5
Alteromonadaceae	902	60	591	309.2
Enterobacteriaceae	790	115	677	380.5
Rhodobacteraceae	767	186	886	373.8
Pseudomonadaceae	675	87	692	381.7
others	4421	73	1344	346.4
Firmicutes	**1567**			
Paenibacillaceae	301	154	686	327.4
Planococcaceae	195	192	1150	322.2
Bacillaceae_A	133	75	820	351.9
Amphibacillaceae	119	70	550	292.0
Bacillaceae	79	160	604	326.1
others	740	70	801	336.8
Firmicutes_A	**1362**			
Lachnospiraceae	713	87	861	367.9
Clostridiaceae	279	73	577	307.0
others	370	54	851	345.1
Spirochaetota	**811**			
Leptospiraceae	275	210	320	281.7
Treponemataceae	186	190	353	284.0
others	350	194	336	286.3
Campylobacterota	**645**			
Campylobacteraceae	207	248	712	429.8
Helicobacteraceae	186	250	564	490.1
Arcobacteraceae	172	203	538	285.2
others	80	232	813	443.7
Actinobacteriota	**571**			
Microbacteriaceae	172	194	441	314.0
Micrococcaceae	63	273	420	306.0
Cellulomonadaceae	58	188	404	331.4
others	278	196	529	313.1
Bdellovibrionota	**481**			
Bacteriovoracaceae	217	69	335	277.3
Bdellovibrionaceae	154	240	282	276.7
others	110	205	307	277.5
Desulfobacterota	**442**			
Desulfovibrionaceae	271	221	617	300.1

others	171	139	933	389.8
Firmicutes_C	**219**			
Selenomonadaceae	180	266	954	460.6
others	39	252	984	446.9
Planctomycetota	**161**			
SM1A02	43	442	530	497.3
Pirellulaceae	37	404	1190	684.1
Phycisphaeraceae	23	484	607	501.2
others	58	201	1041	374.1
Firmicutes_B	**127**			
Desulfitobacteriaceae	26	270	984	454.1
others	101	111	936	408.8
Thermotogota	**77**			
Fervidobacteriaceae	31	118	532	347.3
Petrotogaceae	28	210	876	405.3
Thermotogaceae	18	252	517	345.3
Verrucomicrobiota	**65**			
Opitutaceae	46	241	308	276.0
others	19	267	292	274.2
Nitrospirota	**51**			
Nitrospiraceae	28	274	275	274.6
Leptospirillaceae	8	275	298	283.4
Thermodesulfovibrionaceae	6	502	533	519.7
others	9	249	282	272.0
Others	**440**	76	995	384.3

Table A.9: Primers designed for bacterial *fliC* genes that are likely to occur in human gut microbiome. The primers are degenerate, with ambiguous bases following the IUPAC nucleotide code. d: degeneracy; c: coverage; n:size of the subgroup. The forward and reverse sequences form 1574 unique primer pairs. They are not presented as pairs because most pairs are only able to represent a very small ($n < 10$) subgroup. Unconventional bases: M: A or C; R: A or G; W: A or T; S: C or G; Y: C or T; K: G or T; V: A, C or G; H: A, C or T; D: A, G or T; B: C, G or T; N: A, C, G or T.

Forward primer	d	c	n	**Forward primer**	d	c	n
GTAMAACACAAYATTRCD	24	49	1.0	ATTCTKACSAACAACRGY	16	16	0.94
GTWCAGCAYAAYCTTWSA	32	47	0.98	GTYAAYACGAAYGTGTCS	16	16	1.0
GTWCAACAYAACATGCRR	16	45	1.0	ATYAATAACAAYATTCMR	16	16	1.0
GTACAACAYAATMTTHMA	24	43	1.0	ATCCAACATAATATCRBN	24	15	1.0
GTACARCAYAATMTKCAR	32	38	1.0	ATHGCAACAAAYAYCGCM	24	15	0.8
ATCAATACMAACATSAVS	24	36	1.0	ATMAAYCATAACWTRKCA	32	15	0.87
ATYAAYACTAACAGYHTG	24	35	0.91	ATYAAYACCAACTAYYTK	32	15	0.93
ATCATGACCAAYSCYGCS	16	35	0.97	ATYCARCACAACATVGCW	24	15	1.0
ATCAACACCAAYRTCRGY	16	33	1.0	ATTAAYAMYAACATYATG	16	15	0.93
RTCAAYAMCAACATCGCS	16	32	1.0	ATCAACACKAACGTYHYT	24	15	1.0
ATCAAYCAYAAYATNAGT	32	31	0.97	ATTAATACHAAYRTWTCA	24	15	0.87
GTWAATACWAACGTMKCW	32	31	0.9	GTAAAVHACAACATGTCC	9	15	1.0
ATTAATACYAACWTYGCW	16	30	1.0	AAYACCAAYKTGATGTCS	16	15	0.6
GTACAACAYAAYGTMMCA	16	30	0.97	GTTARTACTAAYGTSTCV	24	15	0.8
ATHAATMATAATATKTCM	24	30	1.0	ATWAATACCAACGTACYV	12	15	1.0
ATYAACACKAACGTDGGC	12	28	0.93	RTCAACACSAAYTCGGGS	16	15	1.0
ATYAATAAYAAYATTCAR	16	28	0.89	GTAAATACWAATGYKRGT	16	14	0.86
ATYAACCABAAYATYGCG	24	28	0.93	ATCAAYCAGAACATYKCY	16	14	0.93
GTACAGCAYAAYKTAWCW	32	27	1.0	ATYCAGCATAATATWGSH	24	14	0.93
RTCAACACCAACRTSTCK	16	27	0.93	ATTRYGACCAATGYGKCG	16	14	0.86
ATTAATCACAATMTDAVT	18	27	0.93	ATCCKGACGAACAYSGCY	16	14	0.79
ATCAACACCAAYGTBSSC	24	26	1.0	GTMAATACYAAYGTRAGC	16	14	1.0
RTCAATACCAACRTTGCD	12	26	1.0	GTTAAYACDAACGTTWCW	24	14	1.0
ATCAATACYAACCTKHTG	12	26	0.96	GTACAGCAYAACWTAAMW	16	14	0.93
ATYAATACAAACGTNBCA	24	25	0.92	GTAAAYASMAACMTTGCK	32	14	1.0
RTACAACACAAYTTAWCV	24	25	1.0	ATCARTACSAATGTTSCH	24	14	1.0
ATCAAYCACAATMTWARY	32	25	0.96	RTTAACWCAAAYRTAATG	16	13	0.92
RTCAATACCAAYRTCAMK	32	25	0.92	GTAAATACWAATATMAKB	24	13	0.85
CACACTAACTMCGCDTCR	12	25	0.96	GTAWCRACAAACATYGCR	16	13	0.85
GTHAARAACAABATGTCG	18	25	0.96	ATYAAYACCAACAGYMTC	16	13	1.0
ATYAAYCACAATATYKCW	32	25	0.96	ATTCAACAYAAYATTWCN	32	13	1.0
ATTCAACACAATATBRCW	12	25	1.0	GTCAACACAAACGTVKCN	24	13	1.0

GTACAGCAYAAYATSACH	24	25	1.0	GTAAMCACYAACRTCASY	32	12	0.92
ATTAACACMAATGTKSCV	24	24	0.88	GTMAAKACMAATRTATCG	16	12	1.0
ATYAACACCAACAWBKCC	24	23	0.96	CACACWAAYTAYGCWTCR	32	12	1.0
ATWYTGACSAAYGTCGCR	32	23	0.96	ATCCKYACCAACAWYKCC	32	12	1.0
ATYAACACCAATRYCRST	32	23	1.0	ATTTCWACAAACRTRGCR	16	11	1.0
ATCAAYACSAAYMTCCWG	32	23	0.87	AATACMAAYATTTCTKCR	16	11	0.45
ATCAAYACSAACSTMGCG	16	23	1.0	ATTCAWACYAACTACWMC	16	11	0.91
GTHAATACCAATGTKTCH	18	23	0.96	GTAAATTATAAYGTDTCM	12	11	1.0
GTHAATACCAAYGTTKCA	12	22	0.95	ATACAACAYAATRTTARY	16	11	0.82
ATYCARCAYAAYATYATG	32	22	1.0	ATTACSAATAAYGTNCAR	32	11	0.91
GTAAATACTAAYGTTDCN	24	22	1.0	RTTAAYTACAAYRTATCS	32	11	0.91
ATTAAYACAAACGTHGSY	24	22	1.0	ATTAAYMAYAACATTCMR	32	11	1.0
ATYAACAMCAACMTSGCY	32	22	0.95	ATAGSAACWAATGTRGCW	16	11	0.82
GTTAAYACCAATGTCABB	18	21	1.0	ATTAAYAAYAAYTTAAWR	32	10	1.0
ATYAAYACCAAYGTSAWG	32	21	0.9	ATCCAGCATAATHTGASS	12	10	1.0
ATWAATWCAAATATYKCY	32	21	0.95	ATCCASAACAACGTSRMK	32	10	0.9
ATCAATACVAATATCGCH	9	21	0.95	ATTWTGACGAAYACYKCS	32	10	1.0
ATYAACACSAACGTYSCW	32	21	0.95	WTGTCWATYTTRAATAAY	32	9	0.89
ATTAAYMACAATATYGCW	16	21	1.0	GTTAACACYAAYGTRAGY	16	9	1.0
ATCAACAAYAATRTTYCW	16	21	1.0	ATCGGAASCAAYAYCKCR	32	9	1.0
ATCAATCACAACDTDAGY	18	20	1.0	GTWMAACACAATATCACH	12	9	1.0
ATTAACAAYAACWTSTCW	16	20	0.95	ATWAACCAYAAYTTAGCM	16	9	1.0
ATWAATCACAACATYGCD	12	20	0.9	ATYAAYKTCAACGCCAGY	16	9	0.89
ATTCTKACCAAYACYTCS	16	20	1.0	ATCAACCACAACMTGRSY	16	9	1.0
ATTAACCAYAATATTBCR	12	20	0.95	ACCAAYATSACGTCGYTR	16	9	0.89
GTCAAYACBAACGTTGCG	6	20	1.0	ATYAATTWTAAYGCATCR	16	8	1.0
ATTYTGACNAACAATGSC	16	20	1.0	CAAAACATCMYRKCTYTG	32	8	0.5
GTACARCACAAYYTDCAG	24	19	1.0	ATYAAYACMAATGYTCYG	32	8	1.0
GTAAACACRAATRTSTCH	24	19	1.0	ATYAATTAYAATGTKTCV	24	8	1.0
ATYAATCACAAYWTGWTG	16	19	0.95	AACACRAAYMTSATGTCT	16	8	0.75
GTACAACACAAYATGKSW	16	19	1.0	ATTTCTACKAACGTAYMH	24	8	1.0
GTMAACACCAACGTRKCN	32	19	0.95	TAYCAAAAYGTASCKGCT	16	7	0.71
GTSAACACCAATRYTGCV	24	19	0.95	ATYAACTTYAAYTCKTCY	32	7	1.0
ATTAACACYAACGTTDCR	12	18	1.0	ATCAATTACAWYSYATCW	32	7	1.0
ATTAAYCAYAATATMCMR	32	18	1.0	ATAAACARDAATWTRAGT	24	6	1.0
ATCAACACAAACATKBCR	12	18	1.0	ATGKCRATCCTGAATAAT	4	5	1.0
RTYAAAAAYAAYATGTCR	32	18	1.0	ATTTCAAACAATGTMCMD	12	4	1.0
ATYAATACMAACACCGCV	12	18	0.94	TTGTCGTCAATYAAWAVY	24	4	0.75
RTAAAYACWAACGTAKYA	32	18	1.0	ACCAACGTGWCHGCRATK	24	4	0.5
GTACAGCAYAAYHTKCAG	24	18	1.0	ACCAATATAGCBTCMATS	12	4	0.75
ATTTTGACAAACACHTCH	9	17	0.94	ACTGATATTGCWGRKGMW	32	3	0.67
ATTCTSACSAAYAMCGGY	32	17	1.0	AGCTCGGGCATGCDSAWY	24	3	0.67
ATTYTGACCAACHCMGCC	12	17	1.0	GTGACGCAGCAAASHYTG	12	3	1.0

ATCAAYACCAACAYCHTK	24	17	0.94	CACACKAACTACGCCAAC	2	2	1.0
ATCAAYACCAAYWCKCTG	16	17	0.94	TTGAATGARWCRCAYTCR	32	2	1.0
ATTCTCACCAACRTHGCD	18	17	0.94	CACACGAACTACGCCAAC	1	1	1.0
ATYAAYASMAACATCARC	32	17	1.0				
Reverse primer	*d*	*c*	*n*	**Reverse primer**	*d*	*c*	*n*
SAKWGAAAGAACDCCCTG	24	50	1.0	SARGCTSAGTACRTTGGA	16	15	0.93
SAGVGMCAGCACGCCCTG	12	46	0.98	AAKCTGAAGNAYCTGKGA	32	15	0.93
AAGWGWMAGAACWCCCTG	16	45	1.0	WAGYTGTARTRCTGCCTG	16	14	1.0
MARCTKCAGCACRCCCTG	16	43	0.95	YAAYTGSAGTGCYARTTG	32	14	0.79
CAGVGACAKGAYGTTCTG	12	43	1.0	YAGTGAHAGDACGTTCTG	18	14	1.0
YARTTGTAAHACRCCTTG	24	38	1.0	SAGCTKSAGCGCGAKCTG	16	14	0.93
CAGNGHCAGMACGTTCTG	24	36	1.0	YAAGCTMAGMGCTGMMGA	32	14	0.79
CAGBGAYAGRGCCAGCTG	12	35	0.86	CARRCTYARTGCTGAGTT	16	14	1.0
VARCGAGAGVACGTTCTG	18	34	1.0	RAGYKRAAGAGCMATCTG	32	14	0.86
YAACTGRAGWACDCCCTG	24	34	1.0	YAACTGYARTAYCTGTGA	16	13	0.92
SARAGACAGRACRKTCTG	32	32	0.97	MAGGCTVAGGAYCGACGA	12	13	0.69
YAAYTGAAGMACDCCTTG	24	31	0.97	TAAGYTKARTGCRTTMTT	32	13	0.77
RAKTGATAATACDCYCTG	24	28	1.0	YAACTGGAGRAYAGMCTG	16	13	1.0
SAGGCTSAGVASGTTCTG	24	28	0.93	MAGYTTYAAWACAGCTTG	16	13	0.62
GAGCTGCAGKACDCCCKS	24	28	0.93	CARCTGCAGYGCMABCTG	24	13	1.0
GAGCGAVAKGAYGYTCTG	24	27	1.0	TAACTGYAWHACACCCTG	12	13	0.92
GAKHGAAAGTACWCCCTG	12	27	1.0	YRATTTCACYACYTGRTC	32	13	0.92
MAKCTGCARWACCTGCTG	16	27	0.96	BAGHGACATTGCCATGCY	18	13	0.92
VAGCGACAGCAYYTGYTG	24	26	0.92	HAGYTGCAACRCCGCCTG	12	12	0.92
SAGGCTGAGYACNGCCTG	16	26	1.0	CARYSKCAGGATGTTTTC	16	12	0.92
YAATGATAAWACRCYCTG	16	26	1.0	WAGYTGTAAYACWGACTG	16	12	0.92
MAKWGAMAGTACWCCCTG	32	25	1.0	CAGGSWMRTKACCATGCC	32	12	0.75
CAGCGMCAGRATSGHCTG	24	25	1.0	RAGCGARAGRACAGCSGA	16	12	0.83
YAACTGMARMACCTGYTG	32	25	0.96	CAAGCTCARWACGCYGYT	16	12	0.83
SAGKGACAGCATGGHCTG	12	25	0.96	CATTATWGMYTGRTTTTT	16	12	1.0
TAAKSMAAGTACRCCCTG	16	24	0.96	YARCTGYAAWGCAGCTTG	16	11	0.91
SAGCGASAKSACRCCCTG	32	23	1.0	CAGGCGSAGDATGTTYTC	12	11	1.0
TAATKGTAATACTRMYTG	16	22	0.86	SAGRCGCAGGATRTTYTS	32	11	0.91
TAARGATAAWACDCCYTG	24	22	1.0	YARTGWCAKYGCAGAGTT	32	11	1.0
TAGHGASATYGCCATRCC	24	22	0.91	WARMGACAATGCTRCCTG	16	11	1.0
RAGCTGMAGYRCCGACTG	16	22	0.91	YARYGAAAGWGCCAWTTG	32	10	0.8
YAATTKAAGWACRTTTTG	16	22	0.91	WARWGAWARTACACTTTG	32	10	1.0
TAWTGARAGAACRCCYTG	16	22	1.0	RAGWTTCAWAACACCYTG	16	10	1.0
SAGRGARAGTACWCCCTG	16	21	1.0	YARRCCAAGWGCAGCRTT	32	10	0.8
SAGCTKRAGCACGCYYTG	32	21	1.0	HARMGTMAGTGCAAGATT	24	10	0.7
TARCTGYARKACWCCTTG	32	21	1.0	YARYTTAAGAAYGGATTG	16	10	0.8
TAARGMMAGAACDCCCTG	24	21	0.95	YAGRCCCATYACCAGRCC	16	10	0.9
MAGDKACAGAACGCYCTG	24	20	0.95	TAARKATAAVGCSATATT	24	10	0.9

GAGCGASAGGAYCGWYTG	16	20	0.95
CAGCGASAGVACCGTYTG	12	20	0.95
TARYTGWAGTACDCCTTG	24	20	0.95
CAATTKCAGWACGCYYTG	16	20	0.95
YAAKGACAWTACRCCCTG	16	20	1.0
KAGWGACAKMACWCCCTG	32	20	1.0
MAGWGACARTACRCCCTG	16	19	0.79
RAGCTKSAGAAYGCCYTG	32	19	1.0
SAGYTKCARGACCTGYTG	32	19	1.0
YAAGGACARYGCCARCTG	16	19	0.95
SAGCTTSAGGAYGYTCTG	16	19	0.95
VAGGCTSAGSACGCCCTG	12	19	1.0
NAGYTGCARTACRCCTTG	32	19	0.95
SAGTTTCAGHRCAGTCTG	12	19	0.84
MAGCTGYAAYACCTSWGA	32	18	1.0
YARTGWYARCGCAATTTG	32	18	0.83
CARYTTCAGTACRGCSGA	16	18	0.72
SAGCGACAGCRCGWTGTY	16	18	0.56
SAGRGASARWACCTGCTG	32	18	1.0
CARCTKCARBACGTTTTG	24	18	0.94
GAGCGWSAGGAYGYTTTC	16	18	1.0
CAGMGWCAGKRCCGWCTG	32	18	0.94
CARYTGTAAAATYCCYTG	16	18	0.83
CARVGMCAGCACACCYTG	24	18	0.78
SAGACTSAGHACAGCCTG	12	18	1.0
CARAGACARHACYTGCTG	24	18	1.0
TARTTKYAATAYATTYTG	32	18	0.94
BAGACTTARWGCAATCTK	24	18	0.78
SARCTTGAGGAYCRACTG	16	17	0.76
GAGCGWSARAAYGTTCTG	16	17	0.94
YARTTGAAGMACYTGYTG	32	17	0.94
CAGAGACARDACWGTYTG	24	17	0.88
AAGYTGNAGAACWCCTTG	16	17	1.0
SAGGCTSAGRACCGWGCT	16	17	0.88
CAADGAHAGWACAGATTG	18	17	0.94
WAGYTKYARAGCCATTTG	32	16	0.81
WAGYTKAAGMACTGATTG	16	16	0.88
SAGGCTGAGNACCGMCTG	16	16	0.94
CARGCTMAGARCWCCYTG	32	16	0.94
CAGCCKCARAATCABCTG	12	16	1.0
YARTTTTARAATSGAYTG	32	16	0.94
WARCTGYAAWACCTGCTG	16	16	0.94
YARRCGCARRATCATCTG	32	16	1.0
CARGCKAAGMGCAAYAGT	16	10	0.7
YAGTCTYAGTGCAGAKTK	16	10	0.9
CARCWGCAKCRGGGTYTG	32	10	0.8
CAKRYTYAATACACCCTG	16	9	0.89
HAGASWCATAGCAATTGA	12	9	0.78
NAGAGWTAGWGCAGCYTG	32	9	0.89
CAGCGCSARSRCTKGCTG	32	9	1.0
SAGGCTSARSGCCAGCTG	16	9	1.0
WAGKCTYAAWGCCGCTTG	16	9	0.89
TAATTKWSTTAVCATTTG	24	9	1.0
GAGCTGVAGRATGTTYTS	24	8	1.0
CAGGCTBGARACCATGCY	12	8	1.0
TARTSCTAATGCWGCAYT	16	8	1.0
AAGCTGCARYATCAKYTS	32	8	0.88
YTTCAKCGCYGCGYYCGG	32	8	0.75
TWMAGCATAATCWACSTC	16	8	0.75
SAGCTGGAGRATRTTYTC	16	8	0.88
TARRGAWARWATGGTCTG	32	8	0.75
YAAKCKCAKYACTGACTG	32	8	0.75
NAGCTSCAGCACCGWMGA	32	8	0.88
DAGTGAAAGTACYTGYTG	12	8	1.0
RAGSCTYAATAYGCTCTG	16	7	0.86
AAKCTGYAATACYTCASY	32	7	1.0
DCCATCKAACAGGTTCWT	12	7	0.43
MAGDSCYAATGCTGCATT	24	7	0.86
SAGSKTYAGAGCCACGGT	16	6	0.83
SATRTTGKTKTTSGCTTA	32	6	0.5
YARWGTCAKCACTTGGTT	16	6	1.0
ACCCTTTAAYARGYATGW	16	6	0.5
YTTAAGGGCCGCWGMRCC	16	6	0.67
YAAVGATARGACATTGTT	12	6	0.83
AYYCWGRATAGCCTGTGS	32	6	0.5
CAGCGMWGCCACMATCTS	16	6	0.83
YTTHARTGCCAATTGRTT	24	6	0.83
CGTSAGGATGYYYTCCGC	16	6	0.67
BARTCYCAGGGCAYTTTT	24	6	0.83
ASCYARTGCAWTCTTTGT	16	5	0.6
YTTSAKYGCAACTTGWGG	32	5	1.0
MKTAAGRGCGTTAKTAST	32	5	0.6
RAGKKTSAGCACGGCACT	16	5	0.8
RTTCTTGSTGTAGKYAAC	16	4	1.0
KCYKAGCACKGCTGAAGS	32	4	0.75
MRTWGCYGTAGCTGCYTG	32	4	0.75

KAGCTKCAKKGCCATCTK	32	15	1.0	TYTTARRGCTAGYTGAGG	16	4	0.5
CARTGAYAATRCTGAYTG	16	15	1.0	DYCTGCCTGAGSAAGAAT	12	4	0.5
CARACTCARYACTGCRCT	16	15	0.87	TCCAAGKGCKWYGTTAGC	16	3	0.67
CARTGACARYGCTGCYTG	16	15	0.8	RDWRTCGGCGTCACGGAT	24	3	0.67
CAGGCKCAGGACGVSCTG	12	15	0.93	YTTTAASMKATAGGTTTC	16	3	0.67
GAGCTGSAGRATRTTYTC	16	15	0.93	TTGATATAGCTTGAGYGC	2	2	1.0
YTTAAGMGCMAAYTGGTT	16	15	0.87	CTCAAGCATTGCKGTCGC	2	2	1.0

A.3 Supplementary figures

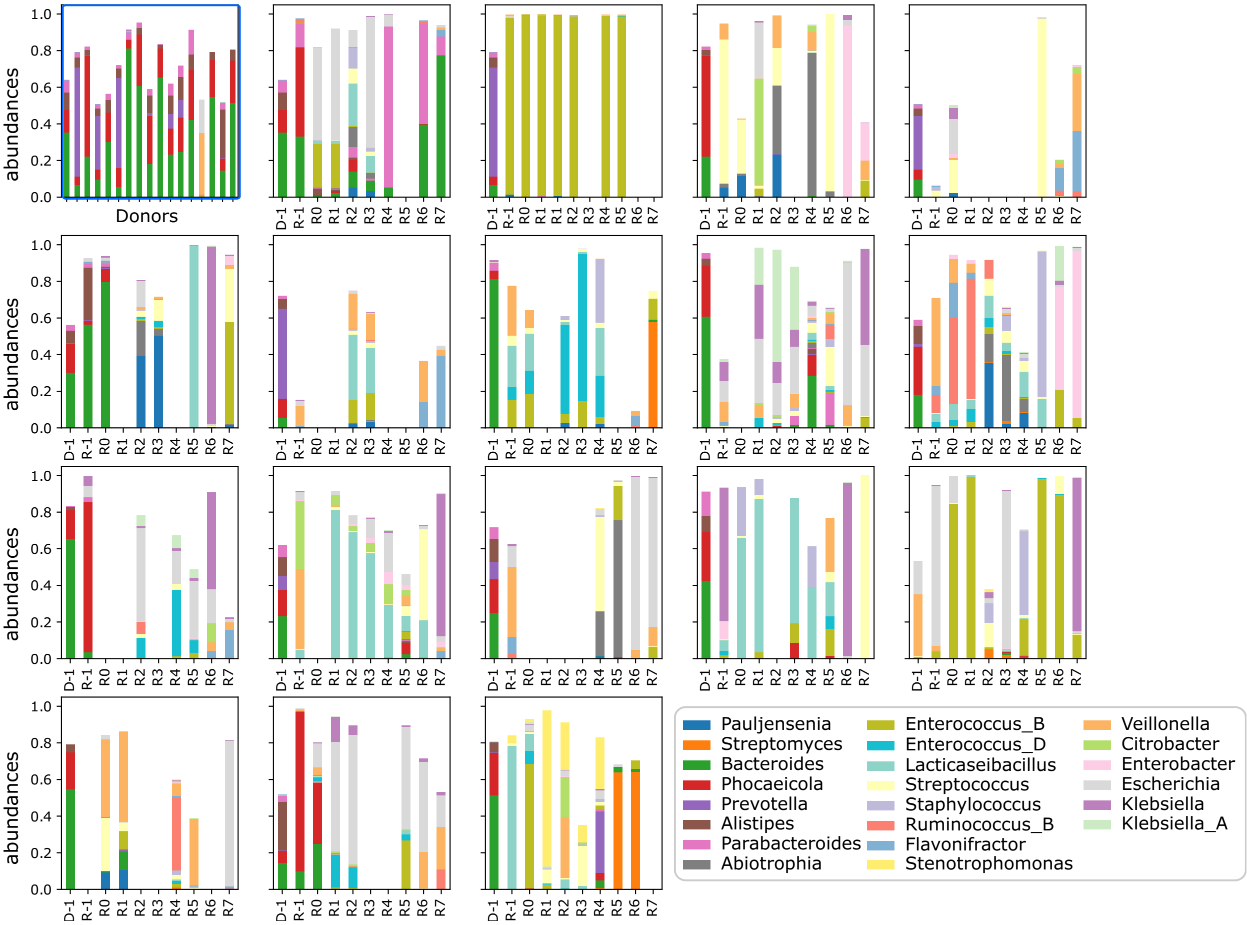


Figure A.2: Gut microbiota composition of all samples at the GTDB genus level, grouped by donors (highlighted with a blue frame) or donor-recipient pairs. The illustration style is similar to Fig. 3.3.

Figure A.3: Primer3 output for the input in Table 4.1

```
Template masking not selected
No mispriming library specified
Using 1-based sequence positions
OLIGO            start  len      tm     gc\%  any_th  3’_th hairpin seq
LEFT PRIMER         15   18   51.58   47.06    0.00   0.00    0.00 ATGACNACTGACGATGCA
RIGHT PRIMER       102   20   55.85   50.00    1.71   0.00    0.00 CGTAGCTATCGATTTGGGTC
SEQUENCE SIZE: 108
INCLUDED REGION SIZE: 108

PRODUCT SIZE: 88, PAIR ANY_TH COMPL: 0.00, PAIR 3’_TH COMPL: 0.00
TARGETS (start len

    1 GTAGTCAGTAGACNATGACNACTGACGATGCAGACNACACACACACACACAGCACACAGG
                    >>>>>>>>>>>>>>>>>>    *********************

   61 TATTAGTGGGCCATTCGATCCCGACCCAAATCGATAGCTACGATGACG
                            <<<<<<<<<<<<<<<<<<<<

KEYS (in order of precedence):
****** target
>>>>>> left primer
<<<<<< right primer

ADDITIONAL OLIGOS
                    start  len      tm     gc%  any_th  3’_th hairpin seq

 1 LEFT PRIMER         15   15   41.61   42.86    0.00   0.00    0.00 ATGACNACTGACGAT
   RIGHT PRIMER       102   16   45.70   43.75    1.71   0.00    0.00 CGTAGCTATCGATTTG
   PRODUCT SIZE: 88, PAIR ANY_TH COMPL: 0.00, PAIR 3’_TH COMPL: 0.00

Statistics
         con   too    in    in   not          no    tm    tm   high  high  high        high
         sid  many   tar  excl    ok   bad    GC   too   too any_th 3’_th hair-  poly   end
        ered    Ns   get   reg   reg   GC\% clamp   low  high  compl compl   pin     X  stab    ok
Left      62    17     0     0     0     0     0     7     0      0     0    23     0     0    15
Right    291     0     0     0     0     0     0     0     0      0     0     4     0     0   287
Pair Stats:
considered 4124, unacceptable product size 1444, tm diff too large 2678,
primer in pair overlaps a primer in a better pair 2379, ok 2

libprimer3 release 2.4.0
```