Characterizing bacterial genetic diversity
In species' pangenomes and microbial communities

van Dijk, L.R.

DOI
10.4233/uuid:10b63def-b224-419f-8817-7aea97448aab

Publication date
2025

Characterizing bacterial genetic diversity

In species' pangenomes and microbial communities

Characterizing bacterial genetic diversity

In species' pangenomes and microbial communities

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op Donderdag, 30 Januari 2025 om 15:00.

door

Lucas Roeland van Dijk

Master of Science in Computer Science,
Delft University of Technology, Delft, the Netherlands,
geboren te 's-Gravenhage, Nederland.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. M.J.T. Reinders	Technische Universiteit Delft, *promotor*
Dr. T.E.P.M.F. Abeel,	Technische Universiteit Delft, *promotor*
Onafhankelijke leden:	
Prof. dr. M.M. de Weerdt	Technische Universiteit Delft
Prof. dr. R.J.L. Willems	Universiteit Utrecht
Prof. dr. M.H. Medema	Wageningen Universiteit
Prof. dr. ir. J. Fostier	Universiteit Gent
dr. N. Yorke-Smith	Technische Universiteit Delft

dr. Ashlee M. Earl, dr. Abigail L. Manson en dr. Kiran V Garimella hebben in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

Keywords: Bacterial genetics, strain-level diversity, microbiome, pangenomics

Printed by: ProefschriftenPrinten.nl

Front & Back: Susan van Dijk.

ISBN 978-94-6384-723-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Contents

Summary

Bacteria are everywhere and play essential roles in Earth's diverse ecosystems and human health. For example, humans harbor a complex and essential gut microbial community comprising thousands of bacterial species (in addition to numerous viruses, fungi, and microbial eukaryotes). This community helps break down and synthesize nutrients, trains the immune system, and keeps pathogens at bay. However, imbalances in this community are associated with several diseases, including obesity, inflammatory bowel syndrome, and recurrent urinary tract infections. Moreover, bacteria can cause deadly infections, and many are developing resistance to our most potent antibiotics. Studying bacteria is thus essential for identifying differences between pathogens and harmless commensals, countering antimicrobial resistance, and understanding their impacts on human health.

To study bacteria, we typically characterize and compare their genomes. The genome comprises all of an organism's hereditary information, and its genes encode the molecular machines necessary for cell function, providing an overview of an organism's capabilities. The hereditary information in genomes additionally enables inferring the organism's evolutionary history, which helps in understanding why specific traits evolved or aid in inferring transmission links in case of an outbreak.

A challenge with comparing large sets of bacterial genomes is the extensive variation in genome content among many species. For example, two *Escherichia coli* strains can share as little as 50% of their genes. Current computational tools offer biased or incomplete views of genetic variation among strains. This hinders the identification of genotype-phenotype associations, prevents tracking mobile genetic elements, and limits our understanding of the microbial communities they are part of.

The central question of this thesis is how to design computational tools that enable accurate characterization of genetic variation among diverse bacterial genomes. This thesis introduces new algorithms to identify and represent genetic variation using graph data structures. It additionally presents a tool that characterizes strain-specific genetic variation in microbial communities, even in the presence of same-species strain mixtures. Finally, this thesis uses the previously mentioned tools to investigate the role of the gut microbiome in women with recurrent urinary tract infections, offering novel insights into the gut and bladder dynamics of *E. coli*.

Collectively, we expect this work to contribute to an improved mechanistic understanding of bacteria's role in human health, help track and counter the spread of antimicrobial resistance, and inform on the development of microbiome-mediated therapeutics.

Samenvatting

Overal om ons heen zijn bacteriën te vinden en ze zijn essentieel in diverse ecosystemen op aarde en belangrijk voor de menselijke gezondheid. De menselijke darmflora is een voorbeeld van een complexe gemeenschap die duizenden bacteriesoorten bevat (naast vele soorten virussen, schimmels, en eencellige eukaryoten). De darmflora helpt met het verwerken van voedsel, traint het immuunsysteem, en houdt ziekteverwekkers buiten de deur. Aan de andere kant, als de darmflora uit balans raakt kan dat leiden tot verschillende ziektes zoals obesitas, prikkelbaredarmsyndroom, of terugkerende blaasontstekingen. Bacteriën kunnen verder ook dodelijke infecties veroorzaken en worden steeds meer resistent tegen onze sterkste antibiotica. Het is dus belangrijk om bacteriën te bestuderen om uit te vinden hoe deze onze gezondheid beïnvloeden, beter te begrijpen wat de verschillen zijn tussen ziekteverwekkers en bacteriën die zonder problemen met ons leven, en het tegengaan van antibiotica resistentie.

We bestuderen bacteriën vaak door hun genomen te vergelijken. Het genoom omvat alle erfelijke informatie van een organisme. De genen erin bevatten de instructies om de moleculaire machines te produceren die een cel nodig heeft voor zijn functioneren. De erfelijke informatie kan verder worden gebruikt om de evolutionaire historie te reconstrueren. Dit helpt met het begrijpen waarom bepaalde eigenschappen zijn ontstaan of het reconstrueren van een transmissienetwerk tijdens een uitbraak.

Het vergelijken van vele genomen wordt bemoeilijkt door de enorme genetische diversiteit in veel bacteriesoorten. Bijvoorbeeld, twee *Escherichia coli* stammen delen soms maar de helft van hun genen. De huidige algoritmes voor het vergelijken van genomen geven een incompleet of vooringenomen beeld tussen de verschillen tussen de stammen. Dit limiteert het vinden van genotype-fenotype associaties, bemoeilijkt het volgen van genen die antibiotica resistentie veroorzaken, en beperkt het inzicht in de gemeenschappen waarin de bacteriën zich bevinden.

De hoofdvraag in deze thesis is hoe we betere algoritmes en software kunnen ontwerpen die beter de genetische verschillen in kaart kunnen brengen tussen diverse bacteriële genomen. We introduceren nieuwe algoritmes om zulke verschillen te vinden en te representeren met behulp van graafdatastructuren. Verder presenteren we een nieuw algoritme die stam-specifieke genetische variaties kan karakteriseren in diverse microbiële gemeenschappen, zelfs als er meerdere stammen van dezelfde soort aanwezig zijn. Deze tool wordt gebruikt om te onderzoeken wat de rol van de darmflora is in vrouwen met terugkerende blaasontstekeningen. Dit levert nieuw inzichten op over de aanwezige *E. coli* in darm en blaas.

Wij verwachten dat deze thesis zal bijdragen aan nieuwe inzichten over hoe bacteriën onze gezondheid beïnvloeden, het volgen en tegengaan van de spreiding van antibioticaresistentie, en de ontwikkeling van nieuwe darmflora-gebaseerde medicijnen.

1

Introduction

BACTERIA are present in countless environments in almost every imaginable corner of our planet. Whether it is near hot vents deep on the ocean floor [1], in hot and acidic springs [2], or in the soil of a backyard [3], bacteria likely inhabit that space. They can be traced back billions of years in the geological record [4], long before the rise of eukaryotic organisms. For all those years, "bacteria [and archea]—the only inhabitants—continuously transformed the planet's surface and atmosphere and invented all of life's essential miniaturized chemical systems" [5].

Their omnipresence also means they shape human health and society in important ways. For example, while many help us digest food and keep pathogens at bay, some can cause deadly infections. We study bacteria to maximize their positive and limit their negative impacts. Studying bacteria frequently starts with characterizing and comparing their genomes, and advances in DNA sequencing technology have made sequencing complete genomes routine.

In this thesis, we focus on computational tools for comparing genomes and how we obtain novel biological insights from them. This chapter will first expand on the numerous impacts of bacteria on our society and why studying them is important. We will explain methods for characterizing the genomes of a single strain as well as methods for characterizing complete microbial communities. We will discuss current algorithms and tools to compare genomes and the common challenges they face. Finally, we will briefly cover the outline of the rest of the thesis and our contributions to the challenges facing bacterial genomics today.

1.1. Many bacteria are beneficial to humans, though some cause deadly infections

Bacteria impact human society in a myriad of ways, both positive and negative. For example, bacteria colonize many sites on and within ourselves, e.g., the skin, the mouth, or the gut [6]. Many of those colonizing bacteria benefit us: they help to prevent pathogens from invading and help to digest food [7]. Another way bacteria have positively impacted human society is through their role in food fermentation. Nearly every human cuisine includes fermented foods, including sourdough, yogurt, kefir, cheese, and kimchi, among many others [8]. Bacteria play a crucial role in many of these. Fermentation aids in preserving food in the absence of refrigeration and likely provided early human societies with a method to store food surpluses of one season to survive more scarce seasons [9].

Bacteria are additionally a rich source of useful molecular tools. For example, the polymerase chain reaction (PCR) method, which rapidly makes many copies of a piece of DNA, enables genetic analyses even when the input DNA quantities are very low. This has many applications and is fundamental to many medical diagnostic tests, including the SARS-CoV-2 test [10]. PCR works by successively cycling between high temperatures, which denatures the two strands of DNA, and low temperatures, at which an enzyme called a polymerase replicates each strand of DNA [11]. Specifically, it relies on the *Taq*-polymerase, which remains stable at higher temperatures and can withstand the high-temperature cycle. This polymerase was isolated from *Thermus*

acquaticus, a bacterial species found near hot springs in Yellowstone National Park [12].

Another important molecular tool discovered in bacteria is the CRISPR/Cas system. CRISPR/Cas was originally discovered as a bacterial defense system against phages, the viruses that infect bacteria [13]. Since its discovery, it has been repurposed as a simple, cheap, and accurate genetic engineering method. It transformed biological research, enabling cheap, functional screens, control of gene expression, and cured people with sickle-cell disease [14, 15].

Bacteria also substantially negatively impact human society by causing life-threatening infections. Worldwide, bacterial infections were responsible for more than 9 million deaths in 2019 [16] (including more than a million deaths from *Mycobacterium tuberculosis* alone [17]). Worryingly, resistance to common antimicrobials is increasing, and in 2019, about 1.2 million deaths could be directly attributed to antimicrobial-resistant (AMR) bacteria [18]. The increased prevalence of bacterial AMR could make simple surgeries again life-threatening because of our inability to suppress bacteria in open wounds.

Contributing to the problem is the lack of new classes of antibiotics. While the early 1900s was a golden age for the discovery of new antimicrobial compounds, between 1962 and 2000, no new classes of antibiotic drugs were approved by the Food and Drug Administration (FDA) [19]. Because of the challenges involved in developing new antibiotics and the lack of business incentives, nearly all pharmaceutical companies have scaled down or even shut down their antibiotic research divisions [19]. To counter AMR, we thus cannot rely on new antibiotics alone.

Alternative strategies to combat AMR include preventing the spread of resistant bacteria and preventing the spread of genes that cause resistance. Bacteria are everywhere, and in our highly connected society, they can rapidly spread around, bringing along the genes responsible for resistance. Common travel routes are through the air, through contaminated water, through contaminated food, or through animals, among numerous other pathways [19]. They can further propagate resistance locally to other bacteria because they have genetic mechanisms to exchange genes if they are near each other [20]. Knowledge about these travel routes can inform on implementing measures to counter this spread.

To maximize bacteria's positive impacts and limit their negative impacts, we need to understand how bacteria function, evolve, and travel. Knowledge about bacterial biology can lead to the discovery of new useful molecular tools or give insights into preventing or treating bacterial infections.

1.2. The genome gives insight into an organism's evolutionary history and functional capabilities

An organism's genome is defined as its complete set of DNA molecules and provides a great starting point for nearly any study. It contains all of an organism's hereditary information and encodes instructions to manufacture almost every cell component. This information is encoded using the using the four nucleobases adenine (A), gua-

nine (G), cytosine (C), and thymine (T). Specific regions in the genome, genes, code for specific molecular machines. To build these, genes are transcribed to RNA; the RNA molecule is then further processed and translated into a protein, the major type of molecular machine in the cell [21]. Some RNA molecules, however, are already functional on their own and do not get translated [22]. Characterizing a genome thus gives us an overview of the major components an organism requires to function [23].

The genome is hereditary and thus is passed on from generation to generation. When a cell replicates, it copies its DNA and transfers it to the daughter cell. While DNA replication is highly accurate, it is not perfect, and a daughter cell might receive a mutated version of its parent's genome [23]. Sometimes, a mutation breaks an essential gene, killing the offspring, while other mutations have a neutral or even a positive effect on the offspring's ability to survive in its environment. In the latter two cases, the daughter cell will pass the acquired mutation to its offspring. Over time, mutations will accumulate, and by comparing the genomes of multiple individuals and inspecting who shares particular mutations with whom, we can estimate their relatedness and infer the evolutionary history of these individuals [23].

Understanding how genetic diversity arises and how genomes evolve is important for accurately inferring evolutionary relationships. This section will first explore common properties of bacterial genomes and sources of genetic diversity. We will explain how evolutionary relationships between genes can be used to infer evolutionary relationships between strains or species and additionally aid in transferring functional knowledge from one species to another. Finally, we will describe several important applications of the wealth of information encoded in the evolutionary histories, e.g., how it enables tracking species or strains across space and time.

Bacterial genomes are highly diverse, driven by horizontal gene transfer

Bacterial genomes typically comprise a single circular chromosome, which is, on average, 3.9 million base pairs (Mbp) in length [24]. Additionally, many bacteria harbor one or more *plasmids*: smaller, usually circular, genetic elements that can replicate independently of the chromosome. Plasmid lengths typically range from a few kilobases to hundreds of kilobases [25]. While bacterial genomes are thus much smaller than many eukaryotic genomes (the human genome is approximately 3.2 Gbp in length), they are densely packed with genes and have an average protein-coding gene density of 87% [26].

A surprising finding since the availability of multiple complete bacterial genomes is the extensive diversity in gene content within the same species [27]. For example, two *Escherichia coli* strains can share as little as 50% of their genes despite being considered the same species [28]. What defines them as *E. coli* is a set of *core* genes shared among all species members. Genes with more variable presence are called *accessory* genes. The combined core and accessory genes are defined as a species's *pangenome*. As a metaphor, core genes can be seen as a smartphone operating system, providing the basic needs to function. In contrast, accessory genes are installable apps, providing additional functionality for specific tasks and environments [29].

How does this diversity in gene content arise? Multiple evolutionary mechanisms could result in altered gene content, and they can be broadly classified into three groups: gene loss, gene gain through duplication, and gene gain through horizontal gene transfer [30]. Genes can be lost due to errors in replication. For example, a mutation could disrupt the promotor region of a gene, preventing the RNA polymerase from transcribing it. Errors in replication could also result in gene duplications [31]. Since the host cell now has two redundant copies, one copy could evolve into a gene with a new function because of reduced selective pressures and increased mutation rates.

However, the biggest driver of gene content diversity is the acquisition of novel genes through horizontal gene transfer (HGT) [32, 33]. Many bacteria have numerous capabilities to obtain DNA through other means than vertical descent [20]: 1) they can pick up extracellular DNA from the environment, a process called transformation (Figure 1.1a); 2) they can exchange DNA directly with their neighbors through a newly formed channel, a process called conjugation (Figure 1.1b); 3) they can get infected by phages, who inject a genetic payload into the bacterial cell, a process called transduction (Figure 1.1c); and 4) they can use specialized outer membrane vesicles to exchange genetic material, a process called vesiduction (Figure 1.1d). Exchange of genetic material generally occurs more frequently between closely related bacteria, though it can happen between distantly related species [34].

HGT enables adaptive evolution without being limited to starting from gene copies within its own genome. Instead of a slow, iterative mutational process to invent new functions, HGT enables the transfer of complete functional genes or even whole metabolic pathways, enabling quick adaptation to new niches or evolutionary pressures [32]. In most cases, the transferred genes retain the same function in the new host cell. However, newly acquired genes can assume new roles, e.g., because of redundancy in metabolic pathways, resulting in reduced selective pressure and increased mutation rate. In turn, this could generate genes with new functionality.

Pangenomes differ in size and reflect a species' lifestyle

A consequence of the resulting diversity in gene content is that a single genome rarely represents a species' total genetic repertoire. This raises questions about how many genomes are required to fully represent a species' pangenome and how pangenomes differ between species.

To gain insight into the growth and size of a species' pangenome, we typically plot a rarefaction curve, which plots pangenome growth as a function of total number of genomes analyzed. Pangenome sizes are frequently modeled as the cumulative sum of a power law function, i.e., the expected number of newly discovered gene families per genome is proportional to a function $N^{-\alpha}$, where N represents the number of genomes. When fit to the data, the value of the exponent α aids in classifying a species' pangenome: if $\alpha > 1$, the size of the pangenome will approach a constant as new genomes are added, and the pangenome is said to be *closed*. If $\alpha \leq 1$, the pangenome will grow indefinitely, and such pangenomes are said to be *open* (Figure 1.1e,f).

Species with closed pangenomes are typically niche specialists, have lower rates

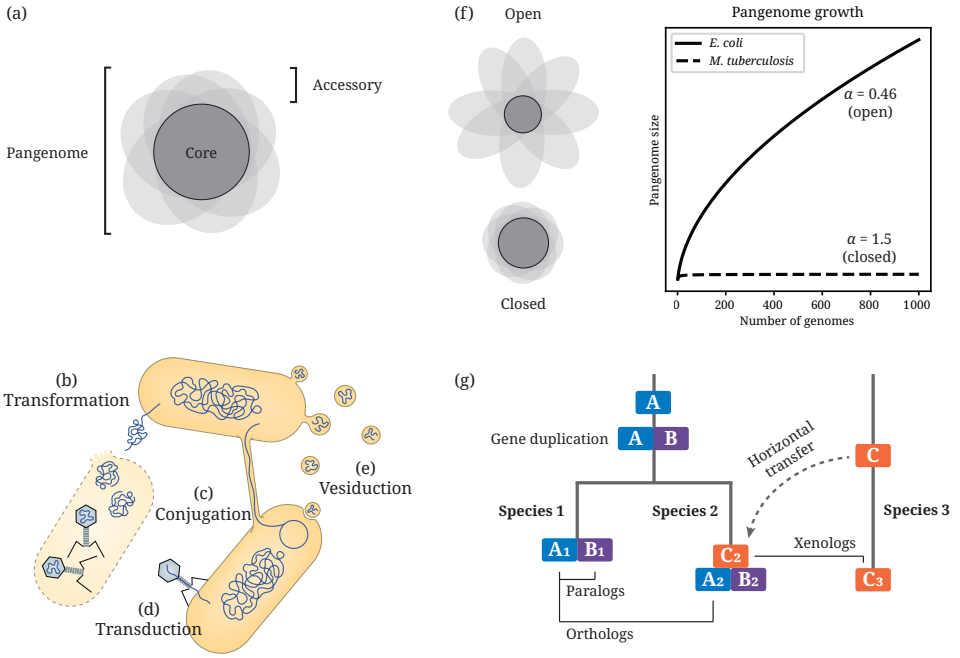

Figure 1.1: Bacteria readily exchange genetic material through other means than vertical descent, resulting in highly diverse genomes. **(a)** Illustration of bacterial pangenomes. Each oval wedge represents the gene set of a single strain. **(b-e)** Mechanisms of horizontal gene transfer include (a) transformation, (b) conjugation, (c) transduction, and (d) vesiduction. Republished from Brito *et al.* [20]. **(f)** Example illustrations and rarefaction curves for open (*E. coli*) and closed (*M. tuberculosis*) pangenomes. **(g)** The evolutionary relationships of three ancestral genes, A, B, and C, across three species.

of HGT, and have smaller population sizes [34]. An example of such a species is *M. tuberculosis*, the causative agent of tuberculosis, which mainly lives in human macrophages. A recent analysis of its pangenome, comprising 4,063 gene families, considered 3,116 (77%) gene families part of the core. The parameter α was estimated to be 1.5, indicating a closed pangenome corresponding to its niche lifestyle in human macrophages.

Species with open pangenomes, however, are often generalists that can adapt to many ecological niches [35]. They frequently interact with other members in diverse communities, have high rates of HGT, and have large population sizes [34]. For example, a large study analyzing 1,294 diverse *E. coli* genomes sampled from humans, non-human vertebrates, and environmental sources revealed a pangenome comprising 75,890 gene families, of which only 2,486 (3%) were considered part of the core [28]. When data was fit to the power law function, they obtained an $\alpha = 0.46$, indicative of an open pangenome, corresponding to *E. coli*'s diverse lifestyles and environments.

Evidence for high rates of HGT in species with open pangenomes includes the numerous *singleton* gene families observed in many species, i.e., gene families attributed to only one strain [30, 36]. This suggests that many acquired genes do not rise to

fixation in the population and are quickly lost. For example, the same *E. coli* study described above showed that *E. coli* singleton gene families are enriched for mobile genetic elements (MGEs), and no MGE family was found to be part of the core, suggesting that many singletons are only transiently present in a strain [28]. Elucidating the interplay between HGT, pangenome dynamics, and evolutionary processes like selection or drift is an active area of research [37, 35, 38]. An improved understanding of the processes that structure pangenomes will help us understand how bacteria evolve in response to new evolutionary pressures [39].

Orthologous genes are key to inferring the evolutionary histories of a species and transferring functional knowledge

By comparing genomes and analyzing who shares which mutations with whom, we infer the evolutionary histories of a set of strains. However, because of frequent HGT, some genes can have a different evolutionary history than the host. To accurately infer the evolutionary relationship between strains or species, we need to consider the evolutionary relationships between their individual genes.

A pair of genes are *homologs* when they derive from a shared ancestor. We can further distinguish between different evolutionary scenarios: 1) a pair of genes are *orthologs* if they are related through a speciation event (i.e., through vertical descent), 2) a pair of genes are *paralogs* if they are related through a gene duplication event, or 3) a pair of genes are *xenologs* if they are related through horizontal gene transfer (Figure 1.1g) [40]. Since orthologs are, by definition, genes inherited vertically, the combined set of orthologous genes between a set of species or strains best describes their evolutionary history.

Identifying orthologs between species or strains is additionally helpful for transferring functional knowledge from one species to another. According to the "ortholog-function conjecture", orthologous genes are most likely to have equivalent functions across species. In contrast, paralogous and xenologous genes are more likely to evolve new functions because of lower selective pressure [41]. Several large-scale efforts aim to catalog known orthologs between species and include predicted and experimentally validated functional annotations [42, 43]. Current gene prediction and annotation tools, including Prokka [44] or Bakta [45], cross-reference predicted genes with these databases. By characterizing the genome and its genes, we obtain an overview of a strain's functional capabilities without the need for experiments.

Genomes enable the tracking of strains across space and time

What information is present in the evolutionary history of a set of strains or species? One useful aspect frequently reflected in the evolutionary tree is the geographical distribution of a species. For example, if a particular strain acquired a mutation that allowed it to colonize a new environment, its descendants could form a new lineage in the evolutionary tree associated with that environment. These environment-lineage associations can help contextualize strains in newly collected samples, e.g., when a strain is genetically similar to strains associated with a known environment.

A well-known example that demonstrated the use of genomes to track strains and identify geographical links is the NextStrain platform during the SARS-CoV-2 pandemic [46]. Across the world, hospitals and other medical institutions collected samples from SARS-CoV-2 infected patients, sequenced the viral genome, and deposited sequences to public databases. The NextStrain platform inferred the evolutionary histories of these strains and linked lineages to geographical location, enabling near real-time detection of highly virulent variants and presenting clues on how the virus spread from country to country.

Other applications of the evolutionary history include uncovering the source of hospital outbreaks. We can track how bacteria spread around in a hospital by collecting samples from infected patients, sinks, door handles, counters, etc. Two genetically similar strains collected from different patients could be evidence of recent patient-to-patient transmission (e.g., when sharing a ward). Similarly, if several strains collected from infected patients are all genetically similar to strains collected from a sink, that could be evidence that the sink is a reservoir of pathogenic bacteria.

The accumulation of mutations and resulting genetic differences additionally reflect time. In current models for molecular evolution, a key assumption is that the mutation rate is fixed as long as the function does not change [47]. By quantifying genetic distances, we can thus estimate when two strains or species diverged.

Linking evolutionary events to points in time can help explain why species or strains evolved in a particular way. For example, the genus *Enterococcus* comprises a diverse set of bacterial species commonly found in the gut microbiome of most land-based insects, invertebrates, and mammals, including humans [48]. Two species, *Enterococcus faecium* and *Enterococcus faecalis*, have independently evolved to become multidrug-resistant, hospital-adapted pathogens, becoming one of the major causes of healthcare-associated bacterial infections [49, 50, 19]. To answer why specifically *Enterococcal* species have adapted so well to the hospital environment, Lebreton *et al.* characterized the genomes of a diverse set of *Enterococcus* species and reconstructed their evolutionary history [50]. *Enterococcus* diverged from its aquatic ancestors, and the estimated time of emergence, as reflected in the evolutionary tree, is concordant with the terrestrialization of animals. This transition from a water-based environment to a land-based environment was accompanied by increased hardening of the cell wall, enabling it to survive longer in harsh conditions on land. This hardened cell wall, which arose about 425 million years ago, enables it to resist many common disinfectants and antibiotics used in hospitals today [50].

1.3. Bacteria live in complex and diverse communities

Bacteria are often important members of diverse communities, which can contain other species of bacteria, archaea, viruses, or single-celled eukaryotic organisms [51]. Some microbial communities have important roles, such as driving earth's geochemical cycles [52] or contributing to human health and disease [6]. It is essential to study such communities holistically to understand the principles governing community assembly [53], elucidate species-species interactions [54], or infer community function.

Several technologies enable the profiling of microbial communities. This includes

metagenomics, which profiles the combined genomes of all community members; *metatranscriptomics*, which profiles the community-wide expressed genes; *metaproteomics*, which profiles protein abundances; and *metabolomics*, which profiles the metabolites present [55]. Reductions in the cost of these technologies and improved protocols have enabled unprecedented insights into numerous microbial communities' structure, composition, dynamics, and functional potential [51, 56].

One of the most widely studied microbial communities is the human gut microbiome [6, 57]. It comprises an estimated 1,000 different bacterial species (and additional archaea, fungi, and viruses) and harbors the same order of magnitude of bacterial cells as the total number of human cells [7, 58]. The community is essential to human health: It aids in nutrient breakdown and synthesis [59] and conditions the immune system to distinguish between pathogen and harmless commensal [60]. Community imbalances have been associated with several diseases, including inflammatory bowel disease (IBD) [61], obesity [62], or colorectal cancer [63] (further reviewed in refs. [64, 7, 56]).

The link between the microbiome and disease can occur at several levels [64, 56]. In some diseases, the absence, presence, or change in abundance of specific species can be directly linked to disease, e.g., in case of a *Clostridium difficile* infection [65]. In other diseases, the cause could be a specific species lineage, e.g., because of toxic gene products produced by particular strains [63]. More often, the disease will likely result from a complex interplay between the host immune system, specific members of the microbiome, and the metabolic state of the gut [64, 61].

While metagenomics, metatranscriptomics, metabolomics, and other profiling technologies each provide valuable and orthogonal insights into a community's state, in the remainder of this thesis, we will mainly focus on metagenomics when analyzing microbial communities. The insights gleaned from genomes, as discussed in previous sections, also apply to the analysis of metagenomes: they enable tracking strains over time and provide an overview of a community's functional potential.

1.4. Advances in sequencing technology enable high throughput characterization of whole (meta)genomes

An organism's genome or a community's metagenome is a rich source of information. However, given a sample specimen collected from soil, blood, urine, or feces, how do we determine the complete sequence of A, C, G, and Ts for the organisms in a sample?

This section will discuss common *DNA sequencing* strategies to characterize genomes in a sample. We first describe methods for preparing and extracting the DNA molecules of interest, e.g., from a single or all community members. We will introduce the three commonly used sequencing platforms and discuss their advantages and disadvantages. Finally, we will discuss algorithmic approaches to infer the sample genome(s) from sequenced *reads*, first focusing on inferring a single strain's genome and later on how these methods can be adapted to profile whole communities.

Preparing DNA for sequencing

Since a sample specimen could harbor multiple organisms and strains, we typically streak it out on a petri dish, allowing individual bacterial colonies to grow in *culture* (Figure 1.2a). This enables picking and isolating specific colonies (representing the clonal expansion of a single strain), extracting DNA from these isolates, and putting them on a sequencing instrument. Sequencing the entire genome of a single isolate from culture is called *whole genome sequencing* (WGS).

Culturing has long been the standard approach to characterize bacterial strains in a sample [66]. Technological improvements in the past decade have enabled culturing of nearly all human-associated bacterial species [67, 68, 69, 70]. However, many other species, especially those from non-human-associated communities, remain challenging to grow under laboratory conditions (if at all). Culturing is also laborious and low throughput, requiring manual picking and isolating of specific colonies. Another limitation is the resulting biased view of genetic diversity in a sample because typically only a subset of colonies are isolated and sequenced, and because of the potential for evolution within a culture [66].

Instead, *whole metagenome sequencing* (WMS) is a culture-free alternative approach for characterizing strains [51]. For WMS, DNA is directly extracted and sequenced from a sample specimen. The sequenced DNA thus represents fragments from all sample members, not just a single strain. WMS offers a more complete and less biased view of the sample's genetic diversity and enables the characterization of genomes from unculturable species.

When DNA has been extracted, either from an isolate or directly from the sample, it is further processed before sequencing. The DNA is fragmented into smaller pieces since no sequencing platform can sequence complete genomes at once (Figure 1.2b). A sequencer can read these smaller fragments, and since these *reads* represent smaller fragments of the sequenced genome, we rely on algorithms to infer complete sample genomes. The accuracy and completeness of these reconstructions depend on the sequencing technology used, as each has limitations regarding fragment lengths, throughput, and sequencing error profiles.

Current sequencing technology platforms

Today's most common sequencing platforms include Illumina, Pacific Biosciences (PacBio), and Oxford Nanopore Technologies (ONT) [71, 72]. Illumina's platform can sequence short (100-250 bp) fragments with high accuracy (an error rate of $< 0.1\%$) and high throughput. Its larger machines can sequence billions of reads per run. This enables amortizing sequencing costs over many samples by pooling them on a single flow cell, making it one of the most cost-effective sequencing solutions. A disadvantage of Illumina's technology is its short read length, which provides limited genomic context and makes it harder to infer what piece of the genome it represents.

The last decade has seen increased adoption of "long read" sequencing platforms, including PacBio and ONT [73]. A major benefit of these platforms is their ability to sequence much longer fragments, with average read lengths of approximately 15 kbp and with ONT up to hundreds of kbp [74]. Longer reads provide more genomic con-

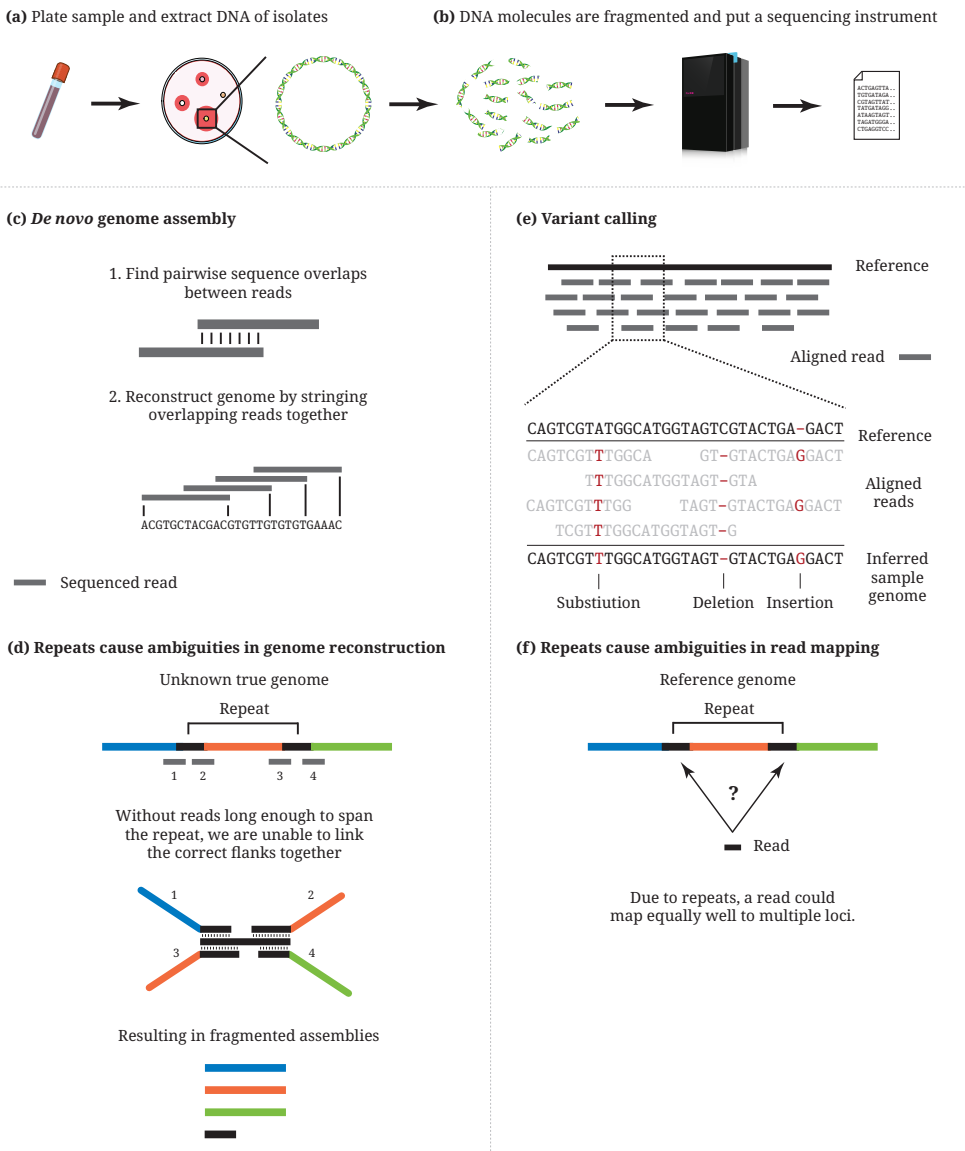

Figure 1.2: Typical whole genome sequencing workflows and algorithmic approaches to infer sample genomes. **(a)** Sample specimens are plated on a petri dish. We isolate individual colonies and extract their DNA. **(b)** Since current sequencing instruments cannot read full chromosomes, the isolated DNA is fragmented before being placed on a sequencing instrument. **(c)** Intituition behind *de novo* assembly. We first identify all-vs-all pairwise overlaps between reads (1). We reconstruct the original genome by iteratively stringing together overlapping reads (2). **(d)** Repeats in the true genome (black lines) prevent complete genome reconstruction, resulting in fragmented assemblies with multiple *contigs*. **(e)** Genome inference through variant calling. Reads are aligned to a previously assembled reference genome. Alignment pileups are analyzed to identify genetic variants, such as substitutions, insertions, or deletions. **(f)** Repeats in the reference are hard to characterize without reads spanning the repeat.

text, making reconstructing complete genomes easier. Analogously to a puzzle, a puzzle with larger pieces is easier to assemble than one with smaller pieces. Long-read sequencing technologies have greatly improved their accuracy in recent years, with PacBio's high-fidelity reads achieving an error rate of 0.1% (on par with Illumina), while ONT's recent nanopore chemistry (R10.4) achieves error rates of 0.5-1% [75]. A disadvantage of long-read sequencing platforms is their relatively low throughput, higher cost, and higher DNA input requirements.

An additional unique feature of ONT's platform is their instruments' range of physical sizes. Most sequencing instruments, including those from Illumina and PacBio, are about the size of an office printer or a large refrigerator. ONT offers similar-sized instruments for higher throughput settings (e.g., large sequencing centers), but additionally offers a handheld device sized like a smartphone. Combined with a relatively simple protocol for DNA extraction, this enables direct sequencing of samples at location, which aids real-time genomic epidemiology during an outbreak, e.g., during the Zika virus epidemic in South America [76, 77].

Reconstructing an isolate genome from scratch using read data only

One common strategy to reconstruct a strain's genome in the case of a sequenced isolate is *de novo* assembly. *De novo* assembly aims to reconstruct genomes from read data only, without aid from a previously completed reference genome. A typical workflow first involves identifying pairwise sequence overlaps between reads. If the suffix of one read overlaps the prefix of another, the latter read can extend the first read's genome sequence. By repeatedly extending the reconstructed sequence with reads that overlap with the end of the sequence, we ultimately assemble the entire genome (Figure 1.2c) [78].

In practice, however, several challenges prevent the complete reconstruction of a genome. One important challenge is repetitive genome content, e.g., a gene present twice. When trying to extend the sequence with an overlapping read during assembly, a repeat results in multiple options for sequence extension: either the flanking sequence of the first copy of the repeat or the flanking sequence of the second copy. If the length of reads is insufficient to span the entire repeat, linking each copy's correct left and right flank will be impossible (Figure 1.2d). This halts the assembly process and results in fragmented assemblies, with the genome split into multiple *contigs*.

Another challenge is the identification of pairwise overlaps between reads. Reads can contain sequencing errors, which means the overlap detection algorithm should be able to identify inexact overlaps. There is a trade-off between being able to detect overlaps in the presence of sequencing error and introducing falsely detected overlaps because of inexact overlap detection. Falsely detected overlaps can lead to ambiguity in the assembly, similar to repetitive genome content, or result in misassemblies.

Modern genome assemblers typically implement these ideas using one of two major computational models: 1) using the *overlap-layout-consensus* (OLC) approach, or 2) using a De Bruijn graph (DBG) [78]. In the first model, pairwise overlaps between reads are computed and explicitly stored in a *string graph* [79]. The graph is cleaned so the remaining paths represent the reconstructed genome with high probability [80].

To construct a DBG, explicitly computing pairwise overlaps between reads is unnecessary. In a DBG, nodes are k-mers, i.e., sequences of a fixed length k, and directed edges connect nodes where the $k-1$ suffix of one node matches the $k-1$ prefix of another. By extracting k-mers from reads and adding nodes and edges to the graph where necessary, we implicitly obtain overlaps between reads because of shared k-mers. After cleaning the graph, removing k-mers likely originating from sequencing error, the remaining paths in the graph represent the genome with high probability [81, 82]. We refer to recent reviews for an in-depth discussion of current tools and practices [78, 83]

Reference-assisted genome inference using read alignment

Instead of reconstructing an isolate genome *de novo*, we can map and compare sequenced reads to a previously characterized *reference* genome. For each read, we search for a locus in the reference that likely represents its origin by looking for regions with high sequence similarity to the read. However, the strain is unlikely to be identical to the reference genome, and the sequenced reads could contain evidence for alleles that differ from the reference.

To infer these differences, each read is *aligned* to the reference genome. Sequence alignment is a form of inexact string matching, which aims to pair nucleotides in the read and the reference that likely share an evolutionary origin, allowing for mismatches, newly inserted nucleotides, and deletions [84, 85, 86]. When all reads are aligned, the resulting *alignment pile-ups* along the reference can be analyzed to infer where the sample strain differs from the reference, a process called *variant calling* (Figure 1.2e) [87, 88]. We can reconstruct the genome of the sample strain by taking the majority allele present among reads at sites with identified differences. A more in-depth discussion of the methods and best practices are reviewed in refs. [89, 88].

A major advantage of variant calling-based genome inference is that the reference will typically be a fully assembled genome accompanied by extensive biological annotations. Identified variants can thus be directly analyzed in their biological context. Other advantages include the lower computational requirements compared to assembly workflows and the simplicity of comparing genetic variation among multiple samples, which we will discuss further in a later section.

While the reference genome might be a genome with repetitive content resolved, repeats still cause challenges for variant calling. For example, when a gene is present twice in a reference, reads from the sample genome homologous to those genes can not be unambiguously mapped (Figure 1.2f). Assigning any identified genetic variation to the correct copy will be impossible.

An additional challenge includes the introduction of *reference bias*. Using a reference limits the identification of variants to genomic content shared with the reference. For example, the reference might lack genes in the sample strain, preventing those reads from aligning. Additionally, if the sample strain diverged substantially from the reference, reads might not align accurately because of low sequence similarity. These issues lower the accuracy of variant calls, especially in bacterial species with open pangenomes, because of extensive diversity in gene content [90, 91].

Profiling metagenomes

The methods discussed so far assume that reads originate from a single isolate. However, sequenced reads could originate from multiple species or strains in metagenomic data. Specialized tools are required to profile genomic content in a metagenome accurately.

Taxonomic profiling tools, including KRAKEN [92] and METAPHLAN [93], are typically reference-assisted approaches that report the species-level composition of metagenomic samples. KRAKEN analyzes the k-mer composition of a sample to the k-mer profiles of references in its database to estimate taxa abundances [92]. METAPHLAN estimates abundances by analyzing read alignments to a set of phylogenetically informative *marker genes*, precomputed by the tool's authors [93]. While these tools enable valuable insights into the high-level composition and dynamics of metagenomic samples, they offer limited insight into the genomic content of sample strains.

Instead of taxonomic profiling, an alternative method to offer insight into sample genomic content is the *de novo* assembly of metagenomes [51]. Similar to the assembly of isolates, these approaches aim to reconstruct all sample genomes from scratch using the sequenced reads.

However, assemblies from metagenomic data are often highly fragmented and incomplete because of three main factors [51, 94]. First, repetitive content is an even larger problem in metagenomic assembly compared to assembly of isolates. Ambiguity in the assembly graph can arise because of genes conserved between species or even across the bacterial kingdom [95]. Another cause of ambiguity is the presence of multiple strains of the same species, who share their core genes but have different accessory genes [94]. Second, the uneven abundance of species in metagenomes makes it more challenging to distinguish between sequencing errors and low-abundance species. Third, an additional "contig binning" step is required to group contigs from the same species. However, this process frequently omits mobile genetic elements, resulting in incomplete views of a genome [20].

Both taxonomic profiling and assembly approaches offer limited or incomplete strain-level insights. Distinguishing between strains is important because of strain-specific biological differences such as antibiotic resistance, pathogenicity, or metabolic capabilities [7, 96]. The current inability to accurately characterize metagenomes at the strain level is a major barrier to understanding species populations, ecologies, transmission patterns, and their role in health and disease [51, 7, 96].

1.5. Biological sequence alignment computes which residues likely have shared evolutionary origin

Genomes evolve and change over time, but given two DNA sequences, how do we determine what has changed? What nucleotides in one sequence have been substituted in the other sequence? What nucleotides are newly inserted or deleted? The identification of genetic changes is at the heart of computational genomics. It enables many important analyses, including computing evolutionary relationships by quanti-

fying differences, inferring a sample genome from read alignments (as described in the previous section), and elucidating molecular mechanisms underlying a phenotype by linking specific variants to the phenotype.

This section will introduce algorithms for computing biological sequence alignments, which enable the identification of common similarities that share an evolutionary origin. An alignment is an ordered arrangement of two sequences (the order of nucleotides in both sequences should remain the same), where nucleotides in one sequence are paired with nucleotides in the other or with a newly introduced *gap*. Paired identical nucleotides are called *matches*; if they are different, we call them *mismatches*. Nucleotides paired with a gap symbol are called *indels*. For example, we find many shared nucleotides by rearranging two seemingly different DNA sequences, ATGCTTA and TGCAATTA (Figure 1.3a). Intuitively, good alignments maximize matching nucleotides between sequences and minimize mismatches, insertions, and deletions [97].

We will first introduce common alignment cost models to quantify good and bad alignments. Next, we describe popular dynamic programming algorithms to compute alignments that minimize the alignment cost.

Alignment cost models to quantify plausible alignments

To obtain biologically plausible alignments, an alignment cost model should reflect the biology of DNA (or proteins if comparing amino acid sequences). The three most commonly used cost models, in order of biological accuracy, are edit distance, linear gap penalties, and affine gap penalties [97, 84]. The edit distance model is the simplest model, in which mismatches and indels have unit costs, while matches have zero cost (Figure 1.3b). The total cost of an alignment would be $C = N_x + N_g$, with N_x and N_g representing the number of mismatches and indels, respectively.

Mismatches and indels, however, can occur at different rates, and the linear gap penalty model addresses this by penalizing these events differently [98]. In this model, matches, mismatches, and indels have different associated costs (Figure 1.3c). The total cost of an alignment can be computed as $C = N_m\Delta_m + N_x\Delta_x + N_g\Delta_g$, with N_m the number of matches and $\Delta_m, \Delta_x, \Delta_g$ the cost of a match, mismatch, and an indel, respectively. One disadvantage of this model is the linear weighting of consecutive indel positions. Because the insertion or deletion of multiple consecutive nucleotides frequently occurs as a single biological event, long indels can get excessively penalized because each gap position increases the cost [84].

To address this, the alignment cost model would ideally include an indel cost function $g(l)$ to consider an indel's length. Supporting any arbitrary function $g(l)$ would complicate the design of an algorithm to compute alignments [84]. Instead, in *affine gap penalty* models, $g(l)$ is required to be in the form of $g(l) = \Delta_o + l \cdot \Delta_g$, i.e., a linear model where opening a new indel is penalized differently than extending an existing indel [99] (Figure 1.3d). In this equation, Δ_o represents the cost of opening a new indel, and Δ_g is the cost of extending an existing indel. The total cost of an alignment would then be $C = N_m\Delta_m + N_x\Delta_x + N_g\Delta_g + N_o\Delta_o$, where N_o represents the number of distinct indels, N_g the total number of indel positions. Affine gap penalty models are the most

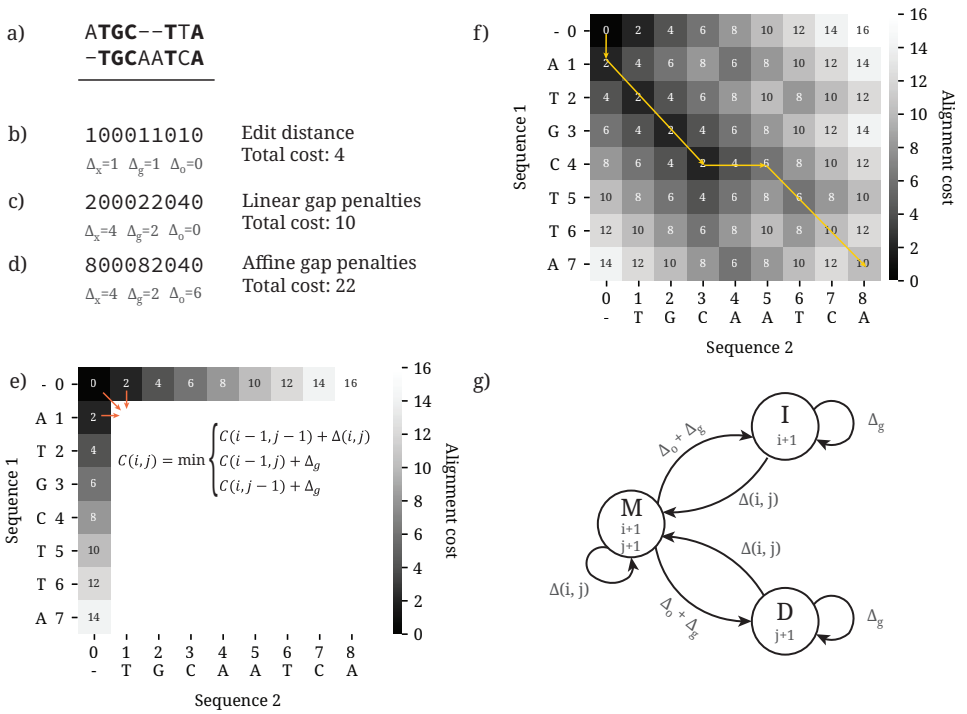

Figure 1.3: Concepts of biological sequence alignment. **(a)** Example alignment of two sequences, with matching nucleotides in bold. **(b-d)** Quantifying the alignment quality using (b) edit distance, (c) linear gap penalties, and (d) affine gap penalties. Δ_x, Δ_g, and Δ_o represent the alignment cost of a mismatch, gap extension, and newly opened gap, respectively. **(e)** Example initialization of the alignment dynamic programming matrix. Each remaining cell is computed by considering the minimum alignment cost of three predecessor cells (orange arrows) plus the additional cost of the alignment extension. **(f)** The completed dynamic programming matrix. We obtain the final alignment (yellow arrows) by tracing which cell is computed from which predecessor. **(g)** A finite state machine diagram representing alignment using affine gap penalties. Circles represent the (mis)match, insertion, and deletion states, which adjust sequence positions i or j when entered. Arrows represent state transitions, labeled with the alignment cost.

commonly used cost models for DNA sequence alignment, as they offer the best balance between computational tractability and biologically accurate alignments [84, 85, 86, 100].

Note that the total alignment cost can be written as a weighted sum of individual alignment events in all cost models. This property enables the design of efficient algorithms to compute accurate alignments.

Computing optimal alignments with dynamic programming

A *dynamic programming*-based algorithm is a common method to compute alignments that minimize the alignment cost. For clarity of presentation, we will first introduce the algorithm using the edit distance or linear gap penalty model. The algorithm is

the same for both models since the edit distance model is a special case of the linear gap penalty model, where $\Delta_m = 0$, $\Delta_x = 1$, and $\Delta_g = 1$. The extension to the affine gap penalty model will be discussed afterward.

The core idea behind the dynamic programming algorithm is to compute the alignment iteratively, reusing solutions for shorter subsequences computed in previous iterations. To see how this works, note that the total costs for each model described above are a weighted sum of (mis)matches and indels. Suppose we know the minimum cost for a previously computed alignment of shorter subsequences. In that case, we can compute the minimum cost of an extension of that alignment by adding the cost of the extension. For example, if extending an alignment with a match, the minimum cost of this extended alignment would be the minimum cost of the previous alignment plus Δ_m; if extending with a mismatch, we would add Δ_x; and if extending with a gap, we would add Δ_g.

Specifically, let $R = r_1 r_2 \dots r_n$ and $Q = q_1 q_2 \dots q_m$ be two DNA sequences to align. Let i and j refer to positions in R and Q, respectively, and we define the subsequence of R up to the ith position as $R[1 \dots i] = r_1 r_2 \dots r_i$, and similarly the subsequence of Q up to the jth position as $Q[1 \dots j] = q_1 q_2 \dots q_j$. We denote $C(i,j)$ as the minimum cost to align $R[1 \dots i]$ and $Q[1 \dots j]$. $C(i,j)$ can be computed recursively by considering three possible scenarios (Figure 1.3e): first, r_i could be aligned to q_j, in which case the alignment cost $C(i,j) = C(i-1, j-1) + \Delta(i,j)$. Here, $\Delta(i,j)$ is a function that returns match cost Δ_m if $r_i = q_j$, and the mismatch cost Δ_x otherwise. Second, r_i could be aligned to a gap, in which $C(i,j) = C(i-1,j) + \Delta_g$. Finally, q_j could be aligned to a gap, in which $C(i,j) = C(i,j-1) + \Delta_g$. The minimum alignment cost $C(i,j)$ would be the minimum of these three cases:

$$C(i,j) = \min \begin{cases} C(i-1, j-1) + \Delta(i,j) & \text{(Mis)match,} \\ C(i-1,j) + \Delta_g & \text{Deletion,} \\ C(i,j-1) + \Delta_g & \text{Insertion.} \end{cases} \tag{1.1}$$

To be able to compute $C(i,j)$ for all possible values, we need to handle the *base cases* separately, i.e., the cases where either $i = 0$ or $j = 0$. Note that the definition of $C(i,j)$ depends on undefined cases when $i = 0$ or $j = 0$ ($i - 1$ or $j - 1$ would be negative). To compute $C(0,j)$ and $C(i,0)$, we observe that those cases correspond to prefixing either Q or R with gaps, and thus we define $C(0,j) = j\Delta_g$ and $C(i,0) = i\Delta_g$. The remainder of $C(i,j)$ cells can be computed by progressing row-by-row and column-by-column. After completing the matrix, the final minimum alignment cost between R and Q is the bottom right cell $C(n,m)$. The alignment itself can be inferred by repeatedly tracing which cell derived from which other cell until it reaches $C(0,0)$ (Figure 1.3f).

To support affine gap penalties we need to keep track of opened gaps. To achieve this, we store three separate costs for each pair (i,j): $M(i,j)$ represents the minimum alignment cost of $R[1 \dots i]$ and $Q[1 \dots j]$ in the case where r_i is aligned to q_j; $D(i,j)$ represents the minimum alignment cost in the case where r_i is aligned to a gap (i.e., a deletion in Q with respect to R); and $I(i,j)$ represents the minimum alignment cost in the case where q_j is aligned to a gap (i.e., an insertion in Q with respect to R). Using separate variables for the alignment costs ending in the (mis)match, deletion, and

insertion states, we can define separate recursive functions considering the different costs of opening or extending an indel. The recurrence relations then become [84, 99]:

$$\begin{aligned} M(i,j) &= \min \begin{cases} M(i-1,j-1) + \Delta(i,j) & \text{(Mis)match,} \\ D(i-1,j-1) + \Delta(i,j) & \text{Close deletion,} \\ I(i-1,j-1) + \Delta(i,j) & \text{Close insertion,} \end{cases} \\ D(i,j) &= \min \begin{cases} M(i-1,j) + \Delta_o + \Delta_g & \text{Open deletion,} \\ D(i-1,j) + \Delta_g & \text{Extend deletion,} \end{cases} \\ I(i,j) &= \min \begin{cases} M(i,j-1) + \Delta_o + \Delta_g & \text{Open insertion,} \\ I(i,j-1) + \Delta_g & \text{Extend insertion.} \end{cases} \end{aligned} \tag{1.2}$$

This set of recurrence relations can be interpreted as a state machine, with M, D, and I the different states, and with varying costs to stay in a state or transition to another (Figure 1.3g) [84].

1.6. Computational methods to characterize genetic variation genome and pangenome-wide

Through sequence alignments, we obtain similarities and differences between biological sequences. However, the algorithms to compute these alignments assume sequences to be collinear, i.e., the shared nucleotides between sequences generally retain the same order save for gaps and mismatches. This assumption is often violated when comparing whole bacterial genomes because of inversions, mobile genes, recombination, and other structural variants [101, 102]. Since the substitution, insertion, and deletion alignment operations do not accurately model these larger changes, additional strategies are required to compare complete genomes.

This section will describe common approaches to compare genome and pangenome-wide genetic variation among two or more bacterial strains. We will discuss methods to compare strains directly from sequenced reads, e.g., by aligning reads to a reference genome, and strategies to compare *de novo* assembled genomes. Finally, we will discuss approaches to characterizing extensive diversity gene content in bacterial genomes, i.e., tools to characterize and analyze bacterial pangenomes.

Direct comparison of whole genome assemblies

To compute whole genome alignments, in the presence of large structural variants and rearrangements, we first identify pairwise homologous (and locally collinear) regions between genomes [101, 103, 104]. This is usually done by first identifying 'seeds,' (short) exact matches between genomes, and chaining seeds in close proximity to each other to obtain larger collinear blocks with high sequence similarity. Tools to compute such homology maps include MUMMER [101, 105], which uses "maximal unique matches" between genomes as seeds, and MashMap [106], which uses "minmers" as

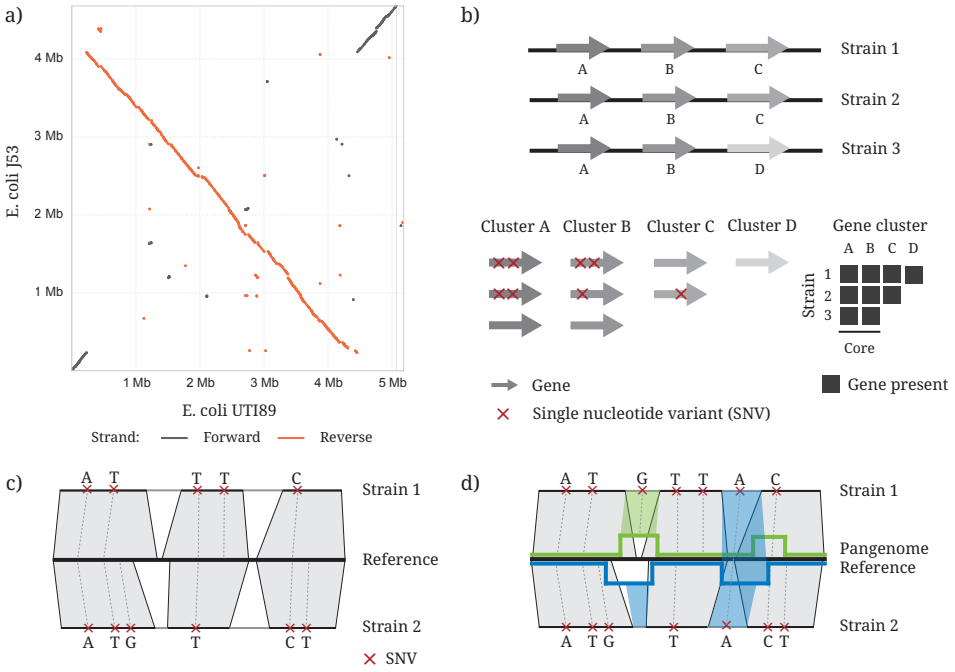

Figure 1.4: Illustration of several genome comparison methods. **(a)** Whole genome alignment between two *E. coli* assemblies visualized as a dot plot. Each line represents a locally collinear aligned region between *E. coli* UTI89 (x-axis) and *E. coli* J53 (y-axis), either on the forward (dark grey) or reverse strand (orange). **(b)** (Pan)genome comparisons through shared genes. Genes (arrows) among a set of input genomes are clustered. Nucleotide-level variation (red crosses) per gene cluster can be obtained through an MSA. The gene presence/absence matrix (black squares) enables separating core from accessory genes. **(c)** Genome comparisons through an intermediate reference genome. Read alignments to a reference enable inference of shared homologous genome content between a strain and the reference (shaded areas). Genetic variants (red crosses and dashed lines) are described with respect to the reference, enabling easy comparisons of variants. **(d)** Genome comparisons through a pangenome reference graph. Multiple reference genomes are combined in a graph (black, blue, and green lines), enabling the identification of variants in a larger fraction of strains' genomes compared to a single reference (blue and green shaded areas).

seeds. Each identified locally collinear block is input into a pairwise sequence alignment algorithm to obtain nucleotide-level differences between genomes.

Besides small genetic variants, homology maps reveal much insight into structural differences between genomes, for example, when visualized as a *dot plot* (Figure 1.4a). In such a plot, an (anti-)diagonal line represents a locally collinear block with high sequence similarity. Horizontal or vertical shifts between lines indicate the presence of large indels. Lines going diagonally from the bottom left to the top right indicate homology between forward strands; lines going anti-diagonally from the top left to the bottom right indicate homology between opposite strands and represent inversions. For example, *E. coli* J53 has much of its genome inverted with respect to *E. coli* UTI89, as indicated by the many lines on the antidiagonal in Figure 1.4a.

While whole genome alignment enables comparison of all shared genome con-

tent between two strains, it is harder to scale comparisons between multiple genomes. Comparing multiple genomes would require all-vs-all whole genome alignments, without an obvious method to identify which regions are homologous between multiple genomes. This hinders analyses of genetic variation at the population level.

Gene-centered pangenomics and comparisons of genomes

An alternative approach to compare genome assemblies that scales to thousands of diverse strains is to characterize and compare their shared genes. Comparing gene content is at the heart of nearly all bacterial pangenome analysis tools [107, 108, 109, 110, 111, 112, 91]. These tools typically take annotated genomes as input and cluster genes based on sequence similarity and gene neighborhoods (Figure 1.4b). Nucleotide-level differences per gene cluster can be obtained by computing an MSA of gene sequences. Comparing genes enables near-complete comparisons of genomes since bacteria have high coding densities.

Gene presence/absence patterns can be further analyzed to separate core from accessory genes. Genes present in all genomes are part of the core (genes A and B in Figure 1.4b), while genes with variable presence are accessory (genes C and D in Figure 1.4b).

An important application of gene-centered pangenome analyses is investigating gene gain and loss events along the evolutionary history. In earlier sections, we discussed that the combined set of orthologous genes encodes the organisms's evolutionary history. Most bacterial pangenome analysis tools include strategies to distinguish between orthologous and paralogous genes post-clustering. To obtain the evolutionary history of all input strains, we thus input the combined MSA of all orthologous core genes to a phylogenetic tree construction tool [113, 114]. Patterns of accessory gene presence and absence can then be analyzed in the context of the strains' phylogeny, e.g., to infer gene gain or loss events in specific lineages.

While these tools enable valuable insights into the evolution of an organism and its genes, the focus on genes also has a significant downside: it prevents characterizing variation in intergenic regions. Variation in intergenic regions is important since it could substantially impact phenotypes, e.g., by changing a promotor, affecting gene expression [115]. Another downside is the reliance on genome assemblies and accurate gene annotations. Genome assemblies, especially those constructed from short reads, can be fragmented and incomplete. Automatic gene prediction tools are imperfect and can miss or misclassify genes or return inaccurate start and end coordinates [116]. Errors in the assembly or annotations impact downstream analysis of pangenome growth, classification of core vs. accessory, and distinguishing between orthologous vs. paralogous genes [39].

Reference-assisted comparison of genetic variation

Comparing variant calls with respect to a reference genome allows for comparing genomes without needing *de novo* assembly [87]. When reads align to a particular locus in the reference, we assume the read's genomic origin is homologous to that

locus (Figure 1.4c; grey-shaded areas). If we run a variant calling workflow for multiple samples, and when reads from different samples align to the same locus in the reference, we use the reference as a proxy to infer homology between sample strains. By assessing inferred alleles in each sample at the same reference positions, obtained variant calls can be easily compared (Figure 1.4c; red crosses).

The simplicity of scaling comparisons to multiple samples is a major benefit of variant calling workflows. Variant calls can be obtained independently for each sample, and since each variant is described with respect to the reference, they can be easily compared. Variant calling workflows typically require few computational resources, enabling the analysis of large datasets.

However, this method's downside is that it only allows comparisons of variant calls in genome content shared with the reference (Figure 1.4c; unshaded areas in strain 1 and 2). If the reference does not contain specific genes in the sample strains, it will be impossible to compare variation within them since the reads from those genes have no place to align. Reads could also misalign to a locus with partial sequence similarity, resulting in false positive variant calls. The more distant the reference from the sample strain, the lower the accuracy of variant calls [90].

Another consequence of the inability to characterize genes not present in the reference is that obtaining a complete picture of gene content diversity among a set of strains is challenging. Reference-assisted genome comparisons are, therefore, unsuitable for the characterization of bacterial pangenomes.

Comparing genetic variation using pangenome-reference graphs

In recent years, there have been increased efforts to extend variant calling to allow for multiple reference genomes [117, 118]. These approaches promise to combine the benefits of variant calling (simple workflows, the ability to obtain variants for each sample independently, and a common coordinate system for comparison) with the benefits of pangenome analysis tools (characterizing a species' total genetic diversity).

References are combined into a *pangenome reference graph* (Figure 1.4d) to support variant calling with respect to multiple references. Pangenome graphs compactly represent multiple references by grouping shared genomic content while also representing each genome's unique content. Generally, nodes represent the sequence of a piece of the genome, either unique or shared with other genomes, while edges connect nodes representing adjacent blocks of genomic content in at least of the genomes [117, 118]. Each input genome can be reconstructed by traversing a specific path through the pangenome graph.

Including multiple references in a pangenome graph increases read alignment accuracy [119] and enables comparison of genetic variation across a larger fraction of sample strain's genomes. For example, the pangenome reference in Figure 1.4d enables the detection of a shared A allele in strains 1 and 2 because of shared homology to the blue reference. Variation in those regions of the genome was previously missed when using a single reference (Figure 1.4c).

Algorithms to construct pangenome reference graphs are still an active area of research. Two recently published approaches mainly focused on eukaryotic (human)

pangenomes include Minigraph-Cactus [120] and the pangenome graph builder (PGGB) [121]. These tools construct pangenome graphs based on whole genome alignments and can represent large structural changes between genomes as well as small nucleotide-level variations. While these pipelines could be used to construct bacterial pangenomes, there has not been any evaluation of these tools' ability to accurately identify homology relations between genomes in the presence of much higher recombination rates and gene content diversity, as observed in many bacterial species.

A pangenome reference approach focusing on bacterial genomes is Pandora [91]. Pandora constructs reference graphs per gene and thus relies on other tools to define what genes are homologous among reference genomes. It enables calling variants directly from sequenced reads using its own read-mapping approach. However, its read-mapping approach is relatively simplistic: it cannot accurately map reads spanning gene boundaries and does not handle situations where reads map (partially) to multiple genes. Pandora also inherits the issues of other gene-centered pangenome analysis tools discussed earlier by relying on gene annotations.

Panaroo is another graph-based and gene-centered bacterial pangenome analysis tool [111]. In the Panaroo graph, nodes represent gene families, and edges connect nodes representing adjacent genes in at least one genome. Panaroo includes several algorithms that alleviate some of the issues with inaccurate gene annotations. It uses the graph topology to recover missing genes, detect misclassified genes, and fix incorrect gene clusterings. The resulting graph gives valuable insights into structural rearrangements between genomes. However, the graph does not include sequence information, making it unsuitable for read alignment or describing variants and thus unsuitable as a pangenome *reference* graph.

To our knowledge, none of the existing tools can construct bacterial pangenome graphs that include both genes and intergenic regions, construct graphs that can serve as a reference for variant calling, and work with many bacterial species' high recombination rates and gene content diversity. This hinders the identification and description of variants across large datasets of diverse strains and is a barrier to improved understanding of genetic variation and its impact on phenotypes.

1.7. Thesis contributions and outline

Comparing genomes is central to bacterial genomics. It enables tracking strains, understanding their evolution, and elucidating the molecular mechanisms underlying a phenotype. **The central question of this thesis is how we can improve our ability to characterize genetic variation in bacteria, considering the extensive strain-level diversity among many species**. In the following chapters, we will introduce new tools and algorithms to tackle this problem and use these tools to gain new insights into strain-level dynamics.

In Chapter 2, we first introduce a new algorithm for partial order alignment (POA). POA is a common multiple sequence alignment approach with many applications in genome assembly, RNA isoform inference, variant calling, and pangenomics. Our algorithm exploits exact matches between a query sequence and the POA graph, reduc-

ing runtime and memory usage and enabling the construction of megabase-length alignments, which was not previously possible.

Chapter 3 introduces the Strain Genome Explorer (StrainGE) suite, a new set of tools to characterize strain-level bacterial genetic variation using WMS data. StrainGE's pipeline comprises two main components: first, it identifies representative reference genomes for each strain in a sample, enabling the detection of same-species strain mixtures. Second, StrainGE further characterizes strain variation by analyzing read alignments and calling variants compared to the reported references. StrainGE was designed to operate at coverages as low as 0.5x, enabling the detection and characterization of previously unnoticed strains.

Chapter 4 covers a large, multi-institute, year-long study investigating the link between gut microbiota and recurrent urinary tract infections (UTIs). For this study, we collected monthly stool, urine, and blood samples from women with and without a history of UTIs. Additional samples were collected during and after a UTI. Using StrainGE, we gained detailed insights into the *E. coli* strain-level diversity in the gut and bladder. Despite divergent outcomes, we found that both women with rUTI and controls share similar *E. coli* dynamics.

We will conclude this thesis by discussing the remaining gaps in analyzing bacterial (pan)genomes and highlighting promising new technologies that will transform our ability to analyze bacterial genetic diversity. We believe these new technologies and computational tools will help counter the growing antibiotic resistance epidemic and help understand the role of bacteria in health and disease.

References

1. Jannasch HW and Mottl MJ. Geomicrobiology of Deep-Sea Hydrothermal Vents. Science 1985 Aug; 229:717–25
2. Meyer-Dombard DR, Shock EL, and Amend JP. Archaeal and bacterial communities in geochemically diverse hot springs of Yellowstone National Park, USA. en. Geobiology 2005; 3:211–27
3. Fierer N and Jackson RB. The diversity and biogeography of soil bacterial communities. Proceedings of the National Academy of Sciences 2006 Jan; 103. Publisher: Proceedings of the National Academy of Sciences:626–31
4. Innamorati KA, Earl JP, Aggarwal SD, Ehrlich GD, and Hiller NL. The Bacterial Guide to Designing a Diversified Gene Portfolio. en. *The Pangenome: Diversity, Dynamics and Evolution of Genomes*. Ed. by Tettelin H and Medini D. Cham: Springer International Publishing, 2020 :51–87
5. Margulis L. Power to the Protoctists. en. *Slanted Truths: Essays on Gaia, Symbiosis and Evolution*. Ed. by Margulis L and Sagan D. New York, NY: Springer, 1997 :75–82
6. Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. Nature 2012 Jun; 486:207–14
7. Gilbert JA et al. Current understanding of the human microbiome. en. Nature Medicine 2018 Apr; 24. Number: 4 Publisher: Nature Publishing Group:392–400
8. Katz SE and Pollan M. The Art of Fermentation. English. Illustrated edition. White River Junction, Vermont: Chelsea Green Publishing, 2012 May
9. Craig OE. Prehistoric Fermentation, Delayed-Return Economies, and the Adoption of Pottery Technology. Current Anthropology 2021 Oct; 62. Publisher: The University of Chicago Press:S233–S241
10. Velavan TP and Meyer CG. COVID-19: A PCR-defined pandemic. English. International Journal of Infectious Diseases 2021 Feb; 103. Publisher: Elsevier:278–9

11. Erlich HA, Gelfand D, and Sninsky JJ. Recent Advances in the Polymerase Chain Reaction. en. Science 1991 Jun; 252:1643–51
12. Brock TD and Freeze H. Thermus aquaticus gen. n. and sp. n., a Nonsporulating Extreme Thermophile. Journal of Bacteriology 1969 Apr; 98. Publisher: American Society for Microbiology:289–97
13. Doudna JA and Charpentier E. The new frontier of genome engineering with CRISPR-Cas9. Science 2014 Nov; 346. Publisher: American Association for the Advancement of Science:1258096
14. Hsu PD, Lander ES, and Zhang F. Development and Applications of CRISPR-Cas9 for Genome Engineering. English. Cell 2014 Jun; 157. Publisher: Elsevier:1262–78
15. Sharma A et al. CRISPR-Cas9 Editing of the HBG1 and HBG2 Promoters to Treat Sickle Cell Disease. New England Journal of Medicine 2023 Aug; 389. Publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJMoa2215643:820–32
16. Ikuta KS et al. Global mortality associated with 33 bacterial pathogens in 2019: a systematic analysis for the Global Burden of Disease Study 2019. English. The Lancet 2022 Dec; 400. Publisher: Elsevier:2221–48
17. Organization WH. Global Tuberculosis Report 2023. en. Tech. rep. World Health Organization, 2023
18. Murray CJL et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. English. The Lancet 2022 Feb; 399. Publisher: Elsevier:629–55
19. Centers for Disease Control and Prevention (U.S.) Antibiotic resistance threats in the United States, 2019. Tech. rep. Centers for Disease Control and Prevention (U.S.), 2019 Nov
20. Brito IL. Examining horizontal gene transfer in microbial communities. en. Nature Reviews Microbiology 2021 Jul; 19. Number: 7 Publisher: Nature Publishing Group:442–53
21. Alberts B et al. Molecular biology of the cell. eng. Seventh edition, international student edition. New York, NY London: W.W. Norton & Company, 2022
22. Gottesman S and Storz G. Bacterial Small RNA Regulators: Versatile Roles and Rapidly Evolving Variations. en. Cold Spring Harbor Perspectives in Biology 2011 Dec; 3. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab:a003798
23. Bromham L. An Introduction to Molecular Evolution and Phylogenetics. Second Edition, Second Edition. Oxford, New York: Oxford University Press, 2016 Mar
24. diCenzo GC and Finan TM. The Divided Bacterial Genome: Structure, Function, and Evolution. Microbiology and Molecular Biology Reviews 2017 Aug; 81. Publisher: American Society for Microbiology:10.1128/mmbr.00019–17
25. Rodríguez-Beltrán J, DelaFuente J, León-Sampedro R, MacLean RC, and San Millán Á. Beyond horizontal gene transfer: the role of plasmids in bacterial evolution. en. Nature Reviews Microbiology 2021 Jun; 19. Number: 6 Publisher: Nature Publishing Group:347–59
26. Land M et al. Insights from 20 years of bacterial genome sequencing. en. Functional & Integrative Genomics 2015 Mar; 15:141–61
27. Tettelin H et al. Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial "pan-genome". Proceedings of the National Academy of Sciences 2005 Sep; 102. Publisher: Proceedings of the National Academy of Sciences:13950–5
28. Touchon M et al. Phylogenetic background and habitat drive the genetic diversification of Escherichia coli. en. PLOS Genetics 2020 Jun; 16. Publisher: Public Library of Science:e1008866
29. Young JPW. Bacteria Are Smartphones and Mobile Genes Are Apps. en. Trends in Microbiology 2016 Dec; 24:931–2
30. Vos M, Hesselman MC, Beek TAt, Passel MWJv, and Eyre-Walker A. Rates of Lateral Gene Transfer in Prokaryotes: High but Why? English. Trends in Microbiology 2015 Oct; 23. Publisher: Elsevier:598–605

31. Anderson RP and Roth JR. Tandem Genetic Duplications in Phage and Bacteria. Annual Review of Microbiology 1977; 31. _eprint: https://doi.org/10.1146/annurev.mi.31.100177.002353:473–505

32. Kirchberger PC, Schmidt ML, and Ochman H. The Ingenuity of Bacterial Genomes. Annual Review of Microbiology 2020; 74. _eprint: https://doi.org/10.1146/annurev-micro-020518-115822:815–34

33. Treangen TJ and Rocha EPC. Horizontal transfer, not duplication, drives the expansion of protein families in prokaryotes. PLoS Genet. 2011; 7

34. Brockhurst MA et al. The Ecology and Evolution of Pangenomes. en. Current Biology 2019 Oct; 29:R1094–R1103

35. McInerney JO, McNally A, and O'Connell MJ. Why prokaryotes have pangenomes. en. Nature Microbiology 2017 Mar; 2. Number: 4 Publisher: Nature Publishing Group:1–5

36. Arnold BJ, Huang IT, and Hanage WP. Horizontal gene transfer and adaptive evolution in bacteria. en. Nature Reviews Microbiology 2022 Apr; 20. Publisher: Nature Publishing Group:206–18

37. Medini D, Donati C, Rappuoli R, and Tettelin H. The Pangenome: A Data-Driven Discovery in Biology. en. *The Pangenome: Diversity, Dynamics and Evolution of Genomes*. Ed. by Tettelin H and Medini D. Cham: Springer International Publishing, 2020 :3–20

38. McInerney JO, Whelan FJ, Domingo-Sananes MR, McNally A, and O'Connell MJ. Pangenomes and Selection: The Public Goods Hypothesis. en. *The Pangenome: Diversity, Dynamics and Evolution of Genomes*. Ed. by Tettelin H and Medini D. Cham: Springer International Publishing, 2020 :151–67

39. Tonkin-Hill G, Corander J, and Parkhill J. Challenges in prokaryote pangenomics. Microbial Genomics 2023; 9. Publisher: Microbiology Society,:001021

40. Koonin EV. Orthologs, Paralogs, and Evolutionary Genomics1. en. Annual Review of Genetics 2005 Dec; 39. Publisher: Annual Reviews:309–38

41. Gabaldón T and Koonin EV. Functional and evolutionary implications of gene orthology. en. Nature Reviews Genetics 2013 May; 14. Publisher: Nature Publishing Group:360–6

42. Hernández-Plaza A et al. eggNOG 6.0: enabling comparative genomics across 12 535 organisms. Nucleic Acids Research 2023 Jan; 51:D389–D394

43. Kuznetsov D et al. OrthoDB v11: annotation of orthologs in the widest sampling of organismal diversity. Nucleic Acids Research 2023 Jan; 51:D445–D451

44. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics 2014 Jul; 30:2068–9

45. Schwengers O et al. Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. Microbial Genomics 2021; 7. Publisher: Microbiology Society,:000685

46. Hadfield J et al. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics 2018 Dec; 34:4121–3

47. Kimura M. Evolutionary Rate at the Molecular Level. en. Nature 1968 Feb; 217. Publisher: Nature Publishing Group:624–6

48. Lebreton F, Van Schaik W, and Manson A. Emergence of Epidemic Multidrug-Resistant Enterococcus faecium. Abstr. Gen. Meet. Am. Soc. Microbiol. 2013; 4:1–10

49. Arias CA and Murray BE. The rise of the Enterococcus: beyond vancomycin resistance. Nat. Rev. Microbiol. 2012 Mar; 10:266–78

50. Lebreton F et al. Tracing the Enterococci from Paleozoic Origins to the Hospital. Cell 2017; 169:849–861.e13

51. Quince C, Walker AW, Simpson JT, Loman NJ, and Segata N. Shotgun metagenomics, from sampling to analysis. Nat. Biotechnol. 2017; 35:833–44

52. Falkowski PG, Fenchel T, and Delong EF. The Microbial Engines That Drive Earth's Biogeochemical Cycles. Science 2008 May; 320. Publisher: American Association for the Advancement of Science:1034–9

53. Debray R et al. Priority effects in microbiome assembly. en. Nature Reviews Microbiology 2022 Feb; 20. Publisher: Nature Publishing Group:109–21

54. Ratzke C, Barrere J, and Gore J. Strength of species interactions determines biodiversity and stability in microbial communities. en. Nature Ecology & Evolution 2020 Mar; 4. Publisher: Nature Publishing Group:376–83

55. Marchesi JR and Ravel J. The vocabulary of microbiome research: a proposal. Microbiome 2015 Jul; 3:31

56. VanEvery H, Franzosa EA, Nguyen LH, and Huttenhower C. Microbiome epidemiology and association studies in human health. en. Nature Reviews Genetics 2023 Feb; 24. Publisher: Nature Publishing Group:109–24

57. The Integrative HMP (iHMP) Research Network Consortium. The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles during Periods of Human Health and Disease. English. Cell Host & Microbe 2014 Sep; 16. Publisher: Elsevier:276–89

58. Lloyd-Price J, Abu-Ali G, and Huttenhower C. The healthy human microbiome. Genome Medicine 2016 Apr; 8:51

59. Kau AL, Ahern PP, Griffin NW, Goodman AL, and Gordon JI. Human nutrition, the gut microbiome and the immune system. en. Nature 2011 Jun; 474. Publisher: Nature Publishing Group:327–36

60. Graham DB and Xavier RJ. Conditioning of the immune system by the microbiome. English. Trends in Immunology 2023 Jul; 44. Publisher: Elsevier:499–511

61. Lloyd-Price J et al. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. Nature 2019 May; 569:655–62

62. Turnbaugh PJ et al. An obesity-associated gut microbiome with increased capacity for energy harvest. en. Nature 2006 Dec; 444. Publisher: Nature Publishing Group:1027–31

63. Pleguezuelos-Manzano C et al. Mutational signature in colorectal cancer caused by genotoxic pks + E. coli. en. Nature 2020 Apr; 580. Number: 7802 Publisher: Nature Publishing Group:269–73

64. Gilbert JA et al. Microbiome-wide association studies link dynamic microbial consortia to disease. en. Nature 2016 Jul; 535. Number: 7610 Publisher: Nature Publishing Group:94–103

65. Gilbert JA. Microbiome therapy for recurrent Clostridioides difficile. English. The Lancet Microbe 2022 May; 3. Publisher: Elsevier:e334

66. Van Rossum T, Ferretti P, Maistrenko OM, and Bork P. Diversity within species: interpreting strains in microbiomes. Nat. Rev. Microbiol. 2020 Jun

67. Browne HP et al. Culturing of 'unculturable' human microbiota reveals novel taxa and extensive sporulation. en. Nature 2016 May; 533. Number: 7604 Publisher: Nature Publishing Group:543–6

68. Lagier JC et al. Culturing the human microbiota and culturomics. en. Nature Reviews Microbiology 2018 Sep; 16. Number: 9 Publisher: Nature Publishing Group:540–50

69. Forster SC et al. A human gut bacterial genome and culture collection for improved metagenomic analyses. en. Nature Biotechnology 2019 Feb; 37. Number: 2 Publisher: Nature Publishing Group:186–92

70. Zou Y et al. 1,520 reference genomes from cultivated human gut bacteria enable functional microbiome analyses. Nat. Biotechnol. 2019 Feb; 37:179–85

71. Loman NJ and Pallen MJ. Twenty years of bacterial genome sequencing. Nat. Rev. Microbiol. 2015 Dec; 13:787–94

72. Dijk EL van, Jaszczyszyn Y, Naquin D, and Thermes C. The Third Revolution in Sequencing Technology. English. Trends in Genetics 2018 Sep; 34. Publisher: Elsevier:666–81

73. Marx V. Method of the year: long-read sequencing. en. Nature Methods 2023 Jan; 20. Publisher: Nature Publishing Group:6–11

74. Jain M et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. en. Nature Biotechnology 2018 Apr; 36. Publisher: Nature Publishing Group:338–45

75. Oehler JB, Wright H, Stark Z, Mallett AJ, and Schmitz U. The application of long-read sequencing in clinical settings. Human Genomics 2023 Aug; 17:73

76. Faria NR et al. Epidemic establishment and cryptic transmission of Zika virus in Brazil and the Americas. en. Pages: 105171 Section: New Results. 2017 Mar

77. Quick J et al. Multiplex PCR method for MinION and Illumina sequencing of Zika and other virus genomes directly from clinical samples. en. Nature Protocols 2017 Jun; 12. Publisher: Nature Publishing Group:1261–76

78. Simpson JT and Pop M. The Theory and Practice of Genome Sequence Assembly. Annual Review of Genomics and Human Genetics 2015; 16:153–72

79. Myers EW. The fragment assembly string graph. Bioinformatics 2005 Jan; 21:ii79–ii85

80. Simpson JT and Durbin R. Efficient construction of an assembly string graph using the FM-index. Bioinformatics 2010; 26:367–73

81. Simpson JT et al. ABySS: A parallel assembler for short read sequence data. Genome Res. 2009 Jun; 19:1117–23

82. Bankevich A et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. Journal of Computational Biology 2012 May; 19. Publisher: Mary Ann Liebert, Inc., publishers:455–77

83. Sohn Ji and Nam JW. The present and future of de novo whole-genome assembly. Briefings in Bioinformatics 2018 Jan; 19:23–40

84. Durbin R, Eddy SR, Krogh A, and Mitchison G. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. en. 1st ed. Cambridge University Press, 1998 Apr

85. Li H and Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009; 25:1754–60

86. Li H. Minimap2: pairwise alignment for nucleotide sequences. en. Bioinformatics 2018 Sep; 34. Publisher: Oxford Academic:3094–100

87. Walker BJ et al. Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One 2014; 9

88. Auwera Gvd and O'Connor BD. Genomics in the cloud: using Docker, GATK, and WDL in Terra. First edition. OCLC: on1148137471. Sebastopol, CA: O'Reilly Media, 2020

89. Koboldt DC. Best practices for variant calling in clinical sequencing. Genome Medicine 2020 Oct; 12:91

90. Bush SJ et al. Genomic diversity affects the accuracy of bacterial single-nucleotide polymorphism–calling pipelines. en. GigaScience 2020 Feb; 9. Publisher: Oxford Academic

91. Colquhoun RM et al. Pandora: nucleotide-resolution bacterial pan-genomics with reference graphs. Genome Biology 2021 Sep; 22:267

92. Wood DE, Lu J, and Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biology 2019 Nov; 20:257

93. Blanco-Miguez A et al. Extending and improving metagenomic taxonomic profiling with uncharacterized species with MetaPhlAn 4. en. Pages: 2022.08.22.504593 Section: New Results. 2022 Aug

94. Ayling M, Clark MD, and Leggett RM. New approaches for metagenome assembly with short reads. Briefings in Bioinformatics 2020 Mar; 21:584–94

95. Wu M and Eisen JA. A simple, fast, and accurate method of phylogenomic inference. Genome Biology 2008 Oct; 9:R151

96. Yan Y, Nguyen LH, Franzosa EA, and Huttenhower C. Strain-level epidemiology of microbial communities and the human microbiome. Genome Medicine 2020 Aug; 12:71

97. Compeau P and Pevzner P. Bioinformatics algorithms: an active learning approach. 3rd edition. La Jolla, CA: Active Learning Publishers, 2018

98. Needleman SB and Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. en. Journal of Molecular Biology 1970 Mar; 48:443–53

99. Gotoh O. An improved algorithm for matching biological sequences. en. Journal of Molecular Biology 1982 Dec; 162:705–8

100. Marco-Sola S, Moure JC, Moreto M, and Espinosa A. Fast gap-affine pairwise alignment using the wavefront algorithm. Bioinformatics 2021 Feb; 37:456–63

101. Kurtz S et al. Versatile and open software for comparing large genomes. Genome Biology 2004 Jan; 5:R12

102. Paten B et al. Cactus: Algorithms for genome multiple sequence alignment. Genome Res. 2011 Sep; 21:1512–28

103. Jain C, Koren S, Dilthey A, Phillippy AM, and Aluru S. A fast adaptive algorithm for computing whole-genome homology maps. Bioinformatics 2018 Sep; 34:i748–i756

104. Minkin I and Medvedev P. Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. en. Nature Communications 2020 Dec; 11. Publisher: Nature Publishing Group:6327

105. Marçais G et al. MUMmer4: A fast and versatile genome alignment system. PLoS Comput. Biol. 2018 Jan; 14:e1005944

106. Kille B, Garrison E, Treangen TJ, and Phillippy AM. Minmers are a generalization of minimizers that enable unbiased local Jaccard estimation. Bioinformatics 2023 Sep; 39:btad512

107. Page AJ et al. Roary: rapid large-scale prokaryote pan genome analysis. Bioinformatics 2015 Nov; 31:3691–3

108. Sheikhizadeh S, Schranz ME, Akdel M, Ridder D de, and Smit S. PanTools: representation, storage and exploration of pan-genomic data. Bioinformatics 2016 Sep; 32:i487–i493

109. Gautreau G et al. PPanGGOLiN: Depicting microbial diversity via a partitioned pangenome graph. en. PLOS Computational Biology 2020 Mar; 16. Publisher: Public Library of Science:e1007732

110. Ding W, Baumdicker F, and Neher RA. panX: pan-genome analysis and exploration. Nucleic Acids Research 2018 Jan; 46:e5

111. Tonkin-Hill G et al. Producing polished prokaryotic pangenomes with the Panaroo pipeline. Genome Biology 2020 Jul; 21:180

112. Georgescu CH et al. SynerClust: a highly scalable, synteny-aware orthologue clustering tool. Microbial Genomics 2018; 4. Publisher: Microbiology Society,

113. Price MN, Dehal PS, and Arkin AP. FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. en. Molecular Biology and Evolution 2009 Jul; 26. Publisher: Oxford Academic:1641–50

114. Minh BQ et al. IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. Molecular Biology and Evolution 2020 May; 37:1530–4

115. Thorpe HA, Bayliss SC, Sheppard SK, and Feil EJ. Piggy: a rapid, large-scale pan-genome analysis tool for intergenic regions in bacteria. GigaScience 2018 Apr; 7:giy015

116. Salzberg SL. Next-generation genome annotation: we still struggle to get it right. Genome Biology 2019 May; 20:92

117. The Computational Pan-Genomics Consortium. Computational pan-genomics: status, promises and challenges. Briefings in Bioinformatics 2018 Jan; 19:118–35

118. Eizenga JM et al. Pangenome Graphs. Annual Review of Genomics and Human Genetics 2020; 21. _eprint: https://doi.org/10.1146/annurev-genom-120219-080406:139–62

119. Garrison E et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. en. Nature Biotechnology 2018 Oct; 36. Number: 9 Publisher: Nature Publishing Group:875–9

120. Hickey G et al. Pangenome graph construction from genome alignments with Minigraph-Cactus. en. Nature Biotechnology 2023 May. Publisher: Nature Publishing Group:1–11

121. Garrison E et al. Building pangenome graphs. en. Pages: 2023.04.05.535718 Section: New Results. 2023 Apr

2

Fast and exact gap-affine partial order alignment with POASTA

Lucas R. van Dijk, Abigail L. Manson, Ashlee M. Earl,
Kiran V Garimella, Thomas Abeel

This chapter has been accepted to Oxford Bioinformatics. DOI: 10.1093/bioinformatics/btae757.

Abstract

Motivation: Partial order alignment is a widely used method for computing multiple sequence alignments, with applications in genome assembly and pangenomics, among many others. Current algorithms to compute the optimal, gap-affine partial order alignment do not scale well to larger graphs and sequences. While heuristic approaches exist, they do not guarantee optimal alignment and sacrifice alignment accuracy.

Results: We present POASTA, a new optimal algorithm for partial order alignment that exploits long stretches of matching sequence between the graph and a query. We benchmarked POASTA against the state-of-the-art on several diverse bacterial gene datasets and demonstrated an average speed-up of 4.1x and up to 9.8x, using less memory. POASTA's memory scaling characteristics enabled the construction of much larger POA graphs than previously possible, as demonstrated by megabase-length alignments of 342 *Mycobacterium tuberculosis* sequences.

Availability and implementation: POASTA is available on Github at `https://github.com/broadinstitute/poasta`.

2.1. Introduction

Multiple sequence alignments (MSAs) are central to computational biology. MSAs have many applications, including computing genetic distances, which can serve as a basis for a phylogeny; determining consensus sequences, e.g., to perform read error correction; and identifying allele frequencies, e.g., for sequence motif identification.

Computing the optimal MSA with the *sum of all pairs* (SP) score is an NP-complete problem [1]. These classical exact algorithms have a runtime exponentially related to the number of sequences and are thus intractable for even modest-sized datasets. Instead, nearly all popular MSA tools, including MAFFT [2] and MUSCLE [3], compute the MSA progressively: first, an alignment between two sequences is computed, then additional sequences are added one by one until all sequences have been aligned. The runtime of these approaches is linear in the number of sequences instead of exponential. While MSAs computed this way do not necessarily find the globally optimal solution for the SP objective, they are still highly useful approximations to otherwise intractable alignment problems.

Partial order alignment (POA) is a well-known progressive MSA approach that pioneered using a graph to represent an MSA rather than a sequence profile [4]. This improved the ability to represent indels, leading to higher-quality alignments. Since POA is a progressive MSA algorithm, the optimal SP score is not guaranteed for the entire MSA. However, POA does guarantee that each individual sequence-to-graph alignment is optimal.

POA is relevant to many applications, including *de novo* genome assembly (e.g.,

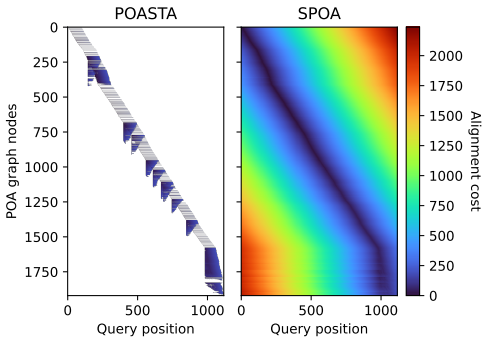

Figure 2.1: Representation of the dynamic programming matrix to compute the global alignment of a *nusA* gene sequence (x-axis) to a POA graph constructed from 50 other *nusA* gene sequences (y-axis). Each pixel represents a computed alignment state, and the color represents the alignment cost of that state. White pixels represent uncomputed states.

read error correction and consensus generation) [5, 6, 7], RNA isoform inference [8], structural variant (SV) characterization [9], and variant phasing [10].

POA is also essential to two recent human pangenome graph construction pipelines [11, 12]. These pipelines are pushing the limits of POA, as aligning long stretches of homologous sequence among input genomes requires substantial computing and memory resources. For example, consider the gap-affine alignment of a 500 kbp sequence to a graph with 500k character-labeled nodes. Conventional POA approaches have a runtime and memory complexity of $O(|V|m)$, i.e., a product of the number of nodes in a POA graph $|V|$ and the sequence length m. This example would, therefore, require about 3 TB of RAM (assuming 32-bit integers for storing alignment costs in three alignment state matrices).

Several tools, including SPOA [7] and abPOA [13], have been developed to address the need for faster and more memory-efficient POA algorithms. The current state-of-the-art, SPOA, is a reimplementation of the original algorithm, which accelerates computing the dynamic programming (DP) matrix by using single-instruction-multiple-data (SIMD) instructions available on modern CPUs. While faster, SPOA still computes the full DP matrix and thus does not ameliorate demands on memory usage. abPOA additionally improves performance by applying an adaptive banding strategy to partially compute the DP matrix. However, this sacrifices the guarantee of finding the optimal sequence-to-graph alignment.

Here, we present POASTA: a fast, memory-efficient, and optimal POA algorithm that computes many fewer alignment states than SPOA, thus enabling the construction of much larger POA graphs (Figure 2.1). It is built on top of the A* algorithm [14], with a new POA-specific heuristic. Inspired by the recently published wavefront algorithm for pairwise alignment [15], it also exploits exact matches between a query sequence and the graph. We additionally introduce a novel superbubble-informed [16] technique for pruning the number of computed alignment states without sacrificing alignment optimality. We benchmarked POASTA against SPOA [7] on diverse sets of bacterial housekeeping genes extracted from RefSeq and demonstrated its in-

creased performance. Additionally, we constructed megabase-length alignments of 342 *Mycobacterium tuberculosis* sequences, demonstrating its reduced memory usage and highlighting POASTA's ability to align much longer sequences than previously possible.

2.2. Methods

POA algorithms compute an MSA by iteratively computing the alignment of a query to a directed acyclic graph (DAG) representing the MSA from the previous iteration [4]. Instead of the original DP formulation (Supplemental Text A; Supplementary Figure A.1a), POASTA's algorithm is based on an *alignment graph* (Supplementary Figure A.1b; not to be confused with the POA graph), enabling the use of common graph traversal algorithms such as the A* algorithm to compute alignments [14, 17, 18, 19]. POASTA further accelerates alignment using three novel techniques: 1) a cheap-to-compute, POA-specific heuristic for the A* algorithm (Figure 2.2a), 2) a depth-first search component, greedily aligning exact matches between the query and the graph (Figure 2.2b); and 3) a method to detect and prune alignment states that are not part of the optimal solution, informed by the POA graph topology (Figure 2.2c). Together, they substantially reduce the number of computed alignment states (Supplementary Figure A.2).

Definitions and notation

To describe the algorithm in detail, we will use the following notation. A POA graph $G = (V, E)$ is a character-labeled DAG, where nodes $v \in V$ represent the symbols in the input sequences, each labeled with a character from an alphabet Σ. Edges $(u, v) \in E$ connect nodes that are adjacent in at least one input sequence. We additionally assume the POA graph has a special start node ν with outgoing edges to all nodes with no other incoming edges and a special termination node τ with incoming edges from all nodes with no other outgoing edges.

The optimal alignment of a query sequence $Q = q_1 q_2 \ldots q_m$ (of length m) to G is the alignment of Q to a path $\pi = \nu v_1 v_2 \ldots v_n \tau$, spelling a sequence R that minimizes the alignment cost C (Supplementary Figure A.1a). Commonly used cost models are linear gap penalties and gap-affine penalties. In the former, each gap position is weighted equally, and the alignment cost is defined as $C = N_m\Delta_m + N_x\Delta_x + N_g\Delta_g$, where N_m represents the number of matches, N_x is the number of mismatches, and N_g is the total length of gaps. The cost of each alignment operation is represented by Δ_m, Δ_x, and Δ_g, representing the cost of a match, mismatch, and a gap, respectively. In the case of gap-affine penalties, opening a new gap has a different (typically higher) cost than extending an existing gap. The total cost is defined as $C = N_m\Delta_m + N_x\Delta_x + N_o\Delta_o + N_g\Delta_e$, with N_o the number of distinct gaps and Δ_o the cost of opening a new gap, and Δ_e the cost of extending a gap [20]. POASTA supports both the gap-linear and the gap-affine cost models, though it constrains Δ_m to be zero and all other costs $\Delta_x, \Delta_o, \Delta_g, \Delta_e$ to be ≥ 0. Additionally, in case of the gap-affine model, it requires that the gap open cost Δ_o is greater than the gap extension cost Δ_e. For clarity, we focus on the gap-linear cost

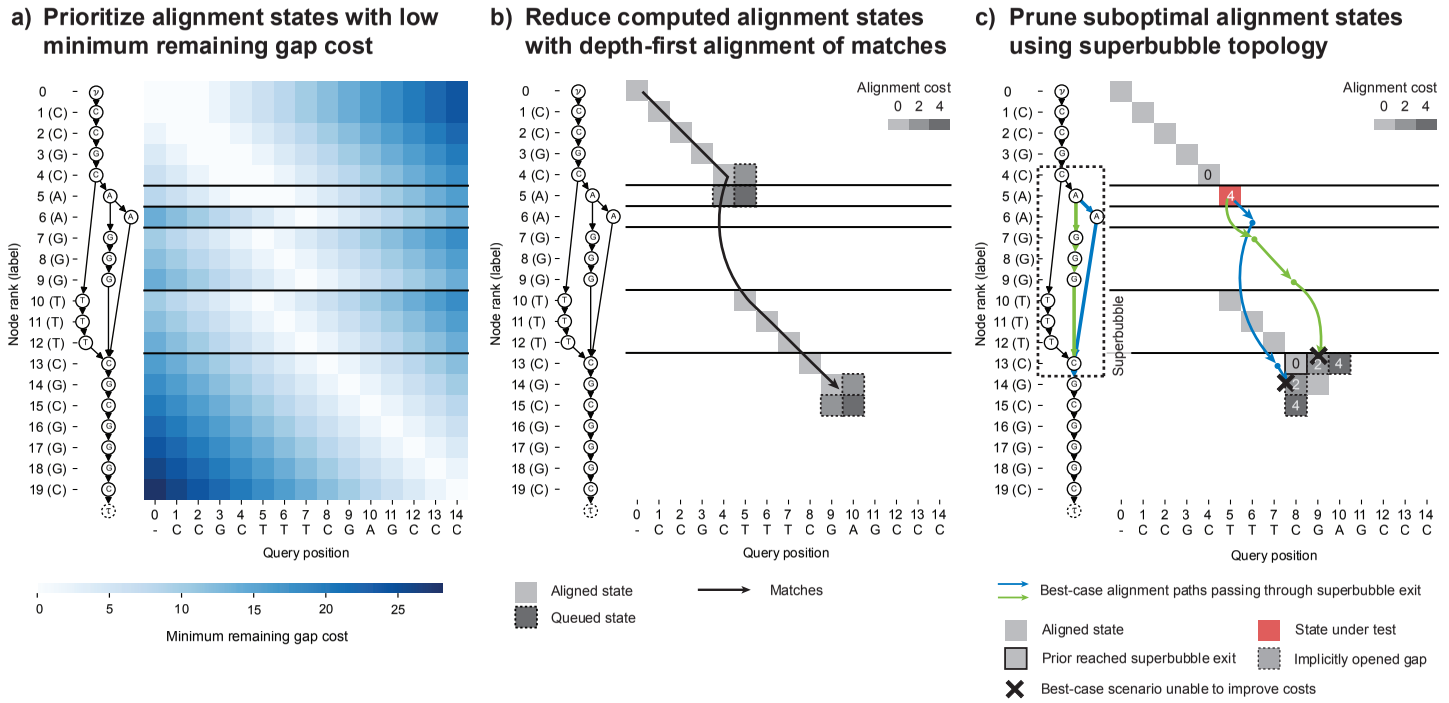

Figure 2.2: POASTA is based on the A* algorithm and accelerates alignment through three algorithmic innovations: **(a)** A novel heuristic for POA that prioritizes alignment states with a low minimum remaining gap cost (light-colored squares); i.e., states where the unaligned query sequence length is similar to the path lengths to the POA graph end node τ. **(b)** Reducing the number of computed alignment states by combining the A* algorithm with a depth-first search component, greedily aligning matches between the query and a path in the graph (black arrow). Adjacent insertion and deletion states are only queued when encountering a mismatch (squares with dashed borders). **(c)** Using knowledge about *superbubble* topology to prune states not part of the optimal solution. POASTA checks whether the best-case alignment paths (blue and green arrows) from a state under test (red square) can improve over the costs of implicitly opened gaps from prior reached bubble exits (bordered squares). All examples use the linear gap cost model with $\Delta_m = 0, \Delta_x = 4, \Delta_g = 2$.

model; the use of POASTA with the gap-affine cost is explained in the Supplemental Text A.2.

The alignment graph $G^A = (V^A, E^A)$ is a product of the POA graph and the query sequence, and paths through it represent possible alignments between them. Nodes $\langle v, i\rangle \in V^A = (V \times \{0, 1, ..., m\})$ represent *alignment states* with a cursor pointing to a node v in the POA graph and a cursor to a query position i (Supplementary Figure A.1b). Edges in the alignment graph correspond to different alignment operations, such as (mis)match, insertion, or deletion, and are weighted with the respective alignment cost. Edges connect alignment states where either one (indel) or both of the cursors have moved ((mis)match), and the construction of edges is further detailed in the Supplementary Text A.2. The lowest cost path in the alignment graph from $\langle v, 0\rangle$ to alignment termination state $\langle \tau, m\rangle$ is equivalent to the optimal alignment of Q to G.

Optimal alignment with A* using a minimum remaining gap cost heuristic

To compute the lowest-cost path in the alignment graph, i.e., the optimal alignment, POASTA uses the A* algorithm [14]. For POASTA, we adapted the widely used gap-cost heuristic for pairwise alignment to POA (Figure 2.2a) [21, 22]. This heuristic is *admissible*, i.e., a lower bound on the true remaining cost, thus guaranteeing that A* finds the lowest-cost path. The intuition behind the heuristic is to prioritize alignment states in which the length of the unaligned query sequence is similar to the path lengths to the end node τ.

To compute heuristic $h\langle v, i\rangle$, POASTA scans the POA graph before alignment starts and stores the shortest and longest path length to the end node τ for all nodes in the graph, denoted as $d^{\min}_{v,\tau}$ and $d^{\max}_{v,\tau}$. This can be computed in $O(V+E)$ time by visiting the nodes in reverse topological order. POASTA compares these path lengths to the length of the unaligned query sequence $l_r = m - i$ and infers the minimum number of indel edges to traverse from $\langle v, i\rangle$ to the alignment termination $\langle \tau, m\rangle$ state as follows:

Definition 1 (Minimum number of indel edges)

$$N_g^{\min} = \begin{cases} l_r - (d^{\max}_{v,\tau} - 1) & \text{if } d^{\max}_{v,\tau} - 1 < l_r \\ (d^{\min}_{v,\tau} - 1) - l_r & \text{if } d^{\min}_{v,\tau} - 1 > l_r \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

We subtract one from $d^{\min}_{v,\tau}$ and $d^{\max}_{v,\tau}$ to exclude the edge towards τ.

Proof See Supplemental Text A.2. □

Combining the computed minimum number of indel edges to traverse with the alignment cost model, e.g., the linear gap cost model, enables us to compute the heuristic.

Definition 2 (Minimum remaining gap cost heuristic)

$$h\langle v,i\rangle = N_g^{\min}\Delta_g \tag{2.2}$$

Lemma 1 (Admissibility) *$h\langle v,i\rangle$ is admissible.*

Proof The true remaining alignment cost, using linear gap penalties and assuming a match cost Δ_m of zero, is defined as $C_r = N_x\Delta_x + N_g\Delta_g$, where N_x and N_g represent the number of remaining mismatches and the total remaining gap length respectively, Δ_x the mismatch cost, and Δ_g the gap cost.

Using Definition 3, we infer that $N_g \geq N_g^{\min}$. Since the mismatch cost $\Delta_x \geq 0$, we note that the $N_x\Delta_x \geq 0$, and thus observe that

$$N_x\Delta_x + N_g\Delta_g \geq N_g^{\min}\Delta_g \Rightarrow C_r \geq h\langle v,i\rangle.$$

$h\langle v,i\rangle$ is thus a lower bound on the true remaining alignment cost. □

Depth-first alignment of exact matches between query and graph

To further speed up alignment and reduce the number of computed alignment states, POASTA greedily aligns exact matches between the query and graph (Figure 2.2b). This is possible because POASTA requires that the alignment cost for a match is zero and all other alignment costs be ≥ 0. Traversing a match edge $\langle u,i\rangle \rightarrow \langle v,i+1\rangle$ will always be the optimal choice if the latter state has not been visited yet since match edges have zero cost and all other paths (requiring indels) will have higher or equal cost [19, 15]. This implies that in the presence of an unvisited match, we can ignore insertion edge $\langle u,i\rangle \rightarrow \langle u,i+1\rangle$ and deletion edge $\langle u,i\rangle \rightarrow \langle v,i\rangle$.

To implement this, POASTA combines the regular A* algorithm with a depth-first search (DFS) component. When a state $\langle u,i\rangle$ is popped from the A* queue, we initiate a DFS from this state. We assess whether a successor state $\langle v,i+1\rangle \quad v : (u,v) \in E$ is a match; if it is, we push it on the stack to be processed in the next DFS iteration; when there is a mismatch, we append it to the A* queue. In the latter case, we no longer can ignore the insertion and deletion edges, so we additionally queue insertion state $\langle u,i+1\rangle$, and deletion state $\langle v,i\rangle$. Note that, just like regular DFS, a state is removed from the stack after all its successors (matches or mismatches) have been explored. Thus, using DFS enables greedily aligning long stretches of exact matches, even in the presence of branches in the graph.

Pruning alignment states not part of the optimal solution

When POASTA's depth-first alignment finds a long stretch of matching sequence, the corresponding path through the POA graph might traverse a *superbubble* [16]. A superbubble (s,t) is a substructure in the POA graph with specific topological features (Supplementary Figure A.3): it is acyclic; it has a single entrance s and a single exit t; all paths leaving s should end in t; and no path from "outside" the superbubble can have an endpoint inside the bubble. The set of nodes U on paths from s to t is called

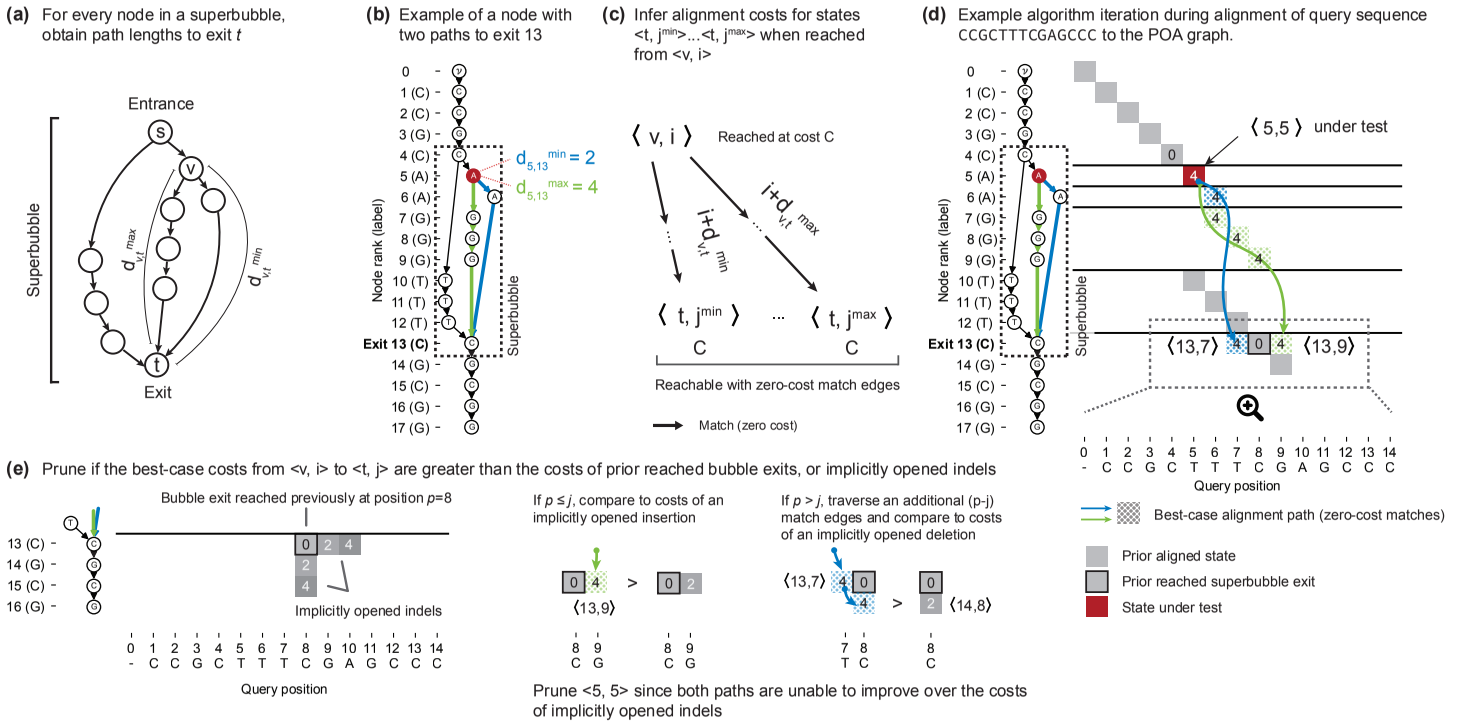

Figure 2.3: POASTA detects and prunes alignment states that are not part of the optimal solution. **(a)** A superbubble with entrance s and exit t. For every node v in the superbubble, POASTA stores the minimum and maximum path length, $d_{v,t}^{\min}$, and $d_{v,t}^{\max}$, to exit t. **(b)** An example POA graph with a superbubble (dashed rectangle) and the path lengths from the highlighted node 5 (red) to the superbubble exit (green and blue paths). **(c)** The path lengths to the superbubble exit are used during alignment to infer the range of states reachable with zero-cost match edges (black arrows) from another state $\langle v, i \rangle$. **(d)** An example aligning the query `CCGCTTTCGAGCCC` to the graph in (b). Grey squares: states aligned in a prior iteration. Red square: state under test. Blue and green arrows and dotted squares: best-case alignment paths from the state under test to the superbubble exit. **(e)** POASTA compares best-case alignment costs from $\langle v, i \rangle$ (blue and green dotted squares) to implicitly opened indels from prior reached bubble exits (grey squares). Implicitly opened indels act as an upper bound for the alignment cost of yet-to-visit states. Examples use the linear gap cost model with $\Delta_m = 0, \Delta_x = 4, \Delta_g = 2$.

the *interior* of a bubble, which can be empty. In a POA graph, superbubbles represent the alleles present at particular loci in the MSA.

POASTA exploits the fact that all paths through a superbubble have a common endpoint, its exit t. If an alignment state $\langle t, p \rangle$ is reached during alignment with a particular cost $C_{\langle t,p \rangle}$, POASTA can detect whether another yet-to-visit state $\langle v, i \rangle : v \in U \cup \{s\}$ that is part of the same superbubble, can improve over this cost. This is especially effective when combined with the depth-first greedy alignment described above; if a bubble exit is reached at a low cost because of a long stretch of matching sequence, we can often prune alignment states on alternative paths through the bubble because they can not improve over the already-found path.

To quickly retrieve topological information about super-bubbles, POASTA constructs a *superbubble index* before alignment. For every node in the POA graph, it stores the superbubbles in which it is contained, along with the shortest and longest path length to the corresponding superbubble exit (Figure 2.3a). For example, the red node (node 5) in the example shown in Figure 2.3b has two paths to the superbubble exit (node 13): one path with length 2 (blue) and one path with length 4 (green). POASTA identifies superbubbles using the $O(V + E)$ algorithm described by Gärtner *et al.* [23]. The shortest path lengths can be computed using a backward breadth-first search (BFS), and the longest path lengths can be computed by recursively visiting nodes in postorder, both $O(V + E)$ operations.

To test if a state $\langle v, i \rangle$ should be pruned, POASTA first uses the superbubble index to infer the range of states $\langle t, j^{\min} \rangle \dots \langle t, j^{\max} \rangle$ reachable from $\langle v, i \rangle$ assuming the best-case scenario of traversing zero-cost match edges (Figure 2.3c). For example, when aligning a query `CCGCTTTCGAGCCC` to the graph in Figure 2.3b, POASTA will initially find a long stretch of matches between the query and a path in the graph, traversing the superbubble $(4, 13)$ (Figure 2.3d; grey squares). In a following iteration, it tests alignment state $\langle 5, 5 \rangle$, where node 5 is part of the same superbubble $(4, 13)$, which is reached with an alignment cost 4 (Figure 2.3d; red square). It looks up the path lengths to the superbubble exit $d^{\min}_{5,13} = 2$ and $d^{\max}_{5,13} = 4$ and infers that we can reach $\langle 13, 7 \rangle$ and $\langle 13, 9 \rangle$ from $\langle 5, 5 \rangle$ with the same alignment cost of four (Figure 2.3d; blue and green arrows and dotted squares).

POASTA can now compare this best-case alignment cost, when reached from a state $\langle v, i \rangle$, to the alignment costs of states that reached the superbubble exit prior, or an *implicitly opened gap* from those. Implicitly opened gap costs are upper bounds on the cost for yet-to-visit alignment states and are computed on the fly when testing to prune a state (Figure 2.3e). For example, the green path in Figure 2.3d could reach alignment state $\langle 13, 9 \rangle$ with an alignment cost of four. However, alignment state $\langle 13, 9 \rangle$ is also reachable from the prior reached bubble exit $\langle 13, 8 \rangle$, by opening an insertion and reaching it with a lower cost of two (Figure 2.3e). Similarly, the blue path in Figure 2.3d could reach alignment state $\langle 13, 7 \rangle$ with an alignment cost of four. This state has not yet been reached and is also not reachable by opening a gap from a previously reached exit. However, suppose we extend the blue path, assuming additional traversal of zero-cost match edges. In that case, we reach an alignment state $\langle 14, 8 \rangle$ which is reachable from a previously reached exit by opening a deletion. The opened deletion would reach $\langle 14, 8 \rangle$ with a cost of two, lower than the cost of four when reached

through the blue path. Since both best-case scenarios from $\langle 5, 5\rangle$ would result in higher alignment costs compared to opened indels from a prior reached exit, POASTA infers $\langle 5, 5\rangle$ will not be part of the optimal solution and prunes it from further consideration.

In the example discussed above, the bubble exit was only reached once. Bubble exits, however, can be reached multiple times during alignment (with varying alignment costs). All previously reached positions should be considered when testing whether a state can be pruned (Supplementary Figure A.4). The Supplemental Methods further detail how POASTA prunes alignment states when the bubble exit has been reached multiple times.

2.3. Results

Benchmarking using bacterial housekeeping genes

To compare POASTA's speed and memory usage to the current state of the art, we generated multiple benchmark datasets from bacterial housekeeping genes (*dnaG*, *nusA*, *pgk*, *pyrG*, and *rpoB*). These genes are present in nearly all bacteria and are commonly used to create bacterial phylogenies, requiring MSA [24]. We downloaded all 40,188 RefSeq-complete genomes representing the breadth of bacterial diversity and extracted genes of interest using the accompanying gene annotations. Gene sequences were deduplicated and coarsely clustered using single-linkage hierarchical clustering. This resulted in multiple genus-spanning clusters. For each gene family, we selected one or more clusters as benchmark datasets, choosing clusters with at least 100 sequences and varying pairwise ANI (Supplemental Methods). The 13 selected benchmark sets each contained 140-2,385 gene sequences, with mean sequence lengths of 1-4kbp and pairwise ANIs of 82%-97% (Supplemental Table 1).

POASTA constructs multiple sequence alignments 4x faster than other optimal methods

We assessed POASTA's runtime and memory compared to SPOA, the only other POA algorithm that guarantees optimal partial order alignments [7]. We did not benchmark against general sequence-to-graph aligners such as Astarix [19], GWFA [25], PaSGAL [26], and GraphAligner [27] since these are unable to compute *multiple* sequence alignments and we would be unable to compare total runtime. We ran POASTA and SPOA to compute the full MSA of the 13 selected datasets and recorded their total runtime and memory usage.

For 12 of 13 datasets, POASTA computed the complete MSA faster than SPOA, achieving an average speed-up of 4.1x. The highest speed-up was 9.8x (Figure 2.4a). The one instance where SPOA was faster corresponded to the gene set with the lowest pairwise ANI (82.6%). POASTA's strongest relative performance was in settings with ANIs of 90-100% and sequences longer than 1,500 bp (Figure 2.4b,c). Furthermore, SPOA required, on average, 2.6x more memory than POASTA (Figure 2.4d-f).

We also compared POASTA's runtime and memory to abPOA, a popular tool for POA that does not guarantee optimal alignment [13]. As expected due to its adaptive

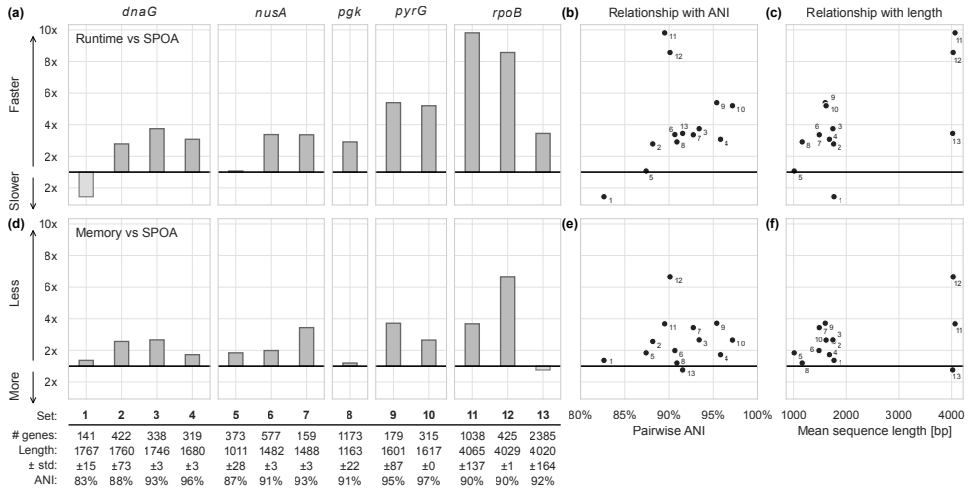

Figure 2.4: POASTA creates multiple sequence alignments of sequences from five bacterial housekeeping genes, in all but one case faster than SPOA and with less memory. **(a)** Relative runtime of POASTA compared to SPOA for each set of gene sequences. **(b)** The relationship between pairwise ANI of each gene sequence set and POASTA's relative runtime. **(c)** The relationship between mean sequence length and POASTA's relative runtime. **(d)** Relative memory usage of POASTA compared to SPOA for each set of gene sequences. **(e)** The relationship between pairwise ANI of each sequence set and POASTA's relative memory usage. **(f)** The relationship between the mean sequence length of each sequence set and POASTA's relative memory usage.

banding strategy, abPOA is faster than POASTA (3.5x; Supplementary Figure A.5a). Surprisingly, abPOA used more memory than POASTA across nearly all benchmark sets (Supplementary Figure A.5b), as it allocates memory for the entire matrix, even though it only computes a fraction of it. For our dataset, we found that abPOA found the optimal alignment the vast majority (99.8%) of the time (Supplemental Text A.2). However, our test dataset had few large indels and adaptive banding strategies are known to miss the optimal alignment more frequently in the presence of indels larger than the band size [28]. For many cases where the optimal alignment was missed in our test dataset, abPOA produced erroneous alignments that started or ended at unexpected nodes. This resulted in alignment costs that were lower than the global optimum reported by SPOA and POASTA, which should be impossible (see Supplemental Text A.2).

POASTA enables the construction of megabase-length POA graphs

To further test POASTA's limits, we benchmarked its ability to align datasets with average sequence lengths of approximately 250 kbp, 500 kbp, and 1 Mbp. We extracted subsequences from all 370 RefSeq-complete whole genome assemblies of *Mycobacterium tuberculosis*, covering a broad range of the species' diversity (including representatives from all known lineages; Mash-estimated average pairwise ANI of 99.3% [29]). *M. tuberculosis* has relatively little large-scale structural variation, including few large inversions or genes translocating to different locations, which POA cannot

model and align accurately. After orienting genomes such that each started with the gene *dnaA*, we truncated at specific shared genes to achieve sequences of the desired length (Supplemental Text A.2). For the 250 kbp, 500 kbp, and 1 Mbp benchmarks, we truncated at the genes *trmB*, *thiE*, and *gltA2*, respectively. Since POA expects sequences to be colinear, we also excluded 28 genomes with more than 15% ($\geq$ 660 kbp) of its complete genome inverted with respect to the canonical reference H37Rv (Supplemental Text A.2).

POASTA successfully computed MSAs for the 250 kbp, 500 kbp, and 1 Mbp benchmark sets with manageable runtimes and memory (Table 2.1). None of these alignments could be completed with SPOA or abPOA, which required more memory than the 240 GB available in the Google Cloud VM used for benchmarking (Supplemental Text A.2). The estimated memory requirements for the 250, 500, and 1,000 kbp benchmarks would be 0.95, 3.5, and 13 TB, respectively (assuming 32-bit integers for storing scores).

Table 2.1: POASTA runtime and peak memory usage for three benchmark sets comprising 342 *M. tuberculosis* sequences of approximately 250, 500, and 1,000 kbp.

Sequence set	Runtime	Max. memory
250 kbp	5.3 h	63.8 GB
500 kbp	24 h	120 GB
1,000 kbp	69 h	231 GB

We assessed computed alignments at a known drug resistance locus to validate that the MSA correctly captured known variation. In *M. tuberculosis*, the S450L change in the *rpoB* gene is one of the most common rifampicin resistance-causing mutations [30, 31]. We first characterized codons representing the 450th amino acid of *rpoB* using just the reference genomes and accompanying gene annotations. We obtained each codon using the start position of the *rpoB* gene to compute the reference locus representing the 450th amino acid of *rpoB*. In our set of genomes, we similarly observed that the S450L mutation is the most common allele present other than the reference or wild-type allele (Table 2.2; 103 genomes have the S450L mutation). To check if the observed codons were correctly aligned in the POA graph, we extracted a small subgraph surrounding the 450th amino acid of *rpoB* in H37Rv (Figure 2.5). While this subgraph was obtained using H37Rv coordinates, all codons listed in Table 2.2 were also represented as different paths in the graph, and the graph edge counts, indicating the number of genomes sharing that edge, matched the codon counts obtained through gene annotations. POASTA thus correctly captured known variation at this locus while the alignments were computed unaware of genes.

2.4. Discussion

In this work, we introduced POASTA, an optimal POA algorithm supporting gap-affine penalties with increased performance. These improvements are achieved using three algorithmic innovations: a minimum remaining gap cost heuristic for A*, depth-first

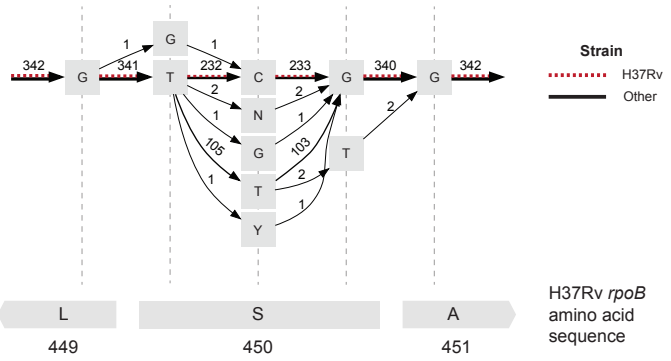

Figure 2.5: The POA subgraph surrounding the 450th amino acid in *M. tuberculosis* H37Rv *rpoB* (red dashed edges) captures extensive allelic diversity in other references (black edges). Grey squares represent nodes in the POA graph labeled with a base or an IUPAC code representing uncertainty about the base at that site (N: any base, Y: C or T). Edges are labeled with the number of genomes that share that edge. The bottom grey rectangles represent the H37Rv amino acid sequence.

greedy alignment of matches, and pruning states not part of the optimal solution using superbubble topology. In benchmarking on short sequences (1-4 kbp), POASTA was, on average, 4.1x faster than the current state-of-the-art SPOA [7] and used 2.6x less memory. On longer sequences (250-1,000 kbp), POASTA generated alignments with manageable runtime and memory, while SPOA failed.

POASTA includes several algorithmic innovations inspired by recent advances in pairwise and graph alignment. For example, POASTA takes inspiration from the recently published wavefront algorithm (WFA), a fast algorithm for pairwise alignment [15]. WFA similarly exploits exact matches between sequences and rapidly computes alignments by only considering the furthest-reaching points on DP matrix diagonals. However, their DP matrix diagonal formulation does not directly apply to graph alignment. In contrast to pairwise alignment, a stretch of exact matches between the query and the graph may span multiple diagonals in the DP matrix because of branches in

Table 2.2: Diversity of codons across 342 *M. tuberculosis* genomes representing the 450th amino acid in the *rpoB* gene. In three genomes, there was uncertainty about the second base in the triplet indicated by IUPAC code N (any base) or Y (C or T).

Codon	Amino acid	Count
TCG (reference)	S	232
TTG	L	103
TTT	F	2
TNG	-	2
GCG	A	1
TGG	W	1
TYG	-	1

the graph, complicating the definition of furthest-reaching points. While others have introduced variants of the WFA for graphs [25, 10], none support the gap-affine scoring model, which is preferred because it gives more biologically relevant alignments [20]. As an alternative to processing only the furthest-reaching points on a diagonal, POASTA uses its knowledge of graph topology, as stored in its superbubble index, to detect and prune alignment states that are not part of the optimal solution, thus speeding up alignment.

POASTA additionally takes inspiration from the recent read-to-graph aligner Astarix. Like POASTA, Astarix uses the A* algorithm for alignment, though with a different heuristic [19, 32]. The benefit of our minimum remaining gap cost A* heuristic is the simplicity of the required computation. All preprocessing can be done in $O(V + E)$ time, and all the necessary data is stored in $O(V)$ additional memory. The fast computation of the heuristic is important because the POA graph is updated at each iteration. Combined, these innovations can substantially reduce the number of computed alignment states, speeding up the construction of the complete MSA and enabling MSAs for longer sequences than was previously possible.

POASTA did not improve over SPOA in every scenario—it performed less well than SPOA in settings with high sequence diversity, where there are fewer stretches of exact matches for POASTA to exploit. In this situation, POASTA must explore more mismatch and indel states, increasing computation time. Though POASTA still computes fewer alignment states than SPOA, its runtime can become longer because the A* algorithm is less predictable and CPU cache-efficient than computing the full DP matrix row-by-row in a contiguous block of memory. Despite POASTA's higher compute time *per alignment state* compared to SPOA, the reduction in computed alignment states is often large enough to gain a net decrease in total runtime. To further develop our understanding of POASTA's performance characteristics, future work could include determining tight upper bounds on its runtime complexity, e.g., by adapting the arguments of Myers' $O(nd)$ algorithm for pairwise alignment to the sequence-to-graph alignment problem [33].

We envision several future improvements to the POASTA algorithm. POASTA could be expanded to support dual gap-affine penalties, enabling computing improved alignments in the presence of large indels [34]. Bi-directed variants of the A* algorithm, where the search for the shortest path is started from both the start and the end, could substantially improve POASTA's runtime with respect to sequence diversity [35]. A more informative A* heuristic, e.g., the recently published seed-heuristic [32] or one inspired by A*PA2 [36], could speed up alignment by improving estimates of the remaining alignment cost, improving the prioritization of alignment states to visit. Other strategies could be to utilize GPUs since massively parallel versions of A* exist [37]. Finally, we could combine the superbubble index with the Gwfa algorithm [25] to link diagonals across nodes and increase power to prune suboptimal alignment states.

2.5. Conclusions

We present POASTA, a novel optimal algorithm for POA. Through several algorithmic innovations, POASTA computed the complete MSA faster than existing tools in diverse bacterial gene sequence sets. It further enabled the creation of much longer MSAs, as demonstrated by successfully constructing MSAs from *M. tuberculosis* sequence sets with average sequence lengths of up to 1 Mbp. The algorithms and ideas presented here will accelerate the development of scalable pangenome construction and analysis tools that will drive the coming era of genome analysis.

Acknowledgments

We would like to thank Fabio Cunial and Ryan Lorig-Roach for their helpful discussions and their reviews of early versions of the manuscript. This project has been funded in part with federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Grant Number U19AI110818 to the Broad Institute.

Availability of data and materials

POASTA is written in Rust and available under the BSD-3-clause license at `https://github.com/broadinstitute/poasta` (DOI: 10.5281/zenodo.11153323). POASTA is available as both a standalone utility and a Rust crate that can be included as part of other software packages. The benchmark suite is written in Rust and Python and is available under the same license at `https://github.com/broadinstitute/poa-bench`. The data underlying this paper are included in the benchmark suite repository (DOI: 10.5281/zenodo.11153368).

References

1. Wang L and Jiang T. On the Complexity of Multiple Sequence Alignment. Journal of Computational Biology 1994 Jan; 1. Publisher: Mary Ann Liebert, Inc., publishers:337–48
2. Katoh K and Standley DM. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. Molecular Biology and Evolution 2013 Apr; 30:772–80
3. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 2004; 32:1792–7
4. Lee C, Grasso C, and Sharlow MF. Multiple sequence alignment using partial order graphs. Bioinformatics 2002 Mar; 18:452–64
5. Chin CS et al. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. en. Nature Methods 2013 Jun; 10. Number: 6 Publisher: Nature Publishing Group:563–9
6. Loman NJ, Quick J, and Simpson JT. A complete bacterial genome assembled de novo using only nanopore sequencing data. en. Nature Methods 2015 Aug; 12. Number: 8 Publisher: Nature Publishing Group:733–5
7. Vaser R, Sović I, Nagarajan N, and Šikić M. Fast and accurate de novo genome assembly from long uncorrected reads. Genome Res. 2017 May; 27:737–46

8. Lee C. Generating consensus sequences from partial order multiple sequence alignment graphs. Bioinformatics 2003 May; 19:999–1008
9. Chaisson MJP et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. en. Nature Communications 2019 Apr; 10. Number: 1 Publisher: Nature Publishing Group:1784
10. Holt JM et al. HiPhase: Jointly phasing small and structural variants from HiFi sequencing. en. Pages: 2023.05.03.539241 Section: New Results. 2023 May
11. Hickey G et al. Pangenome graph construction from genome alignments with Minigraph-Cactus. en. Nature Biotechnology 2023 May. Publisher: Nature Publishing Group:1–11
12. Garrison E et al. Building pangenome graphs. en. Pages: 2023.04.05.535718 Section: New Results. 2023 Apr
13. Gao Y et al. abPOA: an SIMD-based C library for fast partial order alignment using adaptive band. Bioinformatics 2021 Aug; 37:2209–11
14. Hart PE, Nilsson NJ, and Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 1968 Jul; 4. Conference Name: IEEE Transactions on Systems Science and Cybernetics:100–7
15. Marco-Sola S, Moure JC, Moreto M, and Espinosa A. Fast gap-affine pairwise alignment using the wavefront algorithm. Bioinformatics 2021 Feb; 37:456–63
16. Onodera T, Sadakane K, and Shibuya T. Detecting Superbubbles in Assembly Graphs. *Algorithms in Bioinformatics: 13th International Workshop*. Springer, 2013 :338–48
17. Rautiainen M and Marschall T. Aligning sequences to general graphs in O(V + mE) time. en. Pages: 216127 Section: New Results. 2017 Nov
18. Jain C, Zhang H, Gao Y, and Aluru S. On the Complexity of Sequence-to-Graph Alignment. Journal of Computational Biology 2020 Apr; 27. Publisher: Mary Ann Liebert, Inc., publishers:640–54
19. Ivanov P et al. AStarix: Fast and Optimal Sequence-to-Graph Alignment. en. *Research in Computational Molecular Biology*. Ed. by Schwartz R. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020 :104–19
20. Durbin R, Eddy SR, Krogh A, and Mitchison G. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. en. 1st ed. Cambridge University Press, 1998 Apr
21. Ukkonen E. Finding approximate patterns in strings. Journal of Algorithms 1985 Mar; 6:132–7
22. Hadlock F. An efficient algorithm for pattern detection and classification. *Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems - Volume 2*. IEA/AIE '88. New York, NY, USA: Association for Computing Machinery, 1988 Jun :645–53
23. Gärtner F, Müller L, and Stadler PF. Superbubbles revisited. Algorithms for Molecular Biology 2018 Dec; 13:16
24. Wu M and Eisen JA. A simple, fast, and accurate method of phylogenomic inference. Genome Biology 2008 Oct; 9:R151
25. Zhang H, Wu S, Aluru S, and Li H. Fast sequence to graph alignment using the graph wavefront algorithm. arXiv:2206.13574 [q-bio]. 2022 Jun
26. Jain C, Misra S, Zhang H, Dilthey A, and Aluru S. Accelerating Sequence Alignment to Graphs. *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Rio de Janeiro, Brazil: IEEE, 2019 May :451–61
27. Rautiainen M and Marschall T. GraphAligner: rapid and versatile sequence-to-graph alignment. Genome Biology 2020 Sep; 21:253
28. Suzuki H and Kasahara M. Acceleration of Nucleotide Semi-Global Alignment with Adaptive Banded Dynamic Programming. en. 2017 Apr
29. Ondov BD et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016 Jun; 17:132

30. Munir A et al. Identification and Characterization of Genetic Determinants of Isoniazid and Rifampicin Resistance in Mycobacterium tuberculosis in Southern India. en. Scientific Reports 2019 Jul; 9. Publisher: Nature Publishing Group:10283

31. Jamieson FB et al. Profiling of rpoB Mutations and MICs for Rifampin and Rifabutin in Mycobacterium tuberculosis. Journal of Clinical Microbiology 2020 Dec; 52. Publisher: American Society for Microbiology:2157–62

32. Ivanov P, Bichsel B, and Vechev M. Fast and Optimal Sequence-to-Graph Alignment Guided by Seeds. en. *Research in Computational Molecular Biology*. Ed. by Pe'er I. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022 :306–25

33. Myers EW. An O(ND) difference algorithm and its variations. en. Algorithmica 1986 Nov; 1:251–66

34. Sedlazeck FJ et al. Accurate detection of complex structural variations using single-molecule sequencing. en. Nature Methods 2018 Jun; 15:461–8

35. Champeaux D de. Bidirectional Heuristic Search Again. Journal of the ACM 1983 Jan; 30:22–32

36. Groot Koerkamp R. A*PA2: up to 20 times faster exact global alignment. en. 2024 Mar

37. Zhou Y and Zeng J. Massively Parallel A* Search on a GPU. en. Proceedings of the AAAI Conference on Artificial Intelligence 2015 Feb; 29. Number: 1

3

StrainGE: a toolkit to track and characterize low-abundance strains in complex microbial communities

Lucas R. van Dijk*, Bruce J. Walker*, Timothy J. Straub, Colin J. Worby, Alexandra Grote, Henry L. Schreiber IV, Christine Anyansi, Amy J. Pickering, Scott J. Hultgren, Abigail L. Manson, Thomas Abeel, Ashlee M. Earl

This chapter has been published in Genome Biology (2022). DOI: 10.1186/s13059-022-02630-0.
* Equal contribution

Abstract

Human-associated microbial communities comprise not only complex mixtures of bacterial species, but also mixtures of conspecific strains, the implications of which are mostly unknown since strain level dynamics are underexplored due to the difficulties of studying them. We introduce the Strain Genome Explorer (StrainGE) toolkit, which deconvolves strain mixtures and characterizes component strains at the nucleotide level from short-read metagenomic sequencing with higher sensitivity and resolution than other tools. StrainGE is able to identify strains at 0.1x coverage and detect variants for multiple conspecific strains within a sample from coverages as low as 0.5x.

3.1. Background

HUMAN-associated microbial communities include complex mixtures of bacterial species. Many of these species are renowned for their genomic and phenotypic plasticity. For example, strains of *Escherichia coli* share a core genome representing only about half of their genes [1] and cause distinct disease including diarrhea and urinary tract infections, or potentiate tumorigenesis, while other strains are able to co-exist with their host without causing overt illness [2, 3, 4]. Multiple distinct strains of the same species, often from genetically dissimilar phylogroups, frequently coexist within a single human gut community [5, 6], the implications of which are mostly underexplored due to the difficulties of studying strain-level variation from complex community samples.

While culture-based approaches have been a workhorse for dissecting strain-level diversity, these approaches can be slow and unfaithful to the true representation of strains, due to culturing bottlenecks that limit observed diversity, as well as the potential for evolution during culture [7]. Whole metagenome shotgun sequencing approaches offer less perturbed views of strain-level diversity, but require specialized computational tools. However, most current strain-level metagenomic data analytical tools (reviewed in Anyansi et al. [8]) were not designed to work at the low coverages typically found for many clinically relevant organisms in metagenomic samples, such as *E. coli* in the human gut [5]. Existing tools that aim to disentangle within-species strain mixtures include BIB [9], StrainEst [10], and DiTASiC [11], as well as the broader taxonomic profiling tools like Kraken2 [12] and GOTTCHA [13] when given an appropriate database. These tools rely upon a precomputed database of reference genomes, from which the best matches are reported for a sample (or set of samples). Thus, output from these tools is dependent upon database granularity and does not distinguish between distinct strains matching the same reference. Another class of tools characterizes and tracks strains based on single nucleotide variant (SNV) profiles along a single reference or a set of marker genes, including MIDAS [14], StrainPhlan [15] and ConStrains [16]. In the case of strain mixtures, MIDAS and StrainPhlan do not untangle the SNVs coming from different strains, while ConStrains attempts to link

SNVs with similar allele frequencies, though linking SNVs requires high strain coverage to be accurate [16, 17]. A third class of tools aims to recover strain-level variation after de novo metagenomic assembly, including DESMAN [17], inStrain [18] and STRONG [19]. Assembly approaches require higher sequence coverage than typically achieved for lower abundance members of a community. To our knowledge, none of these computational approaches work robustly at low coverages (<10x), accurately disentangle mixtures of same-species strains, and distinguish similar strains at the nucleotide level.

In order to be able to disentangle mixtures of low-abundance, clinically important strains within metagenomic data, we developed the Strain Genome Explorer (StrainGE) toolkit. In an advance over related tools, StrainGE works at exceptionally low sequence coverages (from 0.1x) to identify strains in a sample, and allows the user to characterize and compare strains across samples at the nucleotide level, with high resolution. We have extensively benchmarked StrainGE on synthetic data and compared it against other state-of-the-art strain detection tools. We also applied StrainGE to multiple clinical human gut metagenomic datasets, demonstrating StrainGE's ability to glean insights into biological systems that previous tools could not, including observing previously undetected persistence of low-abundance strains across time. Herein, we applied StrainGE to the analysis of clinically important strains of *E. coli* and *Enterococcus*, but StrainGE can be broadly applied to all community assemblages where same species bacterial strain dynamics are of interest.

3.2. Results

Strain Genome Explorer (StrainGE) toolkit

StrainGE is a toolkit for strain-level characterization and tracking of species (or genera) of interest from short read metagenomic datasets, tuned specifically to capture low abundance strains where data are scant. StrainGE has two key components: Strain Genome Search Tool (StrainGST), and Strain Genome Recovery (StrainGR). StrainGST sensitively reports reference genome(s) from a database that are most similar to the strain(s) in a sample. StrainGR analyzes short read alignments to a reported reference genome(s) to identify single nucleotide variants (SNVs) and large deletions (i.e., gaps in coverage) relative to the reference. Though StrainGST can be used as a standalone tool, the StrainGE tool suite, including StrainGR, enables sensitive nucleotide-level comparison and tracking of strains across multiple samples and provides insights into potential functional variation among individual strains.

In brief, StrainGST builds a database of high-quality reference genomes (e.g., RefSeq assemblies) from a species or genus of interest (Figure 3.1a), filtering them to remove highly similar genomes using a k-mer based clustering approach, with a tunable threshold (Table B.1). StrainGST's default database clustering threshold (0.9 Jaccard similarity) corresponds to an approximate ANI of 99.8% [20], which determines the minimum distance between reference genomes. To identify a similar reference(s) to the strain(s) within a sample and to estimate its relative abundance, StrainGST compares the k-mers in the sample to those of the database reference genomes (Figure 3.1)

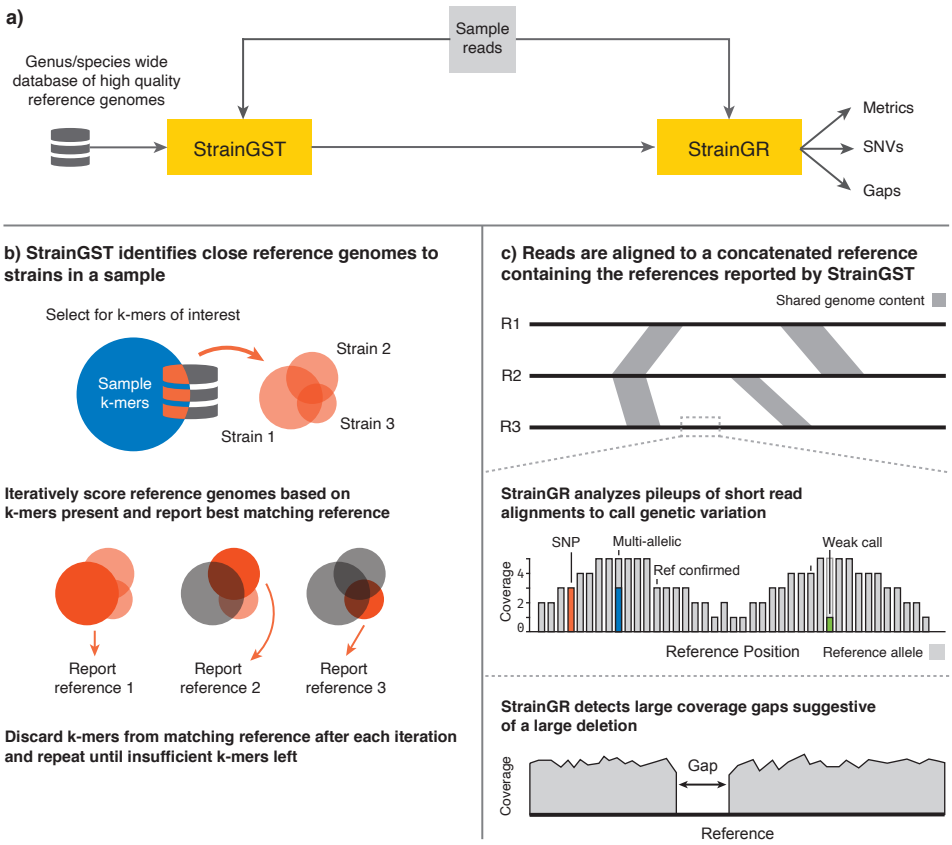

Figure 3.1: StrainGE is a toolkit to track, characterize and compare low-abundance strains in metagenomic samples. **(a)** Overview of StrainGE pipeline. StrainGST uses a database of high quality reference genomes to select those most similar to strains present in a metagenomic sample. StrainGR further characterizes SNVs and gaps that differ between references selected by StrainGST and the actual strain present in the sample. **(b)** At each iteration, StrainGST scores each reference strain by comparing the k-mer profile of the reference to the sample k-mers, reporting the reference closest to the highest abundant strain in the sample. The k-mers in the reported reference are removed from the sample and the process is repeated to search for lower-abundance strains, until there are insufficient k-mers. **(c)** StrainGR uses a short read alignment-based approach to characterize variation (SNVs and gaps) between the reference(s) identified by StrainGST and the metagenomic sample. Regions shared between the concatenated genomes (grey shaded areas) are detected and excluded from variant calling. Alleles are classified as "strong" or "weak". After applying rigorous QC metrics, positions in the reference are classified as i) "reference confirmed" (light grey; a single strong reference allele); ii) "SNV" (red; a single strong alternative allele); or iii) "multi-allelic" (blue; multiple strong alleles present, e.g. the blue allele together with the reference allele in grey). The position with a strong reference allele and a weak alternative allele (green; an allele with only limited support in the reads) is classified as "reference confirmed" because only the reference allele is considered strong at that position. The "callable" genome is defined as all positions within the reference with at least one strong allele call.

and iteratively ranks each reference using three key metrics, similar to QuantTB [21]: 1) the fraction of reference k-mers present in the sample, 2) the fraction of sample k-mer counts explained by a reference, and 3) the evenness of the distribution of shared k-mers along a reference. If the resulting score is above a tunable threshold, the reference strain is reported as present in the sample.

StrainGR was designed to complement StrainGST by providing a more detailed view of the nucleotide- and gene-level differences between a strain in a sample and its closest reference, which can be used to compare across samples having strains that match the same reference. StrainGR analyzes alignments of metagenomic sequencing data to each StrainGST predicted reference (Figure 3.1c). To ensure accurate SNV calls while maintaining sensitivity at low coverage, StrainGR employs stringent quality thresholds and heuristics to filter spurious alignments and reduce the number of incorrect calls.

To separate SNVs belonging to different strains, StrainGR creates a concatenated set of reference genomes, containing all references predicted by StrainGST in a sample or set of samples. It uses this reference set to align metagenomic reads and call variants. While close reference genome(s) generally result in more accurate alignments and variant calls [22], StrainGR still provides meaningful relationships when the reference is more distant, as would be the case in a smaller constructed database or with less well-studied organisms (Supplemental Text B.3; Figure B.1-B.4). To prevent assigning alleles incorrectly, StrainGR only calls variants in regions unique to a single reference by filtering out ambiguously aligned reads. In cases where StrainGST has identified distinct but closely related strains across samples, StrainGR can perform another, coarser round of reference clustering prior to concatenation in order to increase the amount of unique sequence for variant calling.

Variant calls can then be used to compare strains across samples. StrainGR compares positions within the "callable genome", or the set of positions with any reference or alternative allele supported by at least two good reads and >10% of the alignment pileup (Figure 3.1c). To perform a comparison, only "common callable" positions are considered, which represent the subset of the callable genome for a given reference that is shared by two samples. Strain relationships can be assessed using two key metrics: i) the Average Callable Nucleotide Identity (ACNI), or the percentage of common callable positions where both samples have a single identical base call; and ii) a "gap similarity" metric, as patterns of large deletions are often conserved between closely related strains, which can provide an orthogonal metric of strain similarity [23]. The ACNI and gap similarity values that define two samples as containing the same "strain" depend on the research question [7]. For the purposes of this manuscript, we consider two samples to contain the same strain if ACNI is $\geq$ 99.95%, which was based on our benchmarking of in silico *E. coli* spiked metagenomes. This threshold is stricter than our initial database clustering threshold of 99.8%, as samples matching the same reference in the database can contain different strains. As different studies may necessitate different strain definitions, we have intentionally made these thresholds easily tunable. With StrainGST able to accurately report close references to strains at coverages as low as 0.1x, and StrainGR able to track and characterize strains from 0.5x coverage, StrainGE enables sensitive analysis of very low-abundance strains, such as

typical *E. coli* relative abundances of <0.1% within a 3G metagenomic sample.

Benchmarking StrainGE on *Escherichia*

StrainGE was designed to be broadly applicable across different bacterial genera and species, including less well-studied species lacking numerous high quality reference genomes (Supplemental Text B.3). For benchmarking, we focused on *E. coli*, an evolutionarily and functionally diverse species. Despite their importance to human health, *E. coli* are typically found at low (<1%) relative abundance in diverse strain mixtures in human guts [5]. We first used StrainGST to construct an *Escherichia*-specific reference database by downloading all available complete *Escherichia* assemblies from NCBI RefSeq (929 assemblies, July 2019; Materials and Methods; Additional File 2). Because plasmids readily transfer between different genetic backgrounds of the same and/or different species (Supplemental Text B.3) [24], scaffolds labeled as plasmid, or those <1 Mbp were removed. After using the default clustering threshold corresponding to 99.8% ANI, the resulting StrainGST database contained 361 complete *Escherichia* chromosomes, comprising 341 *E. coli* and *Shigella* chromosomes representing all eight phylogroups [1], as well as 20 chromosomes from other *Escherichia* species.

StrainGE can accurately characterize strains and approximate ANI at coverages as low as 0.1x

To assess StrainGE's ability to detect and characterize strains, we first benchmarked each of StrainGE's components, StrainGST and StrainGR, individually. To benchmark StrainGST, we first used *in silico* constructed metagenomes that were spiked with sequences of known *Escherichia* strains at varying relative abundances. We compared StrainGST's ability to identify the correct close reference to that of two similar tools that depend on reference databases, BIB [9] and StrainEst [10]. While the databases used for StrainGST and StrainEst were identical, BIB's database construction method did not scale; thus, we used a smaller database with 20 genomes. StrainGST performed as well as, or better than, the other tools across all scenarios tested, including mixes of up to 3 strains at unequal abundances or 4 strains of equal abundance, and stood out strongly when strains were at very low abundance (Supplemental Text B.3; Figure B.5).

To further benchmark these three tools on real sequencing data of known strain composition, we created and sequenced a mock community containing approximately 99% human DNA and 1% *E. coli* DNA, representing a mixture of four distinct, previously sequenced strains with fully finished genomes mixed in unequal (approximately 80:15:4.9:0.1) relative abundances (Materials and Methods). StrainGST resolved the composition of this in vitro mock community without error (Table 1), while other tools reported two or more false positives (Table B.2).

To benchmark StrainGR's ability to call variants (SNVs and large deletions, or gaps), we used another set of *Escherichia*-spiked metagenomes, with reads simulated from in silico mutated reference genomes (99.9% ANI to reference; 5,000 SNVs). StrainGR accurately called SNVs and large deletions, for both single strain and mixture sam-

ples, providing key information to assess whether two samples shared a strain via the ACNI metric, StrainGR's approximation of ANI, and gap similarity (Supplemental Text B.3; Figure B.6). To assess the accuracy and robustness of ACNI, we generated spiked metagenomes similar to those described above, but we varied the number of SNVs introduced in silico (100%-99.9% ANI to reference; 0-5,000 SNVs) and used different metagenomic background samples, some with other *E. coli* strains present (Table B.4). Identical strains in different samples had high ACNI and gap similarity (Figure 3.2a) and StrainGR's ACNI across all strain pairs correlated strongly with true ANI, even though ACNI is based on unique regions, and ANI is based on the entire genome (Figure 3.2b). In this benchmark, the optimal ACNI threshold to classify two samples as having the same strain was 99.98% (Figure B.7), which is likely higher than the value we would expect from real data due to the artificially uniform distribution of SNVs in our synthetic benchmarks, and that the references used in benchmarking were also present in the StrainGST database. For analysis of real data in this manuscript, we chose a slightly lower value of 99.95%.

StrainGE was the most accurate at detecting shared strains at coverages as low as 0.5x

Having demonstrated that both StrainGST and StrainGR worked well, we aimed to assess StrainGE's complete pipeline to track strains across samples, including in strain mixtures. We compared StrainGE's ability to track strains to two recent, highly cited strain-tracking tools, MIDAS [14] and StrainPhlan [15]. Although MIDAS and StrainPhlan require high strain coverage to run to completion (5x and 10x, respectively), we were able to use manual tuning to allow these tools to accommodate our lower coverage benchmarks (Materials and Methods). We excluded ConStrains [16] because of its high coverage requirements which could not easily be tuned [16, 17]. To assess the sensitivity of these tools to distinguish between similar strains, we generated pairs of spiked metagenomic samples, each containing one or more *Escherichia* strains at 0.1x-10x coverage. Similar strain pairs were derived from the same reference genome, but with a different set of 5,000 random SNVs introduced in silico into each strain's genome (Figure 3.3a-b). This resulted in each strain having 99.9% ANI to the reference and each strain pair having 99.8% ANI to one another. This identity level should result in strain pairs matching the same StrainGST reference but still distinguishable by StrainGR.

At 10x coverage, MIDAS and StrainPhlan performed comparably using tuned (Figure 3.3) and default (Figure B.8) settings. While StrainGE and MIDAS performed well across all scenarios at high coverage, StrainPhlan performed poorly on mixes because it only reported a single SNV profile for each sample. For lower coverage scenarios, StrainGE consistently outperformed the manually tuned versions of MIDAS and StrainPhlan (Figure 3.3). For single strain samples, StrainGE perfectly matched strain pairs down to 0.1x coverage, with MIDAS performing comparably (Figure 3.3c). StrainPhlan performed marginally at 1x coverage, and still was unable to run to completion at coverages lower than 1x. For simple mixtures (Figure 3.3d), only StrainGE and MIDAS correctly matched most pairs, because StrainPhlan was unable to disentangle mixes. StrainGE was the only tool that was able to generate results across the whole range of

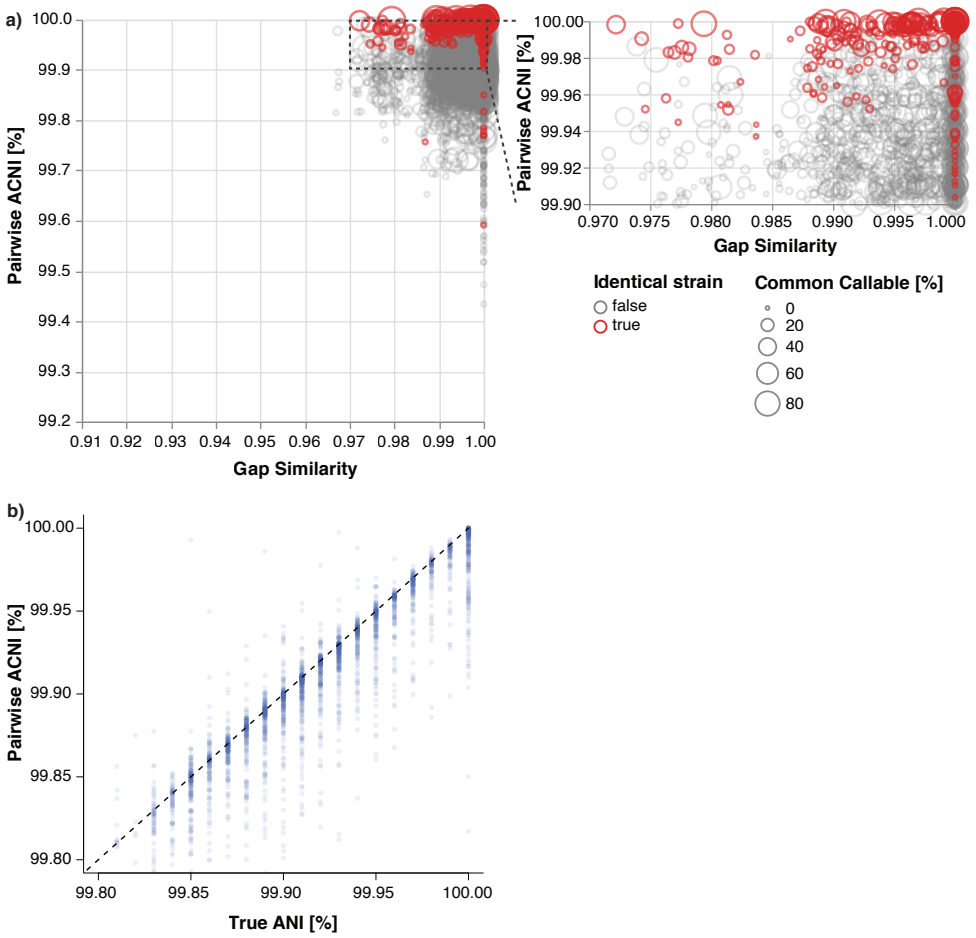

Figure 3.2: StrainGR discriminates between highly similar strains and reports ACNI which strongly correlates with true ANI. **(a)** For all synthetic sample pairs with the same StrainGST reference called, the Jaccard gap similarity index and pairwise ACNI are plotted. Circle size indicates the percentage of the reference genome that was callable across both strains being compared. Red circles indicate comparisons between identical strains. **(b)** For all pairs, the true ANI between spiked isolates is plotted against the ACNI, as estimated by StrainGR. The dashed line indicates parity between these metrics. Pairs of strains could have 0-10,000 SNV differences.

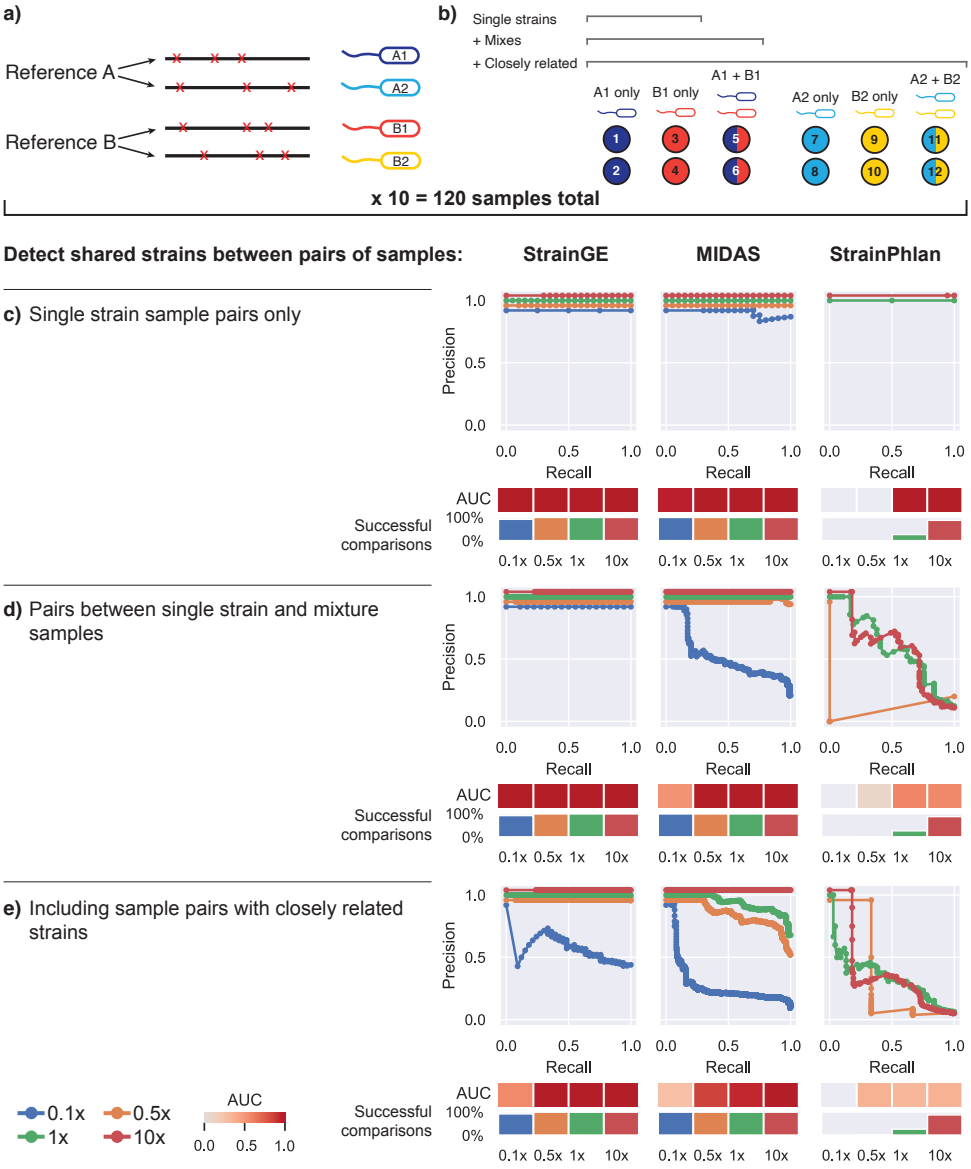

Figure 3.3: StrainGE is the only tool that can detect strain sharing at coverages as low as 0.5x. **(a)** Depiction of how synthetic *Escherichia* genomes were generated from randomly selected NCBI RefSeq genomes to create sets of closely related strains (e.g., A1/A2 and B1/B2) for spike in experiments. **(b)** Depiction of how spiked metagenomes were created using synthetic genomes from (a). Each circle represents a spiked metagenome. The color of the circle indicates which synthetic strain was included: single color circles indicate spiked metagenomes containing a single synthetic strain, and two color circles indicate spiked metagenomes containing two synthetic strains mixed at equal proportions. **(c-e)** Precision-recall curves for each tool and coverage 0.1x-10x, when given the task to detect which sample pairs contain identical strains. The area under the curve (AUC) is depicted as a heatmap below. The "successful comparisons" bar plot indicates the percentage of sample pairs for which a comparison was possible (i.e., tools ran to completion for both samples). **(c)** Limiting to single-strain samples from distinct references. **(d)** Including samples with two strains, but limited to strains from distinct references. **(e)** Including samples with closely related strains.

Table 3.1: StrainGST was the only tool that correctly identified the known composition of a mock community.

True strain (phylogroup)	**Predicted strains**		
	StrainGST	StrainEst	BIB
E. coli SEC460 (A)	✓	✓	*E. coli* K-12 GM4792 99.24%
E. coli UTI89 (B2)	*E. coli* UM146 99.95%	*E. coli* UM146 99.95%	*E. coli* H105 98.49%
E. coli Sakai (E)	*E. coli* 149 99.89%	*E. coli* 149 99.89%	*E. coli* 108 99.97%
E. coli 24377A (B1)	✓	✓	*E. coli* S40 99.01%
		✗ *E. coli* APEC IMT5155 99.51%	✗ *S. flexneri* G1663 97.97%
		✗ *E. coli* RM14721 99.44%	✗ *E. coli* LHM10-1 98.12%
			✗ *E. coli* MSHS 133 97.67%
			✗ *S. dysenteriae* 80-547 97.71%
			✗ *E. coli* IMT16316 97.39%
			✗ *S. dysenteriae* ATCC 12039 97.08%

A check mark indicates that the exact strain was present in our database and correctly identified. A strain name indicates that the exact reference was not in the reference database, but the closest available reference was correctly identified (along with its approximate ANI to the actual strain). A strain name with an "X" indicates a false positive strain identified by the tool that was not present in the mock community. Percentages near strain names indicate approximate ANI to the closest true strain. Relative abundances for each strain are listed in Table B.2.

coverages, scoring almost perfectly down to the lowest tested coverage of 0.1x. When we included samples containing very closely related pairs (Figure 3.3e), StrainGE and MIDAS performed well down to 0.5x coverage, but StrainPhlan could not distinguish between closely related strains, even at 10x coverage, likely due to its reliance on marker genes which comprise only a small fraction of the genome. Whereas StrainGE relied on a mean callable genome of 74% ± 13% (at 10x coverage), StrainPhlan relied on marker genes which only covered on average 1.4% ± 0.3% of the references.

StrainGE achieved this high sensitivity with comparable runtime to MIDAS and StrainPhlan, and its memory usage was well within the range of modern cluster systems or powerful personal computers (Figure B.9). Another key advantage of StrainGE over the other tools is its ability to link a strain in a sample to its specific close reference genome reported by StrainGST, which places an observed strain within the known phylogenetic structure of the reference database (Figure B.10). In contrast, the SNV

profiles outputted by StrainPhlan (based on marker genes) or MIDAS (compared to a single built-in *E. coli* reference) do not offer convenient phylogenetic placement.

In real metagenomic data, StrainGE identifies low-abundance strains and can track strains across samples, including in strain mixtures

StrainGE can identify lower abundance instances of persistent strains previously undetectable by other tools

In order to assess StrainGE's utility to characterize strains from real-world samples, we examined its performance, using default parameters with our *Escherichia* reference database, on a previously published metagenomic dataset of 27 longitudinally collected stool samples from a patient with Crohn's disease, upon which MIDAS was run to delineate *E. coli* strains [25]. MIDAS identified seven dominant "strain types" ("ST1" - "ST7") that varied in abundance over time. Each belonged to a distinct multi-locus sequence type (MLST) and represented one of five *E. coli* phylogroups. StrainGE showed good concordance with results from MIDAS for all high-abundance strains (>10% abundance) (Table 3.2). For the two calls that disagreed, our StrainGST database lacked representatives for the two MLSTs reported by MIDAS. However, StrainGE selected the next closest reference, which we confirmed by comparing the whole genome sequence from a cultured representative of ST1 [25] to our reference database.

StrainGST also identified seven distinct strains missed by MIDAS (Figure 3.4a). While the majority of these were secondary strains found to coexist with a dominant strain predicted by MIDAS, StrainGST also predicted strains at timepoints where MIDAS called none (time points 6-10). In most of these cases, the strains were at 1% relative abundance and were also detected by MIDAS at higher abundance in other time points (e.g., ST3; dark green), lending credence to their existence in these samples and suggesting that some strains were more persistent over time than previously predicted (Figure 3.4a).

We ran StrainGR on all datasets using a concatenated reference including 10 out of 14 total references reported by StrainGST to ensure each genome had at least 20% unique genome content (Materials and Methods; Figure 3.4b). SNV and gap patterns predicted by StrainGR showed that the majority of strains matching the same reference had strikingly high pairwise ACNI (>99.96%) and gap similarity (>0.97) (Figure 3.4c,d), which were within the range of those of same-strain sample pairs in our simulations (Figure 3.2a). However, StrainGR results from strains matching the *E. coli* 118UI (dark green) reference stood out. While 118UI-like strains from samples 3 and 4 had ACNI and gap similarity relationships that were on par with what we observed in same-strain simulations, all other comparisons fell outside of this range, suggesting that this individual carried a mixture of 118UI-like strains in their gut over time that were closely related, but not necessarily the same with respect to gene content and single nucleotide variation (Figure B.11).

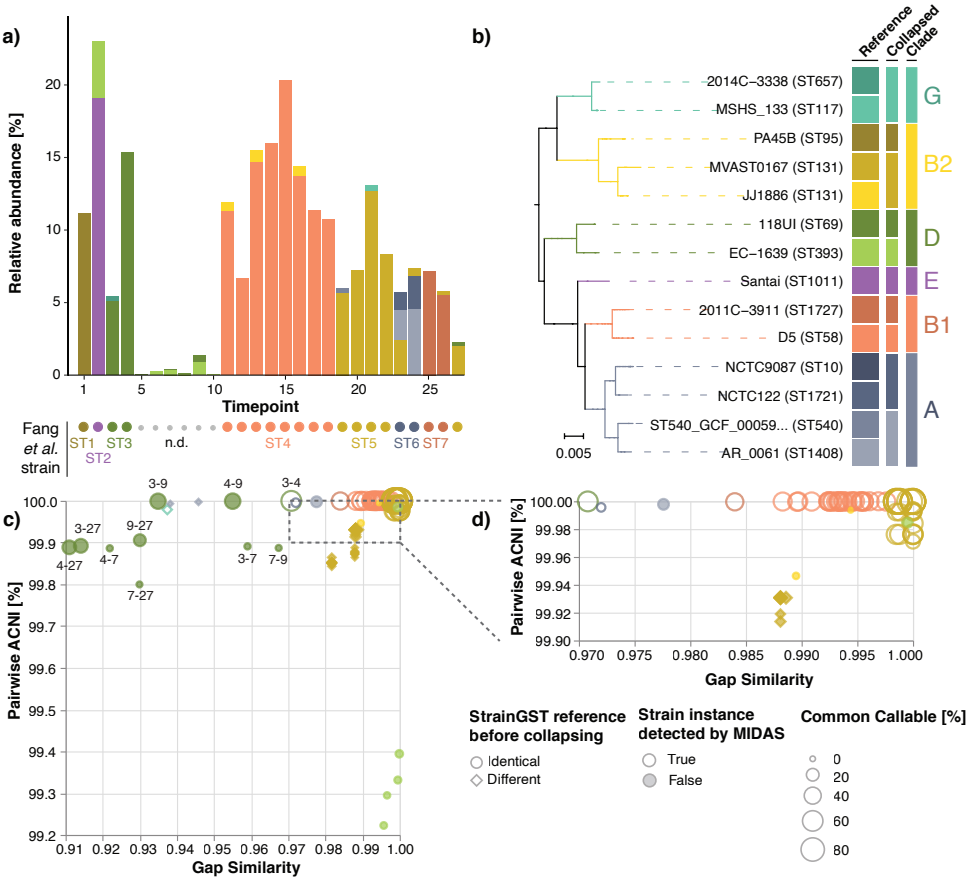

Figure 3.4: StrainGE identified previously undetected low-abundance strains in longitudinal samples from an individual with Crohn's disease. **(a)** Stacked barplot showing the relative abundances of StrainGST calls for each of 27 longitudinal stool metagenomes from Fang et al. [25]. Circles indicate the strain detected in Fang et al., colored by its StrainGST counterpart and labeled using the ST designations (ST1-ST7) assigned by Fang et al. Small grey circles indicate samples where no strain was predicted in Fang et al.; these are labeled with "n.d." **(b)** Single-copy core phylogeny of the 14 StrainGST reference genomes with close matches to strains across samples. Colors are based on the reference's clade; see column "Clade". "Collapsed" column indicates which reference was selected as a representative for subsequent StrainGR analysis, when two or more references shared more than ~99.2% ANI. **(c)** For all sample pairs matching the same collapsed reference, the Jaccard gap similarity index and pairwise ACNI are plotted. Circles indicate comparisons where the predicted reference was the same before collapsing, and diamonds indicate cases where the predicted reference before collapsing was different. Sizes of shapes indicate the percentage of the reference genome that was callable across both strains being compared. Filled in shapes indicate whether this strain instance was undetected by MIDAS. Dark green circles are labelled with the timepoints compared. **(d)** Zoomed in view of the upper right corner of figure c).

Table 3.2: The strains predicted by MIDAS match the dominant strains predicted by StrainGE.

Strain (timepoints)	**MIDAS**		**StrainGE**		
	MLST	*E. coli* phylogroup	MLST	*E. coli* phylogroup	Most abundant strain
ST1 (1)	95	B2	95	B2	PA45B*
ST2 (2)	1629†	E	1011	E	Santai
ST3 (3-4)	69	D	69	D	118UI
ST4 (11-18)	58	B1	58	B1	D5
ST5 (19-22, 27)	131	B2	131	B2	MVAST0167
ST6 (23, 24)	409†	A	1408	A	AR_0061
ST7 (25, 26)	1727	B1	1727	B1	2011C-3911

* The actual strain corresponding to ST1 (3_2_53FAA) was whole-genome sequenced by Fang et al. [25]. PA45B and 3_2_53FAA share 99.9% average nucleotide identity based on whole-genome comparative genomics analysis.
† MLST profile was not represented by any reference genome in the StrainGE database. StrainGST predicted the closest reference within the StrainGE database, which was within the same phylogroup.

StrainGE accurately and sensitively identified a low-abundance, persistent strain of *E. coli* in longitudinal stool samples from a woman with recurrent urinary tract infection

Although the results of StrainGE on the Fang et al. [25] dataset highlighted its ability to resolve strains present at low abundance, the overall *E. coli* relative abundances in these samples were significantly higher (median 7.9%; range 0.05%-27%) than those typically seen in the human gut. Thus, we also tested StrainGE on 12 stool metagenomes having more typical *E. coli* relative abundances (median 0.55%; range 0.006%-17.4%), which originated from a single individual with a history of recurrent urinary tract infection (rUTI) over the span of a year. Given that the gut is a known important reservoir for UTI-causing *E. coli* [26], it was of interest to trace gut *E. coli* strain dynamics and their relationship with UTI.

StrainGST detected a total of five distinct strains of *E. coli* (Figure 3.5a), including a recurrent strain detected in over half of samples. The persistent strain, an *E. coli* 1190-like strain from phylogroup D, had a median relative abundance of only 0.6% (range 0-1.2%) and was detected even in samples that were composed of multiple *E. coli* strains, including at very low (20-fold less) abundance relative to another strain (Figure 3.5a). Despite its low abundance, we were able to confirm that all *E. coli* 1190-like strains had extremely high ACNI (>99.95%) and gap similarity (>0.98) (Figure 3.5b), in line with the identities observed for same-strain benchmarking (Figure 3.2a), suggesting that this strain, also the causative agent of this individual's rUTI, persisted long-term in their gut.

Further, StrainGR output enabled us to look closely at the locations and identities of SNVs and genes within gaps relative to the reference. For example, we consistently identified a large gap across all time points encoding a prophage found in the original reference, but apparently lacking in the *E. coli* 1190-like strain in this individual (Fig-

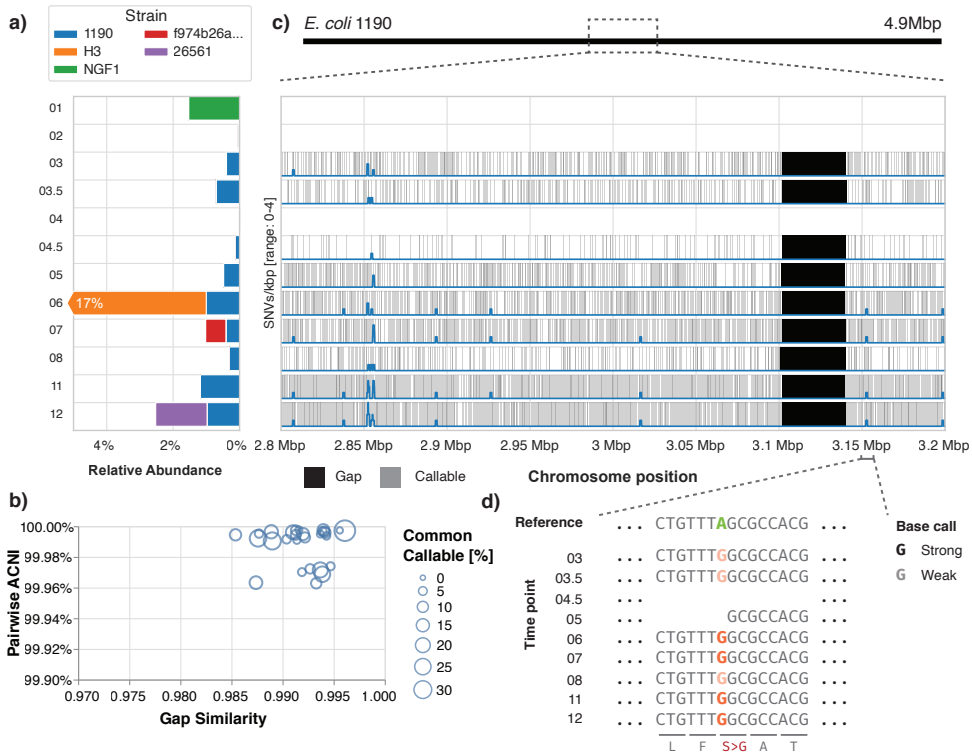

Figure 3.5: StrainGE detected a long-term, persistent strain of *E. coli* in a woman with rUTI. **(a)** Relative abundances predicted by StrainGE are shown for all *E. coli* strains detected. **(b)** For all sample pairs containing a strain matching to *E. coli* 1190, plot shows pairwise ACNI and gap similarity scores. Size of the circle indicates the percentage of the common callable genome. **(c)** Zoom in on a region of the chromosome of *E. coli* 1190. Grey shaded areas indicate "callable" regions, where StrainGR had enough read data to make a strong allele call. Predicted gaps are shaded black. The blue line represents the number of SNVs per 1,000 bp, observed in at least 3 samples. **(d)** Further zoom-in representing a region where StrainGR identified a nonsynonymous SNV that was consistently detected across all 1190-like strains.

ure 3.5c). Using StrainGR output that included both strong and weak variant calls (see Figure 3.1c for strong vs. weak calls; Materials and Methods), we were able to track 839 variant sites across samples, where the corresponding allele was strongly called in at least one sample, and weakly called in at least five samples (e.g., a nonsynonymous SNV in the gene cydC; Figure 3.5d). At each of the 839 variant sites, the called allele was identical across all time points, except for three sites where another secondary weak allele was called, further supporting the persistence of a single UTI-causing strain.

StrainGE accurately recapitulated known strain-level diversity from metagenomes and traced strains from mother to child

To demonstrate StrainGE's applicability to other bacterial genera, we selected a previously published dataset investigating the impact of mode of delivery on the infant gut microbiome, including transmission and carriage of opportunistic pathogens from the *Enterococcus* genus [27]. Shao et al. longitudinally followed 596 babies (and 175 mothers) by collecting stool samples that were then whole metagenome shotgun sequenced and cultured for pathogens, including 451 enterococci that were then whole genome sequenced. This dataset allowed us to evaluate StrainGE's ability to report on i) the relationships between enterococcal strains predicted directly from metagenomes in comparison to those calculated from the genomes of cultured isolates, and ii) mother and child strain sharing. Furthermore, this dataset allowed us to evaluate StrainGE's ability to predict and compare strains across samples using a sparser database, as there were fewer than a third as many RefSeq complete *Enterococcus* genomes than for *Escherichia*.

We built a 163-member StrainGST database representing references from 80 *E. faecium*, 39 *E. faecalis* and 44 other enterococcal species (Materials and Methods; Additional File 2) and ran StrainGE on all 1,679 stool metagenomes. StrainGE identified strain relationships that were very similar to those Shao et al. obtained using bacterial isolate comparisons. For example, the species distributions were roughly similar (Table B.3) and nearly half (42%) of references predicted by StrainGST belonged to one of the five major *E. faecalis* lineages previously identified (Figure 3.6a). The pairwise ACNI distributions for strains matching these references mirrored the tree topology (Figure B.12), and across the whole data set pairwise ACNI correlated strongly with ANI between corresponding isolates (Pearson's r=0.96; Figure 3.6b; Materials and Methods).

Shao et al. used StrainPhlan [15] to predict instances of mother-to-child strain sharing, including 7 *E. faecalis* and 2 *E. faecium* transmission events. Though no direct comparison of transmission predictions could be made (sample names were not reported), we hypothesize that StrainGE's predictions would be more accurate since StrainPhlan's marker genes covered only 3.6% ± 1.7% of reported *Enterococcus* genomes, while StrainGE's callable genome was on average 39% ± 33%. Using StrainGE, we identified 17 mother-baby pairs for which StrainGST reported the same reference, of which six had sufficient common callable genome to calculate ACNI. Three pairs had ACNI <99.7% and three had ACNI near 100%, including an example at 99.999% (Figure 3.6c) suggesting that there were at least three instances of mother-baby strain sharing that we could confidently call based on our "same strain"

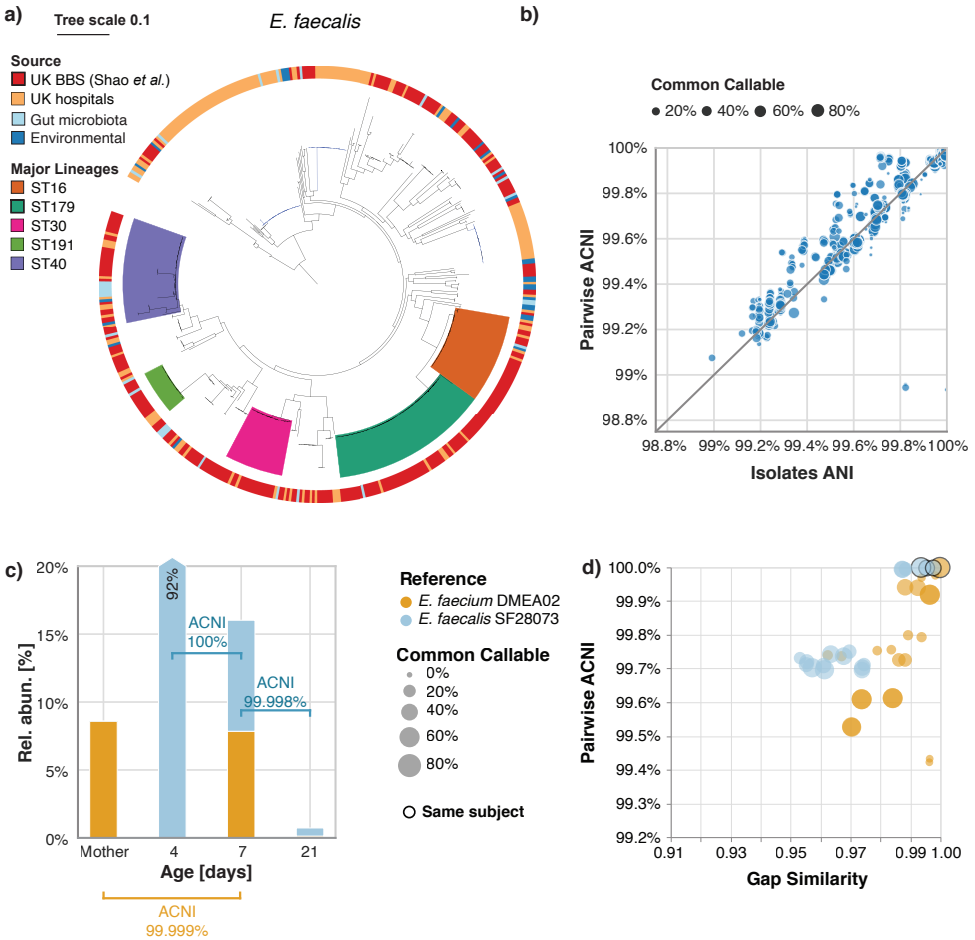

Figure 3.6: StrainGE recapitulates strain-diversity among bacterial isolates using metagenomic data only. **(a)** Single-copy core phylogenetic tree of *E. faecalis* isolates from the UK Baby Biome Study (UK BBS) (n = 282) in the context of isolates from other public UK hospitals (n = 168), human gut microbiota (n = 28), or other environmental sources (n = 27). Five major lineages were identified, represented by ST16, ST179, ST30, ST191 and ST40. Tree republished with permission from Shao et al. [27]. **(b)** Scatterplot relating ANI between isolates (x-axis) to StrainGE's computed ACNI between metagenomes from which the isolates were derived (y-axis). **(c)** Barplot showing StrainGST predicted references and their relative abundances (y-axis) for strains present in metagenomic samples from a mother and her child taken over several days (x-axis). Strains matching the same reference are shown in the same color. Lines connecting bars are labelled with StrainGR computed ACNI. **(d)** For all pairs of samples with a strain close to either *E. faecium* DMEA02 (yellow) or *E. faecalis* SF28073 (blue), ACNI (y-axis) and gap similarity are plotted (x-axis). Circles with a black border represent pairs of samples from the same subject (or its mother). Size of the circle represents the percentage of common callable genome.

ACNI threshold of 99.95%. Comparisons of strains matching the same reference from other mothers or babies revealed that they generally had considerably lower ACNI and gap similarity (Figure 3.6d).

3.3. Discussion

The ability to discern strain-level variation from primary specimens—where the species of interest may be at low abundance—can transform our understanding of species populations, ecologies, and transmission patterns. We have shown that our novel tool suite, StrainGE, is easy to use for ultra sensitive detection of strains in primary specimen metagenomes. StrainGE uses both k-mer and alignment analysis to characterize sample strain genomes, including their i) closest matching reference, which places them phylogenetically, ii) relative abundance, and iii) estimated ANI (ACNI) to other strains, which can be achieved even at very low coverage levels, with more detailed information about specific variants and cross-sample comparisons becoming available as coverage increases. StrainGE can provide nucleotide level resolution for individual bacterial strains or strain mixes that are present at 0.1% relative abundance e.g., 0.5x coverage for a 5Mb genome within 3Gb of sequencing reads. StrainGE provides a substantial advance over previously published tools, which i) were not designed to work at these low coverages [16] ii) report only overall consensus SNV profiles for a mixture [14, 15], or iii) do not offer nucleotide-level resolution [9, 10, 11].

In addition to demonstrating good performance on an extensive array of benchmarking samples, we showed that StrainGE provided insights into the strain-level dynamics of bacteria in three real-world sample sets. For a patient with Crohn's disease, StrainGE identified co-existing strains and strains at timepoints missed by another popular strain-tracking tool. StrainGE similarly was used to identify the long-time gut carriage of a low abundant UTI-causing *E. coli* strain, which we could track via stereotypical gene absence and SNP patterns, reported by StrainGE, that could be discerned even when other strains were present. Finally, using metagenomic data from primary stool specimens, StrainGE was able to recapitulate relationships among *E. faecalis* previously observed using whole genome sequencing of isolates and phylogenetic reconstruction, as well as provide strong evidence for transmission of *E. faecium* strains from mothers to their children. For this vignette, we used an ACNI threshold that we empirically determined to represent the same strain from in silico experiments. However, the measures that define "same" versus "different" strains will depend upon the research question and the species being evaluated [7]. StrainGE provides a compendium of outputs for assessing relationships between strains in detail, which can be used to evaluate appropriate thresholds for any system.

While we demonstrated StrainGE on a narrow set of bacterial species, StrainGE is designed to be broadly applicable to any genus or species, with a wide range of database sizes. While a dense database is generally preferred because the accuracy of variant calls improves with genetically closer references [22], our benchmarking showcased that StrainGST and StrainGR combined can return accurate information about strain relationships, even when few reference genomes are available. Further-

more, the default database clustering threshold of 99.8% is tunable to adjust for the number of references StrainGST considers since, for example, a very dense database could cause StrainGST to report different, but closely related references for two samples containing the same strain. To balance these two factors, we included a tool "prepare-ref" in the StrainGE suite, which performs an additional coarser round of clustering of StrainGST-determined references for a set of samples in order to select a smaller set of representatives prior to running StrainGR. This step increases the total amount of unique content across references to be considered in ACNI calculations and enables direct comparisons of more strains with respect to their nucleotide and gap similarities.

While StrainGST and StrainGR were designed to work together, both tools can work in isolation and provide useful stand alone output. StrainGST with a dense database can provide fast phylogenetic placement of strains. Though not shown here, this also works on whole genome sequence data from bacterial isolates, providing a quick snapshot of phylogenetic relationships without needing to perform reference alignments or other more time-consuming phylogenomic pipelines. StrainGR could be used without StrainGST when good quality assemblies are available for strains present within a mixed community dataset. For example, long read sequencing and assembly of isolates or even whole metagenomes from a select number of time points could provide high quality substrates for StrainGR evaluations of strains in short read time series data. Though competitive and filling a niche left behind by other strain-tracking tools, StrainGE has several limitations. It evaluates the relationships between strains using only unique regions of reference genomes, is unable to detect new genes that occur in strain genomes that are not present in its closest matching reference, and currently only works with Illumina data. Furthermore, StrainGE is currently not designed to phase SNVs from multiple strains matching the same reference in the same sample. In this case, StrainGR will output evidence for multiple alleles, but the frequencies of which cannot be robustly compared to link alleles together at the coverages under which StrainGE was designed to operate.

3.4. Conclusions

Here, we present StrainGE, a novel suite of tools to characterize conspecific strains in complex microbial communities. We have demonstrated its accuracy using benchmarks and have shown that it represents a major advance over other published tools. Using three clinical metagenomic time series, we demonstrated its ability to yield insights into biological systems that previous tools could not, including the persistence of low-abundance strains across time. StrainGE's sensitivity at very low coverages (0.1x and higher) will help to accelerate our understanding of the role of strain-level variation in shaping ecological and disease processes.

StrainGE is installable through bioconda and available at `https://github.com/broadinstitute/strainge`.

3.5. Materials and Methods

Strain Genome Explorer toolkit algorithms

StrainGST: Strain Genome Search Tool

StrainGST is a k-mer based tool used to identify specific strain(s) of a species in a metagenomic sample. StrainGST computes a reference database of previously sequenced strains from this species, and uses it to report close reference genomes to strains present in a metagenomic sample along with their relative abundances. The references reported by StrainGST can be used as input to StrainGR to further characterize genetic variation found within the metagenomic sample.

Creating a StrainGST database. A StrainGST database is constructed from a set of high quality sequenced reference genomes for a single species or genus, such as all complete reference genomes in NCBI RefSeq. From this set of genomes, StrainGST generates a database of k-mer profiles, using a sliding window (window size k) to traverse each genome and count the frequency of each k-mer. To reduce memory usage and computation time, a minhash technique (similarly to Mash [20] is applied to keep 1% of the k-mers with the lowest hashes.

StrainGST next performs clustering to remove highly similar genomes from the reference set. In order to track and compare genomic variation across related samples, StrainGR must be able to align reads to a common reference genome across different sample sets. Therefore, the references reported by StrainGST should not be too closely related, or each sample could end up matching distinct yet closely related references, making comparisons difficult. StrainGST computes pairwise Jaccard similarities using each reference genome's k-mer set, performing single linkage clustering using a Jaccard similarity threshold of τ, and picking a single representative genome for each cluster to include in the reference set. StrainGST selects the genome with the highest mean similarity to all other genomes in that cluster. This process ensures that the k-mer similarities between remaining genomes in the database are all lower than τ. Additionally, to remove genomes from the database that are highly similar to another genome, but that may have lower Jaccard similarity due to the presence of large indels, StrainGST removes genomes where 99% or more k-mers overlapped with those from another genome.

Identifying strains present in a sample. StrainGST uses this database to identify the closest reference genome(s) to the strain(s) present within a sample (Figure 3.1). First, all reads in the sample are k-merized, resulting in the k-mer set K_{sample}. The algorithm then selects for k-mers from the species of interest by taking the intersection between the sample k-mer set and that of the reference database for the species of interest (Figure 3.1b), excluding k-mers not associated with the target species.

StrainGST then uses these k-mers to identify the reference genome(s) with the best k-mer matches to the sample using an iterative process. In each iteration, StrainGST scores each reference genome in the database against the remaining k-mers in K_{sample}

in order to find the reference with the best score, which is reported to the user as the reference with the strongest evidence of being present. The scoring system is described in detail below. If no reference strain is identified that scores above a threshold θ (adjustable by a command line option), the algorithm is terminated. The default value for θ (0.02) was optimized to maximize sensitivity while minimizing false positives. In each iteration, k-mers corresponding to the reference selected are removed from the sample k-mer set in order to enable identification of secondary strains in the next iteration. This process continues until either no strain is reported or the maximum number of iterations is reached (default of 5).

Scoring metric for selecting matching reference strains. To determine which reference strain to report in each iteration, we calculate a score for each reference strain using a combination of three metrics based on: 1) the fraction of matching k-mers in the reference; 2) the fraction sample k-mer counts that could be explained by this reference genome; and 3) the evenness of the distribution of matching k-mers across the genome.

(1) Fraction of matching k-mers in the reference (f). This metric represents the fraction of distinct k-mers in reference j that is present in the sample and has a value between 0 and 1, where 1 would indicate all k-mers of this reference are present in this sample.

$$\begin{aligned} K' &= K_{sample} \cap K_j \\ f &= \frac{|K'|}{|K_j|} \end{aligned} \tag{3.1}$$

K_{sample} represents all k-mers in the sample, K_j represents all k-mers in reference j, and K' represents the set of k-mers both present in the reference and in the sample.

(2) Fraction of sample k-mers that could be explained by this reference (a). To give more weight to reference genomes that are similar to higher abundance strains in the sample, StrainGST calculates the fraction of database k-mers remaining in the sample that could be explained by the k-mers in this reference:

$$a = \frac{\sum_{i \in K'} c_i}{\sum_{i \in K_{sample}} c_i} \tag{3.2}$$

c_i represents the count of k-mer i in the sample. Note that we include k-mer counts, rather than using the fraction of distinct k-mers, which gives more weight to reference strains with high average depth of coverage. This metric has a value between 0 and 1.

(3) Evenness (e). To quantify whether the matching k-mers are evenly distributed across the reference genome, rather than being found predominantly in a small region (e.g., due to a horizontal gene transfer event, or conserved regions attracting reads from different species), we defined the evenness score. First, we assumed that the coverage across the genome follows a Poisson distribution. The rate parameter λ_j of the Poisson distribution specifies the average depth of coverage across the whole genome:

$$\lambda_j = \frac{1}{|K_j|} \sum_{i \in K_j} \frac{c_i}{d_{ij}} \tag{3.3}$$

Here, c_i represents the count of k-mer i in the sample, and d_{ij} represents the count of k-mer i in reference strain j. If X is the random variable that indicates how many reads cover a position, then using the Poisson distribution, the probability of observing x reads at a position is:

$$P(X = x) = \frac{\lambda_j^x \exp(-\lambda_j)}{x!} \tag{3.4}$$

The probability of observing 0 reads at a position is then $P(X = 0) = \exp(-\lambda_j)$. The probability of observing at least one read at a position is [28]:

$$P(X > 0) = 1 - P(X = 0) = 1 - \exp(-j) \tag{3.5}$$

This probability also represents the expected fraction of the genome covered by at least one read given a certain average depth of coverage. The evenness score describes how well the observed fraction of the genome covered by at least one read (which is estimated using the fraction of matching k-mers in the reference defined earlier), matches the expected fraction of the genome covered by at least one read when assuming a Poisson distribution for the depth of coverage:

$$e = \frac{f}{1 - \exp(-j)} \tag{3.6}$$

This score will be close to 1 if the observed fraction of the genome with at least one read matches the expected value for a certain average depth of coverage (assuming a Poisson distribution). It will be closer to zero if only small portions of the genome are well covered. A value higher than 1 indicates that the observed fraction of the genome with at least one read is higher than the expected fraction of the genome with at least one read. To bound this score between 0 and 1, StrainGST uses the minimum of e and its reciprocal:

$$e' = \min(e, \frac{1}{e}) \tag{3.7}$$

Finally, we combined these three metrics together in order to obtain the final score:

$$\text{score} = f \cdot a \cdot e'^2 \tag{3.8}$$

At each iteration, the reference strain with the highest score represents the best match to the highest abundant strain remaining in the sample and is reported to the user.

StrainGR: Strain Genome Recovery

The StrainGR pipeline consists of: 1) building a concatenated reference based on reference strains reported by StrainGST; 2) aligning reads to the concatenated reference; 3) analyzing read alignments to call SNVs and large deletions; and 4) using these variant calls to analyze gene content or track strains across multiple samples.

Preparing a concatenated reference. To analyze a set of related samples together, such as a longitudinal series, StrainGR concatenates a single, unified set of representative references present across the whole dataset. This can facilitate comparisons of alignments or genomic variation across a set of samples, which may contain different strain mixes at different time points. Use of the concatenated reference allows reads with an allele unique to a particular strain to be aligned to the genome of the correct reference strain, thus helping disentangle reads from mixture samples. Genomes from the same species, however, will share conserved genomic regions (i.e., housekeeping and other core genes), where the aligner will be unable to place reads unambiguously within the concatenated reference. StrainGR detects and excludes these conserved regions from variant calling.

In order to minimize conserved regions where StrainGR is unable to call variants, it is important to select a set of reference strains that are not too closely related, which could result in a large fraction of the concatenated reference genome being marked as shared. To construct a concatenated set of references with an optimal degree of similarity, StrainGR includes a tool called prepare-ref that analyzes StrainGST output from a set of samples (e.g., a longitudinal set from a single patient) and generates a concatenated reference ready for use with StrainGR, optionally performing another round of clustering at a stricter threshold to prevent too-closely related genomes from being included. By default, the stricter clustering threshold is set to a Jaccard similarity of 0.7 (99.2% estimated ANI).

Read alignment and filtering. The reads from a metagenomic sample are then aligned to the concatenated reference using BWA-MEM [29], removing read pairs with 1) improper pairing; 2) clipped alignment; or 3) implied insert size smaller than the read length. In order to identify shared regions within the concatenated reference which should be excluded from variant calling, StrainGR tracks the number of "multi-mappable" read alignments (those which map equally well at multiple locations) at each position in the reference. When the majority of aligned reads at a position are multi-mappable, StrainGR excludes this position from variant calling. We rely on BWA's "XA" SAM tag to obtain a read's alternative alignment locations, so aligners other than BWA are not currently supported by StrainGR.

In addition to excluding multi-mappable regions, StrainGR also excludes regions with abnormally high coverage (greater than threshold τ), likely due to genes highly conserved across genera which attract nonspecific reads from other members of the microbial community. was chosen such that the probability of observing a depth of coverage higher than τ is 1×10^{-7} assuming a Poisson distribution. This value results in a threshold of 10x coverage when the mean coverage depth across the genome is 1x, and a threshold of 20x when the mean is 5x.

SNV calling. StrainGR analyzes read alignments to identify single-nucleotide variants (SNVs) between a specific strain within a metagenomic sample and its closest reference genome identified by StrainGST. To filter likely sequencing errors, bases with an Illumina Phred base quality score <5 are ignored by default. An allele is considered strong if the sum of base quality scores supporting that allele is i) higher than

50 (roughly equivalent to having at least two high-quality supporting reads) and ii) at least 10% of the total sum of base quality scores of all alleles at that genomic position. If an allele is present but doesn't match these criteria, it is considered weak. StrainGR stores weak evidence for use when tracking a strain across multiple samples—if a particular strain is highly abundant in some samples, with many strong SNP calls, then weak calls can be useful to discern that allele in low abundance samples from the same sample set.

Based on the observed alleles, StrainGR classifies a genomic position as either "reference confirmed", "SNV" or "multiple alleles". If a position has a single strong allele call, and that allele is the same as the reference, the position is classified as "reference confirmed". A position with a single strong allele call that is different from the reference is classified as a SNV. Any position with multiple strong allele calls (whether they match the reference or not) is classified as "multiple alleles".

To estimate the overall degree of similarity between the strain in the sample and its closest reference, StrainGR computes an estimate of average nucleotide identity (ANI) using StrainGR SNV calls: the average callable nucleotide identity (ACNI) is the percentage of positions marked as "reference confirmed" out of all positions with a single strong allele call.

Large deletion predictions. StrainGR also analyzes the read alignments to identify large deletions present in a specific strain within a sample, as compared to its closest reference identified by StrainGST. Consecutive positions in the reference genome over a specified length (by default 5,000 bp; 2 genes) with no aligned reads could indicate a large deletion. To account for situations with low coverage across the genome (<1x), StrainGR employs a simple heuristic that exponentially scales the threshold for the length of such regions at lower coverages; thus, only longer gaps can be detected at lower coverages. If λ is the average depth of coverage along the genome, and Φ is the unadjusted threshold, then the adjusted minimum size of a "gap" is:

$$\Phi' = 1 - \exp(-\lambda) \tag{3.9}$$

Large deletions are used to assess whether particular genes are absent from the strain in a sample. In addition, the overall pattern of deletions across the genome for the strain in a longitudinal sample set can be used as a strain "fingerprint" to track a particular strain across samples.

Strain comparisons across samples. To assess whether the strains in two samples are the same (or very closely related), we compare both SNV calls (via pairwise ACNI) and patterns of large deletions. StrainGR calculates pairwise ACNI by dividing the number of positions where both samples have the same strong allele by the total number of positions where both samples have a single strong allele. To compare the pattern of predicted deletions between two samples, StrainGR calculates the Jaccard similarity: if G_1 is the set of positions not marked as a large deletion in sample 1, and G_2 is the set of positions not marked as a large deletion in sample 2, then the gap similarity l is defined as

$$l = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|} \tag{3.10}$$

Benchmarking StrainGE using simulated data and mock communities

Spiked metagenome generation

Unless otherwise noted, all synthetic metagenomes used for benchmarking were generated as follows: reads were simulated from the relevant genomes using ART [30] and merged with reads subsampled from a genuine metagenomic data set without detectable *E. coli* (accession SRS014613) as per MetaPhlan2 [31] and StrainGST, up to a fixed depth of 3 Gb. At this depth, strain coverages of 0.1x, 0.5x, 1x and 10x corresponded to relative abundances of 0.016%, 0.083%, 0.16%, and 1.6%, respectively, assuming a 5 Mb *E. coli* genome.

StrainGST database for *Escherichia*

For construction of the *Escherichia* reference database, all complete *Escherichia* genomes available in NCBI RefSeq were downloaded in July 2019 (929 genomes total; Additional File 2). All tools required to construct the StrainGST database are included in the StrainGE suite (kmer counting, clustering, and database construction). The full database with 361 *Escherichia* genomes uses 7.3 Gb of disk space.

In order to set StrainGST's default clustering threshold, we benchmarked its ability to correctly identify single strain and two-strain mixes using the metagenomic spike-in methods described below, using synthetic reads generated from 200 randomly selected *E. coli* genomes spiked into subsets of real metagenomic samples devoid of *E. coli*, to a total of 3 Gb. For the single-strain benchmarks, 200 samples were generated with 10x, 1x, 0.5x, and 0.1x coverage of each of the selected *E. coli* genomes (800 samples total). For the 2-strain mix benchmarks, 100 random 2-strain combinations from the set of 200 selected *E. coli* genomes were spiked in at each combination of 10x, 1x, 0.5x, and 0.1x coverage (10 coverage combinations, 1000 samples total). The 1800 benchmark cases were run using database clustering thresholds of 0.95, 0.90, 0.85, and 0.80 Jaccard k-mer similarity, corresponding to Mash distance ANIs of 99.89%, 99.77%, 99.63%, and 99.49%, respectively. For each threshold, we measured precision, recall, and F1 score for strain identification, with true positives being only those cases in which StrainGST identified the closest reference strain to the true strain as measured by Jaccard k-mer similarity. The clustering threshold of 0.90 generated the best combined results in each of the three metrics (Table B.5).

Phylogenetics and MLST typing of genomes in the *Escherichia* reference database

A single copy core (SCC) phylogeny was generated for the entire database of reference genomes. In brief, SynerClust [32] was used to generate clusters of orthologous genes

(orthogroups). A concatenated alignment was generated for all single-copy, core orthogroups using MUSCLE [33]. A phylogenetic tree was constructed using FastTree v2.1.8 [34]. Phylogenetic trees were visualized using iTol [35].

MLST designations for each reference genome were predicted with the tool mlst (https://github.com/tseemann/mlst). Sequence types reported were based on the Achtman scheme. *E. coli* clade/phylogroup designation was determined using ClermonTyping (https://github.com/A-BN/ClermonTyping). For cases when there were missing or conflicting results between predicted typing and MASH groups, the clade designation for a given genome was selected based on where it was located in the SCC phylogeny with respect to unambiguous genome to clade designations.

Creation of four-strain *E. coli* mock community

Four phylogenetically distinct *E. coli* strains - H10407 (clade A), E24337A (clade B1), UTI89 (clade B2), and Sakai (clade E) - were cultured separately overnight at 37ºC in 2 mL of liquid LB media shaking at 200 rpm. The bacterial number in each culture was estimated via optical density and then combined at a ratio of 80% H10407, 15% UTI89, 4.9% Sakai, and 0.1% E24337A. Genomic DNA was then extracted from this mock community using the Qiagen MagAttract DNA Isolation Kit (Hilden, Germany), following manufacturer's protocols. In two separate tubes, human genomic DNA was then added to the extracted *E. coli* DNA for final ratios of 99% human / 1% *E. coli* (weight / weight). Sequencing data for this mock community has been submitted to NCBI's Sequence Read Archive (SRA) under bioproject PRJNA685748 (biosample SAMN17091845).

Comparison of tools for tracking specific strains across samples using simulated sets of related samples

We compared the ability of StrainGE, StrainPhlan [15], and MIDAS [14] to track strains across samples. We performed strain tracking comparisons across ten sets of twelve spiked metagenomes, where each set of twelve was structured similarly in terms of strain content (Figure 3.32a-b). For each set, we randomly picked two *Escherichia* reference genomes (A and B) from NCBI RefSeq complete, and derived two different but closely related synthetic strains from each reference by introducing ~5,000 random SNVs (99.9% ANI) uniformly across the genome. We spiked reads generated from these synthetic genomes into a real metagenome to generate samples containing these strains in different combinations (Figure 3.3b), at 0.1x, 0.5x, 1x and 10x coverage.

For each data set, we assessed strain similarity metrics calculated by each tool, to determine whether the tool could match i) the identical strain found in different samples (i.e., strain A in sample 1 and 2; Figure 3.3); ii), strains found either in mixtures or single-isolate samples (i.e., strain A in sample 1, 2, 5, and 6; Figure 3.3d); or iii) closely related strains (i.e., the ability to distinguish strain A1 from strain A2; Figure 3.3e). In each case, we compared the tools' predictions to the known strain content of each sample to calculate true positives (TP), false positives (FP), and false negatives (FN). For each tool, we varied the threshold (discussed in detail below) for determining shared strains in order to plot precision-recall curves.

Detecting shared strains using StrainGE. For each sample, we ran the complete StrainGE pipeline: StrainGST was run to identify the closest reference genomes, and StrainGR was run on a sample-specific concatenated reference to call genetic variation. To detect shared strains, we collected all samples predicted to match to the same StrainGST reference, and computed a pairwise ACNI matrix for strain comparisons with at least a 0.5% callable genome. The similarity matrix was transformed to a distance matrix by computing 1-ACNI, and transformed to a genetic distance using the Jukes-Cantor model [36]. If a pair of samples did not share any predicted close reference genomes, we set the distance between those samples to the maximum integer value. To plot the precision-recall curve, we varied the genetic distance threshold that determines when strains are considered the same.

Detecting shared strains with StrainPhlan. We ran StrainPhlan on each sample, using the tool's marker gene database v295 (Jan 2019). Using the marker gene SNV profiles for each sample, StrainPhlan computed the pairwise sample distance matrix using Kimura's two parameter model [37] (as suggested in their user manual). To plot the precision-recall curve, we varied the genetic distance threshold that determines when samples share a strain, as performed for StrainGE. To tune StrainPhlan for lower coverage levels, we ran it using `--relaxed-parameters`.

Detecting shared strains with MIDAS. We ran MIDAS v1.3.2 (database version v1.2) with default parameters. MIDAS includes a strain tracking tool that is first "trained" by giving it a single sample from each patient in a cohort. This training step identifies unique SNV markers for each patient. For our benchmarking, we "trained" MIDAS on samples containing a single strain (sample 1 for strain A1, sample 3 for strain B1, sample 7 for strain A2, and sample 9 for strain A2). (This likely helped the tool in benchmarking since, in a real world scenario, it is likely unknown whether a training sample contains a single strain.) Next, MIDAS compares these SNV markers to alleles in other samples and assesses how much they overlap. To plot precision-recall curves, we varied the percentage of overlapping markers between two samples that serves as a threshold to determine whether two samples share a strain. To tune MIDAS for lower coverage levels we ran its `merge_snvs.py` script with `--all_snvs --all_samples` and its `strain_tracking.py` script with `--min_reads 1`.

Evaluating the ability of StrainGR to quantify strain sharing in distinct metagenomic backgrounds

In order to determine how well StrainGE metrics recapitulated genetic relationships between strains, we generated another set of spiked metagenomic samples, spiked with varying quantities of *E. coli* reads from real, previously sequenced isolates. Ten stool metagenomes were randomly selected from the Human Microbiome Project [5] (Table B.4). The randomly selected samples contained *E. coli* at relative abundances between 0.005% and 0.9%; no two samples contained the same *E. coli* strain based on StrainGST output. Ten complete genome sequences of *E. coli* isolates, distinct from those identified in the background metagenomes, were selected from NCBI RefSeq

database. For each isolate, ten variants were created by generating random mutations, such that the ANI to the original reference ranged from 99.9% to 99.99% at increments of 0.01%. Each reference and variant (110 in total) were spiked into at least two randomly chosen distinct metagenomic backgrounds at coverage levels of 0.1x, 0.5x, 1x, 2x or 5x. A total of 300 synthetic samples were generated, with 350 pairs containing an identical strain in a distinct background. All spiked samples were analyzed with StrainGE; all sample pairs with a matching StrainGST reference were compared using StrainGR. StrainGST hits corresponding to strains present in background samples were not considered further. The ACNI was calculated for every pair.

Evaluation of StrainGE on longitudinal, clinical metagenomic samples

Metagenomic time series from a patient with Crohn's disease

We downloaded from the UCSD Qiita database (https://qiita.ucsd.edu/; Additional File 3) 27 metagenomic data sets representing stool longitudinally collected from a single individual with Crohn's Disease [25]. We ran the full StrainGE pipeline on each sample, using our *Escherichia* database and default parameters, to identify and analyze *E. coli* strains. For StrainGR, to ensure each genome had sufficient unique content, we constructed a concatenated reference using StrainGE's builtin prepare-ref tool, which performed another round of clustering of the StrainGST reported references at a default threshold of 99.2% ANI. The resulting reference contained 10 out of 14 total reported references (Figure 3.4b; phylogroup G, B2 and A). For pairwise strain comparisons, we only included samples where the common callable percentage of the genome was >0.5%.

Metagenomic sequencing of longitudinally collected stool

Twelve longitudinally collected stool samples were extracted with Chemagen Kit CMG-1091 (Baesweiler, Germany). Libraries were generated with NexteraXT (Illumina, San Diego, CA, USA) and sequenced in paired-end mode on an Illumina HiSeq 2500 (101 bp length) and/or Illumina HiSeq X10 (151 bp length). Short-read sequencing data was submitted to the Sequence Read Archive (SRA) with Bioproject accession PRJNA400628 (Additional File 3). We ran the full StrainGE pipeline on each sample, using our *Escherichia* database and default parameters, to identify and analyze *E. coli* strains. For pairwise strain comparisons, we only included samples where the common callable percentage of the genome was >0.5%.

Characterization of *Enterococcus* strain diversity across a large cohort of babies

We downloaded data from by Shao et al. [27] from ENA, including 1679 metagenomes (accession ERP115334) and all isolate samples tagged as *Enterococcus* (accession ERP024601). We ran StrainGE on each metagenomic sample, using our *Enterococcus* database. To compare StrainGE's ACNI to true ANI between the corresponding isolate genomes, we ran StrainGST on the raw isolate reads to identify a close reference

genome, aligned the isolate reads to this reference using BWA-MEM [29], and used Pilon [38] to call variants. To compute the ANI between each pair of isolates that matched the same reference, we compared reference and alternative alleles called by Pilon where both samples had a single base call. For pairwise strain comparisons using StrainGR in the corresponding metagenomic samples, we only included pairs with a common callable genome >0.5%.

Declarations

Availability of data and materials

The data generated and analyzed for this publication is available in SRA, under BioProject accession PRJNA400628 (patient with recurrent UTI), and PRJNA685748 (four *E. coli* strains mock community). The data from an individual with Crohn's disease was previously published by Fang et al. [25] and available on QIITA (Artifact IDs 54476 and 54537). The data analyzed to investigate *Enterococcus* strain diversity in the gut microbiomes of babies was previously published by Shao et al. [25] and available from ENA (ERP115334 and ERP024601).

StrainGE is an open source tool written in Python and C++ released under the BSD 3-clause license, available at https://github.com/broadinstitute/strainge [39]. StrainGE is built on top of several existing Python libraries, including NumPy [40] and SciPy [41]. Analysis scripts to generate the figures in this publication are available at https://github.com/broadinstitute/strainge-paper [42].

Ethics approval and consent to participate

An adult volunteer with a history of UTI was recruited at Washington University, St. Louis under an IRB-approved protocol (principal investigator, Hing Hung (Henry) Lai). Informed consent was obtained.

Consent for publication

Not applicable

Competing interests

BJW is an employee of Applied Invention (Cambridge, MA). No other authors declare competing interests.

Funding

This project has been funded in part with Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Grant Number U19AI110818 and U01AI095776, and National Institute of Diabetes and Digestive and Kidney Disease, National Insti-

tutes of Health, Department of Health and Human Services, under Grant Number R01DK121822.

Acknowledgments

We would like to thank Christopher Desjardins, Theodore Pak, Wen-Chi Chou, Rauf Salamzade and Ryan Bronson for helpful discussions.

Additional Files

Supplemental data tables are available online at `https://doi.org/10.1186/s13059-022-02630-0`.

References

1. Touchon M et al. Phylogenetic background and habitat drive the genetic diversification of Escherichia coli. en. PLOS Genetics 2020 Jun; 16. Publisher: Public Library of Science:e1008866
2. Pleguezuelos-Manzano C et al. Mutational signature in colorectal cancer caused by genotoxic pks + E. coli. en. Nature 2020 Apr; 580. Number: 7802 Publisher: Nature Publishing Group:269–73
3. Leimbach A, Hacker J, and Dobrindt U. E. coli as an All-Rounder: The Thin Line Between Commensalism and Pathogenicity. en. *Between Pathogenicity and Commensalism*. Ed. by Dobrindt U, Hacker JH, and Svanborg C. Current Topics in Microbiology and Immunology. Berlin, Heidelberg: Springer, 2013 :3–32
4. Schreiber HL et al. Bacterial virulence phenotypes of Escherichia coli and host susceptibility determine risk for urinary tract infections. Sci. Transl. Med. 2017 Mar; 9
5. Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. Nature 2012 Jun; 486:207–14
6. Tenaillon O, Skurnik D, Picard B, and Denamur E. The population genetics of commensal Escherichia coli. en. Nature Reviews Microbiology 2010 Mar; 8. Number: 3 Publisher: Nature Publishing Group:207–17
7. Van Rossum T, Ferretti P, Maistrenko OM, and Bork P. Diversity within species: interpreting strains in microbiomes. Nat. Rev. Microbiol. 2020 Jun
8. Anyansi C, Straub TJ, Manson AL, Earl AM, and Abeel T. Computational Methods for Strain-Level Microbial Detection in Colony and Metagenome Sequencing Data. English. Frontiers in Microbiology 2020; 11. Publisher: Frontiers
9. Sankar A et al. Bayesian identification of bacterial strains from sequencing data. Microb Genom 2016 Aug; 2:e000075
10. Albanese D and Donati C. Strain profiling and epidemiology of bacterial species from metagenomic sequencing. Nat. Commun. 2017 Dec; 8:2260
11. Fischer M, Strauch B, and Renard BY. Abundance estimation and differential testing on strain level in metagenomics data. Bioinformatics 2017 Jul; 33:i124–i132
12. Wood DE, Lu J, and Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biology 2019 Nov; 20:257
13. Freitas TAK, Li PE, Scholz MB, and Chain PSG. Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Research 2015 May; 43:e69
14. Nayfach S, Rodriguez-Mueller B, Garud N, and Pollard KS. An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography. Genome Res. 2016 Nov; 26:1612–25

15. Truong DT, Tett A, Pasolli E, Huttenhower C, and Segata N. Microbial strain-level population structure and genetic diversity from metagenomes. Genome Res. 2017; 27:626–38

16. Luo C et al. ConStrains identifies microbial strains in metagenomic datasets. Nat. Biotechnol. 2015 Oct; 33:1045–52

17. Quince C et al. DESMAN: A new tool for de novo extraction of strains from metagenomes. Genome Biol. 2017; 18. Publisher: Genome Biology:1–22

18. Olm MR et al. inStrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains. en. Nature Biotechnology 2021 Jun; 39. Number: 6 Publisher: Nature Publishing Group:727–36

19. Quince C et al. STRONG: metagenomics strain resolution on assembly graphs. Genome Biology 2021 Jul; 22:214

20. Ondov BD et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016 Jun; 17:132

21. Anyansi C et al. QuantTB – a method to classify mixed Mycobacterium tuberculosis infections within whole genome sequencing data. BMC Genomics 2020 Jan; 21:80

22. Bush SJ et al. Genomic diversity affects the accuracy of bacterial single-nucleotide polymorphism–calling pipelines. en. GigaScience 2020 Feb; 9. Publisher: Oxford Academic

23. Darmon E and Leach DRF. Bacterial Genome Instability. en. Microbiology and Molecular Biology Reviews 2014 Mar; 78. Publisher: American Society for Microbiology Section: Review:1–39

24. Acman M, Dorp L van, Santini JM, and Balloux F. Large-scale network analysis captures biological features of bacterial plasmids. en. Nature Communications 2020 May; 11. Number: 1 Publisher: Nature Publishing Group:2452

25. Fang X et al. Metagenomics-Based, Strain-Level Analysis of Escherichia coli From a Time-Series of Microbiome Samples From a Crohn's Disease Patient. Front. Microbiol. 2018 Oct; 9:2559

26. Jones-Freeman B et al. The microbiome and host mucosal interactions in urinary tract diseases. en. Mucosal Immunology 2021 Feb. Publisher: Nature Publishing Group:1–14

27. Shao Y et al. Stunted microbiota and opportunistic pathogen colonization in caesarean-section birth. en. Nature 2019 Oct; 574. Number: 7776 Publisher: Nature Publishing Group:117–21

28. Lander ES and Waterman MS. Genomic mapping by fingerprinting random clones: A mathematical analysis. en. Genomics 1988 Apr; 2:231–9

29. Li H and Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009; 25:1754–60

30. Huang W, Li L, Myers JR, and Marth GT. ART: A next-generation sequencing read simulator. Bioinformatics 2012; 28:593–4

31. Truong DT et al. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. en. Nature Methods 2015 Oct; 12. Number: 10 Publisher: Nature Publishing Group:902–3

32. Georgescu CH et al. SynerClust: a highly scalable, synteny-aware orthologue clustering tool. Microbial Genomics 2018; 4. Publisher: Microbiology Society,

33. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 2004; 32:1792–7

34. Price MN, Dehal PS, and Arkin AP. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. en. PLOS ONE 2010 Mar; 5. Publisher: Public Library of Science:e9490

35. Letunic I and Bork P. Interactive Tree Of Life (iTOL) v4: recent updates and new developments. en. Nucleic Acids Research 2019 Jul; 47. Publisher: Oxford Academic:W256–W259

36. Jukes TH and Cantor CR. Evolution of Protein Models. *Mammalian Protein Metabolism: Volume III, Volume 3*. 1969

37. Kimura M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. en. Journal of Molecular Evolution 1980 Jun; 16:111–20

38. Walker BJ et al. Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One 2014; 9

39. Straub T, Walker B, Dijk Lv, canyansi, and Desjardins C. broadinstitute/StrainGE: v1.2. 2021 Dec

40. Harris CR et al. Array programming with NumPy. en. Nature 2020 Sep; 585. Number: 7825 Publisher: Nature Publishing Group:357–62

41. Virtanen P et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. en. Nature Methods 2020 Mar; 17. Number: 3 Publisher: Nature Publishing Group:261–72

42. Dijk Lv. broadinstitute/strainge-paper: Paper resubmission. 2021 May

4

Longitudinal multi-omics analyses link gut microbiome dysbiosis with recurrent urinary tract infections in women

Colin J. Worby, Henry L. Schreiber IV, Timothy J. Straub, **Lucas R. van Dijk**, Ryan A. Bronson, Benjamin S. Olson, Jerome S. Pinkner, Chloe L. P. Obernuefemann, Vanessa L. Muñoz, Alexandra E. Paharik, Bruce J. Walker, Christopher A. Desjardins, Wen-Chi Chou, Karla Bergeron, Sinéad B. Chapman, Aleksandra Klim, Abigail L. Manson, Thomas J. Hannan, Thomas M. Hooton, Andrew L. Kau, H. Henry Lai, Karen W. Dodson, Scott J. Hultgren, Ashlee M. Earl

My contributions to this chapter are mainly centered around the *E. coli* dynamics in women with rUTI as compared to a healthy control group. I contributed the identification of "persister" strains, which are more likely to originate from phylogroup B2 or D, and frequently cause UTIs. I contributed the finding that persister strains were rarely cleared by antibiotics. I contributed the finding that women in both the rUTI and control groups harbor phylogenetically similar strains and have similar dynamics. Furthermore, I developed and extensively benchmarked and validated StrainGE (Chapter 3) for this project, using additional isolates from matched stool and rectal samples.

This chapter has been published in Nature Microbiology (2022). DOI: 10.1038/s41564-022-01107-x.

Abstract

Recurrent urinary tract infections (rUTIs) are a major health burden worldwide, with history of infection being a significant risk factor. While the gut is a known reservoir for uropathogenic bacteria, the role of the microbiota in rUTI remains unclear. We conducted a year-long study of women with ($n = 15$) and without ($n = 16$) history of rUTI, from whom we collected urine, blood and monthly faecal samples for metagenomic and transcriptomic interrogation. During the study 24 UTIs were reported, with additional samples collected during and after infection. The gut microbiome of individuals with a history of rUTI was significantly depleted in microbial richness and butyrate-producing bacteria compared with controls, reminiscent of other inflammatory conditions. However, *Escherichia coli* gut and bladder populations were comparable between cohorts in both relative abundance and phylogroup. Transcriptional analysis of peripheral blood mononuclear cells revealed expression profiles indicative of differential systemic immunity between cohorts. Altogether, these results suggest that rUTI susceptibility is in part mediated through the gut–bladder axis, comprising gut dysbiosis and differential immune response to bacterial bladder colonization, manifesting in symptoms.

4.1. Introduction

URINARY tract infections (UTIs) are among the most common bacterial infections worldwide and a significant cause of morbidity in females, with uropathogenic *Escherichia coli* (UPEC) being the primary causative agent [1]. One of the strongest risk factors for UTI is a history of prior UTIs [2], but the biological basis and risk factors for long-term recurrence remain unclear in otherwise healthy women. 20-30% of women diagnosed with a UTI will experience a recurrent UTI (rUTI), with some suffering six or more per year. Over one million women in the United States are referred to urologists each year because of rUTIs, and the rapid spread of antibiotic resistance in uropathogens is making treatment more challenging.

The gut is a reservoir for UPEC, and UTIs most commonly arise via the ascension of UPEC from the gut to the urinary tract [3, 4, 5]. Recent studies have explored the 'gut microbiota-UTI axis', showing that uropathogen abundance in the gut is a risk factor for UTI in kidney transplant patients [6], and that a 'bloom' in uropathogen gut abundance may precede infection [7]. Other studies have demonstrated differences in gut microbiome composition associated with children suffering UTIs [8], and with kidney transplant patients developing bacteriuria [9], compared to healthy controls. Furthermore, fecal microbiota transplants to treat *Clostridium difficile* infections may have the collateral effect of reducing the frequency of rUTI [10, 11], suggesting that perturbation of the gut microbiota can modulate rUTI susceptibility.

It is increasingly accepted that the gut microbiota can play a role in conditions affecting distal organs—for instance, the gut-brain and gut-lung axes are the subject of ongoing research [12, 13, 14, 15]. However, the gut-bladder axis—the spectrum

of direct and indirect interactions between gut flora and the bladder immune and/or infection status—remains uncharacterized, and the role of the gut microbiota in rUTI susceptibility is not well understood. No study has yet ascertained whether: i) gut dysbiosis is associated with rUTI susceptibility; ii) rUTI women have unique uropathogen dynamics within and between the gut and the bladder; or iii) microbiome-mediated immunological differences may be linked to rUTI susceptibility, as seen in other diseases [16].

Here, we present results from the UTI microbiome (UMB) project, a year-long clinical study of women with a history of rUTI and a matched cohort of healthy women. Our unique longitudinal study design allowed us to explore the importance and interdependence of the gut microbiota and *E. coli* strain dynamics in rUTI, susceptibility to infection, and host immune responses that may impact these dynamics. Using multi-omic techniques, we determined that: i) compared to healthy controls, women with a history of rUTI had a distinct, less diverse gut microbiota, depleted in butyrate producers and exhibiting characteristics of low-level inflammation; ii) differential immunological biomarkers suggest rUTI women may have a distinct immune state; iii) *E. coli* strains were transmitted from the gut to the bladder in both cohorts, though no UTI symptoms occurred in healthy controls; and iv) UTI-causing *E. coli* strains often persistently colonized the gut and were not permanently cleared by repeated antibiotic exposure. Thus, susceptibility to rUTI is in part mediated through a syndrome involving the gut-bladder axis, comprising a dysbiotic gut microbiome with reduced butyrate production and apparent alterations of systemic immunity. Our work shows that UPEC strains persist in the gut despite antibiotic treatment, which itself may exacerbate gut dysbiosis.

4.2. Results

Frequent antibiotic use and *E. coli* infections in rUTI cohort

Women with a history of rUTI were recruited to the UMB study, along with an age- and community-matched control cohort comprising healthy women (Methods). A total of 16 control and 15 rUTI women participated in the year-long study, providing monthly home-collected stool samples, as well as blood, urine and rectal swabs at enrollment and subsequent clinic visits for UTI treatment (Figure 4.1a). Participants completed monthly questionnaires on diet, symptoms, and behavior (Supplementary Data). There was a greater proportion of white women in the rUTI cohort, and self-reported antibiotic use was higher in this group in line with UTI treatment; otherwise, few dietary or behavioral differences were apparent (Extended Data Table 1).

A total of 24 UTIs occurred during the study, all in rUTI women, who each experienced 0-4 UTIs (Figure 4.1b). Nineteen were diagnosed by clinicians and five were inferred through self-reported symptoms and antibiotic use in the questionnaire during monthly sample collection. UTIs were typically treated with ciprofloxacin or nitrofurantoin. No significant temporal risk factors for UTI were identified amongst dietary or behavioral variables. Sexual intercourse is a well-known risk factor for UTI [2, 17], and all 19 clinically diagnosed UTIs occurred following at least one reported sexual

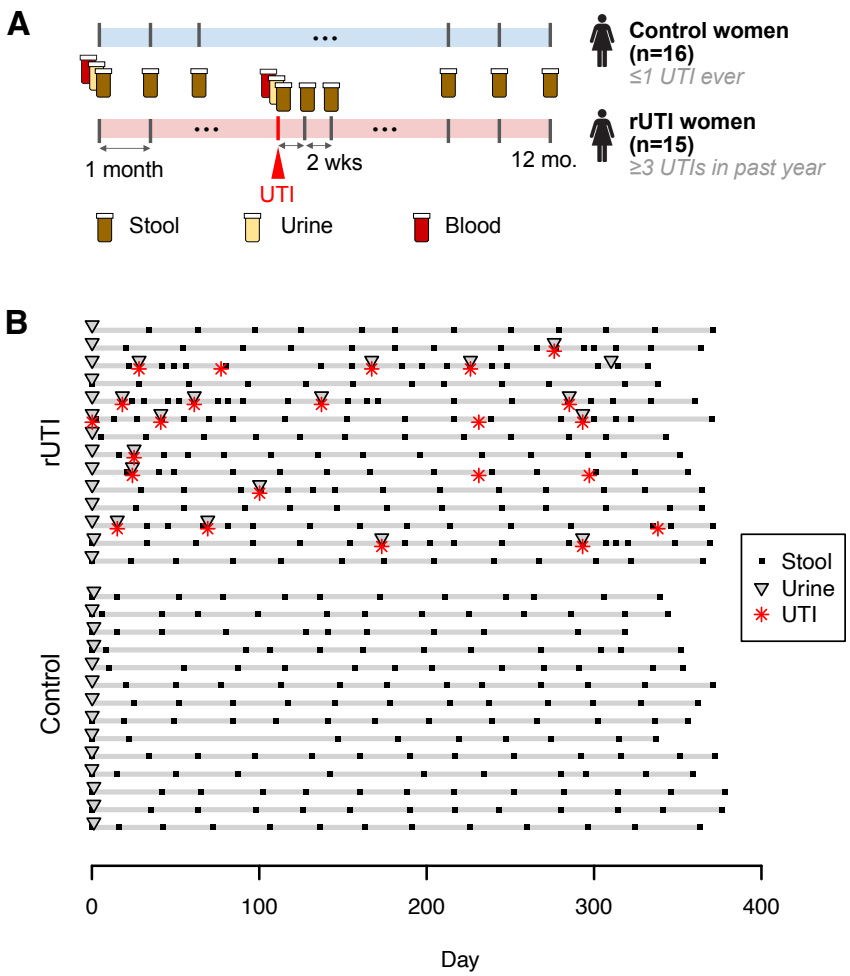

Figure 4.1: Study design and sample collection for the UMB study. **(a)** Stool samples were collected monthly from rUTI and control patients. Stool, urine and blood plasma samples were collected upon enrollment and subsequent UTI clinic visits. Biweekly stool samples were requested following UTI diagnoses. **(b)** Stool and urine samples collected from all rUTI and control participants (excluding one rUTI and two control participants who dropped out of the study prior to completion). Each participant's enrollment timeline is represented by horizontal gray lines, with stool (black dots) and urine (triangles) sample collection times denoted. Red symbols denote diagnosed and inferred UTI events.

encounter in the previous two weeks (Extended Data Figure C.1).

Urine samples collected at the time of clinical UTI diagnoses were plated on MacConkey agar; bacterial growth was detected (> 0 CFU/ml) from the majority (15/19; 79%, Supplementary Table 1). To determine the cause of infection, we sequenced 13 urine cultures, as well as uncultured urine, from all UTI diagnoses, defaulting to results from cultures when available. *E. coli* dominated 12/13 (92%) sequenced outgrowths; the remaining sample was dominated by *Klebsiella pneumoniae*. Sequencing uncultured urine from the remaining UTI samples identified uropathogens in a further four samples, including *E. coli* (two), *Enterococcus faecalis* and *Staphylococcus saprophyticus*, while two yielded no bacterial sequence (Supplementary Table 1). Based on sequencing, we defined 14 *E. coli* UTIs, comprising 82% of infections for which a bacterial cause could be inferred, broadly reflecting previous estimates of the proportion of all UTIs caused by *E. coli* [1].

rUTI gut depleted in microbial richness and butyrate-producers

It is increasingly recognized that the gut microbiota plays a role in a range of autoimmune and inflammatory diseases [18], as well as susceptibility to infection [16], and can alter inflammation in distal organs [19]. While previous studies have highlighted differential abundances of non-uropathogenic gut taxa as risk factors for bacteriuria in kidney transplant patients (reduced *Faecalibacterium* and *Romboutsia* [9]) and UTIs in children (reduced *Peptostreptococcaceae* [8]), it is unclear if these are risk factors for recurrence in otherwise healthy adult women. To explore this, we sequenced and analyzed the metagenomes of 367 longitudinal stool samples from both rUTI ($n = 197$) and control ($n = 170$) women (Figure 4.1b; Methods). Rectal swabs, collected during clinic visits, were not used to determine microbiome profiles.

There were broad differences in the gut microbiota composition between cohorts (Figure 4.2a-c). We fit linear mixed models with individual-level random effects to determine differences in diversity and composition between cohorts, adjusting for recent antibiotic use (Methods). Gut microbial richness was significantly lower, on average, in rUTI women ($p = 0.05$, Figure 4.2c). At the phylum level, we saw elevated levels of *Bacteroidetes* (false discovery rate [FDR] $= 0.003$) and a lower relative abundance of *Firmicutes* (FDR $= 0.02$) in rUTI women. We identified 22 differentially abundant taxa (FDR < 0.25) at lower taxonomic levels, 16 of which were depleted in rUTI women (Supplementary Table 2; Figure 4.2b), including *Faecalibacterium* as previously reported [9].

Several of the taxa reduced in the rUTI gut, including *Faecalibacterium*, *Akkermansia*, *Blautia* and *Eubacterium hallii*, are associated with short chain fatty acid (SCFA) production, including propionate and butyrate, which exert an anti-inflammatory effect in the gut through promotion of the intestinal barrier function and immunomodulation [20, 21]. *Blautia* was additionally identified as the only taxon significantly depleted at the time of UTI relative to non-UTI samples (FDR $= 0.01$). Cumulatively, SCFA producers, particularly butyrate producers, were significantly less abundant in rUTI women ($p = 0.001$) (Figure 4.2d; Extended Data Figure C.2). Four KEGG Orthogroups [22] representing components of butyrate production pathways were significantly re-

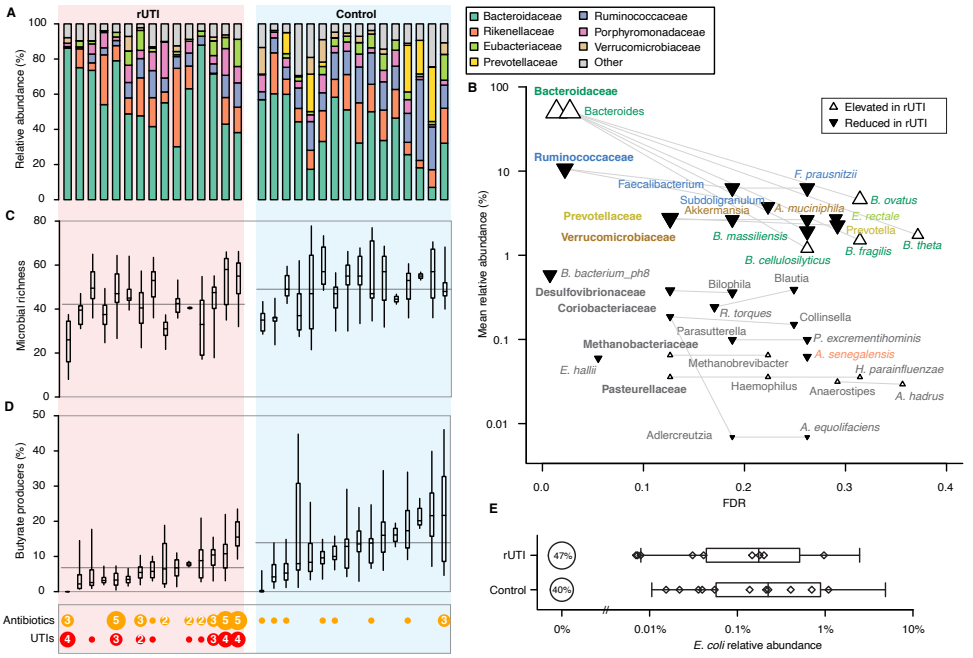

Figure 4.2: rUTI women have a distinct gut microbiome. **(a)** Average relative abundances of bacterial families for each patient in the rUTI (left) and the control (right) cohorts. **(b)** Significance and rUTI group effect size for selected taxa. Each point represents one taxon; its effect size and direction (symbol) for rUTI vs. control, false discovery rate (FDR) and mean relative abundance across all samples. Taxonomic relationships are represented by lines. FDR values calculated independently at each taxonomic level. Bold, regular and italic text denote family, genus and species levels. **(c)** Microbial richness distributions and **(d)** cumulative relative abundance of butyrate-producing species for each study participant. Plot displays the median (center line), 25th and 75th percentiles (box), as well as the 5th and 95th percentiles (whiskers) for each individual. Horizontal lines represent group-level mean of individual means. Antibiotic use and UTI occurrence for each study participant is shown at the bottom left; symbol size and numerals denote the number of UTIs/reported antibiotic courses. **(e)** Relative abundance of *E. coli* in each cohort. Symbols denote median relative abundances of individual patients; box plots display the median (center line), 25th and 75th percentiles (box), as well as the 5th and 95th percentiles (whiskers) of non-zero values.

duced across the rUTI cohort (Supplementary Table 3). Functional analysis with HUMAnN2 (ref. [23]) additionally revealed pathways depleted in the rUTI cohort, including those associated with sugar degradation and biosynthesis of metabolite intermediates and amino acids (Supplementary Table 4), many of which were also found to be differentially abundant in a study of irritable bowel syndrome (IBS) patients with sugar malabsorption [24].

This loss of gut microbial richness, diversity, and butyrate-producing bacteria is also a hallmark of exposure to broad spectrum antibiotics, including ciprofloxacin [25, 26, 27], which was used to treat more than a third of UTIs in our study. Thus, we sought to determine whether antibiotic effects may contribute to the observed shifts in microbiome composition in rUTI women ('rUTI dysbiosis'). Though antibiotic exposure in the previous two weeks was associated with a significant reduction in microbial richness ($p = 0.05$), this loss of richness was not sustained. Samples taken 2-6 weeks after antibiotic exposure were not significantly different from baseline levels ($p = 0.2$). Furthermore, we saw no association between the reported number of antibiotic courses and average richness (Figure 4.2c), and no differences in the overall gut microbiome stability between cohorts, despite more frequent antibiotic treatment among UTI women (Extended Data Figure C.3). We observed no differences in richness or in the abundance of butyrate producers between rUTI women with different antibiotic exposures (Extended Data Figure C.4a-b). Within the rUTI group, the frequency of infections was not associated with microbial richness or the relative abundance of butyrate producers. The microbial richness of women suffering UTIs during the study did not differ significantly from that of rUTI women not reporting infections ($p = 0.4$; Figure 4.2). While we did not detect a lasting impact from individual antibiotic courses – there were few long-term trends among rUTI women over the study (Extended Data Figure C.4c) – it is still possible that repeated antibiotic use over years may have contributed to the observed rUTI dysbiosis.

rUTI gut dysbiosis shares broad similarities with IBD

The depletion of butyrate-producing taxa and microbial richness, key characteristics of rUTI dysbiosis, are also observed in other gut inflammatory conditions, including nosocomial diarrhea [28], IBS [29], and inflammatory bowel disease (IBD) [20], particularly Crohn's disease [30], and thus may be indicative of gut inflammation in rUTI women. While IBD is a multifactorial disorder for which the causative role of gut microbes is incompletely understood [31], mouse models have helped demonstrate a causal relationship between gut dysbiosis and inflammation [32]. We compared our data to longitudinal gut microbiome data from adults with and without IBD in the Human Microbiome Project 2 (HMP2) study [33], which shared the same extraction and sequencing protocols (Methods). Relative to each study's control group, we found that the ten most significantly depleted species in the rUTI gut, including butyrate producers *F. prausnitzii* and *E. hallii*, were also depleted in the IBD gut. We further observed a significant overall correlation in the estimated change of species-level abundances associated with rUTI and IBD (Extended Data Figure C.5), suggesting more general similarities.

There were also some notable differences. *Bacteroides*, significantly elevated in the rUTI group, did not differ between cohorts in the HMP2 study (Extended Data Figure C.5), and were also decreased among IBD patients in other studies [34]. *E. coli* was significantly elevated in IBD patients in the HMP2 study, but showed no difference in average relative abundance between our cohorts (Figure 4.2e). Diminished *Bacteroides* alongside elevated *Enterobacteriaceae* was also observed in patients with nosocomial diarrhea [28]. Diarrhea, also a symptom of IBD, is associated with reduced gut transit time and is known to enrich for organisms common in the upper gastrointestinal tract, including *Enterobacteriaceae* [35], at the expense of anaerobic organisms such as *Bacteroides* [36]. As such, rUTI women with low-level inflammation and no diarrhea may lack the depletion of *Bacteroides* and elevation of *Enterobacteriaceae* observed in diarrhea-associated conditions. It is also possible that the considerable differences in treatment regimens; i.e. antibiotics vs. anti-inflammatories, contribute to divergences of a common underlying inflammatory signal.

Differential host immune response potentially linked to rUTI

rUTI dysbiosis also shares similarities with immunological syndromes affecting distal sites. For example, depletion of butyrate producers has been associated with rheumatoid arthritis, a systemic autoimmune disease which can be partially ameliorated in animal models with oral butyrate supplementation [37, 38]. Patients with chronic kidney disease also exhibit similar dysbiosis, including reduced *Parasutterella* and *Akkermansia*, the latter of which is inversely correlated with interleukin-10 levels, an anti-inflammatory cytokine [39]. We hypothesized that rUTI dysbiosis may also have an immunomodulatory role, potentially eliciting a differential immune response to bacterial invasion of the bladder. Thus, we explored immunological biomarkers from blood samples collected at enrollment and UTI, quantifying (i) a Luminex panel of human cytokines, chemokines, and growth factors involved in inflammation and T cell activation, and (ii) cell types and the transcriptional activity of peripheral blood mononuclear cells (PBMCs) (Methods).

Of the 39 Luminex analytes, one chemokine, plasma eotaxin-1, was higher in rUTI women vs. control women at enrollment, and is associated with intestinal inflammation [40]. Levels of eotaxin-1 are increased in colonic tissue of patients with active IBD [41]. Subsequent human eotaxin-1 ELISAs validated these results, highlighting an additional link to dysbiosis-driven perturbation of the immune state; though, since this result did not hold after adjusting for race, we could not rule out potential demographic confounders. Eotaxin-1 was also higher in blood plasma of rUTI women at the time of UTI vs. enrollment ($p = 0.04$; Extended Data Figure C.6b).

Our small cohort size provided limited statistical power to identify differential expression between cohorts based on PBMC RNA Seq data, and no large-scale differences were observed (Extended Data Figure C.6a). However, we found two genes that were upregulated in the PBMCs of the rUTI cohort (FDR < 0.1), *ZNF266* and the long non-coding RNA *LINC00944* (Supplementary Table 5). *ZNF266* has been previously linked to urological health, as a potential PBMC biomarker for overactive bladder and incontinence in women [42]. *LINC00944* has been associated with inflammatory

and immune-related signaling pathways, as well as tumor invading T lymphocytes in breast cancer, and markers for programmed cell-death [43]. Resting NK cells were significantly reduced at the time of UTI relative to baseline levels ($p = 0.02$; Extended Data Figure C.6c). NK cells help suppress bladder infection by UPEC in mice [44], so the loss of NK cells in the periphery may suggest a migration to the bladder at time of rUTI.

Gut and bladder *E. coli* dynamics similar between cohorts

Previous work has implicated gut dysbiosis and a depletion of butyrate-producing bacteria in enhanced susceptibility to gut colonization by pathogens, including *Salmonella* [45] and *C. difficile* [46]. While we could not quantify absolute species abundances, we observed no significant difference in the average relative abundance of *E. coli* between cohorts (Figure 4.2e), suggesting the rUTI dysbiotic gut is no more hospitable to *E. coli* colonization than controls. Further, we found no relationship between the relative abundances of *Escherichia* and butyrate producers in either cohort, suggesting that depletion of butyrate-producing bacteria does not enhance gut colonization by *Escherichia* (Extended Data Figure C.7). We considered the possibility that a temporal increase, or bloom, in *E. coli* relative abundance is a rUTI risk factor. Of the samples collected in the 14 days preceding an *E. coli* UTI, 75% exhibited *E. coli* relative abundance at or above average levels in the gut (Extended Data Figure C.8a-b). However, elevated *E. coli* levels were not predictive of UTIs; none of the 22 *E. coli* blooms (defined as *E. coli* relative abundance >10-fold higher than the intra-host mean) occurred in the two weeks prior to UTIs. Thänert et al. identified intestinal blooms of uropathogens preceding some UTIs, but similarly noted that blooms often occurred in the absence of infection [7], leading us to conclude that elevated levels of *E. coli* may facilitate transfer to the bladder but rarely manifest in infection. However, without frequent urine collection, we cannot rule out asymptomatic bladder colonization.

Though we did not detect differences in *E. coli* species dynamics, we hypothesized that rUTI dysbiosis may manifest in a qualitatively different *E. coli* population in the gut, contributing to increased rUTI susceptibility. We applied StrainGE [47] to explore *E. coli* strain-level diversity within stool metagenomes (Methods), and classified strains by phylogroup [48]. Patterns of strain carriage were similar in the rUTI (Figure 4.3) and control (Extended Data Figure C.9) cohorts. Both the number of strains per sample and the phylogroup distribution were comparable between cohorts (Figure 4.4, Extended Data Figure C.8c-d). While most *E. coli* strains (62%) were observed in one sample only, 22% were 'persistent', observed in at least one quarter of their carrier's samples. Persistent strains were more likely to originate from phylogroups B2 and D ($p = 0.01$), regardless of cohort, and were slightly more common in control women (OR = 2.1 (0.9, 5.2), $p = 0.1$), at odds with the hypothesis of differential colonization resistance to phylogroups associated with UPEC between cohorts.

We then applied StrainGE to all urine samples, seeking to elucidate differences in strain dynamics in the bladder. We found that 79% (11/14) of *E. coli* UTIs were caused by phylogroup B2 ($n = 7$) or D ($n = 4$) strains (Supplementary Table 1), approximately

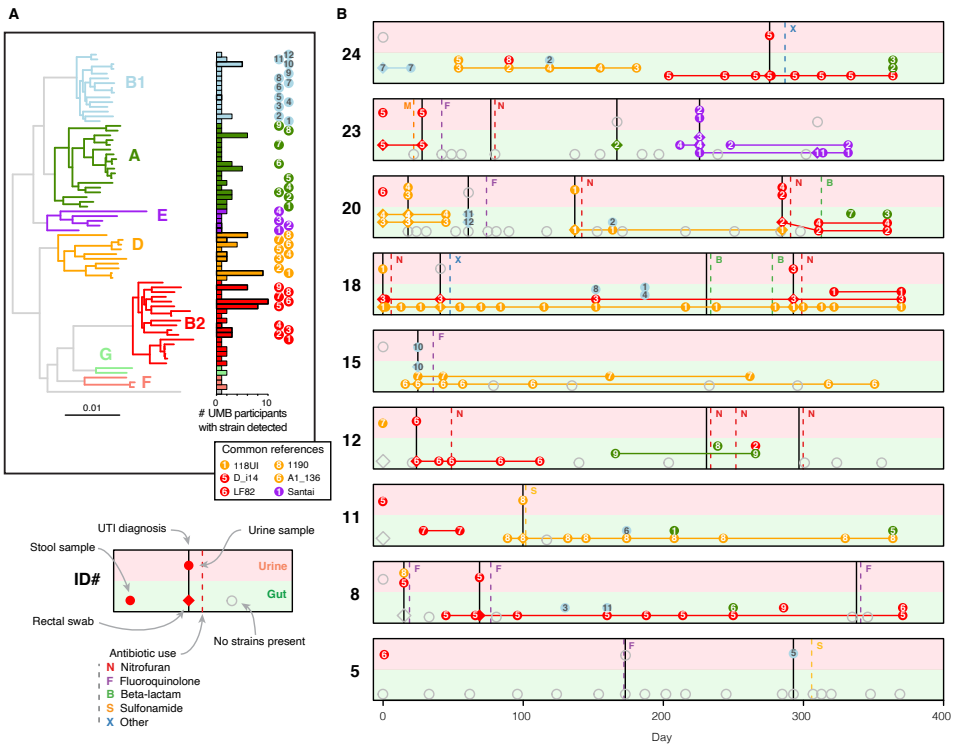

Figure 4.3: Frequent gut-bladder transmission and strain persistence in rUTI patients. Strain dynamics within all participants with *E. coli* UTIs. **(a)** Phylogenetic tree comprising strains called by StrainGE across all stool and urine samples, colored by phylogroup. Bars show number of unique participants with at least one strain observation; bars are bolded if the strain was identified in at least one urine sample. Each strain identified in rUTI patients is uniquely identifiable by the phylogroup (color) and ID (numeral) indicated right. **(b)** Each panel represents longitudinal strain dynamics within one patient. Numerals refer to strain identifiers in (a). All fecal strains are connected to their most recent previous observation in fecal samples. Diamonds denote clinical rectal swabs. Strains identified in urine outgrowth depicted if available; otherwise raw urine strains are shown. Fecal or urine samples with no detected *E. coli* strains represented by open grey symbols. Vertical dashed lines represent self-reported antibiotic use, solid black lines denote UTI events.

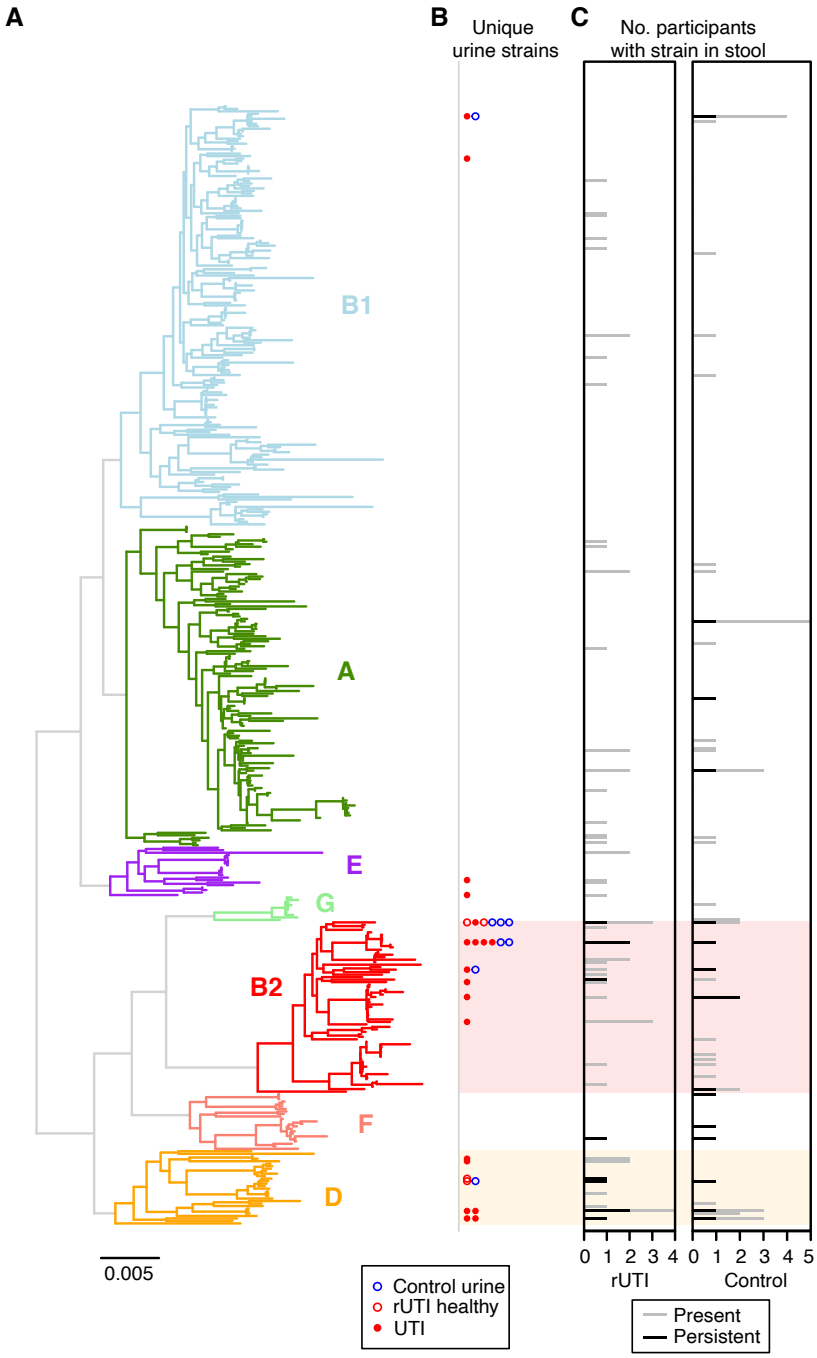

Figure 4.4: Phylogenetic distribution of *E. coli* strains identified in all stool and urine samples. **(a)** The phylogenetic tree of StrainGE reference strains colored and annotated by phylogroup. **(b)** Unique *E. coli* strains identified in urine samples are marked alongside the corresponding reference strain. Filled circles represent UTI-causing strains, blue circles denote strains identified in control hosts. **(c)** The total number of rUTI (left) and control (right) women with the corresponding strain present in stool samples. Black bars denote the number of women for whom the strain was persistent in the gut.

in line with previous studies [4, 49]. Of the 24 healthy enrollment urine samples yielding sufficient bacterial DNA to be sequenced and profiled (Supplementary Table 6), we detected *E. coli* strains in 54% (13/24), including over half of samples (7/13) from control participants, despite the absence of symptoms. All but one of these strains also belonged to phylogroups B2 and D. Control urines carried *E. coli* strains that were phylogenetically similar to UTI-causing strains based on StrainGE predictions (Figure 4.4; Methods), despite divergent clinical outcomes.

Mapping urine metagenome assemblies to a curated virulence factor database showed that UTI-causing strains were enriched in virulence factors (including iron uptake systems (*sit*, *chu*, *iro*, *ybt* operons), colibactin (*clb*), and type 6 secretion systems) relative to an *E. coli* species-wide database, though many of these were also present in the one urine sample from a control participant for which we had sufficient coverage to assess gene content (Methods, Supplementary Table 7). This transition of a likely urovirulent strain to the bladder of healthy women without eliciting UTI symptoms is consistent with previous studies which have been unable to identify genetic markers of urovirulence in mice [49], or consistently discriminate between UTI and asymptomatic bacteriuria strains in women [50]. Nevertheless, the divergence in clinical outcomes after bacterial bladder invasion may still arise due to phenotypic differences in *E. coli* strains reaching the bladder that are not readily apparent in genome comparisons. rUTI dysbiosis could have an impact on UPEC gene expression; it has been shown that higher SCFA levels are associated with down-regulation of *E. coli* virulence factors including fimbrial and flagellar genes [51]. However, such transcriptional analyses fall outside the scope of this study.

Antibiotic treatment fails to clear UTI-causing strains from gut

While it is well known that UTIs are most commonly caused by UPEC resident in the gut, their longitudinal dynamics of these strains within the gut are less well understood, despite the importance of such insights into developing rUTI prophylaxis. We applied StrainGE to all urine samples to identify UTI-causing strains and their gut dynamics, in particular at the time of UTI and after antibiotic exposure. Four rUTI women suffered multiple confirmed *E. coli* UTIs, though only one was a same strain recurrence (individual 8; Figure 4.3b). Comparisons of sequence data from urine samples and cultured rectal swabs from UTI clinic visits revealed that nearly all (11/12) *E. coli* UTIs, for which we had same-day rectal swabs, contained the same UTI strain, underscoring frequent gut to bladder transmission. The dominant *E. coli* strain in four of the rectal swab outgrowths was not the UTI-causing strain, suggesting some UTIs may be caused by minority strains. Only one UTI (individual 5, Figure 4.3) was caused by a strain never observed in another sample from that individual. This phylogroup B1 strain likely arose from a source other than the gut, such as the urinary tract or the vagina, also implicated as UPEC reservoirs [7, 52].

We anticipated that antibiotic exposure—particularly ciprofloxacin—would impact gut carriage of *E. coli* strains, and may explain the lower frequency of persistent colonizers in the rUTI group. Indeed, *E. coli* strains were detected by StrainGE significantly less frequently in stool samples from the two weeks following antibiotic

use (OR = 0.3 (0.13, 0.68); $p = 0.004$). However, many strains apparently cleared by antibiotics were observed again at later time points; in fact, none of the UTI-causing strains observed in the gut was permanently cleared following antibiotic exposure. It has previously been shown that coexistence of susceptible and resistant strains of the same lineage through acquisition/loss of mobile resistance elements can allow UPEC populations to rapidly adapt to repeated antibiotic exposure and persist in the gut [53]. While low-level persistence that is undetectable from sequencing data is a possibility, we plated a subset of post-treatment stool samples onto MacConkey agar to culture *E. coli*. In many cases, we observed no growth, suggesting absence rather than low-level persistence (Supplementary Table 8). Furthermore, profiling of 12 UTI-causing strains isolated from proximate stool samples demonstrated that the majority were susceptible to the antibiotics to which they were exposed (Supplementary Table 9). While a single stool sample is not completely representative of the gut microbiota, this suggests that UTI-causing strains may be frequently reintroduced to the gut from alternative sources following antibiotic clearance of the bladder and gut.

4.3. Discussion

Our study design, data collection and culture-independent metagenomic sequencing approach allowed us to characterize dynamics of the gut-bladder axis in healthy and rUTI women. We propose that rUTI susceptibility is dependent, in part, on perturbation of the gut-bladder axis, which represents a previously undescribed syndrome, comprising gut dysbiosis and differential host immunology. While this study was not designed to identify causal links between gut dysbiosis, immune response and rUTI susceptibility, the proposed model is consistent with our findings and provides a benchmark to be tested in future studies. Compared to healthy controls, women suffering rUTI exhibited gut dysbiosis characterized by depleted levels of butyrate-producing bacteria and diminished microbial richness. This dysbiosis did not appear to impact *E. coli* dynamics within the gut; relative abundances and strain types were similar between cohorts, suggesting that gut carriage of urovirulent bacteria in itself is not a risk factor for rUTIs. Notably, *E. coli* was commonly identified in the urine of healthy women, including strains arising from UPEC-associated clades and harboring similar virulence factors. Based on our observations, rUTI gut dysbiosis is consistent with low-level gut inflammation, and is reminiscent of other disorders in which microbiome-mediated immunomodulation plays a role in disease severity.

Our study had a number of limitations. Firstly, due to the limited collection of urine samples in control women, it was not possible to robustly compare (i) the composition of the urine microbiome, and (ii) the frequency of (asymptomatic) strain transfer from gut to bladder between cohorts. Secondly, we did not assess the role of other potential reservoirs, such as the vagina, which could explain UTIs caused by strains never observed in the gut. Thirdly, while StrainGE offers a high-resolution view of *E. coli* strain dynamics in the gut and bladder, we cannot rule out the presence of additional, low abundance strains which could not be detected from the depth of metagenomic data generated. Finally, the small cohort size and infrequent blood sample collection provided limited power to assess differential expression in PBMCs.

While we identified some indications of immunological differences between cohorts, our findings warrant further investigations to explore microbiome-host mucosal immune interactions in the context of rUTI susceptibility.

While identifying the origins of rUTI dysbiosis is outside the scope of this study, repeated antibiotic exposure is a plausible mechanism through which dysbiosis is maintained. The relatively short study period precluded us from establishing whether dysbiosis is the direct result of long-term antibiotic perturbation. In addition to the potentially detrimental impact of antibiotic use on the gut microbiota, we found that treatment also failed to clear UTI-causing strains from the gut in the long term. rUTI treatment protocols targeting UPEC strains in the gut with minimal disruption to other gut microbiota, such as small molecule therapeutics [54], may offer improved prospects. While more evidence is required to fully characterize the causal mechanisms between dysbiosis and infection, our work highlights the ineffectiveness and potential detrimental impact of current antibiotic therapies, as well as the potential for microbiome therapeutics (e.g. fecal microbiota transplants [10]) to limit infections via restoration of a healthy bacterial community in the gut.

4.4. Methods

Study design and sample collection

Enrollment

This study was conducted with the approval and under the supervision of the Institutional Review Board of Washington University School of Medicine in St. Louis, MO. Women from the St. Louis, MO area reporting three or more UTIs in the past 12 months were recruited into the rUTI study arm, while women with no history of UTI (at most one UTI ever) were recruited into the control arm via the Department of Urological Surgery at Barnes-Jewish Hospital in St. Louis, MO. We excluded women who: i) had inflammatory bowel disease (IBD) or urological developmental defects (e.g., ureteral reflux, kidney agenesis, etc.), ii) were pregnant, iii) take antibiotics as prophylaxis for rUTI, and iv) were younger than 18 years or older than 45 at the time of enrollment. All participants provided informed consent. Microbiological information for previous UTIs was not available. A total of 16 control and 15 rUTI women aged between 18 and 45 were recruited to the study; participants were remunerated with gift cards for participation. 14 women in each cohort completed the entire study collection protocol; no participants who completed the study were excluded from downstream analyses. Participants who did not complete the study were included in cohort-level comparisons, but excluded from longitudinal analyses. No statistical methods were used to pre-determine sample sizes but our sample sizes are similar to those reported in previous publications (e.g., refs [55, 56]). As an observational study with no intervention, with cohort membership based on predetermined criteria defined above, no subject randomization was required. Data collection and analysis were not performed blind to the conditions of the experiments.

Sample collection and storage

Participants provided blood and urine samples, as well as rectal swabs, at the initial clinic visit. UTIs were diagnosed during clinic visits; additional UTIs (not presenting at the study clinic) were inferred based on symptoms (painful urination, increased urgency/frequency of urination, cloudy urine) and antibiotic consumption reported in the monthly questionnaire. Women visiting the clinic during the study with UTI symptoms provided rectal swabs, blood and urine samples, and were requested to submit stool samples as soon as possible (within 24 hours) after the clinic visit, as well as at a two week follow-up time point.

All participants provided monthly stool samples for 12 months. Samples were collected at home, and submitted via mail following procedures developed in the Human Microbiome Project [33]. Briefly, participants collected a fresh fecal sample in a disposable toilet hat and then aliquoted two teaspoon-sized scoops of stool each into a tube containing phosphate buffered saline and a tube containing 100% ethanol. Samples were overnight to the Broad Institute where they were stored at -80C until sample processing. All stool samples were shipped Monday to Thursday within each week to limit samples long term exposure to ambient temperature; samples were stored in patients' home freezers until shipment, if necessary. Questionnaires were completed with all monthly and clinical sample collections; these captured self-reported antibiotic and drug use, dietary intake, sexual intercourse and UTI symptoms. Participants who did not provide stool samples and questionnaires at the beginning of each month were given phone call or email reminders to provide samples.

Sample processing

Blood sample preparation

A total of 15 mL of blood was collected from each patient during initial enrollment and UTI visits. The blood was stored on ice for less than 30 minutes and then mixed with an equal amount PBS with 2% fetal bovine serum (FBS). Peripheral blood mononuclear cell (PBMCs) were then isolated using SepMate PBMC isolation tubes (Stemcell Technologies) with Ficoll-Paque PLUS density gradient medium (Cytiva). Serum was collected during the PBMC isolation process and stored at -80C until use. PBMCs were washed with PBS plus 2% FBS and pelleted via centrifugation at 10,000 x g at room temperature for 5 minutes. PBMC cell pellets were then flash frozen and stored at -80C until RNA extraction.

Rectal swab and urine preparation

Rectal swabs were collected in the clinic and stored on ice for less than 30 minutes. Rectal swabs were washed in 2 mL of PBS. 1 mL of PBS was centrifuged at 10,000 x g at room temperature for 2 minutes and the PBS supernatant was removed. The bacterial/fecal pellet was then flash frozen and stored at -80C until DNA extraction. The remaining 1 mL was then used to make serial dilutions and then plated on both Luria Broth (LB) and MacConkey agar and incubated overnight at 37oC to quantify colony

forming units (CFUs). After bacterial enumeration, bacteria from MacConkey and LB plates were scraped to collect bacterial outgrowths. Bacterial cells were washed with PBS, pelleted at 10,000 x g at room temperature for 2 minutes, flash frozen and then stored at -80oC until DNA extraction.

Mid-stream urine samples were collected in sterile containers and stored on ice for less than 30 minutes. 10 mL of urine was centrifuged at 10,000 x g at room temperature for 5 minutes. The resulting pellet was washed in PBS, pelleted again, and then flash frozen and stored at -80oC until DNA extraction. 1 mL of urine was used to make serial dilutions and then plated onto both LB and MacConkey and incubated overnight at 37oC to enumerate CFUs. After outgrowth, the plates were scraped to collect bacterial colonies, which were then washed with PBS, pelleted at 10,000 x g at room temperature for 2 minutes, flash frozen and then stored at -80oC until DNA extraction.

RNA Extraction - PBMCs

RNA was extracted from stored PBMCs using TRIzol Reagent (cat. no. 15596-026 and 15596-018; Life Technologies), according to the manufacturer's protocol. Briefly, 0.75 mL of TRIzol was added per 0.25 mL of sample and cells were lysed by several rounds of pipetting. Samples were incubated for five minutes at room temperature. Chloroform was added to the samples at the recommended concentration and samples were incubated shaking for 15 seconds and set to rest for 2-3 minutes at room temperature. After incubation, samples were centrifuged at 12,000 x g for 15 minutes at 4oC. The aqueous phase was collected for RNA isolation. RNA was precipitated using 100% isopropanol and incubated at room temperature for 10 minutes, followed by centrifugation at 12,000 x g for 10 minutes at 4oC. The precipitated RNA was washed according to the protocol using 75% ethanol and resuspended in RNase-free water. Extracted RNA was stored at -80oC until further use.

DNA Extraction – Rectal Swabs and Urine

DNA was extracted from rectal swabs and urine samples plated on MacConkey agar using the Wizard Genomic DNA Purification Kit (Promega), according to the manufacturer's protocol. Briefly, samples were resuspended in 600 uL of Nuclei Lysis solution and incubated at 80oC for five minutes, then cooled to room temperature. RNase solution was added to samples and incubated for 15 minutes at 37oC, then cooled to room temperature. 200 uL of Protein Precipitation solution was added to the RNase-treated sample, vortexed for 20 seconds, and incubated on ice for 5 minutes. After incubation, samples were centrifuged for 3 minutes at 16,000 x g and the supernatant was transferred to a 1.5 mL microcentrifuge tube containing 600 uL of isopropanol. Samples were gently mixed and centrifuged for 2 minutes at 16,000 x g. The supernatant was removed and the DNA pellet was washed with 70% ethanol. Samples were centrifuged for 2 minutes at 16,000 x g, ethanol was aspirated and DNA pellets were air-dried for 15 minutes. The DNA pellet was rehydrated with DNA Rehydration solution and incubated at 65oC for 1 hour. Extracted DNA was stored at 4C for short-term storage and at 80oC for long-term storage until further use.

DNA Extraction – Stool

Total nucleic acid from stool was extracted following the HMP2 protocol [33], the basis of which is the Chemagic MSM I with the Chemagic DNA Blood Kit-96 from Perkin Elmer. DNA samples were quantified using a fluorescence-based PicoGreen assay.

WMS sequencing and sequence data processing

Libraries were constructed from extracted DNA from stool, urine, rectal swabs, and plate scrapes using the NexteraXT kit (Illumina). Then, libraries were sequenced on a HiSeq 2500 (Illumina) in 101 bp paired-end read mode and/or a HiSeq X10 (Illumina) in 151 bp paired-end read mode. Sequence data was then demultiplexed. Samples that were sequenced multiple times on different runs were pooled together. Reads were processed with KneadData (v0.7.2, `https://huttenhower.sph.harvard.edu/kneaddata/`) to remove adapter sequence and trim low base qualities (with Trimmomatic), as well as to remove human-derived sequences (by aligning to human genome with bowtie2).

Luminex assays

Custom Luminex magnetic bead assay kit was obtained from R&D systems (product LXSAHM). Analytes from Human Inflammation and Human T Cell Response panels were chosen for the custom kit of 39 analytes: CXCL1/GROalpha, IL-1alpha, M-CSF/CSF1, LIF, Ltalpha/TNF-b, MIF, APRIL, CCL11/Eotaxin, CCL4/MIP-1b, CXCL8/IL-8, IFN-g, IL-1b, IL-11, IL-13, IL-17A, IL-18, IL-21, IL-27, IL-31, IL-4, IL-6, MMP-1, TNF-a, BAFF/BlyS, CCL2/MCP-1, CX3CL1/Fractalkine, CXCL5/ENA-78, GM-CSF, IL-10, IL-12p70, IL-15, IL-17E/IL-25, IL-2, IL-22, IL-28A/INF-12, IL-33, IL-5, IL-7, MMP-3. Detection of the analytes in human plasma samples was performed using the Curiox DropArray system for miniaturization of magnetic bead multiplex kits. Plasma samples were diluted 2-fold for the assay. Results were read and quantified using a BioPlex multiplex plate reader and Microplate Manager software (v5).

Eotaxin ELISA

Plasma eotaxin (CCL11) levels from rUTI and control patients were measured using the Eotaxin (CCL11) Human Simple Step ELISA kit (cat. No. Ab185985; Abcam), according to the manufacturer's protocol. Briefly, plasma samples were diluted into sample diluent and 50 uL of sample and 50 uL of antibody cocktail were added to 96 well plate strips. Plates were sealed and incubated shaking for one hour at room temperature. Wells were washed three times with 1x wash buffer and inverted to remove excess liquid. 100 uL of TMB substrate was added to each well; plates were covered to protect from light and incubated shaking for 10 minutes. Stop solution (100 uL) was added to each well and plates were incubated shaking for one minute. The OD450 was measured and recorded to determine the concentration of Eotaxin in pg/mL.

Sequence data analysis

Community profiling and metrics

Bacterial community composition was determined using MetaPhlAn2 (v2.7.0 with db v20) [57] on KneadData-processed sequences. Functional profiling was performed by HUMAnN2 (v2.8.1, database downloaded in October 2016) [23] on KneadData-processed sequences. Diversity metrics and Bray-Curtis distances were derived from the MetaPhlAn2 relative abundance output using the vegan package in R [`https://cran.r-project.org/web/packages/vegan/`].

PBMC RNASeq analysis

Sequences from PBMC extracted mRNA were aligned to the human reference genome (hg19, Bioproject PRJNA31257) using the STAR aligner [58]. Picard-Tools (`https://broadinstitute.github.io/picard/`) was used to mark duplicate reads. Read counts per gene were generated with subread featureCounts [59]. Read counts were normalized into Counts Per Million (CPM) using edgeR [60]. This normalized read count matrix was then used as input for CIBERSORT using the LM22 signature gene set [61]. Results from CIBERSORT reported the relative abundance of 22 different immune cell types, including both PBMC and non-PBMC cell types, and it was used to remove three samples that were contaminated with 5% or greater of non-PBMC cell types. The CIBERSORT filtered set of samples was used to perform differential gene expression analysis using DESeq2 [62]. Baseline healthy control samples were compared to baseline rUTI samples. Due to limited sample numbers and potential confounding, we included only samples collected from caucasian women in this analysis. Results driven by single outlying data points were not considered.

E. coli strain profiling

In order to track *E. coli* strain dynamics we used Strain Genome Explorer (StrainGE), which we extensively benchmarked for use on low abundance species in the context of typical Illumina sequencer error [47]. We applied the StrainGST module of StrainGE to identify representative *E. coli* strains in all stool, urine and rectal swabs, using an *E. coli* reference database generated from RefSeq complete genomes, as detailed in van Dijk *et al.* [47]. Strains mapping to the same representative reference genome in this database typically have an ANI of at least 99.9%. To provide further evidence that same-strain calls from sample pairs from the same host were indeed matches, we ran the StrainGR module of StrainGE, which calculates alignment-based similarity metrics. We used benchmarked thresholds to determine strain matches; strain pairs with a common callable genome >0.5%, Jaccard gap similarity >0.95 and average callable nucleotide identity >99.95% were deemed matches.

Determination of UTI-causing strains

Urine samples provided at the time of UTI diagnosis were plated on MacConkey agar. Sequence data was generated from DNA extracted from uncultured urine, and/or outgrowth on selective media. The cause of UTI was deemed to be the most abundant uropathogen, using outgrowth data where available, uncultured urine otherwise. Species were determined to be uropathogens based on UTI prevalence studies, for example ref. [1].

Determination of virulence factors

Urine metagenomes for which *E. coli* represented the dominant species were assembled using SPAdes [63]. To detect virulence factors in *E. coli* references (see StrainGST section above) and assembled genomes from study samples, we used the Virulence Factor Database (VFDB) for *E. coli* and the type 6 secretion system (T6SS) database (SecReT6) in genome-wide BLAST+ searches. Though VFDB contains T6SS genes, we removed them in favor of the T6SS-specific database for a T6SS-specific analytical pipeline. Other VFDB hits from blastn were filtered for $\geq$ 90% identity and $\geq$ 90% coverage. All *E. coli* genomes were separated by phylogroup for enrichment analysis, where Fisher's Exact test was used to determine the significance of virulence factor enrichment in a certain phylogroup. T6SS hits were filtered for $\geq$ 90% identity and $\geq$ 90% coverage and the system was considered present where at least 12 different adjacent T6SS genes were present. Again, an enrichment analysis was performed using Fisher's Exact test to determine the significance of T6SS presence in certain phylogroups.

Statistical testing and models

rUTI risk factors

We used questionnaire responses to determine if any dietary or behavioral factors were associated with rUTI. We first compared the proportion of participants in each cohort who responded positively to binary variables (e.g. dairy consumption, alcohol etc. in the previous two weeks) in more than 50% or responses, and used a Fisher's Exact test to determine significance. We next fit mixed effects logistic regression models to determine temporal risk factors for UTIs. Samples collected within 3 days of UTI diagnosis were classified as 'time of UTI'; this binary variable was fit as a function of host (random effects term) and each dietary or behavioral response variable collected in the questionnaire. Variables with limited or no variance were excluded.

Identifying differences at the cohort level and time of UTI

We fit mixed effects linear regression models to compare the structure, diversity and function of the gut microbiome between cohorts, following similar approaches employed by previous studies (for example, ref. [64]). For this purpose, we used sequence data from all collected stool samples, but did not include rectal samples. An arcsine

square root transformation was applied to relative abundance values. Features (transformed relative abundances, diversity, microbial richness) were fit as a function of host (random effects term), cohort (categorical variable), and terms for antibiotic use and race (categorical variable) to adjust for potential confounding effects. To assess change in relative abundances at relevant timepoints, we also fit models including covariates for 'pre-UTI' (14 days preceding UTI diagnosis), 'time of UTI' (three days either side of UTI diagnosis), or post antibiotics (<14 days post antibiotic exposure) as binary variables. All taxa with more than 10% non-zero values were fitted using the lme4 function in R. Significance of associations was determined using Wald's test, and p-values were adjusted for multiple hypothesis testing using Benjamini-Hochberg correction at each taxonomic level.

The relative abundance of SCFA producers was additionally compared between cohorts; butyrate- and propionate-producing species were determined based on functional capacity to produce butyrate and propionate [65]. These species' relative abundances were then aggregated and compared as above.

We compared the stability of the microbiome between cohorts by assessing the distributions of within-host pairwise Bray-Curtis (BC) dissimilarities between individuals. Since rUTI women had, on average, slightly more frequent sampling than control women, due to the additional follow-up samples after UTI diagnoses, this metric may be biased towards smaller values in this cohort. However, we observed no significant trend between BC dissimilarity and time between samples, suggesting no detectable long-term trends. Furthermore, we detected no difference in the distribution of time-adjusted BC distances (BC divided by number of days between samples) between cohorts.

IBD comparisons

To compare rUTI dysbiosis to an IBD gut state, we downloaded MetaPhlAn2 output from the HMP2 study [33], (`https://ibdmdb.org`). We extracted longitudinal samples from adults with IBD (diagnosis='UC' or 'CD') and non-IBD controls (diagnosis='nonIBD'). We fit linear mixed effects models with standardized relative abundances as a function of host (random effects term), race (race='white'; binary term) and recent antibiotic use. Fitted coefficients for the IBD and rUTI cohorts are then plotted in Extended Data Figure C.5.

Data availability

Metagenomic sequence data are available from the Sequence Read Archive under Bioproject PRJNA400628. PBMC RNASeq data are available from the database of Genotypes and Phenotypes (dbGaP) under project number phs002728. Questionnaire data, output files from MetaPhlan2, Humann2, StrainGE are available from `https://github.com/cworby/UMB-study`.

Code availability

Custom R scripts to analyze outputs are available from `https://github.com/cworby/UMB-study`.

Acknowledgements

This project has been funded in part with Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Grant Number U19AI110818 to the Broad Institute, and from the National Institutes of Health Mucosal Immunology Studies Team consortium Grant Number U01AI095542 to Washington University. BSO was supported by grants from the National Institutes of Health, USA (T32GM007067 and T32GM139774). This work was also supported by funds from the Center for Women's Infectious Disease Research (cWIDR) at Washington University School of Medicine.

We would like to acknowledge members of the Broad's Bacterial Genomics group and Hera Vlamakis for helpful conversations. We thank Brian Haas for assistance with PBMC RNA-Seq analysis as well as the Multi-Omics Core and Genomics Platform at the Broad Institute for sample processing and data generation.

References

1. Flores-Mireles AL, Walker JN, Caparon M, and Hultgren SJ. Urinary tract infections: Epidemiology, mechanisms of infection and treatment options. Nat. Rev. Microbiol. 2015; 13. Publisher: Nature Publishing Group:269–84
2. Hooton Thomas M. et al. A Prospective Study of Risk Factors for Symptomatic Urinary Tract Infection in Young Women. New England Journal of Medicine 1996; 335. Publisher: Massachusetts Medical Society _eprint: https://www.nejm.org/doi/pdf/10.1056/NEJM199608153350703:468–74
3. Yamamoto S et al. Genetic Evidence Supporting the Fecal-Perineal-Urethral Hypothesis in Cystitis Caused by Escherichia Coli. The Journal of Urology 1997 Mar; 157:1127–9
4. Nielsen KL, Dynesen P, Larsen P, and Frimodt-Møller N. Faecal Escherichia coli from patients with E. coli urinary tract infection and healthy controls who have never had a urinary tract infection. Journal of Medical Microbiology 2014; 63. Publisher: Microbiology Society,:582–9
5. JANTUNEN ME, SAXÉN H, LUKINMAA S, ALA-HOUHALA M, and SIITONEN A. Genomic identity of pyelonephritogenic Escherichia coli isolated from blood, urine and faeces of children with urosepsis. Journal of Medical Microbiology 2001; 50. Publisher: Microbiology Society,:650–2
6. Magruder M et al. Gut uropathogen abundance is a risk factor for development of bacteriuria and urinary tract infection. en. Nature Communications 2019 Dec; 10. Publisher: Nature Publishing Group:5521
7. Thänert R et al. Comparative Genomics of Antibiotic-Resistant Uropathogens Implicates Three Routes for Recurrence of Urinary Tract Infections. mBio 2019 Aug; 10. Publisher: American Society for Microbiology:10.1128/mbio.01977–19
8. Paalanne N et al. Intestinal microbiome as a risk factor for urinary tract infections in children. en. European Journal of Clinical Microbiology & Infectious Diseases 2018 Oct; 37:1881–91
9. Magruder M et al. Gut commensal microbiota and decreased risk for Enterobacteriaceae bacteriuria and urinary tract infection. Gut Microbes 2020 Nov; 12. Publisher: Taylor & Francis:1805281
10. Tariq R et al. Fecal Microbiota Transplantation for Recurrent Clostridium difficile Infection Reduces Recurrent Urinary Tract Infection Frequency. Clinical Infectious Diseases 2017 Oct; 65:1745–7
11. Wang T, Kraft CS, Woodworth MH, Dhere T, and Eaton ME. Fecal Microbiota Transplant for Refractory Clostridium difficile Infection Interrupts 25-Year History of Recurrent Urinary Tract Infections. Open Forum Infectious Diseases 2018 Feb; 5:ofy016
12. Mayer EA, Tillisch K, and Gupta A. Gut/brain axis and the microbiota. en. The Journal of Clinical Investigation 2015 Mar; 125. Publisher: American Society for Clinical Investigation:926–38

13. Cryan JF et al. The Microbiota-Gut-Brain Axis. Physiological Reviews 2019 Oct; 99. Publisher: American Physiological Society:1877–2013

14. Budden KF et al. Emerging pathogenic links between microbiota and the gut–lung axis. en. Nature Reviews Microbiology 2017 Jan; 15. Publisher: Nature Publishing Group:55–63

15. Dang AT and Marsland BJ. Microbes, metabolites, and the gut–lung axis. en. Mucosal Immunology 2019 Jul; 12. Publisher: Nature Publishing Group:843–50

16. Lazar V et al. Aspects of Gut Microbiota and Immune System Interactions in Infectious Diseases, Immunopathology, and Cancer. English. Frontiers in Immunology 2018 Aug; 9. Publisher: Frontiers

17. Scholes D et al. Risk Factors for Recurrent Urinary Tract Infection in Young Women. The Journal of Infectious Diseases 2000 Oct; 182:1177–82

18. Clemente JC, Manasson J, and Scher JU. The role of the gut microbiome in systemic inflammatory disease. en. BMJ 2018 Jan; 360. Publisher: British Medical Journal Publishing Group Section: Clinical Review:j5145

19. Belkaid Y and Hand TW. Role of the Microbiota in Immunity and Inflammation. English. Cell 2014 Mar; 157. Publisher: Elsevier:121–41

20. Parada Venegas D et al. Short Chain Fatty Acids (SCFAs)-Mediated Gut Epithelial and Immune Regulation and Its Relevance for Inflammatory Bowel Diseases. English. Frontiers in Immunology 2019 Mar; 10. Publisher: Frontiers

21. Liu H et al. Butyrate: A Double-Edged Sword for Health? Advances in Nutrition 2018 Jan; 9:21–9

22. Kanehisa M, Sato Y, Kawashima M, Furumichi M, and Tanabe M. KEGG as a reference resource for gene and protein annotation. Nucleic Acids Research 2016 Jan; 44:D457–D462

23. Franzosa EA et al. Species-level functional profiling of metagenomes and metatranscriptomes. en. Nature Methods 2018 Nov; 15. Number: 11 Publisher: Nature Publishing Group:962–8

24. Mack A et al. Changes in gut microbial metagenomic pathways associated with clinical outcomes after the elimination of malabsorbed sugars in an IBS cohort. Gut Microbes 2020 May; 11. Publisher: Taylor & Francis:620–31

25. Palleja A et al. Recovery of gut microbiota of healthy adults following antibiotic exposure. en. Nature Microbiology 2018 Nov; 3. Publisher: Nature Publishing Group:1255–65

26. Zaura E et al. Same Exposure but Two Radically Different Responses to Antibiotics: Resilience of the Salivary Microbiome versus Long-Term Microbial Shifts in Feces. mBio 2015 Nov; 6. Publisher: American Society for Microbiology:10.1128/mbio.01693–15

27. Rooney AM et al. Each Additional Day of Antibiotics Is Associated With Lower Gut Anaerobes in Neonatal Intensive Care Unit Patients. Clinical Infectious Diseases 2020 Jun; 70:2553–60

28. Schubert AM et al. Microbiome Data Distinguish Patients with Clostridium difficile Infection and Non-C. difficile-Associated Diarrhea from Healthy Controls. mBio 2014 May; 5. Publisher: American Society for Microbiology:10.1128/mbio.01021–14

29. Pozuelo M et al. Reduction of butyrate- and methane-producing microorganisms in patients with Irritable Bowel Syndrome. en. Scientific Reports 2015 Aug; 5. Publisher: Nature Publishing Group:12693

30. Geirnaert A et al. Butyrate-producing bacteria supplemented in vitro to Crohn's disease patient microbiota increased butyrate production and enhanced intestinal epithelial barrier integrity. en. Scientific Reports 2017 Sep; 7. Publisher: Nature Publishing Group:11450

31. Ni J, Wu GD, Albenberg L, and Tomov VT. Gut microbiota and IBD: causation or correlation? en. Nature Reviews Gastroenterology & Hepatology 2017 Oct; 14. Publisher: Nature Publishing Group:573–84

32. Schaubeck M et al. Dysbiotic gut microbiota causes transmissible Crohn's disease-like ileitis independent of failure in antimicrobial defence. en. Gut 2016 Feb; 65. Publisher: BMJ Publishing Group Section: Gut microbiota:225–37

33. The Integrative HMP (iHMP) Research Network Consortium. The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles during Periods of Human Health and Disease. English. Cell Host & Microbe 2014 Sep; 16. Publisher: Elsevier:276–89

34. Zhou Y and Zhi F. Lower Level of *Bacteroides* in the Gut Microbiota Is Associated with Inflammatory Bowel Disease: A Meta-Analysis. en. BioMed Research International 2016 Nov; 2016. Publisher: Hindawi:e5828959

35. Duvallet C, Gibbons SM, Gurry T, Irizarry RA, and Alm EJ. Meta-analysis of gut microbiome studies identifies disease-specific and shared responses. en. Nature Communications 2017 Dec; 8. Publisher: Nature Publishing Group:1784

36. Asnicar F et al. Blue poo: impact of gut transit time on the gut microbiome using a novel marker. en. Gut 2021 Sep; 70. Publisher: BMJ Publishing Group Section: Gut microbiota:1665–74

37. Takahashi D et al. Microbiota-derived butyrate limits the autoimmune response by promoting the differentiation of follicular regulatory T cells. EBioMedicine 2020 Aug; 58:102913

38. Rosser EC et al. Microbiota-Derived Metabolites Suppress Arthritis by Amplifying Aryl-Hydrocarbon Receptor Activation in Regulatory B Cells. Cell Metabolism 2020 Apr; 31:837–851.e10

39. Li F, Wang M, Wang J, Li R, and Zhang Y. Alterations to the Gut Microbiota and Their Correlation With Inflammatory Factors in Chronic Kidney Disease. English. Frontiers in Cellular and Infection Microbiology 2019 Jun; 9. Publisher: Frontiers

40. Adar T, Shteingart S, Ben Ya'acov A, Bar-Gil Shitrit A, and Goldin E. From airway inflammation to inflammatory bowel disease: Eotaxin-1, a key regulator of intestinal inflammation. Clinical Immunology 2014 Jul; 153:199–208

41. Adar T et al. The Importance of Intestinal Eotaxin-1 in Inflammatory Bowel Disease: New Insights and Possible Therapeutic Implications. en. Digestive Diseases and Sciences 2016 Jul; 61:1915–24

42. Cheung W et al. Peripheral Blood Mononuclear Cell Gene Array Profiles in Patients With Overactive Bladder. English. Urology 2010 Apr; 75. Publisher: Elsevier:896–901

43. Santiago PR de et al. Immune-related IncRNA LINC00944 responds to variations in ADAR1 levels and it is associated with breast cancer prognosis. Life Sciences 2021 Mar; 268:118956

44. Gur C et al. Natural Killer Cell-Mediated Host Defense against Uropathogenic E. coli Is Counteracted by Bacterial HemolysinA-Dependent Killing of NK Cells. English. Cell Host & Microbe 2013 Dec; 14. Publisher: Elsevier:664–74

45. Rivera-Chávez F et al. Depletion of Butyrate-Producing Clostridia from the Gut Microbiota Drives an Aerobic Luminal Expansion of Salmonella. English. Cell Host & Microbe 2016 Apr; 19. Publisher: Elsevier:443–54

46. Antharam VC et al. Intestinal Dysbiosis and Depletion of Butyrogenic Bacteria in Clostridium difficile Infection and Nosocomial Diarrhea. Journal of Clinical Microbiology 2020 Dec; 51. Publisher: American Society for Microbiology:2884–92

47. Dijk LR van et al. StrainGE: a toolkit to track and characterize low-abundance strains in complex microbial communities. Genome Biology 2022 Mar; 23:74

48. Clermont O, Bonacorsi S, and Bingen E. Rapid and Simple Determination of theEscherichia coli Phylogenetic Group. Applied and Environmental Microbiology 2000 Oct; 66. Publisher: American Society for Microbiology:4555–8

49. Schreiber HL et al. Bacterial virulence phenotypes of Escherichia coli and host susceptibility determine risk for urinary tract infections. Sci. Transl. Med. 2017 Mar; 9

50. Garretto A et al. Genomic Survey of E. coli From the Bladders of Women With and Without Lower Urinary Tract Symptoms. English. Frontiers in Microbiology 2020 Sep; 11. Publisher: Frontiers

51. Zhang S et al. Short Chain Fatty Acids Modulate the Growth and Virulence of Pathosymbiont Escherichia coli and Host Response. en. Antibiotics 2020 Aug; 9. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute:462

52. Stapleton AE. The Vaginal Microbiota and Urinary Tract Infection. Microbiology Spectrum 2016 Dec; 4. Publisher: American Society for Microbiology:10.1128/microbiolspec.uti–0025–2016

53. Forde BM et al. Population dynamics of an Escherichia coli ST131 lineage during recurrent urinary tract infection. en. Nature Communications 2019 Aug; 10. Publisher: Nature Publishing Group:3643

54. Cusumano CK et al. Treatment and Prevention of Urinary Tract Infection with Orally Active FimH Inhibitors. Science Translational Medicine 2011 Nov; 3. Publisher: American Association for the Advancement of Science:109ra115–109ra115

55. Dethlefsen L and Relman DA. Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. Proceedings of the National Academy of Sciences 2011 Mar; 108. Publisher: Proceedings of the National Academy of Sciences:4554–61

56. Turnbaugh PJ et al. An obesity-associated gut microbiome with increased capacity for energy harvest. en. Nature 2006 Dec; 444. Publisher: Nature Publishing Group:1027–31

57. Truong DT et al. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. en. Nature Methods 2015 Oct; 12. Number: 10 Publisher: Nature Publishing Group:902–3

58. Dobin A et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013 Jan; 29:15–21

59. Liao Y, Smyth GK, and Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 2014 Apr; 30:923–30

60. Robinson MD, McCarthy DJ, and Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 2010 Jan; 26:139–40

61. Newman AM et al. Robust enumeration of cell subsets from tissue expression profiles. en. Nature Methods 2015 May; 12. Publisher: Nature Publishing Group:453–7

62. Love MI, Huber W, and Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. en. Genome Biology 2014 Dec; 15:550

63. Bankevich A et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. Journal of Computational Biology 2012 May; 19. Publisher: Mary Ann Liebert, Inc., publishers:455–77

64. Lloyd-Price J et al. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. Nature 2019 May; 569:655–62

65. Louis P and Flint HJ. Formation of propionate and butyrate by the human colonic microbiota. en. Environmental Microbiology 2017; 19. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1462-2920.13589:29–41

5

Discussion

In this thesis, we have presented several tools and algorithms to improve the identification and comparison of genetic variation in bacterial genomes, enabling new insights into their biology and role in human health and disease. In Chapter 2, we introduced POASTA, a new, faster, partial order alignment algorithm with much-reduced memory usage. This enabled accurate alignments of longer genomic sequences than previously possible, useful for pangenome graph construction. In Chapter 3, we introduced StrainGE, a new tool for detecting and characterizing same-species strains in metagenomic samples. Many clinically relevant species are lowly abundant in the human gut, and StrainGE was the only tool able to deconvolve strain mixtures at coverages as low as 0.5x. In Chapter 4, we characterized the interplay between the gut microbiota and susceptibility for recurrent UTIs, where StrainGE enabled novel insights into the *E. coli* strain-level dynamics in the gut and bladder of women with rUTI and healthy controls. Despite these contributions, characterizing and comparing bacterial (pan)genomes with extensive genomic diversity remains challenging. In this chapter, we will review the remaining limitations of current methods, discuss future avenues for method development, and present impactful future applications.

Moving beyond gene-centered pangenomes

Bacterial pangenome analyses have historically focused on genes [1, 2]. Such approaches have enabled defining core and accessory genes, and since bacterial genomes have high coding densities, enabled characterizing genetic variation between diverse bacterial strains across nearly the entire genome [3]. However, these approaches omit the genome's non-coding regions, while variation in those regions could substantially impact phenotypes [4].

Instead of defining the pangenome based on genes, Marschall et al. generalize the definition of the pangenome as "any collection of genomic sequences to be analyzed jointly or used as reference" [5]. The design of generalized pangenome analysis tools is an active area of research. The ultimate goal of such tools should be to represent multiple genomes or haplotypes compactly and enable quick identification of homologous regions and (shared) genetic variation.

Two recent examples of such general pangenome graph construction tools include Minigraph-Cactus (MC) [6] and the Pangenome Graph Builder (PGGB) [7], both of which were used to construct the draft human pangenome [8]. These tools do not require gene annotations and are not limited to each genome's coding regions since they are based on whole genome sequence alignments. The graphical representation of input genomes encodes homology and represents both small-scale genetic variation (SNPs, indels) and structural rearrangements. The graph can additionally serve as a basis for read alignment, enabling the genotyping of other samples with reduced reference bias [9].

However, neither MC nor PGGB have been extensively tested on bacterial genomes. Since different processes drive the evolution of bacterial and eukaryotic genomes, it is still an open question whether the graphs constructed by either tool accurately encode homology relationships among bacterial genomes and whether they can accurately represent genetic variation in the presence of high recombination rates and

horizontal gene transfer.

In any case, partial order alignment (POA) will likely be a valuable component for such a pangenome graph construction pipeline. It is already an essential component of both the MC and PGGB pipelines, highlighting the broad utility of this algorithm, also outside bacterial genomics. We showed in Chapter 2 that POASTA significantly reduced the computational cost to compute partial order alignments, paving the way for genome-scale alignments and accelerating the development of novel pangenome analysis tools.

The capability to describe genetic variation pangenome-wide will improve many analyses. In the remainder of this chapter, we will further detail three applications where pangenome graphs could have a significant impact. These applications include 1) pangenome-wide association studies, 2) strain-level analysis of microbial communities, and 3) genomic epidemiology.

Elucidating bacteria's many genes with unknown function

A significant barrier to a mechanistic understanding of phenotypes is the large fraction of bacterial genes with unknown functions. For example, of the approximately ten million gene families in the human gut microbiome [10], more than 50% have an unknown function [11].

Characterizing gene function is inherently challenging and hard to scale, and we expect it will remain so for the foreseeable future. However, rapid advances in two complementary genome-wide functional screening methods will likely accelerate the elucidation of gene functions in the coming decade [12].

First, transposon-insertion sequencing (Tn-Seq) has enabled the high-throughput screening of essential genes in controlled experimental conditions [13, 14]. Since its introduction, Tn-Seq has revealed genes conferring resistance to antibiotics in *Staphylococcus aureus* [15], genes in *Streptococcus pyogenes* essential for its survival in human saliva [16], genes essential for *E. coli* capsule production [17], and genes involved in *E. coli*'s capacity to colonize the gastrointestinal tract [18], among many other findings (recently reviewed in ref. [14]). Further scaling to more experimental conditions and testing more strains is an active area of research [19, 14], and we refer to Cain et al. [14] for a more in-depth discussion of future Tn-Seq directions.

Second, inspired by successes in human genome-wide association studies (GWAS) [20], bacterial GWAS have gained traction in the past decade [21, 22]. Such studies have implicated genes in host-adaption [23], identified genomic markers conferring antimicrobial resistance in multiple species [24], and identified the genetic basis underlying invasiveness in *Streptococcus pneumoniae* [25].

A benefit of GWAS is the analysis of strains directly sampled from their environments, considering the genetic variation naturally present in the population [12]. This contrasts with Tn-Seq, where experiments are performed in highly controlled conditions that may not mimic the bacteria's natural environments. The downside of GWAS is that it only provides indirect evidence for associating a variant to a phenotype [12].

Additional follow-up studies (e.g., using engineered knock-out strains) might be required to confirm a causal relationship.

Bacterial GWAS faces additional challenges compared to human GWAS. While meiosis in humans ensures variants are observed on many genetic backgrounds, bacteria reproduce clonally, complicating distinguishing between causal variants and linkage effects [22]. Correcting for strong population structure is thus essential to prevent the identification of variants that correlate with a phenotype because they were co-inherited with another causal variant. Recent methods have addressed this using phylogenetic trees [26], linear mixed models [24, 27], or elastic nets [28].

A second challenge is the high genomic diversity in many bacterial species, and it is important to consider the kind of genomic marker to be associated with a phenotype. In human GWAS, SNPs with respect to a single reference typically fulfill that role. However, limiting bacterial GWAS to a single reference would result in reference biases and omit much variation in the strains' accessory genomes [29]. Instead, several approaches opted to associate *k*-mers to phenotypes [23, 27]. *k*-mers are a flexible approach that can describe variation even in each strain's accessory genome. However, they are also redundant, increasing the number of statistical tests and reducing statistical power, and can be hard to interpret [30]. While this can be partially addressed by constructing a colored, compacted De Bruijn Graph and associating the resulting unitigs to phenotypes [30], another downside is that a causal variant could be obscured by other (unrelated) variants nearby (within the length of a k-mer), reducing statistical power.

Another alternative strategy employed by Panaroo is associating structural rearrangements in its gene-centered pangenome graph with phenotypes [31]. While this enables associating (sets of) genes with particular phenotypes, it cannot associate genetic variation at a lower resolution (such as SNPs) to phenotypes.

Novel pangenome analysis tools, as described in the section "Moving beyond gene-centered pangenomes", could bridge the current gap in our ability to describe bacterial genetic variation. Such tools should be able to represent many kinds of structural rearrangements as well as precisely describe small-scale variation, such as SNPs and indels.

An important feature to consider in the context of GWAS is how to genotype large sets of samples pangenome-wide. GWAS often requires large sample numbers to identify statistically significant hits [22], and we expect that Illumina's short-read sequencing platform will be the platform of choice for the foreseeable future since it is the most cost-effective. In our view, a genotyping approach based on read alignment to a pangenome graph would likely be more powerful than current bacterial pangenome analysis tools, most which rely on *de novo* assembly from short reads which is frequently inaccurate or incomplete [32]). To that end, the ideas we presented in Chapter 2, while currently implemented as a POA tool, could also aid a future read-to-grah aligner, substantially reducing the computational cost of genotyping large sets of samples.

Finally, associating both bacterial and host genetic variation with observed phenotypes will further detail how bacteria influence human health [22]. For example, Lees et al. found that the genetic factors of *S. pneumoniae* largely explained invasiveness,

while host genetic factors explained disease severity [25]. Taken together, we expect such improvements to bacterial GWAS will reveal new insights into the genetic basis of phenotypes such as health and disease, elucidate molecular mechanisms, and inform on novel therapeutics.

The therapeutic potential of the microbiome and strain-resolved metagenomics

The therapeutic potential of the gut microbiome was first demonstrated by treating *Clostridium difficile* infections using fecal microbiota transplantation (FMT) from a healthy donor [33]. FMTs also reduce the frequency of UTI recurrence [34, 35], highlighting a potential avenue for treatment and motivating our investigation into the role of the microbiome in UTI recurrence (Chapter 4).

While FMTs have successfully treated numerous conditions in the past decades, transferring complex and heterogeneous microbial communities bears considerable risk [36]. FMT might transfer antimicrobial-resistant bacteria and increase the risk of additional infections or sepsis, which could result in death [36, 37].

Instead, several companies are developing defined microbiome-based therapeutics to treat *C. difficile* infections [38]. Some therapies, such as SER-109 [39], are derived from purified stool; after purification, the remaining *Firmicutes* spores are packaged into a capsule that can be administered orally. Other therapies, such as VE303 [40], comprise eight individually chosen bacterial strains prepared into a capsule that can be administered orally. VER303 recently successfully finished a phase two trial [41], and SER-109 was recently FDA-approved for treating recurrent *C. difficile* infections. This demonstrates such therapies' safety, efficacy, and potential to replace current FMT-based treatments. These successes additionally pave the way for future microbiome-mediated therapeutics for other diseases.

Improved strain-level insights into the human microbiome will substantially benefit the development of such therapeutics along three axes. First, expanding on the joint host and bacterial GWAS discussed in the previous section, joint host and strain-level microbial genome-wide association studies (mGWAS) could reveal novel mechanisms or identify biomarkers of health and disease. The planned expansion of large biobanks such as All of Us [42] to include human microbiome data could provide the necessary cohort sizes to identify such associations. Early host-microbiome association studies were focused on associating specific species with disease [43] but often failed to identify clear links [44]. More recently, Zahavi et al. performed an mGWAS at a lower resolution, associating specific bacterial SNPs to host body-mass index (BMI) [45], though they did not include host genetics. They identified 40 bacterial SNPs associated with host BMI, which were replicated in two independent cohorts. These SNPs were primarily located in energy production and conversion genes, highlighting the potential to gain insight into mechanisms. While promising, we envision such studies could benefit from better representation of bacterial pangenome-wide genetic variation since Zahavi et al.'s approach used a single reference per bacterial species.

Second, improved insights into strain-level genetics will aid the engineering of

therapeutic strains that can overcome colonization resistance [46, 47]. For example, Zhao et al. identified within-host adaptive mutations in *Bacteroides fragilis*, suggesting potential adaptation to the host diet, pressure from phages, or pressure from the host immune system [48]. They could identify these mutations using an extensive collection of sequenced isolates from a single person spanning several years. Strain-resolved metagenomics would substantially help scale such analysis to larger cohorts, omitting the requirement for large-scale isolate sequencing. The identified adaptive alleles would inform the engineering of strains more likely to stably colonize the gut or could out-compete a pathogen [49].

Third, insight into strain-level community dynamics will help elucidate strain-strain interactions and improve the ability to model and predict microbiome dynamics in response to perturbations. Microbiome dynamics are frequently modeled using the generalized Lotka-Volterra ordinary differential equation model, and several earlier works have used Bayesian approaches to infer the pairwise species interaction parameters from several microbiome time series [50, 51, 52]. For example, Gibson et al.'s model identified species that suppressed *C. difficile*, highlighting how such models could help design consortia of strains to be used as therapeutic [51]. Dynamical system models can also predict a community's response and stability when perturbed by, for example, antibiotics or a probiotic set of strains [53]. While current approaches were all limited to the species level, increasing the resolution of such models to include strain-strain interactions could identify strain-specific competitive or cooperative behavior that could help design a minimal set of strains for therapeutic use.

However, technical challenges have hindered the strain-level characterization of microbial communities. The inherently limited genome context obtained from short reads complicates metagenomic assembly and reference-assisted approaches. Metagenomic assemblies are often highly fragmented, especially in same-species strain mixtures, because of a shared core and variable accessory genome [54]. In Chapter 3, we showed that StrainGE could effectively deconvolve strain mixtures and identify low abundance strains from metagenomic data; however, the application depends on a phylogeny-spanning reference database. Obtaining strain-specific variant calls is an additional challenge since only reads in each reference's unique genome content will map unambiguously.

Recent advances in the quality and throughput of long-read sequencing technologies hold great promise for strain-resolved assembly of metagenomes. Long reads enable genome assemblies with much higher contiguity, and several studies have reported complete, circular chromosomes assembled from metagenomic data [55, 56]. Even when contigs were incomplete, an additional contig binning step could compute lineage-specific bins representing near-complete assemblies of strains [55]. These contiguous, near-complete assemblies were possible for species at high abundance and if no other same-species strains were present.

However, in the case of same-species strain mixtures, recent benchmarking showed an increased risk of misassembly and decreased contiguity [57]. Most assemblers are *strain-oblivious* [58], i.e., they do not try to separate reads from different strains. If two same-species strains have substantially diverged (e.g., differ more

than a few percentage points), read overlaps from different strains will be distinct enough to generate separate contigs; if strains are more closely related, assemblers will generate collapsed contigs that represent mosaics of multiple strains [56].

Methods to improve strain-level genome assembly from metagenomes are an active area of research. The problem is similar to haplotype phasing in polyploid organisms, and Vicedomini *et al.* apply a phasing algorithm to recover strain-specific contigs from a strain-oblivious assembly [58]. In contrast to haplotype phasing, however, where chromosome copies are expected to be sequenced evenly, strain abundances can differ extensively, resulting in more variable allele frequencies. Linking reads with variants having similar allele frequencies could help partition reads from different strains [59, 60].

Another avenue for methodological improvements includes the pooled analysis of related samples since many microbial communities are studied longitudinally. For example, Latent Strain Analysis (LSA) partitions reads before assembly based on covarying patterns of k-mer sequencing depths across samples [61]. LSA was initially designed for short Illumina reads, but given recent progress in lowering sequencing error rates in long reads, such an approach could also benefit current long-read methods.

Metagenomic assembly remains challenging for low-abundance species because of insufficient read coverage, while many are of clinical importance. Reference-assisted methods will remain an alternative to assembly to characterize such species. For example, StrainGE (Chapter 3) could be adapted to support long reads. The longer reads, which provide more genome context, should result in fewer ambiguous alignments, enabling more accurate characterization of larger portions of strains' genomes.

Future improvements of StrainGE could include using pangenome reference graphs. StrainGE currently inherits the limitations of using single references to characterize bacterial genomes. It occasionally reports multiple reference genomes for a single strain because no reference accurately reflects its genome. To solve this issue, future algorithms could build a "strain-specific" reference that includes genome content from multiple database references. Long reads could enable the detection of such breakpoints, where the algorithm would need to switch to a locus on another reference.

We expect these technical challenges to be overcome in the next decade, which will provide valuable insights into the principles underlying microbiome community assembly, community dynamics, and its role in human health. Such insights will advance the development of microbiome-based treatments for diseases such as obesity, IBD, or recurrent urinary tract infections (Chapter 4).

Improved epidemiology of antimicrobial resistance

The rise of antimicrobial resistance (AMR) is already an urgent threat to human health care, causing millions of deaths yearly [62]. Few novel classes of antibiotics are in the pipeline, and annual deaths are expected to rise rapidly in the coming decades [63]. While recent machine-learning-based approaches yielded promising new antibiotic

candidates [64, 65], it could take years before such compounds are approved for clinical use.

An orthogonal approach to counter AMR is through increased surveillance and implementing measures that reduce the dissemination of AMR genes. In this thesis, we have discussed the benefits of genomic epidemiology for tracking strains, e.g., to infer the source of an ongoing outbreak in a hospital or to track strains over time in the human gut (Chapter 3). In our highly connected society, large-scale surveillance of farms, hospitals, or nursing homes is essential to gaining insights into bacteria's travel routes and their resistance genes [63].

StrainGE (Chapter 3) can aid in characterizing bacterial and AMR gene travel routes. For example, Worby et al. used StrainGE to find frequent acquisition of resistant *Enterobacteriaceae* strains in international travelers [66]. Another future application of StrainGE could be tracking strains using sequenced "plate swipes" [67]. Instead of sequencing isolates from a plate culture, the entire plate is deeply sequenced, and the data thus represents DNA from all strains present, which StrainGE could deconvolve. This would enable more sensitive detection of strains present in the sampled environment, enabling more accurate inference of transmission links.

One often overlooked aspect of AMR epidemiology is the spread of individual plasmids as opposed to the bacteria themselves. Plasmids are important vectors for disseminating AMR genes since they are frequently transferred horizontally [68]. Tracking plasmids, however, has been hampered due to the technical challenges of characterizing them with short reads. Plasmids are diverse, frequently recombine, and often highly repetitive. The high repetitiveness complicates their *de novo* assembly, and because of their diversity and frequent rearrangements, a single reference genome will rarely accurately represent a set of plasmids [69].

The importance of tracking plasmids is demonstrated by Salamzade et al. [70], where they discovered a set of plasmids harboring multiple AMR genes that had persisted for over ten years at Massachusetts General Hospital (MGH) despite limited evidence for large-scale bacterial outbreaks. These plasmids were acquired independently by multiple bacterial species, causing hard-to-treat infections. This suggests the presence of a reservoir at MGH where imported bacterial strains can acquire these plasmids and become multi-drug-resistant.

Detecting these plasmids was possible because long-read sequencing enabled the construction of complete, circularized assemblies. Long-read sequencing, however, is still too expensive for large-scale genomic surveillance. Biases in sampling influence the reconstructed evolutionary trees and subsequent inference of transmission links [71, 72]. Increased sampling density of the locations under study results in more accurate inferences of transmission networks [73]. For this reason, we expect Illumina sequencing to remain the most commonly used sequencing platform for large-scale genomic surveillance since it achieves the highest throughput and is the most cost-effective.

New study designs focused on tracking plasmids could use a hybrid approach: long-read sequencing for a subset of samples and short-read sequencing for all other samples. Such a setup would combine accurate long-read plasmid assemblies with the scale of short-read sequencing, in which a panel of long-read assembled plasmids

will aid in tracking plasmids in short-read samples. To maximize the probability that a short-read sample contains a plasmid related to a plasmid in a long-read sample, plasmid content can be estimated beforehand with short reads to guide the selection of samples that best represent plasmid content among the data set.

Novel computational tools are required to characterize and track plasmids in hybrid setups. Similar to a bacterial pangenome reference graph, a pan-plasmid reference graph could be a potential solution to infer what (pieces of) long-read assembled plasmids are present in the short-read samples. We believe that POASTA (Chapter 2) could again be an essential component of such a pipeline to construct and analyze pan-plasmid graphs. Additionally, because plasmids frequently rearrange, novel relatedness measures will be required to infer transmission links accurately. Two recent approaches to achieve that include SHIP [74] and Pling [75].

The spread and persistence of plasmids within hospitals and their role in disseminating AMR genes have, until recently, been underappreciated because of the difficulties in studying them. Improved tools to characterize and track individual plasmids will give insight into their reservoirs and travel routes, informing on measures to curb their spread. Such measures are desperately needed to stem the ever-increasing burden of antimicrobial resistance infections worldwide.

Final remarks

Advances in sequencing technology have revealed the extensive genetic diversity among many bacterial species. Developing novel algorithms that consider this diversity will greatly aid in identifying the genetic basis of phenotypes, understanding microbial community dynamics, and tracking bacteria or their mobile genetic elements. This will lead to increased insight into the molecular mechanisms of disease, advance the development of novel microbiome therapeutics, and reduce the burden of (resistant) bacterial infections.

References

1. Tettelin H et al. Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial “pan-genome”. Proceedings of the National Academy of Sciences 2005 Sep; 102. Publisher: Proceedings of the National Academy of Sciences:13950–5
2. Tettelin H and Medini D, eds. The Pangenome: Diversity, Dynamics and Evolution of Genomes. eng. Cham (CH): Springer, 2020
3. Land M et al. Insights from 20 years of bacterial genome sequencing. en. Functional & Integrative Genomics 2015 Mar; 15:141–61
4. Thorpe HA, Bayliss SC, Sheppard SK, and Feil EJ. Piggy: a rapid, large-scale pan-genome analysis tool for intergenic regions in bacteria. GigaScience 2018 Apr; 7:giy015
5. The Computational Pan-Genomics Consortium. Computational pan-genomics: status, promises and challenges. Briefings in Bioinformatics 2018 Jan; 19:118–35
6. Hickey G et al. Pangenome graph construction from genome alignments with Minigraph-Cactus. en. Nature Biotechnology 2023 May. Publisher: Nature Publishing Group:1–11
7. Garrison E et al. Building pangenome graphs. en. Pages: 2023.04.05.535718 Section: New Results. 2023 Apr

8. Liao WW et al. A draft human pangenome reference. en. Nature 2023 May; 617. Number: 7960 Publisher: Nature Publishing Group:312–24

9. Sirén J et al. Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. Science 2021 Dec; 374. Publisher: American Association for the Advancement of Science:abg8871

10. Li J et al. An integrated catalog of reference genes in the human gut microbiome. en. Nature Biotechnology 2014 Aug; 32. Publisher: Nature Publishing Group:834–41

11. Thomas AM and Segata N. Multiple levels of the unknown in microbiome research. en. BMC Biology 2019 Jun; 17:48

12. Kobras CM, Fenton AK, and Sheppard SK. Next-generation microbiology: from comparative genomics to gene function. Genome Biology 2021 Apr; 22:123

13. Opijnen T van, Bodi KL, and Camilli A. Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms. en. Nature Methods 2009 Oct; 6. Publisher: Nature Publishing Group:767–72

14. Cain AK et al. A decade of advances in transposon-insertion sequencing. en. Nature Reviews Genetics 2020 Sep; 21. Publisher: Nature Publishing Group:526–40

15. Coe KA et al. Multi-strain Tn-Seq reveals common daptomycin resistance determinants in Staphylococcus aureus. en. PLOS Pathogens 2019 Nov; 15. Publisher: Public Library of Science:e1007862

16. Zhu L et al. Novel Genes Required for the Fitness of Streptococcus pyogenes in Human Saliva. mSphere 2017 Nov; 2. Publisher: American Society for Microbiology:10.1128/mspheredirect.00460–17

17. Goh KGK et al. Genome-Wide Discovery of Genes Required for Capsule Production by Uropathogenic Escherichia coli. mBio 2017 Oct; 8. Publisher: American Society for Microbiology:10.1128/mbio.01558–17

18. McCarthy AJ, Stabler RA, and Taylor PW. Genome-Wide Identification by Transposon Insertion Sequencing of Escherichia coli K1 Genes Essential for In Vitro Growth, Gastrointestinal Colonizing Capacity, and Survival in Serum. Journal of Bacteriology 2018 Mar; 200. Publisher: American Society for Microbiology:10.1128/jb.00698–17

19. Price MN et al. Mutant phenotypes for thousands of bacterial genes of unknown function. en. Nature 2018 May; 557. Publisher: Nature Publishing Group:503–9

20. Visscher PM et al. 10 Years of GWAS Discovery: Biology, Function, and Translation. The American Journal of Human Genetics 2017 Jul; 101:5–22

21. Falush D. Bacterial genomics: Microbial GWAS coming of age. Nature Microbiology 2016; 1

22. Power RA, Parkhill J, and Oliveira T de. Microbial genome-wide association studies: lessons from human GWAS. en. Nature Reviews Genetics 2017 Jan; 18. Number: 1 Publisher: Nature Publishing Group:41–50

23. Sheppard SK et al. Genome-wide association study identifies vitamin B5 biosynthesis as a host specificity factor in Campylobacter. Proceedings of the National Academy of Sciences 2013 Jul; 110. Publisher: Proceedings of the National Academy of Sciences:11923–7

24. Earle SG et al. Identifying lineage effects when controlling for population structure improves power in bacterial association studies. Nature Microbiology 2016; 1. Publisher: Nature Publishing Group:1–8

25. Lees JA et al. Joint sequencing of human and pathogen genomes reveals the genetics of pneumococcal meningitis. en. Nature Communications 2019 May; 10. Publisher: Nature Publishing Group:2176

26. Collins C and Didelot X. A phylogenetic method to perform genome-wide association studies in microbes that accounts for population structure and recombination. en. PLOS Computational Biology 2018 Feb; 14. Publisher: Public Library of Science:e1005958

27. Lees JA et al. Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes. en. Nature Communications 2016 Sep; 7. Publisher: Nature Publishing Group:12797

28. Lees JA et al. Improved Prediction of Bacterial Genotype-Phenotype Associations Using Interpretable Pangenome-Spanning Regressions. en. mBio 2020 Aug; 11. Publisher: American Society for Microbiology Section: Research Article

29. Read TD and Massey RC. Characterizing the genetic basis of bacterial phenotypes using genome-wide association studies: a new direction for bacteriology. Genome Medicine 2014 Nov; 6:109

30. Jaillard M et al. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. en. PLOS Genetics 2018 Nov; 14. Publisher: Public Library of Science:e1007758

31. Tonkin-Hill G et al. Producing polished prokaryotic pangenomes with the Panaroo pipeline. Genome Biology 2020 Jul; 21:180

32. Tonkin-Hill G, Corander J, and Parkhill J. Challenges in prokaryote pangenomics. Microbial Genomics 2023; 9. Publisher: Microbiology Society,:001021

33. Eiseman B, Silen W, Bascom GS, and Kauvar AJ. Fecal enema as an adjunct in the treatment of pseudomembranous enterocolitis. eng. Surgery 1958 Nov; 44:854–9

34. Tariq R et al. Fecal Microbiota Transplantation for Recurrent Clostridium difficile Infection Reduces Recurrent Urinary Tract Infection Frequency. Clinical Infectious Diseases 2017 Oct; 65:1745–7

35. Wang T, Kraft CS, Woodworth MH, Dhere T, and Eaton ME. Fecal Microbiota Transplant for Refractory Clostridium difficile Infection Interrupts 25-Year History of Recurrent Urinary Tract Infections. Open Forum Infectious Diseases 2018 Feb; 5:ofy016

36. Papanicolas LE, Gordon DL, Wesselingh SL, and Rogers GB. Improving Risk–Benefit in Faecal Transplantation through Microbiome Screening. English. Trends in Microbiology 2020 May; 28. Publisher: Elsevier:331–9

37. Marcella C et al. Systematic review: the global incidence of faecal microbiota transplantation-related adverse events from 2000 to 2020. en. Alimentary Pharmacology & Therapeutics 2021; 53. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/apt.16148:33–42

38. Gilbert JA. Microbiome therapy for recurrent Clostridioides difficile. English. The Lancet Microbe 2022 May; 3. Publisher: Elsevier:e334

39. Feuerstadt Paul et al. SER-109, an Oral Microbiome Therapy for Recurrent Clostridioides difficile Infection. New England Journal of Medicine 2022 Jan; 386. Publisher: Massachusetts Medical Society _eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMoa2106516:220–9

40. Dsouza M et al. Colonization of the live biotherapeutic product VE303 and modulation of the microbiota and metabolites in healthy volunteers. English. Cell Host & Microbe 2022 Apr; 30. Publisher: Elsevier:583–598.e8

41. Louie T et al. VE303, a Defined Bacterial Consortium, for Prevention of Recurrent Clostridioides difficile Infection: A Randomized Clinical Trial. JAMA 2023 Apr; 329:1356–66

42. Bick AG et al. Genomic data in the All of Us Research Program. en. Nature 2024 Mar; 627. Publisher: Nature Publishing Group:340–6

43. Gilbert JA et al. Microbiome-wide association studies link dynamic microbial consortia to disease. en. Nature 2016 Jul; 535. Number: 7610 Publisher: Nature Publishing Group:94–103

44. Yan Y, Nguyen LH, Franzosa EA, and Huttenhower C. Strain-level epidemiology of microbial communities and the human microbiome. Genome Medicine 2020 Aug; 12:71

45. Zahavi L et al. Bacterial SNPs in the human gut microbiome associate with host BMI. en. Nature Medicine 2023 Nov; 29. Publisher: Nature Publishing Group:2785–92

46. Buffie CG and Pamer EG. Microbiota-mediated colonization resistance against intestinal pathogens. en. Nature Reviews Immunology 2013 Nov; 13. Publisher: Nature Publishing Group:790–801

47. Sorbara MT and Pamer EG. Microbiome-based therapeutics. en. Nature Reviews Microbiology 2022 Jun; 20. Publisher: Nature Publishing Group:365–80

48. Zhao S et al. Adaptive Evolution within Gut Microbiomes of Healthy People. English. Cell Host & Microbe 2019 May; 25. Publisher: Elsevier:656–667.e8

49. Lieberman TD. Detecting bacterial adaptation within individual microbiomes. Philosophical Transactions of the Royal Society B: Biological Sciences 2022 Aug; 377. Publisher: Royal Society:20210243

50. Venturelli OS et al. Deciphering microbial interactions in synthetic human gut microbiome communities. Molecular Systems Biology 2018 Jun; 14. Publisher: John Wiley & Sons, Ltd:e8157

51. Gibson TE and Gerber GK. Robust and Scalable Models of Microbiome Dynamics. 2018

52. Kumar M, Ji B, Zengler K, and Nielsen J. Modelling approaches for studying the microbiome. en. Nature Microbiology 2019 Aug; 4. Publisher: Nature Publishing Group:1253–67

53. Bucci V et al. MDSINE: Microbial Dynamical Systems INference Engine for microbiome time-series analyses. Genome Biol. 2016; 17. Publisher: Genome Biology:1–17

54. Ayling M, Clark MD, and Leggett RM. New approaches for metagenome assembly with short reads. Briefings in Bioinformatics 2020 Mar; 21:584–94

55. Bickhart DM et al. Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. en. Nature Biotechnology 2022 May; 40. Publisher: Nature Publishing Group:711–9

56. Feng X, Cheng H, Portik D, and Li H. Metagenome assembly of high-fidelity long reads with hifiasm-meta. en. Nature Methods 2022 Jun; 19. Publisher: Nature Publishing Group:671–4

57. Yorki S et al. Comparison of long- and short-read metagenomic assembly for low-abundance species and resistance genes. Briefings in Bioinformatics 2023 Mar; 24:bbad050

58. Vicedomini R, Quince C, Darling AE, and Chikhi R. Strainberry: automated strain separation in low-complexity metagenomes using long reads. en. Nature Communications 2021 Jul; 12. Publisher: Nature Publishing Group:4485

59. Luo C et al. ConStrains identifies microbial strains in metagenomic datasets. Nat. Biotechnol. 2015 Oct; 33:1045–52

60. Smillie CS et al. Strain Tracking Reveals the Determinants of Bacterial Engraftment in the Human Gut Following Fecal Microbiota Transplantation. Cell Host Microbe 2018; 23. Publisher: Elsevier Inc.:229–240.e5

61. Cleary B et al. Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. Nat. Biotechnol. 2015 Oct; 33. Publisher: Nature Publishing Group:1053–60

62. Murray CJL et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. English. The Lancet 2022 Feb; 399. Publisher: Elsevier:629–55

63. Centers for Disease Control and Prevention (U.S.) Antibiotic resistance threats in the United States, 2019. Tech. rep. Centers for Disease Control and Prevention (U.S.), 2019 Nov

64. Wong F et al. Discovery of a structural class of antibiotics with explainable deep learning. en. Nature 2024 Feb; 626. Publisher: Nature Publishing Group:177–85

65. Swanson K et al. Generative AI for designing and validating easily synthesizable and structurally novel antibiotics. en. Nature Machine Intelligence 2024 Mar; 6. Publisher: Nature Publishing Group:338–53

66. Worby CJ et al. Gut microbiome perturbation, antibiotic resistance, and Escherichia coli strain dynamics associated with international travel: a metagenomic analysis. English. The Lancet Microbe 2023 Oct; 4. Publisher: Elsevier:e790–e799

67. Mäklin T et al. Bacterial genomic epidemiology with mixed samples. Microbial Genomics 2021; 7. Publisher: Microbiology Society,:000691

68. Rozwandowicz M et al. Plasmids carrying antimicrobial resistance genes in Enterobacteriaceae. Journal of Antimicrobial Chemotherapy 2018 May; 73:1121–37

69. Arredondo-Alonso S, Willems RJ, Schaik W van, and Schürch AC. On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. Microbial Genomics 2017 :1–8

70. Salamzade R et al. Inter-species geographic signatures for tracing horizontal gene transfer and long-term persistence of carbapenem resistance. Genome Medicine 2022 Apr; 14:37

71. Nelson KN et al. Modeling Missing Cases and Transmission Links in Networks of Extensively Drug-Resistant Tuberculosis in KwaZulu-Natal, South Africa. American Journal of Epidemiology 2020 Jul; 189:735–45

72. Liu P, Song Y, Colijn C, and MacPherson A. The impact of sampling bias on viral phylogeographic reconstruction. en. PLOS Global Public Health 2022 Sep; 2. Publisher: Public Library of Science:e0000577

73. Layan M et al. Impact and mitigation of sampling bias to determine viral spread: Evaluating discrete phylogeography through CTMC modeling and structured coalescent model approximations. Virus Evolution 2023 Jan; 9:vead010

74. Teixeira M, Pillay S, Urhan A, and Abeel T. SHIP: identifying antimicrobial resistance gene transfer between plasmids. Bioinformatics 2023 Oct; 39:btad612

75. Frolova D et al. Applying rearrangement distances to enable plasmid epidemiology with pling. en. Pages: 2024.06.12.598623 Section: New Results. 2024 Jun

A

Supplemental Materials - Fast and exact gap-affine partial order alignment with POASTA

A.1. Supplemental Figures

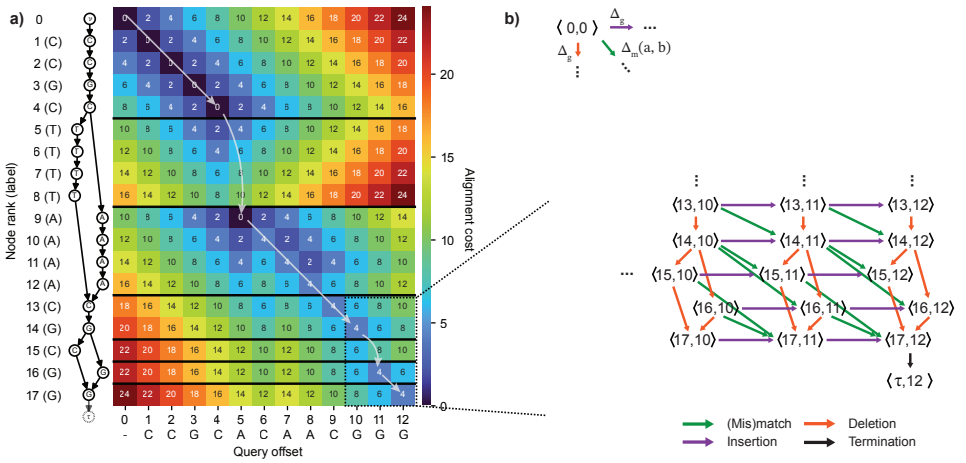

Figure A.1: (a) Example computation of aligning "CCGCACAACGGG" to a POA graph, with mismatch cost $\Delta_x = 4$, and gap cost $\Delta_g = 2$. The white arrows indicate the optimal alignment path. **(b)** A subgraph of the full *alignment graph*, corresponding to POA graph nodes 13-17, and query offset 10-12. A node $\langle v, o \rangle$ in the alignment graph represents a cursor to a node in the POA graph v and a query offset o. The various alignment operations ((mis)match, insertion, deletion) correspond to different kinds of edges. Insertion and deletion edges are weighted with the gap cost Δ_g, and (mis)match edges with a function $\Delta_m(a, b) = \{\Delta_x$ if $a \neq b$, and 0 otherwise.

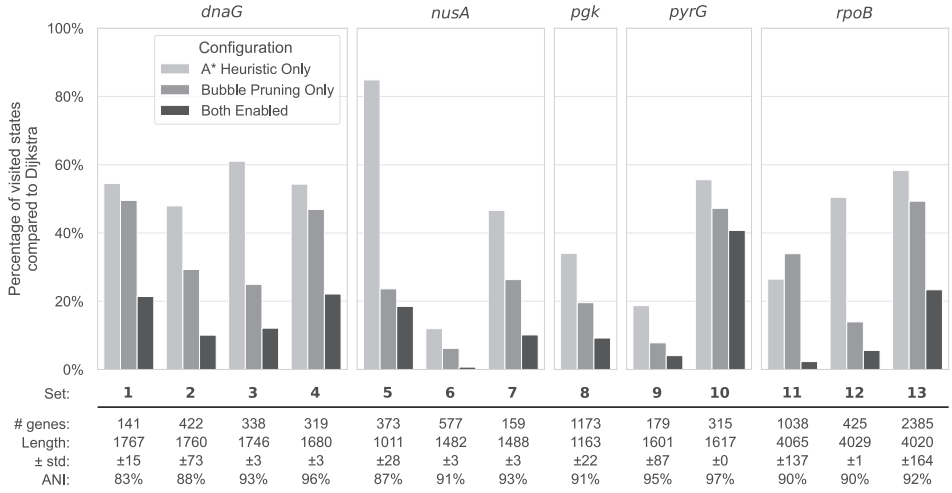

Figure A.2: POASTA's A* heuristic and superbubble-informed pruning substantially reduces the number of visited alignment states. Barplot indicating the percentage of visited alignment states of three POASTA configurations compared to a Dijkstra baseline (i.e., with both the A* heuristic and bubble pruning disabled).

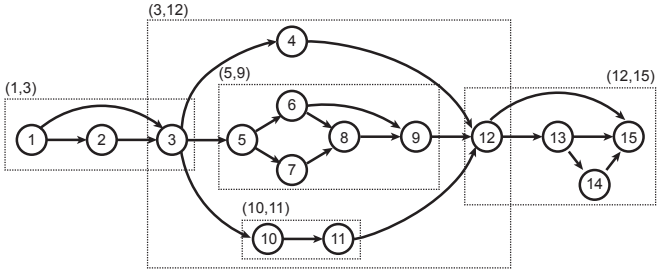

Figure A.3: An example graph containing multiple superbubbles. Each superbubble is marked with a dotted rectangle, labeled with its (*entrance*, *exit*). Superbubbles can be nested within each other: superbubbles (5, 9) and (10, 11) are contained within superbubble (3, 12). Superbubble (10, 11) is an example of a superbubble without an interior.

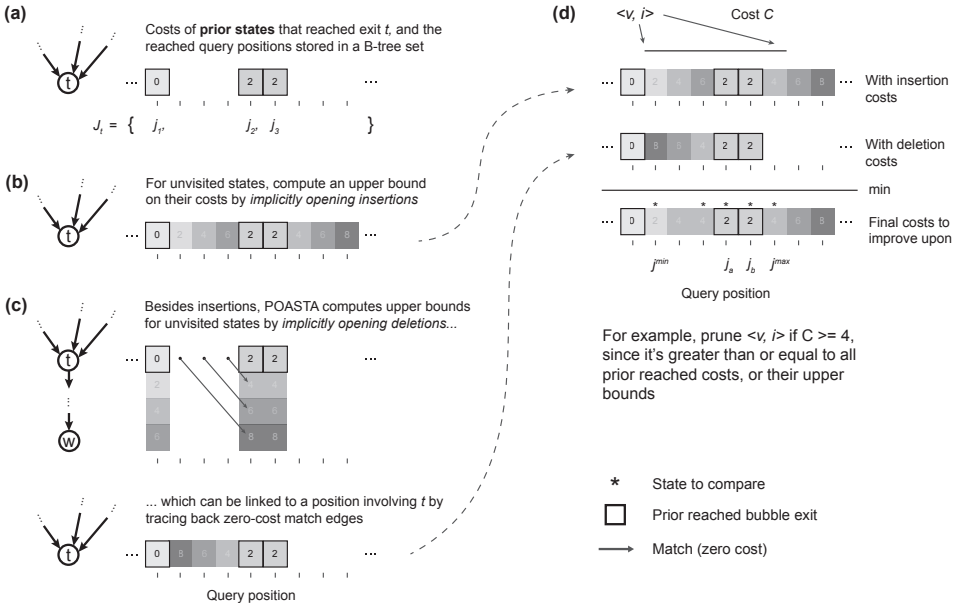

Figure A.4: POASTA considers all prior reached bubble exits when testing if a state can be pruned. **(a)** Example that shows alignment states that reached exit t previously at query position j_1, j_2 and j_3 (bordered squares). Positions are stored as an ordered set J_t. **(b)** Example upper bounds for unvisited states by implicitly opening insertions (squares without border). **(c)** Example upper bounds for unvisited states by implicitly opening deletions (top; squares without border), reaching some node w downstream of t. Tracing back zero-cost match edges (black arrows) from opened deletions enables linking the upper bound to a query position for t (bottom). **(d)** Example of how POASTA determines whether to prune a state $\langle v, i \rangle$ reached with alignment cost c. POASTA determines the lowest upper bound from implicitly opened gaps for each query position by taking the minimum cost of an implicitly opened insertion or deletion. POASTA only needs to compare the alignment cost c with the upper bounds for states marked with a *. All examples use the gap-linear cost model with $\Delta_g = 2$.

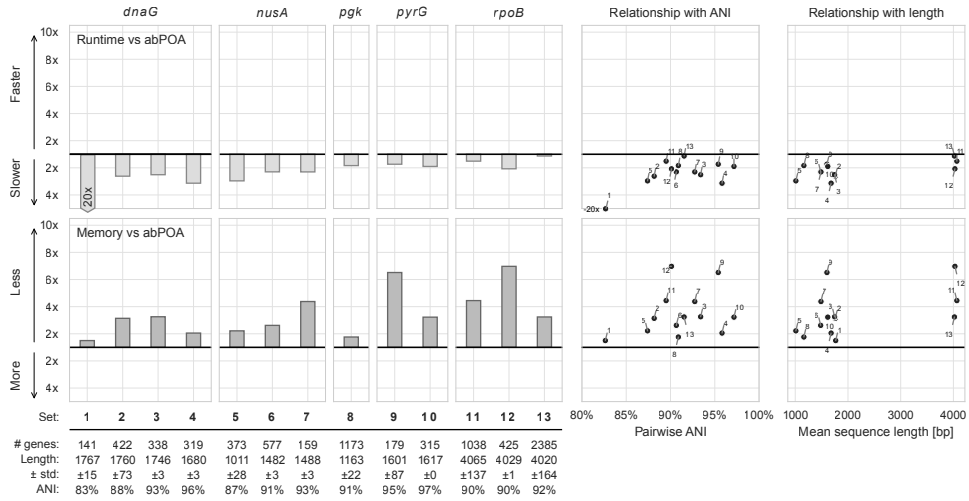

Figure A.5: abPOA is, on average, 3.5x faster than POASTA, but doesn't guarantee optimal alignment and uses more memory. **(a)** Relative runtime of POASTA compared to abPOA for each set of gene sequences. **(b)** The relationship between pairwise ANI of each gene sequence set and POASTA's relative runtime. **(c)** The relationship between mean sequence length and POASTA's relative runtime. **(d)** Relative memory usage of POASTA compared to abPOA for each set of gene sequences. **(e)** The relationship between pairwise ANI of each sequence set and POASTA's relative memory usage. **(f)** The relationship between the mean sequence length of each sequence set and POASTA's relative memory usage.

A.2. Supplemental Methods

The connection between dynamic programming recurrence and the alignment graph

Gap-linear alignment costs

The conventional dynamic programming (DP) recurrence for POA with gap-linear costs is [1]:

$$S_{v,i} = \min \begin{cases} S_{u,i-1} + \Delta(\sigma(v), q_i) & \forall u : (u,v) \in E \\ S_{u,i} + \Delta_g & \forall u : (u,v) \in E \\ S_{v,i-1} + \Delta_g \end{cases} \tag{A.1}$$

Here, $\sigma(v) \to \Sigma$ returns the node label for a node v, and $\Delta(a,b) \to \mathbb{Z}$ is a function that returns the match cost Δ_m if $a = b$, and mismatch cost Δ_x if $a \neq b$. The three cases correspond to a (mis)match between the graph and query, opening or extending a deletion, and opening or extending an insertion.

To translate the recurrence to edges in an alignment graph, we define the edge set E^A as follows. Edges connect two alignment states $\langle u, i\rangle \to \langle v, j\rangle$ if one of the following conditions hold:

- **Match and mismatch.** $(u,v) \in E$, $v \neq \tau$, and $i + 1 = j$, $j \leq m$, with the (mis)match cost $\Delta(\sigma(v), q_j)$ as weight;
- **Deletion.** $(u,v) \in E$, $v \neq \tau$, and $i = j$, with the gap cost Δ_g as weight;
- **Insertion.** $u = v$, $u, v \neq \tau$, and $i + 1 = j$, $j \leq m$, with the gap cost Δ_g as weight;
- **Termination.** $(u,v) \in E$, $v = \tau$, and $i = j = m$, with zero cost.

These edges (except for the termination edge) are analogous to the different cases in Equation A.1. We note that edges originating from the special start node v are analogous to the base cases in the dynamic programming problem, i.e., the first row and column of the matrix initialized with the gap costs. Edges towards to special termination node τ have no analogous case in the dynamic programming recurrence and therefore have zero cost.

Gap-affine alignment costs

To compute the gap-affine alignment, the Smith-Waterman-Gotoh (SWG) DP recurrence for affine pairwise alignment [2] can be adapted to POA as follows:

$$\begin{cases} I_{v,i} & = \min\{M_{v,i-1} + \Delta_o + \Delta_e,\ I_{v,i-1} + \Delta_e\} \\ D_{v,i} & = \min\{M_{u,i} + \Delta_o + \Delta_e,\ D_{u,i} + \Delta_e\} \\ & \quad \forall u : (u,v) \in E \\ M_{v,i} & = \min\{I_{v,i},\ D_{v,i},\ M_{u,i-1} + \Delta(\sigma(v), q_i)\} \\ & \quad \forall u : (u,v) \in E \end{cases} \tag{A.2}$$

The two cases for $I_{v,i}$ correspond to opening an insertion and extending an insertion; the two cases for $D_{v,i}$ correspond to opening a deletion and extending a deletion; and the three cases for $M_{v,i}$ correspond to closing an insertion, deletion, or a (mis)match.

To extend the alignment graph formulation to the gap-affine model, with gap open cost Δ_o and gap extend cost Δ_e, we define the node set of the gap-affine alignment graph as follows: $V^A = (V \times \{0, \ldots, m\} \times \{M, D, I\})$. In other words, for each pair $v \in V, i \in [0, m]$, we now have three possible alignment states: $\langle v, i, M\rangle$, $\langle v, i, D\rangle$, $\langle v, i, I\rangle$, representing the match, deletion, and insertion state, respectively. Edges in the gap-affine alignment graph are defined as follows:

- **Edges ending in the insertion state**
  - $\langle u, i, M\rangle \to \langle u, i+1, I\rangle$, $u \neq \tau$, $i+1 \leq m$, weighted with gap open cost $\Delta_o + \Delta_e$
  - $\langle u, i, I\rangle \to \langle u, i+1, I\rangle$, $u \neq \tau$, $i+1 \leq m$, weighted with gap extend cost Δ_e
- **Edges ending in the deletion state**
  - $\langle u, i, M\rangle \to \langle v, i, D\rangle$, $(u, v) \in E$, $v \neq \tau$, weighted with gap open cost $\Delta_o + \Delta_e$
  - $\langle u, i, D\rangle \to \langle v, i, D\rangle$, $(u, v) \in E$, $v \neq \tau$, weighted with gap extend cost Δ_e
- **Edges ending in the (mis)match state**
  - $\langle u, i, M\rangle \to \langle v, i+1, M\rangle$, $(u, v) \in E$, $v \neq \tau$, $i+1 \leq m$, with (mis)match cost $\Delta(\sigma(v), q_{i+1})$
  - $\langle u, i, I\rangle \to \langle u, i, M\rangle$, $u \neq \tau$, weighted with zero cost
  - $\langle u, i, D\rangle \to \langle u, i, M\rangle$, $u \neq \tau$, weighted with zero cost
- **Termination edges**
  - $\langle u, m, M\rangle \to \langle \tau, m, M\rangle$, $(u, \tau) \in E$, weighted with zero cost

These edges are analogous to the cases in Equation A.2.

Proof of minimum number of indel edges

Given an alignment state $\langle u, i\rangle$, let $d_{u,\tau}^{\min}$ and $d_{u,\tau}^{\max}$ be the minimum and maximum path length in the POA graph from u to end node τ. We additionally compute the length of the unaligned query sequence $l_r = m - i$. The minimum number of indel edges to traverse is then:

Definition 3 (Minimum number of indel edges)

$$N_g^{\min} = \begin{cases} l_r - (d_{u,\tau}^{\max} - 1) & \text{if } d_{u,\tau}^{\max} - 1 < l_r \\ (d_{u,\tau}^{\min} - 1) - l_r & \text{if } d_{u,\tau}^{\min} - 1 > l_r \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

We subtract one from $d_{u,\tau}^{\min}$ and $d_{u,\tau}^{\max}$ to exclude the edge towards τ.

Proof Let $\mathcal{W} \subset V$ be the subset of POA graph nodes with an outgoing edge to τ, i.e., $\mathcal{W} = \{w : (w, \tau) \in E\}$. By definition of the alignment graph, the alignment termination state is only reachable from alignment states $\langle w, m\rangle : w \in \mathcal{U}$. We will prove each case separately.

In the first case, $d_{u,\tau}^{\max} - 1 < l_r$. By definition of $\mathcal{W}$, $\exists w \in \mathcal{W}$ such that $d_{u,w} = d_{u,\tau}^{\max} - 1$, i.e., excluding the last edge towards τ from the maximum path length. The presence of this maximum length path in the POA graph implies a corresponding path of all (mis)match edges in the alignment graph, reaching the alignment state $\langle w, j\rangle, w \in \mathcal{W}, j = i + d_{v,w}$. Since this traversed the maximum length path in the POA graph, j is also the maximum query position reachable from $\langle u, i\rangle$. Since $d_{u,w} < l_r$, we infer that $j < m$. This means that the alignment termination state is not reachable from $\langle u, j\rangle$, and at least $m - j = l_r - (d_{v,\tau}^{\max} - 1)$ insertion edges need to be traversed to be able to reach the alignment termination state.

In the second case, $d_{v,\tau}^{\min} - 1 > l_r$. Similarly as above, $\exists w \in \mathcal{W}$, such that $d_{u,w} = d_{u,\tau}^{\min} - 1$, i.e., excluding the last edge towards τ from the minimum path length. To reach the alignment termination state from $\langle u, i\rangle$, we need to traverse at least $d_{u,w}$ (mis)match or deletion edges, since this is the minimum length path to the POA end node. We can, however, traverse only l_r (mis)match edges, since no (mis)match edges exist that would move the query position beyond the query sequence length m. After traversing l_r (mis)match edges, we would reach some state $\langle v, m\rangle$, with v being a node on the minimum path in the POA graph $u \to ... \to v \to ... \to w \to \tau$. To be able to reach the alignment termination state, we need to traverse at least $d_{v,\tau}^{\min} - 1 - l_r$ deletion edges.

In the last case, $d_{v,\tau}^{\min} - 1 < l_r < d_{v,\tau}^{\max} - 1$, which implies that there exist a path from $\langle u, i\rangle$ to $\langle \tau, m\rangle$ without the need to traverse any indel edges. □

Extension of the minimum gap cost heuristic function to the gap-affine model

To compute the minimum gap cost heuristic using gap-affine model, we need to take into account that insertion or deletion states do not need to incur the gap-open cost again.

A state $\langle v, i, M\rangle$ always needs to incur the gap-open cost, thus the heuristic is computed as follows:

Definition 4 $$h\langle v, i, M\rangle = \begin{cases} 0 & \text{if } N_g^{\min} = 0 \\ \Delta_o + N_g^{\min}\Delta_e & \text{otherwise} \end{cases}$$

A state $\langle v, i, I\rangle$ is already in insertion state and would not have to incur the gap open cost again if $d_{v,\tau}^{\max} - 1 < l_r$, since the minimum number of indel edges (as described above) are all insertion edges in that case. We compute the heuristic as follows:

Definition 5 $$h\langle v, i, I\rangle = \begin{cases} 0 & \text{if } N_g^{\min} = 0 \\ N_g^{\min}\Delta_e & \text{if } d_{v,\tau}^{\max} - 1 < l_r \\ \Delta_o + N_g^{\min}\Delta_e & \text{otherwise} \end{cases}$$

Similarly, for a state $\langle v, i, D\rangle$, we would not have to incur the gap open cost again if $d^{\min}_{v,\tau} - 1 > l_r$, since the minimum number of indel edges are all deletion edges in that case. The heuristic is computed as follows:

Definition 6 $h\langle v, i, D\rangle = \begin{cases} 0 & \text{if } N_g^{\min} = 0 \\ N_g^{\min}\Delta_e & \text{if } d^{\min}_{v,\tau} - 1 > l_r \\ \Delta_o + N_g^{\min}\Delta_e & \text{otherwise} \end{cases}$

Implementation details of superbubble-informed pruning

Effective detection of prunable states by computing implicitly opened gap costs

To test if a state $\langle v, i\rangle$ reached at cost C and contained in a superbubble (s, t) can be pruned, POASTA infers the range of states $\langle t, j^{\min}\rangle, \ldots, \langle t, j^{\max}\rangle$ reachable with zero-cost match edges (Methods; Main Text Figure 3c). A naive approach would scan the entire range $\langle t, j^{\min}\rangle, \ldots, \langle t, j^{\max}\rangle$ and assess whether all of those states were visited prior at a lower or equal cost to C. This would be ineffective for two reasons: 1) for larger and more complex bubbles, the range $j^{\min}, \ldots, j^{\max}$ can be quite large, and 2) at the time of testing, many of those states might not have been reached yet, thus without a known alignment cost to compare to C.

To more effectively detect prunable states, POASTA employs the following: First, POASTA tracks in a B-tree set on which query positions a superbubble exit t have been reached (Supplementary Figure A.4a). Then, POASTA uses the inherent ordering in a B-tree to quickly retrieve which query positions have reached bubble exit t in the range $j^{\min}, \ldots, j^{\max}$. Finally, POASTA computes upper bounds on the alignment costs for unvisited positions in this range by *implicitly opening gaps* from visited positions (Supplementary Figure A.4bc). A state $\langle v, i\rangle$ will be pruned if its alignment cost C is greater than or equal to the (upper bound on) costs for states $\langle t, j^{\min}\rangle, \ldots, \langle t, j^{\max}\rangle$ (Supplementary Figure A.4d). We will detail each step below.

First, the B-tree for each superbubble exit is an ordered set $J_t = \{j_1, j_2, \ldots, j_n\}$ of query positions on which a superbubble exit has been reached. Each time a state $\langle t, j\rangle$ is popped from the A* queue, POASTA inserts j into the B-tree for an exit t (Supplementary Figure A.4a).

Second, as discussed above, POASTA assesses whether a state $\langle v, i\rangle$ can be pruned by comparing its alignment cost C to the alignment costs of $\langle t, j^{\min}\rangle, \ldots, \langle t, j^{\max}\rangle$. Using the inherent ordering in the B-tree, POASTA can quickly find indices a and b using binary search such that $j^{\min} \leq j_a, \ldots, j_b \leq j^{\max}$, i.e., the list of visited query positions in the range $[j^{\min}, j^{\max}]$.

POASTA uses this list of visited query positions to compute upper bounds on the alignment cost for the *unvisited* query positions, i.e., states $\langle t, j'\rangle : j' \in [j^{\min}, j^{\max}], j' \notin J_t$, by *implicitly opening gaps*. We call this implicitly opening gaps since these upper bounds are computed on the fly, not recorded anywhere, and not included in the A* queue.

For example, if $\langle t, j\rangle$ was previously visited with a cost $C_{\langle t,j\rangle}$, then any unvisited state $\langle t, j'\rangle : j' > j$ could also be reached by opening an insertion from $\langle t, j\rangle$. In the

case of linear gap penalties, these states would then be reached at an alignment cost $C_{\langle t,j' \rangle} = C_{\langle t,j \rangle} + \Delta_g(j' - j)$. This is an upper bound on the cost for a state $\langle t, j' \rangle$, since there may exist a path to that state with a lower alignment cost (Supplementary Figure A.4b).

Besides opening an insertion, we could also open a deletion from a previously reached state $\langle t, j \rangle$, reaching some state $\langle w, j \rangle$ where w is a node downstream of t. In the case of linear gap penalties, this state would be reached with an alignment cost of $C_{\langle w,j \rangle} = C_j + \Delta_g d_{t,w}$, where $d_{t,w}$ is the path length between t and w. This is again an upper bound on the cost for state $\langle w, j \rangle$, since there may be other paths with lower alignment costs. We link the upper bound of $\langle w, j \rangle$ to a state involving exit t and a query position $j'' < j$ by noting that any alignment path from a state $\langle v, i \rangle$ to $\langle w, j \rangle$ would need to traverse an alignment state $\langle t, j'' \rangle : i \leq j'' \leq j$ since v is part of a superbubble with exit t. Tracing back the best-case scenario of zero-cost match edges from $\langle w, j \rangle$, we find that $j'' = j - d_{t,w}$. Thus, for $\langle v, i \rangle$ to improve over the upper bound for $\langle w, j \rangle$, it would also need to reach $\langle t, j - d_{t,w} \rangle$ with an alignment cost lower than $C_{\langle w,j \rangle}$ (Supplementary Figure A.4c).

Finally, while implicitly opening gaps enables computing upper bounds of the alignment cost for any state in the range $\langle t, j^{\min} \rangle, \ldots, \langle t, j^{\max} \rangle$, POASTA only needs to check a subset of positions in this range when assessing to prune a state $\langle v, i \rangle$. Specifically, POASTA exploits the fact that the cost of a gap linearly increases with its length. For example, if an exit t has been reached on query positions $j_1, j_2 \in J_T, j_2 \gg j_1$, the position with the *lowest* implicit insertion cost in the range $[j_1, j_2]$ would be $j_1 + 1$, since for all following positions, the insertion cost would only increase. Similarly, the position with the lowest implicit deletion cost in the range would be $j_2 - 1$. By comparing the (upper bound on) alignment costs for the subset of positions $\{j^{\min}, j^{\max}\} \cup \{j - 1, j, j + 1 : j \in J_t^{[a,b]}\}$ to the alignment cost C of a state $\langle v, i \rangle$, POASTA thus determines whether $\langle v, i \rangle$ can improve the alignment score over the entire range $\langle t, j^{\min} \rangle, \ldots, \langle t, j^{\max} \rangle$ (Supplementary Figure A.4d).

Detecting prunable states with gap-affine penalties

Superbubble-informed pruning is straightforward to adapt to the gap-affine cost model. One option would be to keep separate, ordered sets of reached positions J_t^M, J_t^D, J_t^I for matches, deletions, and insertions. When testing to prune a state $\langle v, i, M \rangle$, we could open gaps from positions in J_t^M, while considering the additional gap open cost. When testing to prune a state $\langle v, i, I \rangle$ or $\langle v, i, D \rangle$, we could extend gaps from positions in J_t^I and J_t^D, respectively, without incurring the gap open cost. However, the downside of such an approach is the additional cost of inserting an increased number of positions into a B-tree, which has logarithmic time complexity.

Instead, POASTA employs another option: it tracks only reached (mis)match states in J_t^M, thus substantially reducing the number of times it needs to insert a position in a B-tree. POASTA can still use the positions in J_t^M to compute implicit gap costs and test whether to prune states $\langle v, i, I \rangle$ or $\langle v, i, D \rangle$. One thing to consider is that the latter states will not need to incur the gap open cost for extending the insertion or the deletion, while implicit gaps from positions in J_t^M do. Thus, POASTA will not prune a state $\langle v, i, I \rangle$ or $\langle v, i, D \rangle$, reached at cost C, if $\exists j \in J_t^M : C < C_{\langle t,j \rangle} + \Delta_o$.

Construction of benchmark datasets

To construct our bacterial gene benchmark datasets, we first downloaded all bacterial "complete" genomes from NCBI RefSeq (40,188 genomes total; accessed July 2023). We used the accompanying gene annotations to extract the *dnaG*, *nusA*, *pgk*, *pyrG*, and *rpoB* gene sequences from each genome.

To create each individual benchmark set, we clustered gene sequences using single-linkage hierarchical clustering, as implemented in SciPy [3]. Pairwise genetic distances were estimated using Mash [4] ($k = 15$; sketch size = 5,000), and were additionally used to deduplicate the sequence set, selecting one representative per set of identical sequences. We set the clustering threshold to 0.1, i.e., a new cluster would be formed if no neighbor could be found with a genetic distance < 0.1. This threshold is coarse enough to generate multiple genus and species-level clusters. We picked one or more clusters for each gene family as final datasets, each with at least 100 sequences, and varying the pairwise average nucleotide identities (ANI). Finally, each set was sorted by picking one "center" sequence with the smallest average Mash distance to all others and then ordering the remaining sequences in the set by the distance to the chosen "center" sequence, a strategy commonly applied before POA [5].

Benchmark execution details

We ran POASTA with the following parameters: mismatch cost $\Delta_x = 4$, gap open cost $\Delta_o = 6$, and gap extend cost $\Delta_e = 2$, the same costs as used in the Wavefront Algorithm (WFA) [6]. Our benchmark suite calls POASTA's Rust API directly to perform alignments. Thus, its runtime and memory usage measurements exclude anything related to startup or file input/output.

We wrote Rust bindings to SPOA and abPOA to achieve the same for those tools. All tools were configured to perform global alignment using the same cost model. Tools were run in single-threaded mode on a `c2-standard-8` virtual machine on the Google Cloud Platform, with an Intel Cascade Lake CPU and with 32 GB of RAM.

Assessing the frequency of missed optimal alignments with abPOA

To assess how frequently abPOA missed the optimal alignment, we constructed a graph comprising ten randomly selected gene sequences for each benchmark set. The remaining sequences were then aligned to the graph without updating it. This ensured each alignment was performed against the same graph and alignment scores were not influenced by different alignment backtracking or graph update choices. We performed alignments with SPOA, POASTA, and abPOA (each with the same graph as input) and recorded the alignment score for each non-graph sequence alignment.

We identified a discrepancy in abPOA's graph implemen-tation, which allowed alignments to start at any node that is the start of a previously added sequence to the graph. This enables the alignment to potentially skip nodes, whereas SPOA/POASTA would incur additional indel costs. It similarly allowed alignments to end at any node representing the end of a sequence added to the graph. This discrepancy resulted in

better (i.e., lower) alignment costs than expected since we benchmarked global alignment where indels at the start or end still incur an alignment cost.

Construction of *Mycobacterium tuberculosis* dataset

To construct the benchmark sets with *Mycobacterium tuberculosis* genomic sequences of 250, 500, and 1000 kbp in length, we downloaded all "complete" *M. tuberculosis* genomes available on NCBI RefSeq (370 total; accessed November 2023). To make all genomes colinear, we rotated and reoriented each genome such that each started with the gene *dnaA*, using the `fix-start` utility in Circlator [7]. Additionally, since inversions also break co-linearity, and POA poorly supports aligning large inversions, we excluded 29 genomes with more than 15% ($\geq$ 660 kbp) of its genome inverted with respect to the canonical reference *M. tuberculosis* H37Rv, detected using MUMMER [8].

We truncated genomes at specific genes to obtain sequences of the desired length. For the 250 kbp, 500 kbp, and 1 Mbp datasets, we used the genes *trmB*, *thiE*, and *gltA2* as cutoff points, respectively. We manually confirmed that these genes were located around the 250 kbp, 500 kbp, and 1 Mbp marks in each of the *dnaA* rotated and reoriented genomes. Finally, each dataset was sorted such that references were in ascending order of their Mash distance [4] to H37Rv.

POASTA was executed with the same alignment cost model as described above, but on the larger `c2-standard-60` virtual machine on the Google Cloud Platform, which has 240 GB of RAM available.

References

1. Lee C, Grasso C, and Sharlow MF. Multiple sequence alignment using partial order graphs. Bioinformatics 2002 Mar; 18:452–64
2. Gotoh O. An improved algorithm for matching biological sequences. en. Journal of Molecular Biology 1982 Dec; 162:705–8
3. Virtanen P et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. en. Nature Methods 2020 Mar; 17. Number: 3 Publisher: Nature Publishing Group:261–72
4. Ondov BD et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016 Jun; 17:132
5. Gao Y et al. abPOA: an SIMD-based C library for fast partial order alignment using adaptive band. Bioinformatics 2021 Aug; 37:2209–11
6. Marco-Sola S, Moure JC, Moreto M, and Espinosa A. Fast gap-affine pairwise alignment using the wavefront algorithm. Bioinformatics 2021 Feb; 37:456–63
7. Hunt M et al. Circlator: automated circularization of genome assemblies using long sequencing reads. Genome Biology 2015 Dec; 16:294
8. Marçais G et al. MUMmer4: A fast and versatile genome alignment system. PLoS Comput. Biol. 2018 Jan; 14:e1005944

B

Supplementary Materials - StrainGE: a toolkit to track and characterize low-abundance strains in complex microbial communities

B.1. Supplementary Tables

Table B.1: StrainGST performance using various sized databases.

Threshold (#refs)	TP	FN	FP	Recall	Precision	F1
0.9	2721	57	57	0.979	0.979	0.979
0.8 (213)	2640	119	122	0.957	0.956	0.956
0.7 (88)	2532	159	289	0.941	0.898	0.919
0.6 (42)	2284	336	694	0.872	0.767	0.816
0.5 (14)	2138	120	757	0.947	0.739	0.830

TP: True Positives; FN: False Negatives; FP: False Positives; Recall: TP / (TP+FN); Precision: (TP+FP) / TP; F1 score: 2 * (Recall * Precision) / (Recall + Precision).

Table B.2: Predicted relative abundances of strains in an *in vitro* mock community.

	StrainGST		
True Strain	**Reference**	**Sample**	**DB**
E. coli SEC470	*E. coli* SEC470	0.37%	50%
E. coli UTI89	*E. coli* UM146	0.20%	27%
E. coli Sakai	*E. coli* 149	0.09%	13%
E. coli 24377A	*E. coli* 24377A	0.07%	10%
	StrainEst		
True Strain	**Reference**	**Sample**	**DB**
E. coli SEC470	*E. coli* SEC470	n/a	48%
E. coli UTI89	*E. coli* UM146	n/a	27%
E. coli Sakai	*E. coli* 149	n/a	11%
E. coli 24377A	*E. coli* 24377A	n/a	5%
	E. coli APEC IMT5155	n/a	3%
	E. coli RM14721	n/a	1%
	BIB		
True Strain	**Reference**	**Sample**	**DB**
E. coli SEC470	*E. coli* K-12 substr. GM4792	n/a	15.27%
E. coli UTI89	*E. coli* H105	n/a	8.54%
E. coli Sakai	*E. coli* 108	n/a	7.57%
E. coli 24377A	*E. coli* S40	n/a	4.51%
	S. flexneri G1663	n/a	4.44%
	E. coli LHM10-1	n/a	2.37%
	E. coli $MSHS_1$33	n/a	2.28%
	S. dysenteriae 80-547	n/a	1.66%
	E. coli IMT16316	n/a	1.31%
	S. dysenteriae ATCC 12039	n/a	1.20%

True Strain: True strains present in the mock community. Reference: reference reported by corresponding tool; Sample: predicted relative abundance relative to the whole sample; DB: predicted relative abundance relative to other references in the database.

Table B.3: Species distribution of reported StrainGST references and bacterial isolates.

Species	**StrainGST**	**Isolates**
E. faecalis	64.50%	78.70%
E. faecium	8.40%	13.70%
E. avium	6.90%	0%
E. casseliflavus	6.60%	2.30%
E. durans	3.90%	3.60%
Other enterococci	9.70%	1.70%

Species: which *Enterococcus* species; StrainGST: Percentage of StrainGST reported references of a particular species across all metagenomes; Isolates: Percentage of isolates of a particular species as reported by Shao et al. [1].

Table B.4: Gut microbiome samples used as metagenomic background for ACNI and gap similarity benchmarking.

Accession	Kraken *E. coli*	Num. *E. coli*
SRS017821	0.00056	1
SRS017247	0.00029	1
SRS1041129	0.00083	0
SRS018606	8.00E-04	1
SRS047741	0.00071	1
SRS054590	0.00044	1
SRS064276	0.0092	1
SRS019910	0.00034	0
SRS044365	0.00525	1
SRS050299	1.00E-04	0

Accession: SRA accession; Kraken *E. coli*: *E. coli* relative abundance as predicted by Kraken2 [2]; Num. *E. coli*: Number of *E. coli* strains as predicted by StrainGST.

Table B.5: StrainGST cluster threshold optimization. A threshold of 0.9 performed best overall.

	Treshold	TP	FN	FP	Recall	Precision	F1
Single Strain ($n = 800$)	0.80	799	1	77	0.999	0.912	0.953
	0.85	793	7	23	0.991	0.972	0.981
	0.90	798	2	16	0.998	0.980	0.989
	0.95	790	10	26	0.988	0.968	0.978
Two Strains ($n = 1000$)	0.80	1816	138	111	0.929	0.942	0.936
	0.85	1867	102	60	0.948	0.969	0.958
	0.90	1895	83	51	0.958	0.974	0.966
	0.95	1881	101	84	0.949	0.957	0.953
Combined ($n = 1,800$)	0.80	2615	139	188	0.950	0.933	0.941
	0.85	2660	109	83	0.961	0.970	0.965
	0.90	2693	85	67	0.969	0.976	0.973
	0.95	2671	111	110	0.960	0.960	0.960

TP: True Positives; FN: False Negatives; FP: False Positives; Recall: TP / (TP+FN); Precision: (TP+FP) / TP; F1 score: 2 * (Recall * Precision) / (Recall + Precision).

B.2. Supplementary Figures

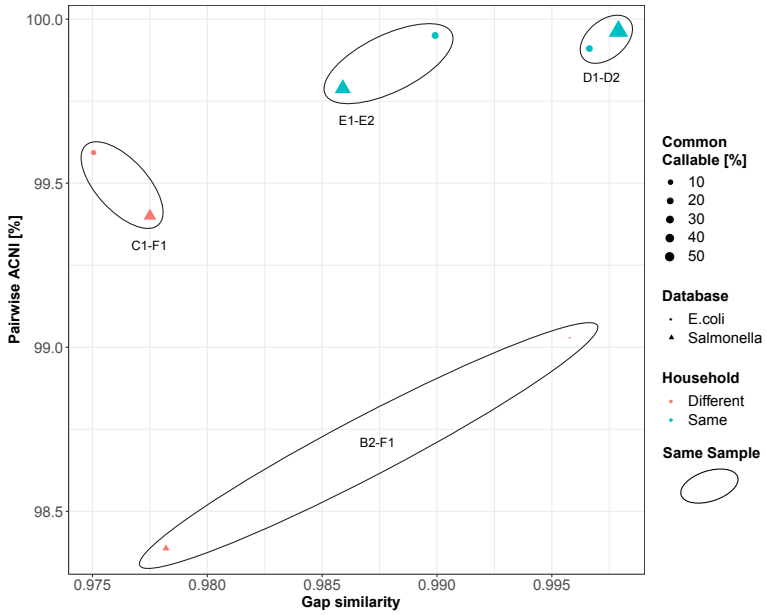

Figure B.1: StrainGE can robustly report on strain relationships even with more distant references. Pairwise ACNI and gap similarity as reported by StrainGE are plotted for the Kenyan household samples which share *E. coli* strains using the *E. coli* database (circle) or the contaminated *Salmonella* database (square). Using the "sparse", or contaminated, *Salmonella* database, StrainGE is still able to discern close strain relationships (same-household comparisons, teal) from more distant ones (different households, orange), almost as well as when the *E. coli* database is used. Point size reflects the common percent between the samples.

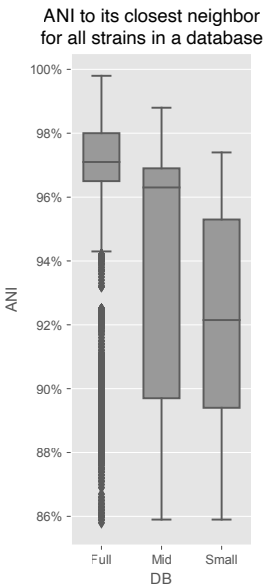

Figure B.2: Boxplot showing ANI between each strain in the reference database and its closest neighbor.

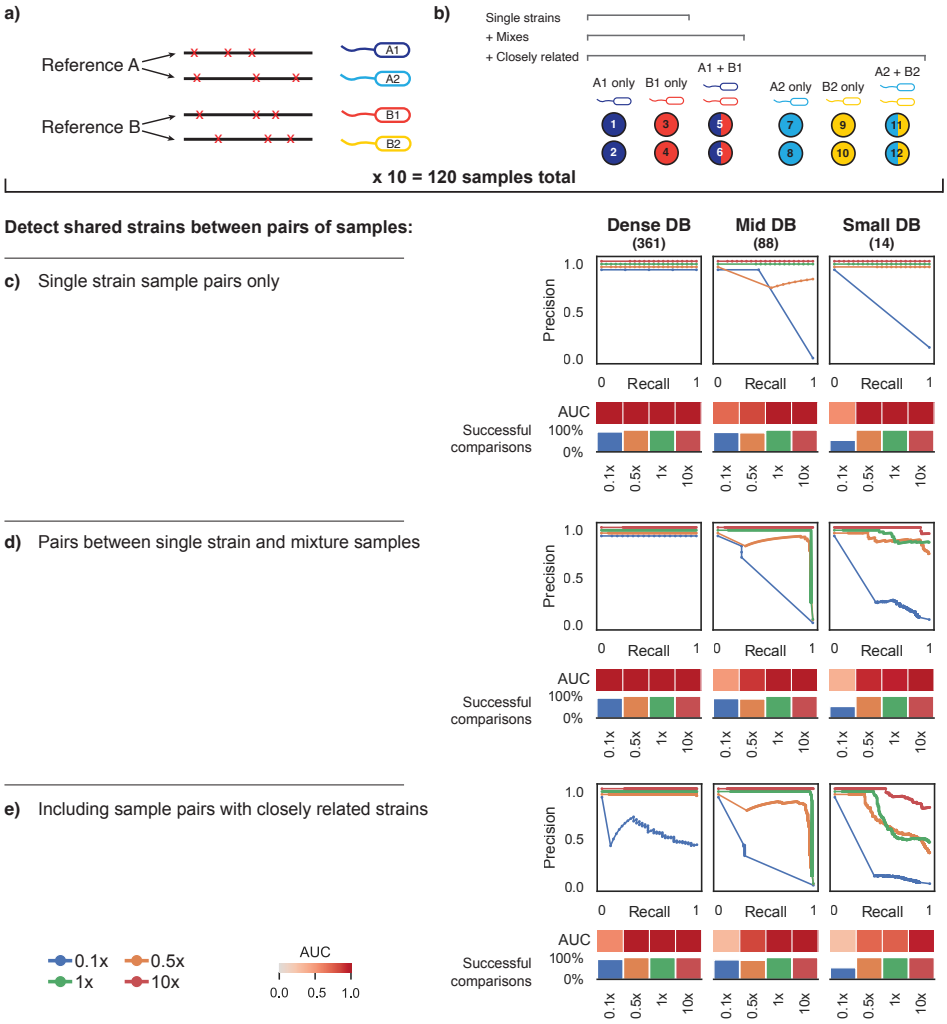

Figure B.3: StrainGE could accurately detect strain sharing even when using smaller databases. **(a)** Depiction of how synthetic *Escherichia* genomes were generated from randomly selected NCBI RefSeq genomes to create sets of closely related strains (e.g., A1/A2 and B1/B2) for spike in experiments. **(b)** Depiction of how spiked metagenomes were created using synthetic genomes from (a). Each circle represents a spiked metagenome. The color of the circle indicates which synthetic strain was included: single color circles indicate spiked metagenomes containing a single synthetic strain, and two color circles indicate spiked metagenomes containing two synthetic strains mixed at equal proportions. **(c-e)** Precision-recall curves for each tool and coverage 0.1x-10x, when given the task to detect which sample pairs contain identical strains. The area under the curve (AUC) is depicted as a heatmap below. The "successful comparisons" bar plot indicates the percentage of sample pairs for which a comparison was possible (i.e., tools ran to completion for both samples). **(c)** Limiting to single-strain samples from distinct references. **(d)** Including samples with two strains, but limited to strains from distinct references. **(e)** Including samples with closely related strains.

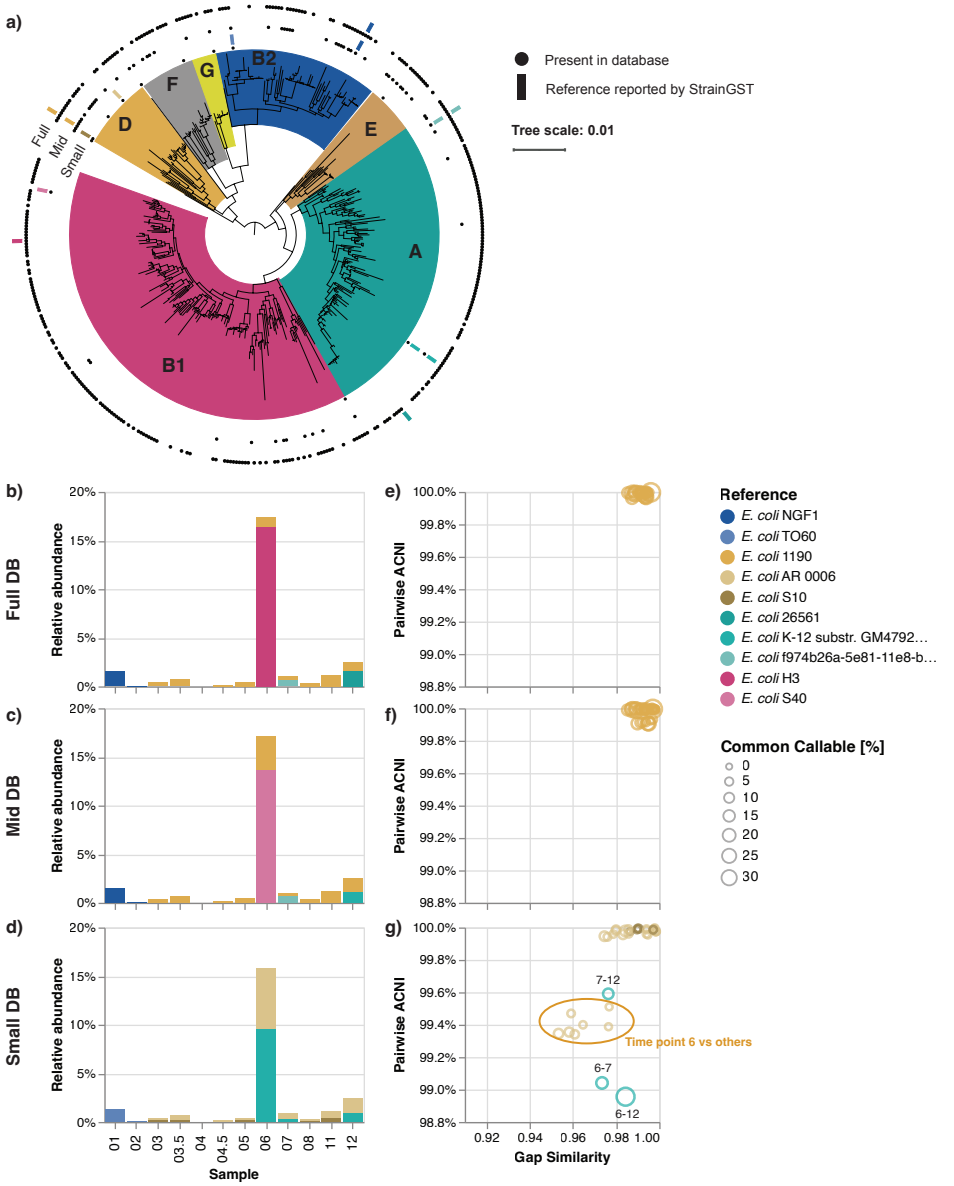

Figure B.4: StrainGE provides useful information using both small and large databases. **(a)** Single-copy core phylogenetic tree of 471 *E. coli* genomes (including the full set of genomes contained in the large reference database) with major phylogroups annotated. Black dots indicate presence of the corresponding genome in the small (inner ring), mid and full database (outer ring). Colored rectangles indicate the genomes reported by StrainGST when using the small, mid or full database. StrainGST reported references and their estimated relative abundances per time point when using the **(b)** full database, **(c)** mid database **(d)** small database. For strains matching the same reference, pairwise ACNI and gap similarities are plotted when using the **(e)** small database, **(f)** mid database and **(g)** full database.

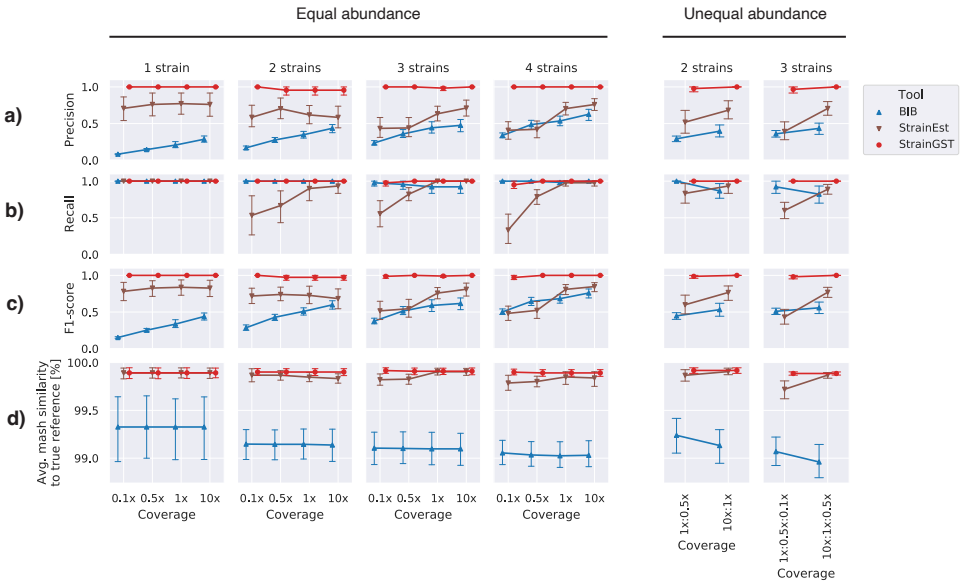

Figure B.5: StrainGST was more sensitive and precise in identifying close reference genomes than other tools. Performance of StrainGST (red circles), StrainEst (brown triangles) and BIB (blue triangles) on 15 sets of metagenomes spiked with known *Escherichia* strains mixed at either equal abundance (left panel; 1-4 strains for each sample, 0.1x-10x coverage) or unequal abundance (right panel; 2 strains mixed at 1x:0.5x and at 10x:1x, or 3 strains mixed 1x:0.5x:0.1x and 10x:1x:0.5x). Performance plotted as **(a)** Precision, **(b)** Recall, **(c)** F1 score, **(d)** Average Mash similarity to closest reference.

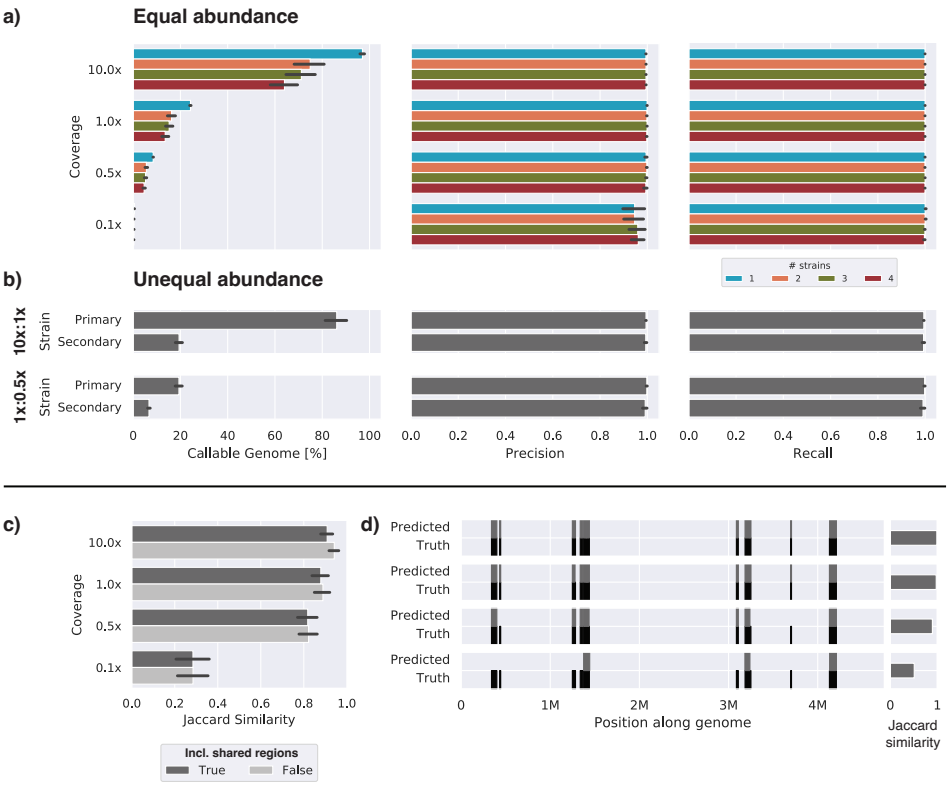

Figure B.6: StrainGR accurately called SNVs and deletions in single strain and mixed samples at low coverages. **(a)** Percent callable genome, precision, and recall for SNVs called by StrainGR on mixtures of 1-4 synthetic genomes spiked into a metagenomic sample at different coverages. The "% callable genome" refers to the fraction of the genome where StrainGR was able to make calls. **(b)** Percent callable genome, precision, and recall for SNVs called by StrainGR (limited to callable genome) on pairs of *Escherichia* strains mixed at unequal abundance (1x:0.5x or 10x:1x). **(c)** Jaccard similarity between gaps predicted by StrainGR and known gaps, at different coverages. Dark grey bars indicate the Jaccard similarity when using the whole genome; light grey indicates the Jaccard similarity when ignoring positions with a majority of multi-mapped reads. **(d)** An example of the pattern of deletions present within a synthetic genome ("truth"; black), compared to the pattern of deletions predicted by StrainGR ("Predictions"; grey).

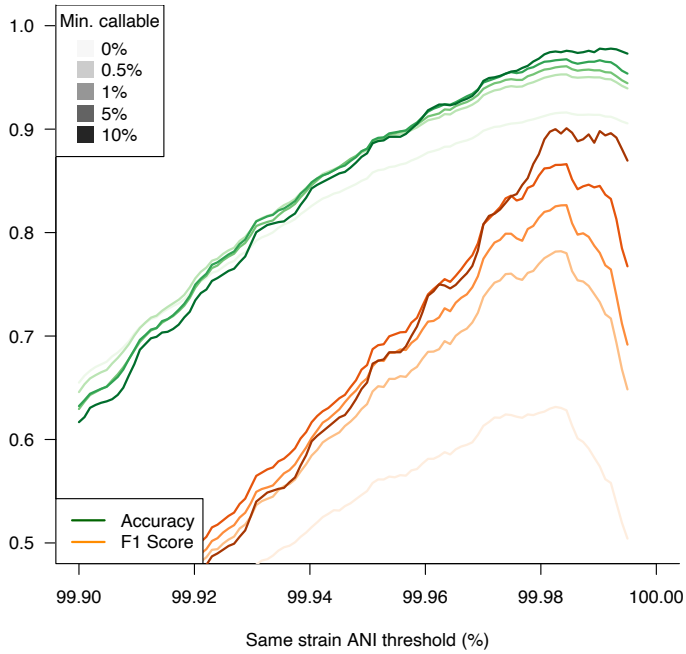

Figure B.7: StrainGR metrics can be used to accurately classify strain sharing in distinct metagenomic backgrounds. Using metagenomic samples spiked with *E. coli* isolates containing *in silico* introduced SNVs, we used different values for StrainGR's ACNI metric to classify sample pairs as containing the same, or different, strains. All pairs with ACNI above the threshold were considered 'shared' between samples. Pairs were considered correctly classified if the true ANI was 100%. Accuracy (green) and F1 score (orange) were calculated for a range of ACNI thresholds, additionally filtering for comparisons with a minimum amount of common callable genome (light to darker lines). StrainGR's ability to correctly delineate identical strain pairs increased with a larger common callable genome, with a substantial drop in accuracy with common callable genome <0.5%.

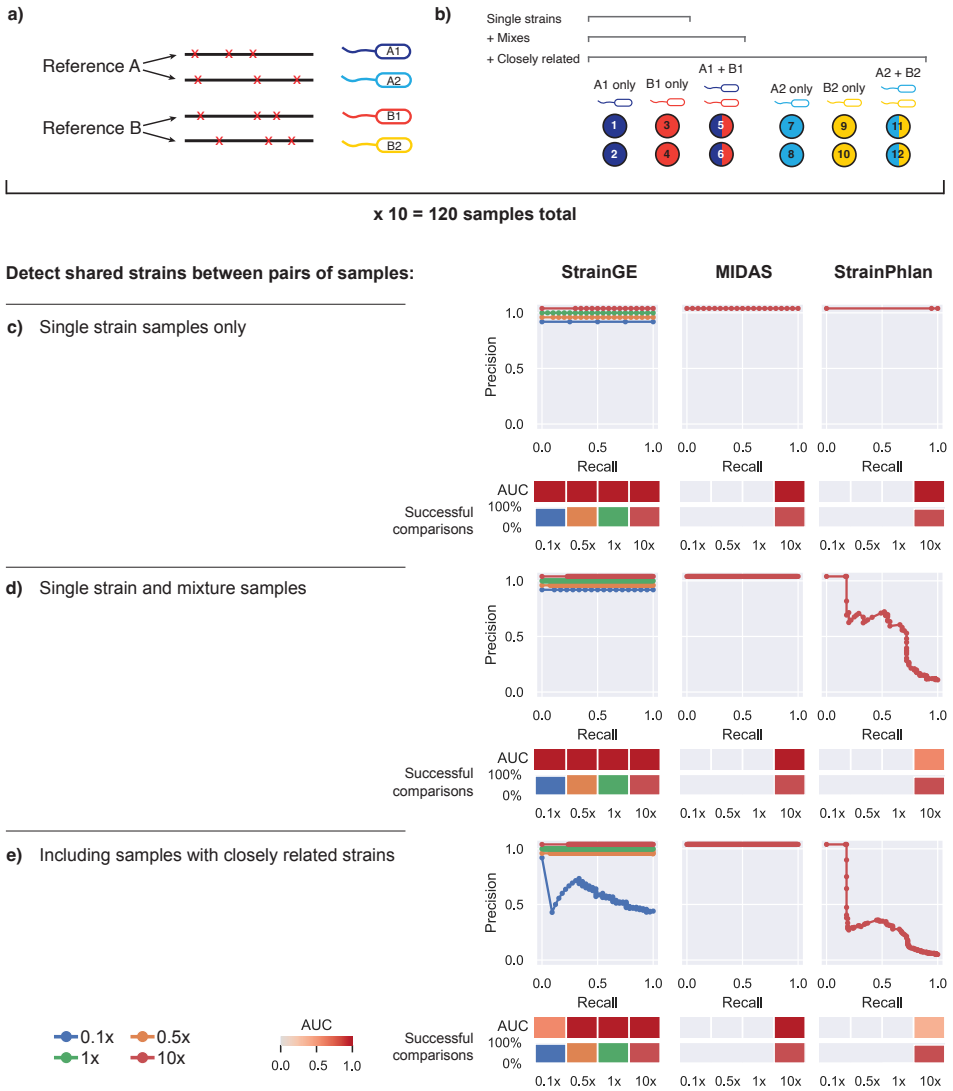

Figure B.8: StrainPhlan and MIDAS did not run to completion at coverages <10x with their default settings. **(a)** Depiction of how synthetic *Escherichia* genomes were generated from randomly selected NCBI RefSeq genomes to create sets of closely related strains (e.g., A1/A2 and B1/B2) for spike in experiments. **(b)** Depiction of how spiked metagenomes were created using synthetic genomes from (a). Each circle represents a spiked metagenome. The color of the circle indicates which synthetic strain was included: single color circles indicate spiked metagenomes containing a single synthetic strain, and two color circles indicate spiked metagenomes containing two synthetic strains mixed at equal proportions. **(c-e)** Precision-recall curves for each tool and coverage 0.1x-10x, when given the task to detect which sample pairs contain identical strains. The area under the curve (AUC) is depicted as a heatmap below. The "successful comparisons" bar plot indicates the percentage of sample pairs for which a comparison was possible (i.e., tools ran to completion for both samples). **(c)** Limiting to single-strain samples from distinct references. **(d)** Including samples with two strains, but limited to strains from distinct references. **(e)** Including samples with closely related strains.

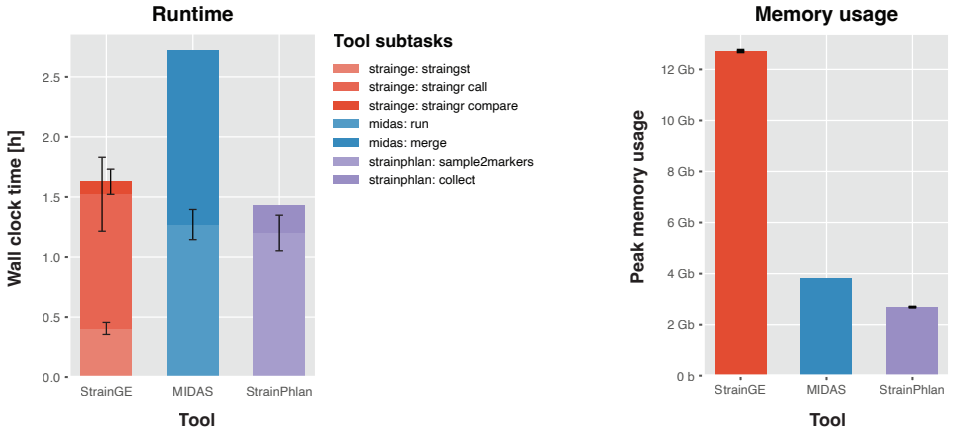

Figure B.9: StrainGE has comparable runtime to other tools and has memory usage within limits of powerful PCs. We tracked runtime and peak memory usage for StrainGE, MIDAS and StrainPhlan when run on samples from our strain sharing benchmarks (Figure 3.3). Tool subtasks are given different shades of color. **(a)** Wall clock runtime for each tool. **(b)** Peak memory usage for each tool.

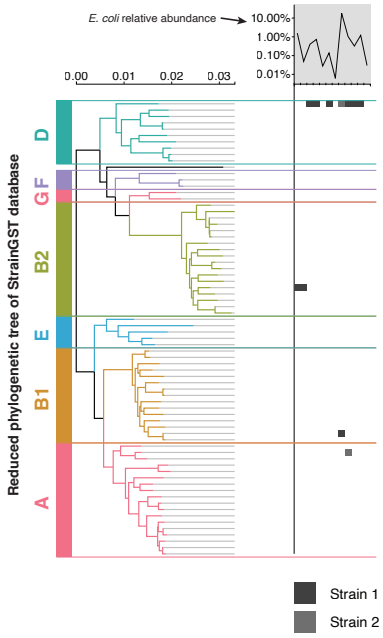

Figure B.10: Strains detected by StrainGE could easily be placed in phylogenetic context. Left panel: reduced phylogenetic tree of the *Escherichia* StrainGST database, with clade designations as obtained using ClermonTyping. Right panel: strains identified from longitudinally collected stool samples from a single individual. Each column represents a time point, and a square within a column indicates the StrainGST match(es) in that sample. Top: overall *E. coli* relative abundance over time.

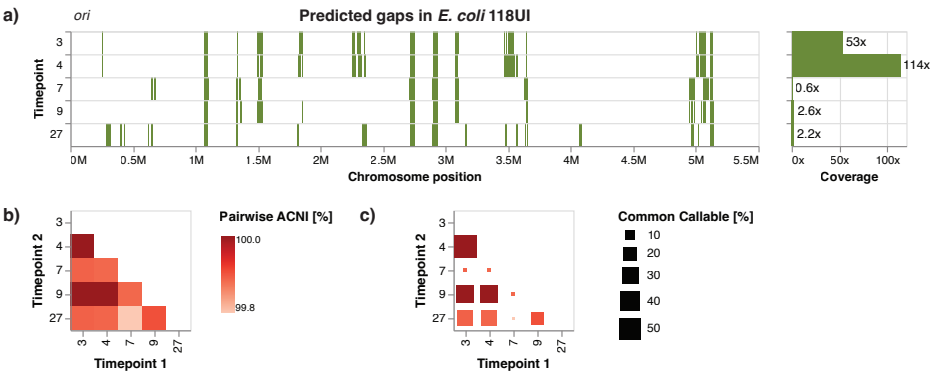

Figure B.11: StrainGR provides detailed insights in the genomic diversity of strains close to *E. coli* 118UI. **(a)** Predicted large deletions in the reference *E. coli* 118UI at multiple time points. Each green rectangle represents a large deletion. **(b)** Pairwise ACNI between strains at different timepoints. **(c)** The same heatmap as in (b), but each square is scaled by the percentage of the common callable genome.

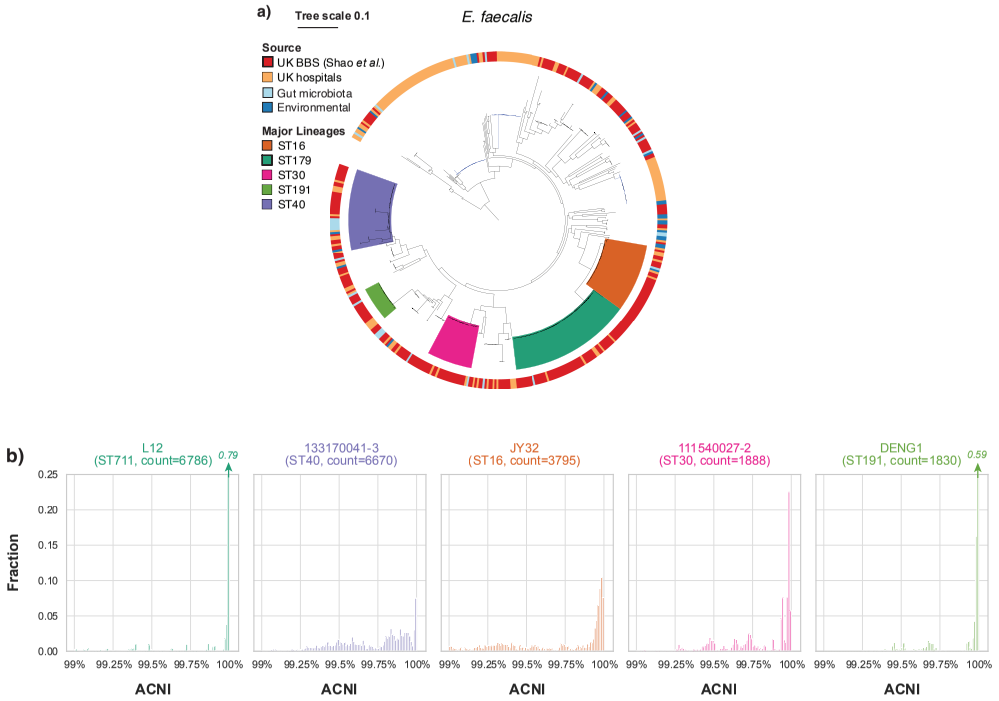

Figure B.12: StrainGE's pairwise ACNI distributions mirror tree topology. **(a)** Single-copy core phylogenetic tree of *E. faecalis* isolates from the UK Baby Biome Study (UK BBS) ($n = 282$) in the context of isolates from other public UK hospitals ($n = 168$), human gut microbiota ($n = 28$), or other environmental sources ($n = 27$). Five major lineages were identified, represented by ST16, ST179, ST30, ST191 and ST40. Tree republished with permission from Shao et al. [1]. **(b)** Pairwise ACNI distribution of strains mapping to references representing the five major lineages identified by Shao et al. ST179 was not represented by any reference in our database, but the closest match was ST711. The tree topologies of some lineages had more consistent short branching (e.g., ST191) than others (e.g., ST40), indicating clear patterns in relationships among close isolates. The StrainGR-computed ACNI values largely mirrored these results in how narrow (e.g., ST191) or wide (e.g., ST40) the ACNI distributions were for comparisons across predicted members of these lineages.

B.3. Supplementary Text

Supplementary Results

StrainGE can robustly report on strain relationships with sparse databases

I. We observed that StrainGE was able to produce reliable results about strain relationships, even when the detected reference strain turned out to be a more distant contaminant in our sequence repository; i.e., from a genetically sparse region of the database. Using a *Salmonella* database, we applied StrainGE to 14 stool microbiome samples obtained from multiple pairs of cohabiting Kenyan siblings, predicted by Kraken2 [2] to contain low levels of *Salmonella* (Supplementary Methods). StrainGST reported that 12 of the 14 samples harbored a match to the same *Salmonella* strain, *Salmonella* sp. HNK130, suggesting that all sibling and non-sibling pairs were colonized with highly similar *Salmonella*, despite households being geographically separated.

Closer examination of StrainGR output revealed, however, that each predicted HNK130-like strain had a different ACNI score relative to the reference, ranging from 98.2% to 99.7%, which were all lower than expected. Subsequent ANI and BLAST analyses (Supplementary Methods) revealed that HNK130 was actually *E. coli*. Recent work has shown that reference genome databases used by Kraken contain cross-species plasmids [3], like those shared by *E. coli* and *Salmonella*, which can lead the tool to incorrectly assign species. To avoid this problem, StrainGST does not include plasmid content in its reference database, and the tool now flags the user when a reference shares less than 90% ANI to other references.

After removing outliers from the *Salmonella* database, we reran the Kenyan samples through StrainGE, which predicted that there were no *Salmonella* in these microbiomes. Remarkably, the output from running StrainGE on the Kenyan samples using our *E. coli* database provided very similar views of the relative closeness of *E. coli* strains in cohabiting siblings as observed in our original run using the *E. coli* contaminated *Salmonella* database (Figure B.1). This suggested that StrainGE can return accurate and consistent information about strain relationships, even when the best matching strain is from a low density area of the reference database, which could occur for species of interest that are under-represented in genome repositories.

II. While the above results suggested that our current default threshold for building the StrainGST database could be looser to make the database more sparse, we predicted that there would be trade-offs including potentially losing the ability to resolve more closely related strains and reducing the amount of genome that could be analyzed, given the known relationship between ANI and gene content similarity for many species i.e., more distantly related isolates tend to share fewer genes [4].

In order to formally test this, we first created *Escherichia* databases with increasingly fewer references by clustering the downloaded *Escherichia* references at Jaccard similarity thresholds of 0.8, 0.7, 0.6 and 0.5 (estimated ANI 98.2%-99.5%), resulting in databases with 213, 88, 42, and 14 *Escherichia* references, respectively. Then, to as-

sess the impact of having fewer references on StrainGST's accuracy, we benchmarked these sparser databases against the original denser database (361 genomes; Jaccard similarity threshold of 0.9) using the same set of 1,800 samples used to optimize the clustering threshold for *E. coli*, each containing 1-2 strains (Materials & Methods). Recall remained consistently high, indicating that StrainGST was able to pick the closest references in the sparser databases. The overall F1-score, however, decreased for smaller databases, driven by an increased number of false positives (Table B.1). As the closest reference could be quite distant from the sample strain, it was often unable to explain significant portions of the sample genome, leaving sufficient k-mers remaining for StrainGST to report an additional false positive reference.

To investigate how database sparseness impacted tracking strains across samples, we repeated the *in silico* strain tracking tests (Figure 3.3) using StrainGR with two smaller databases (Figure B.2), clustered at thresholds of 0.5 (very sparse; 14 *Escherichia* genomes) and 0.7 (intermediate; 88 *Escherichia* genomes). As compared to the original database of 361 references, the intermediate database performed comparably for all tests at coverages of 1x and higher (Figure B.3c,e,d); however, at low coverages (0.5x), a single sample set where StrainGST made incorrect reference calls led to a lower area under the precision-recall curve (AUC) across all tests. For the very sparse database, StrainGE was able to correctly detect shared strains across single strain samples (Figure B.3c), as well as mixes (Figure B.3d), at strain coverages of at least 0.5x; however, at low coverages (0.1x) its performance dropped considerably. In some low-coverage cases, StrainGE was not able to run to completion due to the default minimum coverage requirements not being met (as indicated by the lower "successful comparisons"; Figure B.3c,d,e). In cases where it did complete, the scant read data aligned less accurately to the more distant reference, which could share as little as ~ 98% ANI with the sample strain. In addition, this very sparse database also performed worse than denser databases in distinguishing between closely related strains across all coverages (Figure B.3e), likely because less accurate alignments resulted in lower and less accurate ACNI values.

To examine how the sparser databases would affect results from running StrainGE on real data, we reran the pipeline on the same metagenomic dataset as in Figure 3.5 (woman with recurrent urinary tract infection) using the intermediate and very sparse databases and compared results to those using the original database. StrainGST results were similar across all databases, with nearly identical read-outs of overall *Escherichia* relative abundance, and reported references mostly from the same phylogroups at similar relative abundances (Figure B.4b,c,d; phylogenetic distribution of reported references shown in Figure B.4a). We only observed discordant results with the very sparse database: a strain originally represented by a reference from phylogroup B1 was now represented by a reference from phylogroup A (time point 6); and a phylogroup D strain close to *E. coli* 1190 in the dense database was represented by two phylogroup D references (time points 3-5, 8, 11).

Despite some StrainGST-level differences across database runs, StrainGR was still able to distinguish between "same" and "different" strains that hit the same reference. For example, StrainGR pairwise comparisons of a phylogroup D strain identified by all three (dense, intermediate and very sparse) databases consistently revealed high

ACNI and gap similarity, indicating that the strain was the same across samples (Figure B.4e,f,g). In contrast, for a pair of strains originally mapping to different references with the dense and intermediate databases, but mapping to the same reference with the very sparse database, StrainGR correctly identified that these two strains were not the same, as suggested by the lowered ACNI (Figure B.4g; time points 6-7).

Though "same" versus "different" strain assignments were generally consistent across databases, we observed a notable difference in ACNI estimates using the very sparse database for comparisons involving time point 6, a sample with relatively high abundance of *E. coli* and consistently predicted to carry two strains (Figure B.4g). While we can not confirm why ACNI estimates differed so dramatically for the very sparse database, we hypothesize that the reported references in the medium and full database were better representations of the strains in the sample, able to attract reads to the correct locations which improved deconvolution of a strain mixture, resulting in more accurate ACNI values.

In conclusion, while StrainGE can provide useful information even with a small database (still able to pick the closest references), the accuracy of ACNI improved as the database size got larger. Thus, when using a very sparse database and in case of a strain mixture, we encourage users to be more careful interpreting ACNI, and use any available longitudinal information to confirm the presence of a "same" or "different" strain.

StrainGST works at lower coverages and pinpoints more closely related references than other tools.

In order to assess the sensitivity and specificity of StrainGST compared to similar tools, we constructed *in silico* metagenomes that were spiked with sequences of known strains of *Escherichia* at varying relative abundances. We simulated reads from randomly selected *Escherichia* genomes downloaded from RefSeq, approximately one third of which were also represented in our *Escherichia* reference database, and mixed them with reads from a metagenomic sample from the Human Microbiome Project without any detectable *Escherichia*, to a total of 3 Gb per sample (Materials & Methods). Strains were mixed at both equal (1-4 strains) and unequal (2-3 strains) abundances to achieve between 0.1x and 10x depth of *Escherichia* coverage (roughly 0.02% to 1.6% relative abundance) per sample per strain, designed to cover the typical ranges of complexity and abundance of *E. coli* within metagenomic stool samples. A total of 240 spiked metagenomes were generated with strains mixed at equal abundance, and another 30 with strains mixed at unequal abundance.

We compared StrainGST to two similar tools, which also identify strains in a sample based on those in a reference database: BIB [5] and StrainEst [6]. BIB applies a Bayesian model to sample reads aligned to a core alignment of its database in order to identify the closest strain(s). StrainEst applies a regression model based on unique SNVs in the genomes of strains represented in its database to identify the closest strain(s) in a sample. All tools were run on each spiked metagenome sample, and the fidelity of the results were determined by comparing the reports from each tool against the known composition of the spiked metagenomes. We excluded DiTASiC [7] because the program halts when fewer than 75% of reads can be assigned to reference,

which makes it much less flexible than StrainGE for characterizing low-abundance species, and excluded PathoScope [8] and Sigma [9], because we were unable to run them to completion because of their dependencies on outdated databases or software (Materials & Methods). Because BIB's database construction step, which required generation of a core alignment using progressiveMauve [10], could not scale to include all 361 reference genomes used for benchmarking StrainGST and StrainEst, we also created a smaller database for BIB with only 20 reference genomes.

StrainGST performed as well as, or better than, the other tools across all scenarios tested, and stood out strongly when strains were at very low abundance, either alone or as part of a mixture with other strains. StrainGST had the highest precision (mean 0.99), F1 score (mean 0.99), and its recall and average Mash similarity [11] were at least as good as that of the other tools when given the task of identifying the closest reference(s) to those present in the spiked metagenome (Figure B.5). There was no significant correlation between StrainGST's F1 scores and the number of strains with an exact reference match in the database (Spearman's $\rho = 0.06$, p-value=0.25) suggesting that StrainGST performance was not dependent upon exact matches to references in the database. Although StrainEst was tested using the same reference database as StrainGE, StrainEst often reported a strain different from the true closest strain in the database or none at all, thus lowering both its precision and recall. However, in these cases, StrainEst still selected a strain with relatively high similarity, as reflected in its high Mash similarities. In contrast, while BIB often picked the closest strain in its database (mean recall 0.94), the selected reference was often a poor proxy given BIB's smaller database, resulting in much lower Mash similarities.

The high performance of StrainGST was especially striking at lower coverages. While StrainGST consistently performed well across all coverages and mixtures, both StrainEst and BIB performed poorly at coverages <1x. StrainGST was the only tool able to recover mixtures of strains present at a 20-fold coverage difference, as reflected by a mean precision of 0.98 and mean recall of 1.0 in the 10x:1x:0.5x benchmark (Figure B.5). These results highlight the wide dynamic range over which StrainGST was able to correctly identify the closest strains within mixtures, including for the abundance range typically seen for key organisms such as *E. coli* in the human gut.

StrainGR accurately identifies SNVs at low coverages.

StrainGR is unique in its ability to call SNVs across the close reference genomes of strain(s) identified by StrainGST using metagenomic data. Although other tools can identify nucleotide-level differences across sets of samples, they are limited to either marker gene sets, or a single reference, which may be quite distant. To further characterize the ability of StrainGE to call SNVs within low-abundance strains in a metagenomic sample, we introduced random SNVs into sets of *Escherichia* genomes at equal abundance, or unequal abundance. We mixed simulated reads from these strains into a real metagenomic sample, and compared the SNVs called by StrainGR to the known SNVs (Figure B.6a-b; Supplementary Methods). StrainGR achieved near perfect precision and recall at identifying true SNVs (>0.95 for coverages 0.5x and above). However, the fraction of the genome where StrainGR was able to make calls (the "% callable genome") was reduced when coverage decreased, or when multiple strains

were present, due to there being a greater fraction of shared genome content between references decreasing the unique regions that StrainGE can use for SNV calling. We observed no clear reduction in precision or recall for mixes, either at equal or unequal abundances, highlighting the ability of StrainGR to effectively disentangle SNVs from different strains.

StrainGR accurately identifies large deletions at low coverages

Because of frequent recombination and horizontal gene transfer in bacteria, patterns of large deletions (gaps) provide an orthogonal line of evidence for strain similarity [12]. StrainGR is unique in its ability to call large deletions relative to close reference genomes. In order to benchmark this ability, we introduced random deletions of 5-100kb into *Escherichia* genomes, and mixed simulated reads from these strains into a real metagenomic sample. We then compared the deletions predicted by StrainGR to the known deletions by computing the Jaccard similarity (Supplementary Methods). StrainGR's large deletion predictions closely matched the true deletions, with a Jaccard similarity of approximately 0.8 for coverages 0.5x and higher (Figure B.6c), and high concordance when examining genome-wide patterns of deletions (example in Figure B.6d). Multi-mapping reads (due to repeats in the reference genome) reduced the accuracy of calling deletions, as multi-mapping reads that map to a region of the reference that is present, as well as all or part of a deleted region, will not be properly marked as a deletion. When ignoring positions with a majority of multi-mapped reads, the concordance between predicted and true deletions was even higher, reaching a Jaccard similarity score of 0.9 at 10x coverage (Figure B.6c). The pattern of deletions shared across strains in a dataset should be consistent across all samples to be compared and may reflect evolutionary history, thus providing another key indicator of strain relatedness.

Supplementary Methods

StrainGST benchmarking. We compared the ability of StrainGST to select the closest strain in an *Escherichia* reference database to BIB [5] and StrainEst [6]. We excluded PathoScope [8] because its database construction process required taxonomy IDs in BLAST's NT database, which have been phased out by NCBI. Sigma was excluded because we could not run the pipeline end-to-end, as we were unable to run steps that depended on MPI for compute parallelism. Where possible, we used the same database to ensure fair comparison. For StrainGST and StrainEst, we used the same 361-strain database. For BIB, since BIB's database construction process did not scale, we generated a smaller database containing 20 representative genomes. To select the 20 representatives, we computed pairwise Mash distances [11] between all 929 genomes used as input into the StrainGST *Escherichia* database and performed hierarchical clustering to obtain 20 clusters. The genome from each cluster with the lowest average distance to all other genomes in its cluster was selected.

We took into consideration StrainEst and BIB calls that reported a strain at >1% abundance relative to other strains in the database, the same threshold used in Albanese and Donati [6]. We used pairwise Mash distances to assess how close a re-

ported strain was to the true strain, counting a reported strain as a true positive if it was the closest strain in the database to the strain in the sample. Any other reported strain that was not present in the sample was counted as a false positive. If any of the strains present in the sample were not reported by the tool, it was counted as a false negative.

We ran each tool on 240 spiked metagenomes with 1-4 strains mixed at equal abundance, with average coverage of 0.1x, 0.5x, 1x or 10x; 40 spiked metagenomes with two strains mixed at 10x:1x or 1x:0.1x; and 40 spiked metagenomes with three strains spiked at 10x:1x:0.5x or 1x:0.5x:0.1x. Strains for each sample were randomly selected from NCBI RefSeq and metagenomes were generated as described in the main text Materials & Methods.

Application of StrainGE using a *Salmonella* database

We constructed a StrainGST database from 877 genomes identified as *Salmonella* in NCBI RefSeq. 177 genomes were retained after database clustering using default settings (clustering genomes with ANI higher than approximately 99.8% ANI to another reference in the database, and keeping a single representative from each cluster). We ran StrainGST with this final database, using default settings.

ANI comparisons between HNK130 and the other members of the *Salmonella* reference database were approximated using Mash-based k-mer similarity metrics available in StrainGE. BLAST results for HNK130 revealed close hits to *E. coli* rather than *Salmonella*. ANI comparisons between the HNK130 genome and *E. coli* genomes were performed using Chunlab's ANI calculator tool (https://www.ezbiocloud.net/tools/ani) [13].

As a positive control test set to verify that StrainGE works on *Salmonella*, we ran previously published metagenomic datasets where *Salmonella* content was proven [14] against our cleaned *Salmonella* database where the contaminating *E. coli* genomes had been removed.

Benchmarking of StrainGR SNV calls using simulated data

To benchmark the ability of StrainGR to call SNVs, we introduced random SNVs into randomly drawn genomes from the NCBI RefSeq complete database, such that the average nucleotide identity to the original reference was 99.9% (approximately 5,000 SNPs). We generated synthetic reads from these genomes and spiked them into a metagenomic sample with no *E. coli* as for other benchmarks (Materials and Methods). We constructed a total of 320 synthetic communities with spiked-in strains at equal abundance, including i) 20 sets for each number of strains per sample (1-4 strains); and ii) 20 sets at each coverage (0.1x, 0.5x, 1x and 10x, corresponding to relative abundances of approximately 0.02x - 1.6x). We also created 20 two-strain communities at 10x:1x and another 20 at 1x:0.5x coverage.

Using these simulated metagenomic samples, we used StrainGR to investigate whether we could correctly identify the synthetically introduced SNPs or deletions, even when mixed within a metagenomic background. For each sample, we prepared a concatenated reference containing the original references used to generate the

benchmark sample. We aligned the reads to its concatenated reference, and ran StrainGR to call SNVs. The SNV calls made by StrainGR were compared to the truth (i.e. the known set of mutations introduced into that synthetic strain) using Illumina's som.py (https://github.com/Illumina/hap.py) and each call was classified as either a true positive (TP), false positive (FP), or false negative (FN).

Benchmarking of StrainGR large deletion predictions using simulated data

To benchmark the accuracy of StrainGR in predicting large deletions, we created a separate set of 80 synthetic samples based on 20 randomly selected *E. coli* genomes present in the NCBI RefSeq complete database, in which we deleted random blocks of genes (sized 5-100kb), resulting in loss of approximately 7.5% of the total genes in each reference genome. From these synthetic samples, we simulated reads using ART [15] at fixed coverages of 0.1x, 0.5x, 1x and 10x, and mixed the simulated reads with reads subsampled from a real metagenomic sample, as for the SNV benchmarks described above. Reads were aligned to the original reference, and large deletions predicted by StrainGR were compared to true deletions using the Jaccard similarity metric:

$$\text{Jaccard} = \frac{|G_s \cap G_t|}{|G_s \cup G_t|} \tag{B.1}$$

G_s is the set of positions in the genome where StrainGR predicted a large deletion, and G_t is the actual, known set of positions for large deletions.

References

1. Shao Y et al. Stunted microbiota and opportunistic pathogen colonization in caesarean-section birth. en. Nature 2019 Oct; 574. Number: 7776 Publisher: Nature Publishing Group:117–21
2. Wood DE, Lu J, and Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biology 2019 Nov; 20:257
3. Doster E et al. A Cautionary Report for Pathogen Identification Using Shotgun Metagenomics; A Comparison to Aerobic Culture and Polymerase Chain Reaction for Salmonella enterica Identification. eng. Frontiers in Microbiology 2019; 10:2499
4. Konstantinidis KT, Ramette A, and Tiedje JM. The bacterial species definition in the genomic era. Philosophical Transactions of the Royal Society B: Biological Sciences 2006 Nov; 361:1929–40
5. Sankar A et al. Bayesian identification of bacterial strains from sequencing data. Microb Genom 2016 Aug; 2:e000075
6. Albanese D and Donati C. Strain profiling and epidemiology of bacterial species from metagenomic sequencing. Nat. Commun. 2017 Dec; 8:2260
7. Fischer M, Strauch B, and Renard BY. Abundance estimation and differential testing on strain level in metagenomics data. Bioinformatics 2017 Jul; 33:i124–i132
8. Hong C et al. PathoScope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples. Microbiome 2014 Sep; 2:33
9. Ahn TH, Chai J, and Pan C. Sigma: Strain-level inference of genomes from metagenomic analysis for biosurveillance. Bioinformatics 2015; 31:170–7
10. Darling AE, Mau B, and Perna NT. progressiveMauve: Multiple Genome Alignment with Gene Gain, Loss and Rearrangement. en. PLOS ONE 2010 Jun; 5. Publisher: Public Library of Science:e11147

11. Ondov BD et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016 Jun; 17:132

12. Darmon E and Leach DRF. Bacterial Genome Instability. en. Microbiology and Molecular Biology Reviews 2014 Mar; 78. Publisher: American Society for Microbiology Section: Review:1–39

13. Yoon SH et al. Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies. eng. International Journal of Systematic and Evolutionary Microbiology 2017 May; 67:1613–7

14. Huang AD et al. Metagenomics of Two Severe Foodborne Outbreaks Provides Diagnostic Signatures and Signs of Coinfection Not Attainable by Traditional Methods. Applied and Environmental Microbiology 2017 Jan; 83

15. Huang W, Li L, Myers JR, and Marth GT. ART: A next-generation sequencing read simulator. Bioinformatics 2012; 28:593–4

C

Supplemental Materials - Longitudinal multi-omics analyses link gut microbiome dysbiosis with recurrent urinary tract infections in women

C.1. Extended Data Figures

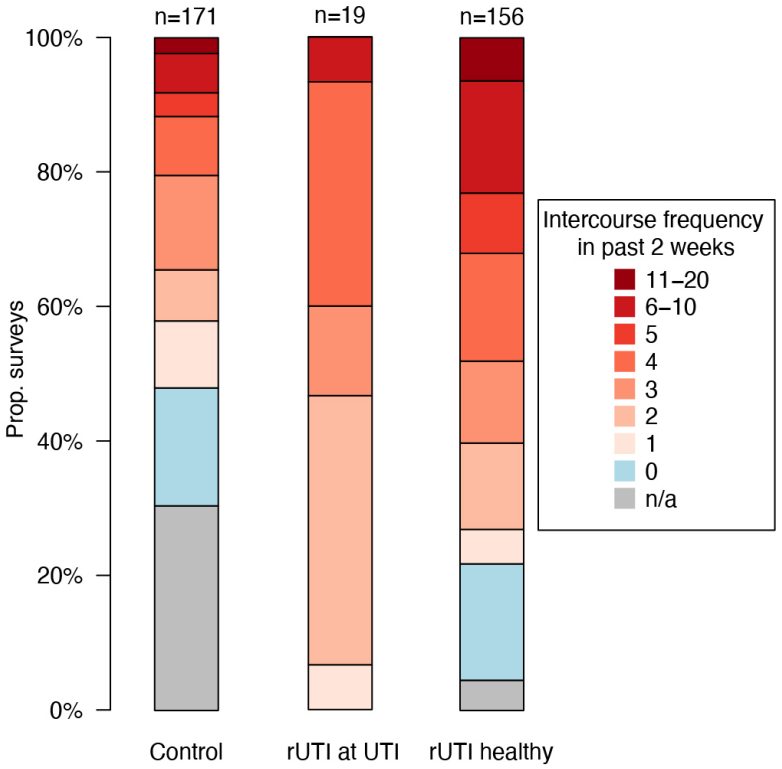

Figure C.1: Sex precedes all clinical UTI events. Survey reports of intercourse frequency in the previous two weeks. Responses are partitioned by (i) control women, (ii) rUTI women at time of UTI, and (iii) rUTI women at non-UTI time points.

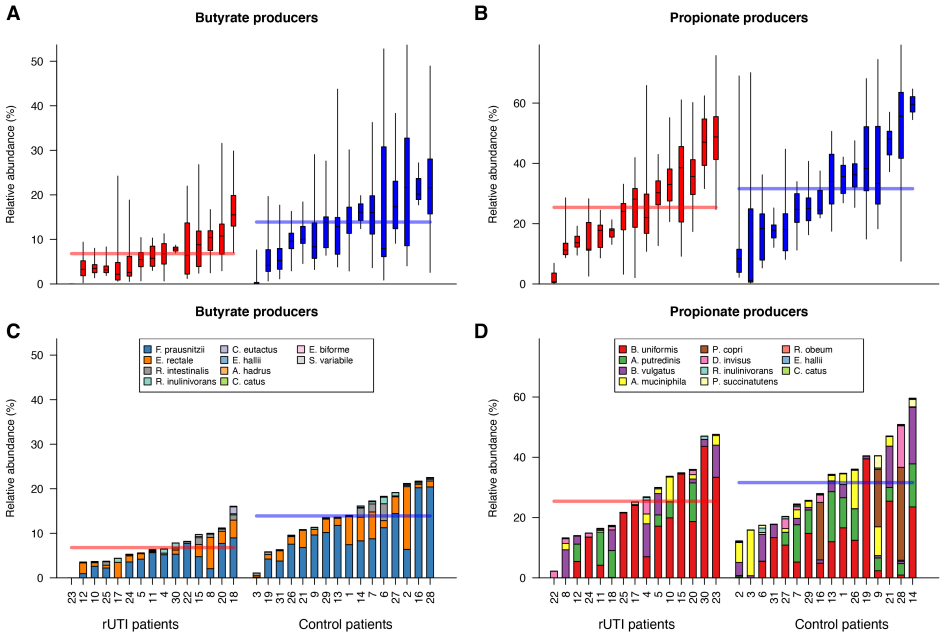

Figure C.2: SCFA producing bacteria are depleted in the rUTI gut. Cumulative relative abundances of **(a)** butyrate and **(b)** propionate producing bacterial species in rUTI and control samples. Box plots display the median (center line), 25th and 75th percentiles (box), as well as the 5th and 95th percentiles (whiskers). Within-host average relative abundances of individual species for **(c)** butyrate and **(d)** propionate producers are also shown. Horizontal lines denote the mean relative abundance in rUTI (red) and control (blue) women.

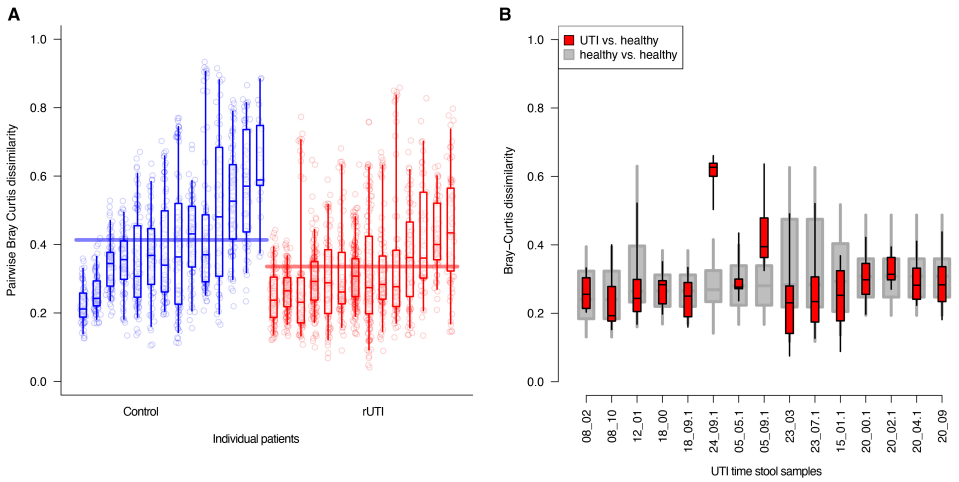

Figure C.3: Bray Curtis dissimilarity across stool samples. (a) For each patient, the distribution of Bray-Curtis dissimilarities between all stool samples, ordered by increasing mean patient values within each cohort. **(b)** Bray-Curtis distributions between samples taken at the time of UTI vs. healthy time points (red), compared to all pairwise healthy sample comparisons. Box plots show the median (center line), 25th and 75th percentiles (box), as well as the 5th and 95th percentiles (whiskers).

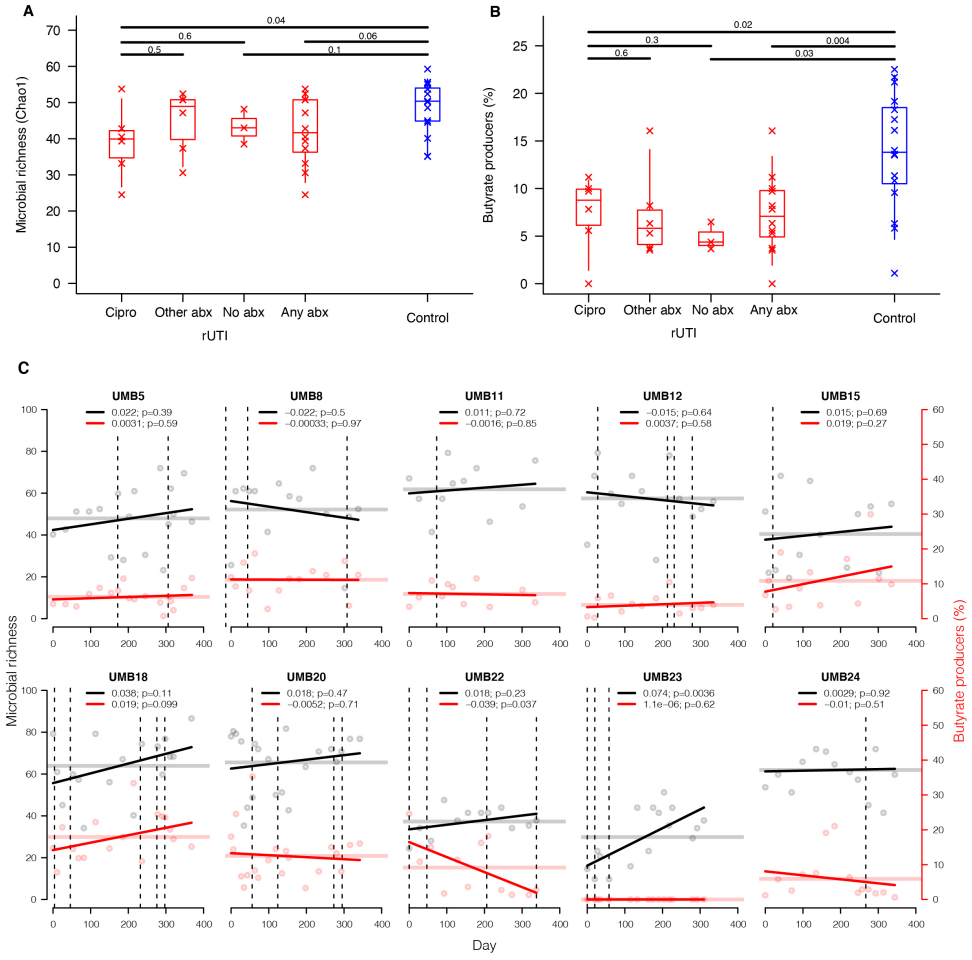

Figure C.4: rUTI dysbiosis is not driven by antibiotic use during the study. We grouped rUTI women according to their antibiotic exposures at any point during the UMB study; (i) ciprofloxacin ($n = 6$) (ii) non-ciprofloxacin antibiotics ($n = 6$); (iii) no antibiotics ($n = 3$); (iv) any antibiotics ($n = 12$). Groups were compared against each other and against the control cohort ($n = 16$) for **(a)** overall microbial richness and **(b)** relative abundance of butyrate producers. Crosses represent mean values for individuals, boxplots denote the IQR and 95% central quantiles for each group. Wilcoxon rank sum tests (two-sided) were applied to group pairs to derive p-values. **(c)** Temporal trends of microbial richness (black) and relative abundance of butyrate producers (red) in all rUTI participants using antibiotics during the study. For each individual, linear models were fit to observations (points) over time; fitted trends are shown, with coefficients & p values reported at the top of each panel. Dashed vertical lines denote antibiotic usage. Participant mean values are represented by horizontal lines.

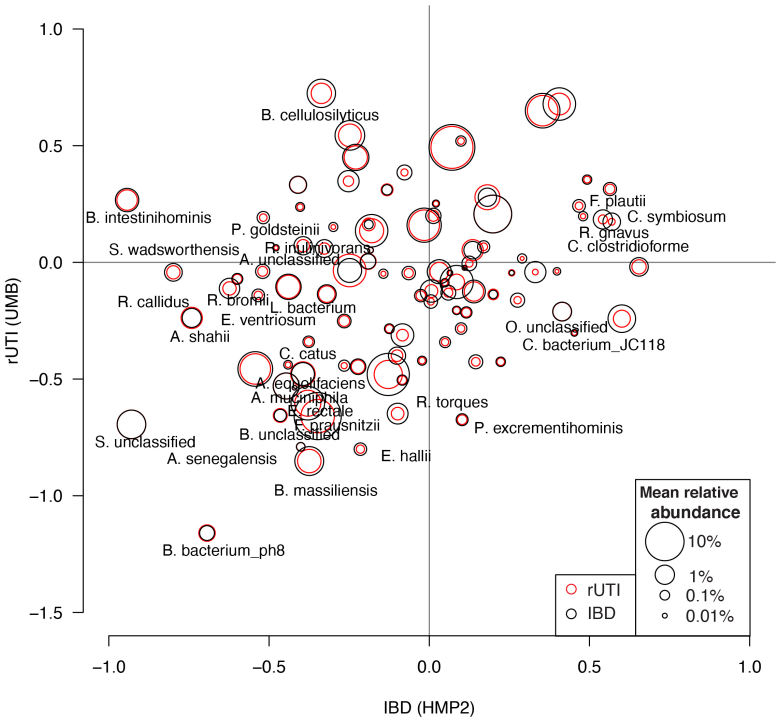

Figure C.5: Most species depleted in the rUTI gut are also depleted in the IBD gut. We compared discriminatory taxa in rUTI women to those in IBD patients using data from adult participants in the HMP2 study [1]. For each study, we fitted mixed effects models to standardized Metaphlan2 relative abundances as a function of categorical disease group (rUTI or IBD respectively, vs. each study's control cohort), including covariates for race and antibiotic use. The disease group coefficients are plotted against each other for each species, with circle pairs representing the average relative abundance in each study. Species with uncorrected p values <0.05 in either study are labelled. Species not present in at least 10% of samples in either study are excluded. IBD comprises patients with either CD or UC.

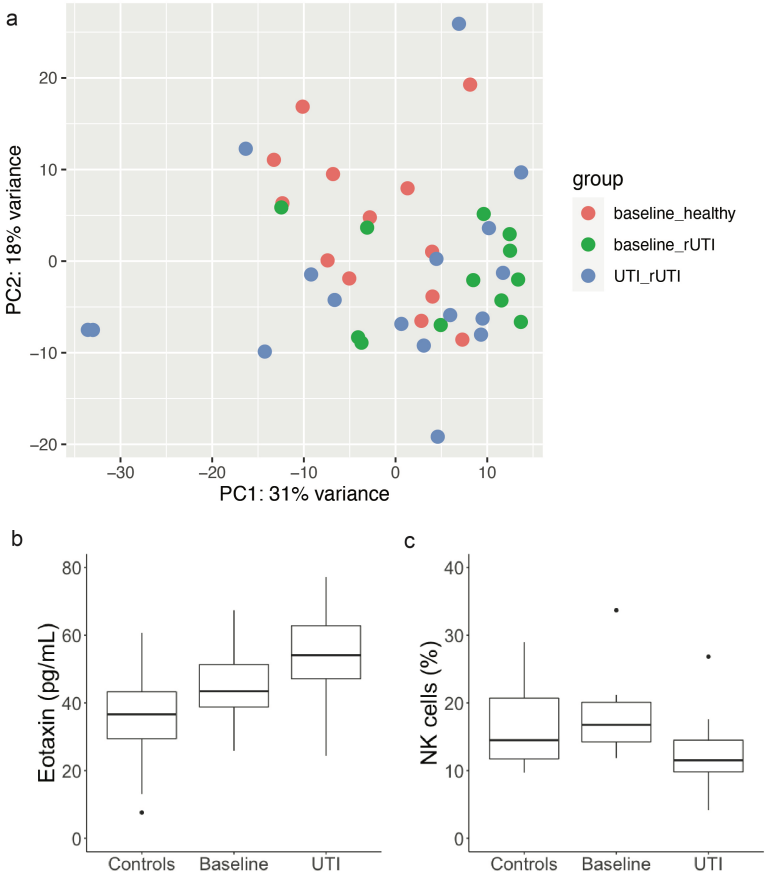

Figure C.6: Immunological differences between cohorts (a) PCA plot of gene expression across cohorts, based on PBMC RNA Seq data. Samples are partitioned into healthy controls ($n = 13$), rUTI patient baseline (enrollment; $n = 12$) and rUTI patient at time of UTI ($n = 17$). **(b)** Plasma eotaxin-1 levels in control women, and rUTI women at healthy enrollment and time of UTI. (c) Relative abundance of NK cells in control and rUTI women based on CIBERSORT output. Box plots display the median (center line), 25th and 75th percentiles (box), as well as data points within 1.5 IQR of the upper & lower quartiles (whiskers), and outliers beyond this range (dots).

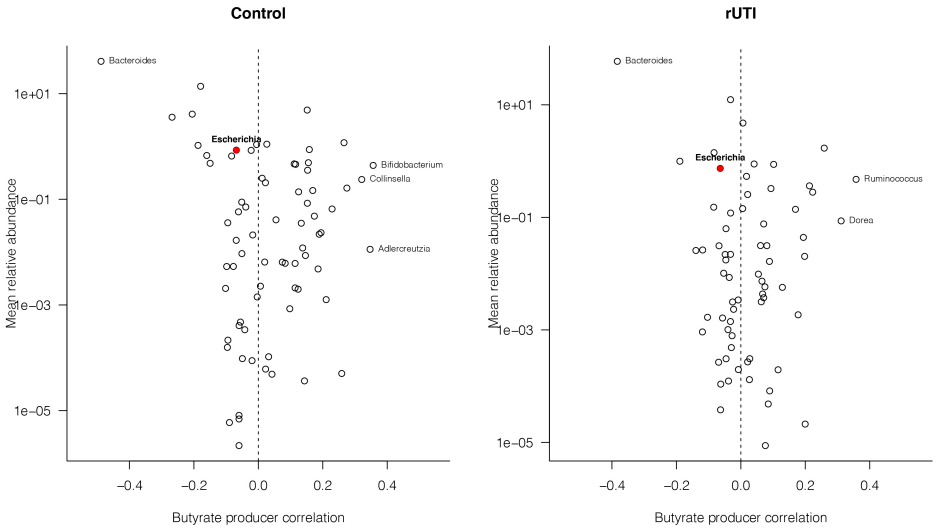

Figure C.7: Limited relationship between non-SCFA-producing taxa with butyrate producers. For all non-SCFA-producing genera detected across all samples, the correlation coefficient between its relative abundance and the relative abundance of butyrate producers was calculated and plotted against its mean relative abundance across **(a)** control ($n = 170$) and **(b)** rUTI ($n = 197$) samples. Genera with an absolute correlation coefficient greater than 0.25 are labeled, along with Escherichia, represented by the red point.

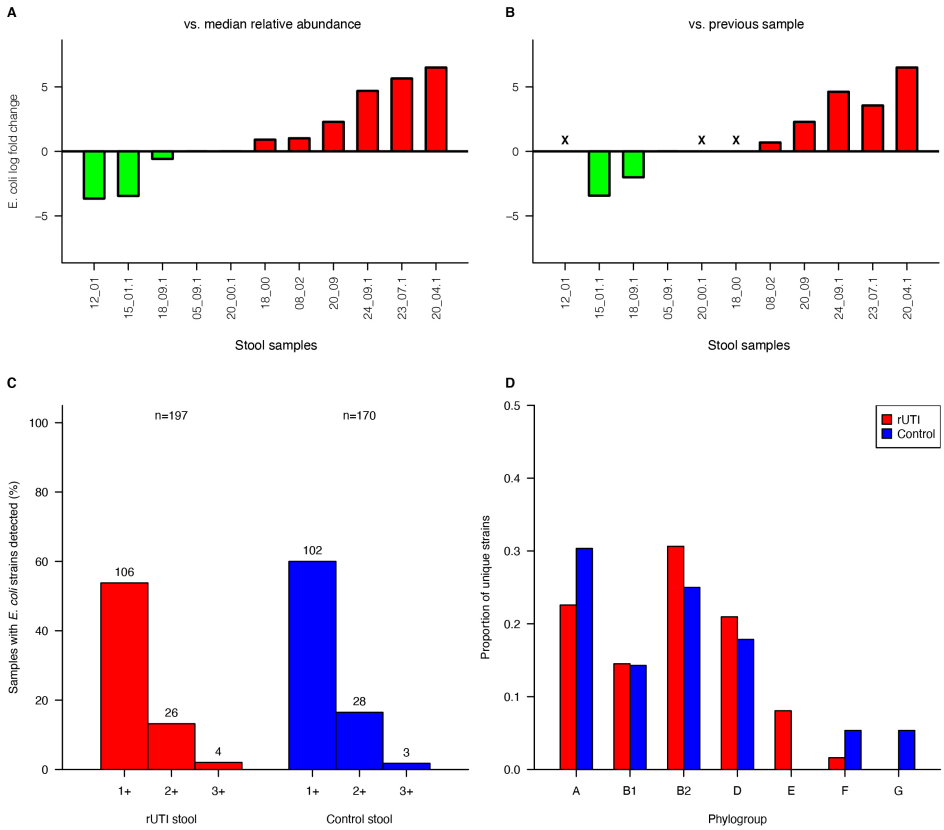

Figure C.8: *E. coli* relative abundance around the time of UTI and phylogroup distributions. For all stool samples taken within 3 days of a UTI event, the log fold change is given relative to **(a)** the median *E. coli* relative abundance in the corresponding patient, excluding samples taken at the time of UTI, and **(b)** the relative abundance of *E. coli* in the preceding stool sample. 'X' denotes samples for which there was no prior sample available. **(c)** Number of detected *E. coli* strains by sample type. **(d)** Number of detected StrainGST reference strains vs. relative abundance of *E. coli.*

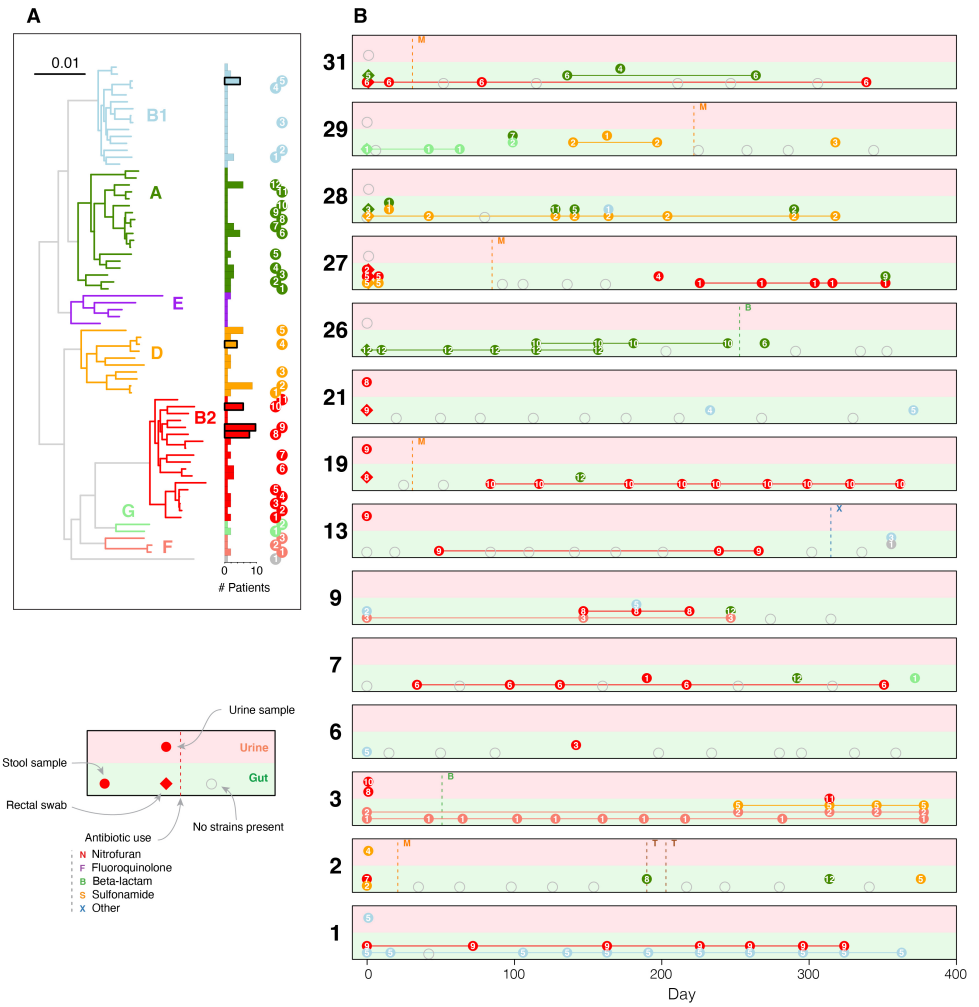

Figure C.9: Strain dynamics in control women. Strain dynamics within all control participants; analogous to Fig. 4.3 **(a)** Phylogenetic tree comprising strains called by StrainGE across all stool and urine samples, colored by phylogroup. Bars show number of unique participants with at least one strain observation; bars are bolded if the strain was identified in at least one urine sample. Each strain identified in control women is uniquely identifiable by the phylogroup (colour) and ID (numeral) indicated right. **(b)** Each panel represents longitudinal strain dynamics within one patient. Numerals refer to strain identifiers in (a). All fecal strains are connected to their most recent previous observation in fecal samples. Diamonds denote clinical rectal swabs. Strains identified in urine outgrowth depicted if available; otherwise raw urine strains are shown. Fecal or urine samples with no detected *E. coli* strains represented by open grey symbols. Vertical dashed lines represent self-reported antibiotic use.

C.2. Extended Data Tables

Additional supplementary data tables are available online at https://doi.org/10.1038/s41564-022-01107-x.

References

1. The Integrative HMP (iHMP) Research Network Consortium. The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles during Periods of Human Health and Disease. English. Cell Host & Microbe 2014 Sep; 16. Publisher: Elsevier:276–89

Acknowledgments

The journey toward the finish line of this PhD has been a long and winding road. It is not unlike traversing a mountainous area in Italy: sharp winding turns, steep hills, and deep valleys. Luckily, I had the support of many people along the way.

First and foremost, a big thank you to my supervisors and mentors at the Broad Institute. Thank you, **Ashlee**, for agreeing to this hybrid employee-PhD setup, which enabled the start of this journey. One day, I hope to reach your ability to see the big picture, tell stories around analyses, and ask the right questions. Your kind and positive approach to mentorship has made me a better scientist. Thank you, **Abby**, for teaching me much about bacterial biology, genomic analyses, and scientific writing. Your attention to detail is fantastic, and your ability to edit a manuscript is truly amazing. I am going to miss the frequent homemade cookies and other baked goods. Thank you, **Kiran**, for your support and shared excitement about bioinformatics algorithms, machine learning, and the Rust programming language. Your enthusiasm is infectious, and your ability to excite people about the need for better tools and algorithms is fantastic. I am looking forward to our future projects together!

Another big thank you to my mentors from the TU Delft. Thank you, **Thomas**, for offering me the option to do this hybrid PhD setup and helping me avoid getting too deep into the engineering side of a project. Thank you, **Marcel**, for your suggestions and help getting this PhD to the finish line.

Thank you to all my amazing colleagues in the bacterial genomics group: **Bruce, Colin, Tim, Terry, Alejandro, Alexandra, Sushmita, Josh, Rauf, Ryan, Mark, Sozie, and Marco**. I have learned so much from you all about bacterial biology and genomic data analyses. Thanks to my colleagues at the Data Sciences Platform, including **Fabio, Ryan, John, Steve, Shadi, Hang, James, and Beri**, for similarly nerding out on genomic graph data structures, sequence alignment, Hidden Markov Models, and machine learning. I have enjoyed the many conversations and discussions we have had.

I owe much gratitude to many people who have made my time in Boston amazing. Thanks to my roommates **Sean, Mohammed, Evan**, and, by extension, **Matte**, who welcomed me warmly. I greatly enjoyed our skiing trips, hikes, and many nights out together. Surviving a pandemic could have been a lot worse if it weren't for you guys. Thanks to my friends at *Minuteman Field Hockey Club*, including **Ed, Kim, Wouter, Jochem, Caroline, Alyn, Andrew, Rishi, Mark, Franny, Maggie**, and many more for the many fun pickup games, tournaments across the country, and fantastic Thanksgiving and Christmas dinners. Thanks to the many fun people of the *Nederlandse Borrels Boston*-group, including **Wim, Suus, Mart, Tjerk, Michel, and Vincent** for the beers, cheese, and Kingsday events.

Thank you to my friends from Delft, **Tom, Dorus, and Joey** for always welcoming me back each time I am back in the country. Our weekly Zoom sessions during the pandemic were also a big help in enduring those crazy times. Similarly, thanks to my

jaarclubgenoten **Frank, Josco, Guus, Marten, Toon, Bastijn, Juriaan, and Marius** for our many New Year's Eve celebrations and the memorable New England trip.

I saved the most important thank yous for last. **Laura**, finishing this PhD would have been more challenging without a supportive partner like you. Our lovely dinner conversations, vacations to the American Southwest, Puerto Rico, and Italy, or cozy evenings playing games on the couch have kept me going. Thanks for all your love and I look forward to building many more memories together. I will also not forget to thank our cats, **Pepito** and **Lola**, for frequently accompanying me in the home office and occasionally contributing to the thesis by stepping on the keyboard. **Mayke and Susan**, I could not have wished for more amazing sisters, and thanks to your love, coming home was always fun. Finally, a big thank you to my parents, **Wim and Eveline**, for their continuous and unconditional love and support throughout each winding turn of my academic career. You have allowed me to explore numerous interests, even if the path to the destination was not always linear. You have always encouraged me when I needed it. Thank you for everything.

Curriculum Vitæ

Lucas Roeland van Dijk

13-08-1990	Born in ’s Gravenhage, the Netherlands.

Education

2002-2008	Pre-university education Staring College, Lochem, the Netherlands
2008-2013	BEng in Electrical Engineering The Hague University of Applied Sciences, Delft, the Netherlands
2013-2017	MSc in Computer Science (cum laude) TU Delft, Delft, the Netherlands
2017-2024	PhD in Computer Science TU Delft, Delft, the Netherlands

Professional Experience

Sep. 2012 - Dec. 2012	Software Engineering Intern Thales Nederland, Delft, the Netherlands
May 2013 - May 2016	Software Engineer Studio Bereikbaar, Rotterdam, the Netherlands
Jun. 2016 - Sep. 2016	Intern DSM, Delft, the Netherlands
Nov. 2017 - Aug. 2024	Computational Associate II The Broad Institute, Cambridge, MA, USA
Sept. 2024 - Present	Computational Scientist II The Broad Institute, Cambridge, MA, USA

List of Publications

1. **Van Dijk, Lucas R.**, Bruce J. Walker, Timothy J. Straub, Colin J. Worby, Alexandra Grote, Henry L. Schreiber, Christine Anyansi, et al. "StrainGE: A Toolkit to Track and Characterize Low-Abundance Strains in Complex Microbial Communities." Genome Biology 23, no. 1 (March 2022): 74. https://doi.org/10.1186/s13059-022-02630-0.

2. Worby, Colin J., Henry L. Schreiber, Timothy J. Straub, **Lucas R. van Dijk**, Ryan A. Bronson, Benjamin S. Olson, Jerome S. Pinkner, et al. "Longitudinal Multi-Omics Analyses Link Gut Microbiome Dysbiosis with Recurrent Urinary Tract Infections in Women." Nature Microbiology 7, no. 5 (May 2022): 630–39. https://doi.org/10.1038/s41564-022-01107-x.

3. Kim, Younhun, Colin J. Worby, Sawal Acharya, **Lucas R. van Dijk**, Daniel Alfonsetti, Zackary Gromko, Philippe Azimzadeh, et al. "Strain Tracking with Uncertainty Quantification." bioRxiv, (January 2023). https://doi.org/10.1101/2023.01.25.525531.

4. Young, Mark G., Timothy J. Straub, Colin J. Worby, Hayden C. Metsky, Andreas Gnirke, Ryan A. Bronson, **Lucas R. van Dijk**, et al. "Distinct Escherichia Coli Transcriptional Profiles in the Guts of Recurrent UTI Sufferers Revealed by Pangenome Hybrid Selection." Nature Communications 15, no. 1 (November 2024): 9466. https://doi.org/10.1038/s41467-024-53829-7.

5. **Van Dijk, Lucas R.**, Abigail L. Manson, Ashlee M. Earl, Kiran V Garimella, and Thomas Abeel. "Fast And Exact Gap-Affine Partial Order Alignment with POASTA." Bioinformatics (January 2025): btae757. https://doi.org/10.1093/bioinformatics/btae757.