Analyzing rare mutations in metagenomes assembled using long and accurate reads

Marcus W. Fedarko, Mikhail Kolmogorov and Pavel A. Pevzner

Genome Res. 2022 32: 2119-2133 originally published online November 23, 2022
Access the most recent version at doi:10.1101/gr.276917.122

Method

Analyzing rare mutations in metagenomes assembled using long and accurate reads

Marcus W. Fedarko,[1,2] Mikhail Kolmogorov,[1,2,3] and Pavel A. Pevzner[1,2]

[1]*Department of Computer Science and Engineering, University of California San Diego, La Jolla, California 92093, USA;* [2]*Center for Microbiome Innovation, University of California San Diego, La Jolla, California 92093, USA;* [3]*UC Santa Cruz Genomics Institute, Santa Cruz, California 95064, USA*

The advent of long and accurate "HiFi" reads has greatly improved our ability to generate complete metagenome-assembled genomes (MAGs), enabling "complete metagenomics" studies that were nearly impossible to conduct with short reads. In particular, HiFi reads simplify the identification and phasing of mutations in MAGs: It is increasingly feasible to distinguish between positions that are prone to mutations and positions that rarely ever mutate, and to identify co-occurring groups of mutations. However, the problems of identifying rare mutations in MAGs, estimating the false-discovery rate (FDR) of these identifications, and phasing identified mutations remain open in the context of HiFi data. We present strainFlye, a pipeline for the FDR-controlled identification and analysis of rare mutations in MAGs assembled using HiFi reads. We show that deep HiFi sequencing has the potential to reveal and phase tens of thousands of rare mutations in a single MAG, identify hotspots and coldspots of these mutations, and detail MAGs' growth dynamics.

[Supplemental material is available for this article.]

Introduction

Deep DNA sequencing and rare mutations

Representing a species using a "reference genome" alone is an inherently limited approach, because any population of a species—even in an isolate (Leyn et al. 2021), let alone a metagenome—has mutations with highly variable frequencies throughout the genome. In the past decade, deep DNA sequencing has enabled studies of diversity in cell populations that represent a departure from the "reference genome" concept (Wilm et al. 2012). These studies' foci span many areas, including viral quasi-species (Henn et al. 2012), bacterial strains (Toprak et al. 2012; Rugbjerg and Sommer 2019; Leyn et al. 2021), cancer evolution (Gerlinger et al. 2012; Andor et al. 2016), cell-free DNA (Fan et al. 2008), forensic applications (Jäger et al. 2017), and somatic mutations (Schmitt et al. 2012; Jaiswal et al. 2014).

Previous experimental evolution studies have greatly contributed to our understanding of mutation rates and the emergence of novel bacterial strains (Barrick et al. 2009). However, they have mainly focused on relatively frequent mutations (e.g., mutations with frequency exceeding 1%) in isolates rather than in metagenomes. Recent studies have shown the importance of detecting rare mutations (Toprak et al. 2012; Leyn et al. 2021; Zlamal et al. 2021): The tasks of identifying rare mutations in a microbial population and monitoring how some of them evolve into frequent mutations are important for understanding the dynamics of the acquisition of antibiotic resistance in a pathogen. For example, the presence of a strain with a drug-resistant mutation with a population frequency of 0.1%—a rare, albeit nonzero, frequency—may lead to faster emergence of drug resistance in this population, compared with a population without a single microbial cell carrying this mutation. However, sequencing errors in reads often prevent the detection of rare mutations, particularly when the mutation frequency is less than the error rate in reads (Wilm et al. 2012).

Short-read DNA sequencing has greatly contributed to the detection of rare mutations (Salk et al. 2018), but phasing these mutations using short reads alone remains challenging: Short-read metagenomic studies rarely result in the complete assembly of even a single metagenome-assembled genome (MAG) (Nurk et al. 2017). The recent emergence of the long and accurate Pacific Biosciences (PacBio) high-fidelity (HiFi) reads (Wenger et al. 2019) has led to the era of "complete metagenomics" by enabling complete or nearly complete assemblies of hundreds of MAGs from a single HiFi data set (Kolmogorov et al. 2020; Bickhart et al. 2022; Kim et al. 2022). This shift allows fundamentally new possibilities in the analysis of metagenomes.

Identifying rare mutations in metagenomic data

Any attempt to identify rare mutations must be mindful of their false-discovery rate (FDR): the fraction of identified mutations that are false. In general, a sequencing technology's error rate is not a single fixed value: Error rates in reads differ depending on many factors (Wilm et al. 2012), for example nucleotide substitution types or sequence-specific errors (Nakamura et al. 2011; Wenger et al. 2019). Identifying rare mutations is therefore a difficult problem that may result in a large FDR.

LoFreq (Wilm et al. 2012) is a variant caller that has been shown to perform well for short-read sequencing data (Wilm et al. 2012; Sandmann et al. 2017; Kille et al. 2021); however, its usual reliance on Phred quality scores limits its applicability to HiFi sequencing data, for which Phred scores are not always available (Fukasawa et al. 2020). If Phred scores are not available, LoFreq's "LoFreq-NQ" module attempts to learn error probabilities

Corresponding author: ppevzner@ucsd.edu
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.276917.122.

for the 12 nucleotide substitution types (A → C, C → A, A → G, etc.); these error probabilities are used to inform variant calling in lieu of Phred scores (Wilm et al. 2012). Although useful, this approach makes the assumption that the probability of a given substitution happening owing to sequencing error—rather than real variation—remains constant for all positions in the genome. This assumption will almost certainly be broken in practice because the effects of a given single-nucleotide mutation vary depending on the position at which this mutation occurs. For example, in the standard genetic code, the mutation A → C is synonymous for the third position in the codon CTA, because both CTA and CTC code for leucine; however, this same mutation is nonsynonymous for the third position in the codon CAA because CAA codes for glutamine but CAC codes for histidine. The probability with which we assume that A → C could happen owing to sequencing error could thus be adjusted to account for this biological context; we contend that knowledge about this sort of "context-dependent" information should improve the quality of mutation identifications.

Other technology-agnostic methods besides Lo-Freq NQ, such as DeepVariant (Poplin et al. 2018), could also be applied to the problem of identifying mutations in HiFi metagenomic data; however, there remains potential to take contextual sequence information into account, as discussed, to further improve the algorithms used by these tools. Additionally, machine-learning-based methods that have been trained on human sequencing data may not perform similarly well for metagenomic data (Sapoval et al. 2022).

Simulation models of sequencing data

The MetaSim simulation model of Illumina reads (Richter et al. 2008), which was used in benchmarking LoFreq (Wilm et al. 2012), has proved to be valuable in many applications; however, it—and other simulation models—still has some limitations. For example, extant simulation models do not model the cross talk between wells in the patterned system on Illumina plates (Wang et al. 2017) or dimness owing to slowly amplifying clusters (affected by insert size or GC bias). Many bioinformaticians may not be aware about these small variations in error rates, because they hardly affect base-calling in reference genomes; however, they nonetheless affect the identification of rare mutations and thus make estimating the FDR of these identifications challenging.

Developing realistic simulation models for HiFi reads represents a similar challenge (Ono et al. 2021). There have been recent developments on this front, including Sim-It (Dierckxsens et al. 2021) and HIsim (Myers 2021; Suzuki and Myers 2022); however, we will propose an approach for estimating the FDRs of identified mutations that can bypass the need for a simulation model and should be applicable to arbitrary long and accurate sequencing data.

Phasing rare mutations in HiFi metagenomic data

Given a set of identified mutations in a metagenome, the next step in assembling strains of a microbe is phasing—for example, answering the question of whether identified mutations at N positions within a MAG represent N alternate strains of this MAG (each with a single mutation), a single alternate strain with N mutations, or something in between. This strain separation problem (Vicedomini et al. 2021) is important because strains with small genomic differences may have vastly different phenotypes: for example, although some *Escherichia coli* strains are harmless, others may cause disease outbreaks (Frank et al. 2011; Vicedomini et al. 2021). The detection and phasing of rare mutations are prerequisites for numerous downstream applications, such as monitoring rare drug-resistant subpopulations (Leyn et al. 2021). It is therefore important to identify as many rare mutations as possible—and to determine which of them are carried together—while simultaneously controlling the FDR of these identifications.

We define a strain as a unique haplotype supported by the available sequencing data, an approach analogous to that taken by Nicholls et al. (2021) and Vicedomini et al. (2021). The field of metagenomic phasing is still in its early stages: Although previous studies have succeeded in separating strains at the level of individual genes or other "genomic features" (Cleary et al. 2015; Luo et al. 2015; Quince et al. 2017; Nicholls et al. 2021), strain separation at the level of a long region or even a complete genome is not yet a solved problem, especially for high-complexity metagenomes (Vicedomini et al. 2021).

Even though long-read technologies promise to resolve complete genomes of strains within a bacterial community (Bertrand et al. 2019; Bickhart et al. 2022; Feng et al. 2022; Sereika et al. 2022), many long-read assemblers suppress rather than reveal small variations to achieve high assembly contiguity (Koren et al. 2017; Kolmogorov et al. 2019; Bickhart et al. 2022). Exceptions include the metaFlye assembler (Kolmogorov et al. 2020), which includes a "strain mode" (the --keep-haplotypes flag) aimed at strain detection, and Strainberry (Vicedomini et al. 2021), which starts from a "strain-oblivious" assembly and iteratively phases it to generate a "strain-aware" assembly. A critical factor for the success of strain separation in both metaFlye and Strainberry is the amount of variation between strains—these tools typically succeed when the percent identity between the strains is below 97%–99% and when there are very few (typically two to three) strains. Another important concern is the FDR of the identified rare mutations, because false-positive mutations mislead strain separation tools. The success of strain separation is thus dependent both on the reliable identification of rare mutations and on the ability to control their FDR.

The strainFlye pipeline

Here we present strainFlye, a pipeline for the FDR-controlled identification, phasing, and analysis of rare mutations in MAGs sequenced using long and accurate reads. Our approach draws on seemingly unrelated yet very relevant prior work on estimating the FDR of peptide identifications in proteomics. The first approach to peptide identification was proposed in 1994 (Eng et al. 1994) and was followed by many other tools addressing the same problem. However, it remained unclear how accurate these tools were and how to benchmark them: The target-decoy approach (TDA) for computing the FDR of peptide identifications, which revolutionized the field of proteomics, was introduced years afterward (Moore et al. 2002; Elias et al. 2005; Elias and Gygi 2007; Käll et al. 2008). Although FDR evaluation is now the requirement in proteomics, this is not the case in studies of rare mutations in metagenomes. As accurate sequencing technologies enable the identification of thousands of rare mutations within complete MAGs, more than can be easily investigated manually, we believe that applying FDR estimation to this process will become a requirement. We describe an analog of the TDA for evaluating the FDR of identified rare mutations in a metagenome and further extend this idea by developing a *context-dependent* TDA.

We show that deep HiFi-based sequencing can reveal tens of thousands of mutations (with controlled FDR) in a MAG and that

these mutations can form hotspots (that may point to selection) and coldspots (that can be used for designing drugs targeting such regions), and we present new methods for phasing these mutations.

Results

Demonstrating strainFlye

Figure 1 illustrates the strainFlye pipeline. The only two required inputs to strainFlye are a HiFi read-set and the contigs (in the case of metaFlye output, edge sequences) assembled from these reads. strainFlye can also optionally take as input an assembly graph indicating contigs' adjacencies for use in adjusting one of the alignment filtering steps (Supplemental Material, "Read alignment").

To benchmark strainFlye, we used a HiFi read-set from a sheep gut metagenome (Methods) (Kolmogorov et al. 2020; Bickhart et al. 2022). This read-set is herein referred to as "SheepGut." We selected this data set because it resulted in, as of writing and to our knowledge, the largest number of complete MAGs among all metagenomic data sets analyzed so far (Bickhart et al. 2022). The data set includes 22,118,393 reads with total length 255,708,236 kbp and average length 11.6 kbp.

We assembled the SheepGut data set using metaFlye (Kolmogorov et al. 2020). The resulting assembly graph includes 78,793 edges (468 of which have lengths of at least 1 Mbp) distributed across 45,988 weakly connected components. The Supplemental Material, "Assembly graph" (Supplemental Fig. S1), provides further details about this assembly graph and how we ran metaFlye.

We classify a contig as *high-coverage* if its coverage is at least *minCov* (default value 1000×) and *long* if its length is at least *minLength* (default value 1 Mbp). We selected three high-coverage and long contigs from the SheepGut metaFlye assembly graph to illustrate various steps of strainFlye's pipeline. These three "selected MAGs" are herein referred to as CAMP, BACT1, and BACT2, as abbreviations of their respective Kaiju (Menzel et al. 2016) taxonomic classifications: *Campylobacter jejuni*, *Bacteria*, and *Bacteroidales*. The Supplemental Material, "Assembly graph" (Supplemental Table S1; Supplemental Fig. S2) and "Coverages and deletion-rich positions" (Supplemental Figs. S3–S5), provide further information about these MAGs and their coverages.

The bulk of this paper's analyses focus on the SheepGut data set; however, we provide a brief demonstration of strainFlye on another HiFi metagenomic data set (referred to as "ChickenGut") in the Supplemental Material, "Demonstrating strainFlye on the ChickenGut dataset."

Computing mutation spectra

We use the term *mutation spectrum* to refer to the collection of mutation frequencies across all positions in a contig. We found that applying LoFreq (Wilm et al. 2012) to the large SheepGut read-set was time-consuming (Supplemental Material, "Applying LoFreq to the SheepGut dataset"). Below we show that, in the case of HiFi reads, a simple variant calling method (herein referred to as NaiveFreq) generates similar sets of rare mutations as LoFreq in a fraction of the time. We limit our focus to substitution mutations because the substitution-based error rate in HiFi reads is an order of magnitude smaller than the indel-based error rate (Wenger et al. 2019).

Given an alignment of reads to contigs (Supplemental Material, "Read alignment"), for each position *pos* in a contig, we consider the number of reads spanning this position with a match/mismatch operation in the alignment (spelling one of the four nucleotides A, C, G, or T; we ignore degenerate nucleotides aligned to a position). We define the sum of the numbers of these reads as *reads*(*pos*). We define the number of reads of the second-most common nucleotide aligned to *pos* as *alt*(*pos*) (breaking ties arbitrarily).

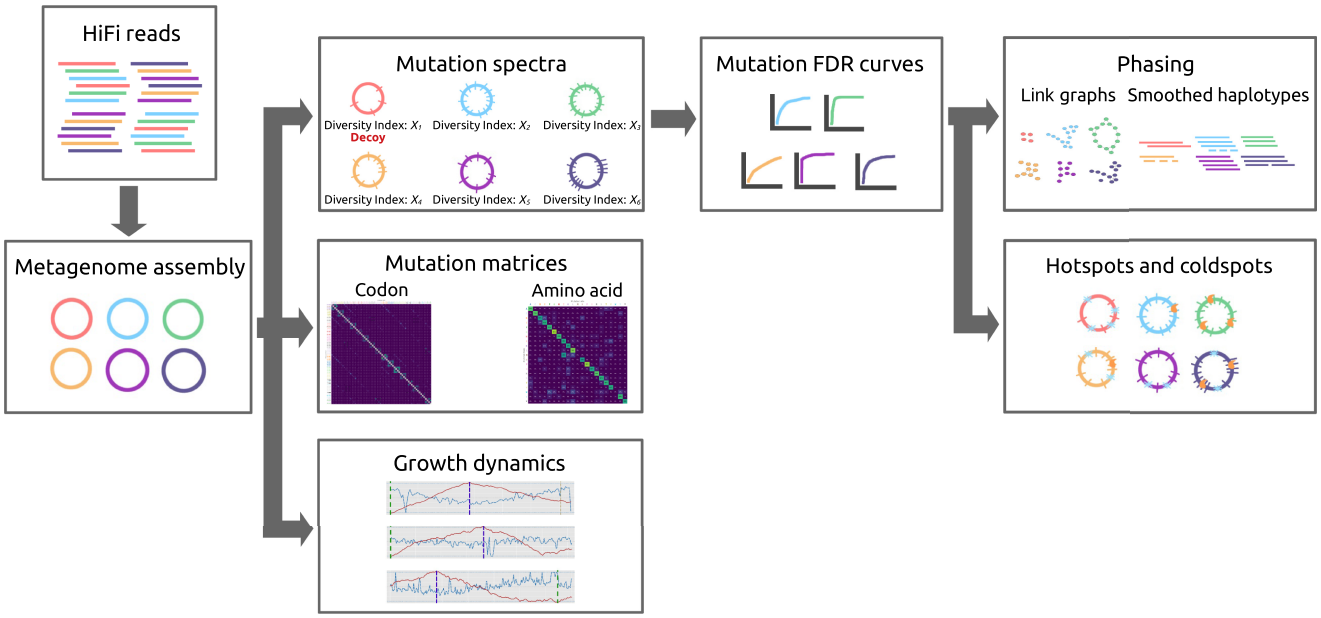

Figure 1. strainFlye pipeline. Given a metagenome assembly of HiFi reads, strainFlye identifies rare mutations in each MAG, estimates their FDR, performs phasing analyses using these mutations, identifies hotspots and coldspots of mutations, produces MAG-specific codon and amino acid mutation matrices, and computes information about MAGs' growth dynamics.

NaiveFreq estimates the mutation frequency of *pos* as $freq(pos) = \frac{alt(pos)}{reads(pos)}$; this value is constrained to the range [0%, 50%]. Given a *frequency threshold* percentage $p \in (0\%, 50\%]$, NaiveFreq classifies a position *pos* as a *p*-mutation if $freq(pos) \geq p$ and if $alt(pos) \geq minAltPos$ (by default, we set $minAltPos = 2$ because, in general, a single read with an alternative nucleotide is an unreliable indicator of a mutation). For the sake of simplicity, we only consider single-allelic mutations (i.e., we do not attempt to call multiple *p*-mutations at a given position), although the methods proposed in this paper could be extended to account for multiallelic mutations.

NaiveFreq is implemented in strainFlye's pipeline in the "strainFlye call p-mutation" command. On its own, NaiveFreq is not especially useful: *p*-mutations represent a very primitive way of defining mutations. NaiveFreq's utility comes from the ease with which we can vary *p* to produce many sets of identified mutations for a contig. As we will show, this property will simplify the process of generating an FDR curve (Käll et al. 2008) that shows how an increase in the number of identified mutations impacts the FDR estimate associated with these identifications.

The target-decoy approach for estimating the FDR of identified mutations

Whether rare mutations are identified by a state-of-the-art algorithm like LoFreq (Wilm et al. 2012) or through NaiveFreq, it is important to estimate their FDR. In the absence of widely adopted realistic simulation models of errors in HiFi reads, we propose an analog of the target-decoy approach (TDA) commonly used in areas of bioinformatics without realistic simulation models for data generation (Elias and Gygi 2007; Gupta et al. 2011).

In proteomics, the key idea of the TDA is to use a *decoy* protein database (that has no real matches with observed mass spectra) to evaluate the FDR of spectral matches against a real protein database (Elias and Gygi 2007; Käll et al. 2008). We contend that the FDR of identified rare mutations in a given contig can be evaluated similarly, using a contig without any (or with very few) real rare mutations as our decoy.

Estimating the FDR of identified mutations in a contig using the TDA necessitates first attempting to select a *decoy contig* within a metagenome that does not contain any real mutations. We can then apply a mutation identification tool to this decoy contig (containing L positions), assume that all M identified mutations in the decoy contig are false, and compute a mutation rate $rate_{decoy} = \frac{M}{3L}$. The multiplication by three in the denominator accounts for how there are three possible single-nucleotide mutations at each position. Afterward, if analysis of a different ("target") contig using the same mutation identification tool results in a mutation rate of $rate_{target}$ (computed analogously as $\frac{M_{target}}{3L_{target}}$), we can estimate the FDR of the identified mutations in the target contig as $\frac{rate_{decoy}}{rate_{target}}$. (We note that our use of the term "mutation rate," which represents the ratio of the numbers of observed and possible mutations, differs somewhat from the commonly used definition of the mutation rate as the number of mutations per base pair per generation—e.g., as used by Barrick et al. 2009.)

Applying the TDA in this way faces the immediate challenge that "ideal" decoy contigs without any real mutations are unlikely to exist within a metagenome. Therefore, the estimate $\frac{rate_{decoy}}{rate_{target}}$ represents an upper bound for the FDR that, depending on the choice of decoy contig, may greatly inflate the estimated FDR. However, because our analysis revealed great variations (by orders of magnitude) in mutation rates across various contigs within a metagenome, we can simply select the least mutated contig and use it as a (nonideal but pragmatic) substitute for a decoy (Methods). We also modify the standard TDA to focus our FDR estimation on "rare" mutations with frequencies below a configurable threshold and thus limit the effects of "indisputable" mutations on the estimation of realistic FDRs (Methods).

We note that the TDA was initially designed to estimate FDRs in situations without an adequate model for data generation where many unknown factors affect the fidelity of the data. As such, the TDA is well-suited for analyzing reads with poorly understood sources of errors—for example, cross-talk between wells in the case of Illumina reads, or the complex derivation of accurate HiFi reads from error-prone CLR reads. We emphasize that the target-decoy approach results in an empirical FDR estimate, rather than an exact formula for computing the FDR. Many recent papers (Gupta et al. 2011; Keich et al. 2015, 2019; Emery et al. 2020) discuss pros and cons of the TDA and describe modifications of it that result in even more accurate FDR estimates.

Similar to how the TDA is used in proteomics (in a way that does not explicitly address complex effects such as internal ions, etc.), our use of the TDA for analyzing rare mutations does not explicitly address complex effects such as well–cross talk, dimness, etc. Instead of attempting to provide some sort of confidence score for individual mutation identifications, the TDA estimates the fraction of erroneously identified mutations "at bulk" within a target contig without trying to investigate their specific sources (Gupta et al. 2011).

Estimating the FDR of identified rare mutations using the TDA

At $p = 0.5\%$, NaiveFreq identifies 249, 17,069, and 1632 rare ($freq(pos) < 5\%$, as described in the Methods) *p*-mutations in CAMP, BACT1, and BACT2, respectively. This illustrates that there exists a difference of nearly two orders of magnitude in mutation rates across these MAGs (6.4×10^{-5}, 2.6×10^{-3}, and 1.9×10^{-4} for CAMP, BACT1, and BACT2, respectively). If CAMP, which has a relatively low mutation rate, is selected as a decoy, then the FDR for BACT1 at $p = 0.5\%$ is estimated as $\frac{6.4 \times 10^{-5}}{2.6 \times 10^{-3}} \approx 2.4\%$ (Fig. 2). For comparison, the Supplemental Material, "Applying LoFreq to the SheepGut dataset," shows this estimation process using LoFreq outputs on the three selected MAGs.

The Supplemental Material, "Growth of the number of *p*-mutations per megabase as *p* decreases" (Supplemental Fig. S8), illustrates how the number of *p*-mutations in the three selected MAGs grows as the frequency threshold p decreases. Each value of p we use to call *p*-mutations implies a mutation rate for the decoy and target contig alike; Figure 2 shows multiple FDR curves (Käll et al. 2008) for eight high-diversity-index (Results subsection "Diversity indices") target contigs in the SheepGut data set, computed by adjusting the values of p used. The orange curves in this figure use the entirety of CAMP as a decoy contig; the other curves use *context-dependent* decoy contigs constructed from CAMP, as will be described shortly, to provide different estimates of the FDR. The Supplemental Material, "Demonstrating strainFlye on the ChickenGut dataset" (Supplemental Fig. S6), provides an analogous version of this figure for the ChickenGut data set.

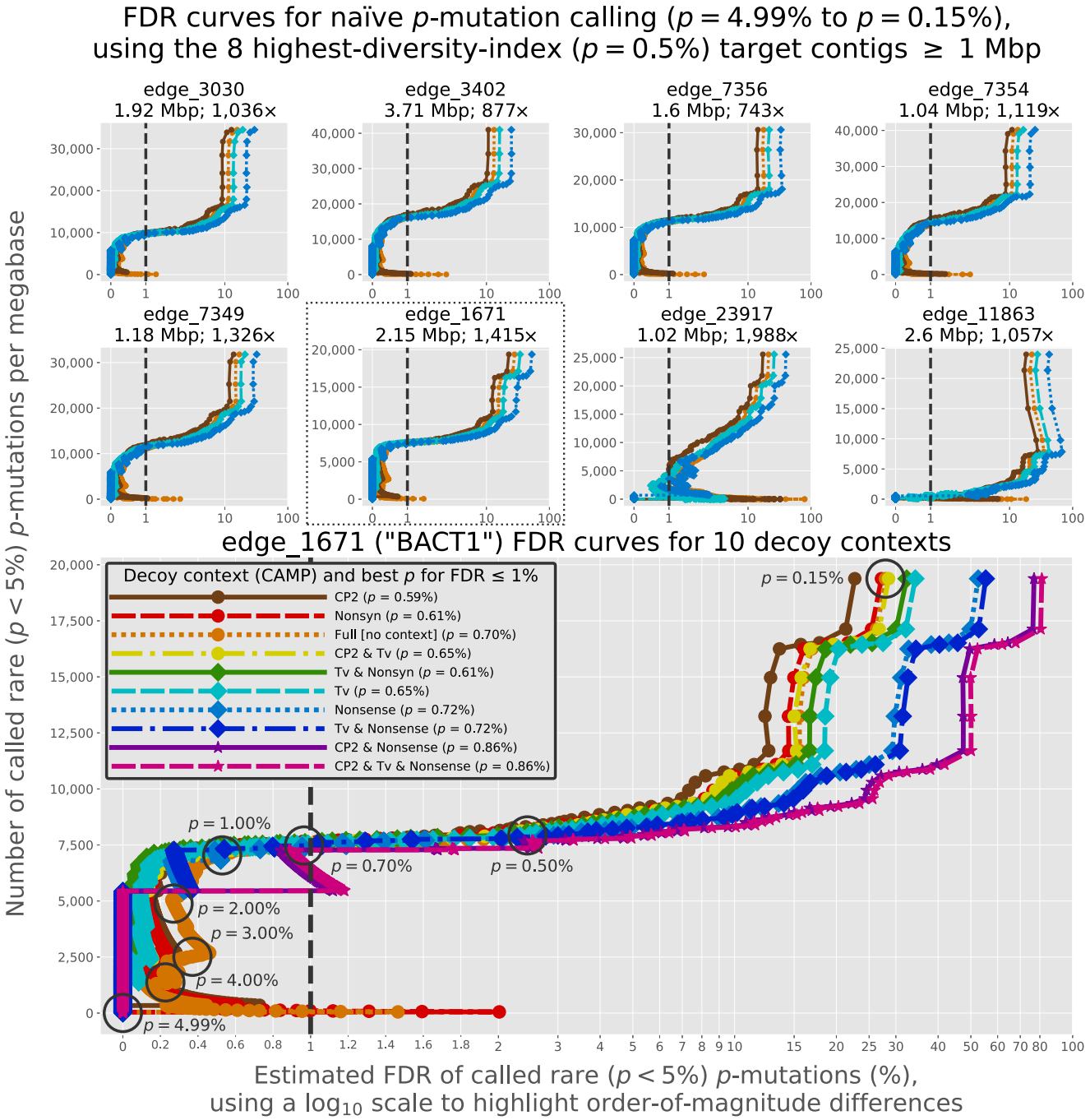

Figure 2. FDR curves for eight target contigs in SheepGut. We generate each FDR curve by using NaiveFreq to identify *p*-mutations in a selected decoy contig (CAMP) as well as the target contig, varying *p* from 4.99% to 0.15% in increments of 0.01%. (*Top*) Demonstration of four basic decoy contexts, resulting in four FDR curves per target contig. The "Full" context considers all mutations in all positions of the decoy contig; the "CP2" context considers all mutations in only positions located in the second codon position of a single predicted gene in the decoy contig; the "Tv" context only considers transversion mutations in all positions of the decoy contig; and the "Nonsense" context only considers single-nucleotide nonsense mutations in only positions located in a single predicted gene in the decoy contig. (*Bottom*) Demonstration of 10 decoy contexts, using BACT1 as the target contig. In addition to the four contexts shown in the above plots, this includes the "Nonsyn" context (corresponding to nonsynonymous mutations) as well as combinations of contexts (Supplemental Material, "Nonsynonymous, nonsense, and transversion decoy contexts"). Fixing the estimated FDR to 1% (indicated by the vertical dashed line shown in all plots) implies a "best" (smallest) value of *p* for a target contig that allows calling the rarest *p*-mutations while keeping the estimated FDR ≤ 1%: For BACT1, these values are listed in the legend for each decoy context. For clarity, we circle and label certain values of *p* on the "Full" curve.

The "strainFlye fdr estimate" command performs decoy contig selection and FDR estimation; furthermore, given the resulting FDR estimates, the "strainFlye fdr fix" command selects the "optimal" value of *p* for each target contig and filters each target contig's identified rare mutations to match this threshold value (Methods).

Context-dependent TDA

The described approach to FDR estimation, although useful, suffers from the fact that—even in relatively-low-mutation-rate contigs like CAMP—many naïvely called mutations represent real rather than false variations. To address this, we describe a *context-dependent target-decoy approach* based on the observation that certain types of mutations are rarer than others. Some positions in a genome are less likely to mutate than other positions, and some specific mutations at a given position are less likely to occur than other mutations. Thus, constructing a decoy contig consisting solely of these relatively mutation-resistant positions and/or mutation types should result in a more accurate FDR estimate.

Codon positions (CPs) are a simple example of this. Mutation rates vary sharply across the first, second, and third CPs (CP1, CP2, and CP3) within protein-coding regions of genomes (Bofkin and Goldman 2007): In general, mutations in CP3 are less likely to change the amino acid encoded by a codon than mutations in CP2 or, to a lesser extent, mutations in CP1. Given predicted protein-coding genes in a data set's contigs (Methods), we define four "groups" of positions for each contig: first/second/third positions of codons (CP1/CP2/CP3) and noncoding positions (positions located outside of predicted genes). We ignore positions that are located within multiple predicted genes owing to gene overlap. If we define M_P as the number of mutated positions identified within a group of positions P, we would expect $M_{CP2} < M_{CP1} < M_{CP3}$ (Bofkin and Goldman 2007). Figure 3 and Supplemental Figure S9 show that, on the three selected MAGs, this property holds for the mutations called by NaiveFreq at many values of p as well as for the mutations called by LoFreq. The Supplemental Material, "Codon position analysis details," presents additional information about the details of this analysis.

This result suggests that computing the decoy contig mutation rate as $\frac{M_{CP2}}{3L_{CP2}}$ may yield a more realistic FDR estimate. We refer to this modification as a *decoy context* that we apply to the original decoy contig; Figure 2 shows the use of this "CP2" decoy context for the CAMP decoy contig, showing that it generally lowers FDR estimates compared with using all of CAMP as a decoy contig.

Besides CPs, there exist other candidate events that show a sharp contrast between various mutation types and thus present promising options for the construction of decoy contigs. Whether owing to selection or other vagaries of the processes by which mutations occur, we expect nonsynonymous, nonsense, or transversion mutations to be rarer, respectively, than synonymous, non-nonsense, or transition mutations (Sonneborn 1965; Vogel 1972). The Supplemental Material, "Nonsynonymous, nonsense, and transversion decoy contexts," confirms this (Supplemental Fig. S10) and describes how strainFlye constructs decoy contexts using these "rarer" types of mutations; Figure 2 includes additional FDR curves produced using decoy contexts of possible nonsynonymous, nonsense, and transversion mutations, as well as various combinations of these contexts and the aforementioned CP2 context.

These analyses focus on p-mutations, in which mutation frequencies are used to call a position as mutated or not. Frequencies are useful for this task because they enable the (imperfect) comparison of positions or contigs with different coverages; the Supplemental Material, "Identifying mutations based solely on read counts," discusses an alternative method for calling mutations based solely on read counts.

Codon and amino acid mutation matrices

Proteins are usually compared using the PAM (Dayhoff et al. 1978) or BLOSUM (Henikoff and Henikoff 1992) matrices, based on frequencies of amino acid substitutions over large evolutionary distances (millions of years). It remains unclear whether these matrices are well suited for comparing proteins encoded by strains that are separated by short evolutionary distances. The complete metagenomics approach (Bickhart et al. 2022) provides the potential to derive MAG-based analogs of these matrices for short evolutionary distances. Moreover, it allows the extension of amino acid substitution matrices into more informative codon substitution matrices. To illustrate the potential of this approach, the Supplemental Material, "Constructing and visualizing mutation matrices," describes the construction of these matrices in detail and shows visualizations of them (Supplemental Figs. S11–S13) for the three selected MAGs.

Diversity indices

Below we describe the *diversity index* that quantifies the widely varying mutation rates across different species within a metagenome. Diversity indices are useful as a way to select a decoy contig for use with the TDA (Methods), and the "strainFlye call p-mutation" command outputs diversity indices for this reason.

We compute diversity indices relative to a given p threshold for naïvely calling p-mutations. We define a position as *sufficiently covered* if its coverage is at least *minSuffCov*, a threshold that is computed as a function of p (such that lower values of p generally correspond to a larger *minSuffCov*). We then define the diversity index of a contig G as the number of mutations called in the sufficiently covered positions in G, divided by the total number of sufficiently covered positions in G (given the selected p threshold value). The Supplemental Material, "Diversity index details," provides details on these definitions and their motivations.

We note that computing the diversity index of a complete bacterial genome in the context of short-read metagenomics is problematic, because studies using short reads alone struggle to assemble complete genomes; however, the emergence of complete metagenomics (Bickhart et al. 2022) has opened the possibility of reconstructing many complete MAGs from a microbial sample and thus analyzing their diversity indices. Figure 4 shows great variation in the diversity indices of the MAGs in the SheepGut data set.

Genomic locations of mutations

Given a set of identified mutations, strainFlye supports the identification of basic hotspots and coldspots (Wilm et al. 2012; Sekowska et al. 2016).

We define the *mutation rate* of a feature (an arbitrary region in a genome, e.g., a predicted gene) as the fraction of mutated positions in this region. The "strainFlye spot hot-features" command takes as input a list of genomic features in contigs and identifies "hotspot" features given simple user-specified thresholds (the minimum number of mutations a feature must have to be considered a hotspot and/or the minimum mutation rate a feature must have to be considered a hotspot). The Supplemental Material, "Hotspot genes in the three selected MAGs" (Supplemental Table S2), provides information about various hotspot genes for each of the three selected MAGs.

Figure 5 shows the mutation spectra of the highest-mutation-rate genes of the three selected MAGs (using NaiveFreq results at $p = 0.5\%$). The genes shown in Figure 5 (top, bottom) show essentially binary splits between their positions' mutation frequencies. However, the gene shown in Figure 5 (middle) offers a less clear interpretation: There exist three main groups of positions with high mutation frequencies (with frequencies ~1%–2%, 4%–5%, and

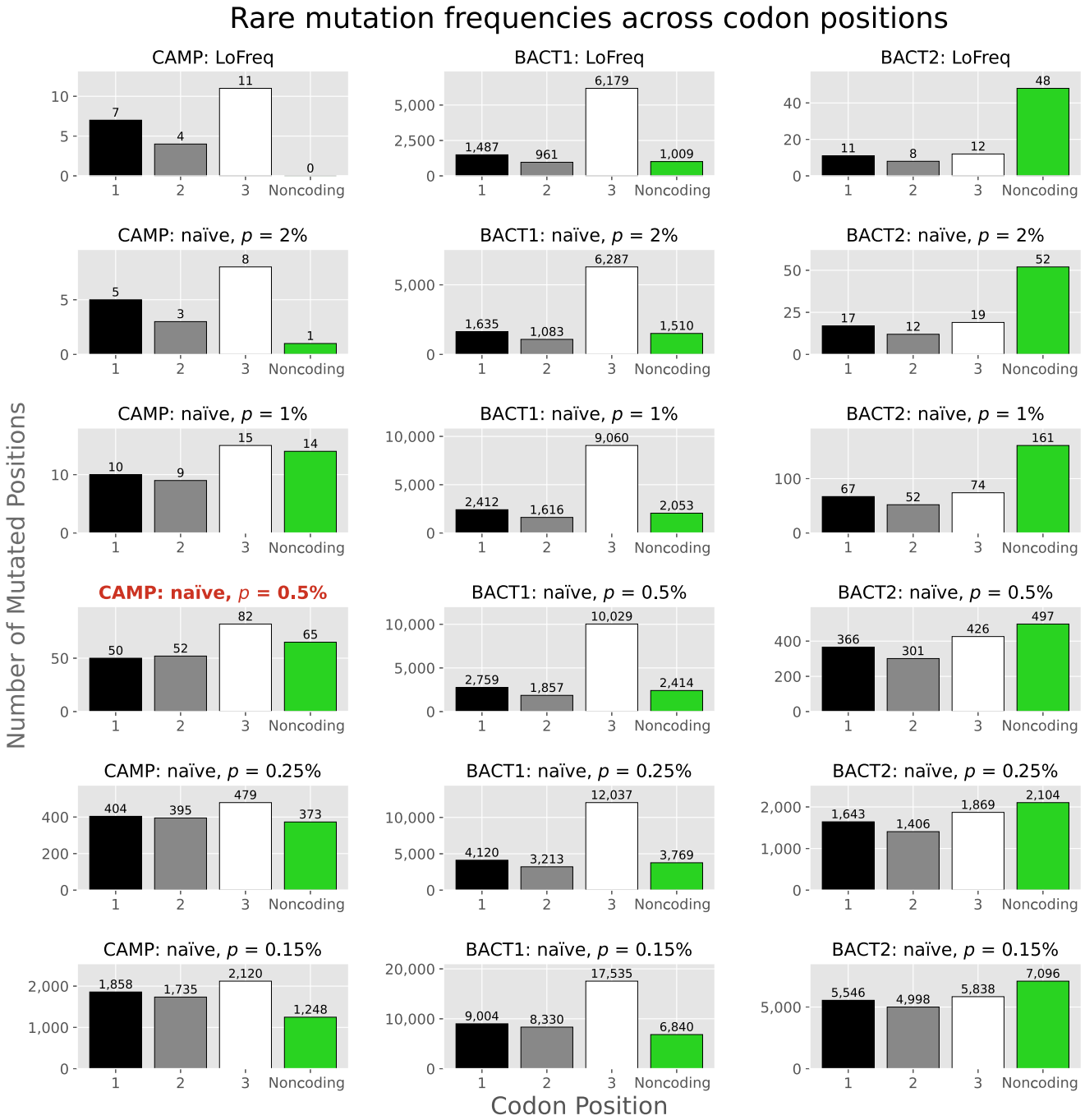

Figure 3. Rare mutation frequencies vary across codon positions in the three selected MAGs. This plot includes variant calls from LoFreq (*top* row) and results from NaiveFreq for $p \in \{2\%, 1\%, 0.5\%, 0.25\%, 0.15\%\}$ (*bottom* rows). The expected "pattern" of codon position mutation frequencies ($M_{CP2} < M_{CP1} < M_{CP3}$) holds for all plots except for the case of CAMP at $p = 0.5\%$ (the title for this plot is highlighted in red). As discussed in the Supplemental Material, "Applying LoFreq to the SheepGut dataset" (Supplemental Fig. S7), LoFreq's results are similar to NaiveFreq's at $p = 2\%$. To focus this visualization on "rare" mutations, we limit the mutations included in all rows (including LoFreq's) to those located in positions at which $freq(pos) < 5\%$ (Methods). We also treat positions where LoFreq called multiple mutations as if only a single mutation were called at this position (Supplemental Material, "Applying LoFreq to the SheepGut dataset"). The Supplemental Material, "Codon position analysis details" (Supplemental Fig. S9), shows an alternate version of this figure, in which *y*-axis values are normalized by the total number of positions considered in order to make plots from different MAGs more easily comparable.

6%–7%) covering this gene, as well as a group of mostly unmutated positions. The Supplemental Material, "Identifying strains in the most mutated gene of BACT1" (Supplemental Fig. S15), uses long reads to inspect the structure of mutations within this gene and analyze whether they are indeed associated with four different strains.

The Supplemental Material, "Plots of mutation locations" (Supplemental Fig. S16), visualizes the locations of mutations in

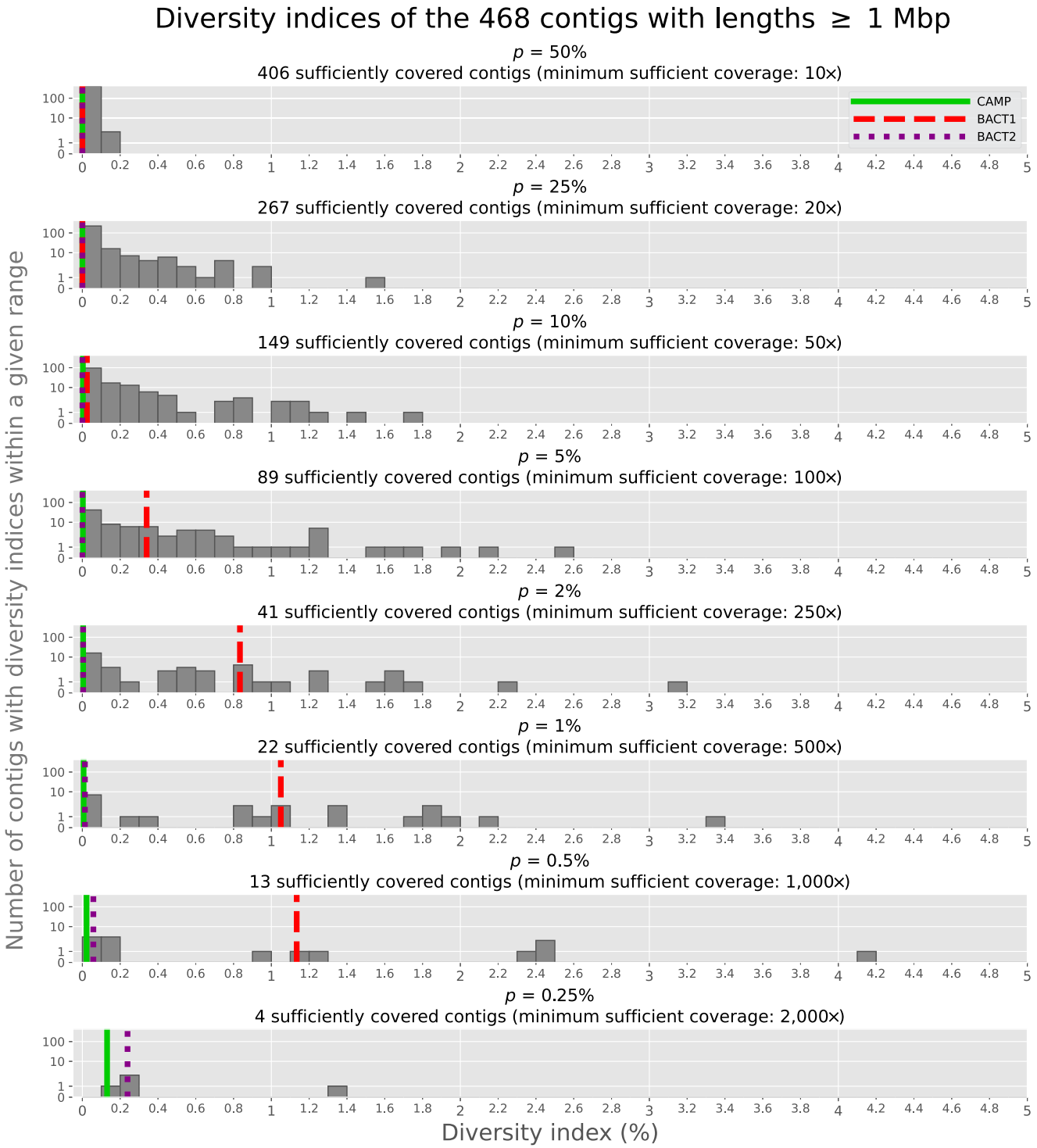

Figure 4. Diversity indices vary widely across the 468 long contigs in SheepGut. Smaller values of *p* allow us to "call" increasing amounts of *p*-mutations in a MAG, at the cost of requiring higher sequencing coverage to reliably distinguish these mutations from errors (we note that this differs from ordinary usage of NaiveFreq, which does not explicitly take coverage into account). The Supplemental Material, "Diversity index details," provides details about some of the high-diversity-index contigs shown in this figure and about how minimum sufficient coverages are determined for each value of *p* (we use *minReadNumber* = 5 here). The diversity index values for the three selected MAGs are highlighted on each row of the histogram as vertical lines. Although it is shown in Figure 3, we have omitted $p = 0.15\%$ from this figure because its minimum sufficient coverage is $3333.\overline{33}\times$; the only MAG that is sufficiently covered for this value of *p* is CAMP.

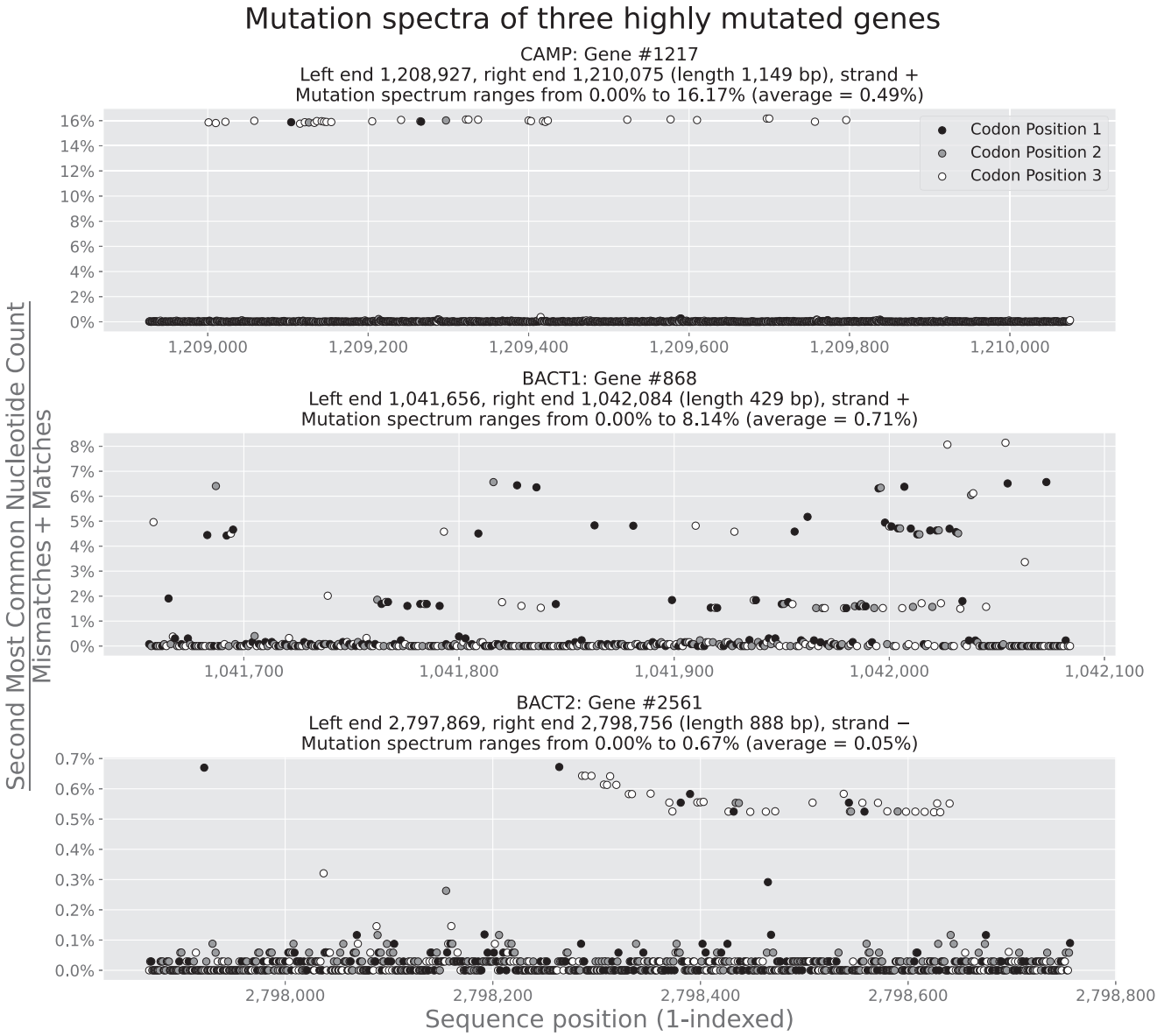

Figure 5. Mutation spectra of highly mutated genes in the three selected MAGs. Each plot shows *freq*(*pos*) for every position *pos* within the selected gene; each of these genes has the highest mutation rate in its parent MAG (for NaiveFreq mutation calls at $p = 0.5\%$) and is thus described in the corresponding MAG's first row in Supplemental Table S2. We exclude deletions from the denominator of *freq*(*pos*) (similar to most of the other analyses in this paper), which is responsible for some of the "jumps" in the *middle* plot (gene 868 in BACT1). Positions are colored by their codon position within the gene, using colors matching those in Figure 3. These figures are analogous to other plots of variation along a gene sequence in the literature, for example Figure 1 in the work by Vasileiadis et al. (2012). Supplemental Figure S14 in the Supplemental Material, "Hotspot genes in the three selected MAGs," visualizes the coverages of each of these genes.

CAMP, BACT1, and BACT2; this visualization shows that there exist clear "coldspot" gaps between mutations, noticeably in BACT1 (Supplemental Table S3). The "strainFlye spot cold-gaps" command can identify these gaps and compute the probability of the largest gap in a contig being of at least a certain length (Supplemental Material, "Investigating coldspots").

Growth dynamics of a metagenome

Because short-read sequencing rarely results in the assembly of complete MAGs, methods for analyzing microbes' growth dynamics using short-read sequencing data (Korem et al. 2015; Joseph et al. 2022) typically rely on reference databases. By improving initial MAG completeness, deep HiFi sequencing provides us the ability to analyze MAGs' growth dynamics completely de novo; the Supplemental Material, "Growth dynamics" (Supplemental Fig. S17), illustrates that this analysis represents yet another benefit of "complete metagenomics."

Phasing identified mutations

The ability to call many mutations with controlled FDR in a contig is essential for phasing the strain-level haplotypes represented in this contig, because haplotype phasing relies on the presence of

reads that span multiple mutations (Nicholls et al. 2021). Fortunately, long and accurate HiFi reads simplify the process of phasing detected mutations and thus performing strain separation.

To reveal the conserved and diverged regions in various strains of a contig, the "strainFlye smooth" module attempts to construct the de Bruijn graph (Compeau et al. 2011) of these strains, not unlike the de Bruijn graphs that are used for analyzing variations among various human genomes (Iqbal et al. 2012). Because errors in reads make this graph very complex, the first command in this module, "strainFlye smooth create," converts reads aligned to a given contig to "smoothed reads," which match the assembled contig at all positions except for those positions in the read aligned to identified mutations (Methods). In addition to smoothed reads, "strainFlye smooth create" also constructs "virtual reads" to fill in low-coverage regions in contigs and maintain contiguity (Methods). The next command, "strainFlye smooth assemble," uses LJA (Bankevich et al. 2022) to construct the multiplex de Bruijn graph of the collection of smoothed and virtual reads for each contig (Methods).

Figure 6 and Supplemental Figure S21 show visualizations of the multiplex de Bruijn graphs produced by LJA given the smoothed and virtual reads produced for each of the three selected MAGs. Ideally, these smoothed reads would have been assembled perfectly, such that one isolated edge would exist for each unique haplotype. The graphs shown here do not reach this ideal, but they come close: Many regions of these MAGs are now represented as linear sequences of "bubbles" representing the variation between haplotypes at a given position in the MAG.

In addition to the "smooth" module, strainFlye also provides the "link" module for strain analysis. This module constructs a *link graph* representing the alleles at each mutated position in a contig, and the frequencies with which these alleles are linked by reads. These graphs, described in the Supplemental Material, "The link graph structure for haplotype visualization" (Supplemental Figs. S18, S19), can be helpful as a visualization tool of what regions of a contig can be phased from the available data.

Discussion

We have presented strainFlye, a pipeline for the identification and analysis of rare mutations within MAGs sequenced using HiFi reads. We showed variations in the frequencies of mutations between CPs and mutation types within MAGs, variations in the numbers of mutations across MAGs in a data set, the identification of hotspots and coldspots of mutations, and the ability of HiFi reads to link mutated positions throughout large portions of MAGs.

These advances set the stage for further improvements in identifying mutations and phasing using complementary linked read technologies, such as reads generated using Hi-C (Burton et al. 2014) or TELL-seq technology (Chen et al. 2020). Complementary short-read sequencing technologies, as well, hold promise for further improving our confidence in the identification of rare mutations (Bickhart et al. 2022; Mc Cartney et al. 2022).

We note that the methodologies developed here could have uses not only in isolation but also as extensions to existing tools for variant calling and metagenomic data analysis. Recent work on optimizing LoFreq in the context of high-coverage viral data sets (Kille et al. 2021) has highlighted the potential for the community to continue to refine existing methods to handle the ever-increasing challenges associated with new types of data.

Although HiFi reads simplify many tasks in metagenomic data analysis, there remain countless open problems. One such problem is the study of lethal positions: A bacterial gene is classified as *lethal* (*nonlethal*) if its deletion allows (does not allow) an organism to survive under some defined conditions (Winzeler et al.

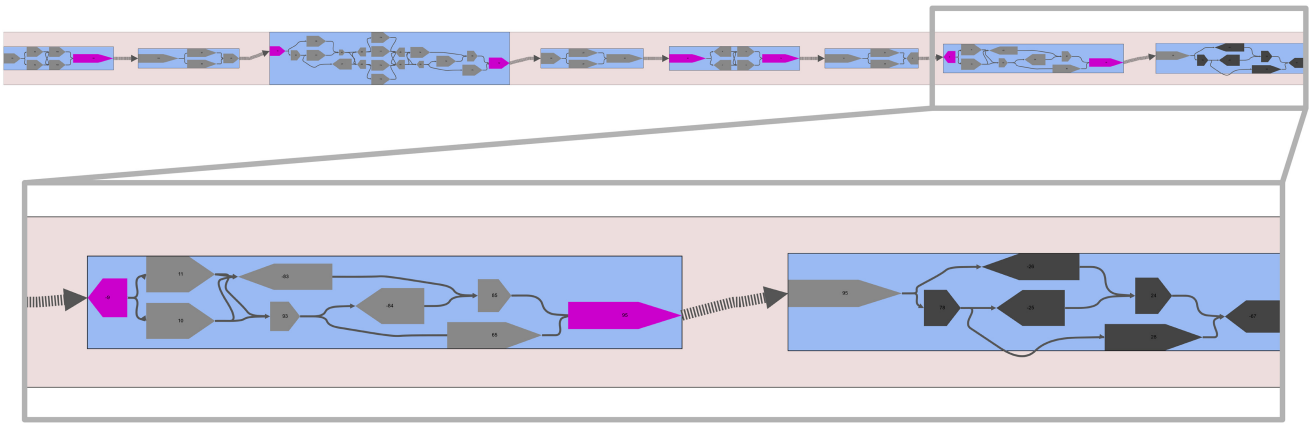

Figure 6. Multiplex de Bruijn graph produced by LJA for CAMP's smoothed and virtual reads. We generated smoothed reads based on mutations identified by NaiveFreq at $p = 1\%$. This is a MetagenomeScope visualization (Fedarko et al. 2017) of the GFA file produced by LJA. Gray pentagons (nodes in this visualization) correspond to "segments" described in this GFA file, which in turn correspond to edges in the de Bruijn graph. Blue regions of the graph indicate "bubble" patterns MetagenomeScope identified in the graph (Miller et al. 2010), highlighted for clarity. Segments colored in pink represent segments that are shared between multiple bubbles: MetagenomeScope duplicates segments (creating a link between the two copies of a duplicate segment) in order to simplify the visualization of adjacent bubbles in the graph. This visualization makes clear that the entire de Bruijn graph can be represented as a linear sequence of bubbles. This topology is consistent with the expected structure of an assembly graph of multiple strains of a genome, for example, as shown in Figure 4 of the work of Kolmogorov et al. (2020); the branching paths in these bubbles likely represent strain-level diversity. The *rightmost* two bubbles in the graph are shown up in a box *below* the main drawing of the graph. The five dark-colored segments highlighted in the *rightmost* bubble correspond to the segments that overlap gene 1217, the most mutated gene in CAMP (Fig. 5, top). There exist three paths through these segments' bubbles: the *top* two paths correspond to the "reference" haplotype of gene 1217 and the *bottom* two paths correspond to the "alternate" haplotype of gene 1217 (Supplemental Material, "Haplotypes of the most mutated gene in CAMP"; Supplemental Fig. S20). The reason for the reference haplotype being represented by two distinct paths is that these paths cover other mutated positions located earlier in CAMP.

1999). Although identification of all lethal SNVs in a bacterial genome provides much more information than identification of all *lethal genes* (Xu et al. 2014), finding these mutations remains an open problem. Moreover, even the simpler problem of identifying a large set of nonlethal mutations in a bacterial genome remains an open problem. Information about such sets is valuable because it can inform various studies, for example, analyses of positions in a genome that lead to antibiotic resistance (Leyn et al. 2021).

We have shown the possibility of revealing tens of thousands of putatively nonlethal mutations in multiple MAGs in SheepGut (Fig. 2) using deep metagenomic sequencing. Even though this example represents—for any of these MAGs alone, to our knowledge—the largest collection of rare mutations in a bacterial genome reported to date, the true extent of diversity in a bacterial community (when the frequency threshold is reduced and the coverage increases by an order of magnitude) remains unknown. Future "ultradeep" sequencing projects (e.g., with coverages as large as 100,000×) may shed light on this question and narrow the set of lethal mutations in bacterial genomes.

Methods

Automatically selecting a decoy contig

In the SheepGut data set, CAMP serves as a long high-coverage decoy contig with relatively few rare mutations (Figs. 3, 4). However, there is no guarantee that such a contig will exist in an arbitrary data set. The "strainFlye fdr estimate" command allows the user to automatically select (or to specify) a decoy contig.

To automatically select a decoy contig, strainFlye first identifies the set of all long high-coverage contigs (Results subsection "Demonstrating strainFlye"). If this set is empty, strainFlye will raise an error explaining the situation to the user; this situation implies that the user will need to lower the *minLength* and *minCov* thresholds in order to find a decoy contig. If this set consists of a single contig, strainFlye will select it as the decoy.

If there are n long high-coverage contigs, strainFlye makes use of diversity index information computed for a set D of various values of the frequency threshold p (Results subsection "Diversity indices"). Each entry in D implies an n-dimensional vector of diversity indices computed for this threshold value (with one entry for each long high-coverage contig). Because the diversity index can be undefined for low-coverage contigs, depending on the minimum sufficient coverage for the corresponding value of p (Fig. 4; Supplemental Material, "Diversity index details"), strainFlye focuses only on the $D_{good} \subseteq D$ threshold values for which at least two long high-coverage contigs have defined diversity indices. If $|D_{good}| = 0$, that is, no vector of diversity indices has at least two long high-coverage contigs with defined diversity indices, strainFlye raises an error explaining the situation to the user.

If $|D_{good}| \geq 1$, then, for each of the corresponding "good" vectors of diversity indices, strainFlye finds the minimum and maximum diversity index across the long high-coverage contigs. It then assigns each of these contigs a score in the range [0, 1] using linear interpolation: The contig with the lowest diversity index in this vector is assigned a score of zero, the contig with the highest diversity index in this vector is assigned a score of one, and all other contigs with defined diversity indices are scaled in between. Contigs in this vector with an undefined diversity index are assigned a score of one in order to penalize them for not being sufficiently covered.

Finally, strainFlye computes a total score for each long high-coverage contig by summing its scores for each of the $|D_{good}|$ diversity index vectors. The contig with the lowest total score is selected as the decoy contig, breaking ties arbitrarily.

We note that there is no guarantee that any of the long high-coverage contigs will have a "low" amount of rare mutations; it may be the case that all of these contigs have high diversity indices. In this case, the automatic selection process will select a decoy contig with a relatively high mutation rate, and strainFlye will thus produce inflated FDR estimates. This is an inherent downside of the TDA; however, we expect that the rising sizes of sequencing data sets will increase the likelihood of suitable decoy contigs existing.

Accounting for indisputable mutations

We classify a p-mutation as *indisputable* if its mutation rate $freq(pos)$ (defined in the Results subsection "Computing mutation spectra") is greater than or equal to a specified *highFrequency* threshold (default value 5%) and as a *rare* mutation otherwise. Although the reliable identification of rare mutations (with frequencies below the error rate in reads) is a challenging task, nearly all indisputable mutations are real, especially in high-coverage MAGs like those studied in this paper. When computing the mutation rates of a decoy or target contig G (with M_G mutations and length L_G), as discussed in the Results subsection "The target-decoy approach for estimating the FDR of identified mutations," we thus redefine $rate_G = \frac{M_{rare}}{3L_G}$, where M_{rare} is the number of rare mutations in the contig. This modification can lower the computed mutation rates for a given contig and thus result in more realistic FDR estimates.

Fixing the estimated FDR of identified rare mutations in a target contig

We have described how to use the TDA to estimate the FDR of a set of identified rare mutations in a contig and to draw an FDR curve for this contig. Given a user-configurable upper bound f on the estimated FDR of rare mutation identifications (e.g., $f = 1\%$, as shown in Fig. 2), the "strainFlye fdr fix" command can fix the estimated FDR for a target contig to be $\leq f$ as follows.

First, we select the lowest value of p at which the estimated FDR for this contig is $\leq f$. Because our FDR estimates do not necessarily increase monotonically as p decreases (Fig. 2), the vertical line implied by fixing the FDR to a given upper bound may be crossed by a target contig's FDR curve multiple times (Käll et al. 2008). However, the number of mutations per megabase produced by a given p value in the target contig does increase monotonically as p decreases. Because of this, the lowest threshold value yielding an acceptable estimated FDR corresponds to the largest set of rare mutations for this target contig with an acceptable estimated FDR.

Because we have assumed that all "indisputable" mutations (Methods) are correct, strainFlye outputs—for each target contig—the set of all rare mutations supported by the "optimal" value of p selected, as well as the set of all indisputable mutations. strainFlye also outputs the set of all indisputable mutations within the decoy contig, although it does not output any rare mutations from the decoy contig (as of writing, this is performed regardless of the decoy context used). These mutations can be used as the starting point for downstream analyses (e.g., phasing).

We note that, for p-mutations, the level of granularity used in varying p can have an impact on the selection of this "optimal" value. strainFlye's pipeline adjusts p in increments of 0.01% (Fig. 2; Supplemental Fig. S8), which we expect should be sufficient for most current data sets.

Predicting protein-coding genes in contigs

Many of the described "decoy contexts" in the Results subsection "Context-dependent TDA," as well as many of the other shown analyses, rely on the availability of gene predictions: We compute these using Prodigal (Hyatt et al. 2010). Notably, we use Prodigal's -c option in order to disallow the prediction of incomplete genes that run off the end of a contig; this restriction simplifies these analyses. For the three selected SheepGut MAGs shown in this paper, we ran Prodigal in its "normal" or "single" mode (processing one contig at a time): This matches the behavior of "strainFlye fdr estimate" when the decoy context(s) requested by a user requires gene predictions in the decoy contig.

We note that our use of Prodigal makes the implicit assumption that the contigs on which we run it are prokaryotic. It should be feasible to extend our analyses to work with alternative gene prediction tools if desired, although for now we have focused our efforts on analyzing prokaryotic contigs.

Constructing smoothed reads

Given a set of identified mutations in a contig, and a read aligned to this contig, we identify all mutated positions in the contig to which a linear alignment of this read has (mis)match operations. We then convert each linear alignment of this read to the contig into a "smoothed read" that completely matches the contig, with the exception of the read's nucleotides at all identified mutated positions spanned by the alignment.

This process involves some simplifying assumptions: For one, it necessarily ignores both indels and non-"identified" single-nucleotide mutations. If a read's nucleotide at any of the identified mutated positions to which it has a (mis)match operation does not match the "reference" or "alternate" nucleotide at this position (in the case of mutations identified by NaiveFreq, these two will always correspond to the first- and second-most common nucleotide at a position), then we discard this alignment of this read and do not generate a smoothed read from it. This operation implicitly ignores the contig's "reference" nucleotide at this mutated position, so there is the (unlikely) possibility for the reference nucleotide at an "unreasonable" position (where the reference and consensus nucleotide disagree) (for discussion, see Supplemental Material, "Hotspot genes in the three selected MAGs") to be completely unrepresented in the smoothed reads if it is not the "reference" or "alternate" nucleotide at a mutated position. Similarly, if a read spans a mutated position with a deletion, we discard this alignment of this read.

We note that these "discarding" steps are only applied to individual linear alignments of a read at a time; for example, a read with two linear alignments (one "representative," one supplementary) (https://samtools.github.io/hts-specs/SAMv1.pdf) to a contig could result in the creation of zero, one, or two smoothed reads for this contig. The input alignment file should not contain any secondary alignments or overlapping supplementary alignments on a contig (Supplemental Material, "Read alignment") so, even if a single read is converted to multiple smoothed reads in a contig, these smoothed reads should all cover disjoint regions within the contig.

This can lead to the generation of smaller haplotype assembly graphs than may be expected, because we effectively ignore haplotypes that disagree with any of the identified mutations in a contig. This is a likely factor in why the BACT1 assembly graph (shown in Supplemental Fig. S21 in Supplemental Material, "Smoothed haplotype assembly graphs") is simple compared with the CAMP and BACT2 assembly graphs: Similar to how BACT1 has an order of magnitude more $p = 1\%$ mutations (22,415) than CAMP (83) and BACT2 (380), BACT1 has an order of magnitude more $p = 1\%$ mutations with at least one deletion in their pileup (8269/22,415) compared with CAMP (57/83) and BACT2 (274/380). This results in a comparatively large number of reads being discarded when attempting to generate smoothed haplotypes for BACT1.

Finally, we note that this process also splits supplementary alignments of a single read into distinct smoothed reads because it is unclear how to produce a single smoothed read from a read that has multiple alignments to a contig. However, because of the filtering of overlapping supplemental alignments described in the Supplemental Material, "Read alignment," these distinct smoothed reads should not overlap with each other on a contig.

Constructing virtual reads

Assembly and/or alignment artifacts can lead to certain positions in a MAG having relatively low coverage. These coverage drops—for example, the drops shown in the Supplemental Material, "Coverages and deletion-rich positions"—complicate the assembly of haplotypes of these nonetheless already-assembled MAGs. Positions without any coverage at all will split the resulting de Bruijn graph, and positions with low coverage can still result in discontiguous assemblies owing to assemblers treating these regions' corresponding smoothed reads as erroneous. To address this problem, we create virtual reads that can span uncovered or low-coverage positions in a MAG. We note that the term "virtual reads" is used in a similar context in the work by Bankevich et al. (2022).

Consider a MAG with average coverage C_m, defining the coverage at each position only based on the number of matching and mismatching reads in the input alignment (ignoring deletions). We define a position with coverage (based only on smoothed reads) C_p in this MAG as *low-coverage* if $C_p < minWellCovFrac \cdot C_m$, where *minWellCovFrac* is a percentage in the range [0%, 100%] (default value 95%). We chose this default value based on testing LJA with the smoothed and virtual reads constructed from CAMP until the resulting assembly became contiguous, as is shown in Figure 6. To construct virtual reads, we identify all "runs" of consecutive low-coverage positions in the MAG. For each of these runs, we define its average coverage as C_r. We then create a virtual read that matches the MAG "reference" sequence at all positions throughout this run, as well as *vrFlank* (default value 100) positions before and after the start and end of the run (clamping to the start or end of the MAG as needed, in case the run is close to a boundary of the MAG). Finally, we generate $round(C_m - C_r)$ copies of this virtual read in order to "lift" the coverage throughout this run of low-coverage positions to roughly match C_m.

Like the construction of smoothed reads, the construction of virtual reads also involves making some concessions; for example, in the case in which an identified mutation occurs within a virtual read or its flanking *vrFlank* positions, we simply set the virtual read's nucleotide at this mutated position equal to the MAG reference. Or, in strange circumstances, the flanking positions in the virtual reads created for a low-coverage region might overlap the virtual reads generated for other nearby low-coverage regions. These artifacts may cause undesirable effects in the haplotype assembly process. We have nonetheless found that using virtual reads, in addition to just smoothed reads, helps improve the contiguity of the multiplex de Bruijn graphs produced by LJA.

Lastly, we note that we do not construct virtual reads that connect the end and start of a MAG, even if this MAG is represented in the original assembly graph as a single circular contig (e.g., as BACT1 and BACT2 are). Although this can have the effect of linearizing circular contigs, we expect that this should not make a large difference in downstream analyses using the resulting smoothed haplotypes.

Assembling smoothed and virtual reads

After constructing smoothed and virtual reads, the "strainFlye smooth assemble" command provides these reads as input to LJA, an assembler designed for HiFi reads (Bankevich et al. 2022). We run LJA without its error correction (mowerDBG) step; instead, we apply a simple *k*-mer coverage filter that removes very-low-coverage edges from the initial de Bruijn graph generated by LJA's jumboDBG tool. This method of running LJA is compared against other assembly methods and discussed in detail in the Supplemental Material, "Haplotypes of the most mutated gene in CAMP."

Data sets

The SheepGut read-set is available at the NCBI BioProject database (https://www.ncbi.nlm.nih.gov/bioproject/) under accession number PRJNA595610. We used the HiFi sequencing data from accession IDs SRX7628648 and SRX10647529, in particular. We note that although there exists additional Hi-C and Illumina short-read sequencing data for SheepGut (Bickhart et al. 2022), we have only made use of the HiFi data (from the two accession IDs given above) in this paper.

To simplify reproduction of our analyses, we have also made the metaFlye assembly graph produced for the SheepGut data set (Supplemental Material, "Assembly graph"), which represents a starting point for the analyses shown in this paper, available on Zenodo (https://doi.org/10.5281/zenodo.6545141).

The chicken gut metagenome read-set is available at the NCBI Sequence Read Archive (SRA; https://www.ncbi.nlm.nih.gov/sra) under accession number SRR15214153. We retrieved the hifiasm-meta (Feng et al. 2022) assembly of this data set produced by Feng et al. (2022) from Zenodo (https://doi.org/10.5281/zenodo.6330282).

Software dependencies

The strainFlye pipeline is implemented as a Python 3 command-line tool. The pipeline code directly relies on pysam (https://github.com/pysam-developers/pysam), pysamstats (https://github.com/alimanfoo/pysamstats), scikit-bio (http://scikit-bio.org), NetworkX (Hagberg et al. 2008), pandas (McKinney 2010; https://doi.org/10.5281/zenodo.4572994), click (https://click.palletsprojects.com/), NumPy (Harris et al. 2020), SciPy (Virtanen et al. 2020), SAMtools (Li et al. 2009; Danecek et al. 2021), BCFtools (Danecek et al. 2021), minimap2 (Li 2018), Prodigal (Hyatt et al. 2010), and LJA (Bankevich et al. 2022).

The analyses shown throughout this paper (including Bash scripts, Python 3 scripts, and Jupyter notebooks) (Kluyver et al. 2016) additionally make use of metaFlye (Kolmogorov et al. 2020), CheckM (Parks et al. 2015), barrnap (T Seemann, https://github.com/tseemann/barrnap), the SRA toolkit (https://github.com/ncbi/sra-tools), LoFreq (Wilm et al. 2012), the Integrative Genomics Viewer (IGV) (Thorvaldsdóttir et al. 2013), Bandage (Wick et al. 2015), MetagenomeScope (Fedarko et al. 2017), Jupyter (Kluyver et al. 2016), nbconvert (https://nbconvert.readthedocs.io), scikit-learn (Pedregosa et al. 2011), Matplotlib (Hunter 2007), Logomaker (Tareen and Kinney 2020), and the neato (Gansner et al. 2004) and sfdp (Hu 2005) layout methods in Graphviz (Gansner and North 2000). We installed software primarily using conda (https://conda.io), mamba (https://mamba.readthedocs.io), and pip (https://pip.pypa.io).

Finally, we note that we produced Figure 1 using LibreOffice Draw (https://www.libreoffice.org/discover/draw/) and GIMP (https://www.gimp.org/). We produced the example link graphs shown within this figure, in particular, using Graphviz' (Gansner and North 2000) neato (Gansner et al. 2004) layout method. We also modified the MetagenomeScope visualization of a multiplex de Bruijn graph shown in Figure 6 using LibreOffice Draw in order to add a box highlighting a region of the graph.

Software availability

strainFlye's code is available on GitHub (https://github.com/fedarko/strainFlye) and as Supplemental Code S1. Additional ad hoc code for performing the analyses shown in this paper is available on GitHub (https://github.com/fedarko/sheepgut) and as Supplemental Code S2.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

We thank the editor and anonymous reviewers for their feedback. We thank Andrey Bzikadze for initial exploration of the TDA and context-dependent TDA in the case of short-read sequencing. We thank Nuno Bandeira for detailed feedback, including bringing the possibility of pseudogenes impacting these analyses to our attention. We thank Anton Bankevich for help with applying LJA (Bankevich et al. 2022) to haplotype assembly. We acknowledge Nicholas Bokulich for bringing Figure 1 of Vasileiadis et al. (2012) to our attention: https://forum.qiime2.org/t/9061/13. We thank Lee Katz and Kai Blin for advice on parsing gene prediction data. We thank Heng Li for answering questions about minimap2. We thank Andrei Osterman and Semen Leyn for helpful discussions and advice on analyzing the SheepGut data set, including alignment filtering and interpretation of variation in CAMP. M.W.F. thanks Alex Richter for bringing the K_a and K_s values to his attention. The rainbow colorscheme used in Figure 2 was partially inspired by Figure 2 of Elias and Gygi (2007). This work was partly supported by IBM Research AI through the AI Horizons Network and the UC San Diego Center for Microbiome Innovation (to M.W.F.); and by National Institutes of Health Common Fund Award, National Institute of Diabetes and Digestive and Kidney Diseases, U24DK131617-01.

References

Andor N, Graham TA, Jansen M, Xia LC, Aktipis CA, Petritsch C, Ji HP, Maley CC. 2016. Pan-cancer analysis of the extent and consequences of intratumor heterogeneity. *Nat Med* **22:** 105–113. doi:10.1038/nm.3984

Bankevich A, Bzikadze AV, Kolmogorov M, Antipov D, Pevzner PA. 2022. Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads. *Nat Biotechnol* **40:** 1075–1081. doi:10.1038/s41587-022-01220-6

Barrick JE, Yu DS, Yoon SH, Jeong H, Oh TK, Schneider D, Lenski RE, Kim JF. 2009. Genome evolution and adaptation in a long-term experiment with *Escherichia coli*. *Nature* **461:** 1243–1247. doi:10.1038/nature08480

Bertrand D, Shaw J, Kalathiyappan M, Ng AHQ, Kumar MS, Li C, Dvornicic M, Soldo JP, Koh JY, Tong C, et al. 2019. Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. *Nat Biotechnol* **37:** 937–944. doi:10.1038/s41587-019-0191-2

Bickhart DM, Kolmogorov M, Tseng E, Portik DM, Korobeynikov A, Tolstoganov I, Uritskiy G, Liachko I, Sullivan ST, Shin SB, et al. 2022. Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. *Nat Biotechnol* **40:** 711–719. doi:10.1038/s41587-021-01130-z

Bofkin L, Goldman N. 2007. Variation in evolutionary processes at different codon positions. *Mol Biol Evol* **24:** 513–521. doi:10.1093/molbev/msl178

Burton JN, Liachko I, Dunham MJ, Shendure J. 2014. Species-level deconvolution of metagenome assemblies with Hi-C–based contact probability maps. *G3 (Bethesda)* **4:** 1339–1346. doi:10.1534/g3.114.011825

Chen Z, Pham L, Wu T-C, Mo G, Xia Y, Chang PL, Porter D, Phan T, Che H, Tran H, et al. 2020. Ultralow-input single-tube linked-read library method enables short-read second-generation sequencing systems to routinely generate highly accurate and economical long-range sequencing information. *Genome Res* **30:** 898–909. doi:10.1101/gr.260380.119

Cleary B, Brito IL, Huang K, Gevers D, Shea T, Young S, Alm EJ. 2015. Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. *Nat Biotechnol* **33:** 1053–1060. doi:10.1038/nbt.3329

Compeau PE, Pevzner PA, Tesler G. 2011. How to apply de Bruijn graphs to genome assembly. *Nat Biotechnol* **29:** 987–991. doi:10.1038/nbt.2023

Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, et al. 2021. Twelve years of SAMtools and BCFtools. *GigaScience* **10:** giab008. doi:10.1093/gigascience/giab008

Dayhoff MO, Schwartz RM, Orcutt BC. 1978. Chap 22. A model of evolutionary change in proteins. In *Atlas of protein sequence and structure* (ed. Dayhoff MO), Vol. 5, pp. 345–352. National Biomedical Research Foundation, Washington, DC.

Dierckxsens N, Li T, Vermeesch JR, Xie Z. 2021. A benchmark of structural variation detection by long reads through a realistic simulated model. *Genome Biol* **22:** 342. doi:10.1186/s13059-021-02551-4

Elias JE, Gygi SP. 2007. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat Methods* **4:** 207–214. doi:10.1038/nmeth1019

Elias JE, Haas W, Faherty BK, Gygi SP. 2005. Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. *Nat Methods* **2:** 667–675. doi:10.1038/nmeth785

Emery K, Hasam S, Noble WS, Keich U. 2020. Multiple competition-based FDR control and its application to peptide detection. In *International Conference on Research in Computational Molecular Biology*, pp. 54–71. Springer, Padua, Italy.

Eng JK, McCormack AL, Yates JR. 1994. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J Am Soc Mass Spectrom* **5:** 976–989. doi:10.1016/1044-0305(94)80016-2

Fan HC, Blumenfeld YJ, Chitkara U, Hudgins L, Quake SR. 2008. Noninvasive diagnosis of fetal aneuploidy by shotgun sequencing DNA from maternal blood. *Proc Natl Acad Sci* **105:** 16266–16271. doi:10.1073/pnas.0808319105

Fedarko M, Ghurye J, Treangen T, Pop M. 2017. MetagenomeScope: web-based hierarchical visualization of metagenome assembly graphs. In *Proceedings of the 25th International Symposium on Graph Drawing and Network Visualization*, Vol. 10692, pp. 630–632. Springer, Boston.

Feng X, Cheng H, Portik D, Li H. 2022. Metagenome assembly of high-fidelity long reads with hifiasm-meta. *Nat Methods* **19:** 671–674. doi:10.1038/s41592-022-01478-3

Frank C, Werber D, Cramer JP, Askar M, Faber M, an der Heiden M, Bernard H, Fruth A, Prager R, Spode A, et al. 2011. Epidemic profile of Shiga-toxin–producing *Escherichia coli* O104:H4 outbreak in Germany. *N Engl J Med* **365:** 1771–1780. doi:10.1056/NEJMoa1106483

Fukasawa Y, Ermini L, Wang H, Carty K, Cheung M-S. 2020. LongQC: a quality control tool for third generation sequencing long read data. *G3 (Bethesda)* **10:** 1193–1196. doi:10.1534/g3.119.400864

Gansner ER, North SC. 2000. An open graph visualization system and its applications to software engineering. *Softw Pract Exp* **30:** 1203–1233. doi:10.1002/1097-024X(200009)30:11<1203::AID-SPE338>3.0.CO;2-N

Gansner ER, Koren Y, North S. 2004. Graph drawing by stress majorization. In *International Symposium on Graph Drawing*, pp. 239–250. Springer, New York.

Gerlinger M, Rowan AJ, Horswell S, Larkin J, Endesfelder D, Gronroos E, Martinez P, Matthews N, Stewart A, Tarpey P, et al. 2012. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *N Engl J Med* **366:** 883–892. doi:10.1056/NEJMoa1113205

Gupta N, Bandeira N, Keich U, Pevzner PA. 2011. Target-decoy approach and false discovery rate: when things may go wrong. *J Am Soc Mass Spectrom* **22:** 1111–1120. doi:10.1007/s13361-011-0139-3

Hagberg AA, Schult DA, Swart PJ. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the Seventh Python in Science Conference*, Pasadena, CA (ed. Varoquaux G, et al.), pp. 11–15.

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, et al. 2020. Array programming with NumPy. *Nature* **585:** 357–362. doi:10.1038/s41586-020-2649-2

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci* **89:** 10915–10919. doi:10.1073/pnas.89.22.10915

Henn MR, Boutwell CL, Charlebois P, Lennon NJ, Power KA, Macalalad AR, Berlin AM, Malboeuf CM, Ryan EM, Gnerre S, et al. 2012. Whole genome deep sequencing of HIV-1 reveals the impact of early minor variants upon immune recognition during acute infection. *PLoS Pathog* **8:** e1002529. doi:10.1371/journal.ppat.1002529

Hu Y. 2005. Efficient, high-quality force-directed graph drawing. *Mathematica J* **10:** 37–71.

Hunter JD. 2007. Matplotlib: a 2D graphics environment. *Comput Sci Eng* **9:** 90–95. doi:10.1109/MCSE.2007.55

Hyatt D, Chen G-L, LoCascio PF, Land ML, Larimer FW, Hauser LJ. 2010. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* **11:** 119. doi:10.1186/1471-2105-11-119

Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. 2012. *De novo* assembly and genotyping of variants using colored de Bruijn graphs. *Nat Genet* **44:** 226–232. doi:10.1038/ng.1028

Jäger AC, Alvarez ML, Davis CP, Guzmán E, Han Y, Way L, Walichiewicz P, Silva D, Pham N, Caves G, et al. 2017. Developmental validation of the MiSeq FGx forensic genomics system for targeted next generation sequencing in forensic DNA casework and database laboratories. *Forensic Sci Int Genet* **28:** 52–70. doi:10.1016/j.fsigen.2017.01.011

Jaiswal S, Fontanillas P, Flannick J, Manning A, Grauman PV, Mar BG, Lindsley RC, Mermel CH, Burtt N, Chavez A, et al. 2014. Age-related clonal hematopoiesis associated with adverse outcomes. *N Engl J Med* **371:** 2488–2498. doi:10.1056/NEJMoa1408617

Joseph TA, Chlenski P, Litman A, Korem T, Pe'er I. 2022. Accurate and robust inference of microbial growth dynamics from metagenomic sequencing reveals personalized growth rates. *Genome Res* **32:** 558–568. doi:10.1101/gr.275533.121

Käll L, Storey JD, MacCoss MJ, Noble WS. 2008. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J Proteome Res* **7:** 29–34. doi:10.1021/pr700600n

Keich U, Kertesz-Farkas A, Noble WS. 2015. Improved false discovery rate estimation procedure for shotgun proteomics. *J Proteome Res* **14:** 3148–3161. doi:10.1021/acs.jproteome.5b00081

Keich U, Tamura K, Noble WS. 2019. Averaging strategy to reduce variability in target-decoy estimates of false discovery rate. *J Proteome Res* **18:** 585–593. doi:10.1021/acs.jproteome.8b00802

Kille B, Liu Y, Sapoval N, Nute M, Rauchwerger L, Amato N, Treangen TJ. 2021. Accelerating SARS-CoV-2 low frequency variant calling on ultra deep sequencing datasets. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Portland, OR, pp. 204–208. IEEE.

Kim CY, Ma J, Lee I. 2022. HiFi metagenomic sequencing enables assembly of accurate and complete genomes from human gut microbiota. *Nat Commun* **13:** 6367. doi:10.1038/s41467-022-34149-0

Kluyver T, Ragan-Kelley B, Pérez F, Granger BE, Bussonnier M, Frederic J, Kelley K, Hamrick JB, Grout J, Corlay S, et al. 2016. Jupyter Notebooks: a publishing format for reproducible computational workflows. In *20th Conference on Electronic Publishing (Elpub)*, Göttingen, Germany, Vol. 2016. IOS Press, Amsterdam.

Kolmogorov M, Yuan J, Lin Y, Pevzner PA. 2019. Assembly of long, error-prone reads using repeat graphs. *Nat Biotechnol* **37:** 540–546. doi:10.1038/s41587-019-0072-8

Kolmogorov M, Bickhart DM, Behsaz B, Gurevich A, Rayko M, Shin SB, Kuhn K, Yuan J, Polevikov E, Smith TP, et al. 2020. metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nat Methods* **17:** 1103–1110. doi:10.1038/s41592-020-00971-x

Korem T, Zeevi D, Suez J, Weinberger A, Avnit-Sagi T, Pompan-Lotan M, Matot E, Jona G, Harmelin A, Cohen N, et al. 2015. Growth dynamics of gut microbiota in health and disease inferred from single metagenomic samples. *Science* **349:** 1101–1106. doi:10.1126/science.aac4812

Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. 2017. Canu: scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation. *Genome Res* **27:** 722–736. doi:10.1101/gr.215087.116

Leyn SA, Zlamal JE, Kurnasov OV, Li X, Elane M, Myjak L, Godzik M, de Crecy A, Garcia-Alcalde F, Ebeling M, et al. 2021. Experimental evolution in morbidostat reveals converging genomic trajectories on the path to triclosan resistance. *Microbial Genomics* **7:** 000553. doi:10.1099/mgen.0.000553

Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34:** 3094–3100. doi:10.1093/bioinformatics/bty191

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25:** 2078–2079. doi:10.1093/bioinformatics/btp352

Luo C, Knight R, Siljander H, Knip M, Xavier RJ, Gevers D. 2015. ConStrains identifies microbial strains in metagenomic datasets. *Nat Biotechnol* **33:** 1045–1052. doi:10.1038/nbt.3319

Mc Cartney AM, Shafin K, Alonge M, Bzikadze AV, Formenti G, Fungtammasan A, Howe K, Jain C, Koren S, Logsdon GA, et al. 2022. Chasing perfection: validation and polishing strategies for telomere-to-telomere genome assemblies. *Nat Methods* **19:** 687–695. doi:10.1038/s41592-022-01440-3

McKinney W. 2010. Data structures for statistical computing in Python. In *Proceedings of the Ninth Python in Science Conference*, Austin, TX (ed. van der Walt S, et al.), pp. 56–61.

Menzel P, Ng KL, Krogh A. 2016. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nat Commun* **7:** 11257. doi:10.1038/ncomms11257

Miller JR, Koren S, Sutton G. 2010. Assembly algorithms for next-generation sequencing data. *Genomics* **95:** 315–327. doi:10.1016/j.ygeno.2010.03.001

Moore RE, Young MK, Lee TD. 2002. Qscore: an algorithm for evaluating SEQUEST database search results. *J Am Soc Mass Spectrom* **13:** 378–386. doi:10.1016/S1044-0305(02)00352-5

Myers EW. 2021. *HISim*. https://github.com/thegenemyers/HI.SIM.

Nakamura K, Oshima T, Morimoto T, Ikeda S, Yoshikawa H, Shiwa Y, Ishikawa S, Linak MC, Hirai A, Takahashi H, et al. 2011. Sequence-specific error profile of illumina sequencers. *Nucleic Acids Res* **39:** e90–e90. doi:10.1093/nar/gkr344

Nicholls SM, Aubrey W, De Grave K, Schietgat L, Creevey CJ, Clare A. 2021. On the complexity of haplotyping a microbial community. *Bioinformatics* **37:** 1360–1366. doi:10.1093/bioinformatics/btaa977

Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. 2017. metaSPAdes: a new versatile metagenomic assembler. *Genome Res* **27:** 824–834. doi:10.1101/gr.213959.116

Ono Y, Asai K, Hamada M. 2021. PBSIM2: a simulator for long-read sequencers with a novel generative model of quality scores. *Bioinformatics* **37:** 589–595. doi:10.1093/bioinformatics/btaa835

Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. 2015. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* **25:** 1043–1055. doi:10.1101/gr.186072.114

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res* **12:** 2825–2830.

Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, et al. 2018. A universal SNP and small-indel variant caller using deep neural networks. *Nat Biotechnol* **36:** 983–987. doi:10.1038/nbt.4235

Quince C, Delmont TO, Raguideau S, Alneberg J, Darling AE, Collins G, Eren AM. 2017. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biol* **18:** 181. doi:10.1186/s13059-017-1309-9

Richter DC, Ott F, Auch AF, Schmid R, Huson DH. 2008. MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS One* **3:** e3373. doi:10.1371/journal.pone.0003373

Rugbjerg P, Sommer MO. 2019. Overcoming genetic heterogeneity in industrial fermentations. *Nat Biotechnol* **37:** 869–876. doi:10.1038/s41587-019-0171-6

Salk JJ, Schmitt MW, Loeb LA. 2018. Enhancing the accuracy of next-generation sequencing for detecting rare and subclonal mutations. *Nat Rev Genet* **19:** 269–285. doi:10.1038/nrg.2017.117

Sandmann S, De Graaf AO, Karimi M, Van Der Reijden BA, Hellström-Lindberg E, Jansen JH, Dugas M. 2017. Evaluating variant calling tools for non-matched next-generation sequencing data. *Sci Rep* **7:** 43169. doi:10.1038/srep43169

Sapoval N, Aghazadeh A, Nute MG, Antunes DA, Balaji A, Baraniuk R, Barberan CJ, Dannenfelser R, Dun C, Edrisi M, et al. 2022. Current progress and open challenges for applying deep learning across the biosciences. *Nat Commun* **13:** 1728. doi:10.1038/s41467-022-29268-7

Schmitt MW, Kennedy SR, Salk JJ, Fox EJ, Hiatt JB, Loeb LA. 2012. Detection of ultra-rare mutations by next-generation sequencing. *Proc Natl Acad Sci* **109:** 14508–14513. doi:10.1073/pnas.1208715109

Sekowska A, Wendel S, Fischer EC, Nørholm MH, Danchin A. 2016. Generation of mutation hotspots in ageing bacterial colonies. *Sci Rep* **6:** 2. doi:10.1038/s41598-016-0005-4

Sereika M, Kirkegaard RH, Karst SM, Michaelsen TY, Sørensen EA, Wollenberg RD, Albertsen M. 2022. Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nat Methods* **19:** 823–826. doi:10.1038/s41592-022-01539-7

Sonneborn TM. 1965. Degeneracy of the genetic code: extent, nature, and genetic implications. In *Evolving genes and proteins*, pp. 377–397. Academic Press, New York.

Suzuki Y, Myers G. 2022. Accurate *k*-mer classification using read profiles. In *22nd International Workshop on Algorithms in Bioinformatics (WABI 2022): Leibniz international proceedings in informatics (LIPIcs)* (ed. Boucher C, et al.), Vol. 242, pp. 10:1–10:20. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany.

Tareen A, Kinney JB. 2020. Logomaker: beautiful sequence logos in Python. *Bioinformatics* **36:** 2272–2274. doi:10.1093/bioinformatics/btz921

Thorvaldsdóttir H, Robinson JT, Mesirov JP. 2013. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief Bioinformatics* **14:** 178–192. doi:10.1093/bib/bbs017

Toprak E, Veres A, Michel J-B, Chait R, Hartl DL, Kishony R. 2012. Evolutionary paths to antibiotic resistance under dynamically sustained drug selection. *Nat Genet* **44:** 101–105. doi:10.1038/ng.1034

Vasileiadis S, Puglisi E, Arena M, Cappa F, Cocconcelli PS, Trevisan M. 2012. Soil bacterial diversity screening using single 16S rRNA gene V regions coupled with multi-million read generating sequencing technologies. *PLoS One* **7:** e42671. doi:10.1371/journal.pone.0042671

Vicedomini R, Quince C, Darling AE, Chikhi R. 2021. Strainberry: automated strain separation in low-complexity metagenomes using long reads. *Nat Commun* **12:** 4485. doi:10.1038/s41467-021-24515-9

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* **17:** 261–272. doi:10.1038/s41592-019-0686-2

Vogel F. 1972. Non-randomness of base replacement in point mutation. *J Mol Evol* **1:** 334–367. doi:10.1007/BF01653962

Wang B, Wan L, Wang A, Li LM. 2017. An adaptive decorrelation method removes Illumina DNA base-calling errors caused by crosstalk between adjacent clusters. *Sci Rep* **7:** 41348. doi:10.1038/srep41348

Wenger AM, Peluso P, Rowell WJ, Chang P-C, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, et al. 2019. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol* **37:** 1155–1162. doi:10.1038/s41587-019-0217-9

Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: interactive visualization of *de novo* genome assemblies. *Bioinformatics* **31:** 3350–3352. doi:10.1093/bioinformatics/btv383

Wilm A, Aw PPK, Bertrand D, Yeo GHT, Ong SH, Wong CH, Khor CC, Petric R, Hibberd ML, Nagarajan N. 2012. LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Res* **40:** 11189–11201. doi:10.1093/nar/gks918

Winzeler EA, Shoemaker DD, Astromoff A, Liang H, Anderson K, Andre B, Bangham R, Benito R, Boeke JD, Bussey H, et al. 1999. Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science* **285:** 901–906. doi:10.1126/science.285.5429.901

Xu Z, Zikos D, Osterrieder N, Tischer BK. 2014. Generation of a complete single-gene knockout bacterial artificial chromosome library of cowpox virus and identification of its essential genes. *J Virol* **88:** 490–502. doi:10.1128/JVI.02385-13

Zlamal JE, Leyn SA, Iyer M, Elane ML, Wong NA, Wamsley JW, Vercruysse M, Garcia-Alcalde F, Osterman AL. 2021. Shared and unique evolutionary trajectories to ciprofloxacin resistance in gram-negative bacterial pathogens. *mBio* **12:** e0098721. doi:10.1128/mBio.00987-21

Received May 10, 2022; accepted in revised form November 16, 2022.