CAMISIM: Simulating metagenomes and microbial communities

ADRIAN FRITZ[1,*], PETER HOFMANN[1,2,*], STEPHAN MAJDA[1,2], EIK DAHMS[1,2], JOHANNES DRÖGE[1,2], JESSIKA FIEDLER[1,2], TILL R. LESKER[1,3], PETER BELMANN[1,4], MATTHEW Z. DEMAERE[5], AARON E. DARLING[5], ALEXANDER SCZYRBA[4], ANDREAS BREMGES[1,3], AND ALICE C. MCHARDY[1,2,†]

[1] *Computational Biology of Infection Research, Helmholtz Centre for Infection Research, 38124 Braunschweig, Germany*
[2] *Formerly Department of Algorithmic Bioinformatics, Heinrich-Heine University Düsseldorf, 40225 Düsseldorf, Germany*
[3] *German Center for Infection Research (DZIF), partner site Hannover-Braunschweig, 38124 Braunschweig, Germany*
[4] *Center for Biotechnology and Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany*
[5] *The ithree institute, University of Technology Sydney, Sydney, New South Wales, Australia*

[*] *These authors contributed equally to this work.*
[†] *Correspondence: alice.mchardy@helmholtz-hzi.de*

Shotgun metagenome data sets of microbial communities are highly diverse, not only due to the natural variation of the underlying biological systems, but also due to differences in laboratory protocols, replicate numbers, and sequencing technologies. Accordingly, to effectively assess the performance of metagenomic analysis software, a wide range of benchmark data sets are required. Here, we describe the CAMISIM microbial community and metagenome simulator. The software can model different microbial abundance profiles, multi-sample time series and differential abundance studies, includes real and simulated strain-level diversity, and generates second and third generation sequencing data from taxonomic profiles or *de novo*. Gold standards are created for sequence assembly, genome binning, taxonomic binning, and taxonomic profiling. CAMSIM generated the benchmark data sets of the first CAMI challenge. For two simulated multi-sample data sets of the human and mouse gut microbiomes we observed high functional congruence to the real data. As further applications, we investigated the effect of varying evolutionary genome divergence, sequencing depth, and read error profiles on two popular metagenome assemblers, MEGAHIT and metaSPAdes, on several thousand small data sets generated with CAMISIM. CAMISIM can simulate a wide variety of microbial communities and metagenome data sets together with truth standards for method evaluation. All data sets and the software are freely available at: https://github.com/CAMI-challenge/CAMISIM

INTRODUCTION

Extensive 16S rRNA gene amplicon and shotgun metagenome sequencing efforts have been and are being undertaken to catalogue the human microbiome in health and disease [1, 2] and to study microbial communities of medical, pharmaceutical, or biotechnological relevance [3–8]. We have since learned that naturally occurring microbial communities cover a wide range of organismal complexities – with populations ranging from half a dozen to likely tens of thousands of members – can include substantial strain level diversity, and vary widely in represented taxa [9–12]. Analyzing these diverse communities is challenging.

The problem is exacerbated by use of a wide range of experimental setups in data generation and the rapid evolution of short- and long-read sequencing technologies [13, 14]. Owing to the large diversity of generated data, the possibility to generate realistic benchmark data sets for particular experimental setups is essential for assessing computational metagenomics software.

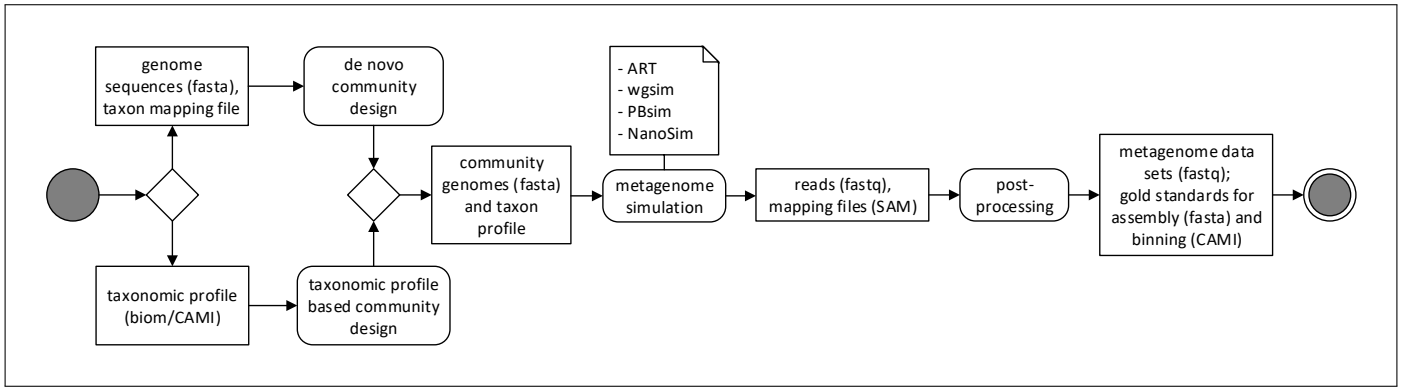

Fig. 1. UML diagram of the CAMISIM workflow. CAMISIM starts with the "Community Design" step, which can either be *de novo*, requiring a taxon mapping file and reference genomes or based on a taxonomic profile. This step produces a community genome and taxon profile which is used for the metagenome simulation using one of currently four read simulators (ART, wgsim, PBsim, NanoSim). The resulting reads and bam-files mapping the reads to the original genomes, are used to create the gold standards before all the files can be anonymized and shuffled in the post-processing step.

CAMI, the initiative for the Critical Assessment of Metagenome Interpretation, is a community effort aiming to generate extensive, objective performance overviews of computational metagenomics software [15]. CAMI organizes benchmarking challenges and encourages the development of standards, and reproducibility in all aspects, such as data generation, software application, and result interpretation [16].

We here describe CAMISIM, which was originally written to generate the simulated metagenome data sets used in the first CAMI challenge. It has since been extended into a versatile and highly modular metagenome simulator. We demonstrate the usability and utility of CAMISIM with several applications. We generated complex, multi-replicate benchmark data sets from taxonomic profiles of human and mouse gut microbiomes [1, 17]. We also simulated thousands of small "minimally challenging metagenomes" to characterize the effect of varying sequencing coverage, evolutionary divergence of genomes, and sequencing error profiles on the popular MEGAHIT [18] and metaSPAdes [19] assemblers.

THE CAMISIM SOFTWARE

CAMISIM allows customization of many properties of the generated communities and data sets, such as the overall number of genomes (community complexity), strain diversity, the community genome abundance distributions, sample sizes, the number of replicates, and sequencing technology used. For setting these options, a configuration file is needed, which is described in the Supplement. Simulation with CAMISIM has 3 stages (Figure 1):

1. design of the community, which includes selection of the community members and their genomes, and assigning them relative abundances,
2. metagenome sequencing data simulation, and
3. postprocessing, where the binning and assembly gold standards are produced.

Community design

In this step, the community genome abundance profiles, called P_{out}, are created. These also represent the gold standard for taxonomic profiling and, from the strain to the superkingdom rank, specify the relative abundances of individual strains (genomes) or their parental taxa in percent. In addition, a genome sequence collection for the strains in P_{out} is generated. Both P_{out} and the genome sequence collection are needed for the metagenome simulation in step 2. The taxonomic composition of the simulated microbial community is either determined by user-specified taxonomic profiles or generated *de novo* by sampling from available genome sequences.

Profile-based design

Taxonomic profiles can be provided in BIOM (Biological Observation Matrix) format [20]. With input profiles, the NCBI complete genomes [21] are used as the sequence collection for creating metagenome data sets. Optionally, the user can choose to also include genomes marked as "scaffold" or "contig" by the NCBI. Input genomes are split at positions with multiple occurrences of ambiguous bases, such that no reads spanning contig borders within larger scaffolds are simulated.

Profiles can include bacterial, archaeal and eukaryotic taxa, as well as viruses. The taxonomic identifiers of BIOM format are interpreted as free text scientific names and are mapped to NCBI taxon IDs (algorithm in the supplement). The so generated input profile P_{in} specifies pairs (t, ab_t) of taxon IDs t and taxon abundances $ab_t \in \mathbb{R}_{\geq 0}$. The profile taxa are usually defined at higher ranks than strain and thus have to be mapped approximately to the genome sequence collection for creating P_{out}.

Given an ordered list of ranks R = (*species, genus, family, order, class, phylum, superkingdom*), CAMISIM requires as an additional parameter a highest rank $r_{max} \in R$. We define the binary operator $\prec$, based on the ordering of the ranks in R. Given two ranks $r_i, r_j \in R$, we write $r_i \prec r_j$, if r_i appears before r_j in R and we say r_i is below r_j. Related complete genomes are searched for all ranks below r_{max}. By default this is the *family* rank. Another parameter is the maximum number of strains m that are included for an input taxon in a simulated sample.

To create P_{out} from P_{in}, the following steps are performed: Let G_{in} be the set of taxon IDs of the genome collection at the lowest annotated taxonomic rank, usually *species* or *strain*. For all $t \in G_{in}$, the reference taxonomy specifies a taxonomic lineage of taxon IDs (or undefined values) across the considered ranks in R. We use these to identify a collection of sets $F = \{G_t \mid t = \text{lineage taxon represented by} \geq 1 \text{ complete genome}\}$,

Algorithm 1. Creating a community genome abundance profile; *genome-select*(F, P_{in}, m, r_{max})

input: Collection of sets F of taxonomic IDs of available complete genomes, taxonomic profile P_{in}, maximum strains per OTU m, highest rank r_{max} considered for similarity
output: Community genome abundance profile P_{out}

1: $P_{out} = \varnothing$
2: **for each** $(t, ab_t) \in P_{in}$ **do**
3: get lineage path tax_t from reference taxonomy
4: **for each** rank $r \in R \prec r_{max}$ **do**
5: $t_r = tax_t$ on rank r ▷ check whether a complete genome for taxon t_r exists
6: **if** $t_r \in F$ **then**
7: G_{t_r} = set of available full genomes corresponding to taxon t_r in F
8: draw a random number X from truncated geometric distribution ▷ (Eq. 1)
9: **if** $X < |G_{t_r}|$ **then**
10: $G_{selected} =$ randomly select X genomes from G_{t_r}
11: **else**
12: $G_{selected} = G_{t_r}$
13: Y = list of $|G_{selected}|$ random numbers from lognormal distribution ▷ (Eq. 2)
14: **for each** $i \in G_{selected}$ **do**
15: $ab_i = \frac{Y_i}{\sum_{i \in G_{selected}} Y_i} \cdot ab_t$ ▷ (Eq. 3)
16: add (i, ab_i) to P_{out}
17: remove i from G_{t_r}
18: **break** ▷ if a complete genome exists: continue with the next taxon instead of rank
19: **else**
20: issue *Unmapped genome* warning
21: **return** P_{out}

which specifies for each lineage taxon the taxon IDs of available genomes from the genome collection. F is used as input for Algorithm 1.

The algorithm retrieves for each t from the tuples $(t, ab_t) \in P_{in}$ the lineage path tax_t across the ranks of R (lines 2–3). Moving from the *species* to the highest considered rank, r_{max}, the algorithm determines whether for a lineage taxon t_r at the considered rank r a complete genome exists, that is, whether $G_t \neq \varnothing$ for $t = t_r$ (lines 4–5). If this is the case, the search ends and t_r is considered further (line 6). If no complete genome is found for a particular lineage, the lineage is not included in the simulated community, and a warning is issued (line 20). Next, the number of genomes X with their taxonomic IDs t_r to be added to P_{out} is drawn from a *truncated geometric distribution* (Eq. 1, line 8) with a mean of $\mu = \frac{m}{2}$ and the parameter k restricted to be less than m.

$$P(X = k) = (1 - \frac{1}{\mu})^k \cdot \frac{1}{\mu} \tag{1}$$

If $|G_{t_r}|$ is less than X, G_{t_r} is used entirely as $G_{selected}$, the genomes of t_r that are to be included in the community. Otherwise X genomes are drawn randomly from G_{t_r} to generate $G_{selected}$ (lines 9–12). It is optional to use genomes multiple times, by default the selected genomes $g \in G_{selected}$ are removed from F, such that no genome is selected twice (line 17). Based on the taxon abundances ab_t from P_{in}, the abundances ab_i of the selected taxa $i \in G_{selected}$ for t are then inferred. First, random variables Y_i are drawn from a configurable lognormal distribution, with by default mean $\mu = 1$ and variance $\sigma = 2$ (Eq. 2) and then the ab_i are set (Eq. 3; lines 13–15). Finally, the created pairs (i, ab_i) are added to P_{out} (line 16) and P_{out} is returned (line 21).

$$\begin{aligned} Y_i &\sim Lognormal(\mu, \sigma) \\ \Leftrightarrow \frac{d}{dx} P(Y_i \leq x) &= \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}} \end{aligned} \tag{2}$$

$$ab_i = \frac{Y_i}{\sum_{j \in G_{selected}} Y_j} \cdot ab_t \tag{3}$$

De novo design

A genome sequence collection to sample and a mapping file have to be specified. The mapping file defines for each genome a taxonomic ID (per default from the NCBI taxonomy), a novelty category and an Operational Taxonomic Unit (OTU) ID. Grouping genomes into OTUs is required for sampling related genomes, to increase strain-level diversity in the simulated microbial communities. The novelty category reflects how closely a query genome is related to draft or complete genomes in a genome sequence reference collection. This is used to maximize the spread of selected genomes across the range of taxonomic distances to the genome reference collection, such that there are genomes included of "novel" strains, species or genera. This distinction is relevant for evaluating reference-based taxonomic binners and profilers, which may perform differently across these different categories. The user can manually generate the mapping file as described in the supplement or in [15].

If controlled sampling of strains is not required, every genome can be assigned to a different OTU ID. If no reference based taxonomic binners or profilers are to be evaluated, or the provided genome sequence collection does not vary much in terms of taxonomic distance to publicly available genomes used as references for these programs, all genomes can be assigned the same novelty category.

In addition, the number of genomes g_{real} to be drawn from the input genome selection and the total number of genomes g_{tot} for the community genome abundance profile P_{out} have to be specified. The g_{real} real genomes are drawn from the provided genome sampling collection. An equal number of genomes is drawn for every novelty category. If the number of genomes

for a category is insufficient, proportionately more are drawn from others. In addition, CAMISIM simulates $g_{sim} = g_{tot} - g_{real}$ genomes of closely related strains from the chosen real genomes in total. These genomes are created with an enhanced version of sgEvolver [22] (Supplementary Methods) from a subset of randomly selected real genomes. Given m, the maximum number of strains per OTU, up to $m-1$ simulated strain genomes are added *per genome*. The exact number of genomes X to be simulated for a selected OTU is drawn from a geometric distribution with mean $\mu = 0.3^{-1}$ (Eq. 1) This procedure is repeated until g_{sim} related genomes have been added to the community genome collection, comprising $g_{tot} = g_{real} + g_{sim}$ genomes [15].

Next, community genomes are assigned abundances. The relevant user-defined parameters for this step are the sample type and the number of samples n. In addition to single samples, multi-sample data sets (with differential abundances, replicates or time series) have become widely used in real sequencing studies [23–26], also due to their utility for genome recovery using covariance-based genome binners such as CONCOCT [27] or MetaBAT [28]. Several options for creating multi-sample metagenome data sets with these setups are provided:

1. If simulating a *single sample data set*, the relative abundances are drawn from a log-normal distribution, which is commonly used to model microbial communities [29–31]. By default, the mean is set to 1 and the standard deviation to 2 (Eq. 2). The two parameters of the lognormal distribution can be changed. Setting the standard deviation σ^2 to 0 results in a uniform distribution.

2. The *differential abundance mode* models a community sampled multiple times after the environmental conditions or the DNA extraction protocols (and accordingly the community abundance profile) have been altered. This mode creates n different lognormally (Eq. 2) distributed genome abundance profiles.

3. Metagenome data sets with multiple samples with very similar genome abundance distributions can be created using the *replicates mode*. Having multiple replicates of the same metagenome has been reported to improve the quality for some metagenome analysis software, such as for genome binners [23, 27, 32, 33]. Based on an initial log-normal distribution D_0, n samples are created by adding Gaussian noise to this initial distribution (Eq. 4). The Gaussian term accounts for all kinds of effects on the genome abundances of the metagenomic replicates including, but not limited to, different experimenters, different place of extraction, or other batch effects.

$$\begin{aligned} D_i = D_0 + \varepsilon \text{ with } \varepsilon \sim N(0,1) \text{ and} \\ \varepsilon \sim N(0,1) \\ \Leftrightarrow \frac{d}{dx} P(\varepsilon \leq x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2} \end{aligned} \tag{4}$$

4. *Time series* metagenome data sets with multiple related samples can be created. For these, a Markov model-like simulation is performed, with the distribution of each of the n samples (Eq. 5) depending on the distribution of the previous sample plus an additional either lognormal (Eq. 2) or Gaussian (Eq. 4) term. This emulates the natural process of fluctuating abundances over time and ensures that the abundance changes to the previously sampled metagenome do not grow very large.

$$\begin{aligned} D_i &= D_{i-1} + \varepsilon && \text{with} \\ D_0 &\sim Lognormal(1,2) && \text{and} \\ \varepsilon &\sim N(0,1) && \text{or} \\ D_i &= \frac{D_{i-1} + \varepsilon}{2} && \text{with} \\ \varepsilon &\sim Lognormal(1,2) \end{aligned} \tag{5}$$

Metagenome simulation

Metagenome data sets are generated from the genome abundance profiles of the community design step. For each genome-specific taxon t and its abundance $(t, ab_t) \in P_{out}$, its genome size s_t, together with the total number of reads n in the sample, determines the number of generated reads n_t (Eq. 6). The total number of reads n is the overall sequence sample size divided by the mean read-length of the utilized sequencing technology.

$$n_t = n \cdot \frac{ab_t \cdot s_t}{\sum_{i \in P_{out}} ab_i \cdot s_i} \tag{6}$$

By default, ART [34] is used to create Illumina 2 x 150 bp paired-end reads with a HiSeq 2500 error profile. The profile has been trained on MBARC-26 [35], a defined mock community that has already been used to benchmark bioinformatics software and a full-length 16S rRNA gene amplicon sequencing protocol [36, 37], and is distributed with CAMISIM. Other ART profiles, such as the one used for the first CAMI challenge, can also be used. Further available read simulators are wgsim (https://github.com/lh3/wgsim, originally part of SAMtools [38]) for simulating error-free short reads, pbsim [39] for simulating Pacific Biosciences data and nanosim [40] for simulating Oxford Nanopore Technologies reads. The read lengths and insert sizes can be varied for some simulators.

For every sample of a data set, CAMISIM generates FASTQ files and a BAM file [38]. The BAM file specifies the alignment of the simulated reads to the reference genomes.

Gold standard creation and postprocessing

From the simulated metagenome data sets – the FASTQ and BAM files – CAMISIM creates the assembly and binning gold standards. The software extracts the perfect assembly for each individual sample, and a perfect co-assembly of all samples together by identifying all genomic regions with a coverage of at least one using SAMtools' mpileup and extracting these as error-free contigs. This gold standard does not include all genome sequences available for the simulation, but the best possible assembly of their sampled reads.

CAMISIM generates the genome and taxon binning gold standards for the reads and assembled contigs, respectively. These specify the genome and taxonomic lineage that the individual sequences belong to. All sequences can be anonymized and shuffled (but tracked throughout the process), to enable their use in benchmarking challenges. Lastly, files are compressed with gzip and written to the specified output location.

RESULTS

Comparison to the state-of-the-art

We tested seven simulators and compared them to CAMISIM (Table 1). All generate Illumina data and some – NeSSM [44], BEAR [45], FASTQSim [46] and Grinder [47] – also use a taxonomic profile. Novel and unique to CAMISIM is the ability to simulate

Table 1. Properties of popular metagenome sequence simulators. The table shows if an abundance profile can be generated by the simulator *de novo* and if an existing *profile* of a microbial community can be used as input. Further inspected features are the ability to simulate *multi*-sample data sets, *strains*, and *non-Illumina data* (e.g. long reads). Lastly, the table states if and how a software is *licensed*, and the date it was last recently *updated*.

Software	*de novo*	profile	multi	strains	non-Illunina data	licensed	updated
MetaSim [41]	✓	X	X	✓	454	P, AU	03/2009
iMESS [42]	✓	X	X	X	454	–	07/2014
BBMap [43]	✓	X	X	X	–	LBL	04/2018
NeSSM [44]	✓	✓	X	X	454	AU	07/2013
BEAR [45]	✓	✓	X	X	–	AU	02/2017
FASTQSim [46]	✓	✓	X	X	SOLiD, IonTorrent, PacBio	GPL	05/2015
Grinder [47]	✓	✓	✓	X	Sanger, 454	GPL	04/2016
CAMISIM	✓	✓	✓	✓	PacBio, ONT, ...	Apache 2.0	04/2018

Abbreviations: P = proprietary software; AU = academic use only; LBL = Lawrence Berkeley Lab.

long-read data from Oxford Nanopore, of hybrid data sets with multiple sequencing technologies and multi-sample data sets, such as with replicates, time series or differential abundances. Grinder [47] can also create multiple samples, but only with differential abundances. In addition, CAMISIM creates gold standards for assembly (single sample assemblies and multi-sample co-assemblies), for taxonomic and genome binning of reads or contigs and for taxonomic profiling. Finally, CAMISIM can evolve multiple strains for selected input genomes, and allows specification of the degree of real and simulated intra-species heterogeneity within a data set.

Effect of data properties on assemblies

We created several thousand "minimally challenging" metagenome samples by varying one data property relevant for assembly, while keeping all others the same. Using these, we studied the effect of evolutionary divergence between genomes, different error profiles and coverage on the popular metaSPAdes [19], version 3.7.0, and MEGAHIT [18], versions 1.1.2 and 1.0.3, assemblers, to systematically investigate reported performance declines for assemblers in the presence of strain-level diversity, uneven coverage distributions and abnormal error profiles [15, 48, 49]. Both MEGAHIT and metaSPAdes work on de Bruijn graphs, which are created by splitting the input reads into smaller parts, the *k*-mers, and connecting two *k*-mers if they overlap by exactly *k-1* letters. For every sequencing error *k* erroneous *k*-mers are introduced into the de Bruijn graph, which might substantially impact assembly (Figure 2).

Varying genome coverage and sequencing error rates

We initially simulated samples from *Escherichia coli* K12 MG1655 with varying coverage and different error rates. Reads were generated at 512x genome coverage and subsampled stepwise by 50% until 2x coverage was reached, resulting in a sample series with 512, 256, 128, 64, 32, 16, 8, 4 and 2-fold coverage, respectively. Subsampling was employed to control variation in the sampling of different genomic regions. To assess the effect of sequencing errors, 4 read data sets were simulated; three using wgsim with uniform error rates of 0%, 2% and 5%, and one using ART with the CAMI challenge error profile (ART CAMI).

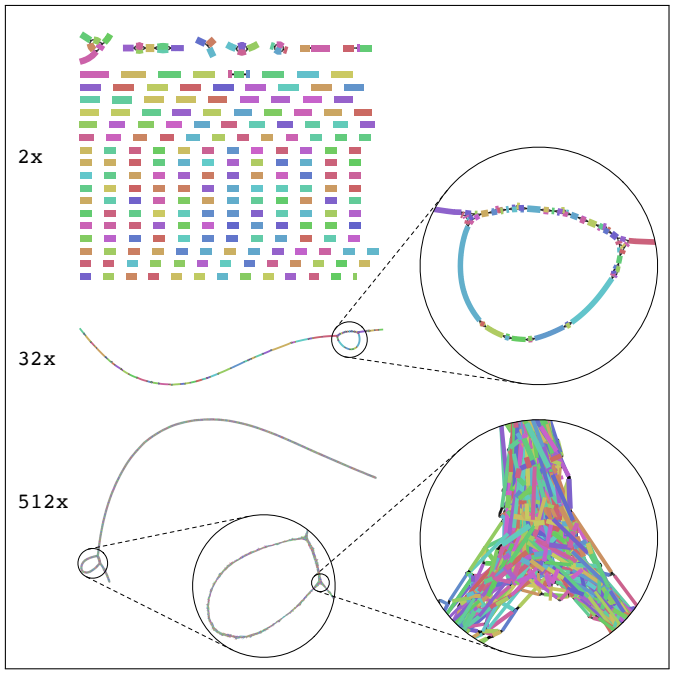

Fig. 2. Assembly graphs become more complex as coverage increases. MEGAHIT assembly graphs (*k*=41) of an *E. coli* K12 genome for 2x, 32x, and 512x per-base coverage, respectively, visualized with Bandage [50]. For 2x coverage, the graph is disconnected and thus the assembly fragmented. With increasing coverage more and more unitigs can be joined, first resulting in a decent assembly for 32x coverage, but – due to sequencing errors adding erroneous edges to the graph – a fragmented assembly again for 512x coverage.

Both assemblers were run on these data sets with default options, except for the phred-offset parameter for metaSPAdes, which was set to 33. Both assemblers performed similar across all error rates and coverages, with assembly quality varying substantially with coverage (Figure 3). Performance on the data generated with the 5% error profile was worst throughout. This

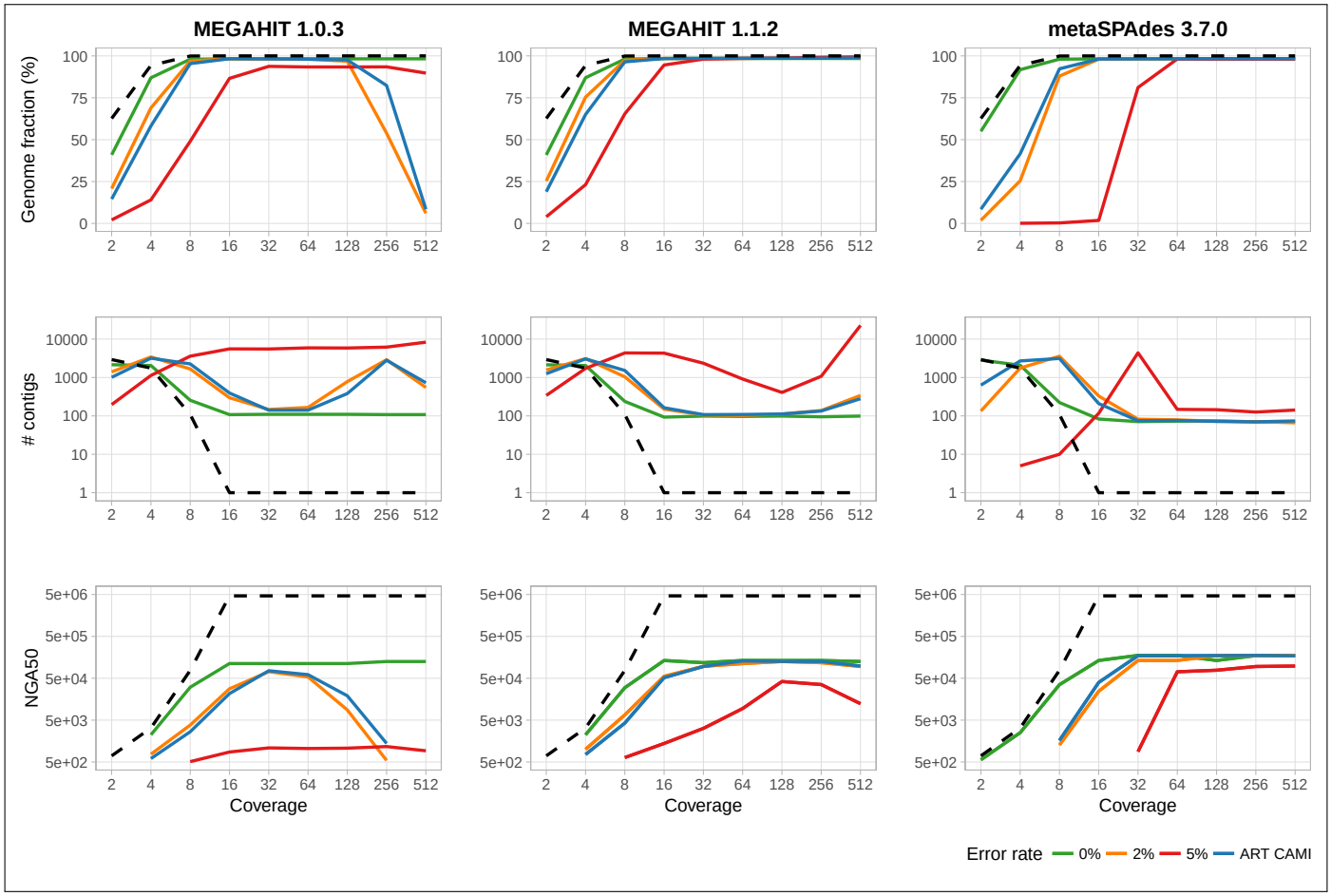

Fig. 3. Coverage dependent assembly performance for MEGAHIT and metaSPAdes. Shown are the metrics, from top to bottom: Genome fraction in %, number of contigs, and NGA50 (as reported by QUAST [51]), for 0%, 2%, and 5% uniform error rate, and with the ART CAMI error profile compared to the best possible metrics (gold standard) on the ART CAMI profile (dashed black).

is an unrealistically high error profile for Illumina data [49] that software need not necessarily be adapted to handle well.

If coverage was low, assembly failed, generating a large number of small (low NGA50) contigs covering only a small genome portion (genome fraction) across all data sets, because of uncovered regions in the genomes. Sequencing errors (denoted ε) do not play a major role (Figure 2). The expected per-base error-rate $E_p = \text{cov} \cdot \varepsilon$ (disregarding the biased errors in the short-read sequencing technologies) is far below 1 ($E_p \ll 1$). With increasing coverage, assembly improved consistently across the 0%, 2% and CAMISIM ART error profile data sets and both assemblers for all metrics (Figure 3), and reaching an early plateau by 8–16x coverage.

Notably, the performance of an earlier version of MEGAHIT (1.0.3) decreased substantially (declining genome fraction and NGA50) for more than 128x coverage, except for error-free reads. For instance, at 5% error rate, MEGAHIT, version 1.0.3, generated an exponential number of contigs at high coverages, which keeps the genome fraction artificially high. For these high coverages and error rates, we expect multiple errors at every position of the genome ($E_p \gg 10$). This creates de Bruijn graphs with many junctions and bubbles (Figure 2) which cannot easily be resolved and may lead to breaking the assembly apart and covering the same part of the genome with multiple, short, erroneous contigs (Figure 3).

Effect of evolutionary relatedness on assembler performances

We systematically investigated the effect of related strains on assembler performances across a wide range of taxa and evolutionary divergences, using the genomes of 152 species from the interactive tree of life iTol [52], which includes bacteria, archaea and eukaryotes. For each genome we evolved 19 related genomes without larger insertions and deletions and an Average Nucleotide Identity (ANI) between 90% and 99.5% to the original one using steps of 0.5%. For each of the $152 \cdot 20 = 3,040$ pairs of original and evolved genome sequences, we simulated single sample minimal metagenomes at equal genome abundances, with error-free reads at $50x$ coverage using wgsim. This constitutes good coverage for the analyzed assemblers, as shown in the previous section. For the resulting samples, variation in assembler performance should thus primarily be caused by differences in ANI.

The presence of closely related genomes substantially affected assembly quality (Figure 4). For up to 95% ANI, the assemblers restored high quality assemblies for both genomes. Between 95% and 99% ANI, the genome fraction and assembly size dropped substantially and contig numbers increased. This was the case if we allowed contigs to either map uniquely to one reference genome or to both, in case of multiple optimal mappings. For more closely related genomes, the number of contigs increased drastically and the assembly size continued to drop. The genome

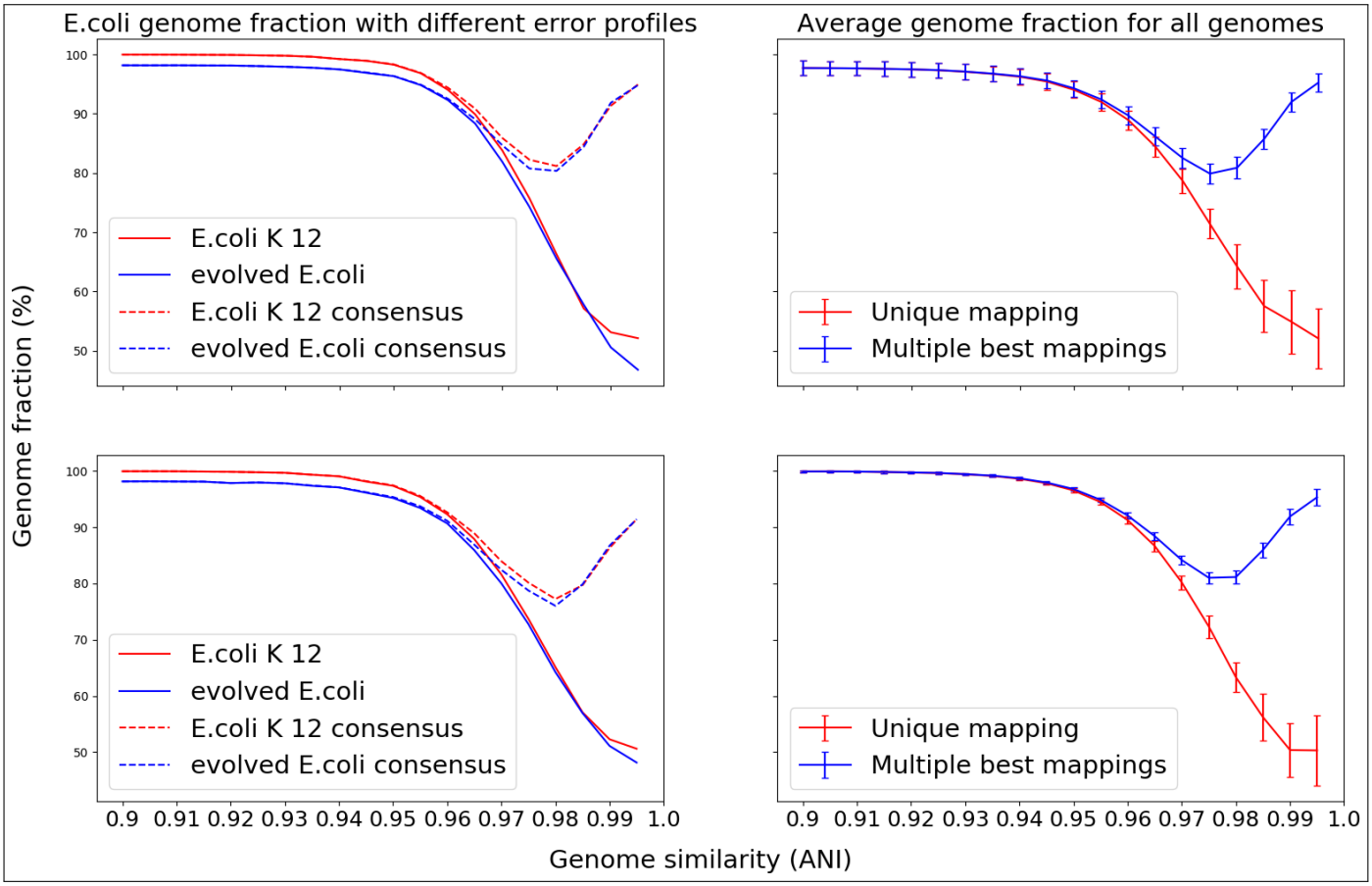

Fig. 4. Genome fraction calculated using unique or multiple best mappings in case of ties to the community genome collection. Left: Genome fraction for the *E. coli* assembly created by MEGAHIT from error-free reads (top) and with ART CAMI error profile (bottom). Right: Average genome fraction and standard deviation for all original 152 iTol genomes created by MEGAHIT from error-free reads (top) and with ART CAMI error profile (bottom). Error bars denote 1x standard deviation.

fraction remained high when considering non-unique mappings, but decreased for unique mappings: the explanation for this observed behavior is that for an ANI of more than 99%, assemblers produced consensus contigs of the two strains that mostly aligned similarly well to both reference genomes. This was the case for all 152 genomes and their evolved counterparts.

Simulating environment-specific data sets

To test the ability to create metagenome data of the human microbiome, we simulated metagenomes from taxonomic profiles of the Human Microbiome Project [9] for different body sites with CAMISIM. We selected 49 samples from the airways, gastrointestinal tract, oral cavity, skin and urogenital tract, with whole genome shotgun (WGS) and 16S rRNA gene amplicon sequence data available. We used the published qiime OTU table (https://hmpdacc.org/hmp/HMQCP/) to generate 5 Gb of simulated reads per sample with CAMISIM, resulting in a data set of 245 Gb of Illumina data, and of PacBio data, respectively. Only genomes tagged as "complete genomes" in the NCBI were considered in the data set generation. To decrease the chance of OTUs not being represented by a genome, the option of allowing multiple OTUs being represented by a single reference genome was turned on. This can be relevant for instance when due to sequencing errors in 16S rRNA data, individual community genomes are represented by multiple OTUs.

For a functional comparison of the simulated data with the original metagenome shotgun data, we inferred KEGG Ortholog family abundance profiles from the raw read data sets [53]. To this end, all reads were searched with Diamond v0.9.10 using its blastx command with default options [54] against the KEGG GENES database (release 77, species_prokaryotes, best-hit approach) and linked to KEGG Orthology (KO) via the KEGG mapping files. KO profile similarity between the simulated and original metagenome samples was calculated with Pearson's correlation coefficient (PCC) and Spearman rank correlation (SRC), and visualized with non-metric multidimensional scaling (NMDS) [55].

For comparison we also created functional profiles with PICRUSt [56], using a prediction model generated from 3772 KEGG genomes and corresponding 16S rRNA gene sequences according to the PICRUSt "Genome Prediction Tutorial" (Supplementary Information). The PCC of the CAMISIM and original samples approached a striking 0.97 for body sites with high bacterial abundances and many sequenced genomes available, such as the GI tract and oral cavity, and still ranged from 0.72 to 0.91 for airways, skin and urogenital tract samples (Figure 5B). All PCCs were 7-30% higher than the PCC of PICRUSt with the original metagenome samples. Thus CAMISIM created metagenome samples functionally even closer to the original metagenome samples than the functional profiles created by PICRUSt. The

higher PCC may also partly be due to the fact that the original and CAMISIM data were annotated by "blasting" reads versus KEGG, while the PICRUSt profiles were directly generated from KEGG genome annotations. The Spearman correlation of the simulated CAMISIM samples to the original metagenome samples was slightly lower than the PCC across all body sites, and very similar for CAMISIM and PICRUSt (0–6% improvement of CAMISIM over PICRUSt). These results demonstrate the quality of the CAMISIM samples.

The NMDS plot (Figure 5A) showed a very distinct clustering of the CAMISIM and original WGS samples by body site, more closely than the original samples clustered with the PICRUSt profiles. Even though the urogenital tract samples did not cluster perfectly, the CAMISIM samples still formed a very distinct cluster close to the original one. Even outliers in the original samples were, at least partly, detected and correctly simulated (both original and simulated sample 26 of urogenital tract cluster most closely with the gastrointestinal tract microbiomes).

We also provide a multi-sample mouse gut data set for software developers to benchmark against. For 64 16S rRNA samples from the mouse gut [17], we simulated 5 Gb of Illumina and PacBio reads each. The mice were obtained from 12 different vendors and the samples characterized by 16S V4 amplicon sequencing (OTU mapping file in the supplement). Since for mouse gut only a few complete reference genomes were available, the "scaffold" quality for downloading genomes was chosen.

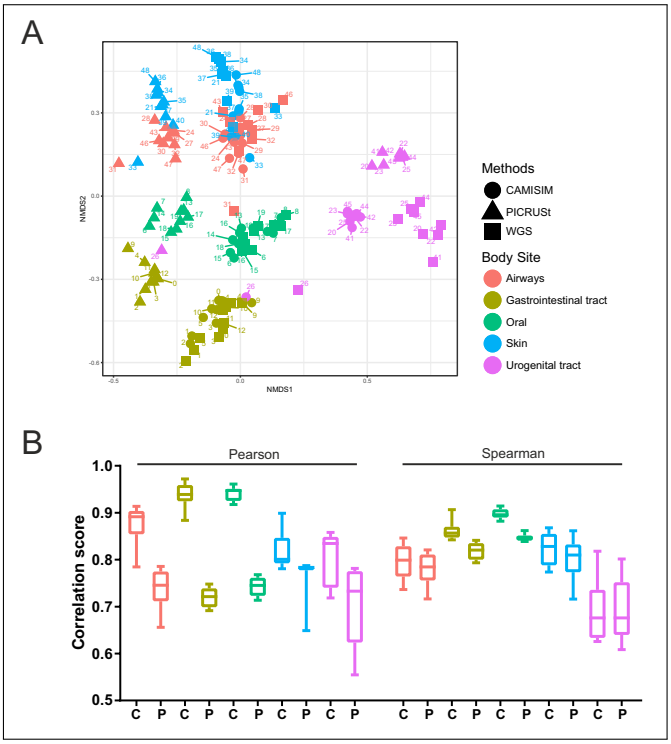

Fig. 5. Comparison of CAMISIM and PICRUSt functional profiles for different body sites. (A) NMDS ordination of the functional predictions of individual samples by the different methods. The different body sites are color-coded and labeled with their sample number. The original WGS is denoted by squares, the CAMISIM result as circles and the PICRUSt result as triangles. (B) Mean and standard deviation of Pearson and Spearman correlation to original WGS samples per body site. C: CAMISIM; P: PICRUSt.

DISCUSSION AND CONCLUSIONS

CAMISIM is a flexible program for simulating a large variety of microbial communities and metagenome samples. To our knowledge it possesses the most complete feature set for simulating realistic microbial communities and metagenome data sets. This feature set includes: simulation from taxonomic profiles as templates, inclusion of both natural and simulated strain level diversity, and modelling multi-sample data sets with different underlying community abundance distributions. Read simulators are included for short read (Illumina) and long read (PacBio, ONT) sequencing technologies, allowing the generation of hybrid data sets. This turns CAMISIM into a versatile metagenome simulation pipeline, as modules for new (or updated) sequencing technologies and emerging experimental setups can easily be incorporated.

We systematically explored the effect of specific data properties on assembler performances on several thousand minimally challenging metagenomes. While low coverage reduced assembly quality for both assemblers, MetaSPAdes and MEGAHIT performed generally well for medium to high coverages and different error profiles. Notably, MEGAHIT is computationally very efficient and overall performed well. As noted before [15, 57], assemblers had problems with resolving closely related genomes in our experiments. For an in-depth investigation, we systematically analyzed the effect of related strains on MEGAHIT's performance across a wide range of taxa and evolutionary divergences. The average nucleotide identity (ANI) between two genomes is a robust measure of genome relatedness; an ANI value of 95% roughly corresponds to a 70% DNA-DNA reassociation value – a historical definition of bacterial species [58, 59]. For an pairwise ANI below 95%, the mixture of strains could be separated quite well and assembled into different contigs. For an ANI of more than 99%, consensus contigs of strains were produced that mostly aligned similarly well to either reference genome. In the "twilight zone" of 95–99% nucleotide identity, assembly performance dropped substantially and MEGAHIT's inability to reliable phase strain variation resulted in many small (and often redundant) contigs. For IDBA-UD [60], another *de Bruijn* graph-based metagenome assembler, a similar pattern has been observed [61], indicating that such behavior is common to many assemblers.

Resolving strains from metagenome shotgun data is an open research question, though recently promising computational approaches were proposed [11, 62]. The hybrid long and short read simulated data sets we are providing for mouse gut and human body sites could enable the development of new appraoches for this task CAMISIM will facilitate the generation of further realistic benchmarking data sets to assess their performances. It can also be used to study the effect of experimental design (e.g. number of replicates, sequencing depth, insert sizes) or intrinsic community properties, such as taxonomic composition, community abundance distributions, and organismal complexities, on program performance. Due to the enormous diversity of naturally occurring microbial communities, experimental and sequencing technology setups used in the field, such explorations are required to determine the most effective combinations for specific research questions.

SOFTWARE AND DATA AVAILABILITY

CAMISIM is implemented in Python 2.7 and available under the Apache 2.0 license. The software, config files, input genomes, and metadata are available at: https://github.com/CAMI-challenge/CAMISIM and https://github.com/CAMI-challenge/CAMISIM-DATA.

The large human and mouse gut microbiome data sets (alongside the BIOM and config files from which they were created) are available at: https://data.cami-challenge.org/participate.

AUTHORS' CONTRIBUTIONS

AF, PH, SM, ED, JD, JF, MZD, AED, and AB implemented CAMISIM; AF and PB tested the software; AF, TRL, and AB performed the experiments; AF, TRL, AS, AB, and ACM interpreted the results; AF, PH, TRL, PB, AB, and ACM wrote the manuscript; AB and ACM conceived the experiments; ACM conceived and supervised the project. All authors read and approved the final manuscript.

ACKNOWLEDGEMENTS

The authors thank the Isaac Newton Institute for Mathematical Sciences for its hospitality during the programme MTG, which was supported by EPSRC Grant Number EP/K032208/1, and Victoria Sack for generating Figure 1. This research project has been supported by the President's Initiative and Networking Funds of the Helmholtz Association of German Research Centres (HGF) under contract number VH-GS-202.

REFERENCES

1. P. J. Turnbaugh, R. E. Ley, M. Hamady, C. Fraser-Liggett, R. Knight, and J. I. Gordon, "The human microbiome project: exploring the microbial part of ourselves in a changing world," Nature **449**, 804–810 (2007).
2. L. M. Proctor, S. Sechi, N. D. DiGiacomo, J. M. Fettweis, K. K. Jefferson *et al.*, "The Integrative Human Microbiome Project: dynamic analysis of microbiome-host omics profiles during periods of human health and disease," Cell Host Microbe **16**, 276–289 (2014).
3. F. Warnecke, P. Luginbühl, N. Ivanova, M. Ghassemian, T. H. Richardson *et al.*, "Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite," Nature **450**, 560–565 (2007).
4. M. Hess, A. Sczyrba, R. Egan, T.-W. Kim, H. Chokhawala *et al.*, "Metagenomic discovery of biomass-degrading genes and genomes from cow rumen," Science **331**, 463–467 (2011).
5. A. Bremges, I. Maus, P. Belmann, F. Eikmeyer, A. Winkler *et al.*, "Deeply sequenced metagenome and metatranscriptome of a biogas-producing microbial community from an agricultural production-scale biogas plant," GigaScience **4**, 33 (2015).
6. S. Sunagawa, L. P. Coelho, S. Chaffron, J. R. Kultima, K. Labadie *et al.*, "Ocean plankton. Structure and function of the global ocean microbiome," Science **348**, 1261359 (2015).
7. L. Xiao, Q. Feng, S. Liang, S. B. Sonne, Z. Xia *et al.*, "A catalog of the mouse gut metagenome," Nat. Biotechnol. **33**, 1103–1108 (2015).
8. B. J. Kunath, A. Bremges, A. Weimann, A. C. McHardy, and P. B. Pope, "Metagenomics and CAZyme Discovery," Methods Mol. Biol. **1588**, 255–277 (2017).
9. C. Huttenhower, D. Gevers, R. Knight, S. Abubucker, J. H. Badger *et al.*, "Structure, function and diversity of the healthy human microbiome," Nature. **486**, 207–214 (2012).
10. M. Scholz, D. V. Ward, E. Pasolli, T. Tolio, M. Zolfo *et al.*, "Strain-level microbial epidemiology and population genomics from shotgun metagenomics," Nat. Methods **13**, 435–438 (2016).
11. C. Quince, T. O. Delmont, S. Raguideau, J. Alneberg, A. E. Darling *et al.*, "DESMAN: a new tool for de novo extraction of strains from metagenomes," Genome Biol. **18**, 181 (2017).
12. L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau *et al.*, "A communal catalogue reveals earth's multiscale microbial diversity," Nature. (2017).
13. C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," Nat. Biotechnol. **35**, 833–844 (2017).
14. S. Goodwin, J. D. McPherson, and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," Nat. Rev. Genet. **17**, 333–351 (2016).
15. A. Sczyrba, P. Hofmann, P. Belmann, D. Koslicki, S. Janssen *et al.*, "Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software," Nat. Methods **14**, 1063–1071 (2017).
16. P. Belmann, J. Dröge, A. Bremges, A. C. McHardy, A. Sczyrba, and M. D. Barton, "Bioboxes: standardised containers for interchangeable bioinformatics software," GigaScience **4**, 47 (2015).
17. U. Roy, E. J. C. Galvez, A. Iljazovic, T. R. Lesker, A. J. Blazejewski *et al.*, "Distinct microbial communities trigger colitis development upon intestinal barrier damage via innate or adaptive immune cells," Cell Reports **21**, 994–1008 (2017).
18. D. Li, C.-M. Liu, R. Luo, K. Sadakane, and T.-W. Lam, "MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph," Bioinformatics **31**, 1674–1676 (2015).
19. S. Nurk, D. Meleshko, A. Korobeynikov, and P. A. Pevzner, "metaSPAdes: a new versatile metagenomic assembler," Genome Res. p. gr.213959.116 (2017).
20. D. McDonald, J. C. Clemente, J. Kuczynski, J. R. Rideout, J. Stombaugh *et al.*, "The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome," GigaScience. **1**, 7 (2012).
21. K. D. Pruitt, T. Tatusova, and D. R. Maglott, "NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins," Nucleic Acids Res. **35**, D61–65 (2007).
22. A. C. E. Darling, B. Mau, F. R. Blattner, and N. T. Perna, "Mauve: multiple alignment of conserved genomic sequence with rearrangements," Genome Res. **14**, 1394–1403 (2004).
23. M. Albertsen, P. Hugenholtz, A. Skarshewski, K. L. Nielsen, G. W. Tyson, and P. H. Nielsen, "Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes," Nat. Biotechnol. **31**, 533–538 (2013).
24. M. L. Bendall, S. L. Stevens, L.-K. Chan, S. Malfatti, P. Schwientek *et al.*, "Genome-wide selective sweeps and gene-specific sweeps in natural bacterial populations," The ISME J. **10**, 1589–1601 (2016).
25. Y. Stolze, A. Bremges, M. Rumming, C. Henke, I. Maus *et al.*, "Identification and genome reconstruction of abundant distinct taxa in microbiomes from one thermophilic and three mesophilic production-scale biogas plants," Biotechnol Bio-

fuels **9**, 156 (2016).
26. S. Roux, L.-K. Chan, R. Egan, R. R. Malmstrom, K. D. McMahon, and M. B. Sullivan, "Ecogenomics of virophages and their giant virus hosts assessed through time series metagenomics," Nat. Commun. **8** (2017).
27. J. Alneberg, B. S. Bjarnason, I. de Bruijn, M. Schirmer, J. Quick *et al.*, "Binning metagenomic contigs by coverage and composition," Nat. Methods **11**, 1144–1146 (2014).
28. D. D. Kang, J. Froula, R. Egan, and Z. Wang, "MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities," PeerJ **3**, e1165 (2015).
29. I. D. Ofiţeru, M. Lunn, T. P. Curtis, G. F. Wells, C. S. Criddle *et al.*, "Combined niche and neutral effects in a microbial wastewater treatment community," Proc. Natl. Acad. Sci. **107**, 15345–15350 (2010).
30. W. Ulrich, M. Ollik, and K. I. Ugland, "A meta-analysis of species–abundance distributions," Oikos. **119**, 1149–1155 (2010).
31. M. Unterseher, A. Jumpponen, M. Opik, L. Tedersoo, M. Moora *et al.*, "Species abundance distributions and richness estimations in fungal metagenomics–lessons learned from community ecology," Mol. Ecol. **20**, 275–285 (2011).
32. H. B. Nielsen, M. Almeida, A. S. Juncker, S. Rasmussen, J. Li *et al.*, "Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes," Nat. Biotechnol. **32**, 822–828 (2014).
33. M. Imelfort, D. Parks, B. J. Woodcroft, P. Dennis, P. Hugenholtz, and G. W. Tyson, "GroopM: an automated tool for the recovery of population genomes from related metagenomes," PeerJ. **2**, e603 (2014).
34. W. Huang, L. Li, J. R. Myers, and G. T. Marth, "ART: a next-generation sequencing read simulator," Bioinformatics. **28**, 593–594 (2012).
35. E. Singer, B. Andreopoulos, R. M. Bowers, J. Lee, S. Deshpande *et al.*, "Next generation sequencing data of a defined microbial mock community," Sci. Data **3**, 160081 (2016).
36. A. Bremges, E. Singer, T. Woyke, and A. Sczyrba, "MeCorS: Metagenome-enabled error correction of single cell sequencing reads," Bioinformatics **32**, 2199–2201 (2016).
37. E. Singer, B. Bushnell, D. Coleman-Derr, B. Bowman, R. M. Bowers *et al.*, "High-resolution phylogenetic microbial community profiling," ISME J **10**, 2020–2032 (2016).
38. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan *et al.*, "The Sequence Alignment/Map format and SAMtools," Bioinformatics **25**, 2078–2079 (2009).
39. Y. Ono, K. Asai, and M. Hamada, "PBSIM: PacBio reads simulator–toward accurate genome assembly," Bioinformatics **29**, 119–121 (2013).
40. C. Yang, J. Chu, R. L. Warren, and I. Birol, "NanoSim: nanopore sequence read simulator based on statistical characterization," GigaScience (2017).
41. D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson, "MetaSim—a sequencing simulator for genomics and metagenomics," PLoS ONE **3**, e3373 (2008).
42. D. R. Mende, A. S. Waller, S. Sunagawa, A. I. Järvelin, M. M. Chan *et al.*, "Assessment of metagenomic assembly using simulated next generation sequencing data," PLoS ONE **7**, e31386 (2012).
43. B. Bushnell, "Bbmap: A fast, accurate, splice-aware aligner," https://sourceforge.net/projects/bbmap (2014).
44. B. Jia, L. Xuan, K. Cai, Z. Hu, L. Ma, and C. Wei, "NeSSM: A next-generation sequencing simulator for metagenomics," PLoS ONE **8**, e75448 (2013).
45. S. Johnson, B. Trost, J. R. Long, V. Pittet, and A. Kusalik, "A better sequence-read simulator program for metagenomics," BMC Bioinforma. **15**, S14 (2014).
46. A. Shcherbina, "FASTQSim: platform-independent data characterization and in silico read generation for NGS datasets," BMC Res. Notes **7**, 533 (2014).
47. F. E. Angly, D. Willner, F. Rohwer, P. Hugenholtz, and G. W. Tyson, "Grinder: a versatile amplicon and shotgun sequence simulator," Nucleic Acids Res. **40**, e94–e94 (2012).
48. C. Rinke, P. Schwientek, A. Sczyrba, N. N. Ivanova, I. J. Anderson *et al.*, "Insights into the phylogeny and coding potential of microbial dark matter," Nature. **499**, 431–437 (2013).
49. D. Laehnemann, A. Borkhardt, and A. C. McHardy, "Denoising DNA deep sequencing data-high-throughput sequencing errors and their correction," Briefings Bioinforma. **17**, 154–179 (2016).
50. R. R. Wick, M. B. Schultz, J. Zobel, and K. E. Holt, "Bandage: interactive visualization ofde novogenome assemblies," Bioinformatics. **31**, 3350–3352 (2015).
51. A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler, "QUAST: quality assessment tool for genome assemblies," Bioinformatics. **29**, 1072–1075 (2013).
52. I. Letunic and P. Bork, "Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation," Bioinformatics. **23**, 127–128 (2007).
53. M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," Nucleic Acids Res. **44**, D457–D462 (2015).
54. B. Buchfink, C. Xie, and D. H. Huson, "Fast and sensitive protein alignment using DIAMOND," Nat. Methods **12**, 59–60 (2014).
55. J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika **29**, 1–27 (1964).
56. M. G. I. Langille, J. Zaneveld, J. G. Caporaso, D. McDonald, D. Knights *et al.*, "Predictive functional profiling of microbial communities using 16s rrna marker gene sequences," Nat Biotech **31**, 814–821 (2013).
57. S. Awad, L. Irber, and C. T. Brown, "Evaluating metagenome assembly on a simple defined community with many strain variants," bioRxiv (2017).
58. K. T. Konstantinidis and J. M. Tiedje, "Genomic insights that advance the species definition for prokaryotes," Proc. Natl. Acad. Sci. U.S.A. (2005).
59. N. J. Varghese, S. Mukherjee, N. Ivanova, K. T. Konstantinidis, K. Mavrommatis *et al.*, "Microbial species delineation using whole genome sequences," Nucleic Acids Res. (2015).
60. Y. Peng, H. C. M. Leung, S. M. Yiu, and F. Y. L. Chin, "IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth," Bioinformatics. **28**, 1420–1428 (2012).
61. M. Z. DeMaere and A. E. Darling, "Deconvoluting simulated metagenomes: the performance of hard- and soft-clustering algorithms applied to metagenomic chromosome conformation capture (3c)," PeerJ. **4**, e2676 (2016).
62. B. Cleary, I. L. Brito, K. Huang, D. Gevers, T. Shea *et al.*, "Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning," Nat. Biotechnol. **33**, 1053–1060 (2015).