Critical Assessment of Metagenome Interpretation

Supplementary Notes

Content

Nature Methods: doi:10.1038/nmeth.4458

Nature Methods: doi:10.1038/nmeth.4458

1 Supplementary Note 1

1.1 Data generation

1.1.1 Generation of plasmid and viral data

Plasmid sequences were identified in samples from brown rat (*Rattus norvegicus*) cecal content, collected from wild rats from an urban area in north Jutland and an area in Copenhagen, Denmark. Circular DNA from the samples was extracted and amplified using Multiple Displacement Amplification as previously described[1,2]. Libraries were constructed with the Illumina Nextera XT kit and dual indexing primers. They were sequenced with the Ilumina HiSeq2000 System (2x100nt), using the TruSeq Paired-End Dual Index Sequencing Primer Box. All bioinformatics work was done in a UNIX environment, running Biopieces version 0.51 (by Martin Asser Hansen, unpublished (www.biopieces.org). The reads were qualitatively trimmed, with parameters identical to the literature[2]. Both single and paired-end reads were used for assembly with IDBA-UD with the same parameters as in literature[2]. Circular contigs were identified using the two-step approach for Illumina sequences validated in [2]. Duplicate sequences were filtered out, leaving a non-redundant dataset of 1.7 Mb of 598 circular sequences. These circular sequences range from 901 to 30,879 bp and have an N50 of 3,658 bp. Plasmid and virus related genes were identified using HMMER3.0[3,4,5] and lists of PFAM families found to be specific for plasmid or virus sequences with a cutoff e-value of 10E-4 (Supplementary Table 11). Circular sequences with either one of the plasmid replication, mobilization, and/or stabilization genes were designated as plasmids, while sequences without plasmid-like genes and with virus replication or capsid genes were as virus-like. This sequential binning has been shown to be very accurate on a mock dataset of annotated viruses and plasmids, with only 20% unassigned sequences and 0.25% misannotation. Circular sequences with neither plasmid- or virus-like genes were defined as 'circular element'. The circular sequences, raw data, and metadata were deposited at the European Nucleotide Archive (ENA) under the project accession number PRJEB20380.

1.1.2 Taxonomic annotation, OTU inference and novelty category assignment for bacterial and archaeal genomes

For assignment of OTU membership and taxonomic novelty relative to sequenced genomes (novelty category) all genomes were analyzed, as follows. For the 310 genome sequences of isolates from culture collections, the already assigned

Nature Methods: doi:10.1038/nmeth.4458

taxonomic IDs were used. For 488 newly sequenced isolate genomes without taxonomic IDs assigned (all genomes isolated from the environment), taxonomic annotation was performed using 16S marker genes, as described below.

For each genome sequence, the extracted marker genes were aligned (step 2) to a manually prepared reference alignment of 16S marker genes (step 1) and then clustered (step 3). The genome was then taxonomically classified based on that clustering and assigned a taxonomic novelty category (step 4). This analysis was partly automated with codes available at https://github.com/CAMI-challenge/MetagenomeSimulationPipeline/releases/tag/0.1.

Step 1. Reference data preparation

This is a manual step. An alignment of marker genes is required and one as such is available from the ARB-SILVA[6], which is a high quality ribosomal RNA database. As the focus is on bacteria and archaea, we used the N99 SILVA small subunit (SSU) 119 dataset of fully aligned sequences, and removed all 18S rRNA sequences from it. The N99 subset of the ARB-SILVA database includes representative sequences that are maximally 99% identical to others, which reduces the calculation burden for compute-intensive applications such as clustering. Furthermore, we focused on the 16S genes of archaea and bacteria, excluding eukaryotic, artificial and unidentified sequences. This reduced the dataset from 534,968 sequences to 482,974 sequences. Of those, 18,797 are archaea and 464,177 bacteria.

SILVA taxonomy to NCBI taxonomy mapping

The ARB-SILVA database offers its own taxonomic classification, along with NCBI taxonomic identifiers. Their analysis determined a small fraction of taxonomic inconsistencies, where distantly related NCBI taxonomic identifiers were assigned to closely related sequences, in conflict with the SILVA taxonomic assignments. For that reason, we performed another mapping of the taxonomic classification by SILVA to the NCBI taxonomy, as in [7], where the assigned taxon ID is the taxon corresponding to the lowest common ancestor of the two taxonomic paths assigned by SILVA and NCBI, respectively. The missing taxon IDs on otherwise consistent taxonomic paths were ignored.

For example, AB004755 belongs to the Escherichia-Shigella group in SILVA, specifically to the lineage

Nature Methods: doi:10.1038/nmeth.4458

“Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacteriales;Enterobacteriacea e;Escherichia-Shigella;Raoultella planticola”.

The group 'Escherichia-Shigella' is not known in the NCBI taxonomy, but the related lineages are otherwise consistent. Therefore, the mapping will result in the assignment of taxon ID '575', Raoultella planticola.

If there are conflicts between taxonomies, higher level assignments, specifically to the taxon being the lowest common ancestor on the two taxonomic paths, are made. For example, FJ268997 is assigned to

“Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacteriales;Enterobacteriacea e;Escherichia-Shigella;uncultured Stenotrophomonas sp.”

in SILVA. This sequence belongs to the 'Xanthomonadales' order in the NCBI taxonomy, which is inconsistent with the rest of the SILVA lineage up to Enterobacteriales. Therefore, an assignment to the first consistent NCBI taxon (the lowest common ancestor in both taxonomic paths) will be performed, specifically to the class “Gammaproteobacteria”.

Overall, the back mapping generated NCBI taxonomic IDs for the following ranks:

- *superkingdom*: 938;
- *phylum*: 20770;
- *class*: 22602;
- *order*: 28459;
- *family*: 82789;
- *genus*: 199909;
- *species*: 127507.

In total, 482974 IDs were assigned.

Preparing the reference sequence collection for analysis with mothur

mothur[8] is an open-source, platform-independent, community-supported software for describing and comparing microbial communities. A distance matrix holding the distances between all pairs of reference sequences was created using the following mothur commands:

Nature Methods: doi:10.1038/nmeth.4458

1. first **unique.seqs**(fasta=<SILVA_alignment_file_path>) generates a names file that maps identical sequences to a single sequence id;
2. then **remove.seqs**(fasta=current, name=current) removes any duplicate sequence;
3. at last **dist.seqs**(cutoff=0.06, processors=<available processors>, calc=onegap, countends=F) calculates the distance between all reference sequences, so that later on they can be reused when new genomes are submitted and annotated. The pairwise relative distances of all sequences from each other were saved in the distance matrix for pairs of sequences with no more than 6% sequence divergence, to save space (parameter cutoff). Testing different values, we found that those higher than 8% required substantially a larger amount of disk space, while for our benchmark datasets (comprising 1,000 sequenced genomes with detected marker genes) and newly generated genome collection it was highly unlikely that any new marker gene sequence was less similar than that to a sequence of the reference collection. Keeping the distances between all sequences would require more than a terabyte of space uncompressed in the format used by mothur. Calculating the distances between the reference sequences took more than a week, using ~50 cores. This process generated a mothur alignment file, a name file, a distances file and a mapping file that maps the sequence IDs to NCBI IDs.

Step 2. Finding and extracting marker genes

This is the first automated step in the taxonomic annotation pipeline that was applied to the newly sequenced microbial genomes. Marker genes were detected in the input genome sequences using HMMER and extracted from the detected regions within a genome. Here, 'RNAmmer 1.2' (www.cbs.dtu.dk/services/RNAmmer) using HMMER 2.0[9,10] with 16S profiles from 2006 was used and found to be more sensitive than HMMER 3.0[3,4,5] with a reference collection from 2010. A minimum sequence length for further analyses of the identified 16S sequences was set to 900 base pairs, meaning that any shorter sequence was discarded, as highly conserved regions within a marker gene can get very high alignment scores, even for very distant relatives.

The output of this step is a single FASTA formatted file containing all marker gene sequences.

Nature Methods: doi:10.1038/nmeth.4458

Step 3. Alignment and clustering

The alignment of the identified marker genes to the SILVA reference sequences (step 1) was performed with mothur[8] Distances were calculated from global pairwise alignments generated with the parameter 'gotoh', and the one gap penalty for indel gaps, not counting ends. mothur was then used to update the existing distance matrix with the values for the identified marker genes, as outlined above. The cutoff was the same as in step 1, in this case 0.06. Based on the genetic distance obtained for all marker genes, including the reference genes, an agglomerative clustering using the furthest neighbor algorithm[11] was applied, using 0.06 as cutoff for cluster determination.

mothur *commands*

- **unique.seqs**(fasta={fasta_file_path_with_unaligned_marker_genes})
- **align.seqs**(candidate=current, template={aligned_silva_reference_fasta_file_path}, align=gotoh, flip=t, processors=<processors>)
- **remove.seqs**(accnos={filename}.unique.flip.accnos, fasta=current, name=current)
- **merge.files**(input={filename}.names-<silva_reference_names_file_path>, output={filename}.merged.names)
- **merge.files**(input={filename}.pick.names-<silva_reference_names_file_path>, output={filename}.merged.names)
- **set.current**(name={filename}.merged.names, column={local_copy_of_silva_reference_distance_file_path})
- **dist.seqs**(oldfasta={aligned_silva_reference_fasta_file_path}, column=current, cutoff={cutoff}, processors={processors}, calc=onegap, countends=F)
- **set.current**(column={local_copy_of_silva_reference_distance_file_path})

Nature Methods: doi:10.1038/nmeth.4458

- **cluster.split**(cutoff={cutoff}, method={method}, precision={precision}, column={local_copy_of_silva_reference_distance_file_path}, name={filename}.merged.names)

The outputs of this procedure are clusters of 16S genes at multiple distances thresholds, up to the cutoff value. The precision parameter (set to 1000) specifies the distance steps that are considered, which means that clusters were generated for every 0.001 increase in distance.

Step 4. Annotation

Based on the clustering of marker genes, the input genomes were then taxonomically classified, organized in OTUs and placed in novelty categories.

Taxonomic classification

Genomes were taxonomically classified using the clusters that they were placed in based on their 16S gene. As the clusters are available for multiple distances, the classification is performed using the first cluster distance setting in which the cluster allows an informative assignment, defined as following: starting from a minimum distance of 0.02, a given cluster is inspected for the taxonomic labels of the included reference gene sequences. Then, the lowest ranking taxon on the respective taxon paths on which at least 90% of cluster reference sequences agree on, is assigned. If, for a given threshold and cluster, there is less agreement even at the rank of domain, the cluster at the next distance threshold is inspected, until a taxon assignment has been made.

Novelty category

A novelty category reflects how closely a query genome is related to sequenced genomes in a reference database (as opposed to the depth of the taxonomic classification assigned). To this end, for every newly sequenced genome used for generation of the challenge datasets, we tested how many sequenced genomes were available in the assigned taxon from a reference collection of 14,957 draft and complete genomes that we compiled from NCBI: RefSeq (microbial part), whole bacterial genomes, draft bacterial genomes and the Human Microbiome Project. The

Nature Methods: doi:10.1038/nmeth.4458

search was started from the taxon assignment at the lowest assigned rank, e.g. the species rank, and by successively moving up the taxonomic path, e.g. to the ranks of genus, family, order, class, phylum and domain, respectively, in case no reference genome was found. If a genome was found, the novelty category assignment was to “new” rank visited below. For example, if a genome was classified on species level and a whole genome in the reference of the same species exists, the novelty was declared as 'new_strain'. If no reference genome of the same species was found, but there was a reference genome of a different species of the same genus, the novelty category will be 'new_species' and so on.

OTU inference

The OTU assigned to each marker gene corresponds to the clustered group at the distance threshold of 3%, which is later used in the metagenome sample simulation.

Output

The output of the taxonomic classification pipeline is a metadata table mapping each genome to a numeric NCBI ID, a novelty category and an OTU membership (Supplementary Table 10).

Quality control

A quality control of the assembled genomes was performed, based on tetranucleotide content analysis, to remove contaminated samples including multiple isolates were removed (24 genomes of the cultured isolates from environmental samples) and taxonomic analyses. The taxonomic assignments were manually inspected, and genome entries with multiple 16S marker gene assignments (48 of environmental genomes, 1 of culture collections) or no marker genes (28 of environmental genomes). The provided taxonomic classification of nine sequenced genomes from the culture collections were in conflict with the inferred ones on 'family' to 'phylum' level, thus these genomes were excluded. The taxonomic classification for the other culture collection genomes were assumed to be correct and used. This included 14 cases of minor taxonomic discrepancies on genus and species levels. This resulted in 388 of 464 genomes of cultured isolates from environmental samples and 301 of 311 of the genomes from the culture collections that were maintained. The novelty categorization was repeated for these genomes using their original taxonomic classifications.

Nature Methods: doi:10.1038/nmeth.4458

The genetic sequences of 26 viruses, 240 plasmids and 332 unknown circular elements were manually assigned to these three groups that were used as OTU and taxonomic novelty categories for the analysis pipeline.

1.1.3 Metagenome simulation

We created three kinds of simulated metagenome datasets in the CAMI challenge (Supplementary Table 3, Supplementary Figs. 47-48): a single sample dataset of a low complexity community (40 genomes and 20 circular elements), a differential abundance dataset with two samples of a medium complexity community (132 genomes and 100 circular elements) and two insert sizes and a time series dataset with five samples from a high complexity community (596 genomes and 478 circular elements). All simulated paired-end sequencing with an Illumina HiSeq System. To generate the three datasets and their samples, the following steps were undertaken: after genome data validation (step 1), the community composition was designed according to specified criteria (step 2) and the metagenome datasets simulated (step 3). The creation of the gold standards (step 4) represents the last part of the pipeline. This analysis was automated with codes available at https://github.com/CAMI-challenge/MetagenomeSimulationPipeline/releases/tag/0.1.

Step 1. Data preprocessing and validation

All sequences shorter than 1000 base pairs were removed from the provided genome assemblies and sequences validated to contain only valid characters. The input genomes can be draft genomes in FASTA format and as such were allowed to contain beside the bases ´A´, ´C´, ´G´, ´T´ also ambiguous DNA characters, such as ´R´, Y´, ´W´, ´S´, ´M´, ´K´, ´H´, ´B´, ´V´, ´D´, ´N´.

Step 2. Community Design

Genome selection

In this step, a specified number of sequences (either circular elements or genomes) is selected according to certain criteria, see below, from all input sequences (e.g. circular elements or genomes) to generate a specific dataset. All sequences of genomes and circular elements were assigned to only one of the three simulated datasets. The random selection of genomes for a particular dataset was performed based on their membership in a novelty category (Taxonomic annotation step 4) and OTUs: to increase the taxonomic spread, for a specified overall number of genomes

Nature Methods: doi:10.1038/nmeth.4458

to be included in a particular dataset, the selection algorithm attempts to draw the same number of genomes from each novelty category (new: strain, species, genus, family, order). If a category does not have enough genomes to thus enable coming up with the specified genome number, more genomes are drawn from the other categories, in equal amounts, if possible. This is repeated until the specified number of genomes has been sampled. To model strain level diversity within a species, for every OTU, all available genomes are included in one particular dataset up to five genomes at most. More than five genomes per OTU were included, if the specified dataset size required more genomes to be added that were not available otherwise.

To increase strain level diversity in the datasets, a number of additional genomes were simulated with sgEvolver[12], representing related strains to the genomes selected above (see below). For this purpose, a genome was randomly chosen, and a strain number was drawn from a geometric distribution (with parameter $p = 0.3$), representing the number of related strains to be created for that particular genome. This was repeated, until the specified overall number of strains to be generated was reached. For each of these genomes, sgEvolver simulated 40 strains, with an increasing genetic distance (in steps of 0.1%) to the preceding ones, up to a total distance of 4%. From the 40 strains, the specified strain number was randomly selected and added to the genome set.

Thus, 28 original genomes selected for the low complexity dataset, together with 12 simulated strains, resulted in an overall dataset size of 40 genomes. Twenty of the circular elements were added. The medium complexity dataset consists of 132 genomes, of which 13 were simulated, and 100 circular elements. The high complexity dataset comprises 542 genomes, 54 simulated genomes and 478 circular elements (Supplementary Table 3).

Creating the community abundances

For dataset simulation, genome and circular element (plasmid, viruses, other circular elements) abundance distributions are required for every dataset. For the single sample dataset, the abundance distribution was created by sampling 28 times (number of genomes) from a log-normal distribution, with parameters: mean = 1 and standard deviation = 2. For the medium complexity dataset, reflecting a differential abundance experiment, two abundance samples of sizes 119 were drawn independently from a log-normal distribution with the same parameter settings. For the high complexity dataset, sampling from a log-normal distribution 544 times with mean = 1.5 and standard deviation = 1 was initially used. For each five consecutive samples, abundances for the next sample were calculated by sampling new values

Nature Methods: doi:10.1038/nmeth.4458

from the log-normal distribution, adding these to the previous value and dividing by two. From the same distributions, circular elements were sampled, with 20 circular elements sampled for the low complexity sample, 100 for the medium complexity and 478 for the high complexity one. The absolute abundance of all circular elements together was then set to be 15 times as high as the abundance of the microbial genomes together, to replicate high natural plasmid abundances (parameter ratio=15). For each sample and the pooled data sets, a gold standard taxonomic profile for taxa at all analysed ranks (strains, species, genus, family, order, class, phylum and domain) was calculated based on the genome abundance values and their taxonomic IDs. Lack of a taxon label at a particular rank for genomes results in classification as "unassigned" for that rank. Additionally, a filtered version of this gold standard was created in which unassigned sequences, i.e. plasmids and viruses were excluded. The taxonomic profile gold standards are available at https://data.cami-challenge.org/participate with the file format described under https://github.com/CAMI-challenge/contest_information/blob/master/file_formats/CAMI_TP_specification.mkd.

Step 3. Metagenome sample simulation

Based on the genome abundance distribution, the genome sizes and the specified metagenome sample output size, for every genome, the coverage in the simulated samples is calculated. The ART read simulator[13] then generates tandem-oriented paired-end reads, sampling each genome proportionally to the specified abundance in the community. Paired-end reads were simulated to represent Illumina data using a high-throughput profile for 150 bp long reads. All reads were generated with a mean insert size of 270 bp and a standard deviation of 10%. The read simulation of the medium complexity dataset was run a second time using a mean insert size of 5000 bp using the same genome abundances as before. The size of every sample of each dataset was set to 15 Gb. Except for the samples with the insert mean size of 5000 bp, which were set to 5 Gb. Outputs of this step are for every sample of a dataset, a FASTQ and a SAM file, which specifies the location of every read in the input sequences.

Step 4. Creation of the gold standards and anonymization

A gold standard assembly was created with SAMtools[14] for each sample individually and for all samples of each dataset together, which included every genome and circular element sequence position with a coverage of at least one, respectively.

Nature Methods: doi:10.1038/nmeth.4458

Specifically, the SAM file specifying the location of each read on the reference genomes was used with SAMtools for calculation of the per-site coverage, thus every part of the genome not covered by at least one read is not part of the gold standard assembly.. Sequences were broken into gold standard contig sequences at zero coverage sites. The contig sequences were then labeled with a unique dataset and sample, identifying prefix and an increasing index as suffix. The gold standard assembly is found under https://data.cami-challenge.org/participate and the format is described under https://github.com/CAMI-challenge/contest_information/blob/master/file_formats/CAMI_A_specification.mkd.

The binning gold standard was also created in this step for the read and assembled samples. It includes for every sequence entry two columns, namely BINID and TAXID, where BINID describes the genome it belongs to and is used for evaluating the genome binners and TAXID is the NCBI taxonomic ID of that particular genome and is used for evaluating taxonomic binners. Finally, to evaluate the effectiveness of genome recovery with taxon assignments (also in the ideal case of the gold standard not always individual strains, but oftentimes species or above), we evaluated the taxon-assignments from the TAXID column as a gold standard for taxonomic binning to resolve genome bins (section 1.4.4.). The exact format of these files is described under https://github.com/CAMI-challenge/contest_information/blob/master/file_formats/CAMI_B_specification.mkd and is available under https://data.cami-challenge.org/participate.

In the last step, sequences of all simulated datasets were shuffled using the UNIX command 'shuf' and anonymized, thus creating the challenge datasets.

1.1.4 Strain genome evolution

Strain genomes were created with an extension of the simulated genome evolution process first described in [12] and implemented in the sgEvolver software. Strain simulation requires as input a fixed phylogenetic tree topology with branch lengths (Fig. S1) along with an ancestral genome sequence and a set of parameters that control evolutionary events. Evolution is simulated in the usual manner as a marked Poisson process. For single nucleotide substitutions, these are simulated according to the F81 model, with locations chosen uniformly at random throughout the genome. Use of the F81 model ensures that the genomic G+C content of the ancestral genome is maintained in simulated descendants. The expected number of single nucleotide substitutions on a branch of the phylogeny is given by the branch length. Small insertions and deletions were simulated to occur with a rate of 0.05 relative to the substitution rate. As with nucleotide substitutions, indel positions were chosen uniformly throughout the genome and indel lengths were Poisson-distributed with a

Nature Methods: doi:10.1038/nmeth.4458

mean of 1. Our simulation of genome evolution also included the large-scale gain and loss of genetic material, as would be created by the natural process of horizontal gene transfer. Losses were simulated as sequence deletions with a mixture of geometric-distributed lengths (rate = 1/200) and uniform lengths in the range [10000, 60000]. Gains were simulated by maintaining a separate pool of donor genomic DNA, from which segments were chosen uniformly at random, with length distributions identical to those used for losses. In all cases, genomic positions of these events were chosen uniformly at random. Small gains and losses with lengths distributed with a geometric distribution (Geom(1/200)) were simulated to occur at a rate of 0.05 relative to nucleotide substitution events, while large gains and losses were simulated with a relative rate of 0.005. Finally, genome rearrangements were simulated as inversion events, with endpoints chosen uniformly, lengths chosen from a geometric distribution with mean 50000, and a rate of 0.005 relative to substitution events.

As originally implemented, the simulated strain evolution process did not model natural selection. The lack of modeled selection can cause the Open Reading Frames (ORFs) of protein encoding genes to break, for example by introduction of a frameshift indel or a premature stop codon, and these broken genes can accumulate quickly even over modest degrees of simulated divergence. Because some metagenomics analysis pipelines might depend on identification of ORFs in the data, the sgEvolver software was extended to include a simulated natural selection step. The sgEvolver was provided with a gene annotation created with Prodigal (Prokaryotic Dynamic Programming Genefinding Algorithm, http://prodigal.ornl.gov/) for the genomes and set to lower mutation rates within these gene locations. We modeled two types of selection: first, substitutions that would introduce a stop codon (assumed here to be TGA, TAA, or TAG) were rejected with a probability of 98%. Second, substitutions that would eliminate an existing in-frame stop codon were rejected with 100% probability. We did not model selection on frameshift mutations introduced by indels, nor ORFs broken by the large scale acquisition or loss of genetic material.

1.1.5 Evaluation nomenclature

Some contest participants analyzed all samples from the different datasets individually. Others combined all samples from an individual dataset and returned one result file for each dataset. The combined samples are referred to as “pooled” samples. For terminology used in the evaluation of the effects of plasmid and viral sequence material on performances, please refer to section 5.3.4.

Nature Methods: doi:10.1038/nmeth.4458

1.1.6 Average Nucleotide Identity (ANI) between reference genomes

For all pairs of reference genomes we calculated the Average Nucleotide Identity (ANI) using pyani (https://github.com/widdowquinn/pyani). The ANIs were used to group the reference genomes into two groups: the first group (unique) contains reference genomes for that the ANI of all pairs of genomes is < 95%. The second group (strains) contains all genomes where a closely related genome with an ANI ≥ 95% exists.

1.2 Evaluation of Assembly Software

Results for all assemblers, except for A*, for which assemblies were submitted separately for each high complexity sample, were submitted for the pooled samples of the three datasets. We used the specified parameters for A* to assemble the pooled high complexity dataset. The gold standard assembly was evaluated in parallel to the challenge submissions and indicates the upper or lower bound attainable with the different performance and error metrics. Results of the assemblers on the different datasets can also be found at https://github.com/CAMI-challenge/firstchallenge_evaluation/tree/master/assembly.

Assemblies were analyzed with MetaQUAST version 4.0[15]. MetaQUAST maps contigs of an assembly to the combination of all reference genomes using MUMmer[16]. Next, MetaQUAST aligns with MUMmer contigs against reference genomes that aligned in the previous “combined reference” step. In the last step, MetaQUAST produces metrics for the unaligned contigs of the previous steps.

Contigs shorter than 500 were excluded (**--min-contig 500**). If contigs mapped equally well to different reference genomes, only one alignment was retained (option **--unique-mapping).**

We used the folder structure of the MetaQUAST output to extract assembly metrics for each reference genome. We combined these metrics with the coverage information of each reference genome and the similarity of the genomes by using the Average Nucleotide Identity (ANI) score. To allow a reproduction of these results, MetaQUAST and the scripts that operate on top of MetaQUAST (QUAST-evaluator[17] https://libraries.io/github/pbelmann/quast-evaluator) are available as Docker containers (Supplementary Table 9).The MetaQUAST Docker container also implements the bioboxes specification[18].

Nature Methods: doi:10.1038/nmeth.4458

1.2.1 Metrics

The following metrics were used for evaluation of assemblies (descriptions adapted from the QUAST website at http://quast.bioinf.spbau.ru/manual.html):

- # **contigs:** total number of contigs in the assembly;

- **Total length / total assembly size:** total number of bases in the assembly (sum of all contig lengths, including misassembled contigs).

- # **misassemblies:** number of positions in the contigs that satisfy one of the following criteria:

  * the left flanking sequence aligns over 1 kbp away from the right flanking sequence on the reference;

  * flanking sequences overlap on more than 1 kbp;

  * flanking sequences align to different strands or different chromosomes;

  * flanking sequences align on different reference genomes.

- # **unaligned length / unaligned bases:** the total length of all unaligned regions in the assembly (sum of lengths of fully unaligned contigs and unaligned parts of partially unaligned ones).

- **Genome fraction (%):** the percentage of a particular reference genome covered by aligned contigs (in bases) from the assembly. A base in the reference genome is considered aligned, if there is at least one contig with at least one alignment to this base. Contigs from repetitive regions may map to multiple places, and thus may be counted multiple times.

Nature Methods: doi:10.1038/nmeth.4458

- **Duplication ratio:** the total number of aligned bases in the assembly divided by the total number of aligned bases in the reference genome (see Genome fraction (%) for the 'aligned base' definition). If the assembly contains many contigs that cover the same regions of the reference, its duplication ratio may be much larger than 1. This may occur due to overestimating repeat multiplicities and due to small overlaps between contigs, among other reasons.

- # **mismatches per 100 kbp:** the average number of mismatches per 100000 aligned bases. True SNPs and sequencing errors are not distinguished and are counted equally.

- # **predicted genes unique:** the number of unique genes in the assembly found by MetaGeneMark (https://www.qiagenbioinformatics.com/plugins/metagenemark/)[19].

- **NA50:** the length for which the collection of all aligned blocks of that length or longer covers at least half an assembly. Aligned blocks are obtained by breaking contigs at misassembly events and removing all unaligned bases.

- **NGA50:** the length for which the collection of all aligned blocks of that length or longer covers at least half the reference genome. Aligned blocks are obtained by breaking contigs at misassembly events and removing all unaligned bases.

- **Mapping statistics:** we mapped all reads back to the assembled contigs to determine the fraction of reads included in the different assemblies. We used bowtie2 v2.2.7[20] with option '--end-to-end' to generate a SAM file for each assembly, followed by using SAMtools v1.3[14] with option 'flagstat' to extract mapping statistics from the SAM files.

1.3 Evaluation of Profiling Software

Results for most programs were submitted for every sample from the three datasets individually. For several programs originally designed as taxonomic binners, such as CLARK[21], individual read predictions were summarized and submitted as a taxonomic profile. Each of the error metrics described below was determined for

Nature Methods: doi:10.1038/nmeth.4458

every submission (representing one profiler with specific parameter settings), and at each taxonomic rank down to the species level; except for Unifrac, which is rank independent. A ranking of tools was performed for all programs where results for the individual samples were available, which was the case for all except for CK_v1[22], FOCUS_v0-v4,v6[23] and DUDes_v0-v1[24] (Supplementary Table 7). Strain level comparisons of taxonomic profilers were not performed, since individual strains do not have reference taxonomic IDs, thus precluding a comparison of predictions to the ground truth.

1.3.1 Metrics

We detail here the precise definition of the metrics utilized for assessing the taxonomic profiling results.

L1 Norm

The L1 norm predominantly measures the accuracy of reconstructing the relative abundance of organisms in a sample. For r, a particular taxonomic rank, let x_r be the actual bacterial frequencies at rank r. That is, x_r is a vector indexed by all taxa at rank r, with entries $(x_r)_i$ given by the relative abundance of taxa i in the sample. Let ${x^*}_r$ be the vector of predicted bacterial frequencies at rank r. Then the L1 norm at rank r is given by:

$$||x_r - {x^*}_r||_1 = \Sigma_i |(x_r)_i - ({x_r}^*)_i|$$

The L1 norm thus gives the total error between the true and predicted abundances of the taxa at rank r. A lower L1 norm indicates that a particular taxonomic profiling algorithm accurately reconstructs the relative abundances of organisms in a sample. The L1 norm ranges between 0 and 2.

Unifrac distance

The weighted Unifrac distance was originally defined in [25] and measures the taxonomic similarity of metagenomic communities. We refer the reader to [25] for the exact definition of the Unifrac distance, but give a brief description here. The actual relative abundances of the sample are placed on the appropriate nodes of a taxonomic tree and the reconstructed relative abundances are also placed on the appropriate nodes of the same taxonomic tree. The Unifrac distance is the total amount of reconstructed relative abundance that must be moved (along the edges of

Nature Methods: doi:10.1038/nmeth.4458

the taxonomic tree, with all branch lengths here set to 1) to cause them to overlap with the true relative abundances. We use the EMDUnifrac implementation of the Unifrac distance (https://github.com/dkoslicki/EMDUnifrac, preprint[26]), as it allows the reconstructed relative abundance to be placed on internal nodes of the taxonomic tree (necessary for when classifications are only predicted at higher taxonomic ranks). A low Unifrac distance indicates that a taxonomic profiling algorithm gives a prediction that is biologically similar to the actual profile of the sample. The Unifrac distance ranges between 0 and twice the height of the taxonomic tree utilized (so here a value of 16).

Binary classification metrics

Standard binary classification metrics were also utilized to assess the accuracy of taxonomic profiles correctly identifying the presence and absence of taxa in a sample, without regard to how well the relative abundance is predicted. As before, let x_r be the actual bacterial frequencies at rank r. That is, x_r is a vector indexed by all taxa at rank r, with entries $(x_r)_i$ given by the relative abundance of taxa i in the sample. Let x_r^* be the vector of predicted bacterial frequencies at rank r. Define the support of a vector x as:

$$\mathrm{supp}(x) = \{i \text{s.t.} x_i > 0\}$$

So $\mathrm{supp}(x_r^*)$ gives the indices of the taxa at rank *r* predicted to be in the sample. Define the true positives, false positives, and false negatives respectively as

$$TP = |\mathrm{supp}(x_r) \cap \mathrm{supp}(x_r^*)|$$

$$FP = |\mathrm{supp}(x_r)^c \cap \mathrm{supp}(x_r^*)|$$

$$FN = |\mathrm{supp}(x_r) \cap \mathrm{supp}(x_r^*)^c|$$

Precision, or purity of a taxonomic profile, and recall, or sensitivity, or completeness of a taxonomic profile, are defined by:

$$precision = \frac{TP}{TP+FP} \qquad recall = \frac{TP}{TP+FN}$$

1.3.2 Absolute Performance Plots

To assess the absolute performance of the profilers, we created spider plots (also known as radar plots) for the taxonomic profilers and their versions, labeling the spokes of the plot in an arbitrary order, and individual metrics depicted as colored

Nature Methods: doi:10.1038/nmeth.4458

lines connecting the spokes. The scale on the plots indicates the value of the error metrics. Since the error metrics vary in their ranges, the Unifrac, True Positives and False Positives metrics are omitted. Additionally, the L1 norm metric was divided by 2, so that it ranged between 0 and 1. Some of the participants combined the multiple FASTA files for a dataset into a single file and used this as input to their program. We indicate these "pooled" samples in the caption of the appropriate figures. Supplementary Figures 29-33 depict these absolute performance plots.

1.3.3 Relative Performance Plots

We used spider plots (also known as radar plots) to assess the relative performance of the taxonomic profilers (and versions). Spokes of the plot are labeled in an arbitrary order and individual metrics are depicted as colored lines connecting the spokes. To plot multiple metrics on each figure, each metric was normalized (divided by) its maximum. Hence there is no scale to the plot as only relative performance is being depicted. By normalizing, the relative strengths and weaknesses of the profilers can be determined, as well as how these strengths and weaknesses change over various taxonomic ranks. If a profiler did not perform a classification at a given taxonomic rank, its name is highlighted in red and two asterisks are added. Some metrics are still shown for such profilers at these ranks, as they are still well defined when there is no classification at that rank. Additionally, some of the participants combined the multiple FASTA files of a sample into a single file and used this as input to their profiler. We indicate these "pooled" samples in the caption of the appropriate figures. Supplementary Figures 34-38 show such relative performance plots (for brevity, only the pooled and one representative sample are shown for the medium and high complexity samples).

1.3.4 Rankings

To get a global sense of relative performance, we ranked the profilers by their relative performance for each metric, sample, and taxonomic rank. Supplementary Figures 22-28 show these results for the low complexity sample, and one sample for each of the medium and high complexity samples. Next, each profiler was assigned a score for its ranking (0 for first place among all profilers at a particular taxonomic rank for a particular sample, 1 for second place, etc.). These scores were then added over the taxonomic ranks. Supplementary Figure 22 demonstrates this procedure. Results were then summed over the samples to give a global ranking/score of the performance of the profilers, and are depicted in Figure 3. Depending on the

Nature Methods: doi:10.1038/nmeth.4458

biological use case, metrics, ranks, or samples can be optionally weighted, as opposed to taking a direct sum, as performed here.

1.3.5 Taxonomic profilers versus taxonomic binners

A taxonomic binning technique can be used to generate a taxonomic profile. Analogous to the coverage-based estimation strategy realized by many profilers, we implemented a simple coverage-estimate based algorithm for assessing taxon abundances from the taxonomic binning results. The idea is that the average coverage of sequences originating from one particular genome can be taken as a proxy of the genome copy number. For bins representing one particular strain genome, the relative abundance can thus be estimated by calculating the fraction that this coverage represents of the sum of the coverages of all bins, including the unassigned one, at a particular taxonomic rank. Some limitations of this approach are that taxonomic bins might integrate data from multiple strains, or data of one strain could be distributed across many bins, or not assigned at all. It is therefore anticipated that as the quality and resolution of the taxonomic binning results improve, so should the profiling estimates.

The resulting relative abundances at each rank give the resulting taxonomic profile.

Algorithm

1. We begin with the lowest rank of the reference taxonomy used, which is in this case the rank of species. For every bin b_i at species level, including the "unassigned" bin, we calculate the average coverage per basepair as:

$$bcov_{b_i} = \frac{\sum_{c \in b_i} cov_c \cdot l_c}{\sum_{c \in b_i} l_c}$$

with $bcov_{b_i}$being the average coverage per base pair of a bin, but only considering the contigs directly assigned to that bin, cov_c the coverage of contig c and $l_c$the length of that same contig.

2. We can then calculate the coverages for all bins b_i on the higher taxonomic ranks, which are composed of the coverage of the sequences in the bin itself (calculated as in 1.) and the coverage of all bins with b_i as lowest common ancestor in the taxonomy

Nature Methods: doi:10.1038/nmeth.4458

$$cov_{b_i} = bcov_{b_i} + \sum_{b \in childbinsof\, b_i} cov_b$$

3. For every rank, we calculate the relative abundances of all bins b_i at this rank as:

$$ra_{b_i} = \frac{cov_{b_i}}{\sum_{b \in binsatrank(b_i)} cov_b}$$

where $rank(b_i)$denotes the taxonomic rank of bin b_i.

The resulting relative abundances at each rank give the resulting taxonomic profile.

Algorithm Pseudocode

```
# sequence might be reads or contigs depending on taxonomic binner
# taxID(sequence) maps every  sequence to its NCBI taxonomy ID
for seq in gold_standard:
      if seq in submission:
            bins[seq.taxID].length += seq.length
            bins[seq.taxID].abundance += seq.coverage * seq.length
      else:
            unassigned.length += seq.length
            unassigned.abundance += seq.coverage * seq.length
taxonomy = {species, genus, family, order, phylum, superkingdom}
for rank in taxonomy:
      ids = rank.taxIDs # all taxIDs who are at level "rank"
      for id in ids:
            bins[id].coverage = bins[id].abundance / bins[id].length
            rank.total_coverage  +=  bins[id].coverage  #  for  abundance
calculation
            lineage = id.lineage # all taxIDs up the tree to superkingdom
            for linid in lineage: # propagate length of bin to higher level
bins
                  bins[linid].length += bins[id].length
```


Nature Methods: doi:10.1038/nmeth.4458

```
                bins[linid].abundance += bins[id].abundance
    for id in ids:
        bins[id].percentage = bins[id].coverage / rank.total_coverage
```

1.4 Evaluation of Genome and Taxonomic Binning Software

Two kinds of binning programs participated in the CAMI challenge, namely taxonomic binning methods that annotate reads or contigs with taxon IDs and binning methods, which group the sequences into bins without assignment of a taxonomic label, aiming to recover individual genomes. For performance evaluation, we applied the measures explained in the following sections. All calculated performance measures can also be found at https://github.com/CAMI-challenge/firstchallenge_evaluation/tree/master/binning. For evaluating genome binners, these are also provided in an easy to use genome binning evaluation package named AMBER (https://github.com/CAMI-challenge/genome_binning_evaluation).

1.4.1 Taxonomic Binning Evaluation

For taxonomic binners, precision (purity) and recall (completeness) were calculated for taxa at the major NCBI taxonomy ranks: species, genus, family, order, class, phylum and superkingdom.

The precision measures the purity of a predicted taxon bin.

$$precision_i = \frac{TP_i}{TP_i + FP_i}$$ (Equation 1.1.1),

where the true positives TP_i are the correct assignments to the i^{th} bin and the false positives FP_i the incorrect assignments to the same bin. For falsely predicted bins, which do not occur in the data, the number of true positives and thus the precision equals zero.

The recall (or sensitivity) is a measure of completeness of the i^{th} real taxon bin.

$$recall_i = \frac{TP_i}{TP_i + FN_i}$$ (Equation 1.1.2),

Nature Methods: doi:10.1038/nmeth.4458

where the false negatives FN_i are the assignments belonging to the i^{th} taxon that were classified to another bin or left unassigned.

To account for uneven taxonomic composition in evaluation datasets and to obtain comparable performance estimates across datasets of different taxonomic composition, we calculated the average precision, also known as the macro-averaged precision[27], [27] either for all predicted bins, or for all predicted bins at a particular rank.

$$average\ precision = \frac{1}{N_p}\sum_{i=1}^{N_p} precision_i \quad \text{(Equation 1.1.3),}$$

where N_p is the number of all predicted bins. The average precision is the fraction of correct sequence assignments of all assignments to a given taxonomic bin, averaged over all predicted bins (for a given rank), where unassigned data is not considered. This value reflects how trustworthy the bin assignments and bins are on average from a user's perspective.

We observed that some programs produced many false positive bins with zero precision but small size (in total length bp). To make the results of those programs more comparable to the others, we calculated a truncated average precision value, where we removed a certain percentage of predictions, corresponding to the sum of the smallest predicted bins. For instance the 99% truncated average precision was calculated by sorting the bins according to their predicted size (in bp) and retaining all larger bins which fall into the 99% quantile, including (equally sized) bins which overlap the threshold.

$$truncated\ average\ precision_\alpha = \frac{1}{N_{r,\alpha}}\sum_{i=1}^{N_p} precision_i \quad \text{(Equation 1.1.4),}$$

where $N_{r,\alpha}$ is the number of bins after applying the α percentile bin size threshold.

Similar to the average precision, we calculated the average recall, or macro-averaged recall[27], as the fraction of correctly assigned sequences for a bin, averaged over all truly existing bins in the test data.

$$average\ recall = \frac{1}{N_r}\sum_{i=1}^{N_r} recall_i \quad \text{(Equation 1.1.5)}$$

N_r is the number of all real bins in the test data. The average recall reflects the average completeness for all real bins.

We furthermore calculated the overall accuracy and misclassification rate for the

Nature Methods: doi:10.1038/nmeth.4458

individual samples, where performance was not averaged over bins, but over assignments of every sequence in bp. Thus, large bins are given more importance in this calculation than small bins.

$$accuracy = \frac{T}{T+F+U} \quad \text{(Equation 1.1.6)}$$

$$misclassification\ rate = \frac{F}{T+F} \quad \text{(Equation 1.1.7)}$$

where T is the sum of sequences in bp which is assigned to the correct bin, F those bps that are assigned to an incorrect bin and U the rest of the data, which is left unassigned.

1.4.2 Genome Binning Evaluation

The bins returned by genome binners typically have no taxonomic label attached. For evaluation of taxonomic binners and genome binners, we calculated the precision (purity; eq. 1.1.1) for every predicted genome bin and the recall (completeness; eq. 1.1.2) for every genome truly present in the benchmark datasets. For these calculations it was necessary to perform a mapping of the genome bins returned by the programs to the original genomes. For every predicted genome bin, the length of all contigs in that bin belonging to the same genome was added up and the bin assigned to the genome to which the most base pairs belonged to, i.e. that was most abundant in the bin in terms of bps. The script calculating this mapping, as well as the following precision and recall calculation, can be found under https://github.com/CAMI-challenge/bbx-binning-evaluation-jd/blob/unstable/opt/cami-evaluate-binning/bin/genome_bins_2.py.

Using this approach, it is possible that multiple genome bins map to the same genome, as well as that some genomes are not being mapped to any bins. In the latter case, recall is 0 for that genome. In the first case, multiple precision (eq 1.1.1) and recall (eq 1.1.2, adapted to considering multiple genome-bin mappings instead of a genome only once) values for a single genome or genome bin are obtained, for every bin which mapped to that genome separately. Thus multiple bins might have a precision of 1.0 for a genome, but the recall for those bins sum up to at most 1.0. This is because recall is calculated by summing up the length of the sequences assigned to that certain bin, divided by the total length of the genome, and every sequence of a genome can be assigned to at most one bin. Using the calculated values for every predicted genome bin, the macro-averaged precision and macro-averaged recall[27] were calculated, as in equations 1.1.3 and 1.1.5. These measures

Nature Methods: doi:10.1038/nmeth.4458

represent the average genome bin purity or genome bin completeness for the respective dataset, with small bins contributing in the same way as large bins to the metric. In these calculations bins representing viruses, circular elements and plasmids were not considered.

In addition, the Adjusted Rand Index (ARI) was calculated, as in [28]. The ARI is a measure of assignment accuracy derived from the Rand Index. The Rand Index compares two clusterings of items from a dataset to each other and determines the extent of their similarity. Here, the clustering of metagenome sequences into predicted genome bins is compared to their underlying correct allocation to the genomes that they originate from. If two sequences from a genome are placed in the same predicted genome bin, these are considered to be true positives. If two sequences from different genomes are placed in different bins, these are considered true negatives. The Rand Index ranges from 0 and 1 and is the sum of true positive and true negative pairs divided by the sum of all possible pairs of sequences, where larger values indicate a higher similarity of the two clusterings. However, for a random clustering of the data set, a Rand index larger than 0 would be returned. The Adjusted Rand Index corrects for this by subtracting the expected value for the Rand Index and normalizing the resulting value such that the values still range from 0 to 1. In our adaptation, units of base pairs instead of sequences were considered, thus giving a measure of assignment accuracy per basepair.

Some genome binning programs exclude sequences from bin assignment, thus assigning only a subset of the sequences from a given dataset. This complicates assessment with the ARI, as the ARI has no concept of unassigned items. When calculating the ARI, we thus considered only the proportion of data that was binned, with results reflecting the assignment accuracy for the assigned portion of the dataset only. Thus, these values should be taken into consideration together with the portion of the sample that was assigned (Fig. 2**a**).

1.4.3 Clarifying the meaning of taxonomic binning metrics with a toy example

Consider a simple dataset, consisting of three sequences from three different strains, one bacterial one with 100 kb length, one archaeal one with 10 kb in length and a viral one of 1 kb in length, so the entire dataset comprises 111 kb. Now assume we have a taxonomic classifier which assigns all data to one bin with a taxonomic label of “bacterial”. For simplification, we do not consider that the taxonomic binner may also have performed assignments at lower taxonomic ranks, which would be analyzed in a similar way as the domain, with assignments to taxa of lower ranks

Nature Methods: doi:10.1038/nmeth.4458

being propagated upwards to the respective higher ranking bins in the reference taxonomy. If an assignment is not present at lower ranks, conversely, the data is considered to be placed in the bin “unassigned”.

At domain level, we can now calculate the average precision, recall, misclassification rate and accuracy of assignments for this particular result. For this, we need the correct assignments (in bp), which are 100 kb of 111 kb, and the false assignments, which are 11 kb. The overall accuracy is therefore 100/111 = 0.90 and the misclassification rate is conversely 11/111 = 0.10. Even though the classifier does not seem to do very well in terms of determining what is in a sample (as it identified only one of three taxa correctly), it did assign the large portion of the dataset correctly, which is reflected in a low misclassification rate and a high accuracy. As you can see, these two metrics reflect how good performance is when considering the sample as a whole. However, they are not indicative of performance for different taxa on average, if the taxon abundances in the sample are vary, as in this case. They mostly reflect performance for the more abundant taxa in a sample. To answer the question whether a classifier is doing well for all taxa that are present or were predicted for a sample, we calculate for every bin the precision (reflecting the purity of a predicted bin and absence of contamination) and the recall, which reflects for the true bins in a dataset, how much of these data was assigned the correct taxon ID. In the example above, we have three real bins and one predicted bin. The precision of the predicted bin is thus 100/111 = 0.9 (Bacteria), and as there is just one predicted bin, this is also the average precision. A user thus can expect a bin to represent to 90% sequences of the taxon that it is assigned to. The recalls for the real bins are 100/100 = 1 (Bacteria), 0/1 = 0 (Viruses) and 0/10 = 0 (Archaea). The average recall for these bins is (0+0+1)/3. As you can see, the average recall reflects more clearly than the accuracy that performance of this classifier is not that impressive, in terms of classifying data from all taxa correctly.

Finally, we can evaluate if the contigs from the same predicted bin were grouped consistently together, which is how binners that do not assign taxonomic labels are evaluated. To this end, we find for a particular predicted bin the true bin that is most abundant within it. In this case, in our predicted bacterial bin with 100 of 111 kb being bacterial, the true bin is indeed the bacterial bin. However, the predicted and true bin could also have a different taxonomic label attached, e.g. if the predicted bacterial bin would have most content from an archaeal bin, then these two would be compared, as we do not consider the taxonomic labels in this evaluation. We can now calculate the precision and recall for the predicted bin, similar as above. The precision is 100/111 = 0.9 and the recall is 100/100 = 1 for this bin. The recall for the archaeal and viral bins is 0/1 and 0/10, as nothing was assigned. Doing this for all predicted or

Nature Methods: doi:10.1038/nmeth.4458

real bins, and averaging over the precision or recall results, respectively, delivers us the average precision (0.9) and recall ((0+0+1)/3 = 1/3) in bin reconstruction for this particular binning. Thus, while the predicted bin to 90% includes sequences from the same taxon, which is reflected in the precision, the low recall denotes that it did badly on resolving data from all bins that were actually present in the sample. Note that in this study, grouping of sequences into bins representing individual genomes was evaluated for the genome binners, while in this example, for simplicity we discussed groupings at higher taxonomic ranks.

1.4.4 Strain-level genome bin recovery with taxonomic binners

To determine whether the data partitioning achieved by taxonomic binners can also be used for strain-level genome recovery, we compared predicted taxon bins of all ranks from domain to species (a strain-level rank does not exist in the reference taxonomy) to the genome bins. Genome bin purity (precision) and completeness (recall) for predicted taxon bins were calculated in the same way as for the genome binners. Thus for taxonomic binners, we evaluated the bin quality relative to a reference genome, but not the taxon assignment.

For the taxonomic binners, the completeness was notably lower than for the genome binners - mostly less than 30% – with that of PhyloPythiaS+ (~20-31%) being the highest, while for all others it was below 10% (Fig. S2). The technical limitations of using taxonomic binners for genome bin recovery is evident by the positioning of the taxon bin gold standard – even when performing perfect binning down to the species level, the presence of multiple strains for many species prevents these approaches from achieving high completeness in genome reconstruction. Notably, the purity had a similar range to that of the genome binners. The most precise was Kraken, with mean values of above 80%, closely followed by the others. This finding, however, does not mean that Kraken assigned many taxonomic labels correctly, but rather that it consistently grouped some fragments of the same genome together.

For the taxonomic binners, both genome purity and completeness improved by around 10% when evaluating “unique” strains for all datasets, with a completeness of up to 40% reached by PhyloPythiaS+, while simultaneously showing more than 70% purity (Supplementary Fig. 15). More than 90% purity, though very low completeness (~1%), was demonstrated by Kraken. A similar behavior was shown by MEGAN 6 and taxator-tk, which have methodological similarities (Supplementary Table 1).

For “common strains” both for genome and taxonomic binners, genome purity and completeness dropped notably (Supplementary Fig. 15). PhyloPythiaS+ again had the highest completeness, which was less than 30% though, at lower purity. Purity was down to 70% for the best performing taxonomic binner, taxator-tk. In part, this is

Nature Methods: doi:10.1038/nmeth.4458

expected even under ideal circumstances, as the reference taxonomy does not include a strain rank, with strains being part of the same species bin in the taxonomic binning gold standard. This effect is evident by the varying, and imperfect performance of the gold standard in recovering the binners achieved a better genome resolution than attributed to the gold standard, by assigning genomes of related strains either not at all or consistently to taxon bins at different ranks.

1.4.5 Evaluating taxon binning performance in relation to taxon neighborhood consistency between NCBI taxonomy and SILVA

We hypothesized that taxon binning performance would be affected if the taxon itself underlying genomes for the "unique" and "common" datasets, where it performed well on the first, but poorly on the second. Interestingly, for the common strains datasets, taxonomic would not be monophyletic or if it would be sitting in a badly structured neighborhood in the tree. To investigate this, we compared the NCBI Taxonomy to SILVA release 128 (28.09.2016) the latter is primarily based on molecular evidence, using the strict and loose mapping dissimilarity measures for taxon placement in two taxonomies defined in [29]. We applied this criterion to the parent of the taxon of interest, to include scoring discrepancy of the taxon surroundings. To make the structure of NCBI taxonomy compatible to SILVA, all nodes with no or intermediate rank were removed from NCBI prior to mapping. Specifically, dissimilarities of their taxonomy neighborhood for individual taxa were computed by considering mapping of the taxon itself, as well as its surroundings as follows: Let T be a taxonomy tree and $t \in T$ be a taxon of interest, with $p(t)$ being its parent node. Then we define surroundings $s(t)$ of t as a set of nodes in the subtree of T of height four with $p(t)$ as its root. That is, if $rank(t)$ equals *order*, then $s(t)$ includes $p(t)$ of rank *class* and all the nodes below $p(t)$ with rank *order*, *family* or *genus*, but no nodes with ranks *domain*, *phylum* or *species*. Mapping dissimilarity Q of $s(t)$ onto SILVA was computed as follows:

$$Q(s(t), SILVA) = \frac{\sum_{n \in s(t)} \left(level(n) - level\left(\mu(n)\right) \right)}{\sum_{n \in s(t)} level(n)}$$

Here, $\mu(n)$ is a node in the SILVA taxonomy that n was mapped to with either strict or loose mapping criteria, and $level(n)$ is a numerical expression of $rank(n)$ defined as $level(root) := 0, \quad level(domain) := 1, \quad level(phylum) := 2, \quad level(class) := 3,$

Nature Methods: doi:10.1038/nmeth.4458

$level(order) := 4$, $level(family) := 5$, $level(genus) := 6$ and $level(species) := 6$[1]. The values of $Q(s(t), SILVA)$ always fall into the range of $[0,1]$.

Calculating the correlation of this neighborhood disagreement between NCBI and SILVA with taxon binning classification performance (precision plus recall, averaged over all tools as samples per taxon) using the Spearman correlation coefficient showed a significant negative correlation (q-value < 0.05) using one or the other dissimilarity measure for the species, genus and family ranks (Supplementary Table 8). In other words; for the lower ranks, taxon binning performance significantly decreased for taxa located in discrepant neighborhoods in SILA and NCBI; indicating that use of the SILVA taxonomy could improve taxon binning performance. For order, a positive correlation was found, indicating that the NCBI taxonomy structure is favorable for taxon binning at this rank over SILVA, while no significant differences were found for higher ranks (class, phylum and domain).

[1] Species is assigned the same level as genus as in the SILVA database the lowest rank is genus, therefore species can only be mapped to genus and we do not want to penalize mappings of species.

Nature Methods: doi:10.1038/nmeth.4458

2 Figures

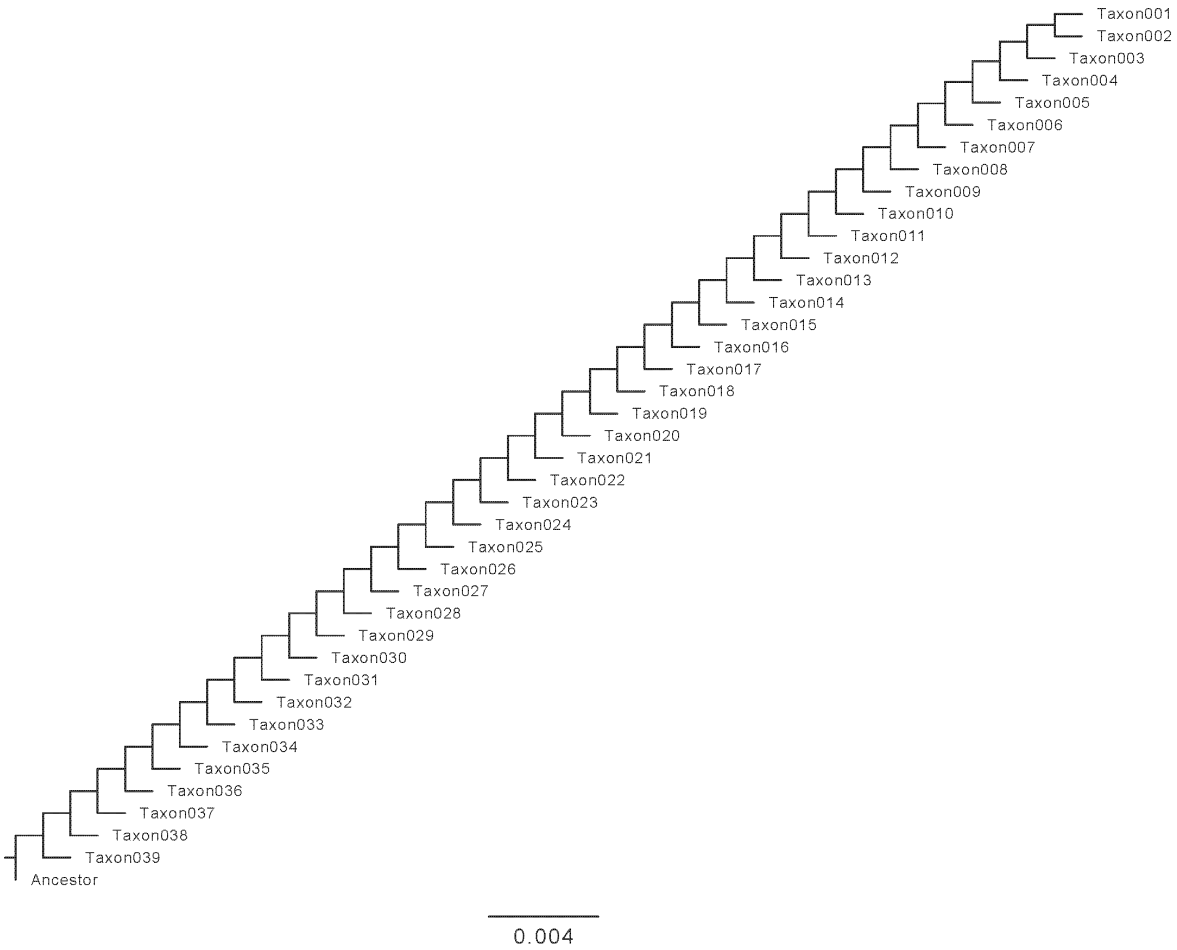


Figure S1. Ladder tree topology used by the strain evolver in the generation of the simulated benchmark datasets.

Nature Methods: doi:10.1038/nmeth.4458

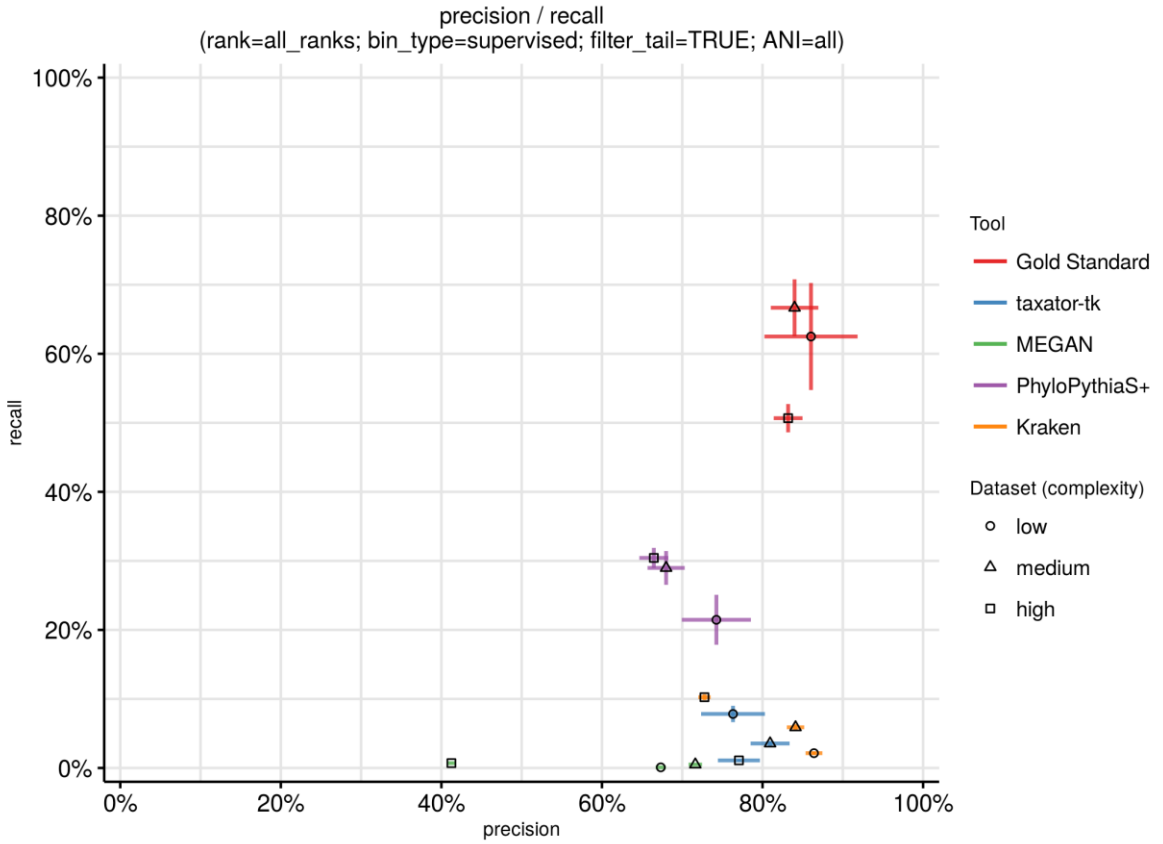
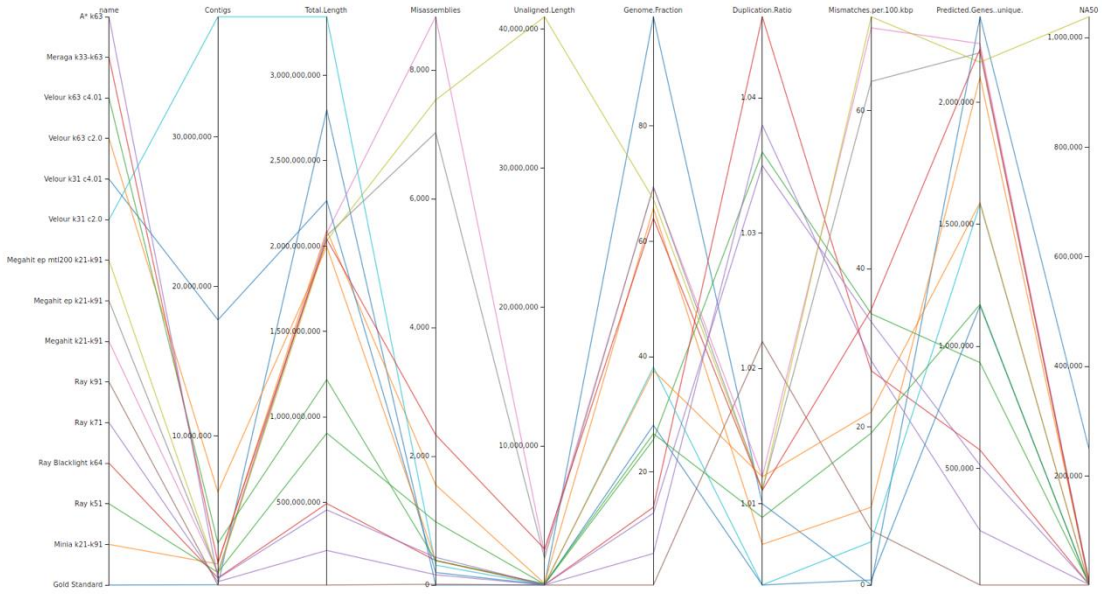


Figure S2. Average precision (or genome bin purity; x-axis) and recall (genome bin completeness; y-axis) for taxonomic binners by genome, for all genomes. For each taxonomic binner and complexity dataset, the submission with the largest sum of precision and recall is shown. Bars denote the standard error of the mean across genome bins. In each case, small bins adding up to 1% of the dataset size overall were removed. For the underlying data, see "Figure_S2_Source_data.xls".

Nature Methods: doi:10.1038/nmeth.4458

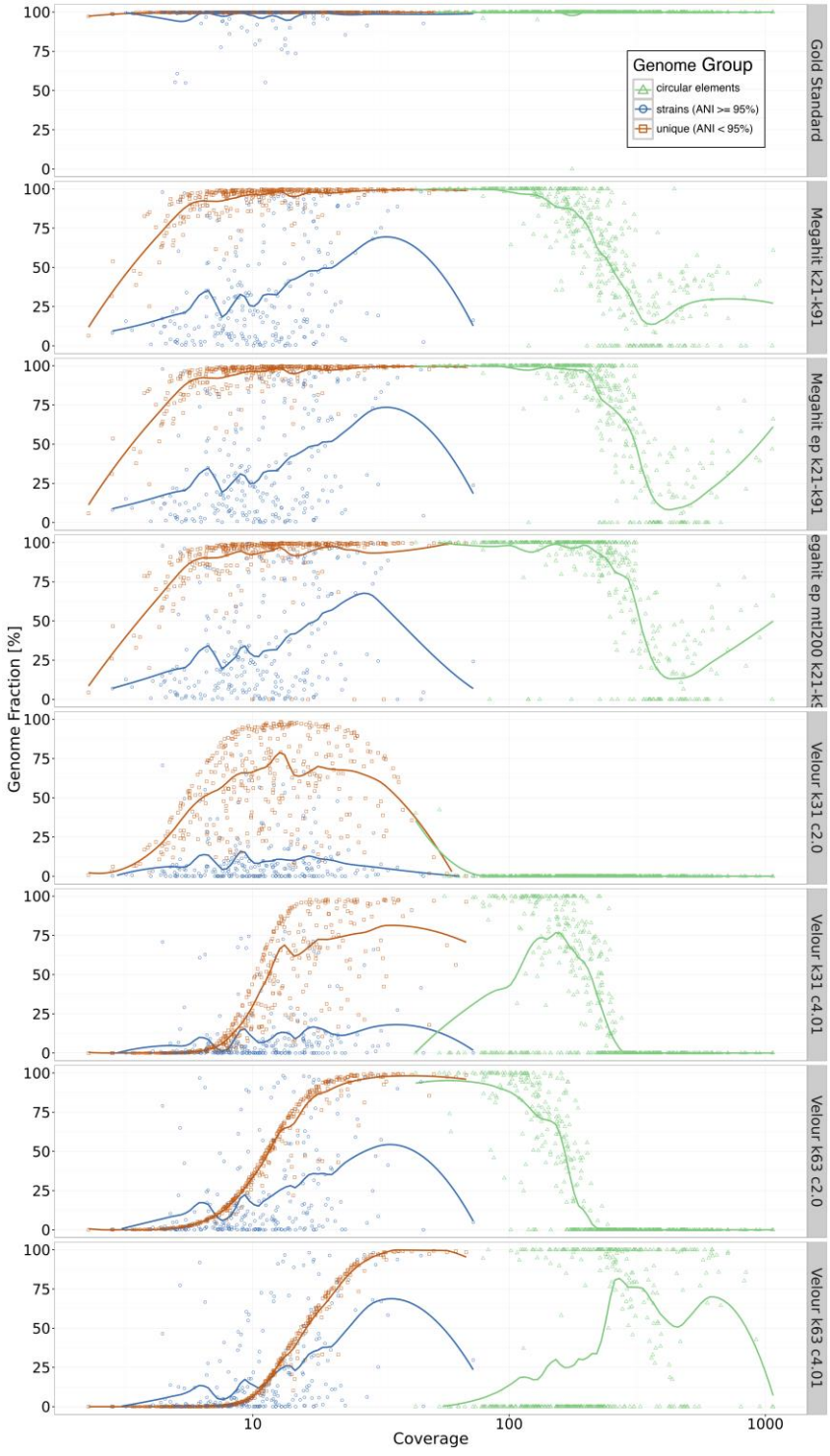
3 Supplementary Figures

3.1 Assembly



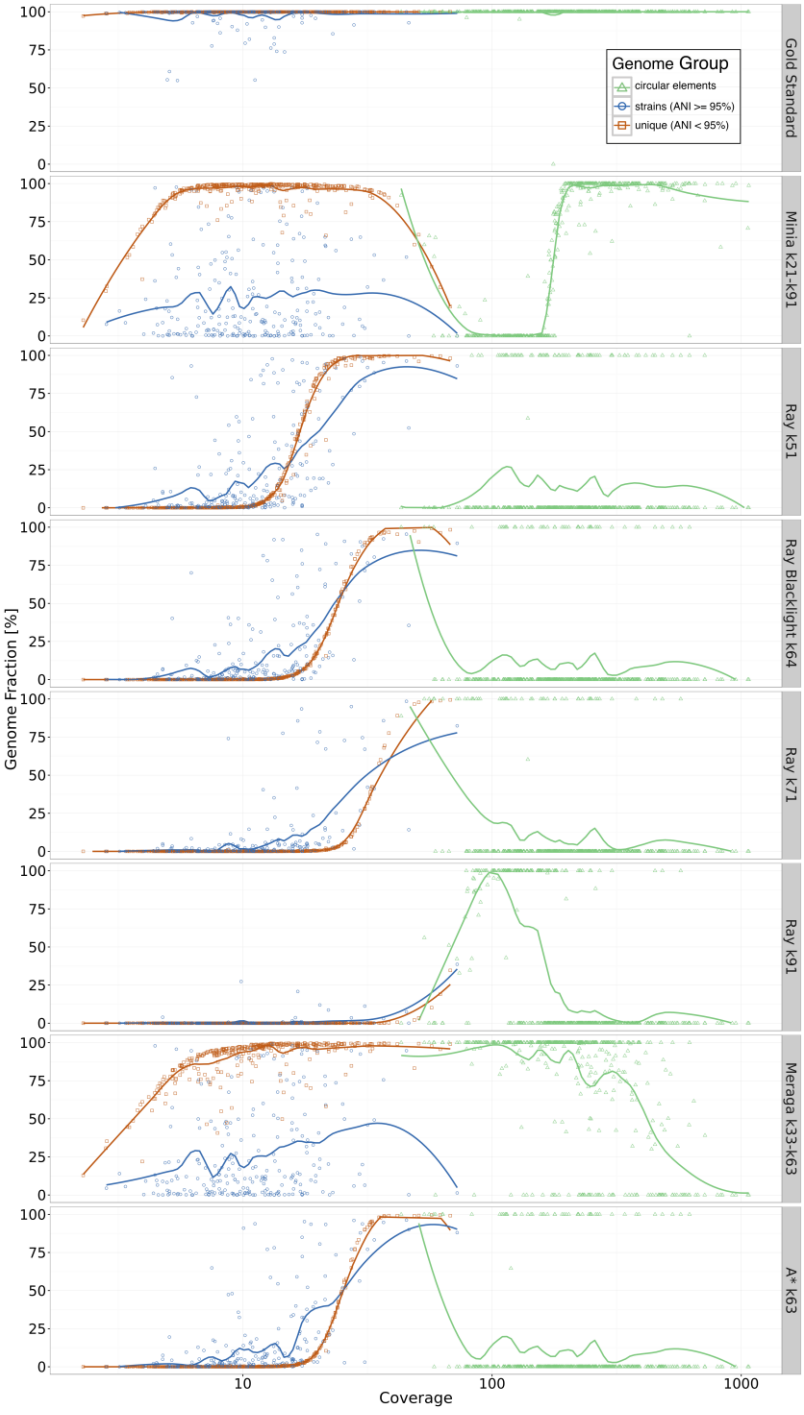
Supplementary Figure 1. Parallel coordinate plot showing performances of all assemblies for the high complexity dataset. Vertical axes represent the combined key QUAST metrics for all reference genomes. Each assembly is represented by a colored line, labels are on the left. Metrics shown are number of contigs (contigs), total length of assembly (Total_length), number of misassemblies (misassemblies), length of unaligned contigs (Unaligned_length), fraction of genomes recovered (Genome_fraction), duplication ratio, mismatches, predicted genes and NA50. For a detailed description of metrics and results for the medium and low complexity datasets see Supplementary Note 1. For raw QUAST results used to generate this plot, see "Supplementary_Figure_1_Source_Data.xlsx".

Nature Methods: doi:10.1038/nmeth.4458



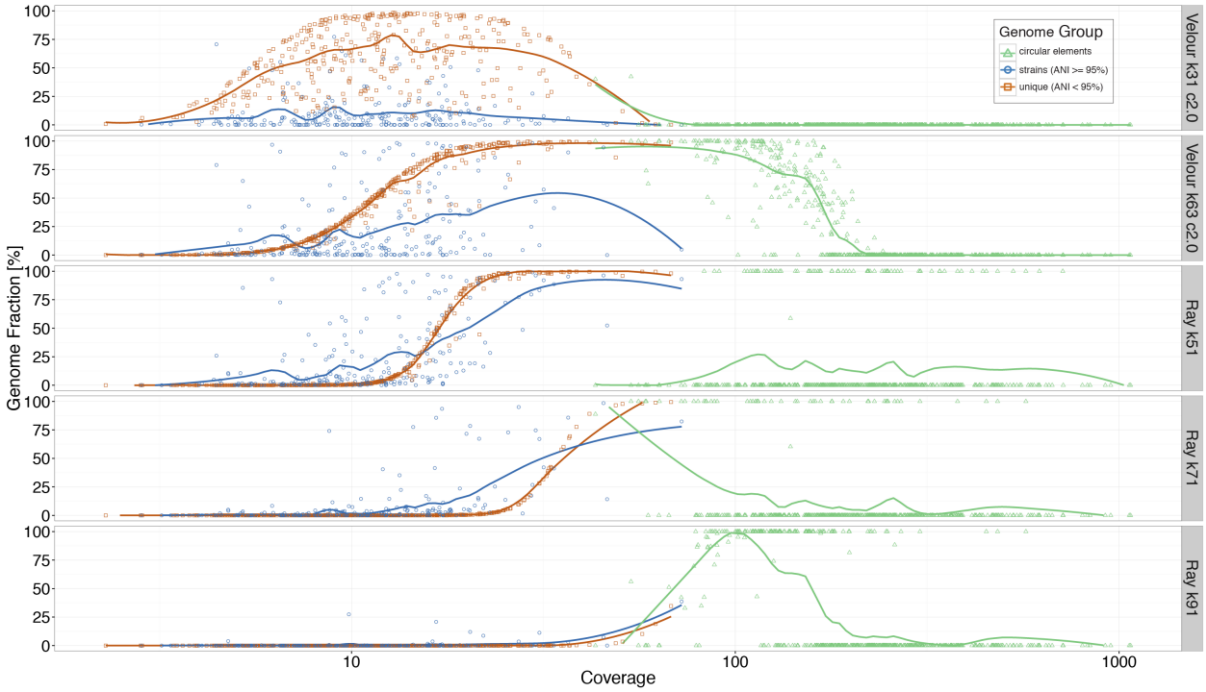
Supplementary Figure 2 (part 1). Genome recovery fraction versus genome sequencing depth (coverage) for the high complexity dataset. Data were classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI ≥ 95%, blue color) and high copy circular elements (green color). For raw QUAST results used to generate this plot, see “Figure_1_Source_Data.xlsx”.

Nature Methods: doi:10.1038/nmeth.4458



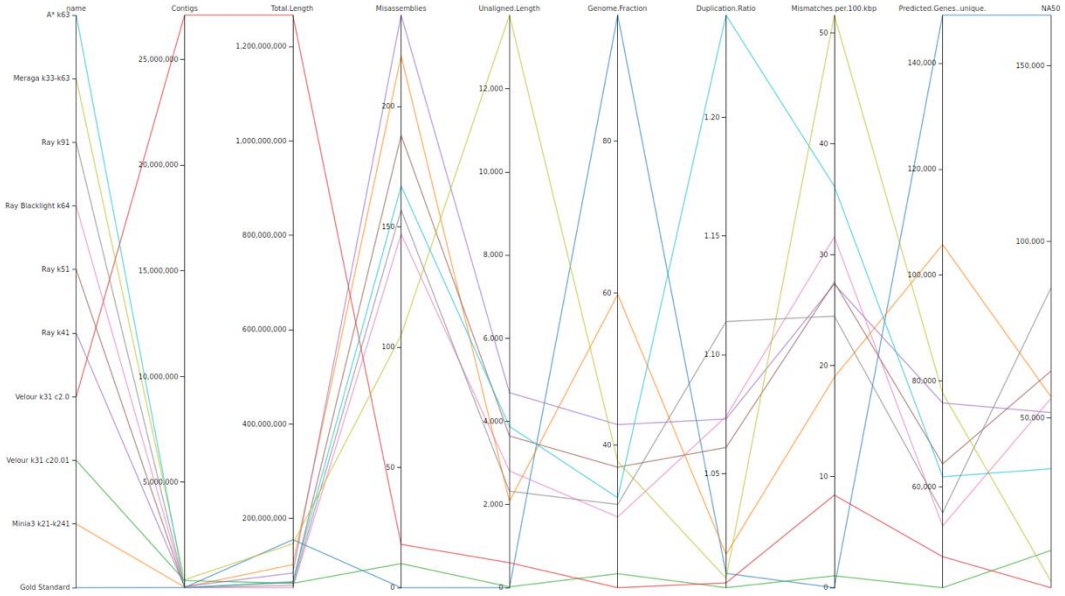
Supplementary Figure 2 (part 2). Genome recovery fraction versus genome sequencing depth (coverage) for the high complexity dataset. Data were classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI ≥ 95%, blue color) and high copy circular elements (green color). For raw QUAST results used to generate this plot, see "Figure_1_Source_Data.xlsx".

Nature Methods: doi:10.1038/nmeth.4458



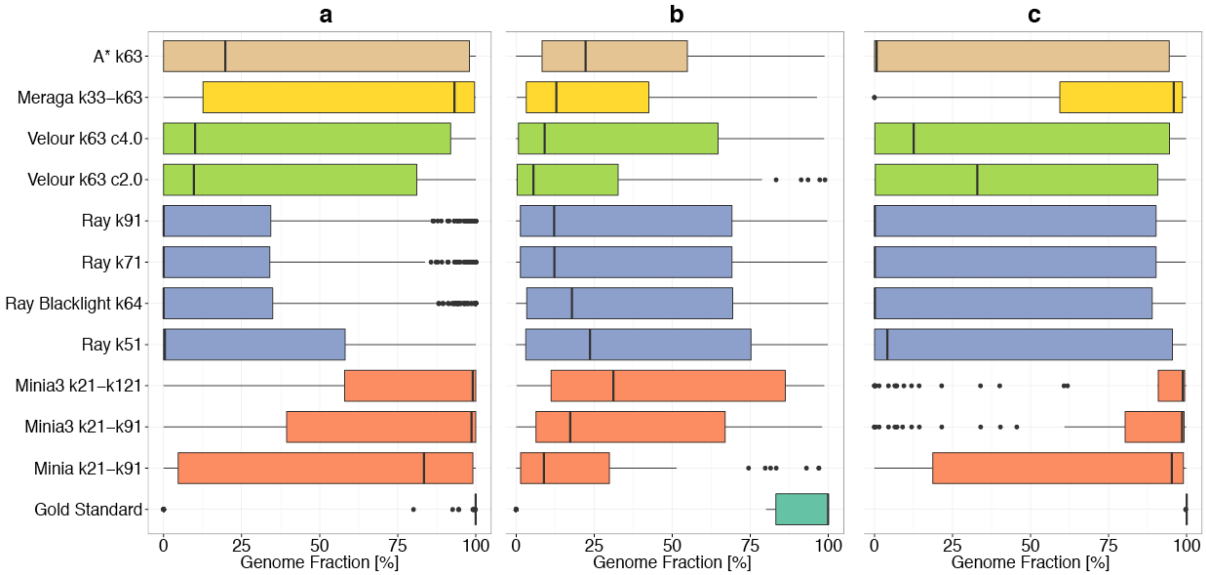
Supplementary Figure 3. Effect of different k-mer sizes on performance for the Velour and Ray assemblers. Shown is the genome fraction versus genome sequencing depth (coverage) for the high complexity dataset. Different k-mer sizes are denoted as 'kNumber' in the assembler names on the right in the individual panels. Data were classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI ≥ 95%, blue color) and high copy circular elements (green color).

Nature Methods: doi:10.1038/nmeth.4458



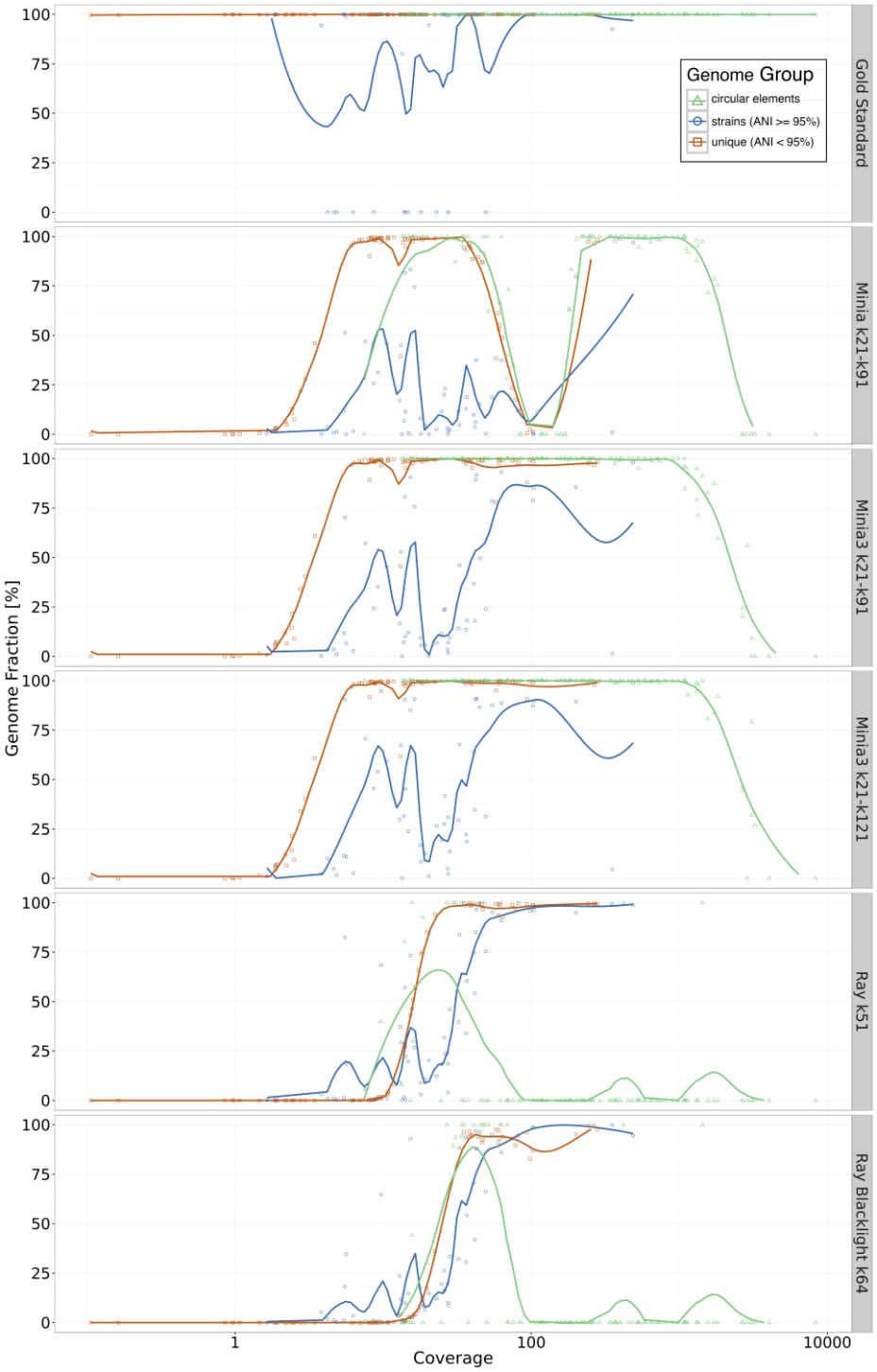
Supplementary Figure 4. Parallel coordinate plot showing performances of all assemblies for the medium complexity dataset. Vertical axes represent the combined key QUAST metrics for all reference genomes. Each assembly is represented by a colored line, labels are on the left. Metrics shown are number of contigs (contigs), total length of assembly (Total_length), number of misassemblies (misassemblies), length of unaligned contigs (Unaligned_length), fraction of genomes recovered (Genome_fraction), duplication ratio, mismatches, predicted genes and NA50. For a detailed description of metrics and results for the medium and low complexity datasets see Supplementary Note 1. For raw QUAST results used to generate this plot, see "Supplementary_Figure_4_Source_Data.xlsx".

Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 5. Boxplots representing the fraction of reference genomes assembled by each assembler for the medium complexity dataset. (**a**) All genomes; (**b**) genomes with ANI ≥ 95%; (**c**) genomes with ANI < 95%. Coloring indicates the results from the same assembler incorporated in different pipelines or with other parameter settings. For raw QUAST results used to generate this plot, see "Supplementary_Figures_5_6_Source_Data.xlsx".

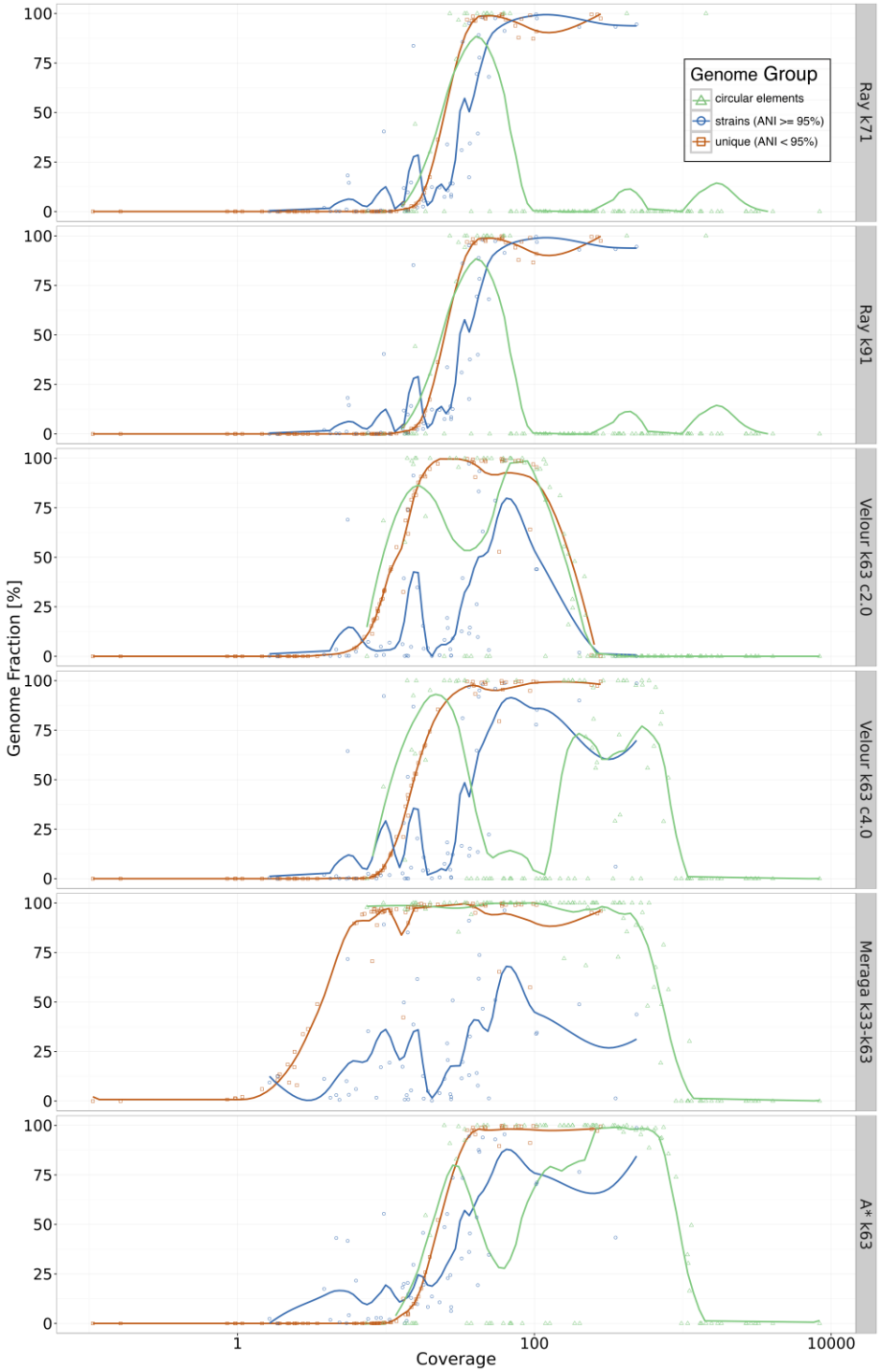
Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 6 (part 1). Genome recovery fraction versus genome sequencing depth (coverage) for the medium complexity dataset. Data were classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI ≥ 95%, blue color) and high copy circular elements (green color).

Nature Methods: doi:10.1038/nmeth.4458

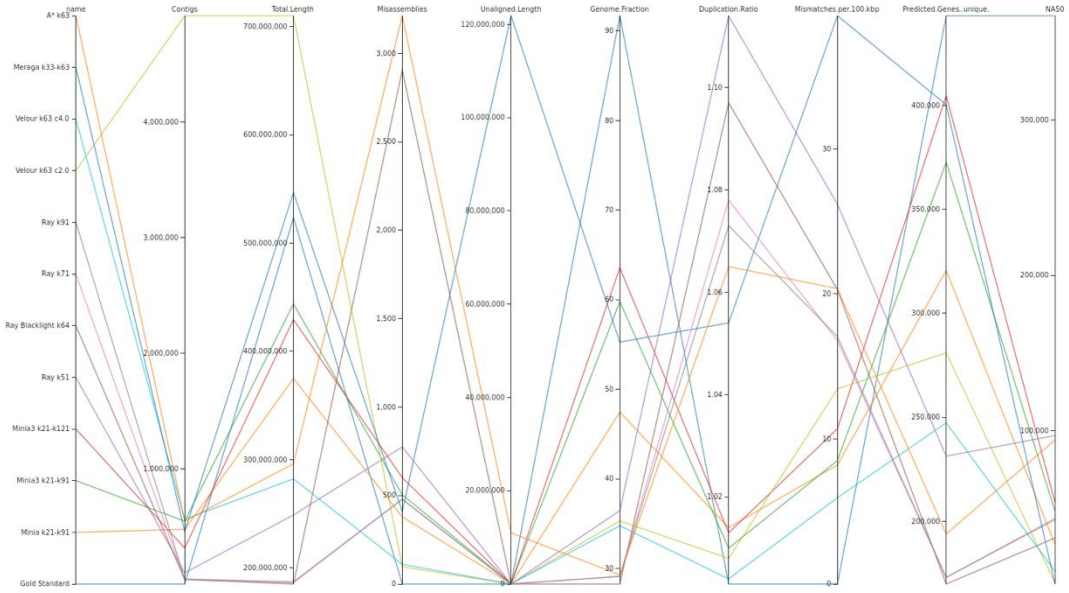
For raw QUAST results used to generate this plot, see "Supplementary_Figures_5_6_Source_Data.xlsx".



Supplementary Figure 6 (part 2). Genome recovery fraction versus genome sequencing depth (coverage) for the medium complexity dataset. Data were

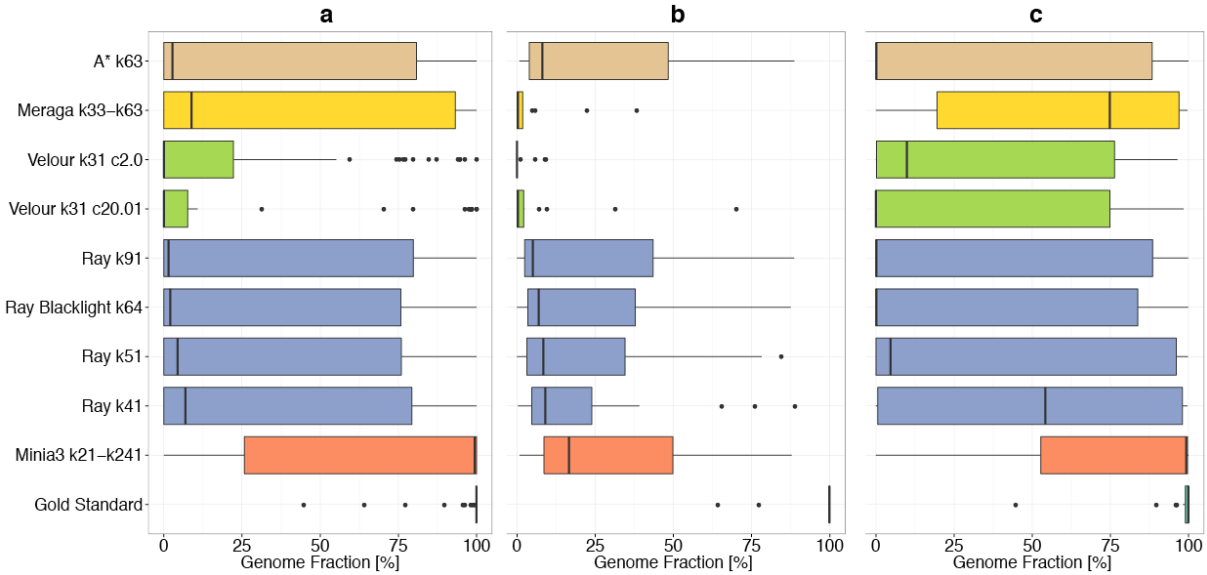
Nature Methods: doi:10.1038/nmeth.4458

classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI >= 95%, blue color) and high copy circular elements (green color). For raw QUAST results used to generate this plot, see "Supplementary_Figures_5_6_Source_Data.xlsx".



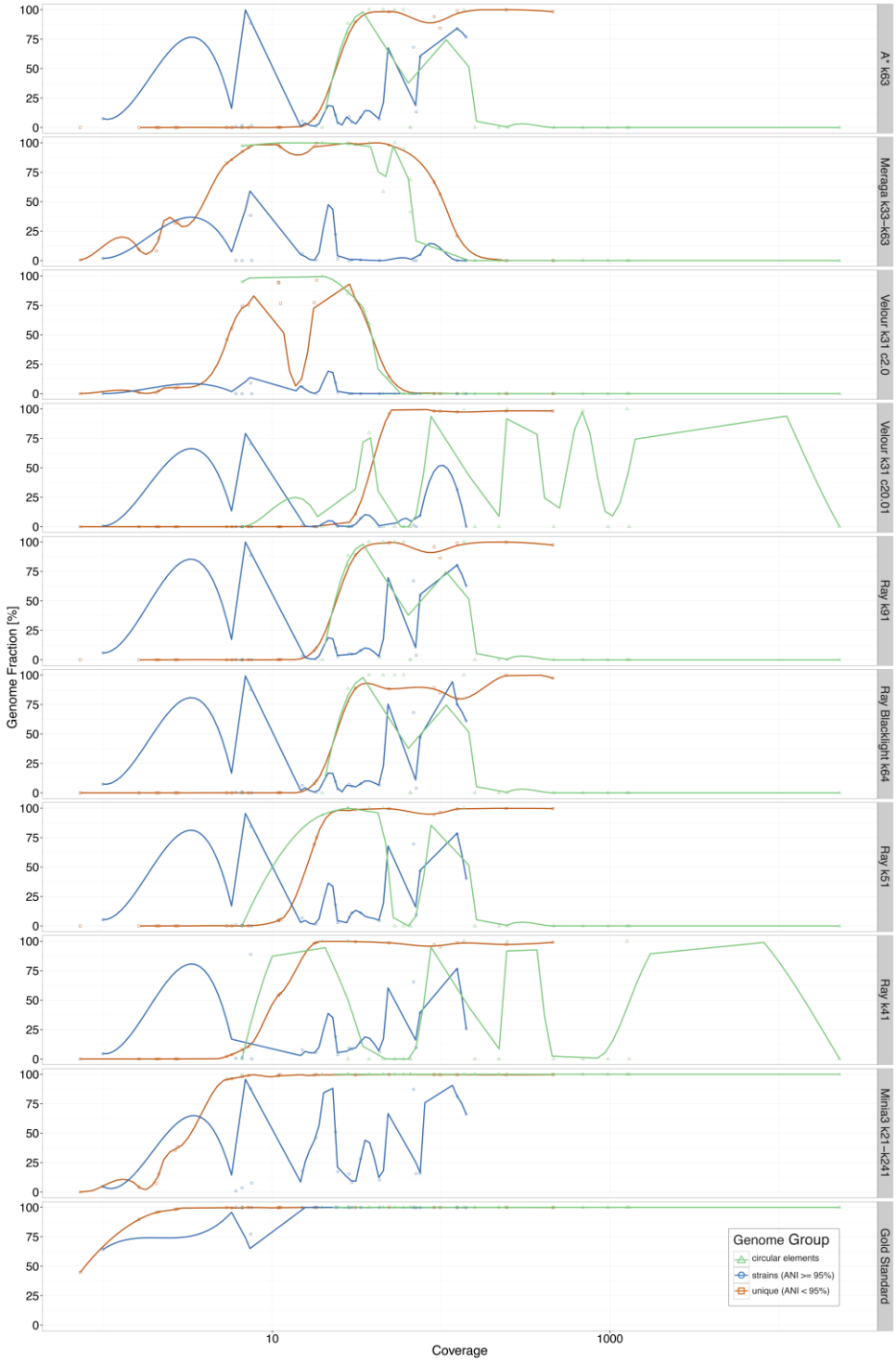
Supplementary Figure 7. Parallel coordinate plot showing performances of all assemblies for the low complexity dataset. Vertical axes represent the combined key QUAST metrics for all reference genomes. Each assembly is represented by a colored line, labels are on the left. Metrics shown are number of contigs (contigs), total length of assembly (Total_length), number of misassemblies (misassemblies), length of unaligned contigs (Unaligned_length), fraction of genomes recovered (Genome_fraction), duplication ratio, mismatches, predicted genes and NA50. For a detailed description of metrics and results for the medium and low complexity datasets see Supplementary Note 1. For QUAST results used to generate this plot, see "Supplementary_Figure_7_Source_Data.xlsx".

Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 8. Boxplots representing the fraction of reference genomes assembled by each assembler for the low complexity dataset. (**a**) All genomes; (**b**) genomes with ANI ≥95%; (**c**) genomes with ANI < 95%. Coloring indicates the results from the same assembler incorporated in different pipelines or with other parameter settings. For raw QUAST results used to generate this plot, see "Supplementary_Figures_8_9_Source_Data.xlsx".

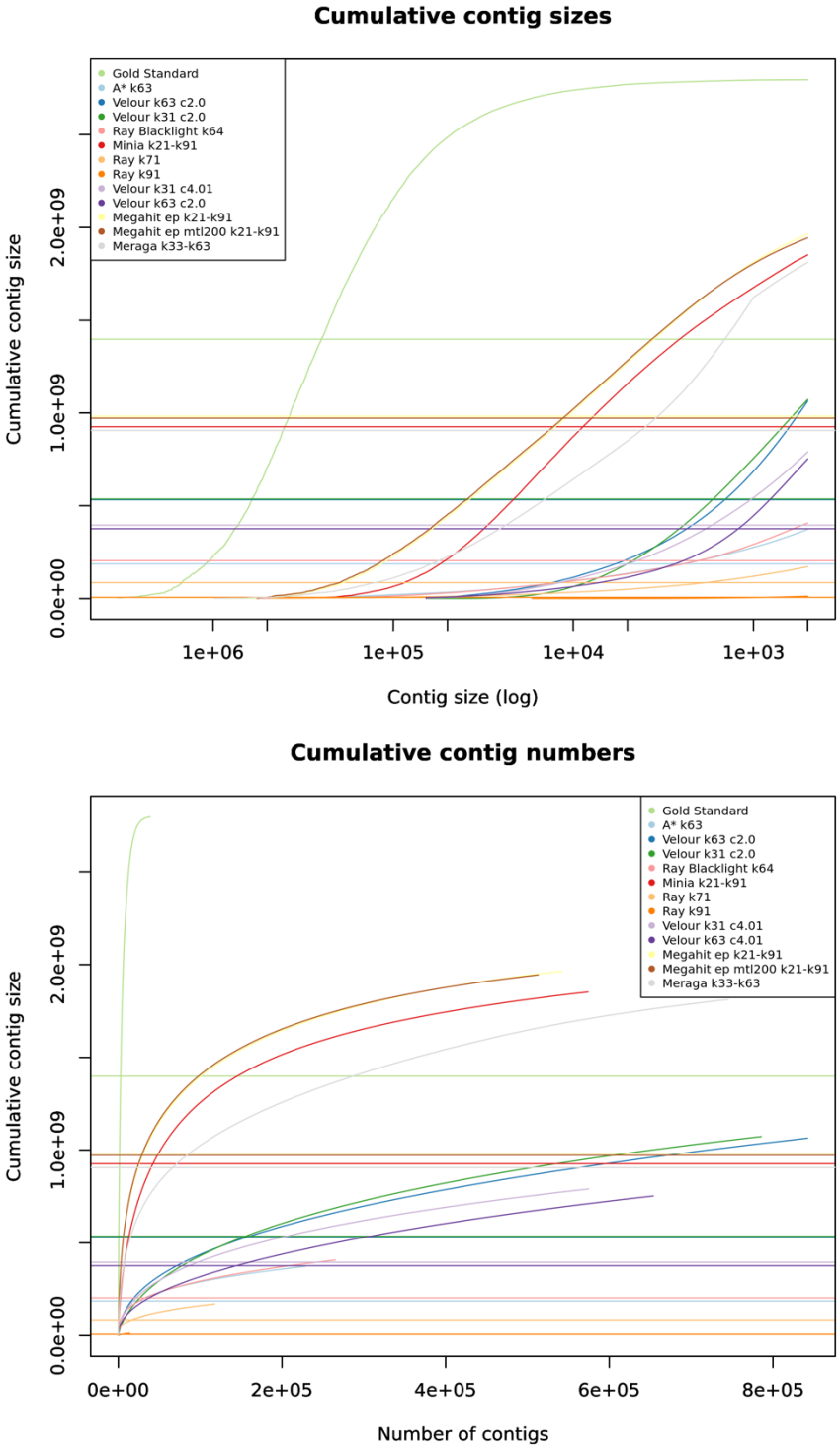
Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 9. Genome recovery fraction versus genome sequencing depth (coverage) for the low complexity dataset. Data were classified as unique genomes (ANI < 95%, brown color), genomes with related strains present (ANI ≥ 95%, blue color) and high copy circular elements (green color). For raw QUAST

Nature Methods: doi:10.1038/nmeth.4458

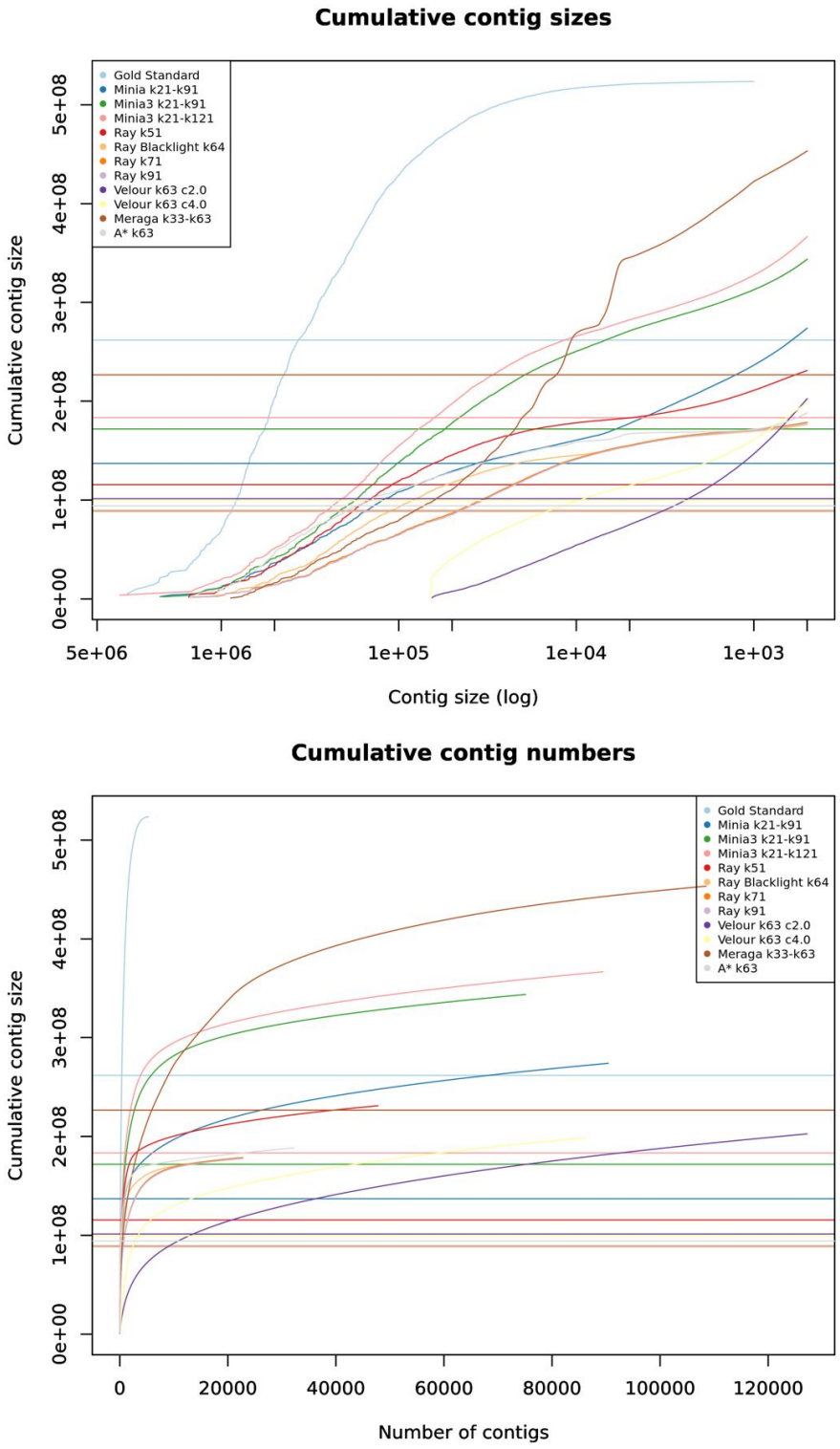
results used to generate this plot, see "Supplementary_Figures_8_9_Source_Data.xlsx".



Nature Methods: doi:10.1038/nmeth.4458

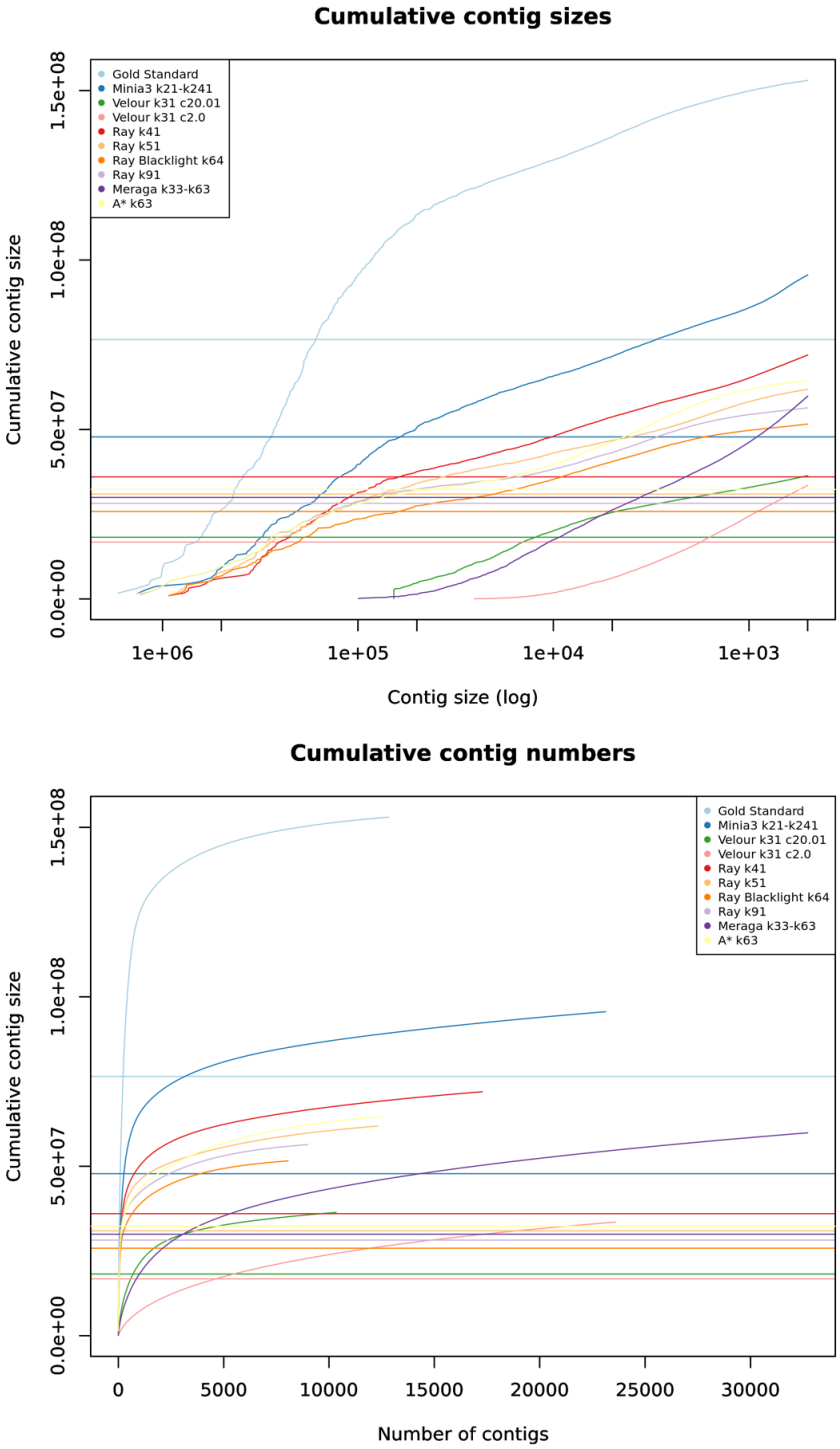
Supplementary Figure 10. Cumulative contig sizes (top) and cumulative contig numbers (bottom) of CAMI high complexity dataset assemblies. Horizontal lines depict the N50 for each assembly.

Nature Methods: doi:10.1038/nmeth.4458



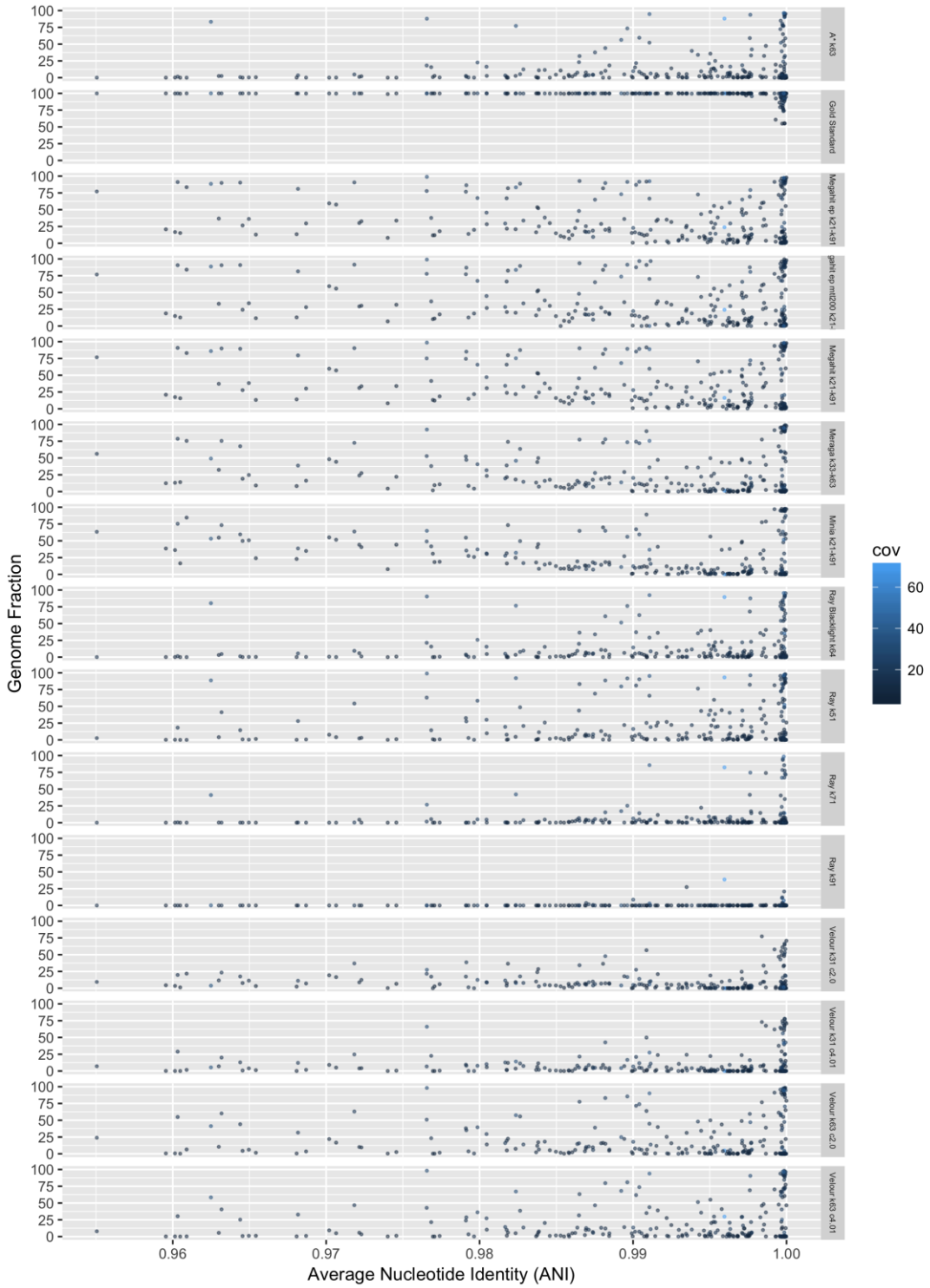
Supplementary Figure 11. Cumulative contig sizes (top) and cumulative contig numbers (bottom) of CAMI medium complexity dataset assemblies. Horizontal lines depict the N50 for each assembly.

Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 12. Cumulative contig sizes (top) and cumulative contig numbers (bottom) of CAMI low complexity dataset assemblies. Horizontal lines depict the N50 for each assembly.

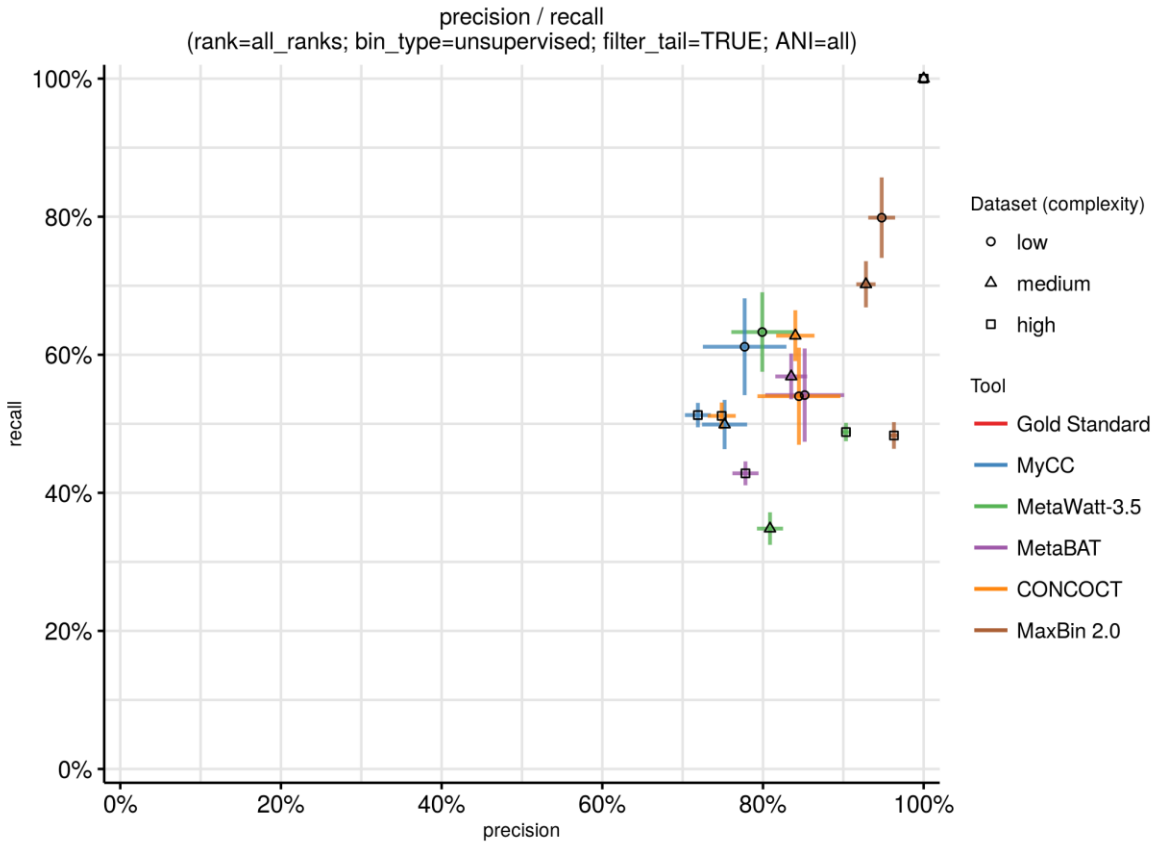
Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 13. ANI versus Genome Fraction for all genomes in the high complexity data set. Color encodes coverage of the genome. For raw QUAST results used to generate this plot, see "Figure_1_Source_Data.xlsx".

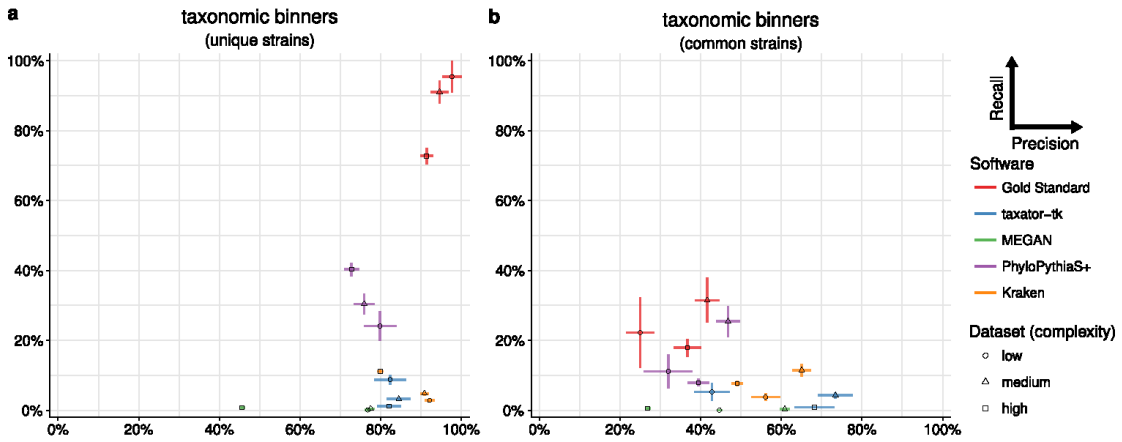
Nature Methods: doi:10.1038/nmeth.4458

3.2 Binning



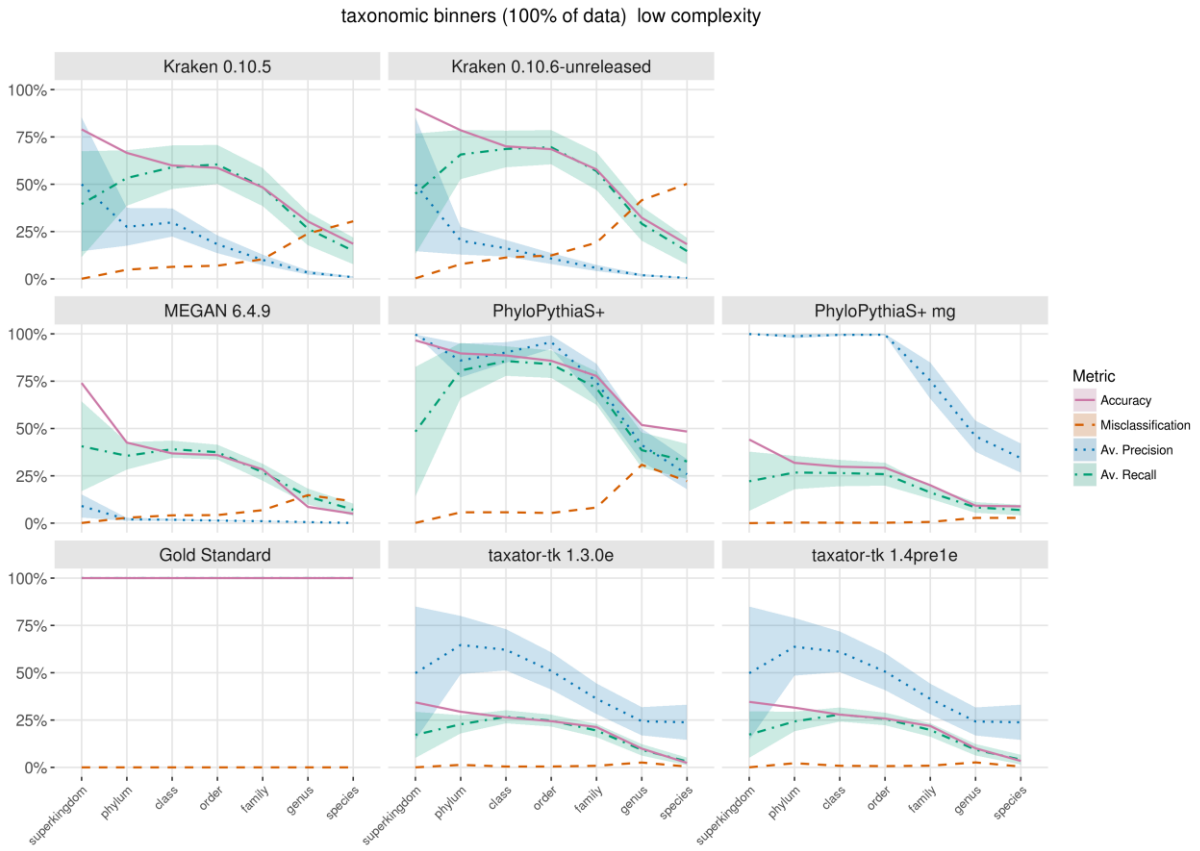
Supplementary Figure 14. Average precision (purity; x-axis) and recall (completeness; y-axis) for genome binners by genome, for all genomes. For each genome binner and complexity dataset, the submission with the largest sum of precision and recall is shown (Supplementary Table 5). Bars denote the standard error of the mean precision and recall across genome bins. In each case, small bins adding up to 1% of the dataset size overall were removed. The gold standard (100% precision, 100% recall, 0% standard error) is shown in the upper right corner.

Nature Methods: doi:10.1038/nmeth.4458



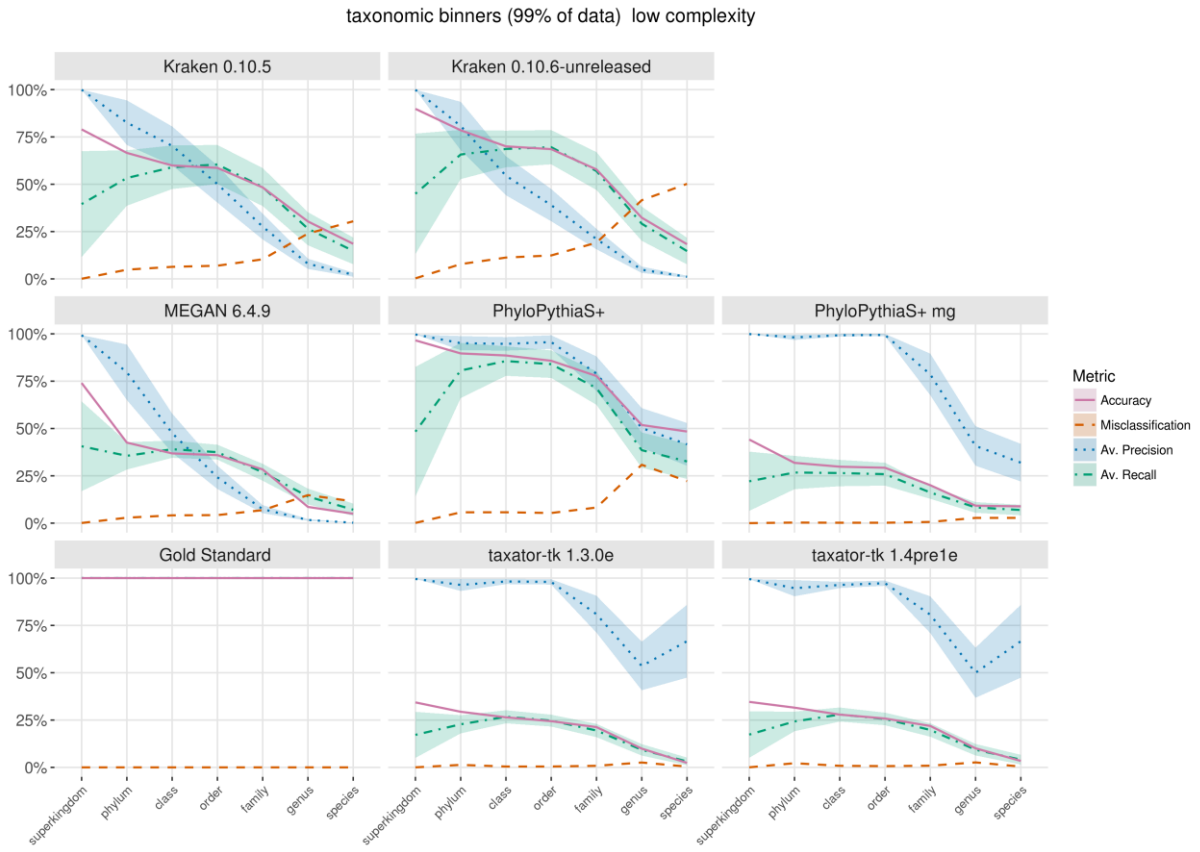
Supplementary Figure 15. Average purity (x-axis) and completeness (y-axis) and their standard errors (bars) for genomes reconstructed by taxonomic binners; for genomes of unique strains with less than 95% Average Nucleotide Identity to others (**a**) and common strains with more than or equal to 95% ANI to each other (**b**). For each program and complexity dataset, the submission with the largest sum of purity and completeness is shown (Supplementary Table 2, "Supplementary_Figure_15_a_Source_Data.xlsx" and "Supplementary_Figure_15_b_Source_Data.xlsx"). In each case, small bins adding up to 1% of the data set size were removed.

Nature Methods: doi:10.1038/nmeth.4458



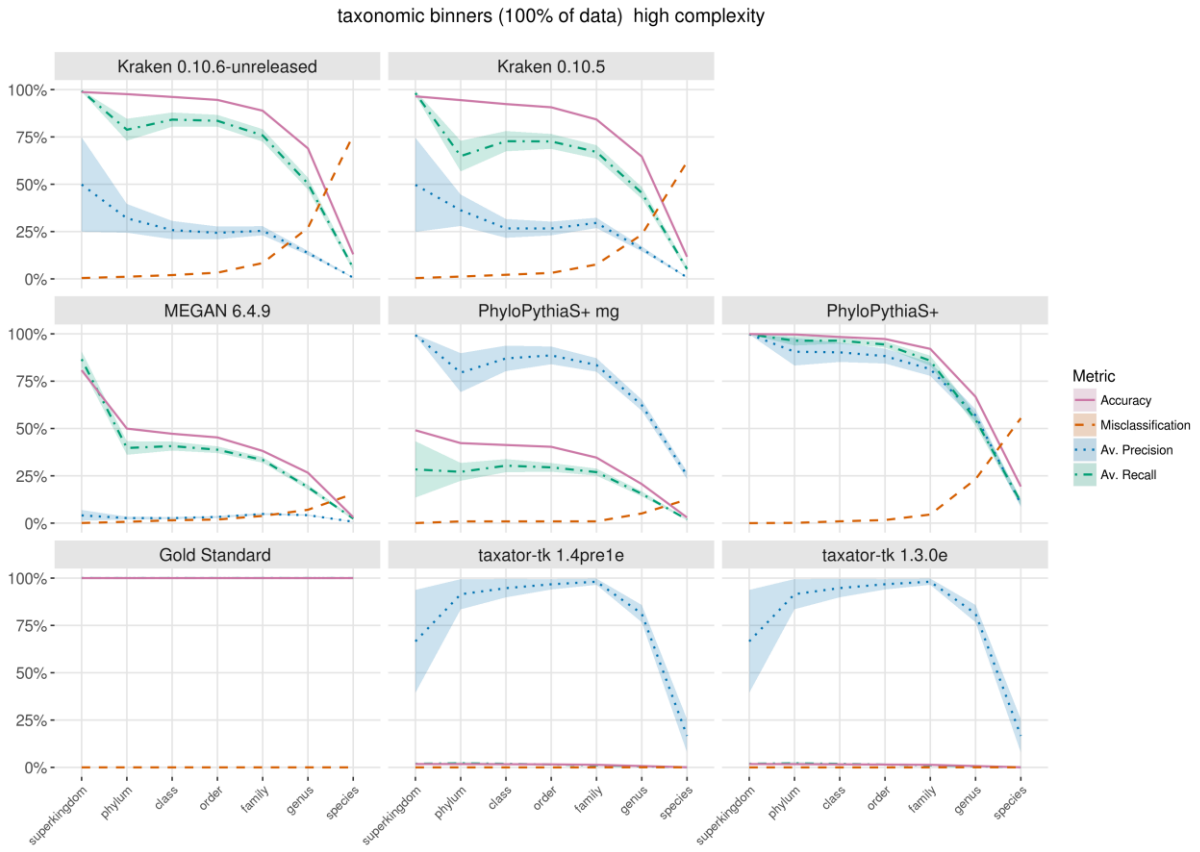
Supplementary Figure 16. Taxonomic binning performance metrics across ranks for the low complexity dataset. Shaded areas indicate the standard error of the mean for precision and recall across taxon bins at a given rank. For several programs, multiple submissions with different parameter settings were evaluated (Supplementary Table 2).

Nature Methods: doi:10.1038/nmeth.4458



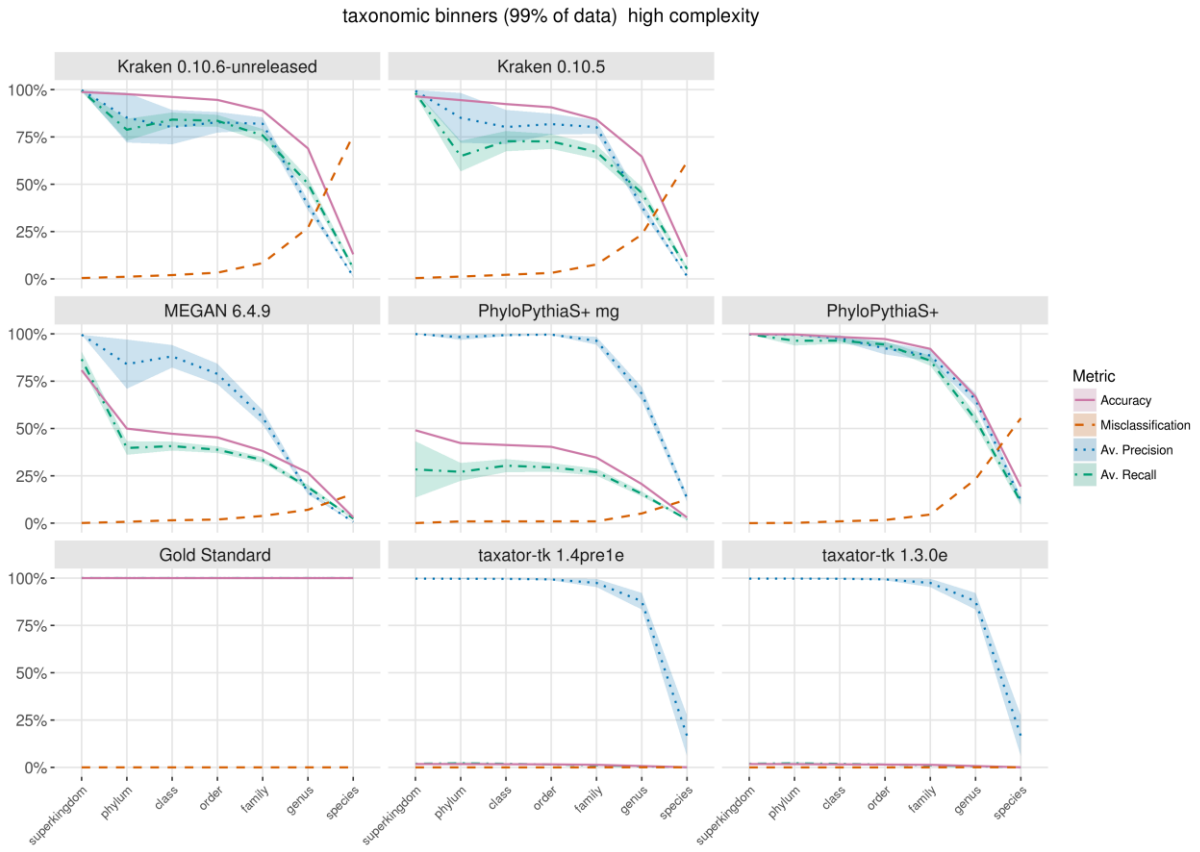
Supplementary Figure 17. Taxonomic binning performance metrics across ranks for the low complexity dataset, with smallest predicted bins summing up to 1% of entire dataset removed. Shaded areas indicate the standard error of the mean for precision and recall across taxon bins at a given rank.

Nature Methods: doi:10.1038/nmeth.4458



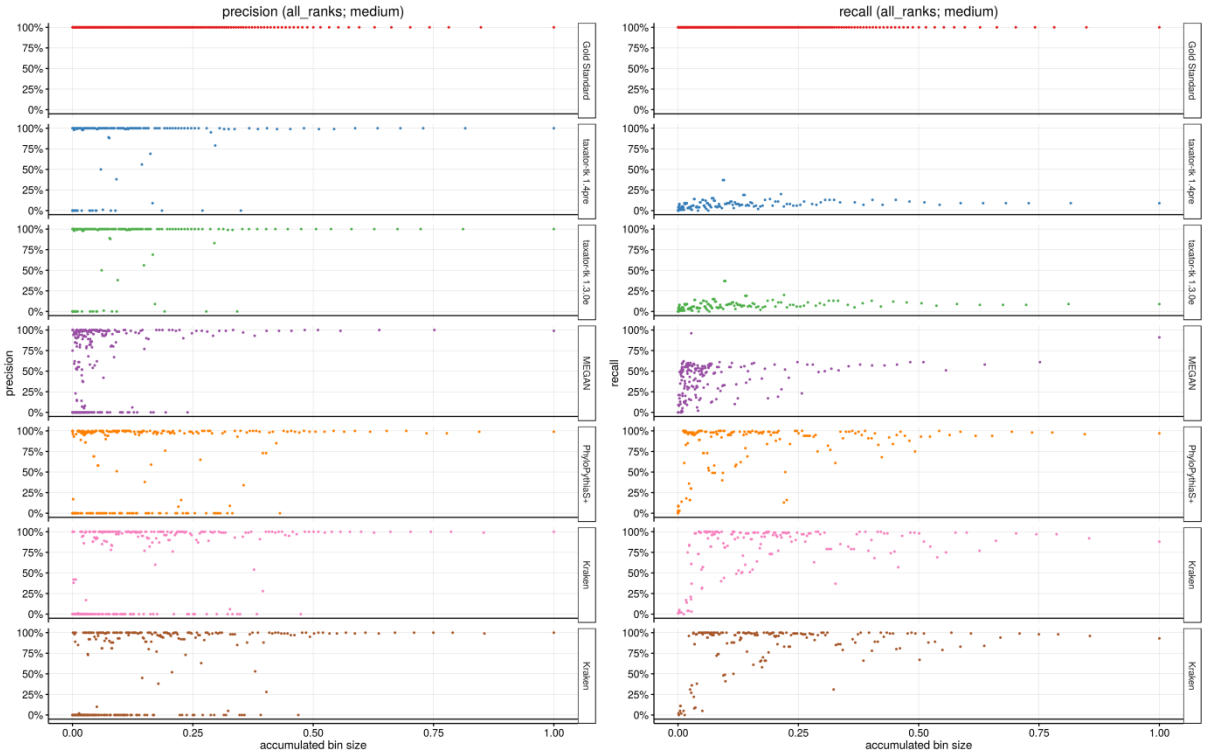
Supplementary Figure 18. Taxonomic binning performance metrics across ranks for the high complexity dataset. For several programs, multiple submissions with different parameter settings were evaluated (Supplementary Table 2). Shaded areas indicate the standard error of the mean for precision and recall across taxon bins at a given rank.

Nature Methods: doi:10.1038/nmeth.4458



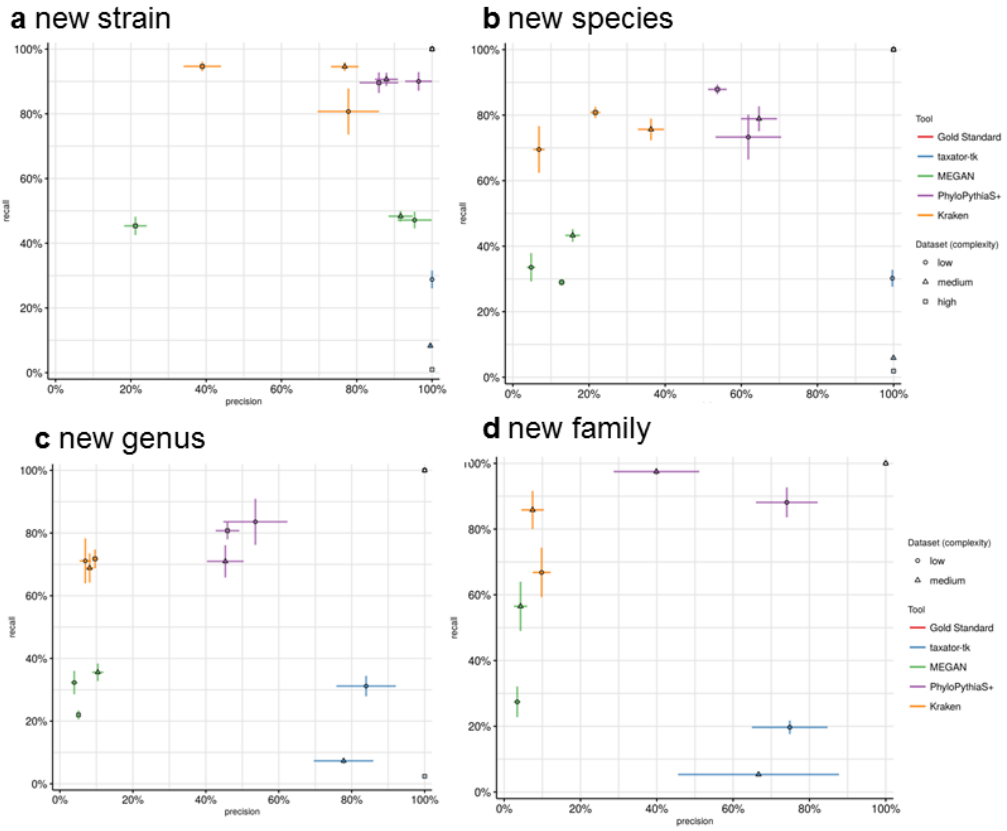
Supplementary Figure 19. Taxonomic binning performance metrics across ranks for the high complexity dataset, with smallest predicted bins summing up to 1% of entire dataset removed. Shaded areas indicate the standard error of the mean for precision and recall across taxon bins at a given rank.

Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 20. Precision and recall by bin across all taxonomic ranks, sorted by real bin sizes for all genomes on the medium complexity datasets. For programs with multiple submissions, the one with the largest sum of precision and recall is shown.

Nature Methods: doi:10.1038/nmeth.4458

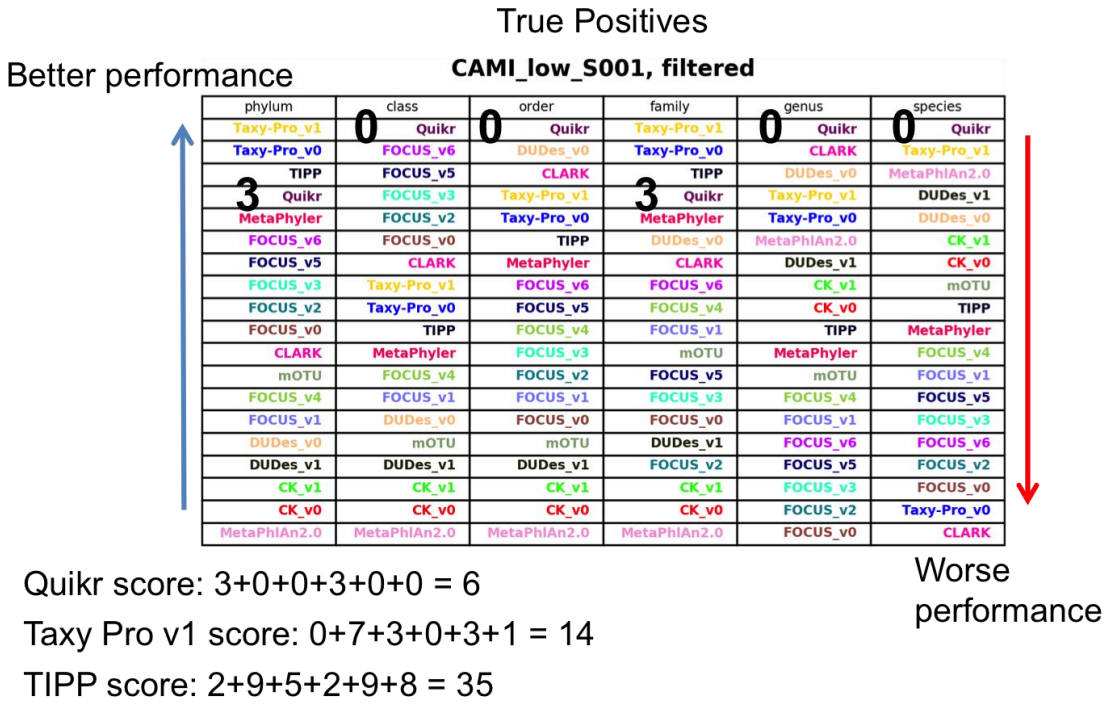


Supplementary Figure 21. Average precision (taxon bin purity) and recall (taxon bin completeness) by bin across all taxonomic ranks and their standard errors (bars), for all genomes representing (**a**) a new strain from a species represented in public genome sequence collections ("new strain" scenario), (**b**) a new species, (**c**) new genus and (**d**) new family relative to taxa represented in public genome sequences collections. The smallest predicted bins summing up to 1% of entire dataset were removed. For programs with multiple submissions, the one with the largest sum of precision and recall is shown. Note that performance is largely influenced by lower ranks, where most bins are from, and quantitative comparisons are most informative of the different categories to each other. The gold standard (precision and recall of 100%) is in the upper right corner (a square without red error bars, as standard error is 0).

Nature Methods: doi:10.1038/nmeth.4458

3.3 Profiling

3.3.1 Rankings



Supplementary Figure 22. Depiction of how the score was determined for the sample in the low complexity dataset for the true positives metric. At each taxonomic rank, a score is assigned corresponding to a profilers ranking (starting at 0 for the highest ranked profiler). Scores are then summed over every taxonomic rank to calculate the final scores for this sample.

CAMI_HIGH_S001, filtered

phylum	class	order	family	genus	species
Quikr	Quikr	CLARK	CLARK	CLARK	Quikr
CLARK	CLARK	TIPP	MetaPhyler	Quikr	CK_v0
TIPP	TIPP	MetaPhyler	TIPP	MetaPhyler	TIPP
MetaPhyler	MetaPhyler	Taxy-Pro_v0	Taxy-Pro_v0	TIPP	MetaPhyler
Taxy-Pro_v1	Taxy-Pro_v1	Quikr	Quikr	Taxy-Pro_v0	MetaPhlAn2.0
Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v1	mOTU
FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	mOTU	Taxy-Pro_v1
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	mOTU	MetaPhlAn2.0	Taxy-Pro_v0
mOTU	mOTU	mOTU	MetaPhlAn2.0	FOCUS_v5	FOCUS_v5
CK_v0	CK_v0	CK_v0	CK_v0	CK_v0	CLARK

Nature Methods: doi:10.1038/nmeth.4458

CAMI_MED_S001, filtered

phylum	class	order	family	genus	species
Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v0	MetaPhyler	CLARK	MetaPhlAn2.0
Taxy-Pro_v0	Taxy-Pro_v0	TIPP	CLARK	MetaPhyler	CK_v1
TIPP	TIPP	MetaPhyler	TIPP	TIPP	DUDes_v1
Quikr	Quikr	CLARK	Taxy-Pro_v1	DUDes_v1	CK_v0
MetaPhyler	MetaPhyler	Taxy-Pro_v1	Taxy-Pro_v0	Quikr	mOTU
FOCUS_v5	FOCUS_v5	Quikr	Quikr	mOTU	TIPP
CLARK	CLARK	DUDes_v1	DUDes_v1	MetaPhlAn2.0	MetaPhyler
mOTU	MetaPhlAn2.0	mOTU	mOTU	CK_v1	Taxy-Pro_v1
MetaPhlAn2.0	DUDes_v1	MetaPhlAn2.0	FOCUS_v5	Taxy-Pro_v0	Quikr
DUDes_v1	mOTU	FOCUS_v5	MetaPhlAn2.0	Taxy-Pro_v1	FOCUS_v5
CK_v1	CK_v1	CK_v1	CK_v1	FOCUS_v5	Taxy-Pro_v0
CK_v0	CK_v0	CK_v0	CK_v0	CK_v0	CLARK

CAMI_low_S001, filtered

phylum	class	order	family	genus	species
Taxy-Pro_v1	Quikr	Quikr	Taxy-Pro_v1	Quikr	Quikr
Taxy-Pro_v0	FOCUS_v6	DUDes_v0	Taxy-Pro_v0	CLARK	Taxy-Pro_v1
TIPP	FOCUS_v5	CLARK	TIPP	DUDes_v0	MetaPhlAn2.0
Quikr	FOCUS_v3	Taxy-Pro_v1	Quikr	Taxy-Pro_v1	DUDes_v1
MetaPhyler	FOCUS_v2	Taxy-Pro_v0	MetaPhyler	Taxy-Pro_v0	DUDes_v0
FOCUS_v6	FOCUS_v0	TIPP	DUDes_v0	MetaPhlAn2.0	CK_v1
FOCUS_v5	CLARK	MetaPhyler	CLARK	DUDes_v1	CK_v0
FOCUS_v3	Taxy-Pro_v1	FOCUS_v6	FOCUS_v6	CK_v1	mOTU
FOCUS_v2	Taxy-Pro_v0	FOCUS_v5	FOCUS_v4	CK_v0	TIPP
FOCUS_v0	TIPP	FOCUS_v4	FOCUS_v1	TIPP	MetaPhyler
CLARK	MetaPhyler	FOCUS_v3	mOTU	MetaPhyler	FOCUS_v4
mOTU	FOCUS_v4	FOCUS_v2	FOCUS_v5	mOTU	FOCUS_v1
FOCUS_v4	FOCUS_v1	FOCUS_v1	FOCUS_v3	FOCUS_v4	FOCUS_v5
FOCUS_v1	DUDes_v0	FOCUS_v0	FOCUS_v0	FOCUS_v1	FOCUS_v3
DUDes_v0	mOTU	mOTU	DUDes_v1	FOCUS_v6	FOCUS_v6
DUDes_v1	DUDes_v1	DUDes_v1	FOCUS_v2	FOCUS_v5	FOCUS_v2
CK_v1	CK_v1	CK_v1	CK_v1	FOCUS_v3	FOCUS_v0
CK_v0	CK_v0	CK_v0	CK_v0	FOCUS_v2	Taxy-Pro_v0
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	FOCUS_v0	CLARK

Supplementary Figure 23. Rankings of the profilers using the true positives metric at each taxonomic rank for three samples from the three datasets (high, medium, and low complexity). A profiler name higher on the list indicates better performance.

Nature Methods: doi:10.1038/nmeth.4458

CAMI_HIGH_S001, filtered

phylum	class	order	family	genus	species
CK_v0	CK_v0	CK_v0	CK_v0	MetaPhlAn2.0	CLARK
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	CK_v0	Taxy-Pro_v0
mOTU	mOTU	mOTU	mOTU	mOTU	MetaPhlAn2.0
TIPP	FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	CK_v0
MetaPhyler	Taxy-Pro_v1	TIPP	Taxy-Pro_v1	TIPP	FOCUS_v5
FOCUS_v5	TIPP	Taxy-Pro_v1	TIPP	Taxy-Pro_v1	mOTU
Taxy-Pro_v1	Taxy-Pro_v0	Taxy-Pro_v0	Quikr	Taxy-Pro_v0	Taxy-Pro_v1
Taxy-Pro_v0	MetaPhyler	Quikr	Taxy-Pro_v0	MetaPhyler	TIPP
Quikr	Quikr	MetaPhyler	MetaPhyler	Quikr	Quikr
CLARK	CLARK	CLARK	CLARK	CLARK	MetaPhyler

CAMI_MED_S001, filtered

phylum	class	order	family	genus	species
CK_v0	CK_v0	CK_v0	CK_v0	MetaPhlAn2.0	CLARK
CK_v1	CK_v1	CK_v1	MetaPhlAn2.0	CK_v0	Taxy-Pro_v0
DUDes_v1	DUDes_v1	DUDes_v1	CK_v1	CK_v1	MetaPhlAn2.0
mOTU	mOTU	MetaPhlAn2.0	DUDes_v1	DUDes_v1	CK_v0
MetaPhlAn2.0	MetaPhlAn2.0	mOTU	mOTU	mOTU	DUDes_v1
FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	CK_v1
MetaPhyler	TIPP	TIPP	Taxy-Pro_v1	TIPP	mOTU
TIPP	Taxy-Pro_v1	Taxy-Pro_v1	TIPP	Taxy-Pro_v1	FOCUS_v5
Taxy-Pro_v1	Taxy-Pro_v0	Taxy-Pro_v0	Quikr	Quikr	Taxy-Pro_v1
Taxy-Pro_v0	MetaPhyler	MetaPhyler	Taxy-Pro_v0	Taxy-Pro_v0	TIPP
Quikr	Quikr	Quikr	MetaPhyler	MetaPhyler	Quikr
CLARK	CLARK	CLARK	CLARK	CLARK	MetaPhyler

CAMI_low_S001, filtered

phylum	class	order	family	genus	species
DUDes_v0	DUDes_v1	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	CLARK
DUDes_v1	MetaPhlAn2.0	CK_v0	CK_v0	DUDes_v1	MetaPhlAn2.0
MetaPhlAn2.0	CK_v0	DUDes_v1	DUDes_v1	CK_v0	Taxy-Pro_v0
mOTU	CK_v1	CK_v1	CK_v1	CK_v1	DUDes_v1
CK_v0	DUDes_v0	mOTU	mOTU	FOCUS_v1	CK_v0
CK_v1	mOTU	DUDes_v0	FOCUS_v2	FOCUS_v4	CK_v1
FOCUS_v2	FOCUS_v2	FOCUS_v2	DUDes_v0	FOCUS_v2	FOCUS_v1
FOCUS_v1	FOCUS_v6	FOCUS_v6	FOCUS_v1	FOCUS_v6	FOCUS_v4
FOCUS_v4	FOCUS_v0	FOCUS_v1	FOCUS_v4	mOTU	FOCUS_v2
FOCUS_v6	FOCUS_v1	FOCUS_v4	FOCUS_v6	FOCUS_v0	FOCUS_v0
FOCUS_v0	FOCUS_v3	FOCUS_v0	FOCUS_v0	FOCUS_v3	FOCUS_v6
FOCUS_v3	FOCUS_v4	FOCUS_v3	FOCUS_v3	FOCUS_v5	mOTU
FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	DUDes_v0	FOCUS_v3
MetaPhyler	Taxy-Pro_v1	Quikr	Taxy-Pro_v1	TIPP	FOCUS_v5
TIPP	Quikr	Taxy-Pro_v1	Quikr	Taxy-Pro_v1	DUDes_v0
Taxy-Pro_v1	MetaPhyler	TIPP	TIPP	Quikr	TIPP
Taxy-Pro_v0	TIPP	Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v1
Quikr	Taxy-Pro_v0	MetaPhyler	MetaPhyler	MetaPhyler	Quikr
CLARK	CLARK	CLARK	CLARK	CLARK	MetaPhyler

Nature Methods: doi:10.1038/nmeth.4458

Supplementary Figure 24. Rankings of the profilers using the false positives metric at each taxonomic rank for three samples from the three datasets (high, medium, and low complexity). A name higher on the list indicates better performance.

CAMI_HIGH_S001, filtered

phylum	class	order	family	genus	species
Quikr	Quikr	CLARK	CLARK	CLARK	Quikr
CLARK	CLARK	TIPP	MetaPhyler	Quikr	CK_v0
TIPP	TIPP	MetaPhyler	TIPP	MetaPhyler	TIPP
MetaPhyler	MetaPhyler	Taxy-Pro_v0	Taxy-Pro_v0	TIPP	MetaPhyler
Taxy-Pro_v1	Taxy-Pro_v1	Quikr	Quikr	Taxy-Pro_v0	MetaPhlAn2.0
Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v1	mOTU
FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	mOTU	Taxy-Pro_v1
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	mOTU	MetaPhlAn2.0	Taxy-Pro_v0
mOTU	mOTU	mOTU	MetaPhlAn2.0	FOCUS_v5	FOCUS_v5
CK_v0	CK_v0	CK_v0	CK_v0	CK_v0	CLARK

CAMI_MED_S001, filtered

phylum	class	order	family	genus	species
Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v0	MetaPhyler	CLARK	MetaPhlAn2.0
Taxy-Pro_v0	Taxy-Pro_v0	TIPP	CLARK	MetaPhyler	CK_v1
TIPP	TIPP	MetaPhyler	TIPP	TIPP	DUDes_v1
Quikr	Quikr	CLARK	Taxy-Pro_v1	DUDes_v1	CK_v0
MetaPhyler	MetaPhyler	Taxy-Pro_v1	Taxy-Pro_v0	Quikr	mOTU
FOCUS_v5	FOCUS_v5	Quikr	Quikr	mOTU	TIPP
CLARK	CLARK	DUDes_v1	DUDes_v1	MetaPhlAn2.0	MetaPhyler
mOTU	MetaPhlAn2.0	mOTU	mOTU	CK_v1	Taxy-Pro_v1
MetaPhlAn2.0	DUDes_v1	MetaPhlAn2.0	FOCUS_v5	Taxy-Pro_v0	Quikr
DUDes_v1	mOTU	FOCUS_v5	MetaPhlAn2.0	Taxy-Pro_v1	FOCUS_v5
CK_v1	CK_v1	CK_v1	CK_v1	FOCUS_v5	Taxy-Pro_v0
CK_v0	CK_v0	CK_v0	CK_v0	CK_v0	CLARK

Nature Methods: doi:10.1038/nmeth.4458

CAMI_low_S001, filtered

phylum	class	order	family	genus	species
Taxy-Pro_v1	Quikr	Quikr	Taxy-Pro_v1	Quikr	Quikr
Taxy-Pro_v0	FOCUS_v6	DUDes_v0	Taxy-Pro_v0	CLARK	Taxy-Pro_v1
TIPP	FOCUS_v5	CLARK	TIPP	DUDes_v0	MetaPhlAn2.0
Quikr	FOCUS_v3	Taxy-Pro_v1	Quikr	Taxy-Pro_v1	DUDes_v1
MetaPhyler	FOCUS_v2	Taxy-Pro_v0	MetaPhyler	Taxy-Pro_v0	DUDes_v0
FOCUS_v6	FOCUS_v0	TIPP	DUDes_v0	MetaPhlAn2.0	CK_v1
FOCUS_v5	CLARK	MetaPhyler	CLARK	DUDes_v1	CK_v0
FOCUS_v3	Taxy-Pro_v1	FOCUS_v6	FOCUS_v6	CK_v1	mOTU
FOCUS_v2	Taxy-Pro_v0	FOCUS_v5	FOCUS_v4	CK_v0	TIPP
FOCUS_v0	TIPP	FOCUS_v4	FOCUS_v1	TIPP	MetaPhyler
CLARK	MetaPhyler	FOCUS_v3	mOTU	MetaPhyler	FOCUS_v4
mOTU	FOCUS_v4	FOCUS_v2	FOCUS_v5	mOTU	FOCUS_v1
FOCUS_v4	FOCUS_v1	FOCUS_v1	FOCUS_v3	FOCUS_v4	FOCUS_v5
FOCUS_v1	DUDes_v0	FOCUS_v0	FOCUS_v0	FOCUS_v1	FOCUS_v3
DUDes_v0	mOTU	mOTU	DUDes_v1	FOCUS_v6	FOCUS_v6
DUDes_v1	DUDes_v1	DUDes_v1	FOCUS_v2	FOCUS_v5	FOCUS_v2
CK_v1	CK_v1	CK_v1	CK_v1	FOCUS_v3	FOCUS_v0
CK_v0	CK_v0	CK_v0	CK_v0	FOCUS_v2	Taxy-Pro_v0
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	FOCUS_v0	CLARK

Supplementary Figure 25. Rankings of the profilers using the recall metric at each taxonomic rank for three samples from the three datasets (high, medium, and low complexity). A name higher on the list indicates better performance.

CAMI_HIGH_S001, filtered

phylum	class	order	family	genus	species
MetaPhlAn2.0	CK_v0	CK_v0	CK_v0	MetaPhlAn2.0	MetaPhlAn2.0
CK_v0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	CK_v0	CK_v0
mOTU	mOTU	mOTU	mOTU	mOTU	Quikr
TIPP	FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	mOTU
MetaPhyler	TIPP	TIPP	TIPP	TIPP	TIPP
Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v1	Quikr	Taxy-Pro_v1
FOCUS_v5	Taxy-Pro_v0	Taxy-Pro_v0	Quikr	Taxy-Pro_v1	MetaPhyler
Taxy-Pro_v0	MetaPhyler	Quikr	Taxy-Pro_v0	MetaPhyler	FOCUS_v5
Quikr	Quikr	MetaPhyler	MetaPhyler	Taxy-Pro_v0	
CLARK	CLARK	CLARK	CLARK	CLARK	

Nature Methods: doi:10.1038/nmeth.4458

CAMI_MED_S001, filtered

phylum	class	order	family	genus	species
mOTU	mOTU	CK_v0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0
DUDes_v1	DUDes_v1	CK_v1	CK_v0	CK_v0	CK_v0
CK_v1	CK_v1	DUDes_v1	DUDes_v1	CK_v1	CK_v1
CK_v0	CK_v0	mOTU	CK_v1	DUDes_v1	DUDes_v1
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	mOTU	mOTU	mOTU
FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5	FOCUS_v5
MetaPhyler	Taxy-Pro_v1	TIPP	Taxy-Pro_v1	TIPP	TIPP
TIPP	TIPP	Taxy-Pro_v1	TIPP	Quikr	Taxy-Pro_v1
Taxy-Pro_v1	Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v0	MetaPhyler	MetaPhyler
Taxy-Pro_v0	MetaPhyler	MetaPhyler	Quikr	Taxy-Pro_v1	Quikr
Quikr	Quikr	Quikr	MetaPhyler	Taxy-Pro_v0	
CLARK	CLARK	CLARK	CLARK	CLARK	

CAMI_low_S001, filtered

phylum	class	order	family	genus	species
mOTU	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0
MetaPhlAn2.0	DUDes_v1	DUDes_v1	CK_v0	DUDes_v1	DUDes_v1
DUDes_v1	DUDes_v0	CK_v0	DUDes_v1	CK_v0	CK_v0
DUDes_v0	mOTU	CK_v1	CK_v1	CK_v1	CK_v1
CK_v1	CK_v1	DUDes_v0	DUDes_v0	DUDes_v0	FOCUS_v4
CK_v0	CK_v0	mOTU	mOTU	FOCUS_v4	FOCUS_v1
FOCUS_v2	FOCUS_v2	FOCUS_v2	FOCUS_v4	FOCUS_v1	mOTU
FOCUS_v6	FOCUS_v6	FOCUS_v6	FOCUS_v1	mOTU	DUDes_v0
FOCUS_v4	FOCUS_v5	FOCUS_v4	FOCUS_v6	FOCUS_v2	FOCUS_v5
FOCUS_v1	FOCUS_v3	FOCUS_v1	FOCUS_v2	FOCUS_v6	FOCUS_v3
FOCUS_v0	FOCUS_v0	FOCUS_v5	FOCUS_v5	FOCUS_v0	Quikr
FOCUS_v5	FOCUS_v4	FOCUS_v3	FOCUS_v3	FOCUS_v5	Taxy-Pro_v1
FOCUS_v3	FOCUS_v1	FOCUS_v0	FOCUS_v0	FOCUS_v3	TIPP
MetaPhyler	Quikr	Quikr	Taxy-Pro_v1	Quikr	FOCUS_v2
TIPP	Taxy-Pro_v1	Taxy-Pro_v1	Quikr	Taxy-Pro_v1	FOCUS_v0
Taxy-Pro_v1	Taxy-Pro_v0	TIPP	TIPP	TIPP	FOCUS_v6
Taxy-Pro_v0	TIPP	Taxy-Pro_v0	Taxy-Pro_v0	Taxy-Pro_v0	MetaPhyler
Quikr	MetaPhyler	MetaPhyler	MetaPhyler	MetaPhyler	
CLARK	CLARK	CLARK	CLARK	CLARK	

Supplementary Figure 26. Rankings of the methods using the precision metric at each taxonomic rank for three samples from the three datasets (high, medium, and low complexity). A method name higher on the list indicates better performance.

Nature Methods: doi:10.1038/nmeth.4458

CAMI_HIGH_S001, filtered

phylum	class	order	family	genus	species
MetaPhyler	FOCUS_v5	MetaPhyler	MetaPhyler	CLARK	CK_v0
Taxy-Pro_v1	MetaPhyler	FOCUS_v5	TIPP	MetaPhyler	MetaPhlAn2.0
Taxy-Pro_v0	Taxy-Pro_v1	TIPP	FOCUS_v5	FOCUS_v5	TIPP
FOCUS_v5	Taxy-Pro_v0	Taxy-Pro_v1	Taxy-Pro_v1	TIPP	MetaPhyler
CLARK	TIPP	Taxy-Pro_v0	Taxy-Pro_v0	CK_v0	Quikr
TIPP	CLARK	CLARK	CLARK	Taxy-Pro_v1	mOTU
CK_v0	Quikr	Quikr	Quikr	Taxy-Pro_v0	Taxy-Pro_v1
Quikr	CK_v0	CK_v0	CK_v0	mOTU	FOCUS_v5
mOTU	mOTU	mOTU	mOTU	Quikr	
MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	MetaPhlAn2.0	

CAMI_MED_S001, filtered

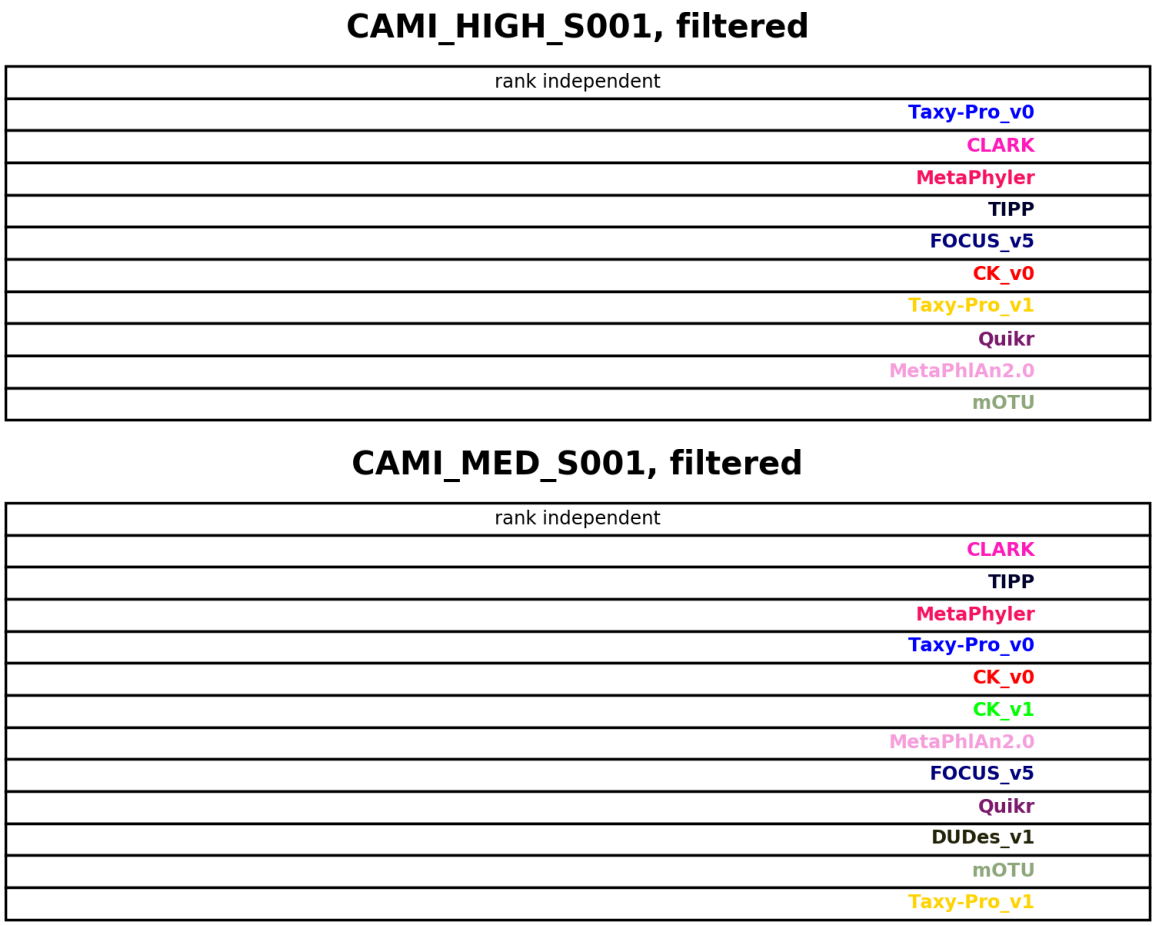
phylum	class	order	family	genus	species
CLARK	CLARK	CLARK	MetaPhyler	CLARK	MetaPhlAn2.0
CK_v0	CK_v0	MetaPhyler	TIPP	MetaPhlAn2.0	mOTU
FOCUS_v5	FOCUS_v5	TIPP	CLARK	TIPP	CK_v0
CK_v1	CK_v1	FOCUS_v5	FOCUS_v5	CK_v1	CK_v1
mOTU	DUDes_v1	CK_v1	Quikr	MetaPhyler	DUDes_v1
MetaPhyler	MetaPhyler	CK_v0	Taxy-Pro_v1	CK_v0	Taxy-Pro_v1
TIPP	TIPP	Taxy-Pro_v1	Taxy-Pro_v0	mOTU	TIPP
DUDes_v1	Taxy-Pro_v1	Taxy-Pro_v0	MetaPhlAn2.0	FOCUS_v5	FOCUS_v5
Taxy-Pro_v1	Taxy-Pro_v0	mOTU	CK_v1	DUDes_v1	MetaPhyler
Taxy-Pro_v0	mOTU	DUDes_v1	mOTU	Taxy-Pro_v1	Quikr
MetaPhlAn2.0	Quikr	Quikr	DUDes_v1	Taxy-Pro_v0	
Quikr	MetaPhlAn2.0	MetaPhlAn2.0	CK_v0	Quikr	

CAMI_low_S001, filtered

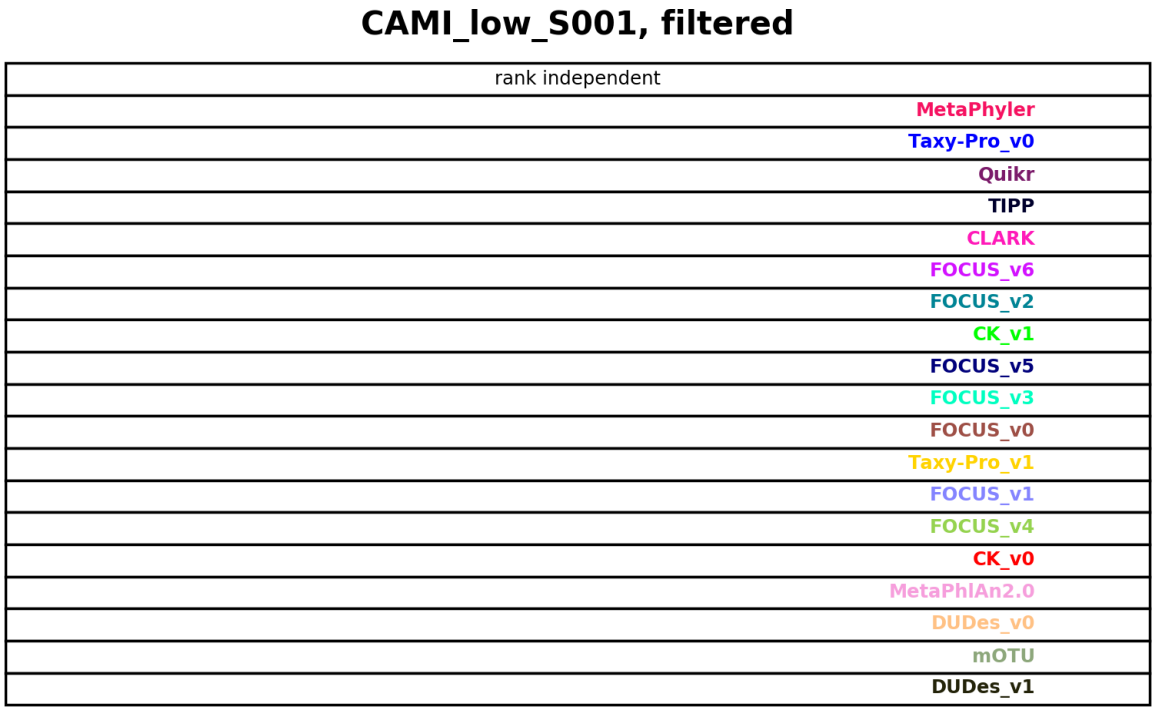
phylum	class	order	family	genus	species
Quikr	Quikr	MetaPhyler	MetaPhyler	Quikr	DUDes_v0
FOCUS_v2	MetaPhyler	Quikr	TIPP	MetaPhlAn2.0	DUDes_v1
FOCUS_v6	FOCUS_v2	TIPP	Taxy-Pro_v1	DUDes_v0	MetaPhlAn2.0
MetaPhyler	FOCUS_v6	FOCUS_v6	FOCUS_v5	FOCUS_v5	MetaPhyler
FOCUS_v5	FOCUS_v3	FOCUS_v3	FOCUS_v3	FOCUS_v3	TIPP
FOCUS_v3	FOCUS_v5	FOCUS_v5	FOCUS_v6	FOCUS_v6	mOTU
FOCUS_v0	FOCUS_v0	FOCUS_v2	FOCUS_v0	CK_v1	CK_v0
FOCUS_v4	TIPP	FOCUS_v0	FOCUS_v2	MetaPhyler	CK_v1
FOCUS_v1	Taxy-Pro_v1	Taxy-Pro_v1	Taxy-Pro_v0	FOCUS_v0	Taxy-Pro_v1
Taxy-Pro_v1	Taxy-Pro_v0	Taxy-Pro_v0	Quikr	CK_v0	Quikr
Taxy-Pro_v0	FOCUS_v4	FOCUS_v4	CK_v1	CLARK	FOCUS_v1
CLARK	FOCUS_v1	FOCUS_v1	CK_v0	FOCUS_v2	FOCUS_v4
TIPP	CLARK	CLARK	FOCUS_v4	TIPP	FOCUS_v5
CK_v1	CK_v1	CK_v1	FOCUS_v1	DUDes_v1	FOCUS_v3
CK_v0	CK_v0	CK_v0	MetaPhlAn2.0	mOTU	FOCUS_v0
DUDes_v0	MetaPhlAn2.0	MetaPhlAn2.0	CLARK	Taxy-Pro_v1	FOCUS_v6
MetaPhlAn2.0	DUDes_v0	DUDes_v0	DUDes_v0	FOCUS_v1	FOCUS_v2
mOTU	mOTU	mOTU	mOTU	FOCUS_v4	
DUDes_v1	DUDes_v1	DUDes_v1	DUDes_v1	Taxy-Pro_v0	

Nature Methods: doi:10.1038/nmeth.4458

Supplementary Figure 27. Rankings of the methods using the L1 norm metric at each taxonomic rank for three samples from the three datasets (high, medium, and low complexity). A method name higher on the list indicates better performance.



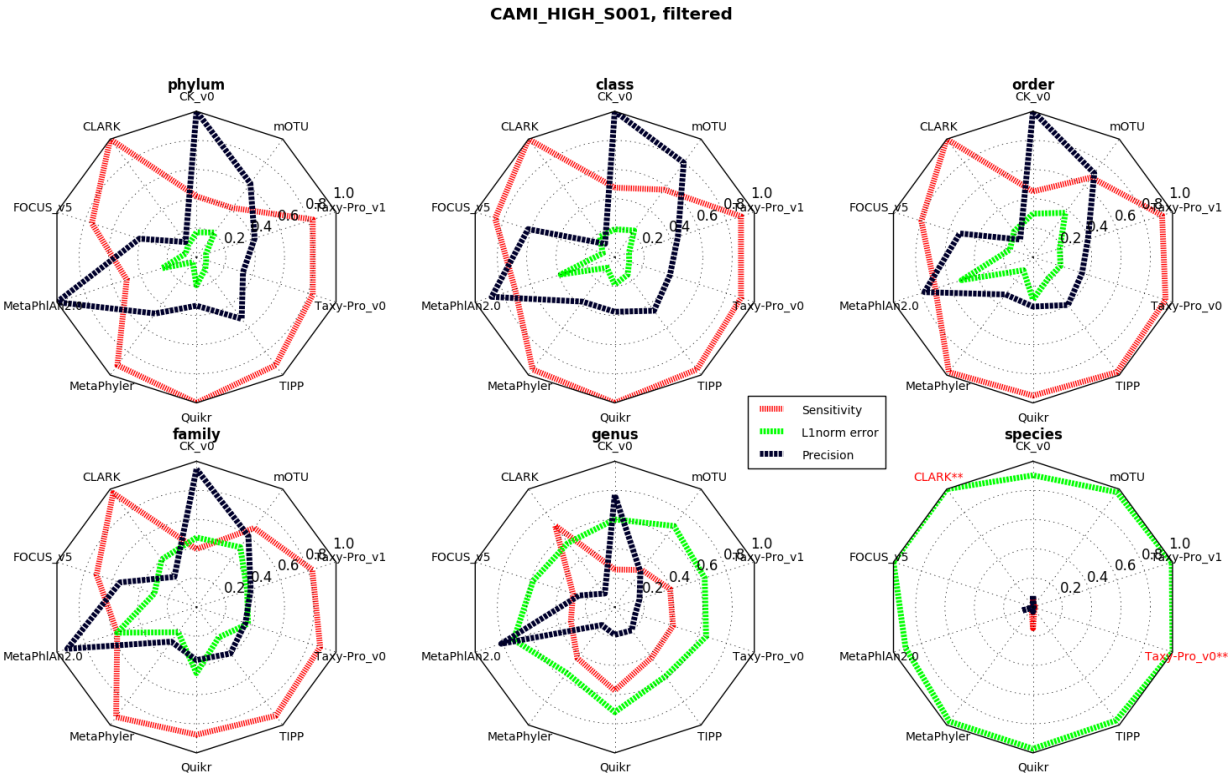
Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 28. Rankings of the methods using the Unifrac metric for three samples from the three datasets (high, medium, and low complexity). A method name higher on the list indicates better performance.

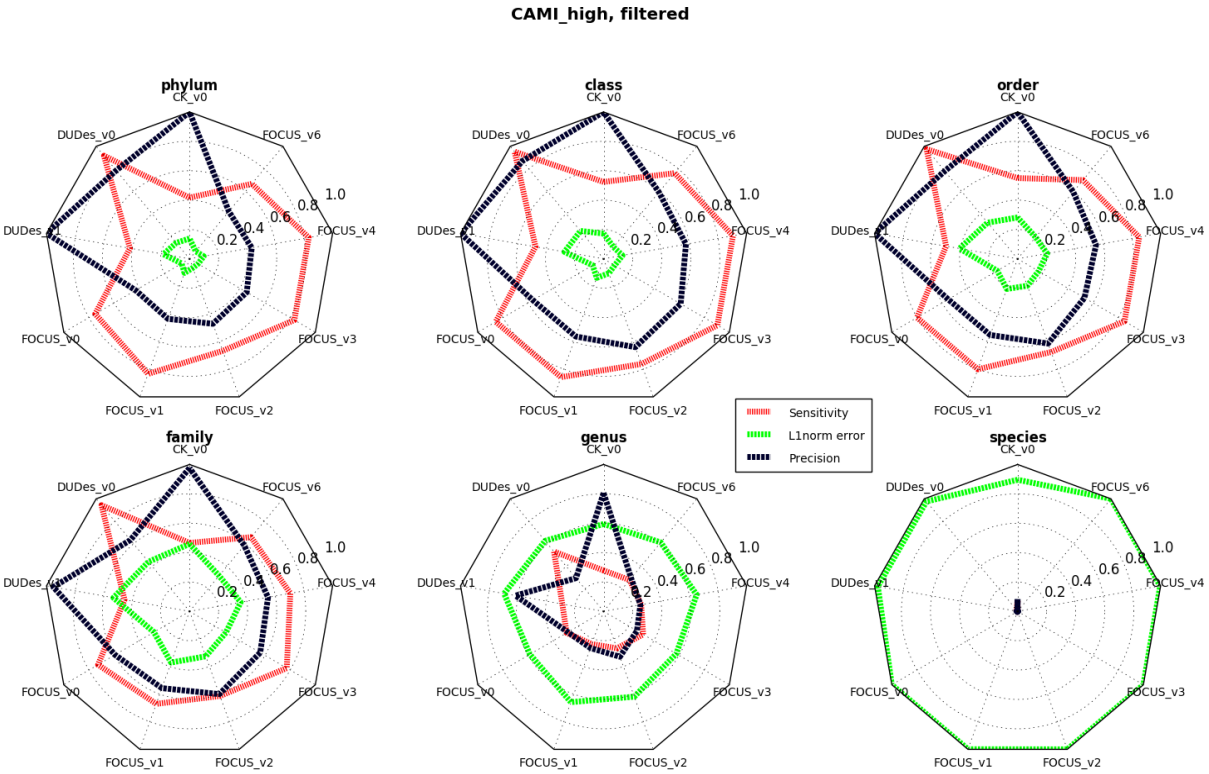
3.3.2 Absolute Performance Plots

Nature Methods: doi:10.1038/nmeth.4458



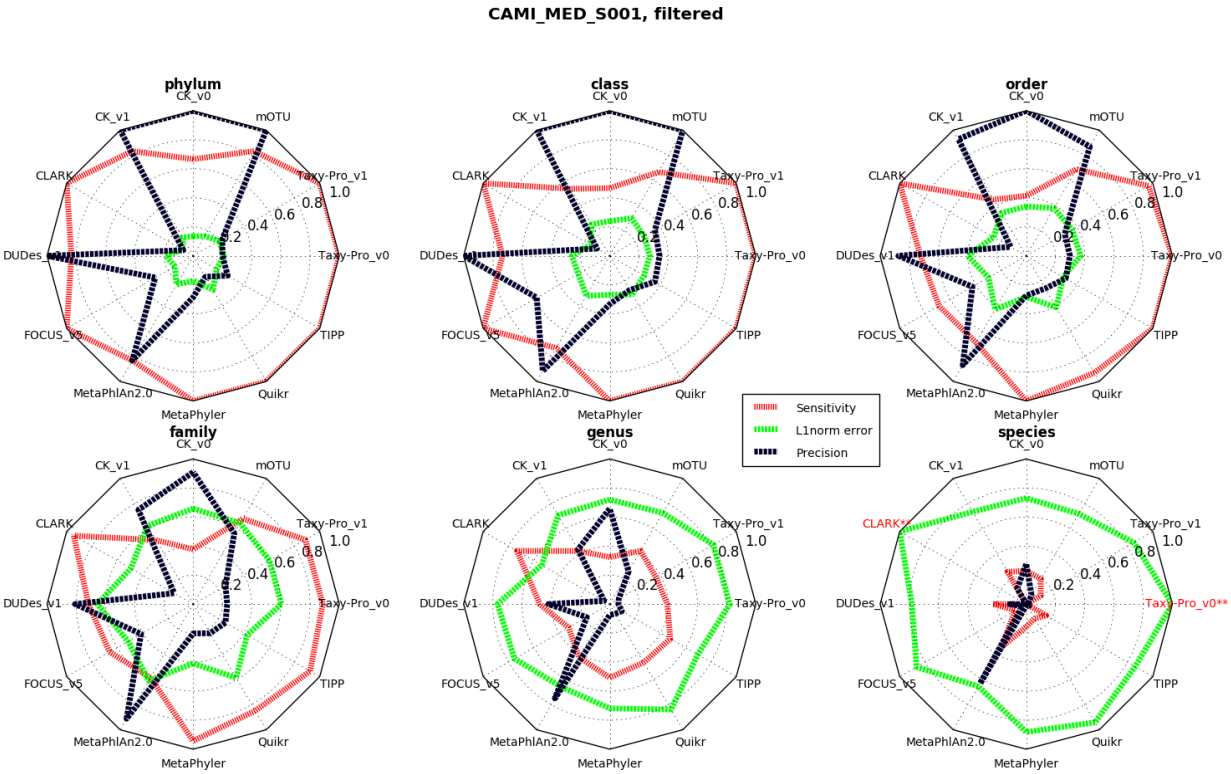
Supplementary Figure 29. Absolute performance plots of recall, precision, and L1 norm error metrics for each profiler over all taxonomic ranks for one of the high complexity samples.

Nature Methods: doi:10.1038/nmeth.4458



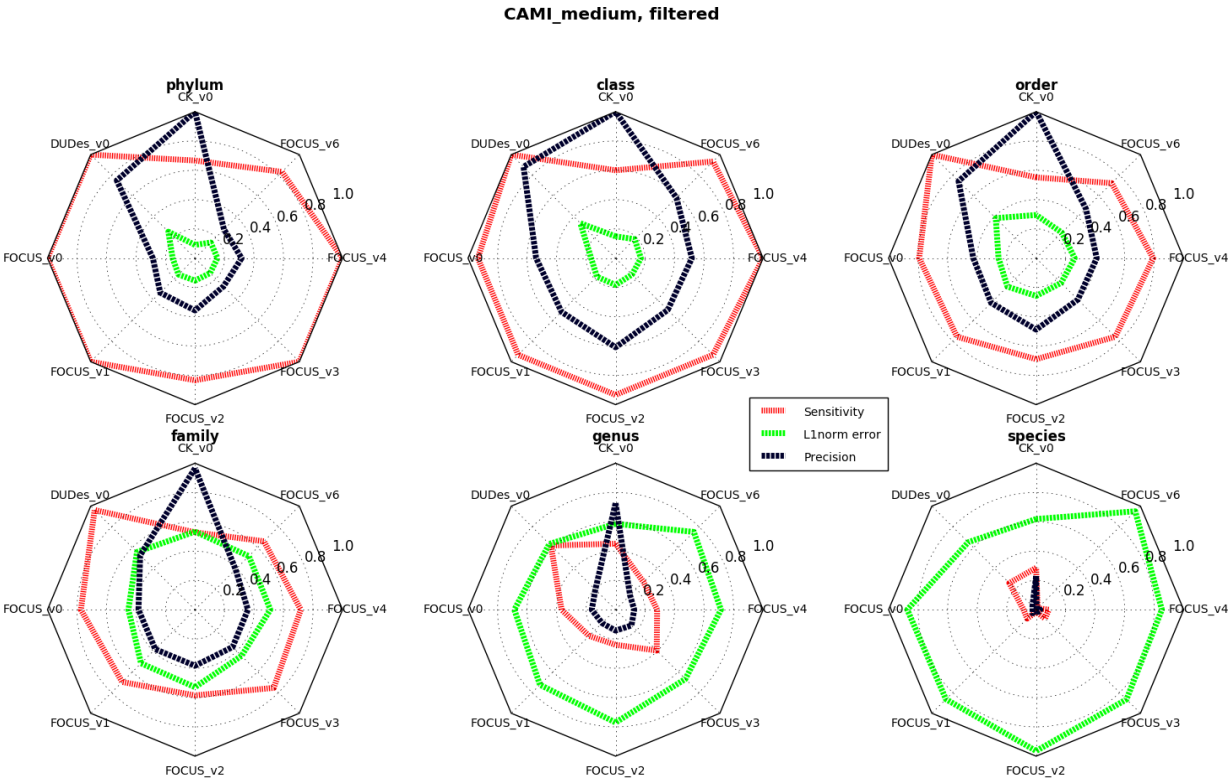
Supplementary Figure 30. Absolute performance plots of recall, precision, and L1 norm error metrics for each method over all taxonomic ranks for the "pooled" high complexity samples (i.e. all high complexity samples combined into a single input).

Nature Methods: doi:10.1038/nmeth.4458



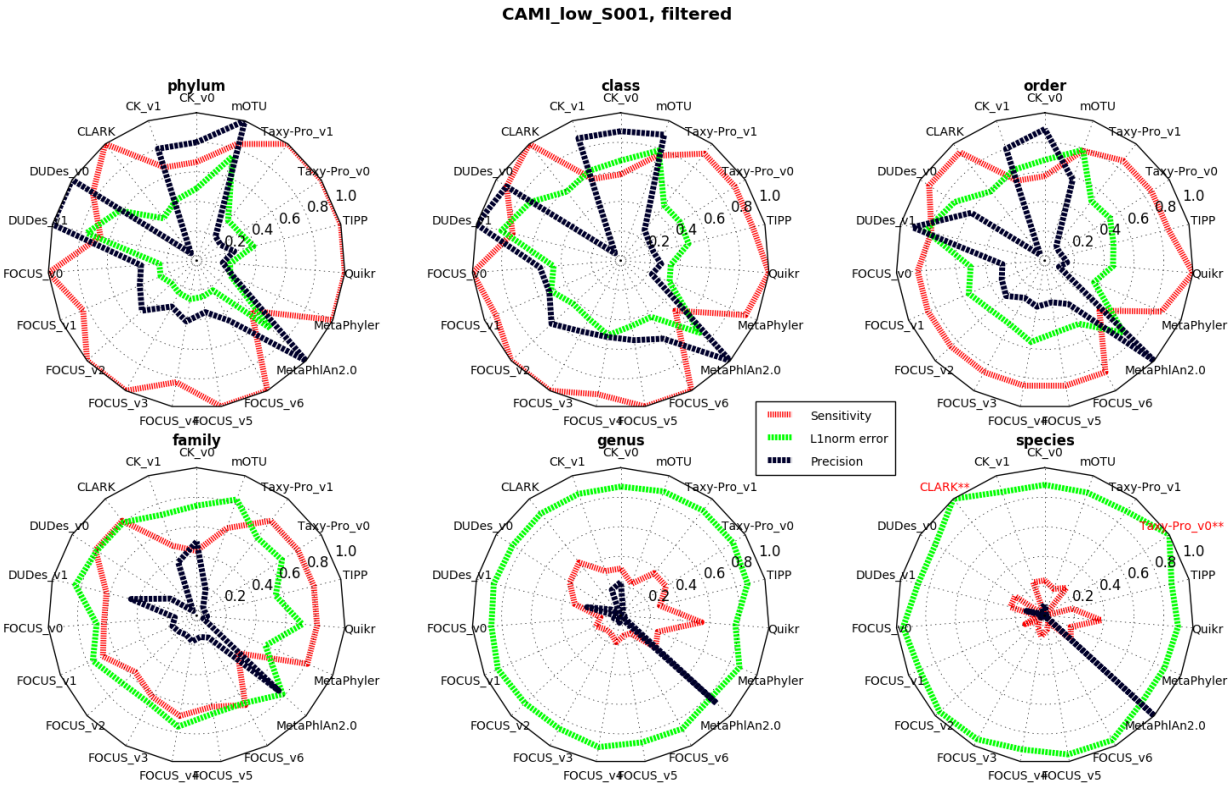
Supplementary Figure 31. Absolute performance plots of recall, precision, and L1 norm error metrics for each profiler over all taxonomic ranks for one of the medium complexity samples.

Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 32. Absolute performance plots of recall, precision, and L1 norm error metrics for each profiler over all taxonomic ranks for the “pooled” medium complexity samples (i.e. all medium complexity samples combined into a single input).

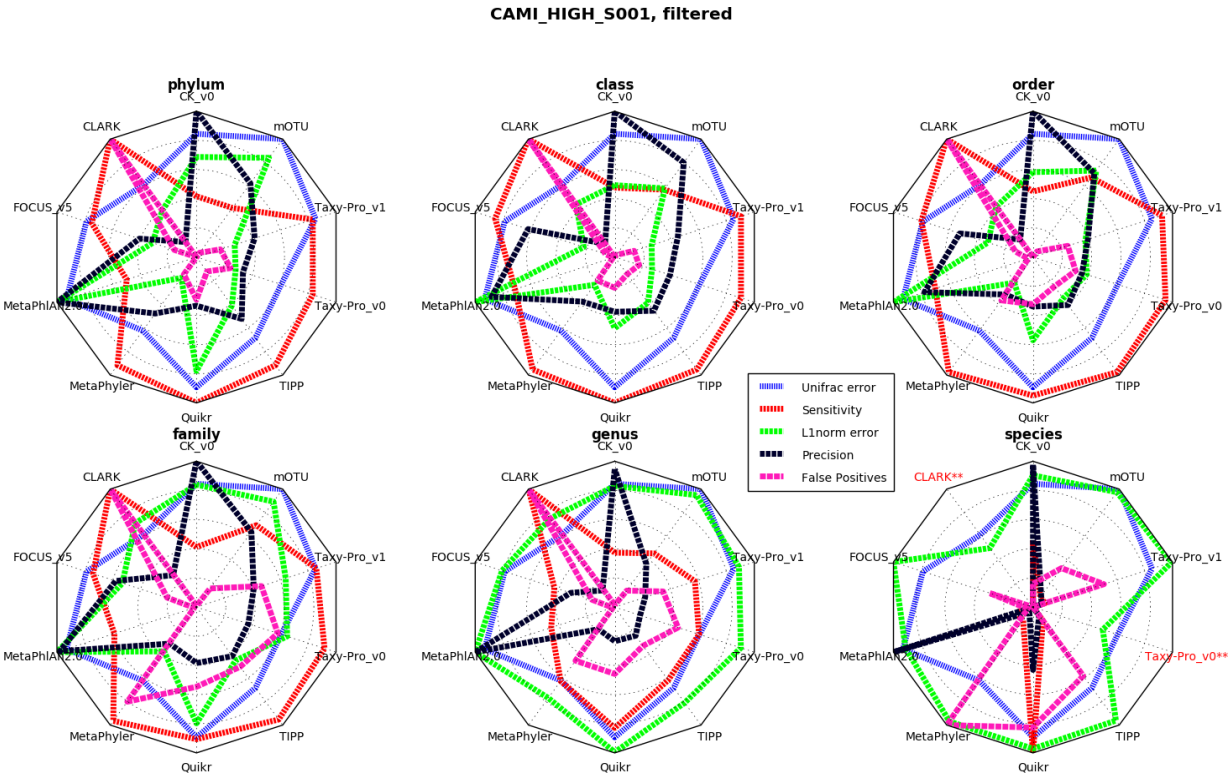
Nature Methods: doi:10.1038/nmeth.4458



Supplementary Figure 33. Absolute performance plots of recall, precision, and L1 norm error metrics for each profiler over all taxonomic ranks for the low complexity sample.

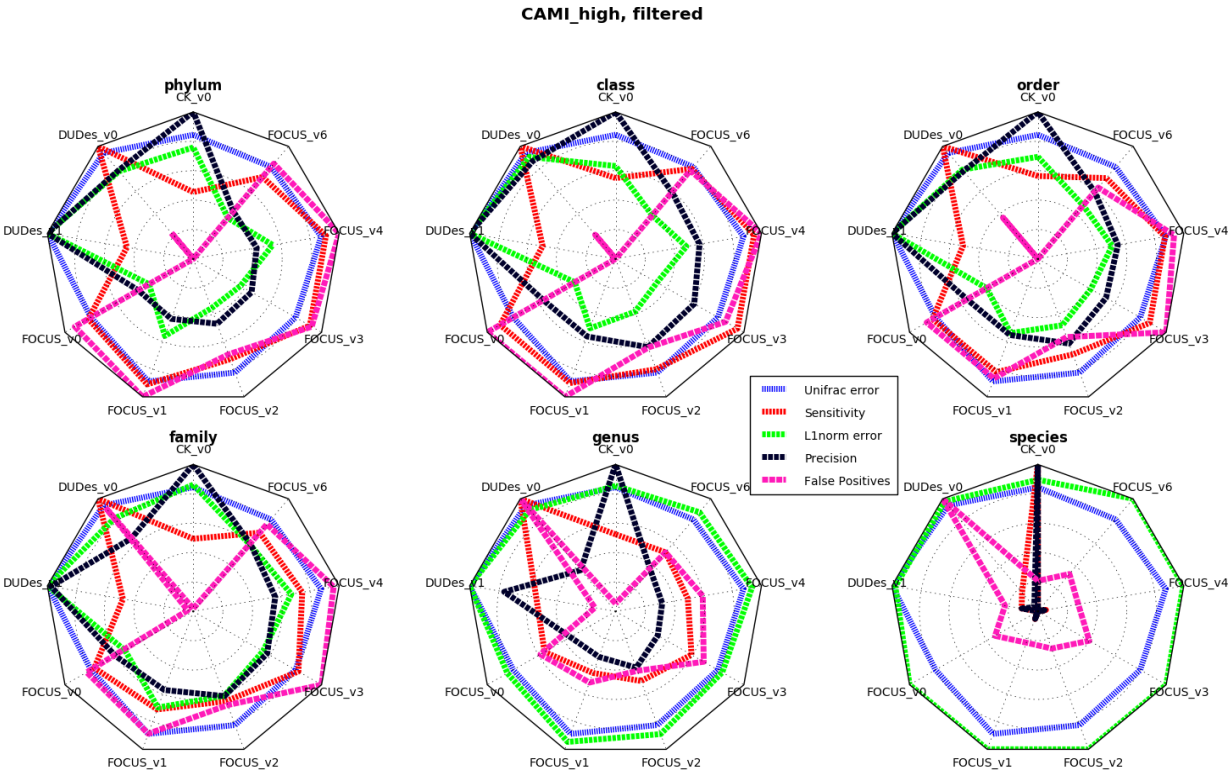
3.3.3 Relative Performance Plots

Nature Methods: doi:10.1038/nmeth.4458

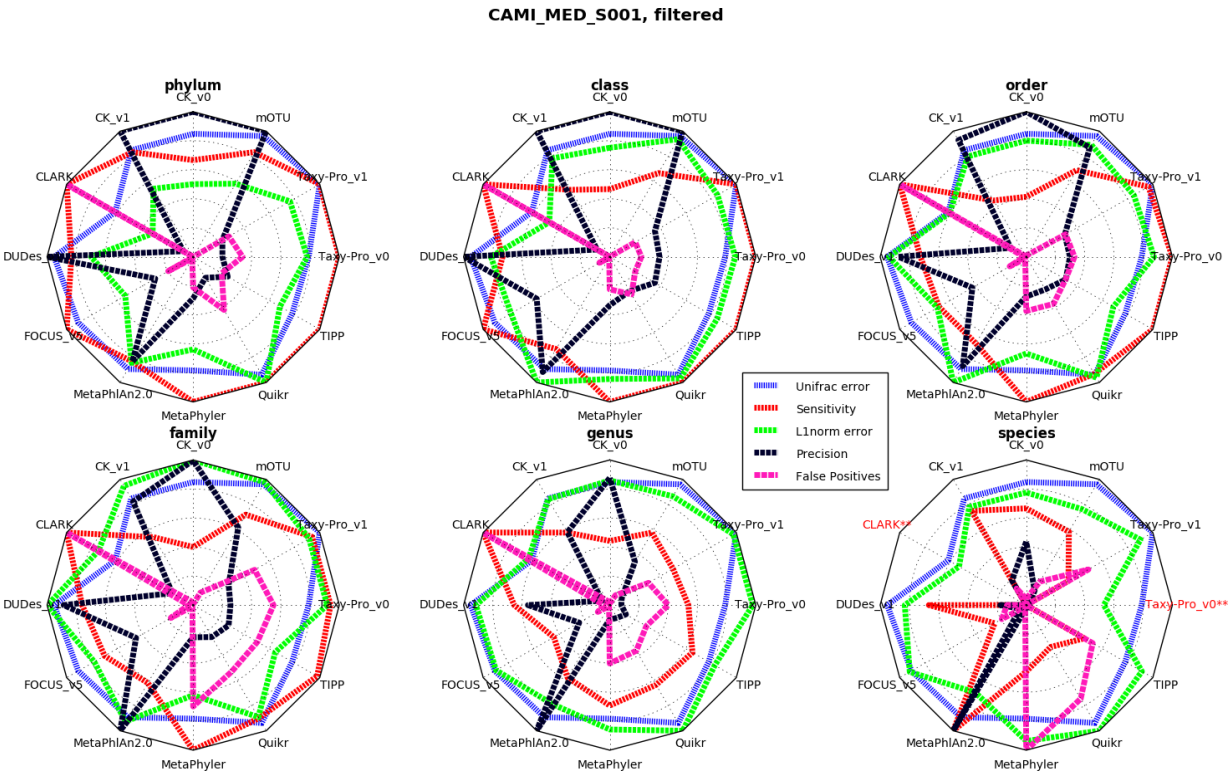


Supplementary Figure 34. Relative performance plots for each profiler and each error metric over all taxonomic ranks for one of the high complexity samples.

Nature Methods: doi:10.1038/nmeth.4458

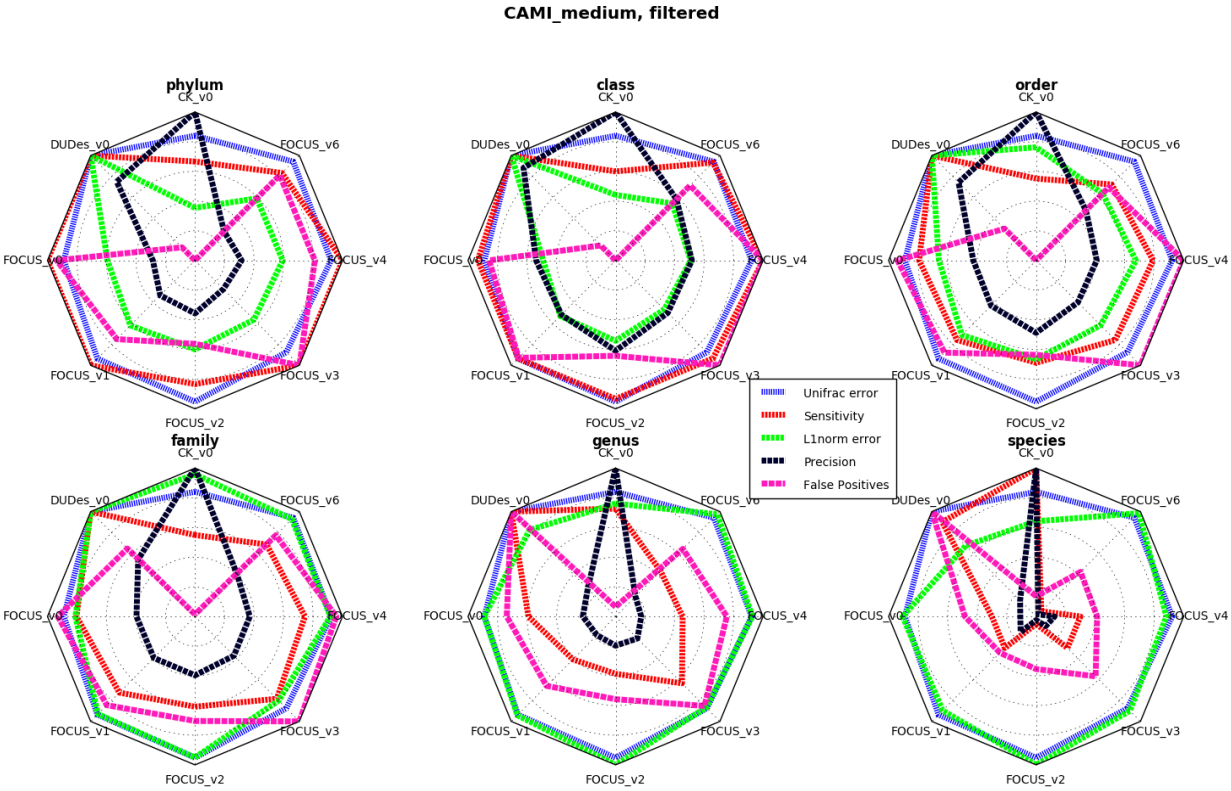


Supplementary Figure 35. Relative performance plots for each profiler and each error metric over all taxonomic ranks for the "pooled" high complexity samples (i.e. all high complexity samples combined into a single input).



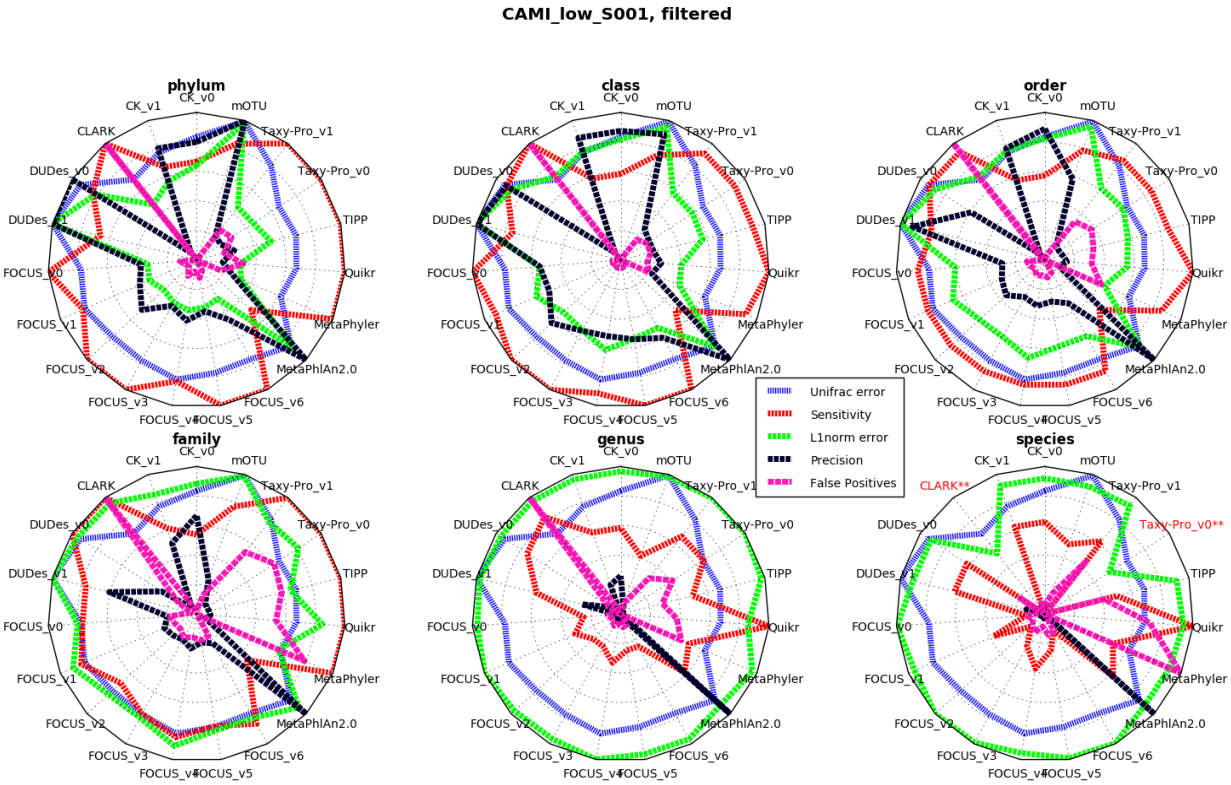
Nature Methods: doi:10.1038/nmeth.4458

Supplementary Figure 36. Relative performance plots for each profiler and each error metric over all taxonomic ranks for one of the medium complexity samples.



Supplementary Figure 37. Relative performance plots for each profiler and each error metric over all taxonomic ranks for the “pooled” medium complexity samples (i.e. all medium complexity samples combined).

Nature Methods: doi:10.1038/nmeth.4458

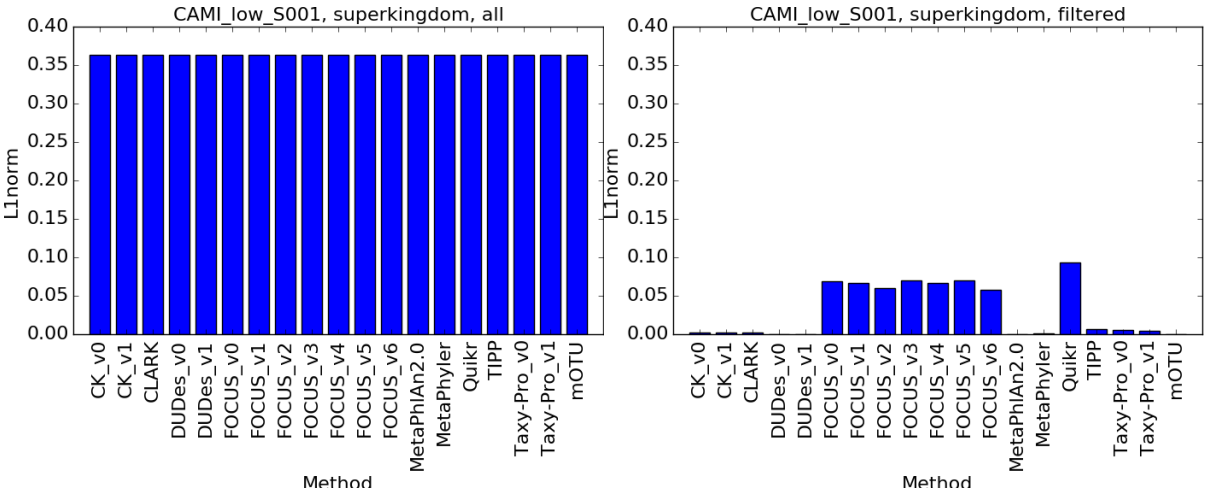


Supplementary Figure 38. Relative performance plots for each profiler and each error metric over all taxonomic ranks for the low complexity sample.

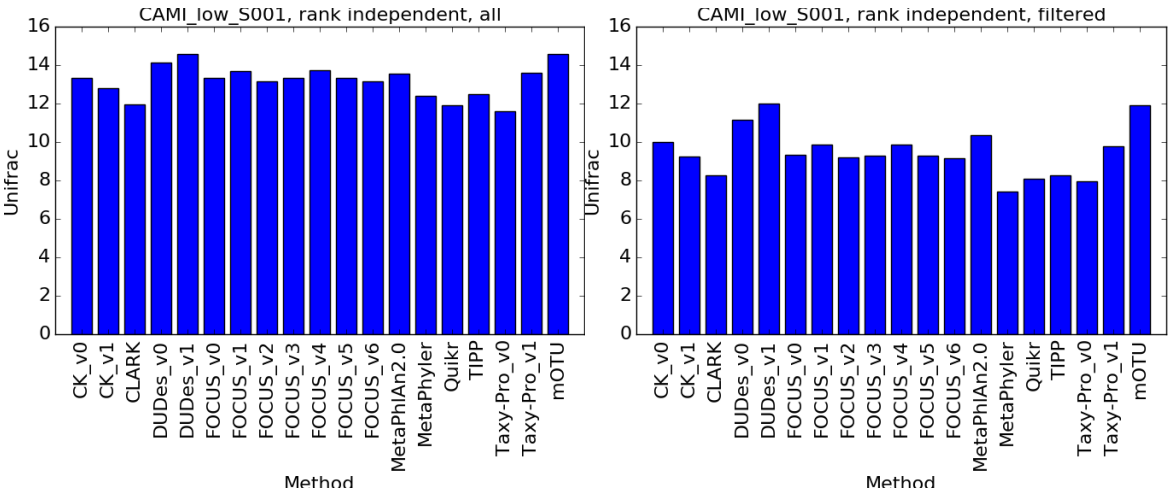
3.3.4 Performance for viruses and plasmids

The challenge datasets included sequences of plasmids, viruses, and other circular elements in addition to microbial sequence material. Viruses are commonly found in metagenome datasets generated without DNA size filtration. Here, the term “filtered” is used to indicate the ground truth did not include these data, and the term “unfiltered” indicates use of these data. Supplementary Figures 39-40 depict the changes in the L1 norm at the superkingdom level and Unifrac metric for unfiltered versus filtered data, and Supplementary Figure 41 depicts the change in recall for the unfiltered versus filtered datasets at the superkingdom rank. Recall and the L1 norm were unaffected by filtering at lower taxonomic ranks and precision was similarly unaffected at all taxonomic ranks.

Nature Methods: doi:10.1038/nmeth.4458

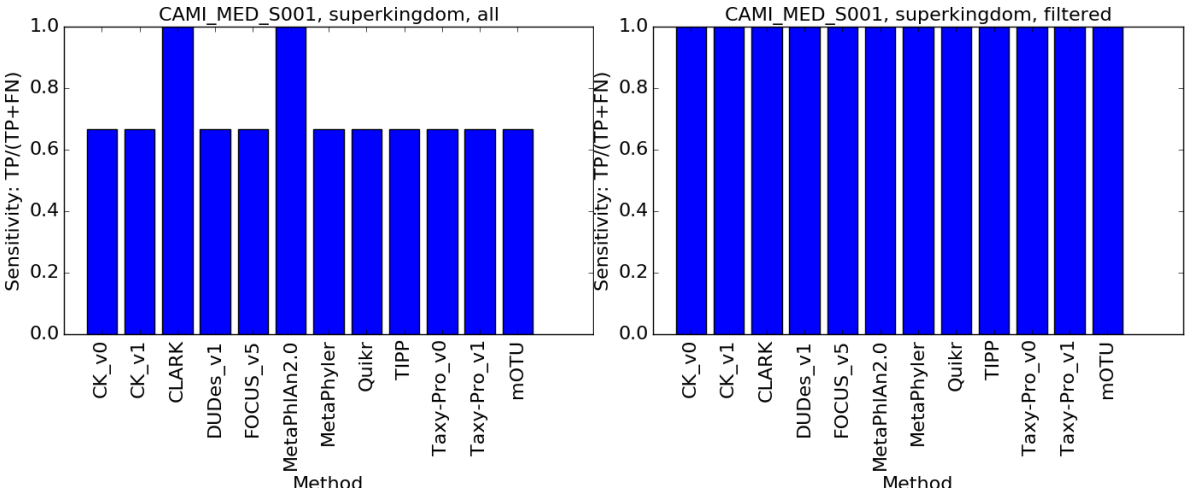


Supplementary Figure 39. L1 norm metric for each profiler on the low complexity, unfiltered sample (left) and filtered sample (right).



Supplementary Figure 40. Unifrac metric for each profiler on the low complexity, unfiltered sample (left) and filtered sample (right).

Nature Methods: doi:10.1038/nmeth.4458

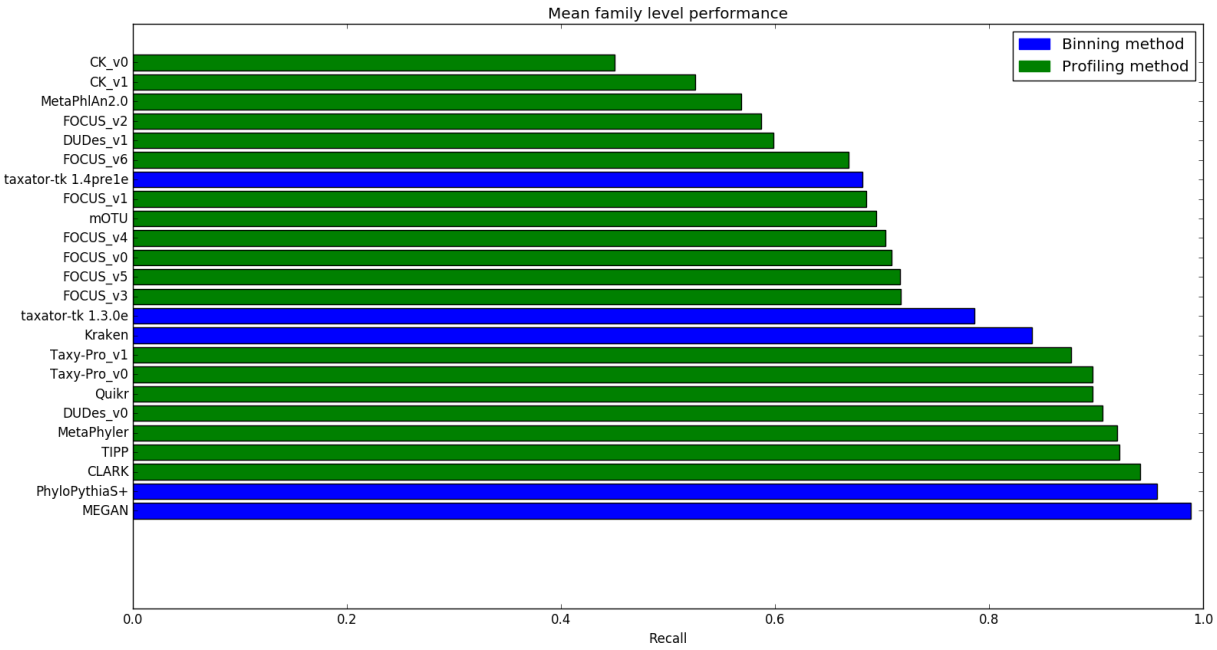


Supplementary Figure 41. Sensitivity metric at the superkingdom rank for each profiler on the low complexity, unfiltered sample (left) and filtered sample (right).

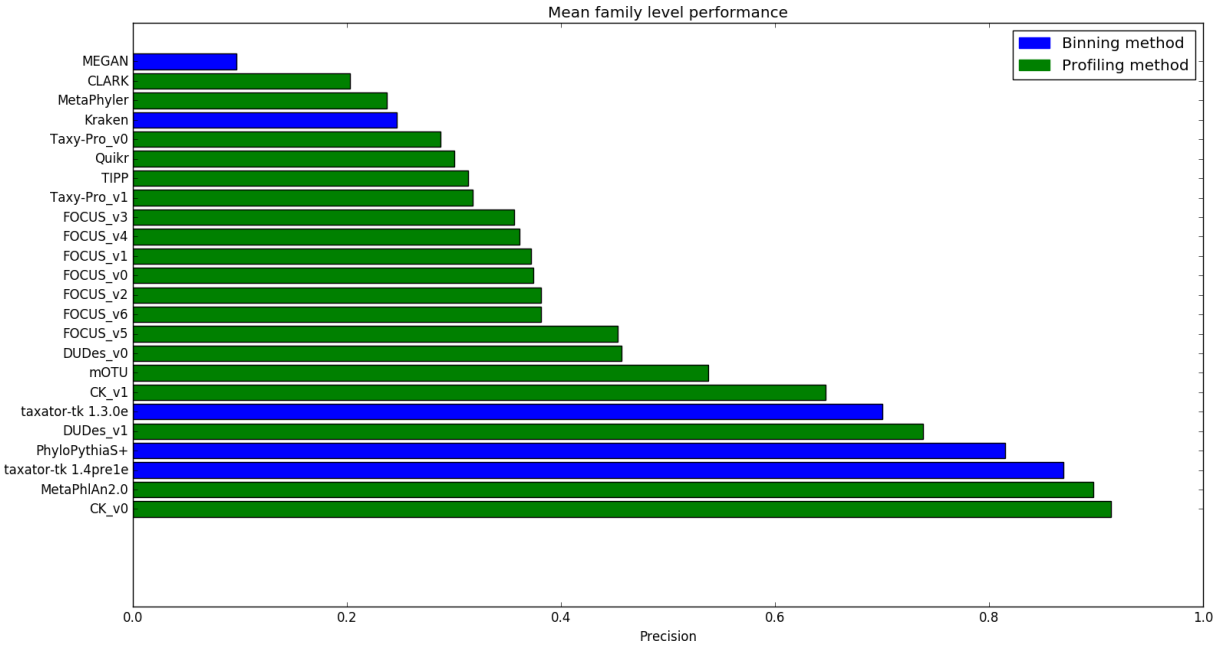
3.3.5 Performance of taxonomic binners versus profilers

We compared the profiling results to those generated from a few of the submitted CAMI taxonomic binning methods. Note that some taxonomic binners run on contigs, whereas profilers run on reads as input, which likely affects the results, as longer sequences contain more information for assignment. Supplementary Figures 42-45 depict bar charts of the various methods averaged over all samples at the family level. Binning algorithms are depicted in blue, while profiling algorithms are shown in green.

Nature Methods: doi:10.1038/nmeth.4458

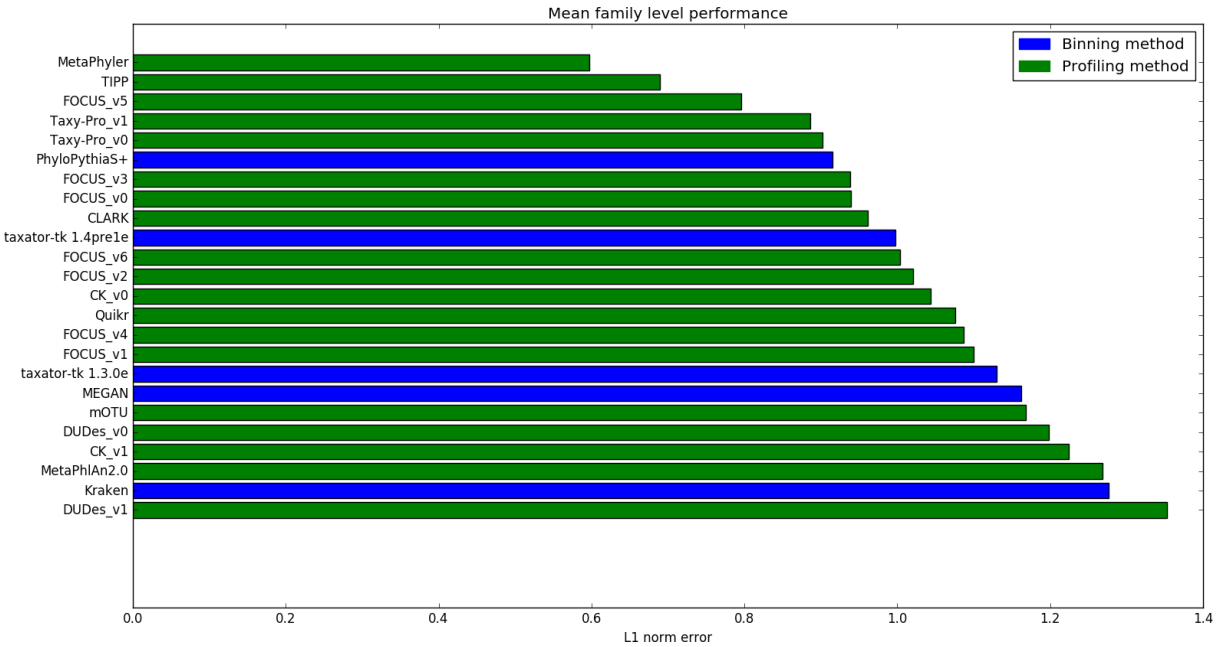


Supplementary Figure 42. Bar chart of recall at the family level for taxonomic profiling and taxonomic binning algorithms averaged over all samples (larger is better).

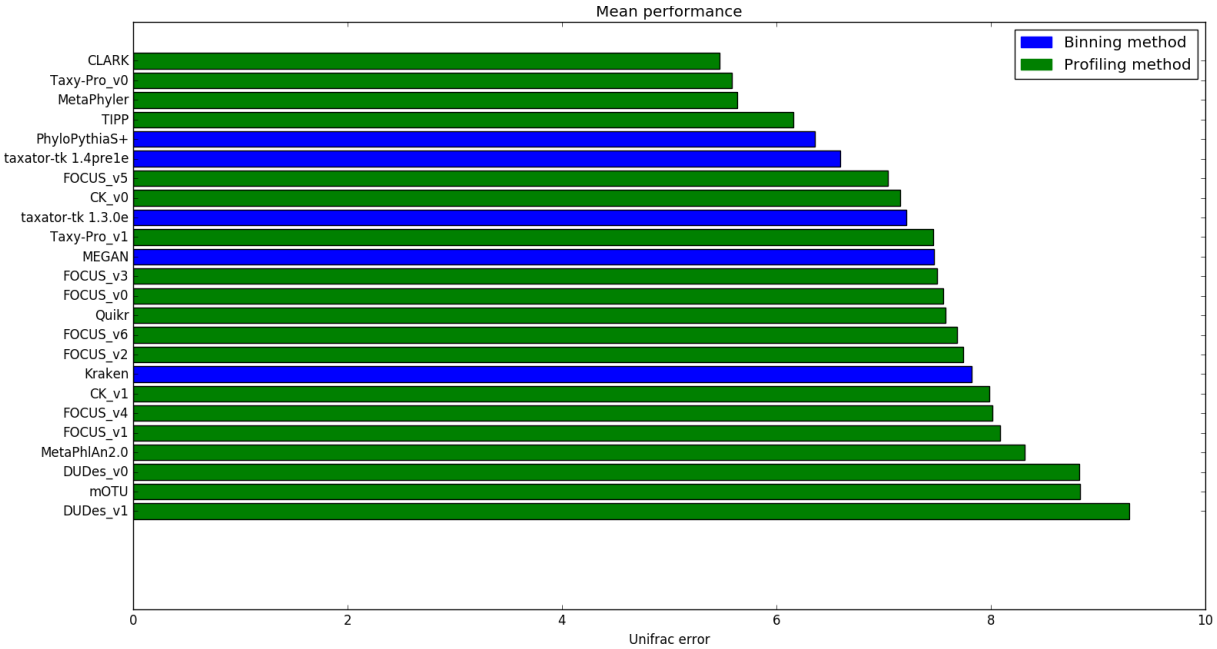


Supplementary Figure 43. Bar chart of precision at the family level for taxonomic profiling and taxonomic binning algorithms averaged over all samples (larger is better).

Nature Methods: doi:10.1038/nmeth.4458



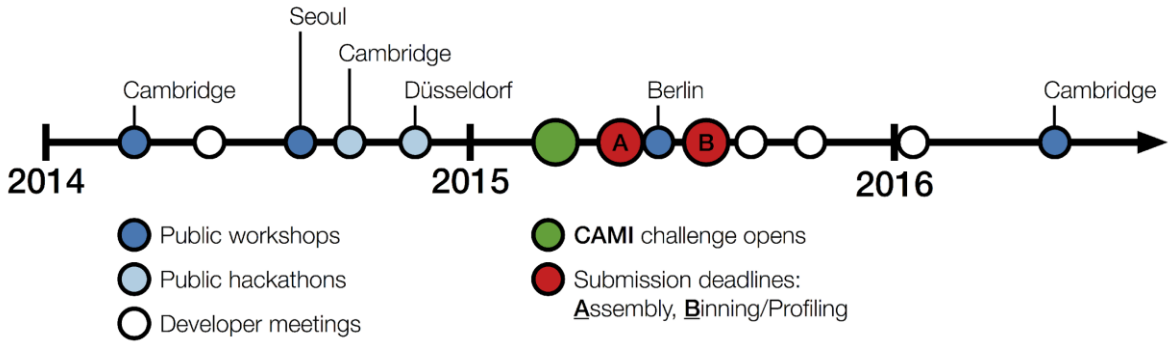
Supplementary Figure 44. Bar chart of the L1 norm error at the family level for taxonomic profiling and taxonomic binning algorithms averaged over all samples (smaller is better).



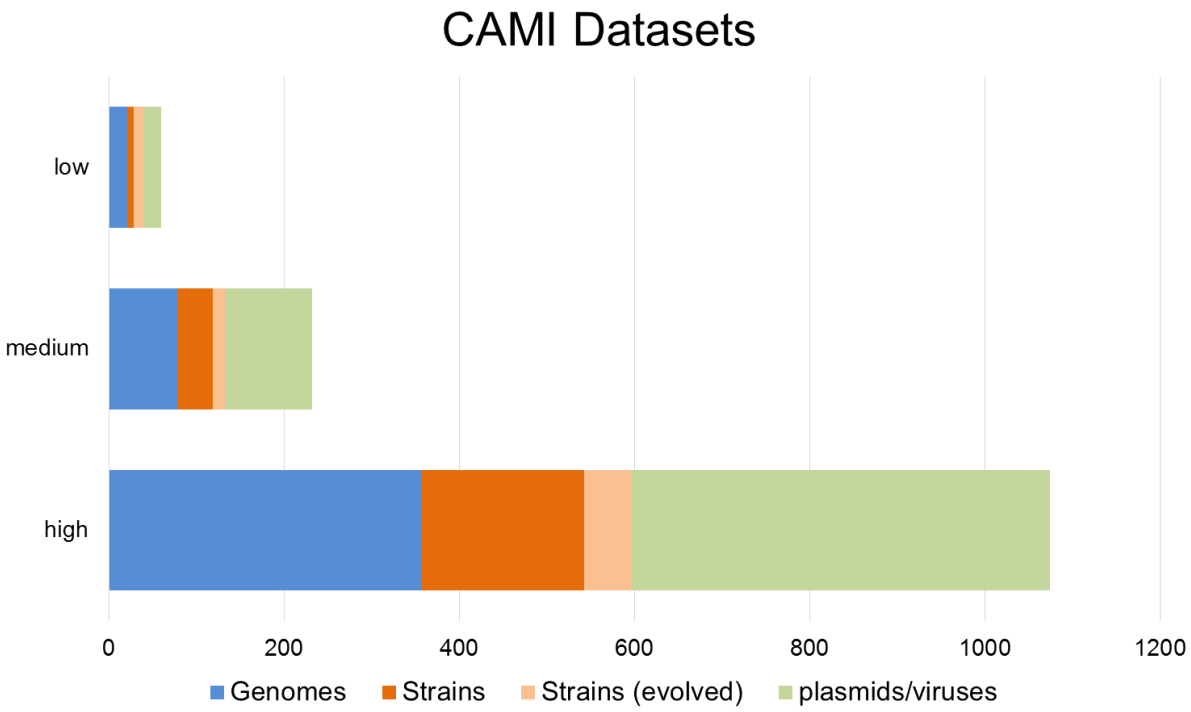
Supplementary Figure 45. Bar chart of the Unifrac error for taxonomic profiling and taxonomic binning algorithms averaged over all samples (smaller is better).

Nature Methods: doi:10.1038/nmeth.4458

3.4 Methods and Datasets



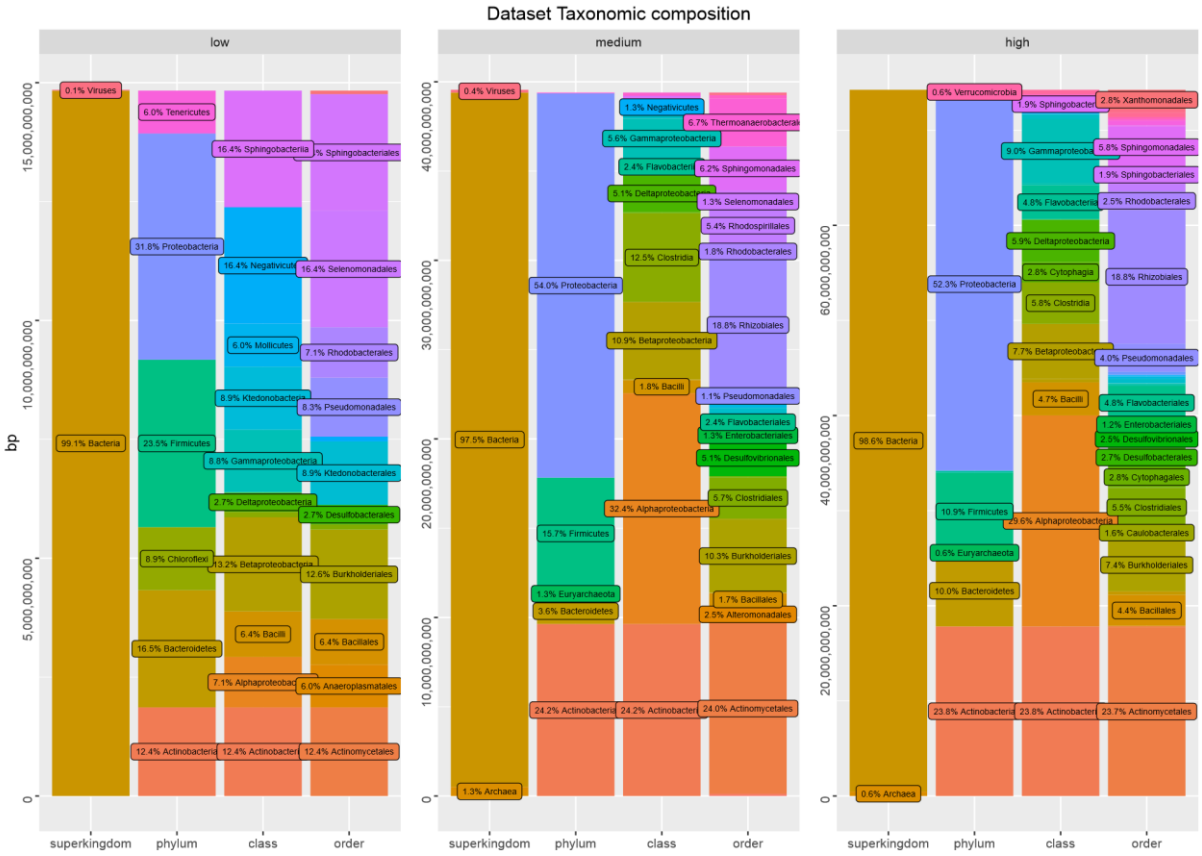
Supplementary Figure 46. Timeline of the first CAMI challenge.



Supplementary Figure 47. Number of genomes, plasmids, viruses and other circular elements in the challenge datasets. Genomes with < 95% ANI to other genomes of

Nature Methods: doi:10.1038/nmeth.4458

the datasets are shown in blue; genomes with ≥ 95% ANI to other genomes in red; simulated evolved strain genomes in light red, plasmids, viruses and other circular elements in green.



Supplementary Figure 48. Taxonomic composition of the challenge datasets for taxa from order to superkingdom (Supplementary Table 12). Sequences originating from plasmids and other circular elements are not shown.

Nature Methods: doi:10.1038/nmeth.4458

Supplementary Figure 49. Geographic locations of registered participants before the start of the CAMI challenge (January 2015).

4 References

1 Norman, A. *et al.* An improved method for including upper size range plasmids in metamobilomes. *PLoS One* **9**, e104405, doi:10.1371/journal.pone.0104405 (2014).

2 Jorgensen, T. S., Xu, Z., Hansen, M. A., Sorensen, S. J. & Hansen, L. H. Hundreds of circular novel plasmids and DNA elements identified in a rat cecum metamobilome. *PLoS One* **9**, e87924, doi:10.1371/journal.pone.0087924 (2014).

3 Eddy, S. R. A new generation of homology search tools based on probabilistic inference. *Genome Inform* **23**, 205-211 (2009).

4 Finn, R. D., Clements, J. & Eddy, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res* **39**, W29-37, doi:10.1093/nar/gkr367 (2011).

5 Eddy, S. R. Accelerated Profile HMM Searches. *PLoS Comput Biol* **7**, e1002195, doi:10.1371/journal.pcbi.1002195 (2011).

6 Pruesse, E. *et al.* SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Res* **35**, 7188-7196, doi:10.1093/nar/gkm864 (2007).

Nature Methods: doi:10.1038/nmeth.4458

7 Mitra, S., Stark, M. & Huson, D. H. Analysis of 16S rRNA environmental sequences using MEGAN. *BMC Genomics* **12 Suppl 3**, S17, doi:10.1186/1471-2164-12-S3-S17 (2011).

8 Schloss, P. D. *et al.* Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* **75**, 7537-7541, doi:10.1128/AEM.01541-09 (2009).

9 Eddy, S. R. Profile hidden Markov models. *Bioinformatics* **14**, 755-763 (1998).

10 Lagesen, K. *et al.* RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Res* **35**, 3100-3108, doi:10.1093/nar/gkm160 (2007).

11 Duda, R. O., Hart, P. E. & Stork, D. G. *Pattern Classification*. Second edn, 680.

12 Darling, A. C., Mau, B., Blattner, F. R. & Perna, N. T. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res* **14**, 1394-1403, doi:10.1101/gr.2289704 (2004).

13 Huang, W., Li, L., Myers, J. R. & Marth, G. T. ART: a next-generation sequencing read simulator. *Bioinformatics* **28**, 593-594, doi:10.1093/bioinformatics/btr708 (2012).

14 Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079, doi:10.1093/bioinformatics/btp352 (2009).

15 Mikheenko, A., Saveliev, V. & Gurevich, A. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* **32**, 1088-1090, doi:10.1093/bioinformatics/btv697 (2016).

16 Kurtz, S. *et al.* Versatile and open software for comparing large genomes. *Genome Biol* **5**, R12, doi:10.1186/gb-2004-5-2-r12 (2004).

17 Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**, 1072-1075, doi:10.1093/bioinformatics/btt086 (2013).

18 Belmann, P. *et al.* Bioboxes: standardised containers for interchangeable bioinformatics software. *Gigascience* **4**, 47, doi:10.1186/s13742-015-0087-0 (2015).

19 Zhu, W., Lomsadze, A. & Borodovsky, M. Ab initio gene identification in metagenomic sequences. *Nucleic Acids Res* **38**, e132, doi:10.1093/nar/gkq275 (2010).

20 Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9**, 357-359, doi:10.1038/nmeth.1923 (2012).

21 Ounit, R., Wanamaker, S., Close, T. J. & Lonardi, S. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* **16**, 236, doi:10.1186/s12864-015-1419-2 (2015).

22 Koslicki, D. & Falush, D. MetaPalette: a k-mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation. *mSystems* **1**, doi:10.1128/mSystems.00020-16 (2016).

23 Silva, G. G., Cuevas, D. A., Dutilh, B. E. & Edwards, R. A. FOCUS: an alignment-free model to identify organisms in metagenomes using non-negative least squares. *PeerJ* **2**, e425, doi:10.7717/peerj.425 (2014).
24 Piro, V. C., Lindner, M. S. & Renard, B. Y. DUDes: a top-down taxonomic profiler for metagenomics. *Bioinformatics* **32**, 2272-2280, doi:10.1093/bioinformatics/btw150 (2016).
25 Lozupone, C. & Knight, R. UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microbiol* **71**, 8228-8235, doi:10.1128/AEM.71.12.8228-8235.2005 (2005).
26 McClelland, J. & Koslicki, D. EMDUnifrac: Exact linear time computation of the Unifrac metric and identification of differentially abundant organisms. *bioRxiv*, doi:10.1101/087171 (2016).
27 Tsoumakas, G., Katakis, I. & Vlahavas, I. P. in *Data Mining and Knowledge Discovery Handbook* (eds O. Maimon & L. Rokach) 667–685 (Springer-Verlag, 2010).
28 Alneberg, J. *et al.* Binning metagenomic contigs by coverage and composition. *Nat Methods* **11**, 1144-1146, doi:10.1038/nmeth.3103 (2014).
29 Balvociute, M. & Huson, D. H. SILVA, RDP, Greengenes, NCBI and OTT - how do these taxonomies compare? *BMC Genomics* **18**, 114, doi:10.1186/s12864-017-3501-4 (2017).

Nature Methods: doi:10.1038/nmeth.4458