# Supplemental Information

## Benchmarking Metagenomics Tools for Taxonomic Classification

Simon H Ye, Katherine J Siddle, Daniel J Park, Pardis C Sabeti

# Supplemental Information

## In silico benchmark datasets

The benchmark datasets consist of computationally simulated reads with varying mixtures of bacterial species. Specifically, we used the 8 single ended unambiguously-mapped datasets from the NIST IMMSA repository previously evaluated in (McIntyre et al., 2017). Six of these datasets ("Buc12," "CParMed48," "Gut20," "Hou31," "Hou21," and "Soi50") were simulated to reflect distinct microbial habitats based on real human or environmental metagenomes and contain between 12 and 50 species per dataset. The seventh ("simBA525") contains 525 randomly selected species (Ounit and Lonardi, 2016) and the eighth ("NYCSM20") contains 20 species representative of the organisms of the New York City subway system. Simulation of these datasets is described elsewhere (Ounit and Lonardi, 2016).

We simulated two additional datasets to represent the ATCC Even and ATCC Staggered 20 species microbiome standards available from ATCC (ATCC ® MSA-1003™ ; ATCC ® MSA-2002™). The ATCC microbiome standards include a mix of 20 bacterial species with varying phenotypic characteristics found in the human microbiome. Simulated reads were generated using the ART Illumina read simulator modified to generate exact numbers of reads per input fasta file. Program parameters used were: HiSeq 2500 error model (HS25), paired end reads with length 150 bp and insert size of 400 with standard deviation of 50 bp. Apart from the ATCC Staggered dataset, the species abundance profiles in the rest of the datasets were evenly distributed among all species. The ATCC Staggered mixture had species abundances ranging four orders of magnitude, comprising between 0.02%-18% of the sample. For all of the ATCC and NIST samples, the sequences were simulated from the genomic DNA of component bacteria species.

The CAMI benchmarking datasets include 1 low complexity metagenome containing bacteria, and viruses, 2 medium complexity metagenomes containing archaea, bacteria and viruses, and 5 high complexity metagenomes containing archaea and bacteria. These metagenomes contain only about 30-40% of classified abundance from known taxa in these kingdoms. The rest of the abundance in these samples is characterized by added plasmids, novel new species, genera and other taxa at higher ranks, or simulated evolved strains from existing reference genomes. These short reads were simulated as described in (Sczyrba et al., 2017). There were two additional sample datasets with identical abundance profiles as the two medium complexity metagenomes but with a simulated insert size of 5000bp instead of 270bp to simulate mate-pair libraries. We excluded these long insert libraries because they are less commonly used for bulk sequencing of metagenomes and are inconsistent with the rest of our simulated datasets. Since the ground truth gold standard profiles of the CAMI samples are genome-length corrected, we instead generate ground truth profiles based on the proportions of taxa classified by individual reads.

Simulated reads for the *hg38* misclassification test were created using *wgsim* v1.8 with default parameters, for a total of 10 million paired-end reads from the GRCh38.p12 assembly of the human genome.

## In-vitro ATCC Even sequencing

The ATCC 20 Strain Even Mix Genomic Material (ATCC® MSA-1002™) in-vitro mixture was resuspended in PBS and then extracted using the MagMAX™ Pathogen RNA/DNA Kit (Thermo Cat. No. 4462359). A corresponding NTC sample was extracted using DNase/RNase clean water. The extracted RNA underwent double stranded cDNA synthesis using random hexamers. Sequencing libraries were prepared as previously described in (Gire et al., 2014). The extracted DNA and cDNA were used as input for Nextera XT library construction with 1 ng input (c)DNA and 16 cycles of Nextera XT PCR with two replicates each. The resulting tagmented libraries pooled at equal 4 nM concentrations as analyzed by a TapeStation D1000 High Sensitive tape. The libraries were paired-end sequenced on a MiSeq using a MiSeq V2 Micro kit with 300 cycles.

## Database construction

The RefSeq Complete Genomes (RefSeq CG) database was constructed using RefSeq assemblies at the "Complete Genome" assembly quality comprising 254 Archaea genomes representing 201 species, 9434 Bacteria genomes representing 3130 species, and 7530 Viral genomes representing 7346 species (Nasko et al., 2018).

## Performance metrics

The most important metrics for metagenomic classification are precision and recall. These metrics are chosen because they focus on the positive class of identified taxa, and typically not much can be said about the negative class which contains unknown unknowns. Precision is defined as the proportion of true classified taxa over all classified taxa. Recall is the proportion of correctly classified abundances over all truth abundances. The precision-recall curve was generated with abundance threshold as the classification boundary. To assess the area under the precision-recall curve, the average precision score (APS) sums the stepwise marginal gain in precision-recall, whereas the area under the precision-recall curve (AUPR) is a function which approximates the area using the trapezoidal rule on the plotted points. The precision-recall curve was calculated using scipy's precision_recall_curve, while APS and AUPR were measured using the average_precision_score and auc functions in sklearn v0.19.2 (Pedregosa et al., 2011). While APS and AUPR are normally quite similar, a quirk in scipy's precision_recall_curve inserts a final precision-recall coordinate at recall = 1.0 with precision equal to the last precision. Since this falsely indicates that the classifier eventually identified all of the truth taxa, we instead set the precision to 0 at the highest observed recall onward to adjust the AUPR calculation downward. Additionally, when multiple precision points are present for a single recall value, we take the maximal precision value to remove the effects of random ordering of taxa. Performance metrics that require the calculation of false negatives are not assessed because false negatives are poorly defined in real-world metagenomic samples. The universe of unknown unknowns cannot be known in a biological context. Therefore metrics such as the ROC curve and area under the ROC curve were not utilized because they are less relevant for real-world biological samples.

### Abundance profile similarity

To assess the quality of the abundance profiles, norms were calculated between the vector of classified species and the ground truth abundance profiles. To generate the abundance profile vector, each individual species sum is divided by the total abundance classified at a given taxonomic rank - either species or genus. Therefore each vector should have normalized magnitude of 1. To generate the 2D matrix of method similarity, pairwise distances were computed between the species abundance profiles for each classification method. The median L2 distance across the benchmarking samples was taken as the pairwise similarity for each (*method1*, *method2*) pair. The hierarchical clustering was performed using the Nearest Point algorithm on the median L2 distances. The L2 distance and hierarchical clustering for all comparisons were calculated using scipy v1.1.0 (Jones et al., 2001)

### Individual classifier quirks

The classifiers frequently had individual quirks complicating cross-classifier comparisons. A common individual classifier quirk is not outputting the unclassified abundance. For example, GOTTCHA classifies bacteria and viruses with separate abundance reports which must be merged, while PathSeq and Bracken performs genome-length normalization for abundances even though we're evaluating based on read count abundances. Many methods did not output a specific abundance profile, so a profile was generated from the read-level taxon hits by assigning to the LCA taxon of all hits for each individual read, then cumulatively summing up the taxonomic tree.

### Supporting software used

Snakemake v5.2.2 was used to manage the workflow and benchmarking of the classifiers (Köster and Rahmann, 2012). Duplicate content was estimated using a bugfixed version of *cd-hit-dup* v4.6.1. Jupyter notebook v5.6.0 (Kluyver et al., 2016), pandas 0.23.4 (McKinney, 2010) and matplotlib v2.2.3 (Hunter, 2007) on Python v3.5.4 was used for analysis and plotting. The necessary Python software was installed via miniconda v4.4.10 and the bioconda package channel (Grüning et al., 2018).

### Computational Environment

The classifiers were benchmarked on AWS's EC2 platform using the r4.8xlarge instance type. This instance utilizes Intel Xeon E5-2686 v4 processors with 16 cores (32 HyperThreaded cores) has 240 Gb of memory, and has support for all instruction set extensions used by the tested classifiers for optimized runtime performance. For some classifiers, database creation required more than 240 Gb of memory. These classifiers were benchmarked on the r4.16xlarge instance type with 480 Gb of memory. Disk storage was comprised of st1 EBS instances, which are backed by hard disk drives. While st1 drives have poor random read/write performance, they have high linear throughput which match the workloads of metagenomics tools - they primarily load large databases into memory and sequentially process large FASTQ files. EBS instances have varying IO performance by drive size, which slightly affected the speed of database loading for certain classifiers such as Kraken. The computational time benchmark was measured by running a single instance of each classifier provided all 32 cores and memory, even if the classifier cannot properly utilize these resources with efficient parallelization and could have higher throughput if processed concurrently with other samples.

### Individual Classifier Descriptions

A description of all classifiers, as well as example simplified commands used with paired-end input and quirks of each method is listed below. Filenames in commands are templated in the curly braces such as: {input}, {output}, {db} representing input files, output files, and database-related options.

Bracken v1.0.0-9aaaec is an add-on that utilizes the read classification output from standard Kraken (Lu et al., 2017). It re-estimates the taxonomic abundance profile by accounting for the uniqueness of the reference genomes based on self-classification of its reference genomes, which strongly reduces the false-positive rate of standard Kraken and implicitly normalizes for genome length. Databases were constructed using their respective kraken library genomes for the default and RefSeq CG databases. *kraken-filter* was used to filter raw classifications at the 0.05 threshold.

Bracken command:

```
est_abundance.py -i {input} -k {db} -o {output}
```

Centrifuge v1.0.4-91dfff is an FM-index approach that first compresses the reference genomes by removing redundant segments among highly similar genomes (Kim et al., 2016). It searches for exact matches of at least 22 bp, and the query search is restarted each time after a mismatch occurs. Instead of just classifying a single taxon label or LCA taxon, each matched segment can have up to 5 labels applied. Expectation maximization (EM) is performed on the raw abundance profiles to generate a finalized abundance profile. The default database selected was the compressed database from ftp://ftp.ccb.jhu.edu/pub/infphilo/centrifuge/data/p_compressed+h+v.tar.gz containing human, viruses, and compressed bacteria/archaea sequences updated 2016/12/06.

Centrifuge command:

```
centrifuge -x {params.db} -1 {input.1} -2 {input.2} --report-file
{output.report} --threads {threads} | pigz -9 > {output.data}
```

CLARK v1.2.5 is a method similar to Kraken which uses 31bp k-mers to classify reads. Instead of storing the LCA taxon of each k-mer, it removes non-unique k-mers (except for those that occur on the same arm of a chromosome) and rare k-mers to reduce the noise is inherent in using raw k-mers for classification (Ounit et al., 2015). CLARK-S is a further extension that utilizes spaced k-mers with wildcard gaps instead of contiguous 31bp k-mers. The default database contains  RefSeq bacteria and viruses and was constructed on 2018/04/25.

CLARK command:

```
CLARK -n {threads} {db} -P {input} -R {output}
```

CLARK-S command

```
CLARK-S -n {threads} {db} -P {input} -R {output}
```

DIAMOND v0.9.21-131bd4 is a BLASTx-like aligner that takes DNA query sequences and aligns them to a protein database, typically the nr database used by BLASTx (Buchfink et al., 2015). To improve on the speed of BLASTx while maintaining sensitivity, it uses relatively long spaced seeds with weight of 12 and lengths of 15-24 letters by default. Instead of using the full amino acid alphabet, it uses a reduced alphabet of only 11 letters. Additionally it does a double-indexed sort and linear iteration of the seed locations in both the query and reference database to improve cache locality. DIAMOND does not natively handle paired-end reads, so paired ends were considered as separate reads in the fasta input file. After taxonomic assignment of each paired end, an additional processing step was run to compute the LCA taxon of the two paired ends and reassigned to the combined read. The default database selected used the BLAST *nr* reference from 2018/04/20.

DIAMOND command:

```
diamond blastx --verbose -p {threads} --outfmt 102 -q {input} -o
{output.tax}
```

GOTTCHA v1.0c-e4067a uses databases of unique 24 length k-mers to each organism (Freitas et al., 2015). These databases are specific to each taxonomic family (such as species, genus, family etc), and have variants with human k-mers removed. Query reads are broken into 30 bp fragments and searched for maximal exact matches using bwa mem to the reference database. The default databases for bacteria and viruses at the species and genus levels were downloaded from ftp://ftp.lanl.gov/public/genome/gottcha/, version v20150825 with the xHUMAN3x variant. Since GOTTCHA does not generate a combined abundance of bacteria and viruses, the abundance of each taxa was determined by dividing the total sum of taxa covering bp by the cumulative number of bp covered in the concatenated bacteria and viral reports.

GOTTCHA command:

```
gottcha.pl --database {db.bacteria} --input {input} --mode all
--dumpSam --prefix {output} --threads {threads}
gottcha.pl --database {db.viral} --input {input} --mode all --dumpSam
--prefix {output} --threads {threads}
```

Kaiju v1.7.0 is a DNA to protein classifier that relies on the FM-index to reduce memory requirements (Menzel et al., 2016). Six-frame translated query sequences are first split into fragments ending with putative stop codons and searched for maximal exact matches (MEMs) in the FM-index. Queries are taxonomically assigned to the longest MEM, or to the LCA taxon if it matches multiple taxons. Kaiju also implements a greedy search mode which allows some mismatches at the left end of fragments, by searching backwards in the BWT. The default database used the BLAST *nr* reference from 2018/04/20.

Kaiju command:

```
kaiju -t {db.nodes} -f {db.fmi} -i {input.1} -j {input.2} -o {output}
-z {threads}
```

Karp v1.0-88c5b1 is a method that utilizes the Kraken approach of matching 31 bp k-mers as a first step (Reppell and Novembre, 2017). Afterwards, it performs local alignment and likelihood estimation based on the read qualities encoded in the input FASTQ. The harp filter was disabled and likelihood threshold set to a low value due to not having enough classified reads on our datasets otherwise. Finally, it performs EM on the abundance profiles to generate the final species profile. The default database was constructed using bacteria rRNA sequences from SILVA release 132.

Karp command:

```
karp -c quantify --threads {threads} {db} -f {input.1} -q {input.2}
--out {output} --like_thresh 19 --no_harp_filter --collapse
```

Kraken v1.0-352e78 is k-mer based classification method that searches for 31bp k-mers from the query sequence in a precomputed database that matches k-mers to the lowest common ancestor (LCA) taxon of all genomes that contain that taxon (Wood and Salzberg, 2014). The default database selected includes RefSeq complete bacterial and viral genomes constructed on 2018/05/13. A filtering threshold of 0.05 was selected due to it showing the best precision/recall by the authors' own measurements.

Kraken command:

```
kraken --threads 32 --fastq-input --bzip2-compressed
{input_1.fastq.bz2} {input_2.fastq.bz2} | tee >(pigz --best >
{output.reads}) | kraken-filter --threshold 0.05 | kraken-report >
{output.report}
```

Kraken2 v2.0.7-beta-fb4522 is k-mer based classification method that searches for 35bp k-mers from the query sequence in a precomputed database that matches k-mers to the lowest common ancestor (LCA) taxon of all genomes that contain that taxon (Wood and Salzberg, 2014). It is faster than Kraken and requires less memory, but has a slight chance of false positive classifications. The default database selected includes RefSeq complete bacterial and viral genomes from 2018/05/13. A filtering threshold of 0.05 was selected due to it showing the best precision/recall by the authors' own measurements.

Kraken2 command:

```
kraken2 --threads 32 --fastq-input --confidence 0.05
--bzip2-compressed {input_1.fastq.bz2} {input_2.fastq.bz2} --output
{output.reads}) --report {output.report}
```

KrakenUniq v0.4.8-70cd32 (formerly named KrakenHLL) is an extension on the standard Kraken algorithm which outputs additional information about the uniqueness of k-mers assigned to each taxa (Breitwieser and Salzberg, 2018). Although not used for benchmarking, this additional information can be used to filter false-positive calls for taxa with low k-mer uniqueness. The default database containing RefSeq complete archaea, bacterial, and viral genomes was constructed on 2018/04/25.

KrakenUniq command:

```
krakenuniq --preload --db {db} --threads {threads} --paired
--bzip2-compressed --fastq-input {input_1.fastq.bz2}
{input_2.fastq.bz2} --output {output.reads} --report-file
{output.report}
```

k-SLAM v1.0-6cbf5a is a k-mer based classification method that first queries the database index using 32-bp k-mers and performs local alignment on any resultant hits (Ainsworth et al., 2017). It also incorporates specialized handling of paired-end reads as well as methods to differentiate between alignments to regions of high sequence conservation. The default database is created using the built-in `install_slam.sh` script to download bacterial and viral genomes from RefSeq on 2018/04/25.

k-SLAM command:

```
SLAM --db {db} --num-reads-at-once 50000 --output-file {output}
```

MegaBLAST v2.7.1+ is an accelerated version of BLAST that uses longer initial seed lengths at 28 nucleotides compared to 11 nucleotides in standard BLASTn to reduce the number of alignments that need to be calculated (Morgulis et al., 2008). Although BLASTn was not evaluated here because it cannot process large numbers of metagenomic sequencing reads in a reasonable amount of time, MegaBLAST is fast enough to be feasible for this task. It still remains the slowest DNA classifier overall. The option of maximum 10 HSPs was used to reduce the number of hits reported for the same subject sequence. BLAST can output a list of taxa for each individual BLAST hit for a query sequence. For each query read, hits with a minimum bit score of 50 and e-value of 0.01 within 10% of the best hit (in terms of bit-score) were kept. The LCA taxon of all the kept hits was taken as the taxon assignment for the sequence.

MegaBLAST command:

```
blastn -db {db} -out {output} -max_hsps 10 -num_threads {threads}
-task megablast -outfmt "6 qseqid sseqid pident length mismatch
gapopen qstart qend sstart send evalue bitscore sgi sacc staxids
sscinames scomnames stitle"
```

MetaOthello (git rev. 15ded5) is a fairly unique approach that relies on probabilistic hashing of k-mers (Liu et al., 2018). The database is constructed with two hash functions calculated on taxonomically unique k-mers extracted from the reference database. The two hash functions index into arrays which are bitwise XOR'd to generate the taxon integer id. Hash collisions of different k-mers to the same taxa would lead to ambiguous assignments and are removed from the database. K-mers not seen during database construction may lead to spurious taxon assignments if the XOR'd result matches a valid taxon id. This probabilistic approach offers significant memory savings over other k-mer scanning methods, but can lead to spurious classifications. Since metaOthello does not produce an abundance profile, individual read classifications at the species and genus levels were divided by the total number of reads. The default database was downloaded from https://drive.google.com/open?id=0BxgO-FKbbXRIa0Flc3Q4bWtycGM containing 31-bp long bacterial k-mers.

MetaOthello command:

```
classifier {db} {output.dir} 31 {threads} fq PE {db.idmap} {db.names}
{input} | pigz --stdout -9 {output.reads}/taxo_assignment.txt >
{output}
```

MetaPhlAn2 v2.6.0-06c962 is marker gene alignment approach that uses precomputed customized databases containing clade-specific marker genes (Truong et al., 2015). Query reads are aligned via bowtie2 to the marker genes. Classification speed is very high since the reference database is much smaller than approaches utilizing full genomes. The default database version is the v20 db.

MetaPhlAn2 command:

```
metaphlan2.py --input_type fastq --nproc {threads} --bowtie2out
{output.bowtie2out} {db} {input} {output.report}
```

MMseqs2 v3-be8f6 is another BLASTx alternative aligning DNA to protein sequences (Steinegger and Söding, 2017). For the lookup phase, it looks for two consecutive weight 7 spaced seed matches to a target sequence. These two seeds have to be positionally-consistent, (on the diagonal of the query/target position graph) before continuing to the alignment phase. MMseqs2 has a specialized taxonomy output which was used for benchmarking. MMseqs2 did not have any special handling for paired end reads and they were treated as separate sequences. The default database used the BLAST *nr* reference from 2018/04/20.

MMSeqs2 command:

```
mmseqs2 createdb {input} {input.db}
mmseqs2 taxonomy {input.db} {db} {db.lca} /tmpdir
mmseqs2 createtsv {input.db} {db.lca} {output.tsv}
```

mOTUs2 v2.0.1 is a marker-based method that compiles a large variety of marker genes from multiple biomes: four body sites and the ocean biome. (Milanese et al., 2019) This set of marker genes contains 3x and 7x the known and unknown species from mOTUs version 1. Query reads are aligned using bwa mem and further processed to generate an abundance profile.

mOTUs2 command:

```
motus profile -pu {input} -o {output.report} -t {threads}
```

PathSeq (GATK v4.1.2) is a pipeline that has steps for host (human) read depletion and microbial genome read alignment using bwa (Walker et al., 2018). The output of PathSeq reports the “score” of each taxa according to the amount of read evidence present for each taxa. This score is normalized by genome length, unlike most other classifiers, and PathSeq does not readily provide an unnormalized score. The taxa abundance was taken to be the score divided by the total number of reads. The default host and microbe databases were downloaded from ftp://ftp.broadinstitute.org/bundle/pathseq/ corresponding to the 2017/12/19 versions of these databases. Although custom images can be constructed for the microbial reference database, it was not performed for this analysis.

PathSeq command:

```
gatk PathSeqPipelineSpark --input {input} --microbe-bwa-image {db.bwa}
--microbe-fasta {db.microbe} --taxonomy-file {db.taxonomy}
--scores-output {output.report} --output {output.reads}
```

Prophyle v0.3.0.3-64f187 is another alignment based approach that propagates k-mers up the taxonomic tree such that contigs are assembled at each taxonomic node (Břinda et al., 2017). These node-specific contigs are indexed by BWT and searched using a BWA-like algorithm (Li, 2013). The default database contains RefSeq bacteria, plasmids, and viruses constructed on 2017/11/14 .

Prophyle command:

```
prophyle classify {db} {input} > {output}
```

TaxMaps v0.2.1-7e0ec7 is an FM-index based alignment approach that uses the GEM mapper (Corvelo et al., 2018; Marco-Sola et al., 2012). The reference sequences are first processed to classify the LCA of each k-mer. These k-mers are extended into segments that share the LCA, where are used to generate the FM-index used by the GEM mapper. Combining segments that share an LCA compresses the original input database. The default database was RefSeq Complete Bacteria/Archaea/Viral genomes 300 bp GEM index downloaded from ftp://ftp.nygenome.org/taxmaps and updated 2018/03/15.

TaxMaps command:

```
taxMaps -f {input} -d {db} -c {threads} -t {db.tax} -p sample -o
{output.dir}
```

## References

Ainsworth, D., Sternberg, M.J.E., Raczy, C., and Butcher, S.A. (2017). k-SLAM: accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets. Nucleic Acids Res. *45*, 1649–1656.

Breitwieser, F.P., and Salzberg, S.L. (2018). KrakenHLL: Confident and fast metagenomics classification using unique k-mer counts.

Břinda, K., Salikhov, K., Pignotti, S., and Kucherov, G. (2017). karel-brinda/prophyle: ProPhyle 0.3.1.0.

Buchfink, B., Xie, C., and Huson, D.H. (2015). Fast and sensitive protein alignment using DIAMOND. Nat. Methods *12*, 59–60.

Corvelo, A., Clarke, W.E., Robine, N., and Zody, M.C. (2018). taxMaps: Comprehensive and highly accurate taxonomic classification of short-read data in reasonable time. Genome Res.

Freitas, T.A.K., Li, P.-E., Scholz, M.B., and Chain, P.S.G. (2015). Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Res. *43*, e69.

Gire, S.K., Goba, A., Andersen, K.G., Sealfon, R.S.G., Park, D.J., Kanneh, L., Jalloh, S., Momoh, M., Fullah, M., Dudas, G., et al. (2014). Genomic surveillance elucidates Ebola virus origin and transmission during the 2014 outbreak. Science *345*, 1369–1372.

Grüning, B., Dale, R., Sjödin, A., Chapman, B.A., Rowe, J., Tomkins-Tinch, C.H., Valieris, R., Köster, J., and Bioconda Team (2018). Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat. Methods *15*, 475–476.

Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering *9*, 90–95.

Jones, E., Oliphant, T., Peterson, P., and Others (2001). SciPy: Open source scientific tools for

Python.

Kim, D., Song, L., Breitwieser, F.P., and Salzberg, S.L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. *26*, 1721–1729.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., et al. (2016). Jupyter Notebooks -- a publishing format for reproducible computational workflows. In Positioning and Power in Academic Publishing: Players, Agents and Agendas, F. Loizides, and B. Schmidt, eds. (IOS Press), pp. 87–90.

Köster, J., and Rahmann, S. (2012). Snakemake--a scalable bioinformatics workflow engine. Bioinformatics *28*, 2520–2522.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM.

Liu, X., Yu, Y., Liu, J., Elliott, C.F., Qian, C., and Liu, J. (2018). A novel data structure to support ultra-fast taxonomic classification of metagenomic sequences with k-mer signatures. Bioinformatics *34*, 171–178.

Lu, J., Breitwieser, F.P., Thielen, P., and Salzberg, S.L. (2017). Bracken: estimating species abundance in metagenomics data. PeerJ Comput. Sci. *3*, e104.

Marco-Sola, S., Sammeth, M., Guigó, R., and Ribeca, P. (2012). The GEM mapper: fast, accurate and versatile alignment by filtration. Nat. Methods *9*, 1185–1188.

McIntyre, A.B.R., Ounit, R., Afshinnekoo, E., Prill, R.J., Hénaff, E., Alexander, N., Minot, S.S., Danko, D., Foox, J., Ahsanuddin, S., et al. (2017). Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. Genome Biol. *18*, 182.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, S. van der Walt, and J. Millman, eds. pp. 51–56.

Menzel, P., Ng, K.L., and Krogh, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju. Nat. Commun. *7*, 11257.

Milanese, A., Mende, D.R., Paoli, L., Salazar, G., Ruscheweyh, H.-J., Cuenca, M., Hingamp, P., Alves, R., Costea, P.I., Coelho, L.P., et al. (2019). Microbial abundance, activity and population genomic profiling with mOTUs2. Nat. Commun. *10*, 1014.

Morgulis, A., Coulouris, G., Raytselis, Y., Madden, T.L., Agarwala, R., and Schäffer, A.A. (2008). Database indexing for production MegaBLAST searches. Bioinformatics *24*, 1757–1764.

Nasko, D.J., Koren, S., Phillippy, A.M., and Treangen, T.J. (2018). RefSeq database growth influences the accuracy of k-mer-based species identification.

Ounit, R., and Lonardi, S. (2016). Higher classification sensitivity of short metagenomic reads with CLARK-S. Bioinformatics *32*, 3823–3825.

Ounit, R., Wanamaker, S., Close, T.J., and Lonardi, S. (2015). CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. BMC

Genomics *16*, 236.

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

Reppell, M., and Novembre, J. (2017). Using pseudoalignment and base quality to accurately quantify microbial community composition.

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., et al. (2017). Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. Nat. Methods *14*, 1063–1071.

Steinegger, M., and Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat. Biotechnol. *35*, 1026–1028.

Truong, D.T., Franzosa, E.A., Tickle, T.L., Scholz, M., Weingart, G., Pasolli, E., Tett, A., Huttenhower, C., and Segata, N. (2015). MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat. Methods *12*, 902–903.

Walker, M.A., Pedamallu, C.S., Ojesina, A.I., Bullman, S., Sharpe, T., Whelan, C.W., and Meyerson, M. (2018). GATK PathSeq: a customizable computational tool for the discovery and identification of microbial sequences in libraries from eukaryotic hosts. Bioinformatics *34*, 4287–4289.

Wood, D.E., and Salzberg, S.L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. *15*, R46.

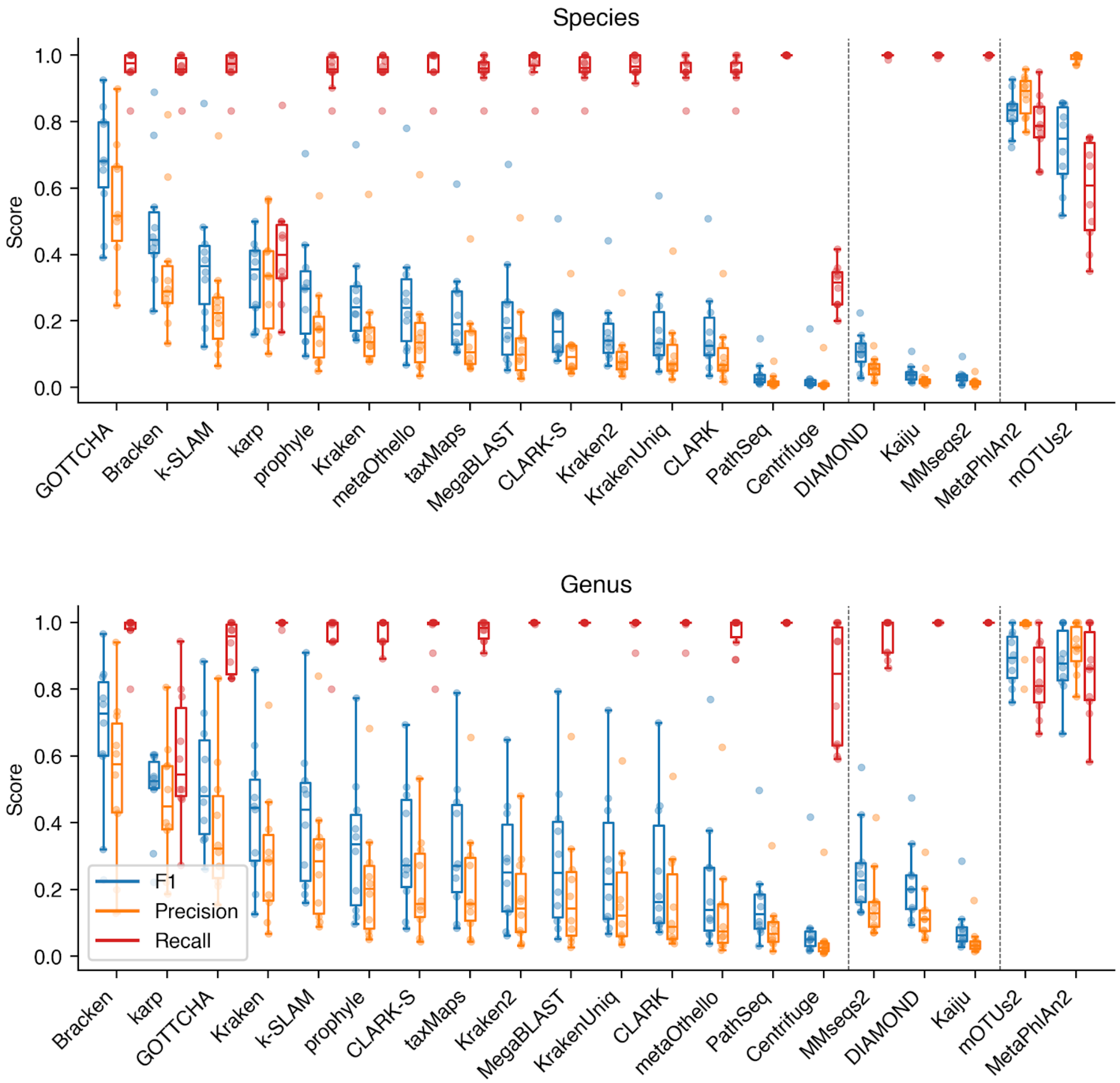


**Figure S1.** Baseline precision, recall, and F1 statistics on unfiltered abundance reports with no abundance threshold (considering all classified taxa regardless of abundance) at the species and genus levels using default databases. Related to Figure 3.

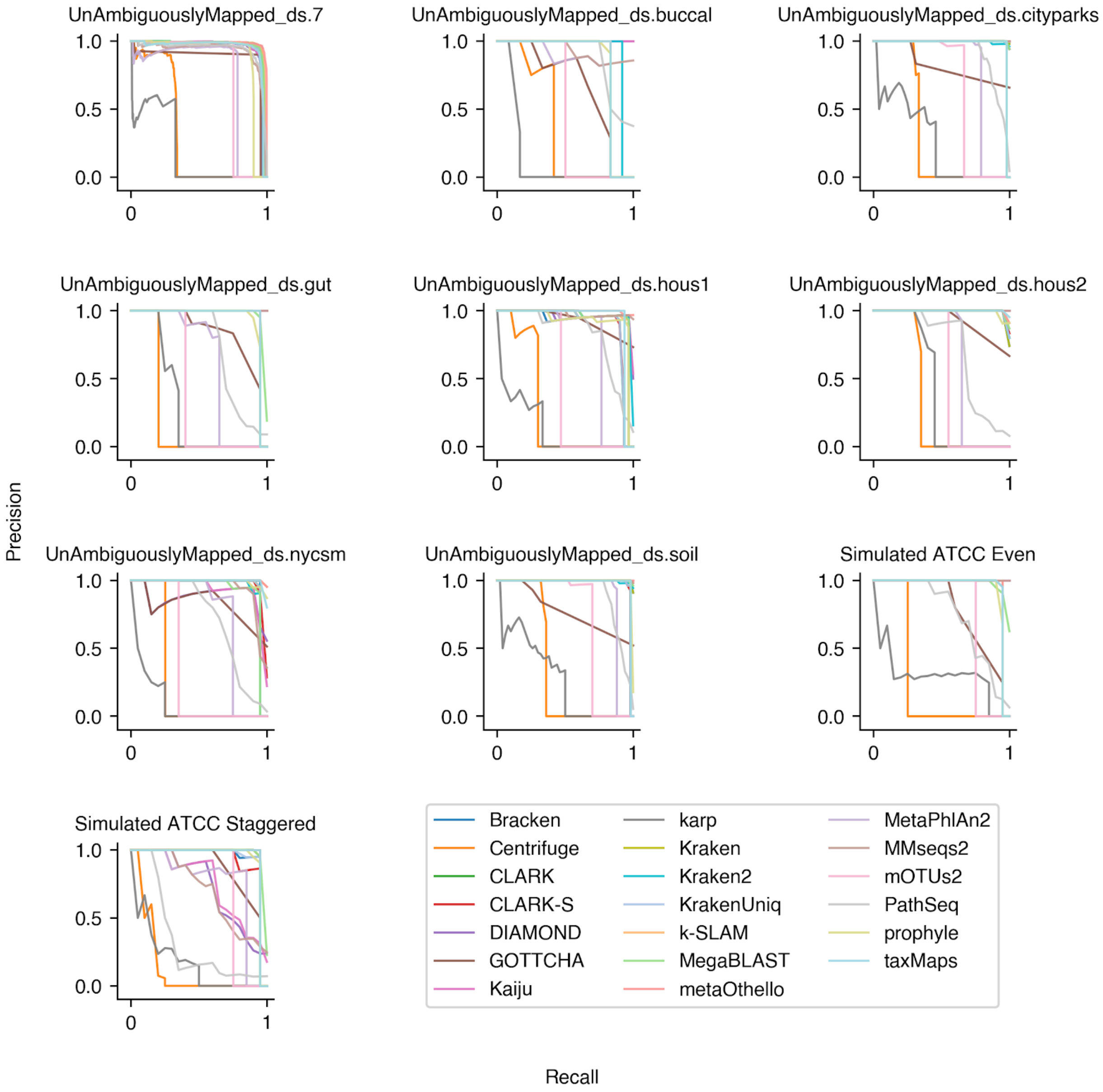


**Figure S2.** Precision-Recall curves with each sample as a separate plot for classifications using default databases. Related to Figure 3.

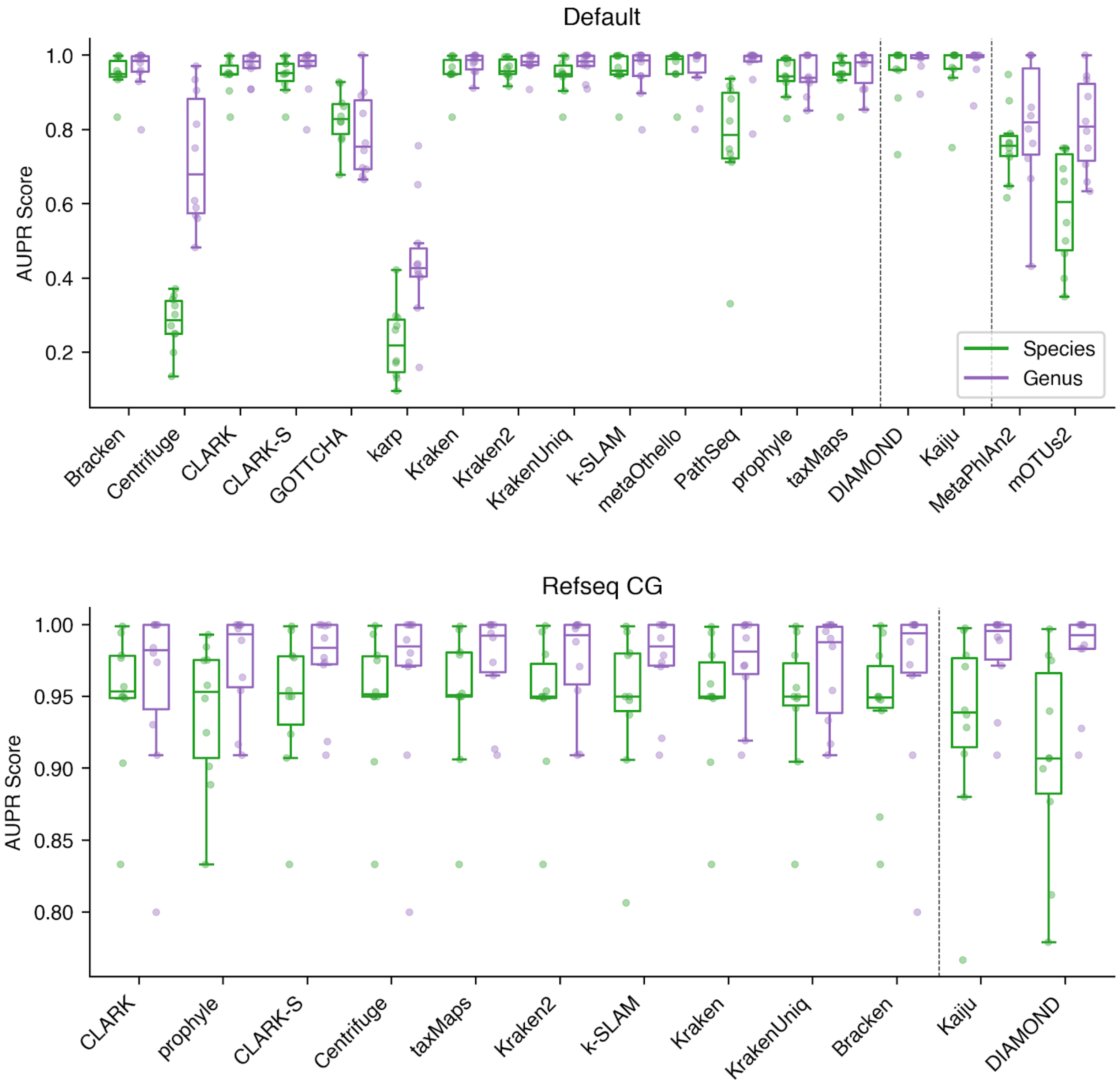


**Figure S3.** Area under precision-recall curve (AUPR) scores for each classifier for simulated datasets at the species and genus ranks using both the default and RefSeq CG databases. Related to Figure 3.

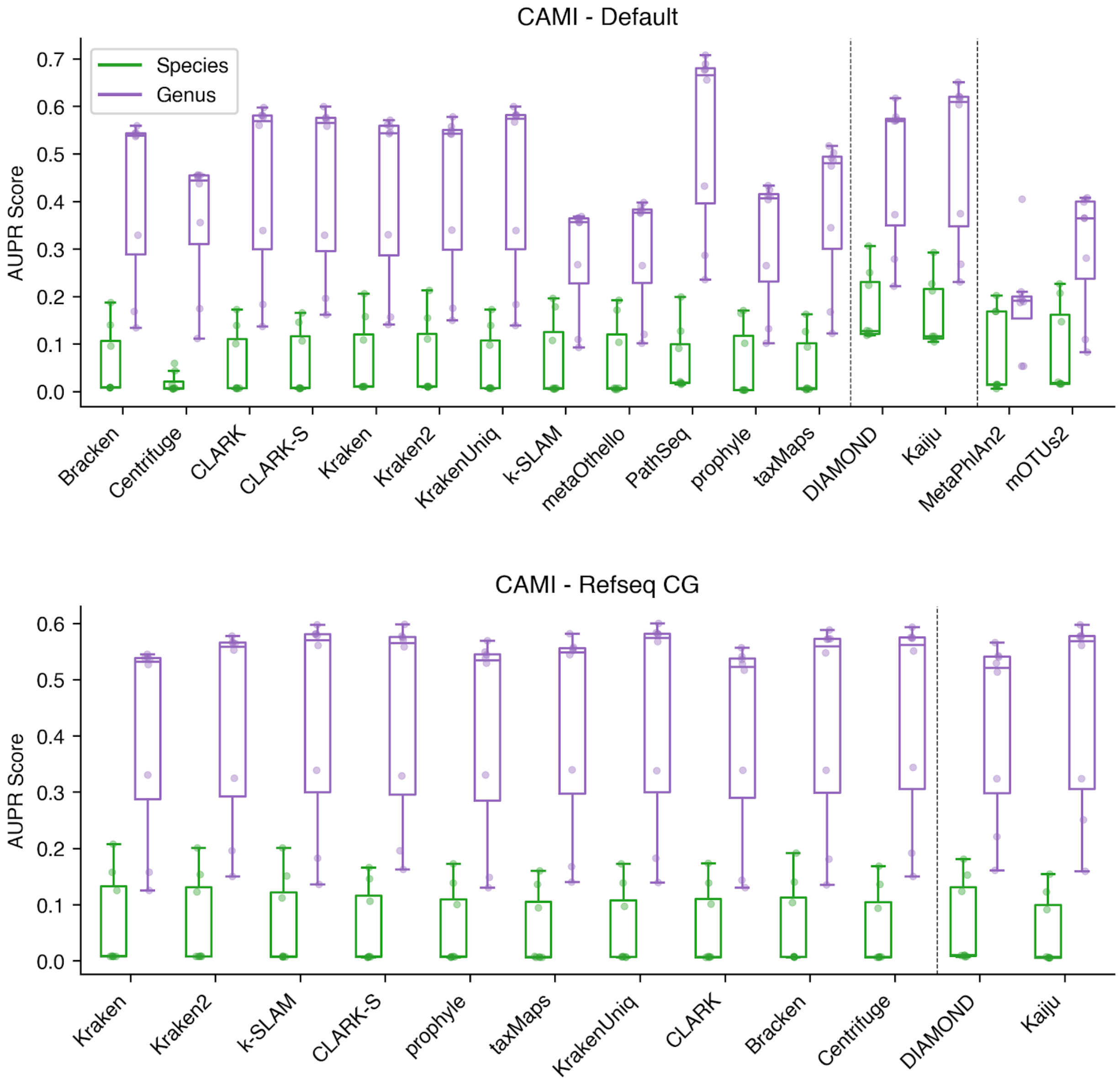
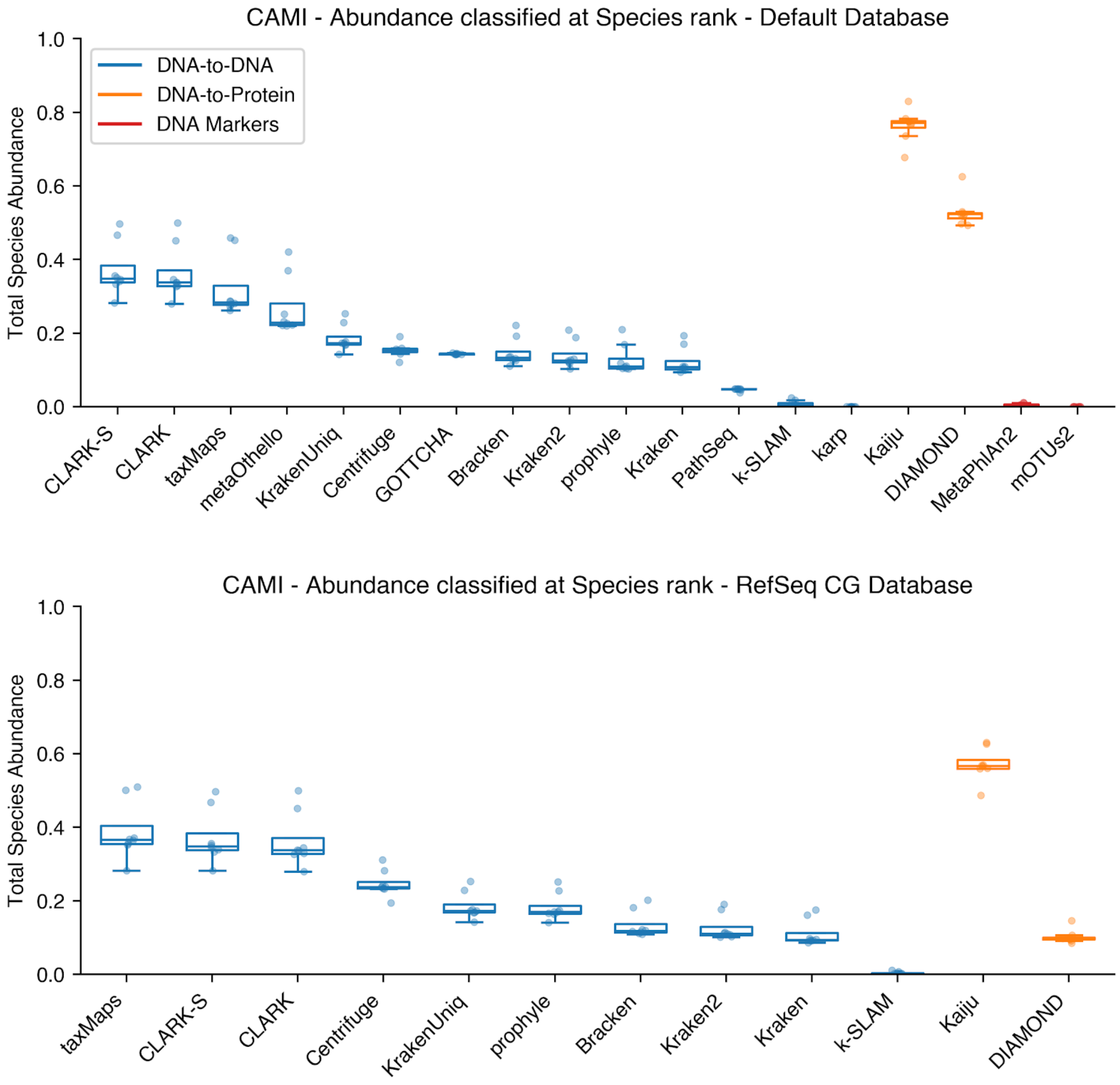


**Figure S4.** Area under precision-recall curve (AUPR) scores for each classifier for CAMI datasets at the species and genus ranks using both the default and RefSeq CG databases. MegaBLAST and MMseqs2 are omitted due to high computational runtime. Related to Figure 3.


CAMI - Abundance classified at Species rank - Default Database

CAMI - Abundance classified at Species rank - RefSeq CG Database

**Figure S5.** Proportion of abundance classified at the Species rank for CAMI dataset. Proportion of sample abundance classified at the species rank with default databases and using uniform RefSeq CG databases. Related to Figure 5.

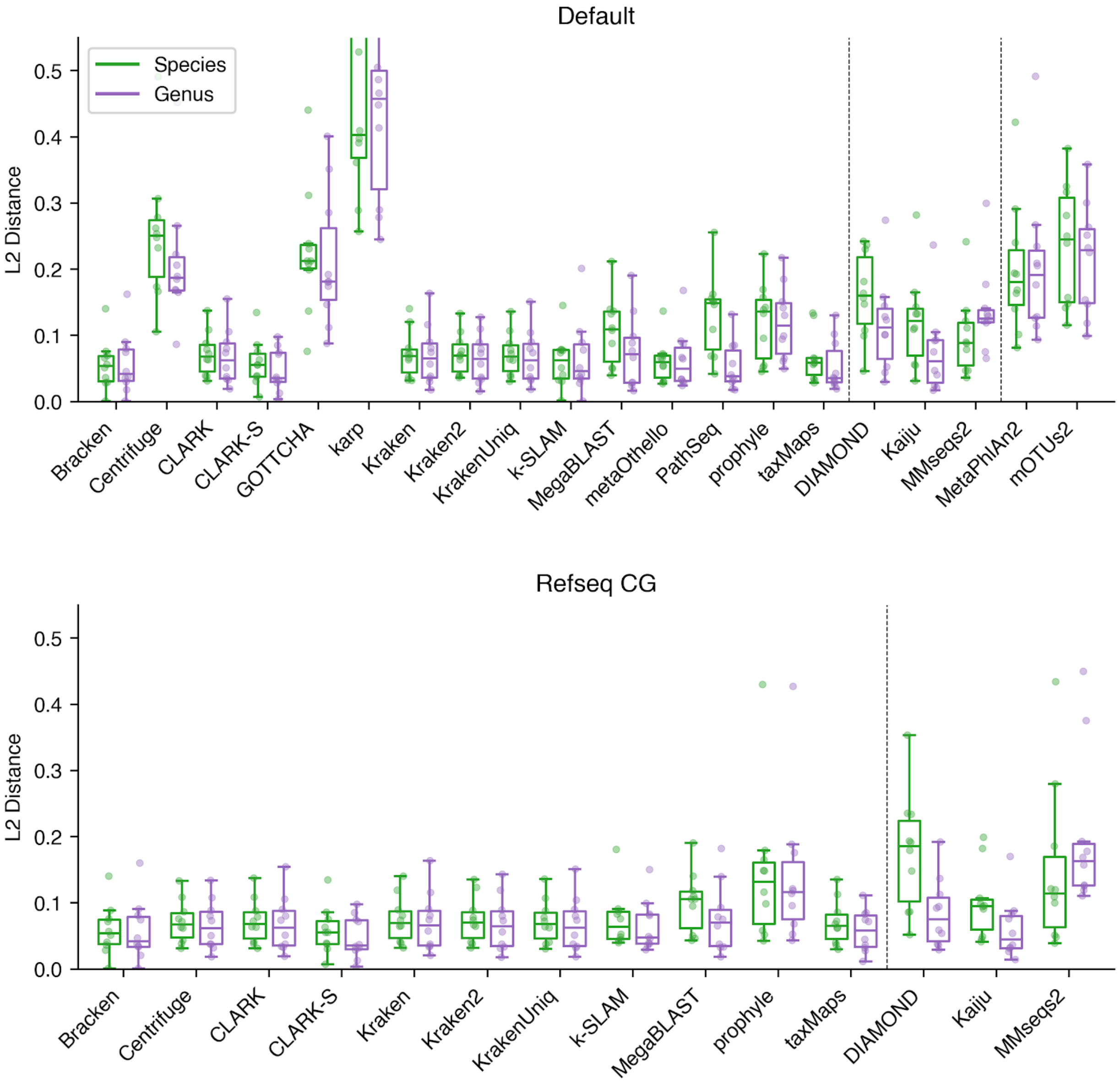


**Figure S6.** L2 abundance profile norms for each classifier for simulated datasets at the species and genus ranks using both the default and RefSeq CG databases. For L2 a lower value indicates more accurate abundance estimates. Related to Figure 4.

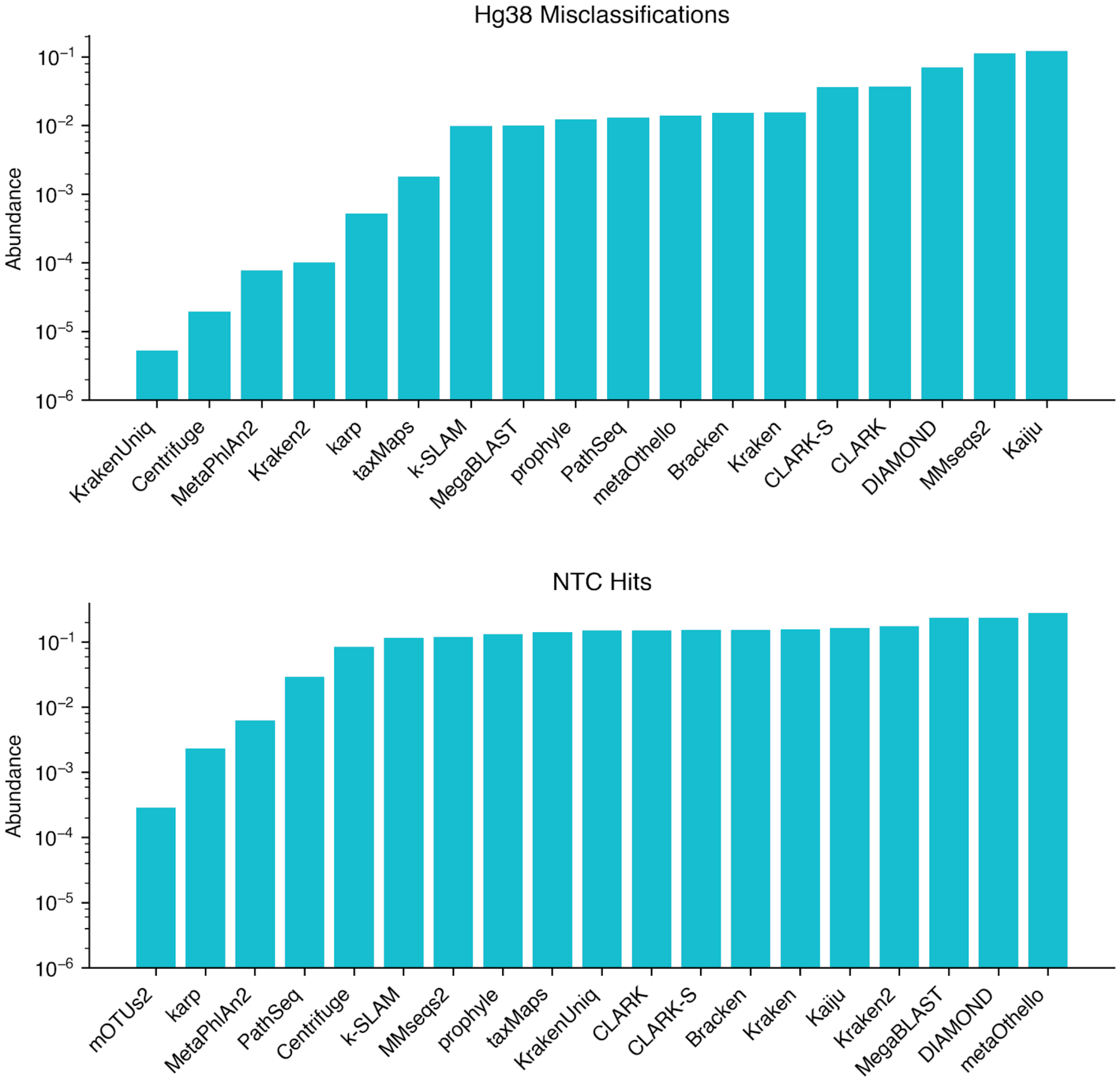


**Figure S7.** Proportion of simulated hg38 reads and ATCC Even In-Vitro NTC sample reads misclassified to non-metazoan taxa excluding the root taxon. Classifiers that classified less than one in a million reads were omitted. Related to Figure 6.

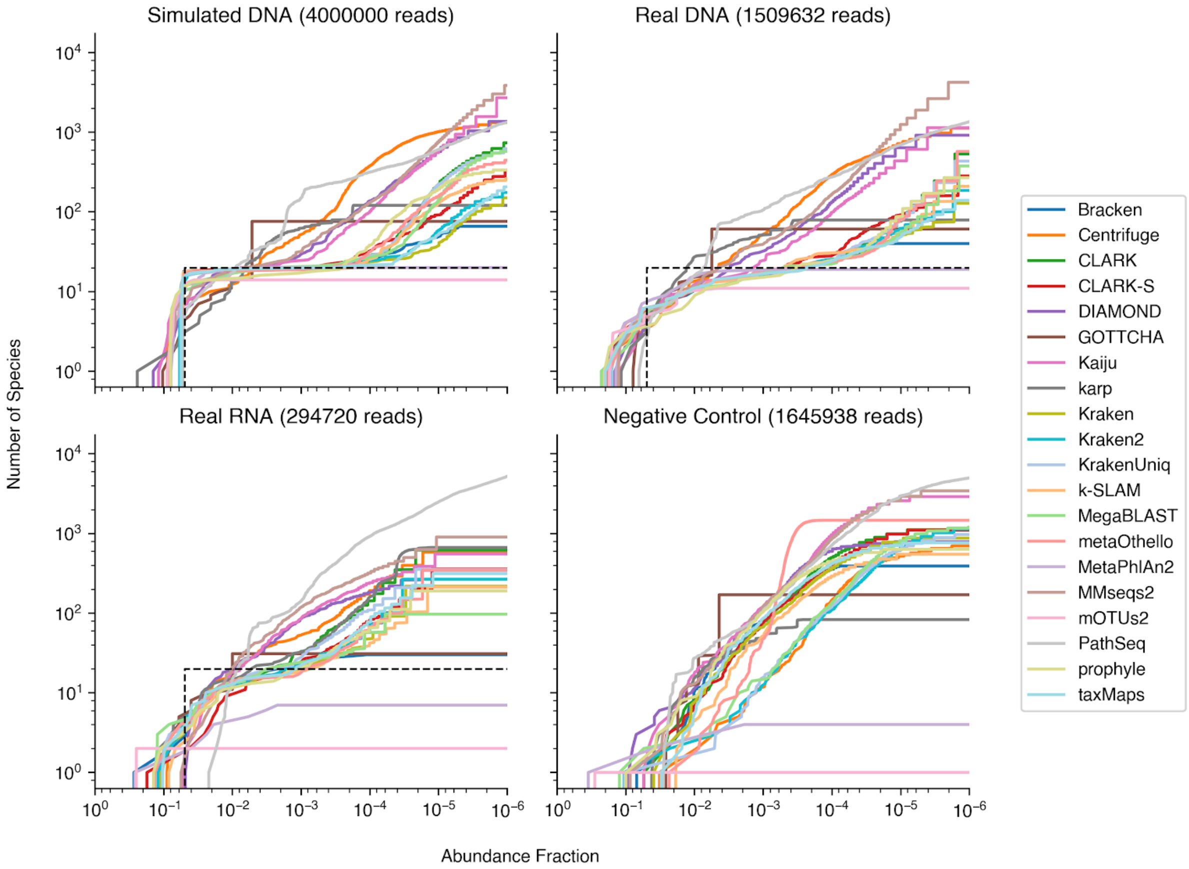
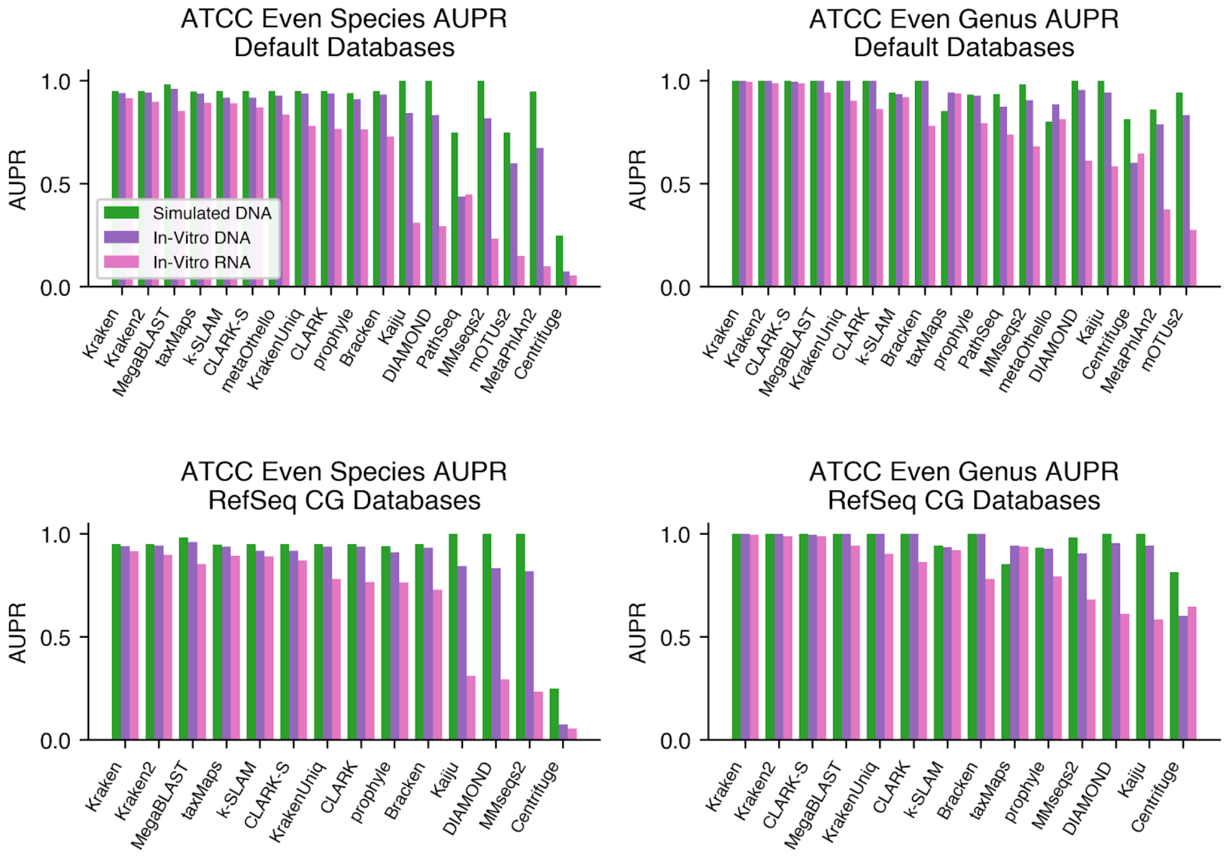


**Figure S8.** Number of species classified versus minimum abundance threshold detected in ATCC Even sample datasets. The truth abundance of 20 species at 0.05 abundance each is depicted as the black dotted line. Related to Figure 6.



ATCC Even Species AUPR Default Databases

ATCC Even Genus AUPR Default Databases

ATCC Even Species AUPR RefSeq CG Databases

ATCC Even Genus AUPR RefSeq CG Databases

**Figure S9.** AUPR for three variants of the ATCC Even sample: Simulated DNA, In-Vitro DNA, and In-Vitro RNA from whole cell material. Comparisons are made across default versus Refseq CG databases as well as the species vs genus level. The classifiers are ordered according to their average AUPR across the three sample variants. Related to Figure 6.