Title

Leviathan: A fast, memory-efficient, and scalable taxonomic and pathway profiler for next generation sequencing (pan)genome-resolved metagenomics and metatranscriptomics

Authors

Josh L. Espinoza[1,2]

Affiliations

[1]*NewAtlantis Labs*
[2]*J. Craig Venter Institute*

Contact

jol.espinoz@gmail.com

Abstract

Metagenomic and metatranscriptomic functional profiling is crucial for understanding microbial community capabilities, yet current tools often face challenges in computational efficiency, scalability, and integrated genome-resolved references. Here, I introduce *Leviathan*, an open-source software package designed to address these limitations. *Leviathan* implements taxonomic profiling via *Sylph* and a novel highly optimized functional profiling workflow. The functional profiling workflow uniquely combines the speed of *PyKOfamSearch* (*PyHMMER*-accelerated *KofamScan* implementation) for feature annotation, the accuracy of *Salmon* for read quantification against genome-resolved reference gene catalogs, and the novel graph-based pathway completeness assessment of the *KEGG Pathway Profiler*. I demonstrate *Leviathan's* capabilities using the *CAMI* low, medium, and high complexity datasets. Compared to the widely used tool *HUMAnN*, *Leviathan* exhibits significantly reduced runtimes (up to ~72-fold faster) and memory usage (up to ~14-fold lower), while achieving competitive or superior accuracy gains (up to 12%) in identifying functional features at both individual genome and pangenome levels. Notably, *Leviathan* natively supports and streamlines pangenome-level analysis, a critical aspect for understanding functional redundancy and diversity within microbial communities. *Leviathan* is available as an open-source software package (Python and CLI

APIs) offering a powerful and accessible solution for comprehensive genome-resolved metagenomic profiling.

Introduction:

The advent of high-throughput sequencing has revolutionized microbiology, enabling deep insights into the composition and function of complex microbial communities through metagenomics and metatranscriptomics. While taxonomic profiling reveals "who is there" and “how abundant they are relative to each other”, functional profiling uncovers "what they can do" providing crucial information for understanding ecosystem dynamics, host-microbe interactions, and biogeochemical cycles. Contextualizing genetic assets with sample metadata is essential for translating basic research into actionable insight with genome-resolved metagenomics as a powerful approach for building scalable science.

Taxonomic profiling can be implemented using *k*-mer based methods (e.g., *Sylph* (Shaw and Yu, 2024), *Kraken2* (Wood et al., 2019), *Ganon* (Piro et al., 2020) requiring only a reference database of genomes or marker-based methods (*MetaPhlAn* (Blanco-Míguez et al., 2023), *mOTUs* (Ruscheweyh et al., 2021) requiring both reference genomes and taxa-specific marker sets. While the methodologies for taxonomic profiling are well established and scalable, the development of functional profiling methodologies has not received the same advancements. Accurate and comprehensive functional profiling from meta-omics datasets requires: 1) establishing a reference catalog of (pan)genomes (ideally from *de novo* genomes recovered from said dataset) and genome-specific pathway markers; 2) profiling this reference database using FASTQ reads; 3) aggregating counts relative to each pathway within each (pan)genome; and 4) estimating the “completeness” of essential steps within each pathway for each (pan)genome (i.e., pathway coverage). These (pan)genome-resolved functional abundance and coverage artifacts are the main outputs used for downstream analysis in ecosystem-level modeling.

Existing tools such as *HUMAnN* (Beghini et al., 2021) have been instrumental in establishing a reproducible framework for functional profiling in metagenomics and metatranscriptomics. However, as sequencing costs decline faster than Moore’s Law predictions, the volume of meta-omic sequencing data available has grown substantially (“DNA Sequencing Costs,” n.d.) and the current resource-intensive functional profiling methodologies are not designed for the new paradigm of genome-resolved metagenomics with massive genomes catalogs (*GTDB* (Parks et al., 2022), *OceanDNA* (Nishimura and Yoshizawa, 2022), *Soil MAGs* (Ma et al., 2023)). Furthermore, with a shift towards sample-specific genome-resolved metagenomics (Espinoza et al., 2024; Espinoza and Dupont, 2022; Shaiber and Eren, 2019) and Average Nucleotide Identity (ANI) based pangenome clustering methods (Espinoza et al., 2024; Olm et al., 2017; Shaw and Yu, 2023), the need for

scalable profiling methods that can natively handle strain-level and pangenome-level resolution are essential for leveraging the power of rapid profiling techniques. Pangenome analysis, which considers the entire gene repertoire of closely related genomes (e.g., strains clustered at a specific ANI and alignment fraction), is increasingly recognized as essential for capturing the full functional landscape and understanding functional redundancy or specialization within microbial populations (Molina-Pardines et al., 2025).

To address these challenges, I developed *Leviathan,* a rapid, modular, and scalable software package for integrated taxonomic and functional profiling of metagenomes and metatranscriptomes designed specifically for large *de novo* genome-resolved (pan)genomic databases. *Leviathan* streamlines the workflows for 1) building taxonomic and functional profiling databases; 2) profiling taxonomic/sequence abundance; 3) profiling pathway abundance/coverage; and 4) merging sample-specific outputs lazily into *Xarray NetCDF* and *Apache Parquet* artifacts that be seamlessly sliced into tabular dataframes. This work introduces the architecture and capabilities of *Leviathan* by demonstrating the significant performance improvements in accuracy, speed and memory usage when benchmarked against *HUMAnN* (Beghini et al., 2021) using the standardized *CAMI I* (*Critical Assessment of Metagenome Interpretation*) low, medium, and high complexity datasets (Sczyrba et al., 2017) and the marine *CAMI II* dataset (Meyer et al., 2022). These synthetic case studies demonstrate common paradigms in metagenomics while showcasing *Leviathan's* scalability and native support for streamlined (pan)genome-level functional profiling representing a key advancement for comprehensive modern meta-omics research. *Leviathan* is an *Apache 2.0*-licensed open-source software package available at https://github.com/jolespin/leviathan.

Methods

Data resources

CAMI datasets were downloaded from *GigaDB* (https://gigadb.org/dataset/100344).

Benchmarking *Leviathan* and *HUMAnN*

All benchmarking and analysis was performed using a virtual machine with the following specifications: *Linux Ubuntu* 22.04 64-bit (x86_64), 30 *Intel Xeon Platinum 8358* CPU, 222 GB memory, and 1 *NVIDIA A10* GPU. Benchmarking and analysis was performed using 16 threads running 2 jobs simultaneously for *Leviathan* and *HUMAnN*. The following versions were used for all benchmarking: *Sylph v0.8.1*, *Salmon v1.10.3*, *HUMAnN v3.9, Leviathan v2025.7.3, BBMap v39.28, VEBA v2.5.1.* Genome and pangenome level accuracies were determined by comparing the *mapped.sam* from *Salmon* for *Leviathan* and *humann_diamond_aligned.tsv* from *Diamond* for

HUMAnN. In the case of multiple alignments in SAM files, performance was assessed using the primary alignment for determining the predicted source genome and pangenome.

Universal CAMI preprocessing

The following commands were run for low, medium, and high complexity datasets separately:

Clustering genomes into pangenomes using *skani* and *VEBA*

```
skani triangle --sparse -t 16 -o ${out_dir}/skani_output.tsv -l
${genome_filepaths} --min-af 15 -s 80 -c 125 -m 1000 --ci

cat ${output_directory}/skani_output.tsv | cut -f1-5 | tail -n +2 |
edgelist_to_clusters.py --basename -t 95 -a 50 --af_mode relaxed
--cluster_prefix "PSLC-" -o ${output_directory}/genome_clusters.tsv
--identifiers ${genome_identifiers} --export_graph
${output_directory}/networkx_graph.pkl --export_dict ${out_dir}/dict.pkl
--export_representatives ${output_directory}/representatives.tsv
```

Predicting prokaryotic genes with *Pyrodigal*

```
pyrodigal -i ${fp} -a ${dir}/${id}.faa -d ${dir}/${id}.ffn -j 16 |
append_geneid_to_prodigal_gff.py > ${dir}/${id}.gff
```

Quality assessing genomes with *CheckM2*

```
checkm2 predict -i ${genome_directory} -o ${output_directory} --threads 16
--genes -x faa --database_path ${checkm2_database}
```

Leviathan preprocessing and benchmarking *CAMI*

Identifying *KEGG Ortholog* pathway markers:

```
pykofamsearch -i ${faa} -o ${out} -p=16 -b ${db}
```

Preprocessing genomes and annotations for *Leviathan*:

```
leviathan-preprocess.py -i ${manifest} -a ${annotations} -o
${output_directory} --annotation_format pykofamsearch
```

Building the *Leviathan* index:

```
leviathan-index.py -f ${fasta} -m ${feature_mapping} -g ${genomes} -d
${index_directory} -p=16 --pathway_database ${pathway_database}
```

Taxonomic profiling with *Leviathan*:

```
leviathan-profile-taxonomy.py -1 ${r1} -2 ${r2} -n ${id} -d ${index} -p=16
-o ${output_directory}
```

Functional profiling with *Leviathan*:

```
leviathan-profile-pathway.py -1 ${r1} -2 ${r2} -n ${id} -d ${index} -p=16
-o ${output_directory} --salmon_include_mappings --alignment_format sam
```

HUMAnN preprocessing and benchmarking *CAMI*

Diamond alignment against *UniRef50*:

```
diamond blastp --db ${db} --query ${faa} --threads 4 -f 6 qseqid sseqid
stitle pident evalue bitscore qcovhsp scovhsp --max-target-seqs 1 --header
simple -o ${out}
```

UniRef50 is part of *UniRef Clusters* (Suzek et al., 2015).

Compiling annotations for custom *HUMAnN* database:

```
compile_custom_humann_database_from_annotations.py -a ${annot} -s ${faa}
-o ${out} -t ${tax} -i ${ids} -m header
```

Building custom *HUMAnN*-compatible *Diamond* database:

```
diamond makedb --threads 16 --in
${output_directory}/diamond_hits.proteins.fasta --db
${output_directory}/diamond_hits.proteins${db_suffix}
```

Joining forward and reverse reads for *HUMAnN* profiling:

```
repair.sh in1=${r1} in2=${r2} out=stdout.fastq | bbmerge.sh in=stdin.fastq
int=t out=${joined_fastq} minoverlap=12 itn
```

Functional profiling with *HUMAnN*:

```
humann -i ${fastq} -o ${output_directory} --threads 16  -v --search-mode
${db_name} --memory-use minimum --input-format "fastq.gz"
--bypass-nucleotide-search --diamond ${DIAMOND_PATH} --evalue 1.0
--protein-database ${db_directory} --output-basename humann --id-mapping
${id_mapping} --o-log ${output_directory}/humann.log
```

repair.sh and bbmerge.sh are provided in the *BBMap* software suite (Bushnell, 2014)

Building the *Leviathan* reference index

The construction of a comprehensive reference index is a prerequisite for analysis with the *Leviathan* framework and is performed using the `leviathan-index.py` command-line executable. This executable requires a set of coding DNA sequences (CDS) in a FASTA file (`--fasta`), and a corresponding tab-separated feature mapping file (`--feature_mapping`) that links each gene ID to its functional annotations (as a Python-style set or list string) and its source genome ID. The executable first validates data integrity by ensuring a perfect one-to-one correspondence between gene identifiers in the FASTA and feature mapping files before compiling the metadata into efficient, pickled Python dictionaries. For gene abundance/expression quantification, it then builds a CDS-level index using *Salmon* (Patro et al., 2017), employing the `--keepDuplicates` flag to retain all provided sequences. Concurrently, if a mapping file of genome IDs to their assembly file paths is provided (`--genomes`), the executable constructs a taxonomic profiling index by generating a single sketch of all genome assemblies using *Sylph* (Shaw and Yu, 2024), with default parameters of a 31-bp k-mer, a minimum spacing of 30, and a subsampling rate of 200. If pangenome IDs are also provided, then the pangenome information is incorporated into the index and is used for downstream profiling. Pathway-level context is integrated by either using a pre-formatted custom database or by automatically downloading and processing the *KEGG* database if the provided functional features are identified as *KEGG Orthologs* (Aramaki et al., 2020). The executable ensures relevance by verifying that features in the pathway database overlap with those in the user's dataset. The final output is a structured directory containing the *Salmon* index, the *Sylph* sketch, the compiled databases, configuration and log files, and a summary of MD5 hashes for all components. An existing index can be updated *post hoc* to include genome sketches by re-running the executable with the `--update_with_genomes` flag. A typical command to build a new index is:

```
leviathan-index.py --fasta pathway_markers.fasta[.gz] --feature_mapping
gene_to_ko.tsv[.gz] --genomes genome_paths.tsv[.gz] --index_directory
leviathan_index/ --n_jobs 8
```

Taxonomic and sequence abundance with *Leviathan*

The taxonomic composition of a metagenomic/metatranscriptomic sample is determined using the `leviathan-profile-taxonomy.py` command-line executable, which leverages the alignment-free *k*-mer-based profiler *Sylph* (Shaw and Yu, 2024). The workflow begins by generating a *k*-mer sketch from the input paired-end sequencing reads (FASTQ format). This is handled internally by

executing the `sylph sketch` command with default parameters of a 31-bp k-mer size (`-k 31`), a minimum spacing of 30 between selected *k*-mers, and a subsampling rate of 200 (`-c 200`); alternatively, a pre-computed reads sketch can be provided directly. The resulting reads sketch is then compared against the reference genome database (`genomes.syldb`) previously built and stored within the *Leviathan* index. The core profiling is performed by the `sylph profile` command, which estimates the relative abundance of each reference genome in the sample. To ensure high-confidence assignments, *Leviathan* uses a minimum ANI threshold of 95% (`--minimum-ani 95`), and a minimum of 50 shared *k*-mers to consider a genome present (`--min-number-kmers 50`). The executable also invokes the `--estimate-unknown` flag to account for the fraction of the community not represented in the reference database which is reported under the *"Sequence_abundance*" variable of the resulting artifacts. Following profiling, *Leviathan* parses the *Sylph* output, converting genome file paths back to the user-defined genome identifiers from the index. This module generates two primary metrics: *"Taxonomic_abundance"*, representing the relative proportion of each genome in the community, and *"Sequence_abundance"*, the fraction of total reads derived from each genome. If genome clusters (i.e., pangenomes) were defined during index creation, abundances are automatically aggregated to this higher level. The final abundance tables are saved for each sample in either compressed TSV or *Apache Parquet* format at both the genome and genome-cluster levels. A typical command for taxonomic profiling is as follows:

```
leviathan-profile-taxonomy.py -1 forward.fastq.gz -2 reverse.fastq.gz -n
Sample_1 -d path/to/leviathan_index/ -o output_directory/
```

(Pan)genome-resolved graph-based pathway completion and abundance with *Leviathan*

The functional potential and transcriptional activity of microbial communities is profiled using the `leviathan-profile-pathway.py` executable, which integrates abundance/expression quantification with the genomic and pathway context stored in the *Leviathan* reference index. The workflow initiates by quantifying gene expression from paired-end sequencing reads using *Salmon* (Patro et al., 2017) in its `--meta` mode, which is specifically optimized for accurately assigning reads in complex metagenomic samples with strain-level heterogeneity similar to metazoan single-cell isoforms. To maintain high-confidence mappings, a stringent alignment score threshold is used by default (`--minScoreFraction 0.87`), ensuring that only reads with high-quality alignments contribute to abundance estimates. The output of this step is a per-gene quantification table providing both Transcripts Per Million (TPM) and raw read counts.

Following quantification, these gene-level abundances are contextualized by mapping each gene to its source genome and its annotated functional features (e.g., *KEGG Orthologs*) using the metadata from the index. The abundance/expression value of a gene annotated with multiple features is, by default, distributed equally among them, providing a scaled abundance for each feature but this can be turned off using the `--no_split_feature_abundances` option. A core feature of this module is its dual analysis of both functional potential or transcriptional activity for each pathway, resolved at the (pan)genome level.

First, to assess genomic potential, *Leviathan* calculates pathway completion. This is achieved through a graph-based approach via *KEGG Pathway Profiler* where each metabolic pathway is treated as a network of functional nodes (e.g., *KEGG Orthologs*). For each genome or pangenome cluster, the complete set of detected functional features are mapped onto these pathway graphs and a coverage score is computed representing the fraction of a pathway's enzymatic steps encoded within that specific genomic context. Second, to measure the functional potential or transcriptional activity, the executable aggregates the expression levels of all features constituting a given pathway within the same genomic unit. This yields a total pathway abundance value (in TPM or read counts) that reflects the collective abundance/expression of that pathway by a specific genome or pangenome cluster.

This entire analysis is performed first at the individual genome level. If the index was constructed with genome cluster definitions (i.e., pangenomes), the executable performs a subsequent aggregation, summing abundances and recalculating prevalence metrics for each cluster. The final outputs are a series of structured tables (in *Parquet* or compressed TSV format) detailing gene, feature, and pathway metrics, including the dual measures of pathway abundance (activity) and coverage (potential), resolved at both the genome and, if applicable, the pangenome level. A typical command for the pathway profiling is as follows: `leviathan-profile-pathway.py -1 forward.fq.gz -2 reverse.fq.gz -n Sample_1 -d path/to/leviathan_index/ -o output_directory/`

Merging sample-specific profiling into unified *Xarray NetCDF*

To facilitate study-level comparative analyses and downstream statistical modeling, *Leviathan* provides a dedicated `leviathan-merge.py` executable that consolidates the per-sample outputs from the profiling workflows into unified, analysis-ready data structures. This executable takes as input the parent directories containing the individual sample outputs from both the taxonomic and pathway profiling modules. It systematically discovers all per-sample result files (e.g., `.parquet` tables) and parses them to build comprehensive, multi-sample *N*-dimensional data structures that can be accessed via lazy loading without loading the entire file into memory.

The core of this process relies on the *Xarray* library (Hoyer and Hamman, 2017), which is used to construct labeled, multi-dimensional arrays that preserve the relationships between samples, genomic units (genomes or pangenomes), and features (e.g., pathways, *KEGG Orthologs*). This approach intelligently groups related metrics into single, coherent data objects. For instance, when merging pathway profiling results, the executable generates a single `pathways.genomes.nc` file. This artifact contains an `xarray.Dataset` where distinct data variables for `number_of_reads`, `tpm`, and `coverage` share common dimensions for `samples`, `genomes`, and `pathways`. This structure is vastly more efficient for storage and analysis than managing separate flat files for each metric which is required for the scalability and performance needed for modern datasets.

A similar aggregation is performed for all other data types, including taxonomic abundances, feature abundances, and feature prevalence, at both the genome and genome-cluster levels. The final outputs are persisted as compressed NetCDF (`.nc`) files, a widely-used and highly scalable format. This data model simplifies complex data wrangling, prevents data misalignment, and provides a memory-efficient foundation for streamlined access to the entire project's results, enabling robust, integrated multi-omic analyses.

Pathway marker detection and completeness estimation

The core innovation of *Leviathan* lies in its functional profiling engine, which synergistically combines: (i) *PyKOfamSearch* (https://github.com/jolespin/pykofamsearch) or *PyHMMSearch* (https://github.com/jolespin/pyhmmsearch), optimized implementations of *KofamScan* (Aramaki et al., 2020) and *HMMSEARCH* (Finn et al., 2011) utilizing *PyHMMER* (Larralde and Zeller, 2023) for rapid marker annotation; (ii) *Salmon* (Patro et al., 2017) for highly accurate and efficient pseudo-alignment of reads to gene catalogs for quantification; and (iii) the *KEGG Pathway Profiler* (https://github.com/jolespin/kegg_pathway_profiler) for robust, reaction graph-based assessment of KEGG pathway coverage and abundance. *KEGG Pathway Profiler* is an open-source reimplementation of *MGNify's kegg-pathways-completeness-tool* (https://github.com/EBI-Metagenomics/kegg-pathways-completeness-tool) optimized for high-memory systems but could not have been developed without the fundamental work pioneered by EBI's developers. *PyKOfamSearch*, *PyHMMSearch*, and *KEGG Pathway Profiler* have been developed to optimize annotation and are available within the *VEBA* software ecosystem (Espinoza et al., 2024).

Results

Overview of *Leviathan*

Leviathan is an open-source, command-line-driven software package designed for rapid, scalable, and integrated taxonomic and functional profiling of metagenomic and metatranscriptomic data. The entire workflow is engineered for high performance and is centered around a modular, three-stage architecture that streamlines analysis from FASTQ reads to multi-sample, analysis-ready data artifacts (Figure 1). *Leviathan* is run at the sample level, optimized for distributed compute environments, and the individual tabular artifacts are merged into *Xarray NetCDF* files for lazy loading and efficient access to large structured database tables.

The first stage is database construction via `leviathan-index`. *Leviathan* builds a single, unified reference index containing a *Sylph* sketch for *k*-mer-based taxonomic profiling, a *Salmon* index of coding sequences for high resolution functional quantification, and pre-compiled databases that map genes to functional features (e.g., *KEGG Orthologs*) and pathways. Optionally, genome clusters (i.e., pangenomes) can be provided during the database construction which will be used automatically in profiling stages. This step efficiently prepares all necessary assets for downstream analysis.

The second stage is profiling. Taxonomic profiling via `leviathan-profile-taxonomy` is essentially a convenient wrapper around *Sylph* to rapidly estimate the relative abundance of reference genomes in a sample. The core innovation of *Leviathan* lies in its functional profiling module via `leviathan-profile-pathway`, which first quantifies gene-level abundance/expression using *Salmon's* meta-mode for high accuracy in complex communities. It then uniquely computes two distinct metrics for each pathway at the (pan)genome level: 1) pathway coverage, a graph-based assessment of the completeness of a pathway's enzymatic steps, representing its genomic potential; and 2) pathway abundance, the aggregated abundance/expression of all genes within that pathway, representing its functional potential/activity.

The final stage is merging the artifacts into individual data structures via `leviathan-merge`. *Leviathan* consolidates the individual per-sample outputs into individual multi-dimensional *Xarray NetCDF* files. This creates a memory-efficient, analysis-ready data structure that contains all metrics (e.g., taxonomic abundance, pathway coverage, pathway abundance) neatly organized by sample, feature, and genomic unit. A key design principle throughout this workflow is *Leviathan's* native support for pangenome-resolved analysis, where all metrics can be seamlessly aggregated from individual genomes to user-defined pangenome clusters. Comparisons between *Leviathan* and *HUMAnN* capabilities are detailed in Table 1.

Benchmarking Strategy

The *CAMI* datasets were used to benchmark the pathway profiling tools (Meyer et al., 2022; Sczyrba et al., 2017). *CAMI I* and *II* datasets include 8875 genomes that cluster into 3129 pangenomes (Table 2). Of the 8875 genomes, 856 genomes were insufficient for *Pyrodigal* to identify any protein-coding genes, 3129 genomes passed quality control (*CheckM2* completeness ≥ 50% and contamination < 10%), and 7202 genomes contained high-quality KEGG pathway *KOfam* annotations. *Leviathan* can only incorporate genomes that have pathway annotations (i.e., *KOfam*) therefore any genome that did not have *KOfam* annotations that exceeded the HMM threshold were excluded from the analysis. However, every genome that lacked *KOfam* annotations also failed medium stringency quality control based on *CheckM2* so they would be excluded from best-practice workflows. Within a single dataset there are multiple samples and each sample may or may not contain a separate strain of an organism. In the case of this study, each sample-specific strain is considered a genome and all genomes from a dataset (e.g., *CAMI_high_toy*) are used to build pangenomes. These genome and pangenome collections are used for building taxonomic and functional profiles. Co-assembly or pooled samples from the *CAMI* dataset were excluded to provide highly informative benchmarks with strain-level resolution.

Native support for genome and pangenome-level profiling of microbial communities

Next-generation sequencing instruments produce data that is inherently compositional as they estimate the relative abundance of discrete biological components (e.g., transcripts, marker genes) within a community by sampling from a pool of nucleic acid fragments (Espinoza et al., 2020). Biological features are naturally hierarchical (e.g., genes → chromosomes/plasmids → genomes → pangenomes) and individual components can be amalgamated up the hierarchy while retaining compositionality (Quinn and Erb, 2020). In the case of *Leviathan*, the relative abundance and counts tables produced by *Sylph* and *Salmon* in the backend, respectively, are compositional and, thus, individual component abundances can be aggregated with respect to higher order biological structures such as pangenomes and pathways. In the context of this study, a strain will be referred to as a genome and this genome is recovered from an individual sample. Each output artifact contains a genome and pangenome version allowing users to easily select which level would be appropriate for their analysis. One caveat for estimating the abundance of a feature or pathway is that a single gene can be assigned to multiple features (e.g., *KEGG Orthologs* or enzymes) and a feature can be represented in multiple pathways. To address this, *Leviathan* implements a fractional abundance for these scenarios which can be toggled off. An example of this would be *Salmon* mapping a read to *gene_a* which is annotated by 3 features {*KOfam_1, KOfam_2, KOfam_3*}. When

aggregating the gene-level counts with respect to each feature, the read count would be equally distributed across the features where each feature would receive ⅓ of a count for each read mapped to *gene_a* (or ⅓ of a TPM value if using TPM as a metric). This approach ensures that the sum of feature counts equals the sum of gene counts which is necessary for interpreting relative abundances for compositional data analysis.

While *HUMAnN* technically supports pangenome analysis, the methodology for building custom databases is not user-friendly and requires users to identify marker genes that are specific to each pangenome to build *ChocoPhlAn* and *MetaPhlAn* databases. We were unable to build these extended databases but our performance benchmarks are computed at the genome-level and accuracy calculations are performed post hoc so are not affected by the lack of *HUMAnN*-compatible pangenome databases.

Leviathan integrates *Sylph* for rapid (pan)genome-resolved taxonomic profiling

Leviathan provides users with `leviathan-profile-taxonomy`, a convenient wrapper around *Sylph* which is a state-of-the-art taxonomic profiler (Shaw and Yu, 2024). *Sylph* is a species-level metagenome profiler that estimates genome-to-metagenome containment ANI through zero-inflated Poisson *k*-mer statistics and has superior performance relative to other approaches with support for genomes at low coverage levels (See Shaw and Yu, 2024 for benchmarking details). After running *Sylph*, *Leviathan* sums the taxonomic and sequence abundance values with respect to pangenomes and outputs relative abundance tables for genomes and pangenomes. Once the taxonomic profiling has been completed for each sample, the abundance tables are merged to build *Xarray NetCDF* artifacts where users can access data using lazy loading without storing everything into memory.

Leviathan enables rapid, low-resource functional profiling

The main innovation of *Leviathan* is in the functional and pathway profiling. To quantitatively assess the performance of this approach, a direct benchmark was conducted comparing the `leviathan-profile-pathway` module against the widely-used *HUMAnN* method. The evaluation was performed on *CAMI I* and *CAMI II* datasets to measure performance across a range of microbial community structures. Both *Leviathan* and *HUMAnN* were executed with 16 CPU threads to ensure a fair comparison, and performance was measured by recording total wall-clock time for speed and maximum resident set size for memory usage (Table 3).

The results revealed that *Leviathan* offers a substantial improvement in computational efficiency. Across all tested datasets, *Leviathan* was up to 74-fold faster (7.3x - 74.2x) than *HUMAnN*. For example, on the high-complexity *CAMI* samples, *Leviathan* completed its functional and pathway analysis in approximately 14 minutes, whereas *HUMAnN* required between 13-18 hours for the same task. This performance advantage was coupled with a significant reduction in memory consumption. *Leviathan's* peak memory usage was consistently around 14-fold lower (6.5x - 13.9x), consuming approximately 2.2 GB of RAM on the high-complexity samples compared to over 32 GB utilized by *HUMAnN*. These performance gains are attributed to *Leviathan's* architecture, which leverages a pre-compiled *Salmon* index for highly efficient, alignment-free quantification and employs optimized, in-memory aggregation strategies to rapidly compute genome-resolved pathway metrics without the overhead of intermediate file parsing and alignment-based methods.

The major performance gains attributed to the pseudo-alignment approach *Leviathan* takes relative to the translated blast search that *HUMAnN* uses in the backend. More specifically, *Leviathan* employs *Salmon* to map paired-end reads directly to the CDS sequences in DNA-space while *HUMAnN* uses *Diamond* to translate reads from DNA-space into protein-space and aligns those predicted proteins to a marker protein database. Further, since *HUMAnN* leverages translated blast searches it requires users to join reads together (e.g., bbmerge.sh (Bushnell, 2014)) prior to running the pipeline as it cannot operate on paired-end reads directly. Bypassing the translated search and leveraging state-of-the-art pseudo-alignment algorithms gives *Leviathan* greater performance and accuracy.

To leverage *HUMAnN* for custom databases, users are required to align proteins to *UniRef* (Suzek et al., 2015) and these proteins IDs are cross-referenced to enzymes in *MetaCyc* (Caspi et al., 2020) and *MinPath* (Ye and Doak, 2009). However, not all of the *UniRef* proteins have representatives in *MetaCyc* and *MinPath* pathways. Considering level-4 enzymes (e.g., EC-1.1.1.1), there are 1952 enzymes shared between *HUMAnN* and *MinPath*, 236 enzymes unique to *MinPath,* and 3571 unique to *HUMAnN*. For level-3 enzymes (e.g., EC-1.1.1.-), there are 186 shared enzymes, 73 unique to *HUMAnN*, and 9 unique to *MinPath*. These set overlaps are based on cross-referencing *HUMAnN (metacyc_reactions_level4ec_only.uniref.bz2* and *metacyc_pathways)* and *MinPath* (*MetaCyc-mapping.txt)* backend files. These findings show that many of the proteins in the *UniRef*-based custom database are not being used and are likely underestimating the true pathway coverage.

Leviathan achieves superior accuracy with strain-level resolution

The ability to accurately profile microbial communities with high strain heterogeneity is a major challenge in complex systems. *Leviathan* addresses this challenge by achieving competitive or superior accuracy on *CAMI* datasets while also maintaining optimal runtime performance. Using the same *CAMI I* toy (low, medium, and high complexity) and *CAMI II* Marine datasets, I compared the genome and pangenome assignments from the pathway abundance profiles generated by *Leviathan* and *HUMAnN* against the ground-truth profiles (Table 4).

The results revealed that *Leviathan* provides improvements in genome-level accuracy (i.e., strain resolution) while achieving near-perfect performance at the pangenome level. Across all tested datasets, *Leviathan* was up to 12% more accurate (1.5% - 12.1%) at the genome level and up to 4.9% (1.5% - 4.9%) at the pangenome-level compared to *HUMAnN*. *Leviathan's* accuracy ranged from 92.3% accuracy (*CAMI II Marine sample_7*) and 99.9% accuracy (*CAMI_low_toy*) at the genome level and 98.2% accuracy (*CAMI II Marine sample_2*) and 99.9% accuracy (*CAMI_low_toy*). For the high-complexity CAMI samples, *Leviathan* consistently performed at over 94% accuracy (94.2% - 95.6%) at the genome-level and over 98% accuracy (98.7% - 99.7%) at the pangenome level. This shows that most misalignment discrepancies that align from strain heterogeneity are resolved when aggregating genome-level features to the pangenome level even in high complexity samples. In comparison, *HUMAnN* performed at ~90% accuracy (89% - 90.5%) at the genome level and ~96% accuracy (95.7% - 96.1%) at the pangenome level.

Again, the major accuracy gains are attributed to the pseudo-alignment approach *Leviathan* takes relative to the translated blast search that *HUMAnN* uses in the backend. More specifically, the use of *Salmon* and operating in DNA-space for pathway features rather than best-hit alignments in protein-space. For higher strain-level accuracy, users can specify more strict `--minimum_score_fraction` but *Leviathan* uses 0.87 which is the default for *Alevin* single-cell analysis and designed for differentiating cell-type heterogeneity (Srivastava et al., 2019).

Discussion

This work introduced *Leviathan*, a high-performance software package designed to address the critical computational bottlenecks in modern, genome-resolved metagenomic and metatranscriptomic functional profiling. *Leviathan* is designed for both taxonomic and functional profiling by leveraging the strengths of state-of-the-art methods such as *Sylph* (Shaw and Yu, 2024) and *Salmon* (Patro et al., 2017) while also building on prior conceptual work pioneered by *HUMAnN* (Beghini et al., 2021). *Leviathan* dramatically improves computational efficiency of functional

profiling while simultaneously enhancing accuracy. The benchmarking results on the *CAMI* datasets demonstrate that Leviathan is not merely an incremental improvement but a significant leap forward, offering up to an 74-fold reduction in runtime and 14-fold decrease in memory usage compared to the widely used tool *HUMAnN*. Crucially, these gains are achieved alongside superior accuracy in resolving functional profiles at both the strain (genome) and pangenome levels where accuracy is improved up to 12% and 5%, respectively.

The substantial performance gains observed in *Leviathan* are primarily attributed to a fundamental shift in its core algorithmic strategy. Unlike *HUMAnN*, which relies on computationally expensive translated-blast searches to align reads from DNA-space into protein-space, *Leviathan* leverages the sensitivity of pseudo-alignment capabilities employed via *Salmon*. By mapping reads directly to a reference catalog of coding sequence in DNA-space, *Leviathan* bypasses the costly translation step and benefits from *Salmon's* highly optimized algorithms. This architectural choice transforms functional profiling from a process that can take many hours on high-performance compute nodes into a task that can be completed in minutes on standard workstations. This democratization of high-throughput analysis lowers the barrier to entry and enables researchers to investigate larger datasets in a scalable way.

Importantly, these performance improvements do not come at the cost of accuracy as *Leviathan* demonstrates superior or competitive performance compared to *HUMAnN*. The enhanced accuracy, particularly at the strain-level, can be credited to *Salmon's* `meta` mode, which is specifically designed to handle the quantification of features in complex mixtures with high sequence similarity, analogous to isoform quantification in single-cell transcriptomics. This is a more nuanced approach than the "best-hit" protein alignment strategy, which can struggle to differentiate reads from closely related genomes and confirmed by the increase in accuracy of *HUMAnN* when measured at the pangenome level. These results show that while *Leviathan* achieves higher accuracy at the genome-level, any minor misassignments between closely-related strains are effectively resolved when aggregating to the pangenome-level, resulting in near-perfect accuracy. This demonstrates the robustness of the pangenome as the ideal unit for functional analysis in complex communities.

A key advance offered by *Leviathan* is its seamless, native support for pangenome-resolved analysis and flexibility for custom reference databases. While the concept of pangenome analysis is not new, its practical implementation in existing functional profiling tools is often cumbersome, requiring users to navigate complex, multi-step processes to build custom databases. *Leviathan* streamlines this entirely, integrating pangenome definitions at the indexing stage and automatically generating both genome and pangenome level outputs. This functionality makes it trivial for researchers to investigate critical biological questions related to functional redundancy, niche specialization, and the distribution of metabolic capabilities across related microbial populations.

The synergy of massive performance gains and high accuracy opens the door to previously intractable research at a global scale such as scalable biodiversity monitoring. As sequencing projects generate petabytes of data from vast environmental surveys like global ocean transects (e.g., *Tara* (Sunagawa et al., 2020)*, GOS* (Rusch et al., 2007)) or continent-spanning soil studies (Ma et al., 2023), the need for tools that can keep pace is paramount. *Leviathan's* speed and low resource footprint make it possible to process thousands of metagenomes and metatranscriptomes, enabling a dynamic, near-real-time view of ecosystem function. This work contributes to the paradigm shift in cataloging "who is there" to robustly quantifying "what can they or are doing" across entire biomes at different points in time, tracking shifts in biogeochemical cycles and ecosystem responses to climate change with unprecedented resolution.

In the realm of human health, the implications are equally profound. The speed and accuracy of *Leviathan* could significantly reduce the turnaround time for clinical metagenomic analyses, moving from days to hours. Its strain-level accuracy is critical for distinguishing between pathogenic and commensal strains of the same species or for tracking the transmission of an antibiotic-resistant bacterium during a hospital outbreak. Furthermore, by enabling routine pangenome-level analysis, *Leviathan* provides clinicians and researchers with a powerful tool to survey the complete functional repertoire of a patient's microbiota, identifying virulence factors or metabolic pathways that may be linked to disease states.

While *Leviathan* represents a significant advance, there are areas for future development. Its current functional annotation partially relies on KEGG, but the modular architecture is designed to readily incorporate other functional databases such as *MetaCyc* if pathway graphs are available which would broaden its utility. Future work will focus on expanding this database support and further optimizing data structures for even larger-scale meta-analyses.

In conclusion, *Leviathan* addresses a critical and growing need in the field of meta-omics. By delivering a solution that is orders of magnitude faster, more memory-efficient, and more accurate than current standards, it effectively removes the computational bottleneck in functional profiling. *Leviathan* makes comprehensive, (pan)genome-resolved functional analysis a routine, scalable, and accessible task, empowering researchers to ask more complex questions of their data and paving the way for new discoveries in both environmental science and human medicine.

Data Availability

Leviathan is an *Apache 2.0*-licensed open-source software package available at https://github.com/jolespin/leviathan.

Acknowledgements

HUMAnN developers for laying the foundation for this research. Courtney Nichols Gould and Gordon Gould for their support in developing this software. Jayson A. Gutierrez Betancur for inspiring the development of this work for conservation. Steve Sperandeo for setting up computational infrastructure and wisdom on distributing workflows. Ekaterina Sakharova (https://github.com/KateSakharova) the developer for the original implementation *kegg-pathways-completeness-tool*. Antonio P. Camargo for assistance in preparing *PyKOfamSearch* and *PyHMMSearch* for *bioconda*.

Figures

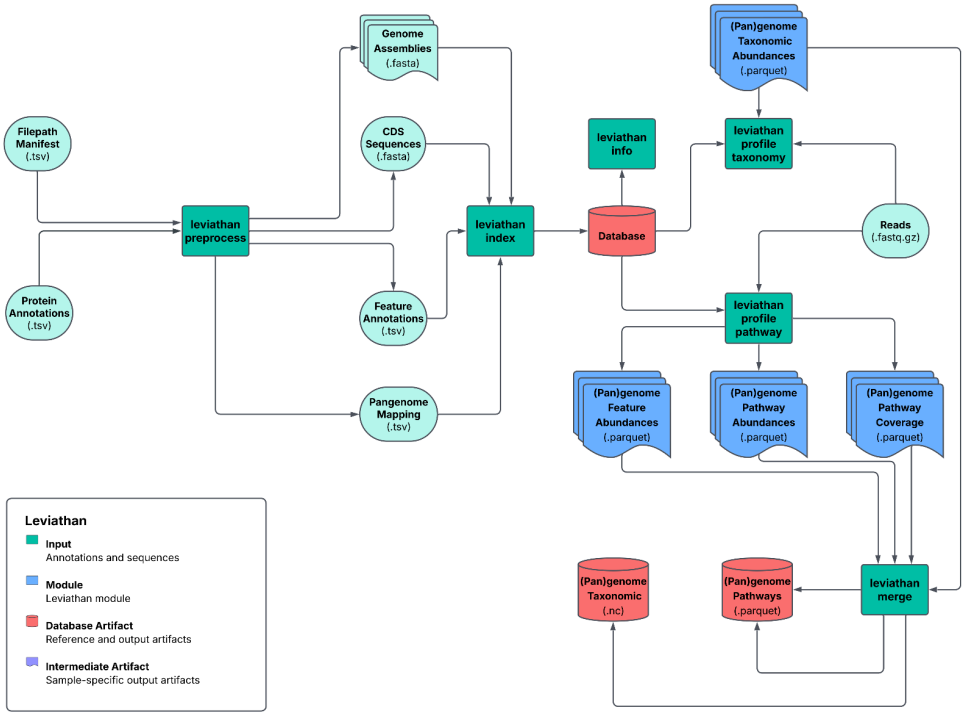

Figure 1 - Flowchart of *Leviathan* core modules

Tables

Feature	***Leviathan***	***HUMAnN***
Alignment method	salmon quant	diamond tblastx
Alignment space	DNA	Protein
Requires MinPath	No	Yes
Requires taxonomic lineage	No	Yes
Requires translated blast	No	Yes
Requires UniRef for custom databases	No	Yes
Support custom databases	Yes	Yes
Supports functional profiling	Yes	Yes
Supports paired-end Reads	Yes	No
Supports custom pangenomes*	Yes	Yes
Supports taxonomic profiling	Yes	No
Supports stratified counts tables	No	Yes
Supports lazy loading data structures	Yes	No

Table 1 - Comparison of features between *Leviathan* and *HUMAnN*. Asterisks indicates that *HUMAnN* can support custom pangenomes but only through building custom *MetaPhlAn* and *Chocophlan* databases.

	CAMI-I			**CAMI-II**	**CAMI**
	CAMI_high_toy	**CAMI_low_toy**	**CAMI_medium_toy**	**Marine**	**Total**
HUMAnN	2034	30	449	5506	8019
Leviathan	2029	30	447	4696	7202
Genomes With Gene Predictions	2034	30	449	5506	8019
Genomes Passed QC	1280	23	255	1571	3129
Genomes With Pathway *KOfams*	2029	30	447	4696	7202
Total Genomes	2036	30	450	6359	8875
Total Pangenomes	789	30	270	2647	3736

Table 2 - Comparison of genomes from CAMI datasets

		Leviathan		*HUMAnN*		Fold Improvement	
		Duration (minutes)	**Peak Memory (GB)**	**Duration (minutes)**	**Peak Memory (GB)**	**Duration**	**Memory**
CAMI_high_toy	H_S001	14.61	2.34	1083.57	32.31	74.19	13.84
	H_S002	14.89	2.35	949.73	32.28	63.78	13.75
	H_S003	14.96	2.34	875.83	32.64	58.56	13.97
	H_S004	15.02	2.35	852.18	32.33	56.72	13.79
	H_S005	15.27	2.33	826.25	32.23	54.13	13.82
CAMI_medium_toy	M2_S001	5.78	1.62	219.25	15.95	37.96	9.83
	M2_S002	5.81	1.62	174.12	16.96	29.95	10.45
CAMI_low_toy	S_S001	3.29	1.27	76.90	10.00	23.40	7.87
Marine	sample_0	13.78	2.70	119.52	17.92	8.68	6.63
	sample_1	15.22	2.69	121.30	18.00	7.97	6.69
	sample_2	14.97	2.71	120.27	17.99	8.03	6.65
	sample_3	17.10	2.71	124.05	17.83	7.25	6.59
	sample_4	14.32	2.74	118.47	17.82	8.27	6.51
	sample_5	15.53	2.72	119.40	17.80	7.69	6.54
	sample_6	16.09	2.71	119.72	17.91	7.44	6.62
	sample_7	14.90	2.73	119.92	17.92	8.05	6.56
	sample_8	16.41	2.72	121.73	17.95	7.42	6.61
	sample_9	14.45	2.73	118.87	17.79	8.23	6.51

Table 3 - Runtime performance benchmarks for pathway profiling of CAMI genomes using 16 threads

	Accuracy	*Leviathan*		*HUMAnN*		**Improvement**	
Dataset	**SampleID**	**Genome**	**Pangenome**	**Genome**	**Pangenome**	**Genome**	**Pangenome**
CAMI_high_toy	H_S001	0.9492	0.9970	0.9049	0.9610	0.0442	0.0360
	H_S002	0.9551	0.9899	0.8992	0.9591	0.0558	0.0308
	H_S003	0.9556	0.9888	0.9004	0.9598	0.0553	0.0290

	H_S004	0.9496	0.9872	0.8947	0.9588	0.0548	0.0284
	H_S005	0.9420	0.9877	0.8901	0.9573	0.0519	0.0304
CAMI_medium_toy	M2_S001	0.9692	0.9983	0.9101	0.9620	0.0591	0.0363
	M2_S002	0.9762	0.9988	0.9177	0.9650	0.0585	0.0338
CAMI_low_toy	S_S001	1.0000	1.0000	0.9845	0.9845	0.0155	0.0155
Marine	sample_0	0.9727	0.9933	0.8783	0.9538	0.0944	0.0396
	sample_1	0.9298	0.9922	0.8793	0.9554	0.0505	0.0367
	sample_2	0.9686	0.9817	0.8768	0.9393	0.0918	0.0424
	sample_3	0.9706	0.9842	0.8596	0.9517	0.1110	0.0325
	sample_4	0.9661	0.9880	0.8454	0.9389	0.1207	0.0491
	sample_5	0.9614	0.9856	0.8740	0.9612	0.0874	0.0244
	sample_6	0.9283	0.9869	0.8684	0.9574	0.0599	0.0295
	sample_7	0.9231	0.9942	0.8719	0.9466	0.0512	0.0476
	sample_8	0.9703	0.9889	0.8764	0.9488	0.0940	0.0401
	sample_9	0.9459	0.9859	0.8657	0.9548	0.0802	0.0311

Table 4 - Accuracy performance benchmarks for pathway profiling of CAMI genomes and pangenomes ranging from 0.0 to 1.0 where 1.0 represents perfect accuracy

References

Aramaki, T., Blanc-Mathieu, R., Endo, H., Ohkubo, K., Kanehisa, M., Goto, S., Ogata, H., 2020. KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold. Bioinformatics 36, 2251–2252. https://doi.org/10.1093/bioinformatics/btz859

Beghini, F., McIver, L.J., Blanco-Míguez, A., Dubois, L., Asnicar, F., Maharjan, S., Mailyan, A., Manghi, P., Scholz, M., Thomas, A.M., Valles-Colomer, M., Weingart, G., Zhang, Y., Zolfo, M., Huttenhower, C., Franzosa, E.A., Segata, N., 2021. Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3. eLife 10, e65088. https://doi.org/10.7554/eLife.65088

Blanco-Míguez, A., Beghini, F., Cumbo, F., McIver, L.J., Thompson, K.N., Zolfo, M., Manghi, P., Dubois, L., Huang, K.D., Thomas, A.M., Nickols, W.A., Piccinno, G., Piperni, E., Punčochář, M., Valles-Colomer, M., Tett, A., Giordano, F., Davies, R., Wolf, J., Berry, S.E., Spector, T.D., Franzosa, E.A., Pasolli, E., Asnicar, F., Huttenhower, C., Segata, N., 2023. Extending and improving metagenomic taxonomic profiling with uncharacterized species using MetaPhlAn 4. Nat. Biotechnol. 41, 1633–1644. https://doi.org/10.1038/s41587-023-01688-w

Bushnell, B., 2014. BBMap: A Fast, Accurate, Splice-Aware Aligner (No. LBNL-7065E). Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States).

Caspi, R., Billington, R., Keseler, I.M., Kothari, A., Krummenacker, M., Midford, P.E., Ong, W.K., Paley, S., Subhraveti, P., Karp, P.D., 2020. The MetaCyc database of metabolic pathways and enzymes - a 2019 update. Nucleic Acids Res. 48, D445–D453. https://doi.org/10.1093/nar/gkz862

DNA Sequencing Costs: Data [WWW Document], n.d. URL https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data (accessed 6.25.25).

Espinoza, J.L., Dupont, C.L., 2022. VEBA: a modular end-to-end suite for in silico recovery, clustering, and analysis of prokaryotic, microeukaryotic, and viral genomes from metagenomes. BMC Bioinformatics 23, 419. https://doi.org/10.1186/s12859-022-04973-8

Espinoza, J.L., Phillips, A., Prentice, M.B., Tan, G.S., Kamath, P.L., Lloyd, K.G., Dupont, C.L., 2024. Unveiling the microbial realm with VEBA 2.0: a modular bioinformatics suite for end-to-end genome-resolved prokaryotic, (micro)eukaryotic and viral multi-omics from either short- or long-read sequencing. Nucleic Acids Res. 52, e63. https://doi.org/10.1093/nar/gkae528

Espinoza, J.L., Shah, N., Singh, S., Nelson, K.E., Dupont, C.L., 2020. Applications of weighted association networks applied to

compositional data in biology. Environ. Microbiol. 22, 3020–3038. https://doi.org/10.1111/1462-2920.15091
Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. 39, W29–W37. https://doi.org/10.1093/nar/gkr367
Hoyer, S., Hamman, J., 2017. xarray: N-D labeled Arrays and Datasets in Python. J. Open Res. Softw. 5. https://doi.org/10.5334/jors.148
Larralde, M., Zeller, G., 2023. PyHMMER: a Python library binding to HMMER for efficient sequence analysis. Bioinformatics 39, btad214. https://doi.org/10.1093/bioinformatics/btad214
Ma, B., Lu, C., Wang, Y., Yu, J., Zhao, K., Xue, R., Ren, H., Lv, X., Pan, R., Zhang, J., Zhu, Y., Xu, J., 2023. A genomic catalogue of soil microbiomes boosts mining of biodiversity and genetic resources. Nat. Commun. 14, 7318. https://doi.org/10.1038/s41467-023-43000-z
Meyer, F., Fritz, A., Deng, Z.-L., Koslicki, D., Lesker, T.R., Gurevich, A., Robertson, G., Alser, M., Antipov, D., Beghini, F., Bertrand, D., Brito, J.J., Brown, C.T., Buchmann, J., Buluç, A., Chen, B., Chikhi, R., Clausen, P.T.L.C., Cristian, A., Dabrowski, P.W., Darling, A.E., Egan, R., Eskin, E., Georganas, E., Goltsman, E., Gray, M.A., Hansen, L.H., Hofmeyr, S., Huang, P., Irber, L., Jia, H., Jørgensen, T.S., Kieser, S.D., Klemetsen, T., Kola, A., Kolmogorov, M., Korobeynikov, A., Kwan, J., LaPierre, N., Lemaitre, C., Li, C., Limasset, A., Malcher-Miranda, F., Mangul, S., Marcelino, V.R., Marchet, C., Marijon, P., Meleshko, D., Mende, D.R., Milanese, A., Nagarajan, N., Nissen, J., Nurk, S., Oliker, L., Paoli, L., Peterlongo, P., Piro, V.C., Porter, J.S., Rasmussen, S., Rees, E.R., Reinert, K., Renard, B., Robertsen, E.M., Rosen, G.L., Ruscheweyh, H.-J., Sarwal, V., Segata, N., Seiler, E., Shi, L., Sun, F., Sunagawa, S., Sørensen, S.J., Thomas, A., Tong, C., Trajkovski, M., Tremblay, J., Uritskiy, G., Vicedomini, R., Wang, Zhengyang, Wang, Ziye, Wang, Zhong, Warren, A., Willassen, N.P., Yelick, K., You, R., Zeller, G., Zhao, Z., Zhu, S., Zhu, J., Garrido-Oter, R., Gastmeier, P., Hacquard, S., Häußler, S., Khaledi, A., Maechler, F., Mesny, F., Radutoiu, S., Schulze-Lefert, P., Smit, N., Strowig, T., Bremges, A., Sczyrba, A., McHardy, A.C., 2022. Critical Assessment of Metagenome Interpretation: the second round of challenges. Nat. Methods 19, 429–440. https://doi.org/10.1038/s41592-022-01431-4
Molina-Pardines, C., Haro-Moreno, J.M., Rodriguez-Valera, F., López-Pérez, M., 2025. Extensive paralogism in the environmental pangenome: a key factor in the ecological success of natural SAR11 populations. Microbiome 13, 41. https://doi.org/10.1186/s40168-025-02037-6
Nishimura, Y., Yoshizawa, S., 2022. The OceanDNA MAG catalog contains over 50,000 prokaryotic genomes originated from various marine environments. Sci. Data 9, 305. https://doi.org/10.1038/s41597-022-01392-5
Olm, M.R., Brown, C.T., Brooks, B., Banfield, J.F., 2017. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. ISME J. 11, 2864–2868. https://doi.org/10.1038/ismej.2017.126
Parks, D.H., Chuvochina, M., Rinke, C., Mussig, A.J., Chaumeil, P.-A., Hugenholtz, P., 2022. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. Nucleic Acids Res. 50, D785–D794. https://doi.org/10.1093/nar/gkab776
Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C., 2017. Salmon provides fast and bias-aware quantification of transcript expression. Nat. Methods 14, 417–419. https://doi.org/10.1038/nmeth.4197
Piro, V.C., Dadi, T.H., Seiler, E., Reinert, K., Renard, B.Y., 2020. ganon: precise metagenomics classification against large and up-to-date sets of reference sequences. Bioinformatics 36, i12–i20. https://doi.org/10.1093/bioinformatics/btaa458
Quinn, T.P., Erb, I., 2020. Amalgams: data-driven amalgamation for the dimensionality reduction of compositional data. NAR Genomics Bioinforma. 2, lqaa076. https://doi.org/10.1093/nargab/lqaa076
Rusch, D.B., Halpern, A.L., Sutton, G., Heidelberg, K.B., Williamson, S., Yooseph, S., Wu, D., Eisen, J.A., Hoffman, J.M., Remington, K., Beeson, K., Tran, B., Smith, H., Baden-Tillson, H., Stewart, C., Thorpe, J., Freeman, J., Andrews-Pfannkoch, C., Venter, J.E., Li, K., Kravitz, S., Heidelberg, J.F., Utterback, T., Rogers, Y.-H., Falcón, L.I., Souza, V., Bonilla-Rosso, G., Eguiarte, L.E., Karl, D.M., Sathyendranath, S., Platt, T., Bermingham, E., Gallardo, V., Tamayo-Castillo, G., Ferrari, M.R., Strausberg, R.L., Nealson, K., Friedman, R., Frazier, M., Venter, J.C., 2007. The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pacific. PLOS Biol. 5, e77. https://doi.org/10.1371/journal.pbio.0050077
Ruscheweyh, H.-J., Milanese, A., Paoli, L., Sintsova, A., Mende, D.R., Zeller, G., Sunagawa, S., 2021. mOTUs: Profiling Taxonomic Composition, Transcriptional Activity and Strain Populations of Microbial Communities. Curr. Protoc. 1, e218. https://doi.org/10.1002/cpz1.218
Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., Bremges, A., Fritz, A., Garrido-Oter, R., Jørgensen, T.S., Shapiro, N., Blood, P.D., Gurevich, A., Bai, Y., Turaev, D., DeMaere, M.Z., Chikhi, R., Nagarajan, N., Quince, C., Meyer, F., Balvočiūtė, M., Hansen, L.H., Sørensen, S.J., Chia, B.K.H., Denis, B., Froula, J.L., Wang, Z., Egan, R., Don Kang, D., Cook, J.J., Deltel, C., Beckstette, M., Lemaitre, C., Peterlongo, P., Rizk, G., Lavenier, D., Wu, Y.-W., Singer, S.W., Jain, C., Strous, M., Klingenberg, H., Meinicke, P., Barton, M.D., Lingner, T., Lin, H.-H., Liao, Y.-C., Silva, G.G.Z., Cuevas, D.A., Edwards, R.A., Saha, S., Piro, V.C., Renard, B.Y., Pop, M., Klenk, H.-P., Göker, M., Kyrpides, N.C., Woyke, T., Vorholt, J.A., Schulze-Lefert, P., Rubin, E.M., Darling, A.E., Rattei, T., McHardy, A.C., 2017. Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. Nat. Methods 14, 1063–1071. https://doi.org/10.1038/nmeth.4458
Shaiber, A., Eren, A.M., 2019. Composite Metagenome-Assembled Genomes Reduce the Quality of Public Genome Repositories. mBio 10, 10.1128/mbio.00725-19. https://doi.org/10.1128/mbio.00725-19
Shaw, J., Yu, Y.W., 2024. Rapid species-level metagenome profiling and containment estimation with sylph. Nat. Biotechnol. 1–12. https://doi.org/10.1038/s41587-024-02412-y
Shaw, J., Yu, Y.W., 2023. Fast and robust metagenomic sequence comparison through sparse chaining with skani. Nat. Methods 20, 1661–1665. https://doi.org/10.1038/s41592-023-02018-3
Srivastava, A., Malik, L., Smith, T., Sudbery, I., Patro, R., 2019. Alevin efficiently estimates accurate gene abundances from dscRNA-seq data. Genome Biol. 20, 65. https://doi.org/10.1186/s13059-019-1670-y
Sunagawa, S., Acinas, S.G., Bork, P., Bowler, C., Eveillard, D., Gorsky, G., Guidi, L., Iudicone, D., Karsenti, E., Lombard, F., Ogata, H., Pesant, S., Sullivan, M.B., Wincker, P., de Vargas, C., 2020. Tara Oceans: towards global ocean ecosystems biology. Nat. Rev. Microbiol. 18, 428–445. https://doi.org/10.1038/s41579-020-0364-5

Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C.H., the UniProt Consortium, 2015. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics 31, 926–932. https://doi.org/10.1093/bioinformatics/btu739

Wood, D.E., Lu, J., Langmead, B., 2019. Improved metagenomic analysis with Kraken 2. Genome Biol. 20, 257. https://doi.org/10.1186/s13059-019-1891-0

Ye, Y., Doak, T.G., 2009. A Parsimony Approach to Biological Pathway Reconstruction/Inference for Genomes and Metagenomes. PLoS Comput. Biol. 5, e1000465. https://doi.org/10.1371/journal.pcbi.1000465