

# Metaphor—A workflow for streamlined assembly and binning of metagenomes

Vinícius W. Salazar<sup>1</sup>, Babak Shaban<sup>2,†</sup>, Maria del Mar Quiroga<sup>2</sup>, Robert Turnbull<sup>2</sup>, Edoardo Tescari<sup>2</sup>, Vanessa Rossetto Marcelino<sup>3,4,5,6</sup>, Heroen Verbruggen<sup>5,†</sup> and Kim-Anh Lê Cao<sup>1,\*</sup>

<sup>1</sup>Melbourne Integrative Genomics, School of Mathematics & Statistics, University of Melbourne, Parkville, VIC 3052, Victoria, Australia

<sup>2</sup>Melbourne Data Analytics Platform (MDAP), University of Melbourne, Carlton, VIC 3053, Victoria, Australia

<sup>3</sup>Department of Molecular and Translational Sciences, Monash University, Clayton, VIC 3168, Victoria, Australia

<sup>4</sup>Centre for Innate Immunity and Infectious Diseases, Hudson Institute of Medical Research, Clayton, VIC 3168, Victoria, Australia

<sup>5</sup>School of BioSciences, University of Melbourne, Parkville, VIC 3052, Victoria, Australia

<sup>6</sup>Department of Microbiology and Immunology, The University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Parkville, VIC 3052, Victoria, Australia

\*Correspondence address. Kim-Anh Lê Cao, School of Mathematics and Statistics, Peter Hall Building, Bldg 160, University of Melbourne Parkville VIC 3010, Australia. E-mail: [kimanh.lecao@unimelb.edu.au](mailto:kimanh.lecao@unimelb.edu.au)

<sup>†</sup>Deceased.

<sup>‡</sup>Contributed equally.

## Abstract

Recent advances in bioinformatics and high-throughput sequencing have enabled the large-scale recovery of genomes from metagenomes. This has the potential to bring important insights as researchers can bypass cultivation and analyze genomes sourced directly from environmental samples. There are, however, technical challenges associated with this process, most notably the complexity of computational workflows required to process metagenomic data, which include dozens of bioinformatics software tools, each with their own set of customizable parameters that affect the final output of the workflow. At the core of these workflows are the processes of assembly—combining the short-input reads into longer, contiguous fragments (contigs)—and binning, clustering these contigs into individual genome bins. The limitations of assembly and binning algorithms also pose different challenges depending on the selected strategy to execute them. Both of these processes can be done for each sample separately or by pooling together multiple samples to leverage information from a combination of samples. Here we present Metaphor, a fully automated workflow for genome-resolved metagenomics (GRM). Metaphor differs from existing GRM workflows by offering flexible approaches for the assembly and binning of the input data and by combining multiple binning algorithms with a bin refinement step to achieve high-quality genome bins. Moreover, Metaphor generates reports to evaluate the performance of the workflow. We showcase the functionality of Metaphor on different synthetic datasets and the impact of available assembly and binning strategies on the final results.

**Keywords:** bioinformatics, pipeline, MAGs, Snakemake, high-throughput sequencing, microbial genomics

## Introduction

Genome-resolved metagenomics (GRM) is a set of techniques for the recovery of genomes from high-throughput sequencing data. Applications of GRM have led to unprecedented insight into microbial diversity, ecology, and evolution, due to the recovery of (mostly uncultivated) metagenome-assembled genomes (MAGs) [1–4]. MAGs are essentially “bins” of contigs that are clustered together based on differential coverage and sequence composition; a bin is considered a MAG when it displays a high degree of completeness and a low degree of redundancy/contamination, which is usually calculated through the presence of marker genes in the bin. Advances in GRM have consistently improved the quality of recovered MAGs, and large-scale studies reconstructing and analyzing thousands of MAGs have become prominent in microbiology research. Even with the inherent biases that accompany the generation of MAGs, it is evident that the benefits outweigh the risks, and researchers are increasingly in need of automated data processing methods for assembling and binning metagenomes [5]. Data pipelines that perform such experiments are inherently complex, have high computing cost, use heteroge-

neous data sources, have dozens of customizable parameters, and depend on several specialized bioinformatics software [6, 7].

An additional domain-specific challenge for GRM studies is the strategy used for assembling and binning each sequenced sample. Data (raw reads generated by the sequencer) originating from multiple samples may be assembled separately or pooled together, depending whether they come from the same population, specimen, or environment. This results in either a set of contigs for each sample or a “coassembly” of the pooled samples. Similarly, in the metagenome binning step, where contigs are clustered into genome bins, one may do this individually for each set of assembled contigs or by pooling together contigs from multiple samples and then mapping each individual sample to this catalog of contigs (“cobinning”) [8]. The latter approach allows binning algorithms to account for differential coverage of contigs across samples, enriching the information available for clustering. The chosen strategy for assembly and binning may have important consequences for the final results (i.e., the quality of the assembly and of the recovered bins) [8]. It is hypothesized that pooled assembly and binning may lead to improved results when

Received: March 20, 2023. Revised: June 5, 2023. Accepted: July 4, 2023

© The Author(s) 2023. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

analyzing communities with high genetic diversity and to poorer results when there is a high level of intraspecies/strain-level diversity [9].

Here we present Metaphor, an automated and flexible workflow for the assembly and binning of metagenomes, which recovers prokaryotic genomes from metagenomes efficiently and with high sensitivity, and it provides taxonomic and functional abundance data for quantitative metagenome analyses. Our software advances existing metagenomic pipelines by combining 2 core features: the usage of multiple binning software, along with a binning refinement step, and the possibility of defining groups for assembly and binning of samples. This effectively allows scaling Metaphor to process multiple datasets in a single execution, performing assembly and binning in separate batches for each dataset, and avoiding the need for repeated executions with different input datasets. The workflow includes native functionality for downstream integration with omics statistical toolkits [10, 11], so that abundance data can be easily imported into these tools, and with the Anvi'o [12] platform, which allows importing the collections of bins generated by Metaphor along with contig coverage data. Metaphor generates detailed performance metrics at the end of each module of the workflow to provide users with a high-level summary of their analysis, and it has been designed to be user-friendly, portable, and flexible, as users can choose between different strategies for assembly and binning. We demonstrate its functionality using different synthetic datasets and discuss how these different strategies can impact data analyses in terms of quality of the resulting assembly and genome bins.

## Design and Implementation

Metaphor stands out from existing GRM pipelines by offering flexible options for assembly and binning combined with multiple binning software and a binning refinement step. See Table 1 for a comparison of Metaphor's features with other state-of-the-art GRM workflows. The workflow is implemented with Snakemake [13], a widely used scientific workflow management system. In each module, computing steps (called "rules" by Snakemake) consist of both third-party bioinformatics software [14–28] and custom scripts that connect different parts of the workflow, listed in Table 2.

The workflow consists of 6 modules: quality control (QC), assembly, annotation, mapping, binning, and postprocessing. In the QC module, raw sequencing reads are filtered and trimmed. Metagenomic assembly is then performed. Coding sequences are predicted from the assembled contigs and used for functional and taxonomic annotation. The quality-filtered reads are mapped against the contigs, generating coverage statistics employed by the binning algorithms. After binning is complete, bins are refined and dereplicated. Lastly, the postprocessing module renders runtime and memory usage metrics and generates an HTML report. A simplified version of the flow of data between the different modules of the workflow is shown in Fig. 1.

The choice of bioinformatics tools was informed by the results of the Second Critical Assessment for Metagenome Interpretation (CAMI II) [8, 36], striving for the maximum trade-off between performance, efficiency, and software sustainability. Although the latter is a subjective factor, selecting and streamlining dependencies with regard to code quality, maintenance, and community support is a critical factor when maintaining complex bioinformatics pipelines [6, 37]. Each third-party software (along with its

version) is defined in an individual requirements file that is used by Snakemake to create a virtual environment and run that particular step. To facilitate citing these tools, Metaphor packages a `bibs/` directory containing all citations in the Bibtext format.

The workflow takes 2 files as input: a tab-delimited file containing sample names and file paths to the raw reads and a configuration file in the YAML format, which will set the workflow parameters (see Fig. 1). These files can be automatically generated by Metaphor and edited by the user or created from scratch. The output of Metaphor consists of a directory for each module, further subdivided into the rules within each module. This is described in detail in the documentation [38].

## Assessment on CAMI II synthetic datasets

To demonstrate the functionality of Metaphor, we analyzed datasets from CAMI II [8], described in Table 3. All datasets consist of short and long reads generated by simulation of collections of reference genomes [39]. Only short reads were used for each dataset, as Metaphor does not yet support long reads. Specifically, we used the Marine metagenome dataset (identified as "marmg"), the Strain Madness dataset (identified as "strmg"), and the Human Microbiome dataset, which consists of 5 sets of samples, each corresponding to a different sampling location in the human body, which were treated as distinct datasets [3]. The following strategies were employed for each dataset: single assembly, single binning (SASB), where each sample is individually assembled and binned; single assembly, cobinning (SACB), where each sample is assembled individually and then binned with other samples from the same dataset; and coassembly, cobinning (CACB), where all samples from the dataset are assembled and binned together. Table 4 illustrates how this works in practice, in terms of generated output files. Metaphor allows defining multiple groups for coassembly or cobinning to analyze multiple independent datasets with a single execution.

In order to assess the effect of different assembly strategies, we used MetaQUAST [18] to compare the assemblies generated by the workflow with the collections of reference genomes. For the different binning strategies, we compared metrics obtained from DAS Tool, the software used for dereplicating and evaluating genome bins, after a second round of dereplication with dRep [40]. This is because data generated with the SASB strategy will likely result in redundant bins, as for that strategy, there is no dereplication between samples, and since samples within a dataset have similar composition, it is likely that a genome bin can be generated repeatedly by different samples. dRep performs dereplication based on the average nucleotide identity between genomes, a metric that has been consistently used as a proxy to differentiate taxonomy at the species and strain levels [41]. dRep was run with default clustering parameters and without any length, completeness, or contamination cutoffs. We used Spartan [42], the High Performance Computing system at the University of Melbourne, to run the pipeline. Jobs were dispatched to nodes with the SLURM scheduler, using up to 64 processors and 300 GB RAM per node.

## Results and Discussion

After running Metaphor on the CAMI II Marine, Strain Madness, and Human Microbiome datasets, we illustrate the different outputs generated by the workflow and compare the effects of different assembly and binning strategies on workflow performance.

**Table 1:** Comparison of features between Metaphor and state-of-the-art GRM workflows as listed by [29]. Data adapted to include Metaphor

Features	Metaphor v1.7.7	ATLAS [30]	MetaWRAP [31]	nf-core/mag [32]	MAGNETO [29]
Preprocessing					
Reads trimming	✓	✓	✓	✓	✓
Contamination	✓	✓	✓	✓	✓
Assembly					
Coassembly possible	✓		✓	✓	✓
Coassembly by groups	✓				
Compute sets to coassemble				✓	
Assembly evaluation	✓				
Binning					
Cobinning possible	✓		✓	✓	✓
Multiple binning software	✓	✓	✓		
Bin refinement	✓	✓	✓		
Bin reassembly		✓	✓		
Postprocessing					
MAGs quality check	✓	✓	✓	✓	✓
Dereplication step	✓	✓	✓	✓	✓
Genome annotation	✓	✓	✓	✓	✓
Gene catalog	✓			✓	✓
HTML report	✓	✓		✓	✓
Reproducibility					
Workflow management	✓	✓		✓	✓
Packages management	✓	✓		✓	✓

**Table 2:** Modules, steps, and software used in Metaphor

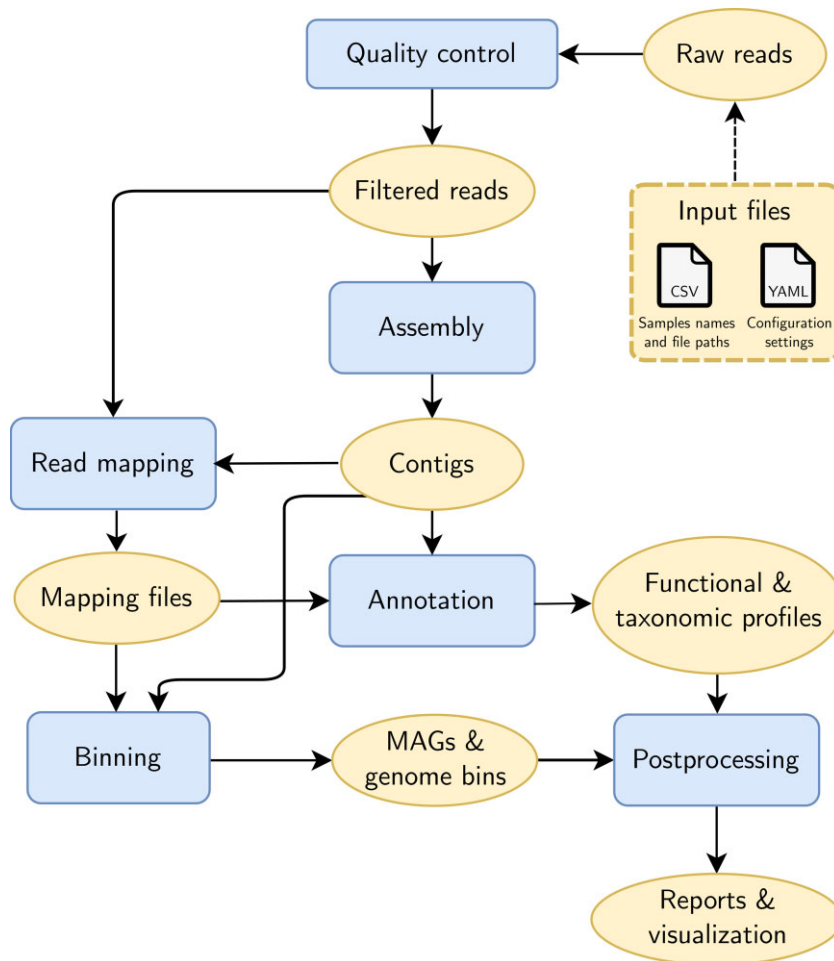
Module	Step	Software
Quality control (QC)	Trim adapters and filter low-quality reads	fastp [14]
	Generate QC reports	FastQC [15]
	Combine QC reports	MultiQC [16]
Assembly	Assemble filtered and merged reads into contigs	MegaHit [17]
	Perform assembly evaluation	MetaQUAST [18]
Mapping	Assemble report and plots	Metaphor script*
	Map reads	MiniMap2 [19]
	Sort and index mapped reads	Samtools [20]
Annotation	Predict coding sequences from contigs	Prodigal[21]
	Annotate coding sequences	Diamond, NCBI COG [22, 23]
	Annotate MAGs	Prokka [24]
Binning	Annotate report and plots	Metaphor script*
	Cluster contigs into bins	VAMB [25]
	Cluster contigs into bins	MetaBAT2 [26]
	Cluster contigs into bins	CONCOCT [27]
	Dereplicate and score bins	DAS Tool [28]
Postprocessing	Binning report and plots	Metaphor script*
	Concatenate benchmarks	Metaphor script*
	Plot benchmarks	Metaphor script*

\* External libraries used in Metaphor scripts: [33–35].

## Reconstruction of metagenome-assembled genomes

Metaphor produces genome bins generated with 3 tools—Vamb, MetaBAT2, and CONCOCT [25–27]—that are refined with the DAS Tool [28]. The DAS Tool performs bin refinement through a “dereplication, aggregation, and scoring” process, in which candidate bins are initially scored based on the presence/absence of single-copy marker genes (SCGs, which are a proxy for bin completeness). Redundant candidate bin sets are then aggregated, and an iterative scoring process is performed, so only the best-quality, nonredundant bins remain; the bin score ( $S_b$ ) increases

with the number of SCGs and decreases with duplicate SCGs per bin. Please refer to [28] for an overview of the DAS algorithm and the formula to determine the bin score. The input for each binning tool differs slightly, but they all rely on the catalog of contigs obtained from the assembly and the coverage files obtained from the read mapping module (see Fig. 1). A report is generated for each of the binning groups (only 1 is generated if cobinning is performed), which highlights 3 key metrics: completeness, redundancy, and bin score. The first 2 metrics are calculated by the presence/absence of single-copy genes, and the latter is a function of the former two. Plots generated by an



**Figure 1:** Simplified workflow diagram. Workflow modules are represented by rectangular blue shapes and data files are represented by oval yellow shapes, except for entrypoint files shown in a dashed yellow rectangle. Arrows indicate input and output of data between modules.

**Table 3:** Datasets from CAMI II used to assess the workflow. Columns show the number of samples and size in gigabytes of each dataset, along with the amount of reference genomes used to generate the dataset

Dataset	Identifier	No. of samples	Size (GB)	No. reference genomes
Marine	marmg	10	50	622
Strain Madness	strmg	100	200	408
Human Airways	h_airways	10	44	1,394
Human Genital	h_urogenital	9	39	1,394
Human Gut	h_gastrointestinal	10	44	1,057
Human Oral	h_oral	10	43	1,057
Human Skin	h_skin	10	44	1,394

example report are shown in Fig. 2. It is possible to compare the performance of the different binning software and obtain the proportion of bins above a specified particular quality threshold based on the bin score. The source table for the report is provided, so that users can generate custom reports and inspect specific individual bins. Bins that pass the quality threshold are stored in individual FASTA files, so they can easily be used for downstream analyses with tools such as CheckM or GTDB-Tk [43, 44]. We chose not to include these software in the workflow as they rely on fairly large reference databases and/or contain several different steps that are dependent on third-party software, which would affect Metaphor’s portability. Bin collections generated with Metaphor

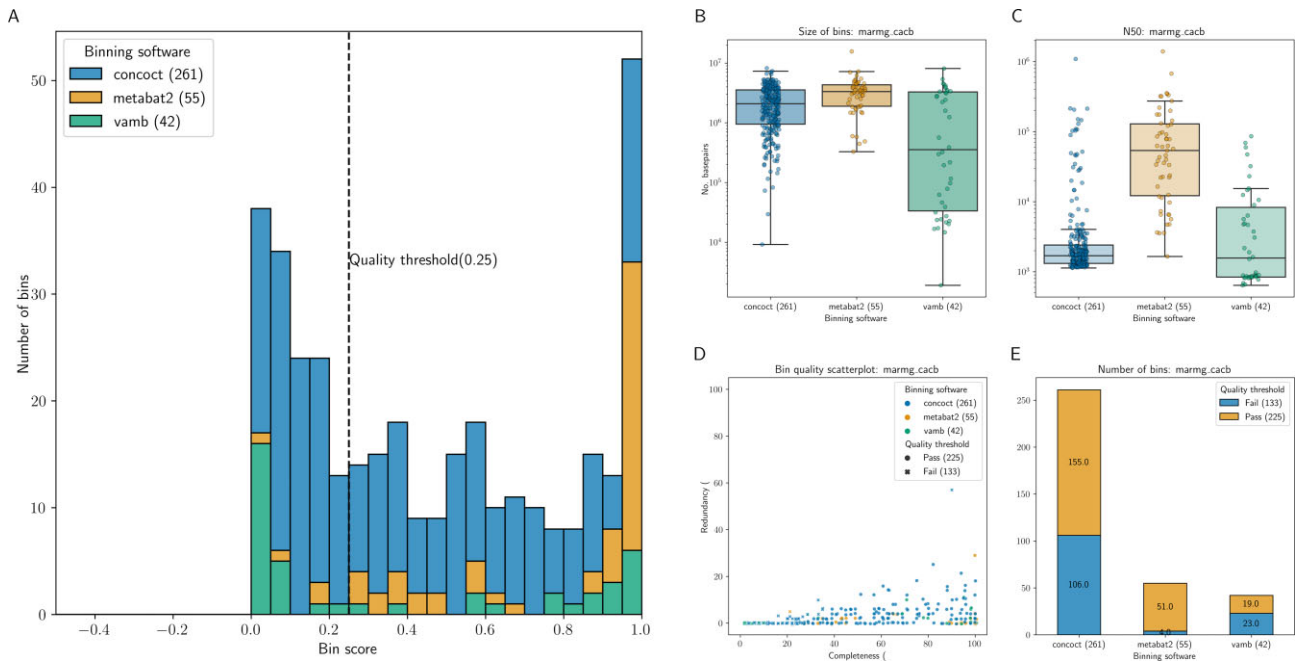
can be imported into the Anvi’o along with coverage data (BAM files), so users can use the interactive interface of Anvi’o to examine the bins.

### Contig-level taxonomic and functional profiling

To facilitate quantitative metagenomics applications, Metaphor’s annotation module generates contig-level functional and taxonomic profiles based on the NCBI COG database [23]. These are obtained by predicting coding sequences with Prodigal and then aligning the resulting amino acid files with Diamond [21, 22] in the “iterative” mode. This setting performs repeated rounds of alignment, with an increasing degree of sensitivity when no hits are

**Table 4:** Output files for each strategy. If only 1 dataset/group is being analyzed, assembly and binning results are named as “Coassembly” and “Cobinning,” respectively. If multiple datasets/groups are used, the results are named according to the group/dataset’s name

Strategy	Description	Reads files	Assemblies	Bins
SASB	Single assembly, single binning	Sample_0.fastq Sample_1.fastq Sample_2.fastq	Sample_0_contigs.fasta Sample_1_contigs.fasta Sample_2_contigs.fasta	Sample_0_bins/ Sample_1_bins/ Sample_2_bins/
SACB	Single assembly, cobinning	Sample_0.fastq Sample_1.fastq Sample_2.fastq	Sample_0_contigs.fasta Sample_1_contigs.fasta Sample_2_contigs.fasta	Cobinning_bins/
CACB	Coassembly, cobinning	Sample_0.fastq Sample_1.fastq Sample_2.fastq	Coassembly_contigs.fasta	Cobinning_bins/



**Figure 2:** Binning report generated by Metaphor for the CAMI II Marine metagenome dataset processed with the CACB (coassembly, cobinning) setting. (A) A stacked histogram of the distribution of bin scores, with the defined quality threshold highlighted as a dashed line. (B) The size (in base pairs) and (C) N50 of bins. The y-axis is in log-scale. (D) Scatterplot of completeness and redundancy for each bin. Colors indicate the tool used to generate the bin, and the symbols indicate whether that bin passed or failed the bin score quality threshold (corresponding to the same value in the dashed line of A). (E) The number of bins that passed or failed the quality threshold for each binning tool.

detected in the previous round. Abundances for each feature are calculated based on the coverage of all coding sequences that align to that feature. Figure 3 illustrates the profile visualizations offered by Metaphor: a heatmap of COG categories for the functional profile and a stacked barplot for the most abundant taxa (for the latter, 1 plot is generated for each taxonomic rank). The annotation module outputs count tables with both absolute and relative abundance values of taxa and functional categories and may be directly imported by downstream statistical toolkits such as MixOmics or PhyloSeq [10, 11].

### Quality control and performance metrics

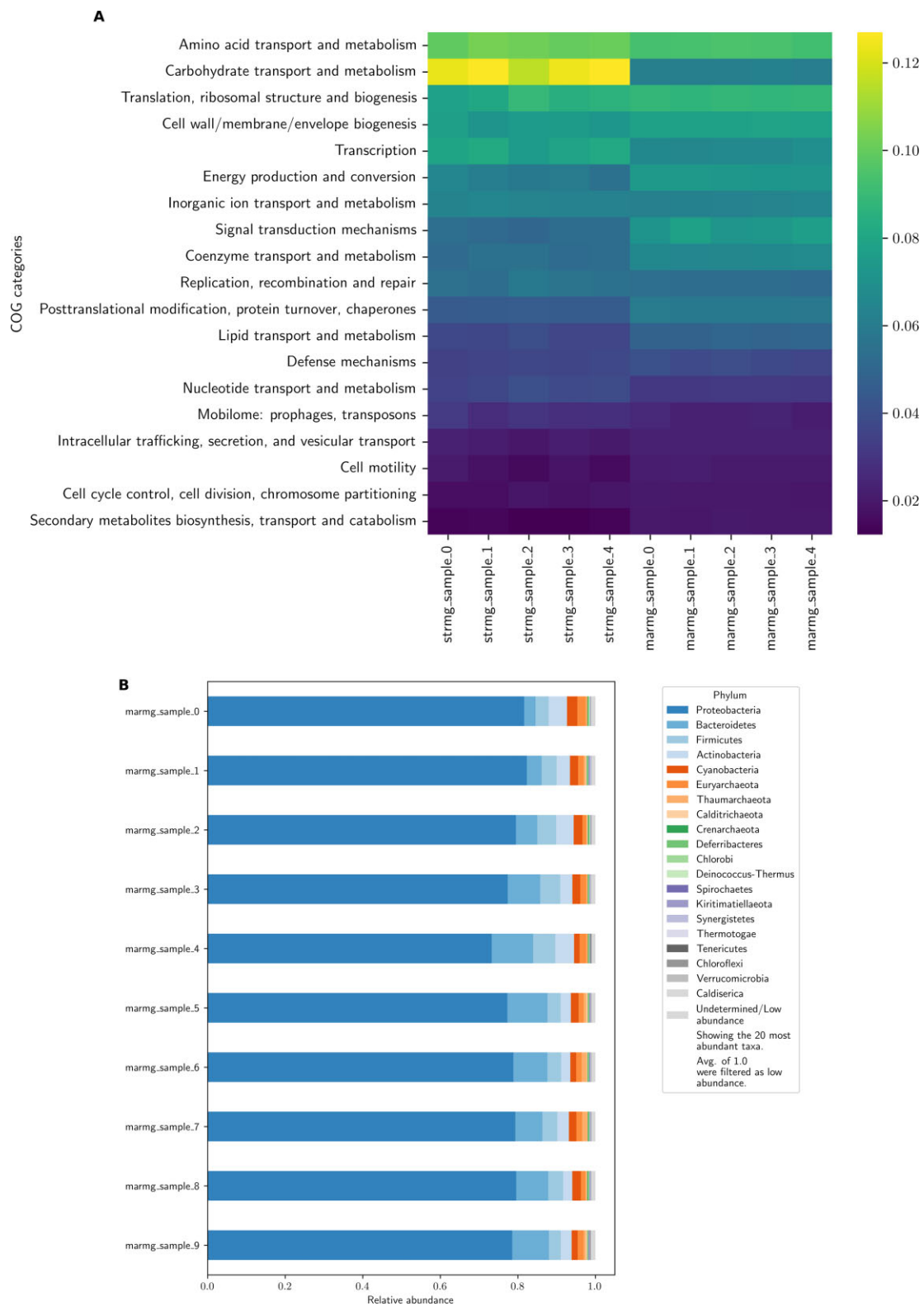
Additional outputs produced by Metaphor include the quality control reports from the fastp and FastQC tools, with a summary of FastQC outputs being produced by MultiQC [14–16]. A simple report is produced by the assembly module with sequence statistics of the assembled contigs (e.g., N50, number of contigs, total and mean length of contigs), and performance metrics. At the

end of the workflow execution, the postprocessing module generates figures obtained from the “benchmark” files provided by Snakemake. These files contain process information such as runtime and memory consumption. Metaphor plots these metrics in 2 ways: total per rule and per-sample mean (Fig. 4) as some rules run only once across all samples, while other rules run per sample. These plots help identify computational bottlenecks and assess whether computing resources are adequate.

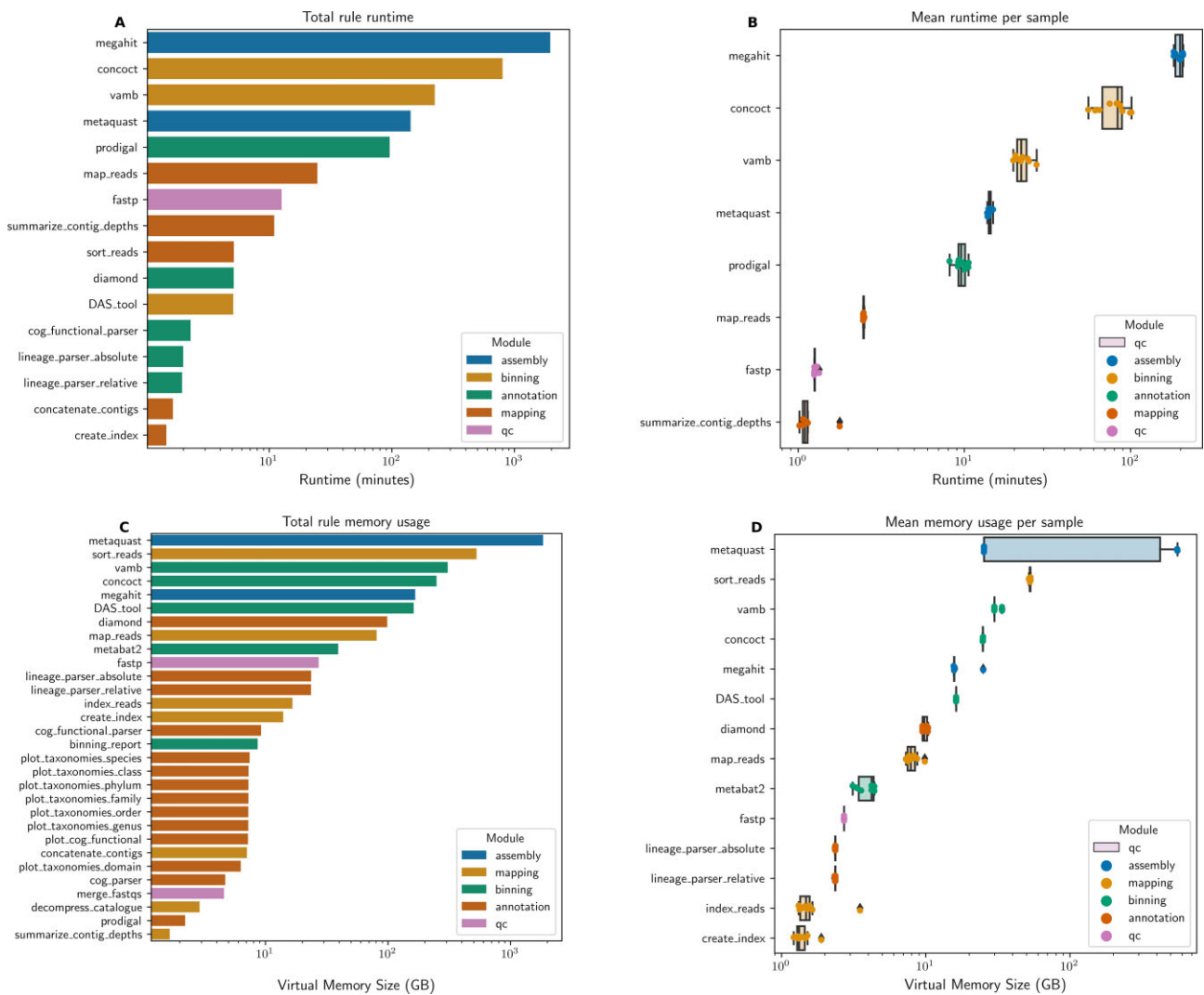
### Assembly and binning strategies

The effects of distinct assembly and binning strategies on the final output of metagenomic workflows are highly dependent on the data source and research context [8]. As such, the choice of individual or group assembly and binning can only be assessed a posteriori. We compared 3 different strategies: SASB, SACB, and CACB; see Tables 3 and 4 and “Assessment on CAMI II synthetic datasets” section for details. For assembly, we used the 5 different groups in the Human Microbiome dataset along with the Strain





**Figure 3:** Annotation plots generated by Metaphor on the Strain Madness (“strmg”) and the Marine (“marmg”) datasets. (A) The functional profile as a heatmap of the relative abundance of functional COG categories (y-axis) across samples (x-axis) for 5 samples from the Strain Madness and Marine datasets. (B) The taxonomic profile of the Marine dataset as a stacked barplot of relative abundance of taxa. In this case, the phylum rank was used, but Metaphor generates this for the most common taxonomic ranks (phylum, class, order, family, genus, species). The number of abundant taxa can be easily adjusted in the workflow settings. For both taxonomic and functional profiles, abundance of each feature is calculated from coverage values for each gene.



**Figure 4:** Performance metrics report generated by Metaphor on the Marine dataset processed with the SASB strategy. Total runtime per rule (A), mean runtime per sample (B), total memory usage per rule (C), and mean memory usage per sample (D). The x-axis is in log format. Cutoffs are applied to omit rules with short runtime or low memory usage. Colors indicate the workflow module of each rule.

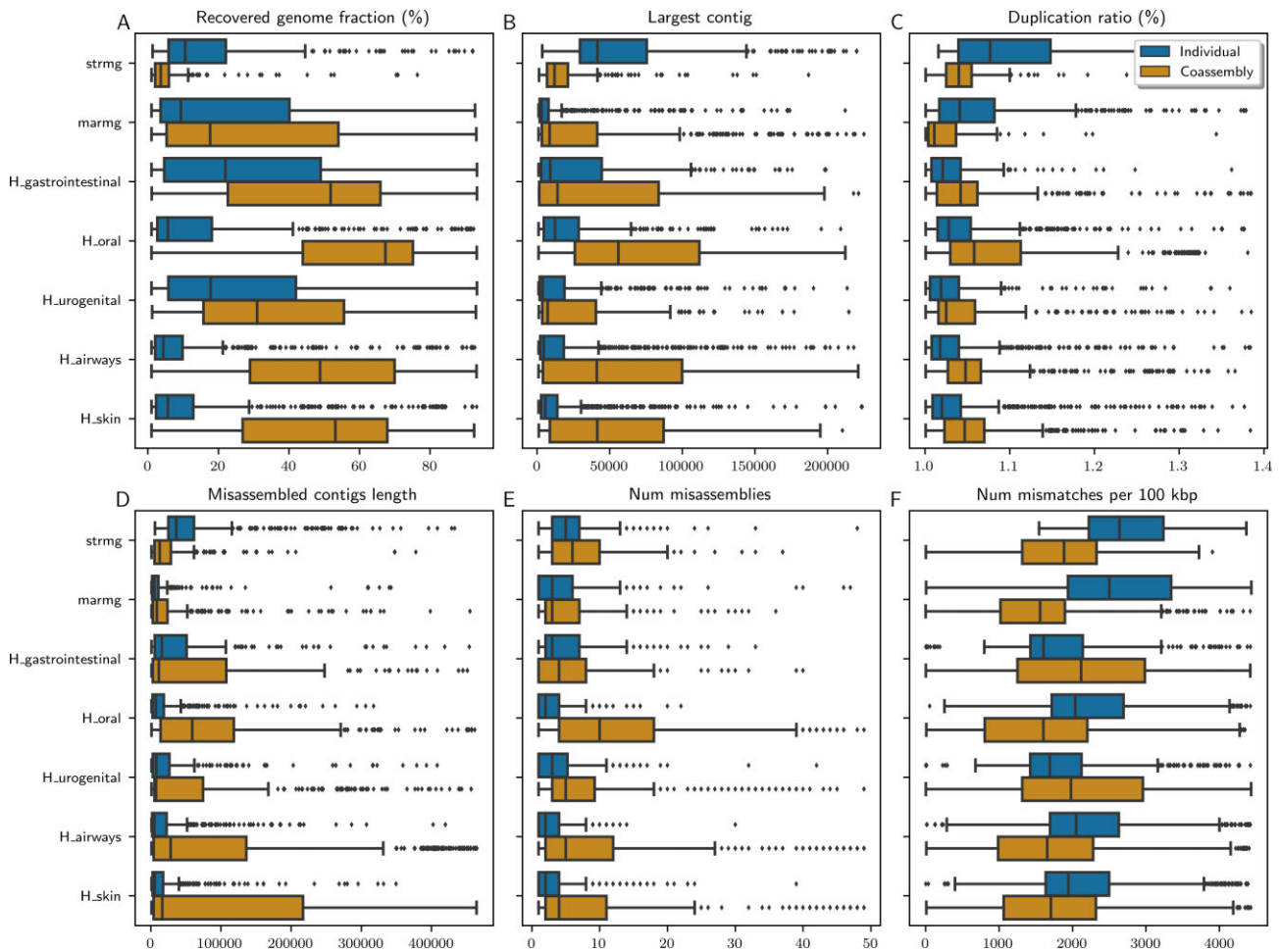
Madness and Marine datasets. We only used the latter 2 datasets for the binning assessment.

We used 6 metrics to evaluate assembly performance: percentage of recovered genome fraction, size of the largest contig, duplication ratio, length of misassembled contigs, number of mis-assemblies, and number of mismatches per 100,000 base pairs. High values for the first 2 metrics and low values for the last 4 indicate better performance. We observed a general trade-off between assembly completeness (represented by the first 2 metrics) and the number of errors in the assembly (represented by the last 4 metrics), shown in Fig. 5 (Supplementary Fig. S1). In most datasets, assemblies were more complete and contiguous, albeit with more errors when the coassembly strategy was used. The exception was the Strain Madness (“strmg”) dataset, for which the individual assembly was more complete and contiguous, albeit with more errors. This may be attributed to the high degree of strain/intraspecies diversity in that dataset [8]. A high degree of similarity between the related genomes likely confounds assembly algorithms, and pooling samples together may aggravate this effect [5].

To evaluate differences between binning strategies, we compared the number and quality of bins after refinement with the

DAS Tool. Bins generated with each approach were further dereplicated with dRep [40]. This is because the SASB strategy generates a set of bins for each sample, and datasets with similar composition will likely generate redundant bins, as there is no dereplication of bins between samples. Results varied significantly between the Marine and Strain Madness datasets. In both datasets, the mean bin score was the highest for the CACB strategy (Supplementary Fig. S2). However, in the Strain Madness dataset, CACB produced a significantly lower number of bins (33 compared with 259 and 215 generated with SASB and SACB, respectively), which did not occur in the Marine dataset. The performance of each binning tool is also variable between strategies and conditional on the characteristics of the original dataset, with no clear “winner,” and each tool favoring particular performance metrics, in agreement with results from the CAMI II challenge [8]. Tools like DAS Tool attempt to conciliate the output of multiple binning algorithms to generate a consensus output that theoretically outperforms each individual algorithm.

Since the binning performance is assessed as a proxy of the combination of quantity and quality of generated bins, rather than only one metric or the other, we calculated the cumulative bin score (the sum of scores of all bins) and the number

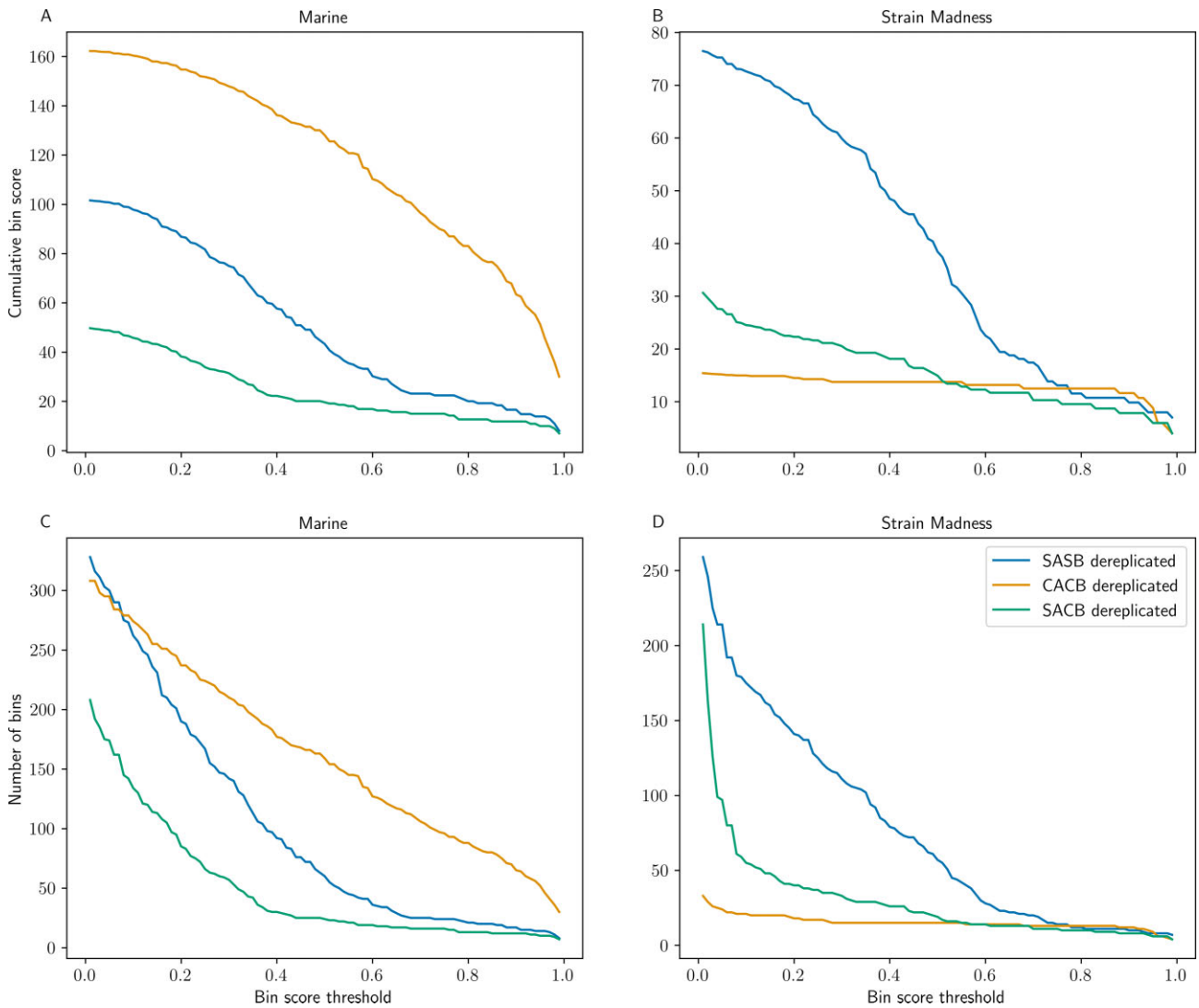


**Figure 5:** Differences between assembly strategies for each dataset. Each data point corresponds to a reference genome evaluated with the MetaQUAST tool. Data points above the 98th percentile were classified as outliers and removed from the figure to improve visualization. See Supplementary Fig. S1 for the full data. The title at the top of each panel indicates the plotted metric. Panels A and C show percentages along the x-axis, while the remainder show absolute values.

of bins above an increasing score threshold, shown in Fig. 6. The higher the threshold, the more significant the differences between the cumulative scores, as only bins with the highest quality compose the score. For the Marine dataset, we observed a higher score and a larger number of bins in the CACB strategy and the exact opposite in the Strain Madness dataset. In both datasets, there was a clear difference between SASB before dereplication and the other strategies, confirming that several highly similar samples produce redundant bins. That difference was also present in the SACB strategy, albeit not so pronounced (see Supplementary Figs. S2, S3 and S4 for the comparison of dereplicated and non-dereplicated data). This suggests that for both of these strategies, further dereplication is recommended [5]. Although the Strain Madness dataset shows fewer bins generated with CACB, a summary of the bins recovered with that dataset is displayed in Supplementary Table S1. The cumulative bin score for that strategy remained similar to SACB and SASB above the 0.8 score threshold, since there are fewer bins with a score lower than that. In that same dataset, SASB showed the best performance, although differences were small above the 0.8 threshold. In the Marine dataset, there were more pronounced differences between strategies. CACB produced the larger quantity and higher cumulative score of bins, followed by SASB and SACB.

In summary, our results indicate that, for most metagenomic analysis scenarios, coassembly followed by cobinning is recommended, assuming that samples are sourced from a similar environment or population. The exception to this is when there is a high level of intraspecies/strain-level diversity across samples, like in the Strain Madness dataset. In that scenario, single assembly followed by single binning is preferred, followed by dereplication of bins between samples. There is, however, a trade-off between the different approaches, as computational requirements are higher for the pooled strategies. Coassembly resulted in higher genome recovery fractions and larger contigs, although usually at the expense of a higher number of misassemblies and higher duplication ratio. When combining coassembly with cobinning, there is a remarkable improvement in the quantity and quality of bins generated for a diverse dataset (represented by the Marine dataset), whereas the difference was negligible in the Strain Madness dataset. Therefore, when deciding the assembly and binning strategy, it is important to consider the expected strain-level diversity and abundances of each individual genome, as the interaction between these factors is likely to limit the resolution of recovered bins. This is shown in the CAMI II challenge [8] (see Fig. 1G); genomes with low strain diversity (i.e., are less than 95% similar to any other genome) have a higher correlation between sequencing coverage and recovered fraction than common genomes ( $\geq 95\%$





**Figure 6:** Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets. Lines show the cumulative bin score (A, B) and number of bins (C, D) along the y-axis, for bins above a certain score threshold (x-axis). Left column shows Marine dataset, and right column shows Strain Madness dataset.

similar to other genomes in the sample), although many times, sequencing coverage was not all correlated with genome recovery fraction, especially for smaller bins that represent plasmids or circular elements.

## Availability and Future Directions

Metaphor is available through Bioconda [45], a popular repository of bioinformatics software. It can be installed with a single command from the conda package manager [46] or from source using pip, the Python package manager. The installation of all third-party software used by Metaphor is handled automatically by Snakemake and conda. It can be easily deployed in different computing environments, such as high-performance computing clusters and cloud instances, due to Snakemake's support of execution profiles. Metaphor is developed with documented best practices in workflow development [6, 47], striving for reproducibility and transparency of its results. Data used for testing Metaphor's installation (see documentation for details) are available from GitHub at <https://github.com/vinisalazar/mg-example-data>. These data are a subset of the CAMI I challenge data [36] that

are reduced in size in order to run test commands in a reasonable time.

The workflow may be extended to support downstream tools such for genome analysis such as GTDB-Tk, CheckM, and dRep. This may help with further improvement of strain-level resolution in bins; there are a number of strategies for that, such as identification of misassembled contigs or using the assembly graph for variant detection [48, 49]. New functionality may also be added for the identification of eukaryotic and viral contigs; Metaphor would benefit from new third-party software to facilitate the generation of non-prokaryotic bins in the near future. The output of Metaphor's "annotation" module is suitable for *ad hoc* identification of eukaryotic and viral contigs; after selecting the annotated prokaryotic contigs, it is possible to filter them out, leaving unannotated (putative) eukaryotic and viral contigs. These can then be used as input for a eukaryotic or viral discovery pipeline [50–52], but this process could be further improved by facilitating the use of custom reference databases in the annotation module. This can also be done directly with the output of the assembly module, but in that case, there will not be any screening for prokaryotic contigs. One drawback of this approach is that

each eukaryotic/viral discovery pipeline has specific input data formatting requirements. This integration with non-prokaryotic pipelines, along with support for long reads, are priority features to be added to future major versions of Metaphor.

## Availability of Source Code and Requirements

Project name: Metaphor  
 Project homepage: <https://github.com/vinisalazar/metaphor>  
 Documentation: <https://metaphor-workflow.readthedocs.io/>  
 Operating system(s): Linux, Mac OS (Intel)  
 Programming language: Snakemake (Python 3)  
 Other requirements: Conda, Snakemake v7 or higher, Python 3.7 or higher.  
 License: MIT  
 RRID number: SCR\_023701

## Additional Files

**Supplementary Fig. S1.** Differences between assembly strategies across datasets. Same data as Fig. 5, but including outliers.

**Supplementary Fig. S2.** Boxplot of bin scores across different strategies. Each data point is a genome bin, and the y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colors represent different strategies. Numbers underneath each bar show the number of data points for that bar. Bin sets were dereplicated with dRep.

**Supplementary Fig. S3.** Boxplot of bin scores across different strategies for non-dereplicated data. Same as Fig. S2, but with non-dereplicated data. Each data point is a genome bin, and the y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colors represent different strategies. Numbers underneath each bar show the number of data points for that bar.

**Supplementary Fig. S4.** Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets. Solid lines show the same data as Fig. 6, and dashed lines show data based on bins prior to dereplication with dRep.

**Supplementary Table S1.** Summary of genome bins recovered from the Strain Madness dataset, CACB strategy. “Bin ID” indicates the binning algorithm that generated the bin, “Bin score  $S_b$ ” is the relative bin score, SCG refers to single-copy gene in “SCG completeness,” and “SCG redundancy,” “FastANI reference,” and “GTDB classification” refer to the reference genome and corresponding taxonomy assignment. Taxonomy determined with GTDB-Tk v2.3.0, reference data r214 [44].

## Abbreviations

CACB: coassembly, cobinning; CAMI II: Second Critical Assessment for Metagenome Interpretation; GRM: genome-resolved metagenomics; MAG: metagenome-assembled genome; NCBI: The National Center for Biotechnology Information; QC: quality control; SACB: single assembly, cobinning SASB: single assembly, single binning; SCG: single-copy marker gene.

## Data Availability

This work uses data from the CAMI II challenge, available from [53–55]. Analysis code used in this article is available from [56]. Snapshots of our code and other data further supporting this work are openly available from the GigaDB repository [57].

## Competing Interests

The authors declare they have no competing interests.

## Funding

V.W.S. is funded by a Melbourne Research Scholarship from the University of Melbourne. V.R.M. is funded by an Australian Research Council DECRA Fellowship DE220100965. K.-A.L.C. was supported in part by the National Health and Medical Research Council (NHMRC) Career Development fellowship (GNT1159458). This research was also funded by the Australian Research Council project DP200101613.

## Authors’ Contributions

V.W.S.: conceptualization, data curation, methodology, investigation, software, writing—original draft. B.S., M.M.Q., R.T., E.T.: conceptualization, writing—review and editing. V.R.M., H.V., K.-A.L.C.: conceptualization, supervision, funding acquisition, writing—review and editing.

## Acknowledgments

Metaphor benefited strongly from experience gained developing MetaGenePipe [58], a Cromwell-based workflow for assembly and annotation of metagenomic contigs. This research was supported by the University of Melbourne’s Research Computing Services and the Petascale Campus Initiative. We thank Francesco Ricci and Uthpala Pushpakumara for providing datasets for early trials of Metaphor and colleagues from the Lê Cao lab for sharing their feedback.

## References

- Almeida A, Nayfach S, Boland M, et al. A unified catalog of 204,938 reference genomes from the human gut microbiome. *Nat Biotech* 2021;39(1):105–14.
- Parks DH, Rinke C, Chuvochina M, et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nat Microbiol* 2017;2(11):1533–42.
- Tully BJ, Graham ED, Heidelberg JF. The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. *Sci Data* 2018;5(1):170203.
- Setubal JC. Metagenome-assembled genomes: concepts, analogies, and challenges. *Biophys Rev* 2021;13(6):905–9.
- Nelson WC, Tully BJ, Mobberley JM. Biases in genome reconstruction from metagenomic data. *PeerJ* 2020;8:e10119.
- Reiter T, Brooks PT, Irber L, et al. Streamlining data-intensive biology with workflow systems. *Gigascience* 2021;10(1):g1aa140.
- Quince C, Walker AW, Simpson JT, et al. Shotgun metagenomics, from sampling to analysis. *Nat Biotech* 2017;35(9):833–44.
- Meyer F, Fritz A, Deng ZL, et al. Critical assessment of metagenome interpretation: the second round of challenges. *Nat Methods* 2022;19(4):429–40.
- Delgado LF, Andersson AF. Evaluating metagenomic assembly approaches for biome-specific gene catalogues. *Microbiome* 2022;10(1):72.
- Rohart F, Gautier B, Singh A, et al. mixOmics: An R package for ‘omics feature selection and multiple data integration. *PLoS Comput Biol* 2017;13(11):e1005752.

11. McMurdie PJ, Holmes S. phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* 2013;8(4):e61217.
12. Eren AM, Kiefl E, Shaiber A, et al. Community-led, integrated, reproducible multi-omics with anvio. *Nat Microbiol* 2020;6(1):3–6.
13. Mölder F, Jablonski KP, Letcher B, et al. Sustainable data analysis with Snakemake. *F1000 Res* 2021;10:33.
14. Chen S, Zhou Y, Chen Y, et al. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* 2018;34(17):i884–90.
15. Andrews S. FastQC A Quality Control tool for High Throughput Sequence Data. Online Resource 2020. <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>. Accessed 19 November 2021.
16. Ewels P, Magnusson M, Lundin S, et al. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* 2016;32(19):3047–8.
17. Li D, Liu CM, Luo R. et al. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 2015;31(10):1674–6.
18. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* 2016;32(7):1088–90.
19. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018;34(18):3094–100.
20. Danecek P, Bonfield JK, Liddle J, et al. Twelve years of SAMtools and BCFtools. *Gigascience* 2021;10(2):giab008.
21. Hyatt D, Chen GL, Locascio PF, et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* 2010;11:119.
22. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. *Nat Methods* 2014;12:59–60.
23. Galperin MY, Wolf YI, Makarova KS, et al. COG database update: focus on microbial diversity, model organisms, and widespread pathogens. *Nucleic Acids Res* 2021;49(D1):D274–81.
24. Seemann T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* 2014;30(14):2069.
25. Nissen JN, Johansen J, Allesøe RL, et al. Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotech* 2021;39:555–60.
26. Kang DD, Li F, Kirton E, et al. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* 2019;2019(7):1–13.
27. Alneberg J, Bjarnason BS, de Bruijn I, et al. Binning metagenomic contigs by coverage and composition. *Nat Methods* 2014;11(11):1144–6.
28. Sieber CMK, Probst AJ, Sharrar A, et al. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nat Microbiol* 2018;3(7):836–43.
29. Churchward B, Millet M, Bihouée A, et al. MAGNETO: an automated workflow for genome-resolved metagenomics. *mSystems* 2022;0(0):e00432–22.
30. Kieser S, Brown J, Zdobnov EM, et al. ATLAS: a Snake-make workflow for assembly, annotation, and genomic binning of metagenome sequence data. *BMC Bioinform* 2020;21(1):1–8.
31. Uritskiy GV, DiRuggiero J, Taylor J. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome* 2018;6(1):158.
32. Krakau S, Straub D, Gourel H, et al. nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning. *NAR Genomics Bioinform* 2022;4(1):lqac007.
33. McKinney W. pandas: a foundational Python library for data analysis and statistics. *Python High Performance Sci Comput* 2011;14(9):1–9.
34. Hunter JD. Matplotlib: A 2D graphics environment. *Comput Sci Eng* 2007;9(3):90–5.
35. Waskom M, Botvinnik O, Ostblom J, et al. Seaborn v0.10.0. 2020. <https://seaborn.pydata.org/>. Accessed 1 February 2023.
36. Sczyrba A, Hofmann P, Belmann P, et al. Critical assessment of metagenome interpretation - a benchmark of metagenomics software. *Nat Methods* 2017;14(11):1063–71.
37. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nat Methods* 2021;18:1161–8.
38. Salazar VW. Metaphor's documentation. 2023. <https://metaphor-workflow.readthedocs.io/en/latest/>. Accessed 1 February 2023.
39. Fritz A, Hofmann P, Majda S, et al. CAMISIM: simulating metagenomes and microbial communities. *Microbiome* 2019;7(1):17.
40. Olm MR, Brown CT, Brooks B, et al. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *ISME J* 2017;11(12):2864–8.
41. Jain C, Rodriguez-R LM, Phillippy AM, et al. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. *Nat Commun* 2018;9(1):5114.
42. Lafayette L, Wiebelt B. Spartan and NEMO: two HPC-cloud hybrid implementations. In: 2017 IEEE 13th International Conference on e-Science Auckland, New Zealand: e-Science, 2017, 458–9.
43. Parks DH, Imelfort M, Skennerton CT, et al. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* 2015;25(7):1043–55.
44. Chaumeil PA, Mussig AJ, Hugenholtz P, et al. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. *Bioinformatics* 2019;36(6):1925–7.
45. Grüning B, Dale R, Sjödin A, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods* 2018;15(7):475–6.
46. Inc A. Conda – Conda documentation. 2023. <https://docs.conda.io/en/latest/>. Accessed 1 February 2023.
47. Jackson M, Kavoussanakis K, Wallace EWJ. Using prototyping to choose a bioinformatics workflow management system. *PLoS Comput Biol* 2021;17(2):e1008622.
48. Lai S, Pan S, Sun C, et al. metaMIC: reference-free misassembly identification and correction of de novo metagenomic assemblies. *Genome Biol* 2022;23(1):242.
49. Quince C, Nurk S, Raguideau S, et al. STRONG: metagenomics strain resolution on assembly graphs. *Genome Biol* 2021;22(1):214.
50. Pandolfo M, Telatin A, Lazzari G, et al. MetaPhage: an automated pipeline for analyzing, annotating, and classifying bacteriophages in metagenomics sequencing data. *mSystems* 2022;7(5):e00741–22.
51. Karlicki M, Antonowicz S, Karnkowska A. Tiara: deep learning-based classification system for eukaryotic sequences. *Bioinformatics* 2022;38(2):344–50.
52. Pronk L, Medema M. Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structure. *Microbial Genomics* 2021;8(5). <https://doi.org/10.1099%2Fmgen.0.000823>.
53. Fritz A, Mächler M, McHardy A, et al. CAMI 2—Challenge Datasets. 2021. <https://repository.publisso.de/resource/frl:6425521>. Accessed 28 June 2023.

54. Fritz A, McHardy A, Lesker T, et al. CAMI 2—Multisample Benchmark Dataset of Human Microbiome Project. 2021. <https://repository.publisso.de/resource/frl:6425518>. Accessed 28 June 2023.
56. Salazar V. Metaphor supplementary material. 2023. <https://github.com/vinisalazar/manuscript-notebooks-metaphor>. Accessed 28 June 2023.
57. Vinicius SW, Babak S, Maria QD, et al. Supporting data for “Metaphor—A Workflow for Streamlined Assembly and Binning of Metagenomes.” GigaScience Database. 2023. <http://dx.doi.org/10.5524/102408>.
58. Shaban B, Quiroga MdM, Turnbull R, et al. MetaGenePipe: an automated, portable pipeline for contig-based functional and taxonomic analysis. *J Open Source Softw* 2023;8(82):4851.