GigaScience

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

--Manuscript Draft--

| | |
|---|---|
| **Manuscript Number:** | GIGA-D-23-00067R1 |
| **Full Title:** | Metaphor - A workflow for streamlined assembly and binning of metagenomes |
| **Article Type:** | Technical Note |
| **Funding Information:** | Australian Research Council (DE220100965) — Dr Vanessa Rossetto Marcelino<br>National Health and Medical Research Council (GNT1159458) — Prof. Kim-Anh Le Cao<br>Australian Research Council (DP200101613) — A/Prof Heroen Verbruggen |
| **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 analyse 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 customisable 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. Both 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. |
| **Corresponding Author:** | Kim-Anh Le Cao<br>The University of Melbourne<br>Parkville, VIC AUSTRALIA |
| **Corresponding Author Secondary Information:** | |
| **Corresponding Author's Institution:** | The University of Melbourne |
| **Corresponding Author's Secondary Institution:** | |
| **First Author:** | Vinícius W. Salazar |
| **First Author Secondary Information:** | |
| **Order of Authors:** | Vinícius W. Salazar<br>Babak Shaban<br>Maria del Mar Quiroga<br>Robert Turnbull<br>Edoardo Tescari<br>Vanessa Rossetto Marcelino<br>Heroen Verbruggen<br>Kim-Anh Le Cao |

| | |
|---|---|
| **Order of Authors Secondary Information:** | |
| **Response to Reviewers:** | Dear editor,<br>See attached files in the submission |
| **Additional Information:** | |
| **Question** | **Response** |
| Are you submitting this manuscript to a special series or article collection? | No |
| **Experimental design and statistics**<br><br>Full details of the experimental design and statistical methods used should be given in the Methods section, as detailed in our Minimum Standards Reporting Checklist. Information essential to interpreting the data presented should be made available in the figure legends.<br><br>Have you included all the information requested in your manuscript? | Yes |
| **Resources**<br><br>A description of all resources used, including antibodies, cell lines, animals and software tools, with enough information to allow them to be uniquely identified, should be included in the Methods section. Authors are strongly encouraged to cite Research Resource Identifiers (RRIDs) for antibodies, model organisms and tools, where possible.<br><br>Have you included the information requested as detailed in our Minimum Standards Reporting Checklist? | Yes |
| **Availability of data and materials**<br><br>All datasets and code on which the conclusions of the paper rely must be either included in your submission or deposited in publicly available repositories (where available and ethically appropriate), referencing such data using a unique identifier in the references and in | Yes |

| | |
|---|---|
| the “Availability of Data and Materials” section of your manuscript.<br><br>Have you have met the above requirement as detailed in our Minimum Standards Reporting Checklist? | |

*GigaScience*, 2023, 1–16

doi: xx.xxxx/xxxx
**Manuscript in Preparation**
**Technical Note**

TECHNICAL NOTE

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

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

[1]Melbourne Integrative Genomics, School of Mathematics & Statistics, University of Melbourne, Parkville, Victoria, Australia and [2]Melbourne Data Analytics Platform (MDAP), University of Melbourne, Parkville, Victoria, Australia and [3]Department of Molecular and Translational Sciences, Monash University, Clayton, Victoria, Australia and [4]Centre for Innate Immunity and Infectious Diseases, Hudson Institute of Medical Research, Clayton, Victoria, Australia and [5]School of BioSciences, University of Melbourne, Parkville, Victoria, Australia and [6]Department of Microbiology and Immunology, The University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Parkville, Victoria, Australia.

[*]kimanh.lecao@unimelb.edu.au
[†]Deceased.
[§]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 analyse 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 customisable 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.

**Key words**: 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)

**Compiled on:** June 5, 2023.
Draft manuscript prepared by the author.

[1, 2, 3, 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 analysing 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 heterogeneous data sources, have dozens of customisable parameters, and depend on several specialised 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 catalogue 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 hypothesised that pooled assembly and binning may lead to improved results when analysing 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 provides taxonomic and functional abundance data for quantitative metagenome analyses. Our software advances existing metagenomic pipelines by combining two 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 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, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] and custom scripts that connect different parts of the workflow, listed on Table 2.

The workflow consists of six 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 show on Fig 1.

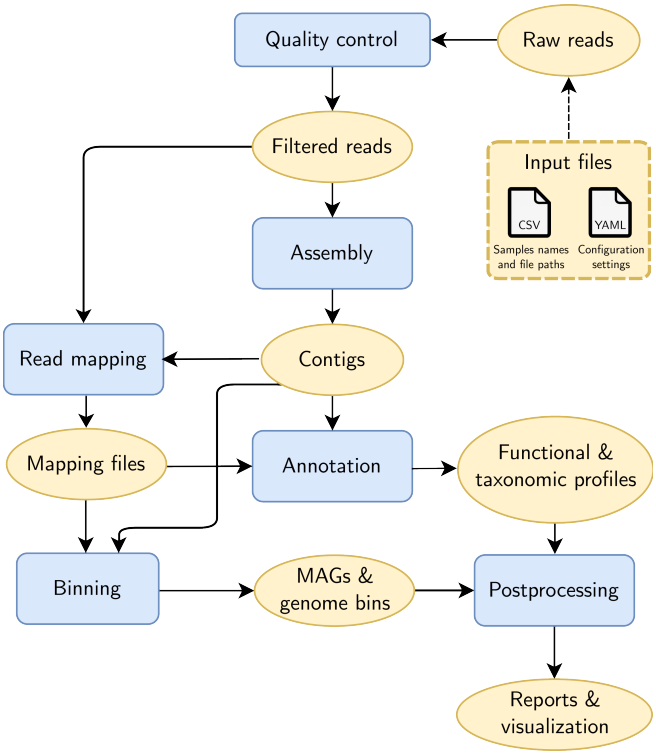


**Figure 1. Simplified workflow diagram.** Workflow modules are represented by rectangular blue shapes and data files are represented by oval yellow shapes, except for entrypoint files shown in a dashed yellow rectangle. Arrows indicate input and output of data between modules.

The choice of bioinformatics tools was informed by the results of the 2nd 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 two files as input: a tab-delimited file con-

**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 catalogue | ✓ | | | ✓ | ✓ |
| 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] |
| | Assembly report and plots | Metaphor script* |
| Mapping | Map reads | MiniMap2 [19] |
| | Sort and index mapped reads | Samtools [20] |
| Annotation | Prediction of coding sequences from contigs | Prodigal[21] |
| | Annotation of coding sequences | Diamond, NCBI COG [22, 23] |
| | Annotation of MAGs | Prokka [24] |
| | Annotation report and plots | Metaphor script* |
| Binning | 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] |
| | Binning report and plots | Metaphor script* |
| Postprocessing | Concatenate benchmarks | Metaphor script* |
| | Plot benchmarks | Metaphor script* |

* External libraries used in Metaphor scripts: [33, 34, 35].

**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 | 1394 |
| Human Genital | h_urogenital | 9 | 39 | 1394 |
| Human Gut | h_gastrointestinal | 10 | 44 | 1057 |
| Human Oral | h_oral | 10 | 43 | 1057 |
| Human Skin | h_skin | 10 | 44 | 1394 |

**Table 4.** Output files for each strategy. If only one dataset/group is being analysed, 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<br>Sample_1.fastq<br>Sample_2.fastq | Sample_0_contigs.fasta<br>Sample_1_contigs.fasta<br>Sample_2_contigs.fasta | Sample_0_bins/<br>Sample_1_bins/<br>Sample_2_bins/ |
| SACB | Single assembly, Cobinning | Sample_0.fastq<br>Sample_1.fastq<br>Sample_2.fastq | Sample_0_contigs.fasta<br>Sample_1_contigs.fasta<br>Sample_2_contigs.fasta | Cobinning_bins/ |
| CACB | Coassembly, Cobinning | Sample_0.fastq<br>Sample_1.fastq<br>Sample_2.fastq | Coassembly_contigs.fasta | Cobinning_bins/ |

taining 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 analysed datasets from CAMI II [8]. 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 five 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 datasets: 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; coassembly, cobinning ('CACB'), where all samples from the dataset were 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 analyse 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 results 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 which 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 (HPC) 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.

### Reconstruction of metagenome-assembled genomes

Metaphor produces genome bins generated with three tools: Vamb, MetaBAT2 and CONCOCT [25, 26, 27] that are refined with DAS Tool [28]. 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, non-redundant bins remain; the bin score ($S_b$) increases with the number of SCGs and

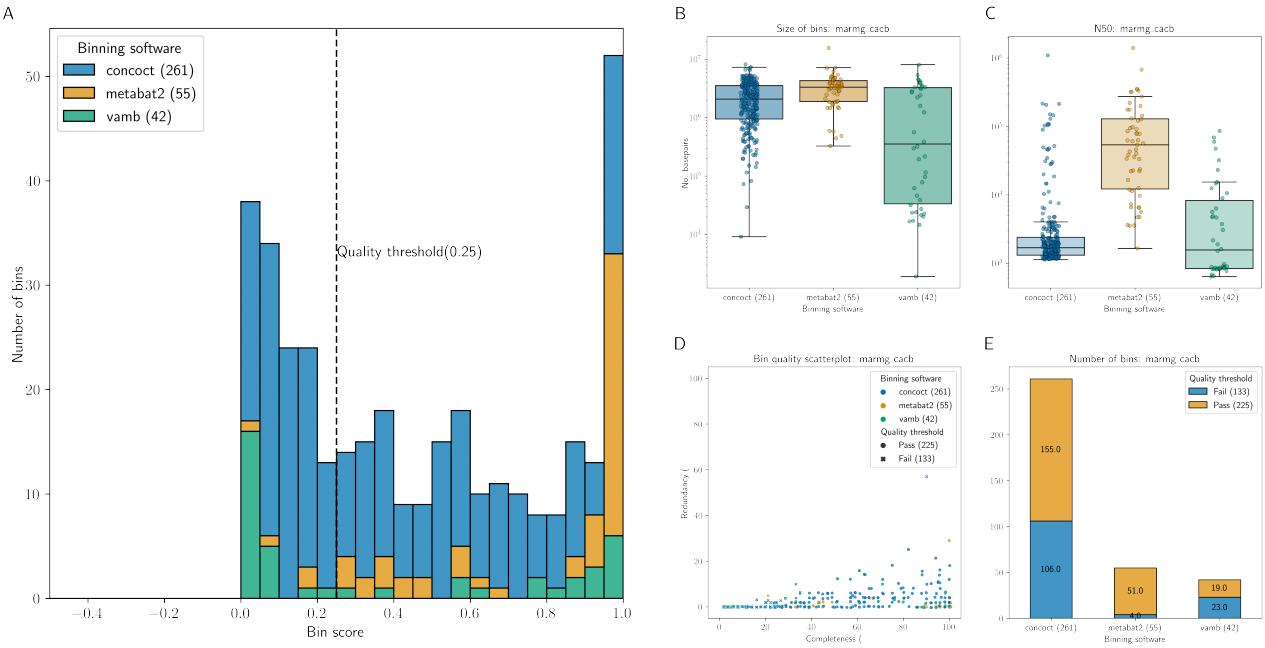


**Figure 2. Binning report generated by Metaphor for the CAMI II Marine metagenome dataset processed with the 'CACB' (coassembly, cobinning) setting.** Panel **A** shows a stacked histogram of the distribution of bin scores, with the defined quality threshold highlighted as a dashed line. Panels **B** and **C** show, respectively, the size (in base pairs) and N50 of bins. The Y-axis is in log-scale. Panel **D** shows a scatterplot of completeness and redundancy for each bin. Colours 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 Panel **A**). Panel **E** shows how many bins passed or failed the quality threshold for each binning tool.

decreases with duplicate SCGs per bin. Please refer to [28], Figure 1 and Equation 1 for an overview of the DAS algorithm and the formula to determine the bin score, respectively. The input for each binning tool differs slightly, but they all rely on the catalogue 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 one is generated if cobinning is performed), which highlights three key metrics: completeness, redundancy, and bin score. The first two 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 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 detected in the previous round. Abundances for each feature are calculated based on the coverage of all coding sequences which align to that feature. Fig 3 illustrates the profile visualisations offered by Metaphor: a heatmap of COG categories for the functional profile and a stacked barplot for the most abundant taxa (for the latter, one 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, 15, 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 two 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 three different strategies: single assembly and single binning ('SASB'), single assembly and cobinning ('SACB'), and coassembly and cobinning ('CACB'), see Table 4 and Section 'As-

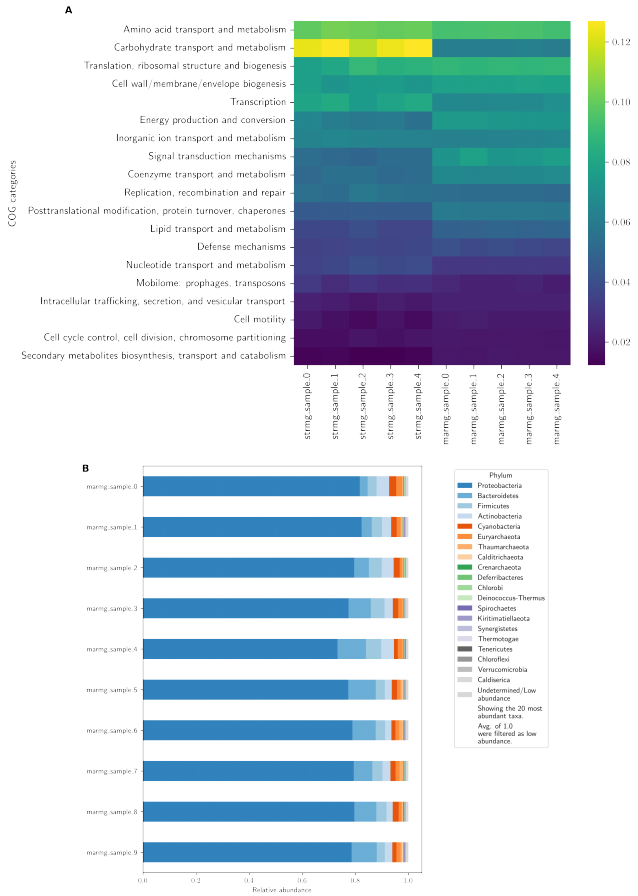


**Figure 3. Annotation plots generated by Metaphor on the Strain Madness ('strmg') and the Marine ('marmg') datasets.** Panel **A** displays the functional profile as a heatmap of the relative abundance of functional COG categories (Y-axis) across samples (X-axis) for five samples from Strain Madness and Marine datasets. Panel **B** displays 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.

sessment on CAMI II synthetic datasets' for details. For assembly, we used the five different groups in the Human Microbiome dataset along with the Strain Madness and Marine datasets. We only used the latter two datasets for the binning assessment.

We used six metrics to evaluate assembly performance: percentage of recovered genome fraction, size of the largest contig, duplication ratio, length of misassembled contigs, number of misassemblies, and number of mismatches per 100 thousand base pairs. High values for the first two metrics and low values for the last four indicate better performance. We observed a general trade-off between assembly completeness (represented by the first two metrics), and the number of errors in the assembly (represented by the last four metrics) (Figure 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 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 (**Figure S3**). 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), which did not occur in the Marine dataset. The performance of each binning tool is also variable between strategies and is conditional on the characteristics of the original dataset, with no clear "winner", and each tool favouring particular performance metrics, in agreement with results from the 2nd CAMI Challenge [8]. Tools like DAS Tool attempt to conciliate the output of multiple binning algorithms to generate a consensus output which 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 of bins above an increasing score threshold, shown on 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 **Figure 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 on **??**. 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 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), where 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 Figure 1g); genomes with low strain diversity (*i.e.* are less than 95% similar with any other genomes) have higher correlation between sequencing coverage and recovered fraction than common genomes ($\geq$ 95% similar to other genomes in the sample), although many times sequencing coverage was not all correlated with genome recovery fraction, specially for smaller bins that represent plasmids or circular elements.

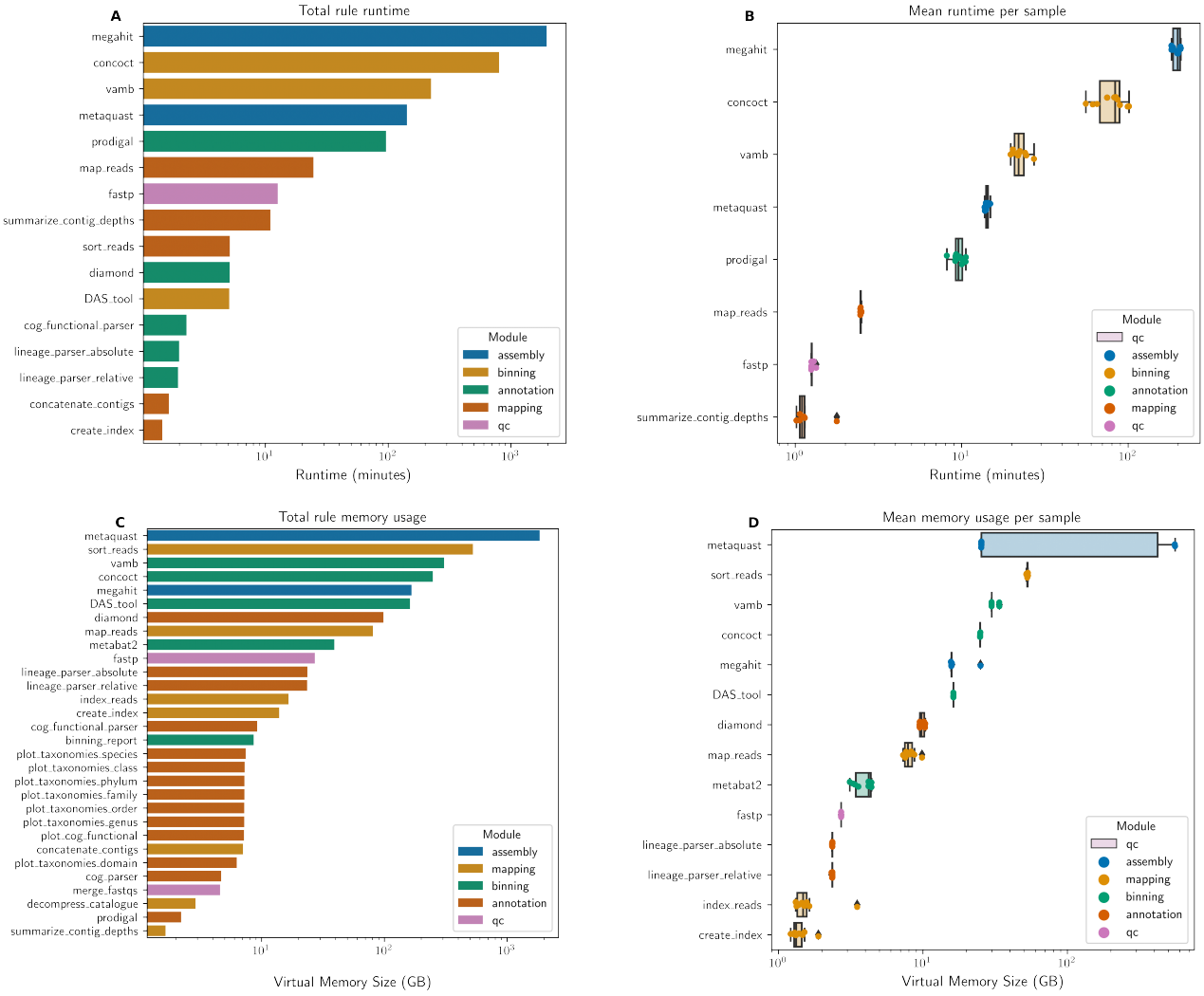


**Figure 4. Performance metrics report generated by Metaphor on the Marine dataset processed with the SASB strategy.** Total runtime per rule (**A**), mean runtime per sample (**B**), total memory usage per rule (**C**), and mean memory usage per sample (**D**). X-axis is in log format. Cutoffs are applied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

## 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 the testing Metaphor's installation (see documentation for details) is available from GitHub at https://github.com/vinisalazar/mg-example-data. This data is a subset of the CAMI I challenge data [36] that is 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, 51, 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 won't 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 checklist

Project name: Metaphor
Project home page: https://github.com/vinisalazar/metaphor
Documentation: https://metaphor-workflow.readthedocs.io/

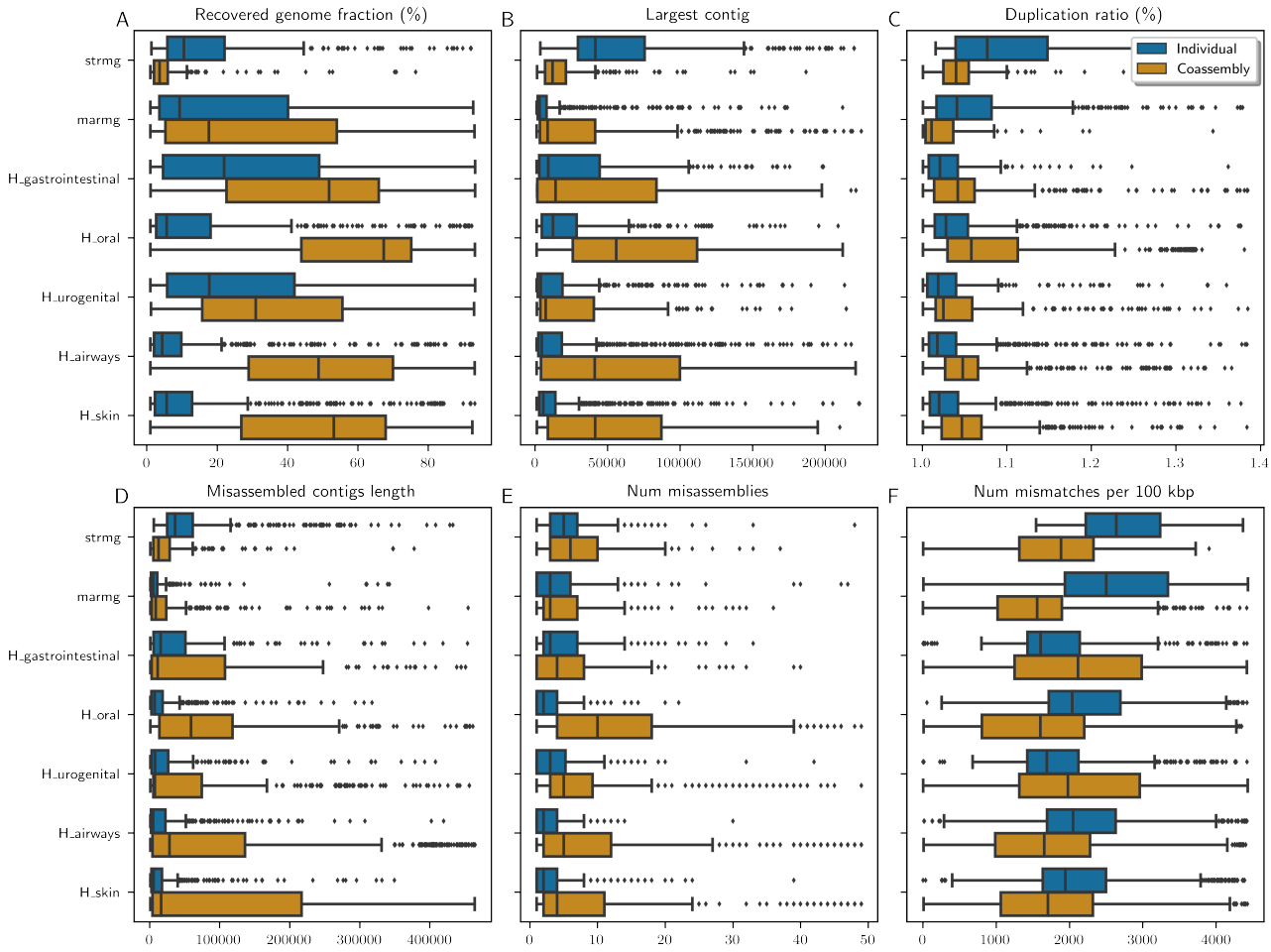


**Figure 5. Differences between assembly strategies for each dataset.** Each data point corresponds to a reference genome evaluated with the MetaQUAST tool. Data points above the 98th percentile were classified as outliers and removed from the figure to improve visualisation. See **Figure S1** for the 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.

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

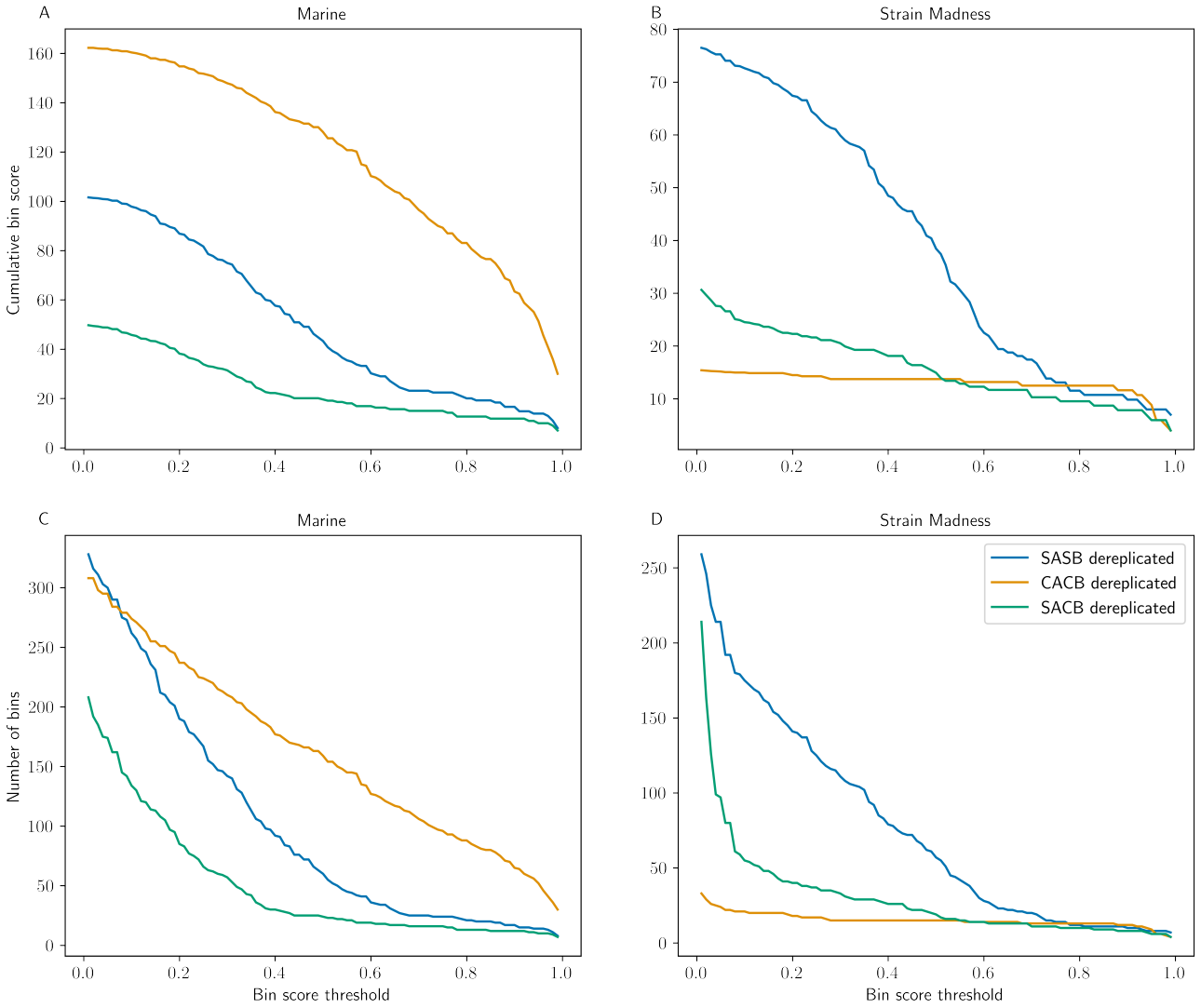


**Figure 6. Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets.** Lines show the cumulative bin score (**A** and **B**) and number of bins (**C** and **D**) along the Y-axis, for bins above a certain score threshold (X-axis). Left column shows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

## Declarations

The authors declare they have no competing interests.


## Funding

VWS is funded by a Melbourne Research Scholarship from The University of Melbourne. VRM is funded by an Australian Research Council DECRA Fellowship DE220100965. KALC 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.


## Author's Contributions

VWS - Conceptualization, Data curation, Methodology, Investigation Software, Writing - original draft; BS, MMQ, RT, ET - Conceptualization, Writing - review and editing; VRM, HV, KALC - Conceptualization, Supervision, Funding Acquisition, Writing - review and editing.


## Acknowledgments

Metaphor benefited strongly from experience gained developing MetaGenePipe [53], 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


1. Almeida A, Nayfach S, Boland M, Strozzi F, Beracochea M, Shi ZJ, et al. A unified catalog of 204,938 reference genomes from the human gut microbiome. Nature Biotechnology 2021 Jan;39(1):105–114. https://www.nature.com/articles/s41587-020-0603-3, number: 1 Publisher: Nature Publishing Group.
2. Parks DH, Rinke C, Chuvochina M, Chaumeil PA, Woodcroft BJ, Evans PN, et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nature Microbiology 2017;2(11):1533–1542. http://dx.doi.org/10.1038/s41564-017-0012-7, publisher: Springer US ISBN: 4156401700127.
3. Tully BJ, Graham ED, Heidelberg JF. The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. Scientific Data 2018 Jan;5(1):170203. https://www.nature.com/articles/sdata2017203, bandiera_abtest: a Cc_license_type: cc_publicdomain Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Bioinformatics;Genome;Metagenomics;Water microbiology Subject_term_id: bioinformatics;genome;metagenomics;water-microbiology.
4. Setubal JC. Metagenome-assembled genomes: concepts, analogies, and challenges. Biophysical Reviews 2021 Dec;13(6):905–909. https://doi.org/10.1007/s12551-021-00865-y.
5. Nelson WC, Tully BJ, Mobberley JM. Biases in genome reconstruction from metagenomic data. PeerJ 2020 Oct;8:e10119. https://peerj.com/articles/10119.
6. Reiter T, Brooks PT, Irber L, Joslin SEK, Reid CM, Scott C, et al. Streamlining data-intensive biology with workflow systems. GigaScience 2021 Jan;10(1). https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giaa140/6092773.
7. Quince C, Walker AW, Simpson JT, Loman NJ, Segata N. Shotgun metagenomics, from sampling to analysis. Nature Biotechnology 2017 Sep;35(9):833–844. https://www.nature.com/articles/nbt.3935, bandiera_abtest: a Cg_type: Nature Research Journals Number: 9 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Microbial communities;Computational biology and bioinformatics;Metagenomics Subject_term_id: communities;computational-biology-and-bioinformatics;metagenomics.
8. Meyer F, Fritz A, Deng ZL, Koslicki D, Lesker TR, Gurevich A, et al. Critical Assessment of Metagenome Interpretation: the second round of challenges. Nature Methods 2022 Apr;19(4):429–440. https://www.nature.com/articles/s41592-022-01431-4, number: 4 Publisher: Nature Publishing Group.
9. Delgado LF, Andersson AF. Evaluating metagenomic assembly approaches for biome-specific gene catalogues. Microbiome 2022 May;10(1):72. https://doi.org/10.1186/s40168-022-01259-2.
10. Rohart F, Gautier B, Singh A, Lê Cao KA. mixOmics: An R package for 'omics feature selection and multiple data integration. PLoS Computational Biology 2017;.
11. McMurdie PJ, Holmes S. phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. PLOS ONE 2013 Apr;8(4):e61217. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0061217, publisher: Public Library of Science.
12. Eren AM, Kiefl E, Shaiber A, Veseli I, Miller SE, Schechter MS, et al. Community-led, integrated, reproducible multi-omics with anvi'o. Nature Microbiology 2020 Dec;6(1):3–6. https://www.nature.com/articles/s41564-020-00834-3.
13. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with Snakemake. F1000 Research 2021 Apr;https://f1000research.com/articles/10-33, type: article.
14. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 2018 Sep;34(17):i884–i890. https://doi.org/10.1093/bioinformatics/bty560.
15. Andrews S. FastQC A Quality Control tool for High Throughput Sequence Data. Online resource 2020 Jan;https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
16. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: Summarize analysis results for multiple tools and samples in a single report. Bioinformatics 2016;.
17. Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 2015 May;31(10):1674–1676. https://doi.org/10.1093/bioinformatics/btv033.
18. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics 2016 Apr;32(7):1088–1090. https://doi.org/10.1093/bioinformatics/btv697.
19. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 2018 Sep;34(18):3094–3100. https://doi.org/10.1093/bioinformatics/bty191.
20. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Twelve years of SAMtools and BCFtools. GigaScience 2021 Feb;10(2):giab008.
21. Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics 2010;11:119. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2848648&tool=pmcentrez&rendertype=abstract, iSBN: 1471-2105 (Electronic)\r1471-2105 (Linking).

22. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nature Methods 2014;https://github.com/bbuchfink/diamond.
23. Galperin MY, Wolf YI, Makarova KS, Vera Alvarez R, Landsman D, Koonin EV. COG database update: focus on microbial diversity, model organisms, and widespread pathogens. Nucleic Acids Research 2021 Jan;49(D1):D274–D281.
24. Seemann T. Prokka: Rapid prokaryotic genome annotation. Bioinformatics 2014;30(14).
25. Nissen JN, Johansen J, Allesøe RL, Sønderby CK, Armenteros JJA, Grønbech CH, et al. Improved metagenome binning and assembly using deep variational autoencoders. Nature Biotechnology 2021 Jan;http://www.nature.com/articles/s41587-020-00777-4, publisher: Nature Research.
26. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, 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, Schirmer M, Quick J, Ijaz UZ, et al. Binning metagenomic contigs by coverage and composition. Nature Methods 2014 Nov;11(11):1144–1146. https://www.nature.com/articles/nmeth.3103, bandiera_abtest: a Cg_type: Nature Research Journals Number: 11 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Genome informatics;Machine learning;Metagenomics Subject_term_id: genome-informatics;machine-learning;metagenomics.
28. Sieber CMK, Probst AJ, Sharrar A, Thomas BC, Hess M, Tringe SG, et al. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. Nature Microbiology 2018 Jul;3(7):836–843. https://www.nature.com/articles/s41564-018-0171-1, number: 7 Publisher: Nature Publishing Group.
29. Churcheward B, Millet M, Bihouée A, Fertin G, Chaffron S. MAGNETO: An Automated Workflow for Genome-Resolved Metagenomics. mSystems 2022 Jun;0(0):e00432–22. https://journals.asm.org/doi/10.1128/msystems.00432-22, publisher: American Society for Microbiology.
30. Kieser S, Brown J, Zdobnov EM, Trajkovski M, McCue LA. ATLAS: a Snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data. BMC Bioinformatics 2020 Dec;21(1):1–8. https://link.springer.com/article/10.1186/s12859-020-03585-4, number: 1 Publisher: BioMed Central.
31. Uritskiy GV, DiRuggiero J, Taylor J. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. Microbiome 2018 Sep;6(1):158. https://doi.org/10.1186/s40168-018-0541-1.
32. Krakau S, Straub D, Gourlé H, Gabernet G, Nahnsen S. nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning. NAR Genomics and Bioinformatics 2022 Mar;4(1):lqac007. https://doi.org/10.1093/nargab/lqac007.
33. McKinney W. pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing 2011;14(9).
34. Hunter JD. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 2007;9(3):90–95. http://ieeexplore.ieee.org/document/4160265/.
35. Waskom M, Botvinnik O, Ostblom J, Gelbart M, Lukauskas S, Hobson P, et al. Seaborn v0.10.0. Online resource 2020 Apr;https://zenodo.org/record/3767070.
36. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Dröge J, et al. Critical Assessment of Metagenome Interpretation - A benchmark of metagenomics software. Nature Methods 2017;14(11):1063–1071.
37. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nature Methods 2021 Sep;p. 1–8. https://www.nature.com/articles/s41592-021-01254-9, bandiera_abtest: a Cg_type: Nature Research Journals Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Computational platforms and environments;Programming language;Software Subject_term_id: computational-platforms-and-environments;programming-language-and-code;software.
38. Salazar VW. Metaphor's documentation. Online resource 2023;https://metaphor-workflow.readthedocs.io/en/latest/.
39. Fritz A, Hofmann P, Majda S, Dahms E, Dröge J, Fiedler J, et al. CAMISIM: simulating metagenomes and microbial communities. Microbiome 2019 Feb;7(1):17. https://doi.org/10.1186/s40168-019-0633-6.
40. Olm MR, Brown CT, Brooks B, Banfield JF. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. The ISME Journal 2017 Dec;11(12):2864–2868. https://www.nature.com/articles/ismej2017126, bandiera_abtest: a Cg_type: Nature Research Journals Number: 12 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Metagenomics;Next-generation sequencing Subject_term_id: metagenomics;next-generation-sequencing.
41. Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT, Aluru S. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. Nature Communications 2018 Nov;9(1):5114. https://www.ncbi.nlm.nih.gov/pubmed/30504855, publisher: Nature Publishing Group UK.
42. Lafayette L, Wiebelt B. Spartan and NEMO: Two HPC-Cloud Hybrid Implementations. 2017 IEEE 13th International Conference on e-Science (e-Science) 2017 Oct;p. 458–459.
43. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Research 2015;25(7):1043–1055.
44. Chaumeil PA, Mussig AJ, Hugenholtz P, Parks DH. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. Bioinformatics 2019;36(6):1925–1927. https://academic.oup.com/bioinformatics/article-abstract/36/6/1925/5626182.
45. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nature Methods 2018 Jul;15(7):475–476. https://www.nature.com/articles/s41592-018-0046-7, number: 7 Publisher: Nature Publishing Group.
46. Inc A. Conda — Conda documentation. Online resource 2023;https://docs.conda.io/en/latest/.
47. Jackson M, Kavoussanakis K, Wallace EWJ. Using prototyping to choose a bioinformatics workflow management system. PLOS Computational Biology 2021 Feb;17(2):e1008622. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008622, publisher: Public Library of Science.
48. Lai S, Pan S, Sun C, Coelho LP, Chen WH, Zhao XM. metaMIC: reference-free misassembly identification and correction of de novo metagenomic assemblies. Genome Biology 2022 Nov;23(1):242. https://doi.org/10.1186/s13059-022-02810-y.
49. Quince C, Nurk S, Raguideau S, James R, Soyer OS, Summers JK, et al. STRONG: metagenomics strain resolution on assembly graphs. Genome Biology 2021 Jul;22(1):214. https://doi.org/10.1186/s13059-021-02419-7.
50. Pandolfo M, Telatin A, Lazzari G, Adriaenssens EM, Vitulo N. MetaPhage: an Automated Pipeline for Analyzing, Annotating, and Classifying Bacteriophages in Metagenomics Sequencing Data. mSystems 2022 Sep;7(5):e00741–22. https://journals.asm.org/doi/10.1128/msystems.00741-22, publisher: Ameri-

can Society for Microbiology.
51. Karlicki M, Antonowicz S, Karnkowska A. Tiara: deep learning-based classification system for eukaryotic sequences. Bioinformatics 2022 Jan;38(2):344–350. https://doi.org/10.1093/bioinformatics/btab672.
52. Pronk L, Medema M. Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structur; 2021.
53. Shaban B, Quiroga MdM, Turnbull R, Tescari E, Lê Cao KA, Verbruggen H. MetaGenePipe: An Automated, Portable Pipeline for Contig-based Functional and Taxonomic Analysis. Journal of Open Source Software 2023 Feb;https://joss.theoj.org/papers/c9c52942084258507eeb1693b83153ba.

## Supplementary Material

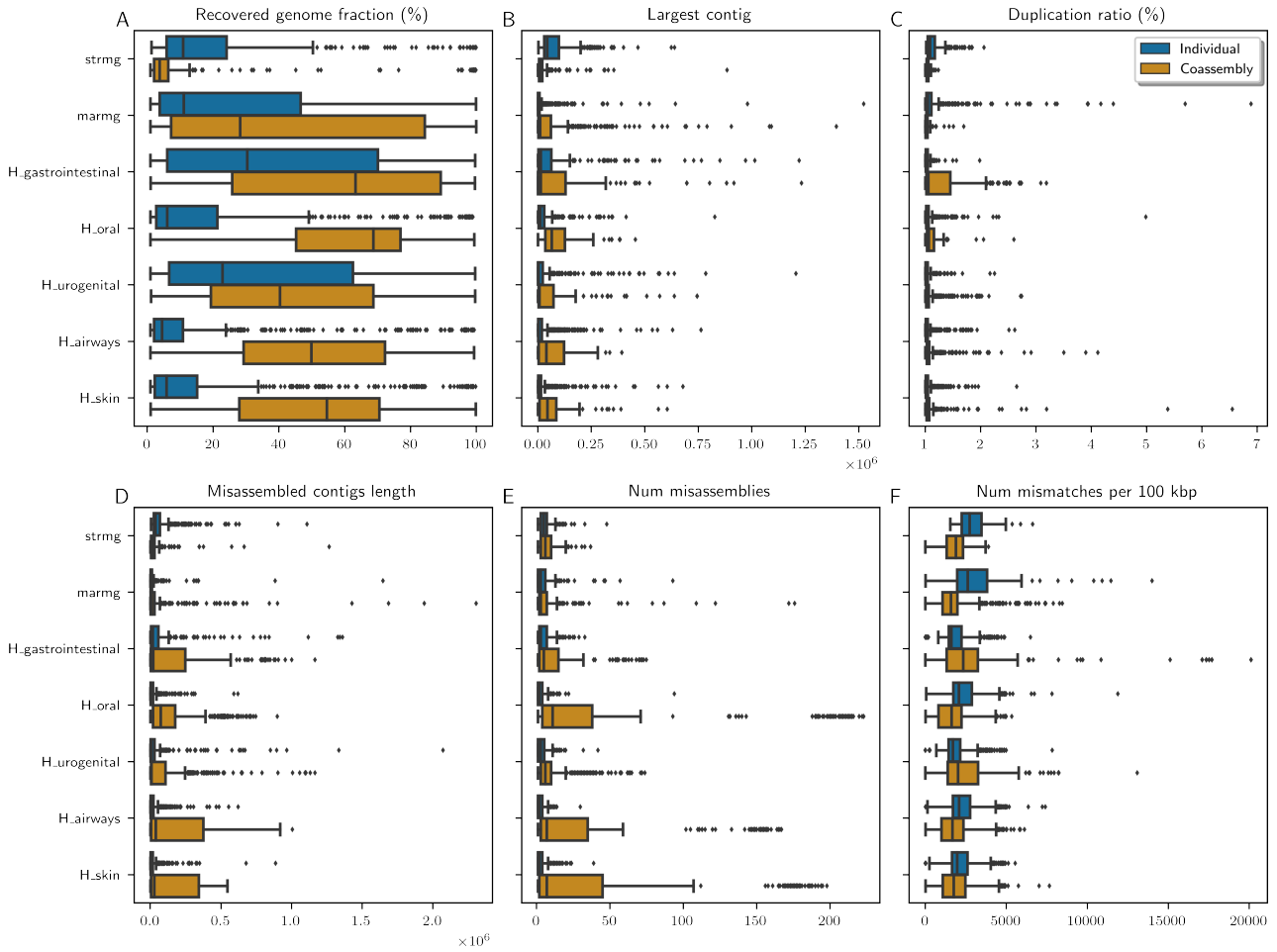


***Figure S1.*** **Differences between assembly strategies across datasets.** Same data as Fig 5, but including outliers.

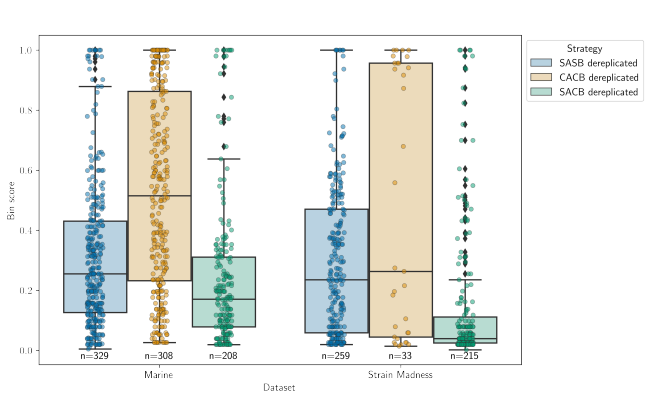


***Figure S2.*** **Boxplot of bin scores across different strategies.** Each data point is a genome bin, and Y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colours represent different strategies. Numbers underneath each bar show the number of data points for that bar. Bins sets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

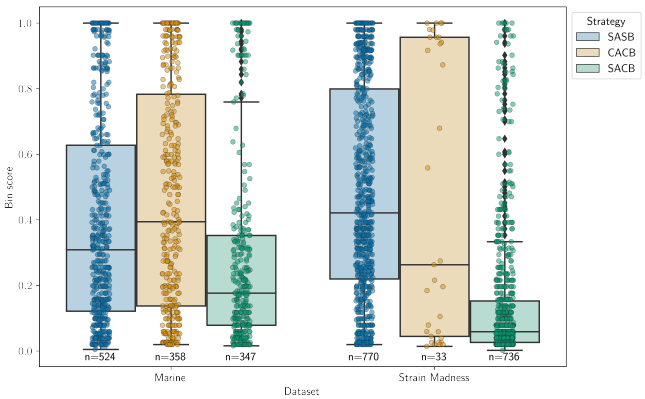


***Figure S3.*** **Boxplot of bin scores across different strategies for non-dereplicated data.** Same as **Figure S2**, but with non-dereplicated data. Each data point is a genome bin, and Y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colours represent different strategies. Numbers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

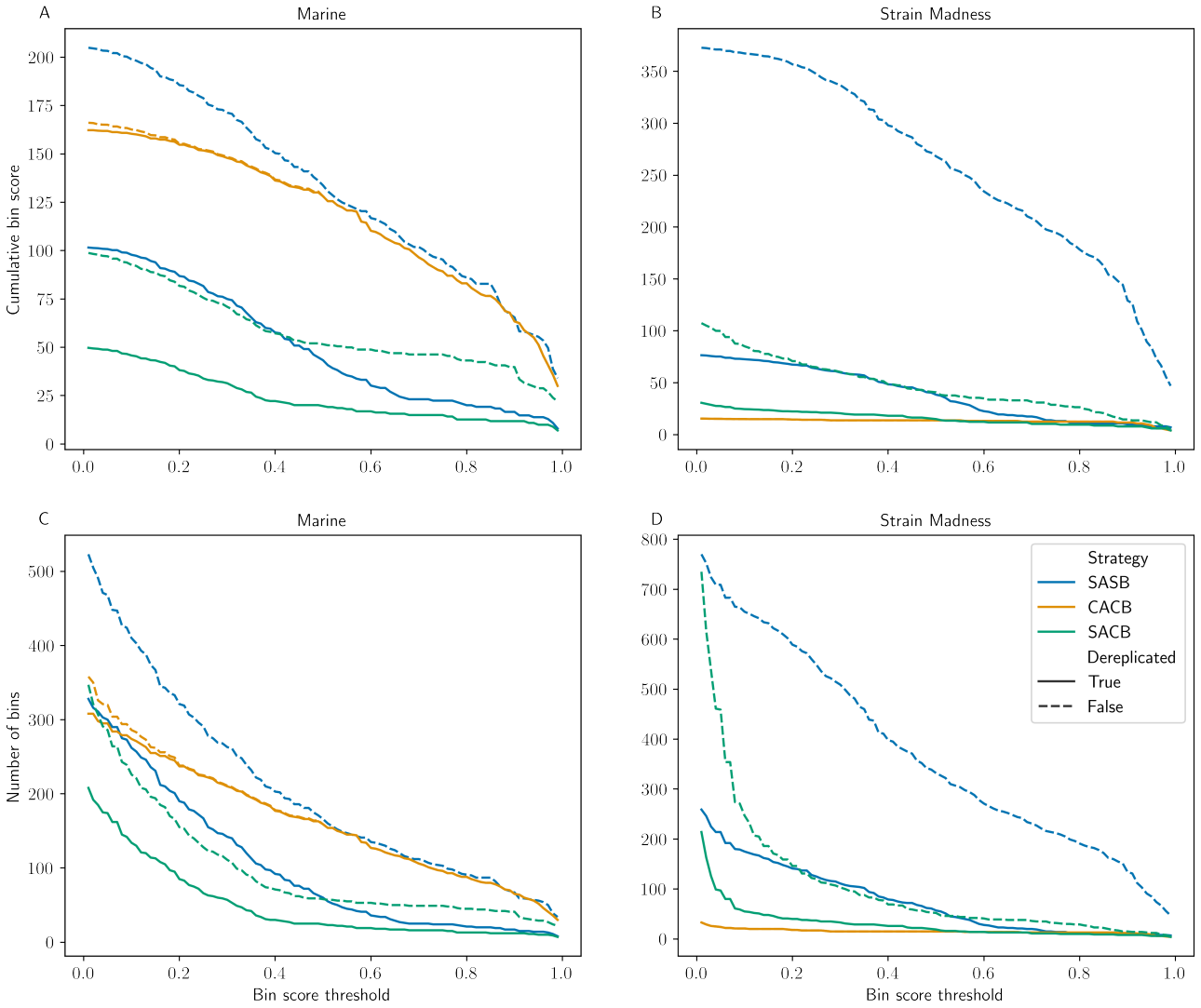


***Figure S4.*** **Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets.** Solid lines show the same data as Fig 6, and dashed lines show data based on bins prior to dereplication with dRep.

***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].

| Bin ID | Bin score $S_b$ | SCG completeness | SCG redundancy | FastANI reference | GTDB Classification |
|---|---|---|---|---|---|
| metabat2.1221 | 1 | 100 | 0 | GCF_004793475.1 | Bacteroides sp002491635 |
| concoct.122 | 1 | 100 | 0 | GCF_024397795.1 | Lactobacillus intestinalis |
| vamb.S1C5590 | 1 | 100 | 0 | GCF_000614185.1 | Phocaeicola sartorii |
| metabat2.4898 | 1 | 100 | 0 | GCF_000969835.1 | Parabacteroides goldsteinii |
| concoct.156 | 0.98039216 | 98 | 0 | GCF_003030305.1 | Cutibacterium acnes |
| concoct.148 | 0.97843137 | 100 | 2 | GCF_001436695.1 | Lactobacillus taiwanensis |
| concoct.92 | 0.95686275 | 100 | 4 | GCF_014863545.1 | Paenibacillus lautus_A |
| concoct.121 | 0.95686275 | 100 | 4 | GCF_000012845.1 | Parabacteroides distasonis |
| metabat2.328 | 0.95686275 | 100 | 4 | GCF_000016825.1 | Limosilactobacillus reuteri |
| vamb.S1C971 | 0.94117647 | 94 | 0 | GCF_000392875.1 | Enterococcus faecalis |
| metabat2.3846 | 0.93678431 | 98 | 4 | GCA_009911065.1 | Ventrimonas sp009911065 |
| concoct.136 | 0.91668667 | 96 | 4 | GCF_001027105.1 | Staphylococcus aureus |
| metabat2.1266 | 0.87258904 | 96 | 8 | GCF_001544255.1 | Enterococcus_B faecium |
| concoct.115 | 0.67941176 | 71 | 2 | GCF_016758115.1 | Lactococcus sp002492185 |
| concoct.58 | 0.55843137 | 59 | 2 | GCF_013394695.1 | Streptococcus sp013394695 |
| metabat2.4512 | 0.2745098 | 27 | 0 | GCF_001729805.1 | Enterobacter roggenkampii |
| metabat2.2064 | 0.26315789 | 26 | 0 | GCF_000742135.1 | Klebsiella pneumoniae |
| metabat2.1951 | 0.21568627 | 22 | 0 | GCF_001457635.1 | Streptococcus pneumoniae |
| metabat2.3969 | 0.19607843 | 20 | 0 | GCF_011064845.1 | Citrobacter freundii |
| metabat2.1470 | 0.18421053 | 18 | 0 | GCF_000215745.1 | Klebsiella aerogenes |
| concoct.22 | 0.10526316 | 11 | 0 | GCF_001729745.1 | Enterobacter hormaechei_A |
| concoct.103_sub | 0.07894737 | 8 | 0 | | Unclassified Bacteria |
| concoct.97_sub | 0.05882353 | 6 | 0 | | Citrobacter |
| concoct.124_sub | 0.05882353 | 6 | 0 | | Unclassified Bacteria |
| concoct.27_sub | 0.04473684 | 16 | 3 | | Enterobacter |
| concoct.91_sub | 0.03921569 | 4 | 0 | GCF_001729745.1 | Enterobacter hormaechei_A |
| concoct.159 | 0.02631579 | 3 | 0 | | Unclassified Bacteria |
| concoct.64_sub | 0.02631579 | 3 | 0 | | Unclassified Bacteria |
| vamb.S1C21648 | 0.02631579 | 3 | 0 | | Unclassified |
| metabat2.3037_sub | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| concoct.13 | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| vamb.S1C7072 | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| concoct.35_sub | 0.01417112 | 86 | 53 | | Klebsiella |

Professor Kim-Anh Lê Cao
Statistical Genomics
NHMRC Career Development Fellow
School of Mathematics and Statistics
Melbourne Integrative Genomics
The University of Melbourne | VIC 3010
T: +61 (0)3834 43971 | kimanh.lecao@unimelb.edu.au

June 5 2023

RE: Revision of manuscript GIGA-D-23-00067

Dear Dr. Nogoy,

On behalf of the authors of the Metaphor manuscript, please find attached our review response letter outlined. In the submission, you will also find a track-change PDF indicating the sections that have been modified. We have also fixed the titles in Panels A and C of Figures 5 and S1.

Thank you for handling our manuscript, and we hope to hear from you soon,

Yours sincerely,

Prof Kim-Anh Lê Cao

Reviewer reports:

Reviewer #1: the authors present a snakemake-based workflow to automate and chain the main computational ingredients (assembly and binning) of genome-centric metagenomics; the authors developed a technically sound tool for this purpose, and by itself it is certainly valuable to the research community and worth of publication. however, even if the article is casted as a technical note -hence with an emphasis on the design, implementation and assessment of the tool-, I feel that a more thorough discussion of both its abilities and inabilities (e.g. strain resolution, detection of low abundance organisms, identification of virus bins, etc) would be worth for a more general audience. On the same token, a more deep discussion of some of the results obtained with their tool (see below) would be of interest and would also illustrate useful use cases.

We thank Reviewer #1 for their suggestions and believe that they have greatly improved the quality of the manuscript. We address each comment point-by-point in the following sections.

I would suggest the following modifications/additions:

-the experiments with the strain madness dataset suggest that the genomes (or fragments thereof, i.e. the bins) resolved should be viewed as "species" genomes, or composite genomes possibly originating from multiple strains. if so, do the authors think this represents a hard limit to the assembly + binning approach, or could further existing tools (e.g. performing variant detection on top of cross-assembly before the binning step) be integrated or developed in the future for strain-resolution (i.e. to identify strains not dominant in any sample)?

Yes, it would be possible to integrate additional tools to further refine strain-resolution with. Metaphor. Currently, Metaphor uses DAS Tool as a bin refinement tool which selects a set of best-quality, non-redundant bins. In scenarios where there is a high level of strain diversity, such as the Strain Madness dataset, the selection of bins performed by DAS Tool would indeed present a limit to the strain resolution, regardless of the selected strategy. However, in the “Single Binning” (SB) strategy, where each sample is binned individually, one can use tools like dRep (Olm *et al* 2017) to identify species clusters from the bins generated by DAS Tool for each sample. It is also possible to perform pre-binning steps which could aid binning resolution, such as evaluating the assembly with MetaQUAST (Mikheenko *et al* 2016) (which is supported in Metaphor), or detecting misassembled contigs with a tool such as metaMIC (Lai *et al* 2022). We added a section to the **Availability and future directions** section addressing this:

> 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].

Something which would also help increase strain-level resolution of bins would be to add support for long-reads data. This will be a priority for a future version of Metaphor. Adding such feature depends on the complexity, degree of “user-friendliness”, and code quality of tools that integrate short and long-reads data. We discuss this in the manuscript in the third paragraph of the **Results and Discussion** section:

> [...] The choice of bioinformatics tools was informed by the results of the 2nd 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].

To conclude, it would definitely be possible to integrate existing tools to Metaphor to further refine strain resolution of metagenome-assembled genomes (MAGs). However, there is evidence to support that strain-level resolution is limited by sequencing technology, with long reads or hybrid approaches usually outperforming short reads (Gehrig *et al* 2022, Meyer *et al* 2022). There are tools which focus exclusively in obtaining higher resolution in metagenomic assemblies, such as STRONG (Quince et al 2021), and groups have achieved lineage-level resolution in MAGs using “a combination of HiFi sequencing, Hi-C binning and a computational phasing approach to resolve genome bins.” (Reiter & Brown 2022, Bickhart 2022). Currently, Metaphor supports only the analysis of short-reads data, as it is designed for large-scale, general-purpose analyses of massive (short-reads) datasets. We do have plans to integrate long-reads support as we continue to maintain Metaphor in the future, but for the current version, we do not intend to add this particular feature as it may affect the stability of the workflow. As we discuss in the text (see: first paragraph of Introduction, third paragraph of Design and Implementation) one of the main challenges associated with genome-resolved metagenomic (GRM) workflows is that it is difficult to support and maintain such tools, due to the high number of dependencies and steps involved in the data analysis. Thus, it is important to manage workflow complexity and ensure that third-party tools use best practices in packaging and maintenance. A tool such as STRONG (Quince *et al* 2021), for example, is a separate workflow in and of itself and integrating it into Metaphor would greatly increase the complexity of the latter.

-related, a simple summary of the number of individual strains recovered in individual bins for the strain madness experiment would be interesting.

We now provide a supplementary table “Table S1” describing the retrieved strains for the CACB strategy. We omitted other strategies as they present a similar number of recovered high-quality bins (with score ≥ 0.8, **Figure 6d**).

-another issue that would be worth discussing in my opinion is the impact of genome abundance on the recovery of corresponding bins and their quality. the platform developed by the authors appears to be well suited for such kind of analyses and the results would be of both theoretical and practical interest. to put it simply, what is the minimal initial coverage of genomes required in order for them to be recovered in bins of a given size and quality?

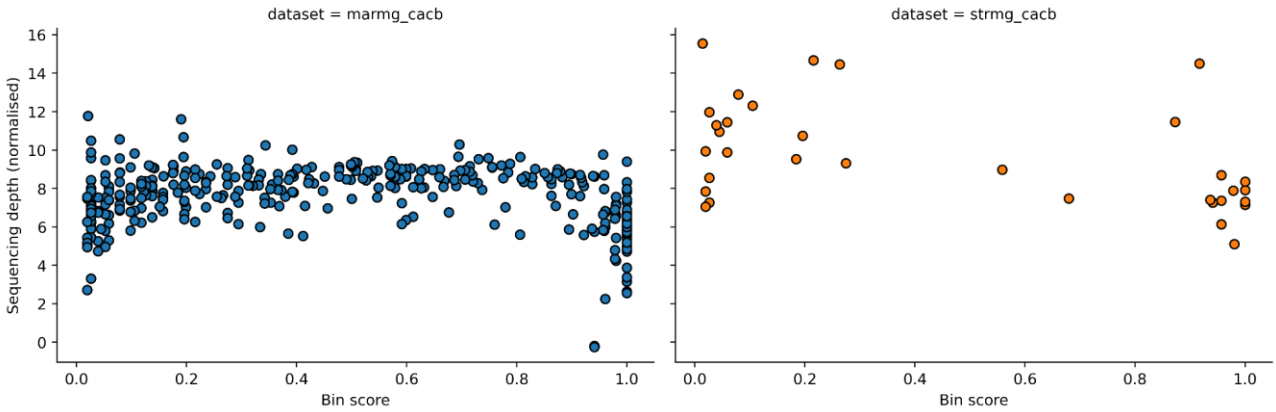


**Figure a:** Bin score (x axis) by normalised sequencing depth across all contigs in the bin (y axis) for two datasets: marmg_cacb on the left and strmg_cacb on the right. Each point is a bin of contigs. Normalised sequencing depth of each bin is equal to the the sum of average depth of all contigs in the bin, divided by contig lengths, divided by total sequencing depth in all binned contigs of sample, multiplied by 1e9, and log-transformed (analogous to a log transcripts-per-million value).

It is difficult to determine a minimal sequencing depth threshold in order to obtain a bin with a certain size and quality. As discussed by Nelson *et al* 2018, some regions of the genome may be harder to assemble and bin due to repetitiveness or peculiar nucleotide composition. Our data shows no correlation between sequencing depth across contigs in a bin and the resulting bin score (**Figure a** in this document). Meyer *et al* 2022 also discuss why some bins are more difficult to recover than others, due to factors such as e.g. uniqueness of the genome in the sample (low level of strain variation). We argue that this question will likely depend on the type of data that is being analysed, subject to variations due to source of sample, library preparation and sequencing protocol, and community structure. This is discussed in the last paragraph of the Results and Discussion, which also addresses the next point raised by the reviewer.

-rem: theses two issues (strain-level diversity and individual strain genome abundances) likely interact to limit bin resolution, and this could be mentioned by the authors.

We have edited the last paragraph of the **Results and Discussion** section to address this comment:

> [...] 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 Figure 1g); genomes with low strain diversity (*i.e.* are less than 95% similar with any other genomes) have higher correlation between sequencing coverage and recovered fraction than common genomes (≥ 95% similar to other genomes in the sample), although many times sequencing coverage was not all correlated with genome recovery fraction, specially for smaller bins that represent plasmids or circular elements.

-the data presented by the authors suggest that the metabat binning engine significantly outperforms the other two tools (concoct and vamb, which are both widely used), see e.g Figure 2; what would

account for that, and do the authors think this is a general observation (i.e. beyond the specific CACB setting or marine metagenome shown in Fig 2)?

To answer the reviewer's question, we believe that this is not a generalised characteristic, but rather is due to the interaction between the binning algorithm and input data characteristics, but pinpointing the exact causes would be a challenging task which is out of the scope of the present manuscript. Meyer et al (2022) showed that different binning tools perform differently depending on the original dataset, with no clear "winner". The idea behind a tool like DAS Tool is to conciliate the output of multiple binners to generate a consensus output which theoretically outperforms each individual binner. So, even if MetaBAT may have performed better in these particular datasets/contexts, by combining it with CONCOCT and vamb, we are able to obtain an improved end result. We have addressed this in the third paragraph to the **Results and discussion: Assembly and binning strategies** section:

> [...] The performance of each binning tool is also variable between strategies and is conditional on the characteristics of the original dataset, with no clear "winner", and each tool favouring particular performance metrics, in agreement with results from the 2nd CAMI Challenge [8]. Tools like DAS Tool attempt to conciliate the output of multiple binning algorithms to generate a consensus output which theoretically outperforms each individual algorithm.

-a bin refinement step (based on the DAS tool and dereplication) is frequently mentioned but should be more detailed (including a precise definition of the bin quality metric used).

We have added a section to **Results and Discussion: Reconstruction of metagenome-assembled genomes** addressing this:

> [...] 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, non-redundant bins remain; the bin score (Sb) increases with the number of SCGs and decreases with duplicate SCGs per bin. Please refer to [28], Figure 1
> and Equation 1 for an overview of the DAS algorithm and the formula to determine the bin score, respectively. [...]

further rather minor comments:

-in the abstract, when mentioning "technical challenges associated with...", it would be worth mentioning that algorithmic challenges are present as well.

We have edited the **Abstract** to include this:

> [...] 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 [...]

-in the introduction, "It is hypothesised that pooled assembly and binning may lead to improved results when analysing communities with high genetic diversity, and to poorer results when there is a high level of intraspecies/strain-level diversity". I would assume there are many instances in the real world that are both, i.e. that present both high inter-species and intra-species genetic diversity, what then?

Indeed, and Metaphor's flexibility to change between the different assembly/binning strategies would be an advantage when dealing with that, as users can easily tweak their settings to run the workflow with different strategies. If that is possible, users could then combine all of the bins generated with the different strategies with a tool such as dRep (used in the manuscript). The choice will depend on the underlying biological question and on the available sequencing depth. We did not add anything to the manuscript for this point.

-in the future directions, the authors mention the identification of eukaryotic and viral contigs and bins, and could shortly elaborate how this could be done properly.

We added the following to the **Results and discussion** section:

> [...] 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 unnannotated (putative) eukaryotic and viral contigs. These can then be used as input for a eukaryotic or viral discovery pipeline [48, 49, 50 ]. This can also be done directly with the output of the assembly module, but in that case there won't 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.

-the sentence "In summary, our assessment of ..." at the end of the ms appears to have a syntactic problem.

We have rephrased this sentence:

> 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 approaches, as computational requirements are higher for the pooled strategies. [...]

Reviewer #2: The Metaphor is a workflow with high completeness for short-read-based metagenomic analysis. I look forward to its compatibility with long-read platforms (ONT and PacBio). This work is worth publishing. However, it is still a bioinformatic knowledge and skill-required toolkit. If the Metaphor can be integrated into a web-based platform, such as Galaxy or Kbase, it would be more user-friendly for much more users.

We thank Reviewer #2 for their comments. We are considering adding the following two features to Metaphor in a future major version release:

- Support for long reads (as discussed in our answer to Reviewer #1 above)
- Deployment in a web-based workflow platform. For the latter, we have started drafting an XML configuration file to investigate the deployment of Metaphor on Galaxy (see https://github.com/vinisalazar/metaphor/tree/dev/.github/planemo), using the Planemo (https://planemo.readthedocs.io/) tool, that is officially supported by the Galaxy community. In the next months we will investigate whether this could be an appealing option to our users. This implementation will need to be accompanied by extensive tests, therefore we expect it to be released with Metaphor v2.0.

**References**


Bickhart DM, Kolmogorov M, Tseng E, Portik DM, Korobeynikov A, Tolstoganov I, et al.. Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. *Nat Biotechnol*. Nature Publishing Group; 2022; doi: 10.1038/s41587-021-01130-z.

Chaumeil P-A, Mussig AJ, Hugenholtz P, Parks DH. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. *Bioinformatics*. 2019; doi: 10.1093/bioinformatics/btz848.

Gehrig JL, Portik DM, Driscoll MD, Jackson E, Chakraborty S, Gratalo D, et al.. Finding the right fit: evaluation of short-read and long-read sequencing approaches to maximize the utility of clinical microbiome data. *Microb Genom*. 2022; doi: 10.1099/mgen.0.000794.

Lai S, Pan S, Sun C, Coelho LP, Chen W-H, Zhao X-M. metaMIC: reference-free misassembly identification and correction of de novo metagenomic assemblies. *Genome Biology*. 2022; doi: 10.1186/s13059-022-02810-y.

Meyer F, Fritz A, Deng Z-L, Koslicki D, Lesker TR, Gurevich A, et al.. Critical Assessment of Metagenome Interpretation: the second round of challenges. *Nat Methods*. Nature Publishing Group; 2022; doi: 10.1038/s41592-022-01431-4.

Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*. 2016; doi: 10.1093/bioinformatics/btv697.

Nelson WC, Tully BJ, Mobberley JM. Biases in genome reconstruction from metagenomic data. *PeerJ*. 2020; doi: 10.7717/peerj.10119.

Olm MR, Brown CT, Brooks B, Banfield JF. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *ISME J*. Nature Publishing Group; 2017; doi: 10.1038/ismej.2017.126.

Quince C, Nurk S, Raguideau S, James R, Soyer OS, Summers JK, et al.. STRONG: metagenomics strain resolution on assembly graphs. *Genome Biology*. 2021; doi: 10.1186/s13059-021-02419-7.

Reiter TE, Brown CT. MAGs achieve lineage resolution. *Nat Microbiol*. Nature Publishing Group; 2022; doi: 10.1038/s41564-021-01027-2.

*GigaScience*, 2023, 1–16

doi: xx.xxxx/xxxx
**Manuscript in Preparation**
**Technical Note**

TECHNICAL NOTE

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

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

[1]Melbourne Integrative Genomics, School of Mathematics & Statistics, University of Melbourne, Parkville, Victoria, Australia and [2]Melbourne Data Analytics Platform (MDAP), University of Melbourne, Parkville, Victoria, Australia and [3]Department of Molecular and Translational Sciences, Monash University, Clayton, Victoria, Australia and [4]Centre for Innate Immunity and Infectious Diseases, Hudson Institute of Medical Research, Clayton, Victoria, Australia and [5]School of BioSciences, University of Melbourne, Parkville, Victoria, Australia and [6]Department of Microbiology and Immunology, The University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Parkville, Victoria, Australia.

[*]kimanh.lecao@unimelb.edu.au
[†]Deceased.
[§]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 analyse 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 customisable 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. ~~Both~~ 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.

**Key words**: 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. ~~In recent years, applications~~ Applications of GRM have led to unprecedented insight into microbial diversity, ecology, and evolution, due to the recovery of (mostly uncultivated) metagenome-

**Compiled on:** June 5, 2023.
Draft manuscript prepared by the author.

assembled genomes (MAGs) [1, 2, 3, 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 analysing 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 heterogeneous data sources, have dozens of customisable parameters, and depend on several specialised 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 catalogue 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 hypothesised that pooled assembly and binning may lead to improved results when analysing 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 provides taxonomic and functional abundance data for quantitative metagenome analyses. Our software advances existing metagenomic pipelines by combining two 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 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, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] and custom scripts that connect different parts of the workflow, listed on Table 2.

The workflow consists of six 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 show on Fig 1.

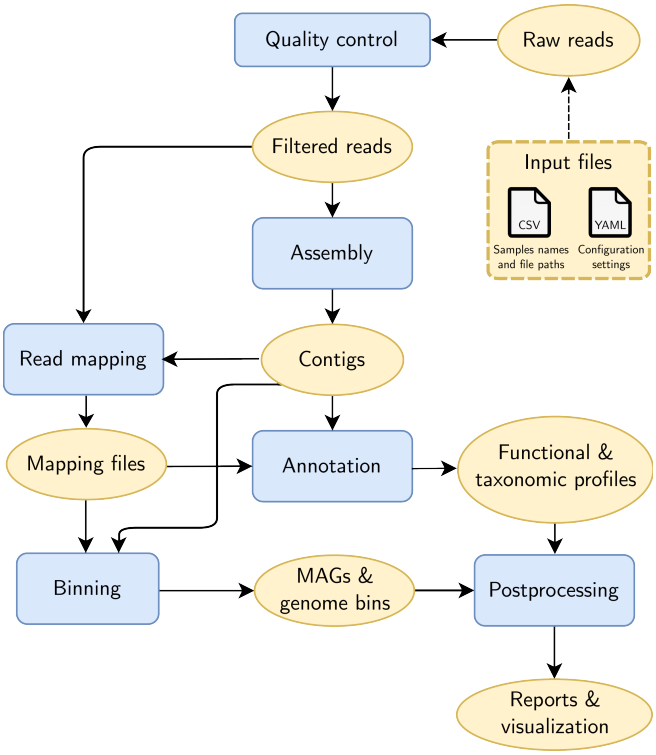


**Figure 1. Simplified workflow diagram.** Workflow modules are represented by rectangular blue shapes and data files are represented by oval yellow shapes, except for entrypoint files shown in a dashed yellow rectangle. Arrows indicate input and output of data between modules.

The choice of bioinformatics tools was informed by the results of the 2nd 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.

**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 catalogue | ✓ | | | ✓ | ✓ |
| 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] |
| | Assembly report and plots | Metaphor script* |
| Mapping | Map reads | MiniMap2 [19] |
| | Sort and index mapped reads | Samtools [20] |
| Annotation | Prediction of coding sequences from contigs | Prodigal[21] |
| | Annotation of coding sequences | Diamond, NCBI COG [22, 23] |
| | Annotation of MAGs | Prokka [24] |
| | Annotation report and plots | Metaphor script* |
| Binning | 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] |
| | Binning report and plots | Metaphor script* |
| Postprocessing | Concatenate benchmarks | Metaphor script* |
| | Plot benchmarks | Metaphor script* |

* External libraries used in Metaphor scripts: [33, 34, 35].

**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 | 1394 |
| Human Genital | h_urogenital | 9 | 39 | 1394 |
| Human Gut | h_gastrointestinal | 10 | 44 | 1057 |
| Human Oral | h_oral | 10 | 43 | 1057 |
| Human Skin | h_skin | 10 | 44 | 1394 |

**Table 4.** Output files for each strategy. If only one dataset/group is being analysed, 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<br>Sample_1.fastq<br>Sample_2.fastq | Sample_0_contigs.fasta<br>Sample_1_contigs.fasta<br>Sample_2_contigs.fasta | Sample_0_bins/<br>Sample_1_bins/<br>Sample_2_bins/ |
| SACB | Single assembly, Cobinning | Sample_0.fastq<br>Sample_1.fastq<br>Sample_2.fastq | Sample_0_contigs.fasta<br>Sample_1_contigs.fasta<br>Sample_2_contigs.fasta | Cobinning_bins/ |
| CACB | Coassembly, Cobinning | Sample_0.fastq<br>Sample_1.fastq<br>Sample_2.fastq | Coassembly_contigs.fasta | Cobinning_bins/ |

The workflow takes two 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 analysed datasets from CAMI II [8]. 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 five 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 datasets: 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; coassembly, cobinning ('CACB'), where all samples from the dataset were 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 analyse 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 results 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 which 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 (HPC) 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.

### Reconstruction of metagenome-assembled genomes

Metaphor produces genome bins generated with three tools: Vamb, MetaBAT2 and CONCOCT [25, 26, 27] that are refined with DAS Tool [28]. 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

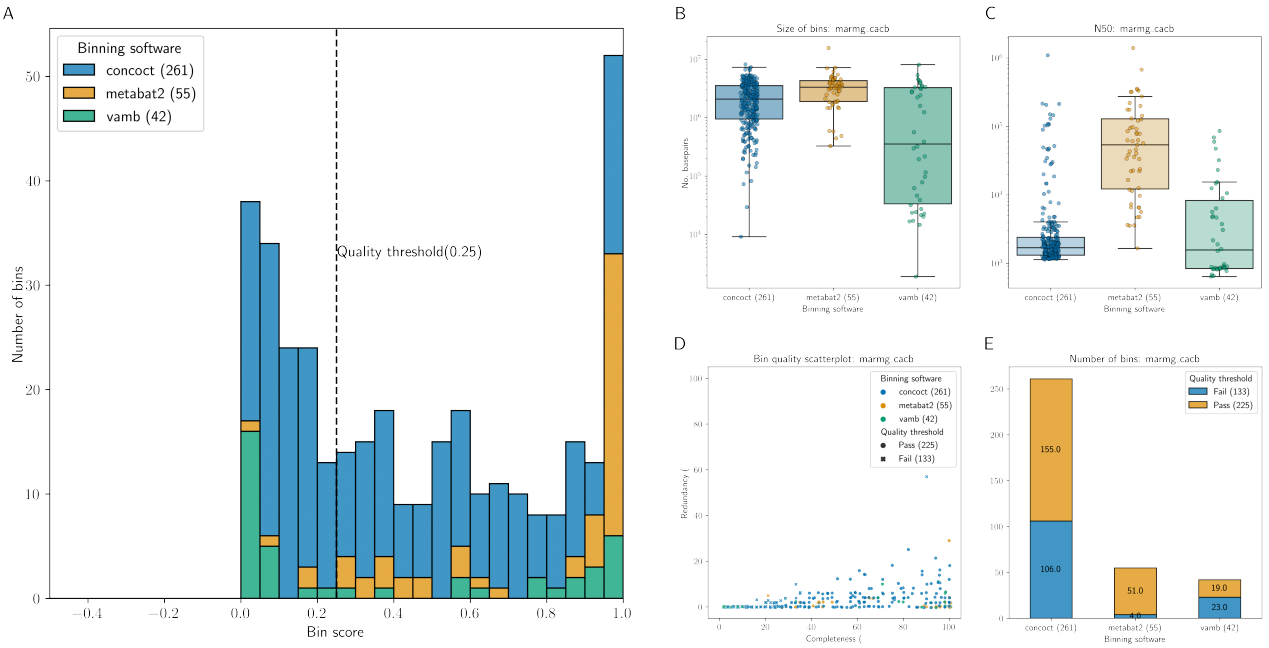


**Figure 2. Binning report generated by Metaphor for the CAMI II Marine metagenome dataset processed with the 'CACB' (coassembly, cobinning) setting.** Panel **A** shows a stacked histogram of the distribution of bin scores, with the defined quality threshold highlighted as a dashed line. Panels **B** and **C** show, respectively, the size (in base pairs) and N50 of bins. The Y-axis is in log-scale. Panel **D** shows a scatterplot of completeness and redundancy for each bin. Colours 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 Panel **A**). Panel **E** shows how many bins passed or failed the quality threshold for each binning tool.

best-quality, non-redundant bins remain; the bin score ($S_b$) increases with the number of SCGs and decreases with duplicate SCGs per bin. Please refer to [28], Figure 1 and Equation 1 for an overview of the DAS algorithm and the formula to determine the bin score, respectively. The input for each binning tool differs slightly, but they all rely on the catalogue 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 one is generated if cobinning is performed), which highlights three key metrics: completeness, redundancy, and bin score. The first two 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 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 detected in the previous round. Abundances for each feature are calculated based on the coverage of all coding sequences which align to that feature. Fig 3 illustrates the profile visualisations offered by Metaphor: a heatmap of COG categories for the functional profile and a stacked barplot for the most abundant taxa (for the latter, one 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, 15, 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 two 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 three different strategies: single assembly and single

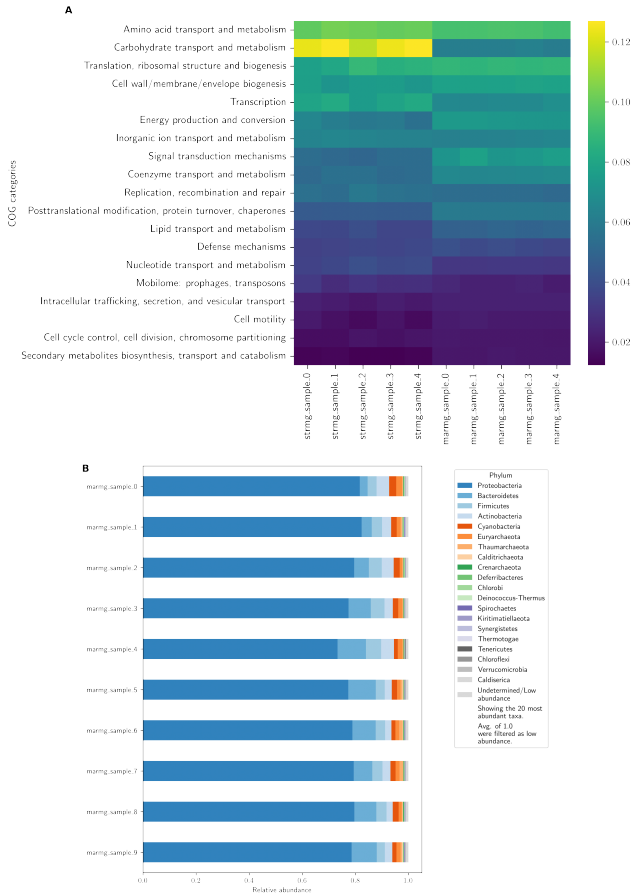


**Figure 3. Annotation plots generated by Metaphor on the Strain Madness ('strmg') and the Marine ('marmg') datasets.** Panel **A** displays the functional profile as a heatmap of the relative abundance of functional COG categories (Y-axis) across samples (X-axis) for five samples from Strain Madness and Marine datasets. Panel **B** displays 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.

binning ('SASB'), single assembly and cobinning ('SACB'), and coassembly and cobinning ('CACB'), see Table 4 and Section 'Assessment on CAMI II synthetic datasets' for details. For assembly, we used the five different groups in the Human Microbiome dataset along with the Strain Madness and Marine datasets. We only used the latter two datasets for the binning assessment.

We used six metrics to evaluate assembly performance: percentage of recovered genome fraction, size of the largest contig, duplication ratio, length of misassembled contigs, number of misassemblies, and number of mismatches per 100 thousand base pairs. High values for the first two metrics and low values for the last four indicate better performance. We observed a general trade-off between assembly completeness (represented by the first two metrics), and the number of errors in the assembly (represented by the last four metrics) (Figure 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 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 (**Figure S3**). 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), which did not occur in the Marine dataset. The performance of each binning tool is also variable between strategies and is conditional on the characteristics of the original dataset, with no clear "winner", and each tool favouring particular performance metrics, in agreement with results from the 2nd CAMI Challenge [8]. Tools like DAS Tool attempt to conciliate the output of multiple binning algorithms to generate a consensus output which 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 of bins above an increasing score threshold, shown on 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 **Figure 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 ~~, the~~ —a summary of the bins recovered with that dataset is displayed on ??. 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 ~~assessment of different assembly and binning strategies~~ 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~~, except when~~. The exception to this is when 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), where 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 Figure 1g); genomes with low strain diversity (*i.e.* are less than 95% similar with any other genomes) have

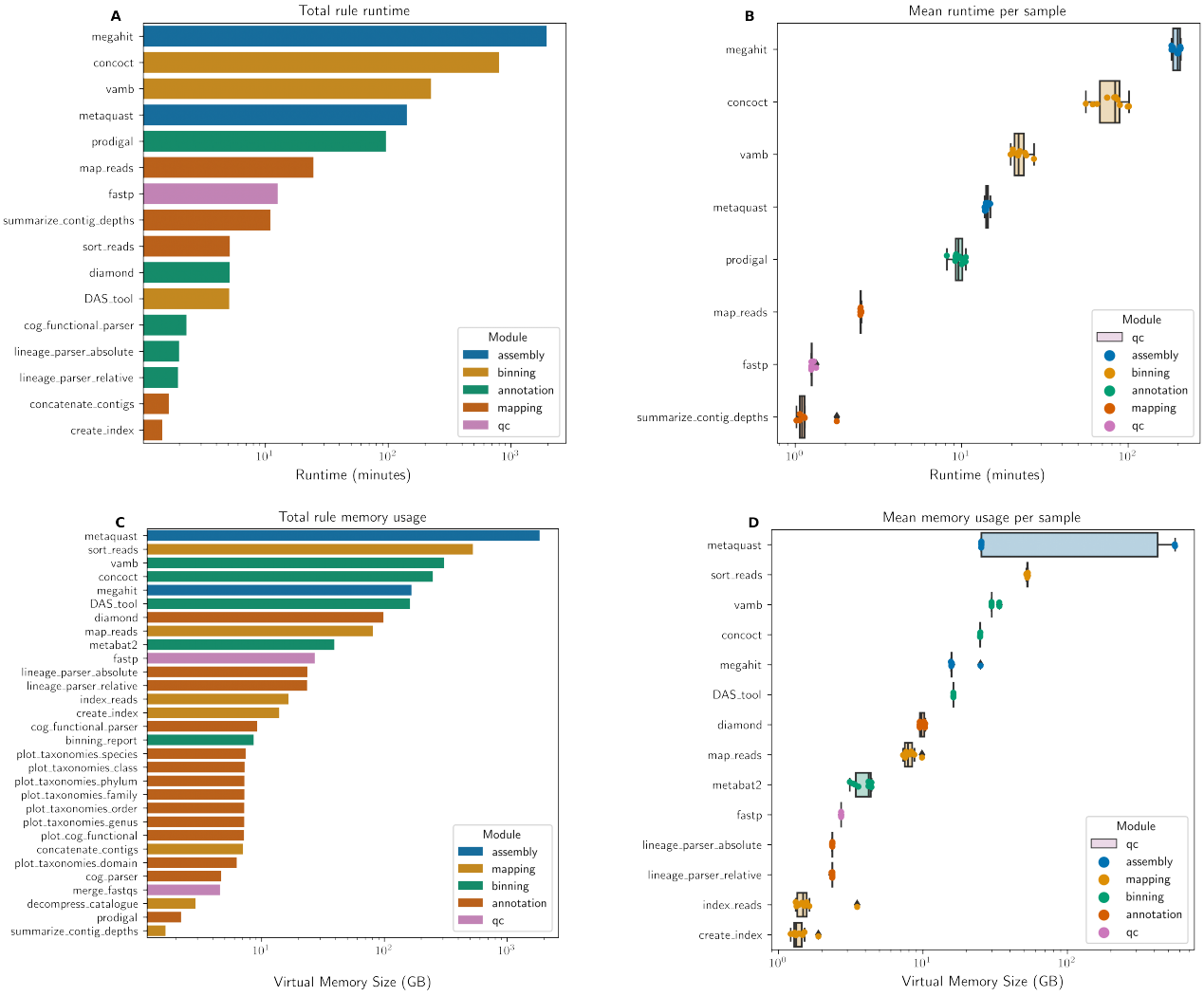


**Figure 4. Performance metrics report generated by Metaphor on the Marine dataset processed with the SASB strategy.** Total runtime per rule (**A**), mean runtime per sample (**B**), total memory usage per rule (**C**), and mean memory usage per sample (**D**). X-axis is in log format. Cutoffs are applied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

higher correlation between sequencing coverage and recovered fraction than common genomes (≥ 95% similar to other genomes in the sample), although many times sequencing coverage was not all correlated with genome recovery fraction, specially 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 the testing Metaphor's installation (see documentation for details) is available from GitHub at https://github.com/vinisalazar/mg-example-data. This data is a subset of the CAMI I challenge data [36] that is 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~~and~~ , CheckM, and ~~a new functionality~~ 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~~and bins . The annotation module can also be improved to facilitate~~ ; 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, 51, 52], but this process could be further improved by facilitating the use of custom reference databases ~~. In addition, Metaphor would benefit from new third-party software to facilitate the generation of~~ in the annotation module. This can also be done directly with the output of the assembly module, but in that case there won't 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

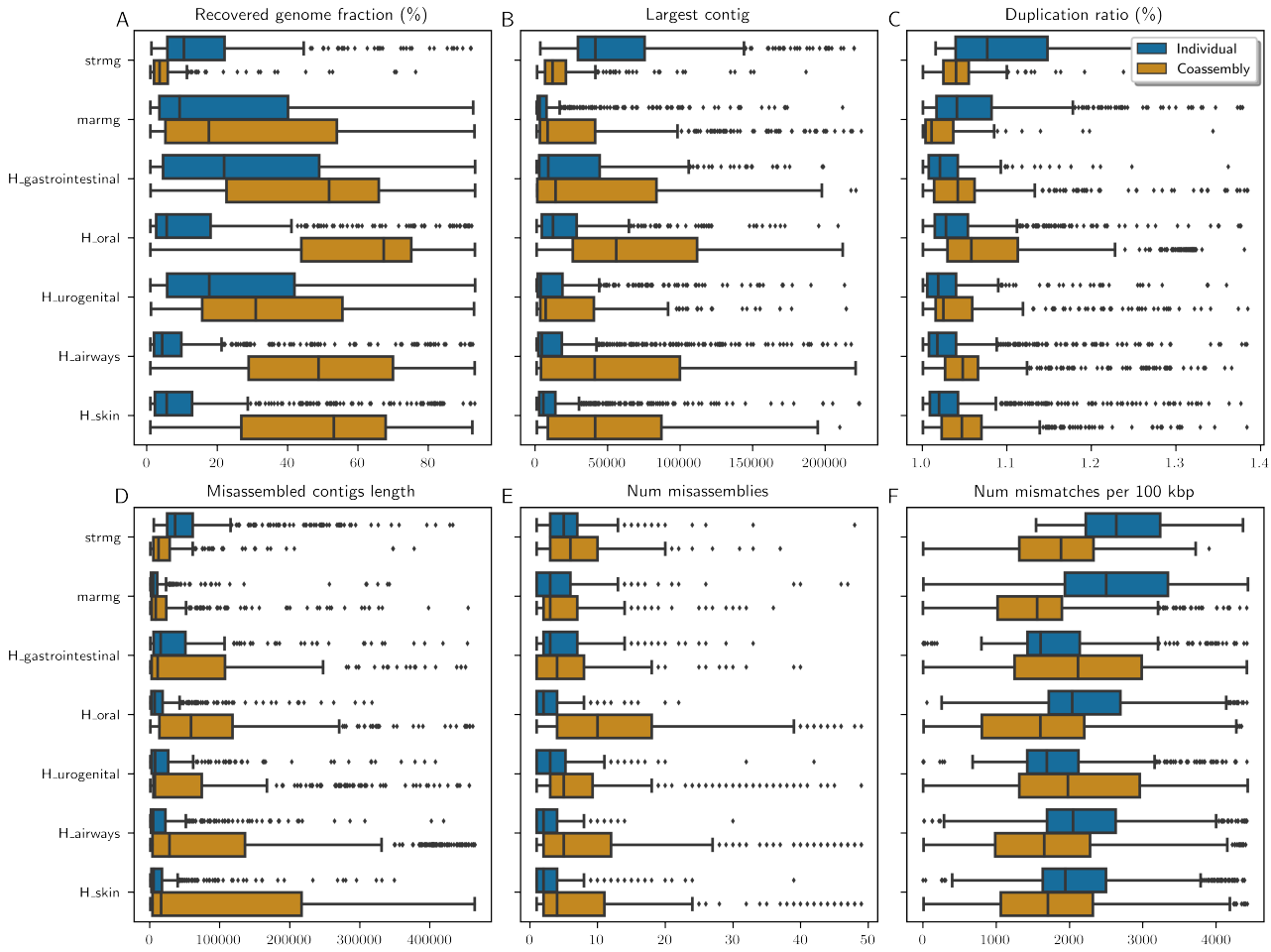


**Figure 5. Differences between assembly strategies for each dataset.** Each data point corresponds to a reference genome evaluated with the MetaQUAST tool. Data points above the 98th percentile were classified as outliers and removed from the figure to improve visualisation. See **Figure S1** for the 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.

integration with non-prokaryotic ~~bins in the near future~~ pipelines, along with support for long reads, are priority features to be added to future major versions of Metaphor.

## Availability checklist

Project name: Metaphor
Project home page: 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

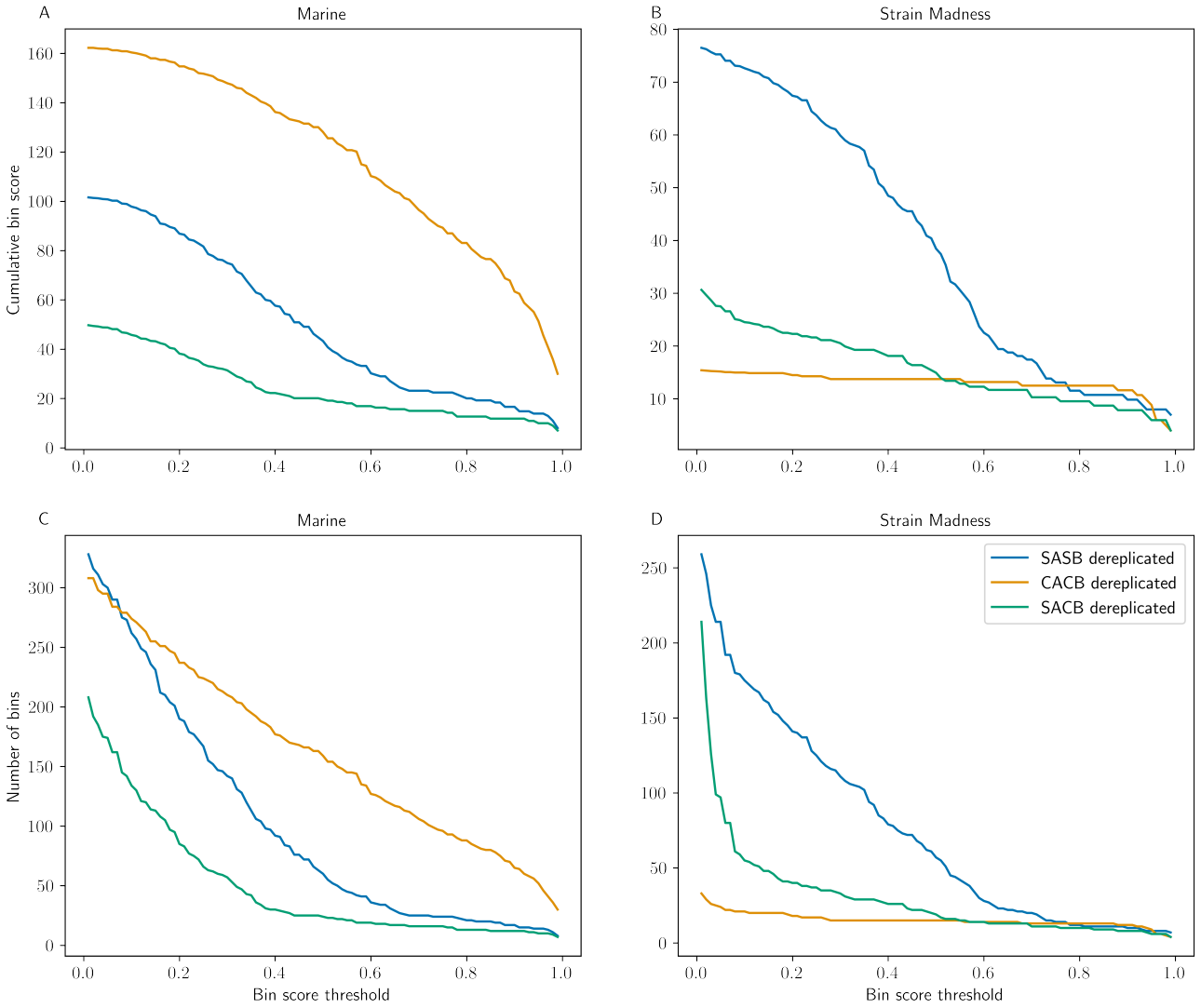


**Figure 6. Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets.** Lines show the cumulative bin score (**A** and **B**) and number of bins (**C** and **D**) along the Y-axis, for bins above a certain score threshold (X-axis). Left column shows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

## Declarations

The authors declare they have no competing interests.


## Funding

VWS is funded by a Melbourne Research Scholarship from The University of Melbourne. VRM is funded by an Australian Research Council DECRA Fellowship DE220100965. KALC 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.


## Author's Contributions

VWS - Conceptualization, Data curation, Methodology, Investigation Software, Writing - original draft; BS, MMQ, RT, ET - Conceptualization, Writing - review and editing; VRM, HV, KALC - Conceptualization, Supervision, Funding Acquisition, Writing - review and editing.


## Acknowledgments

Metaphor benefited strongly from experience gained developing MetaGenePipe [53], 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


1. Almeida A, Nayfach S, Boland M, Strozzi F, Beracochea M, Shi ZJ, et al. A unified catalog of 204,938 reference genomes from the human gut microbiome. Nature Biotechnology 2021 Jan;39(1):105–114. https://www.nature.com/articles/s41587-020-0603-3, number: 1 Publisher: Nature Publishing Group.
2. Parks DH, Rinke C, Chuvochina M, Chaumeil PA, Woodcroft BJ, Evans PN, et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nature Microbiology 2017;2(11):1533–1542. http://dx.doi.org/10.1038/s41564-017-0012-7, publisher: Springer US ISBN: 4156401700127.
3. Tully BJ, Graham ED, Heidelberg JF. The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. Scientific Data 2018 Jan;5(1):170203. https://www.nature.com/articles/sdata2017203, bandiera_abtest: a Cc_license_type: cc_publicdomain Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Bioinformatics;Genome;Metagenomics;Water microbiology Subject_term_id: bioinformatics;genome;metagenomics;water-microbiology.
4. Setubal JC. Metagenome-assembled genomes: concepts, analogies, and challenges. Biophysical Reviews 2021 Dec;13(6):905–909. https://doi.org/10.1007/s12551-021-00865-y.
5. Nelson WC, Tully BJ, Mobberley JM. Biases in genome reconstruction from metagenomic data. PeerJ 2020 Oct;8:e10119. https://peerj.com/articles/10119.
6. Reiter T, Brooks PT, Irber L, Joslin SEK, Reid CM, Scott C, et al. Streamlining data-intensive biology with workflow systems. GigaScience 2021 Jan;10(1). https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giaa140/6092773.
7. Quince C, Walker AW, Simpson JT, Loman NJ, Segata N. Shotgun metagenomics, from sampling to analysis. Nature Biotechnology 2017 Sep;35(9):833–844. https://www.nature.com/articles/nbt.3935, bandiera_abtest: a Cg_type: Nature Research Journals Number: 9 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Microbial communities;Computational biology and bioinformatics;Metagenomics Subject_term_id: communities;computational-biology-and-bioinformatics;metagenomics.
8. Meyer F, Fritz A, Deng ZL, Koslicki D, Lesker TR, Gurevich A, et al. Critical Assessment of Metagenome Interpretation: the second round of challenges. Nature Methods 2022 Apr;19(4):429–440. https://www.nature.com/articles/s41592-022-01431-4, number: 4 Publisher: Nature Publishing Group.
9. Delgado LF, Andersson AF. Evaluating metagenomic assembly approaches for biome-specific gene catalogues. Microbiome 2022 May;10(1):72. https://doi.org/10.1186/s40168-022-01259-2.
10. Rohart F, Gautier B, Singh A, Lê Cao KA. mixOmics: An R package for 'omics feature selection and multiple data integration. PLoS Computational Biology 2017;.
11. McMurdie PJ, Holmes S. phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. PLOS ONE 2013 Apr;8(4):e61217. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0061217, publisher: Public Library of Science.
12. Eren AM, Kiefl E, Shaiber A, Veseli I, Miller SE, Schechter MS, et al. Community-led, integrated, reproducible multi-omics with anvi'o. Nature Microbiology 2020 Dec;6(1):3–6. https://www.nature.com/articles/s41564-020-00834-3.
13. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with Snakemake. F1000 Research 2021 Apr;https://f1000research.com/articles/10-33, type: article.
14. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 2018 Sep;34(17):i884–i890. https://doi.org/10.1093/bioinformatics/bty560.
15. Andrews S. FastQC A Quality Control tool for High Throughput Sequence Data. Online resource 2020 Jan;https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
16. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: Summarize analysis results for multiple tools and samples in a single report. Bioinformatics 2016;.
17. Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 2015 May;31(10):1674–1676. https://doi.org/10.1093/bioinformatics/btv033.
18. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics 2016 Apr;32(7):1088–1090. https://doi.org/10.1093/bioinformatics/btv697.
19. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 2018 Sep;34(18):3094–3100. https://doi.org/10.1093/bioinformatics/bty191.
20. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Twelve years of SAMtools and BCFtools. GigaScience 2021 Feb;10(2):giab008.
21. Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics 2010;11:119. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2848648&tool=pmcentrez&rendertype=abstract, iSBN: 1471-2105 (Electronic)\r1471-2105 (Linking).

22. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nature Methods 2014;https://github.com/bbuchfink/diamond.
23. Galperin MY, Wolf YI, Makarova KS, Vera Alvarez R, Landsman D, Koonin EV. COG database update: focus on microbial diversity, model organisms, and widespread pathogens. Nucleic Acids Research 2021 Jan;49(D1):D274–D281.
24. Seemann T. Prokka: Rapid prokaryotic genome annotation. Bioinformatics 2014;30(14).
25. Nissen JN, Johansen J, Allesøe RL, Sønderby CK, Armenteros JJA, Grønbech CH, et al. Improved metagenome binning and assembly using deep variational autoencoders. Nature Biotechnology 2021 Jan;http://www.nature.com/articles/s41587-020-00777-4, publisher: Nature Research.
26. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, 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, Schirmer M, Quick J, Ijaz UZ, et al. Binning metagenomic contigs by coverage and composition. Nature Methods 2014 Nov;11(11):1144–1146. https://www.nature.com/articles/nmeth.3103, bandiera_abtest: a Cg_type: Nature Research Journals Number: 11 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Genome informatics;Machine learning;Metagenomics Subject_term_id: genome-informatics;machine-learning;metagenomics.
28. Sieber CMK, Probst AJ, Sharrar A, Thomas BC, Hess M, Tringe SG, et al. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. Nature Microbiology 2018 Jul;3(7):836–843. https://www.nature.com/articles/s41564-018-0171-1, number: 7 Publisher: Nature Publishing Group.
29. Churcheward B, Millet M, Bihouée A, Fertin G, Chaffron S. MAGNETO: An Automated Workflow for Genome-Resolved Metagenomics. mSystems 2022 Jun;0(0):e00432–22. https://journals.asm.org/doi/10.1128/msystems.00432-22, publisher: American Society for Microbiology.
30. Kieser S, Brown J, Zdobnov EM, Trajkovski M, McCue LA. ATLAS: a Snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data. BMC Bioinformatics 2020 Dec;21(1):1–8. https://link.springer.com/article/10.1186/s12859-020-03585-4, number: 1 Publisher: BioMed Central.
31. Uritskiy GV, DiRuggiero J, Taylor J. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. Microbiome 2018 Sep;6(1):158. https://doi.org/10.1186/s40168-018-0541-1.
32. Krakau S, Straub D, Gourlé H, Gabernet G, Nahnsen S. nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning. NAR Genomics and Bioinformatics 2022 Mar;4(1):lqac007. https://doi.org/10.1093/nargab/lqac007.
33. McKinney W. pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing 2011;14(9).
34. Hunter JD. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 2007;9(3):90–95. http://ieeexplore.ieee.org/document/4160265/.
35. Waskom M, Botvinnik O, Ostblom J, Gelbart M, Lukauskas S, Hobson P, et al. Seaborn v0.10.0. Online resource 2020 Apr;https://zenodo.org/record/3767070.
36. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Dröge J, et al. Critical Assessment of Metagenome Interpretation - A benchmark of metagenomics software. Nature Methods 2017;14(11):1063–1071.
37. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nature Methods 2021 Sep;p. 1–8. https://www.nature.com/articles/s41592-021-01254-9, bandiera_abtest: a Cg_type: Nature Research Journals Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Computational platforms and environments;Programming language;Software Subject_term_id: computational-platforms-and-environments;programming-language-and-code;software.
38. Salazar VW. Metaphor's documentation. Online resource 2023;https://metaphor-workflow.readthedocs.io/en/latest/.
39. Fritz A, Hofmann P, Majda S, Dahms E, Dröge J, Fiedler J, et al. CAMISIM: simulating metagenomes and microbial communities. Microbiome 2019 Feb;7(1):17. https://doi.org/10.1186/s40168-019-0633-6.
40. Olm MR, Brown CT, Brooks B, Banfield JF. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. The ISME Journal 2017 Dec;11(12):2864–2868. https://www.nature.com/articles/ismej2017126, bandiera_abtest: a Cg_type: Nature Research Journals Number: 12 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Metagenomics;Next-generation sequencing Subject_term_id: metagenomics;next-generation-sequencing.
41. Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT, Aluru S. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. Nature Communications 2018 Nov;9(1):5114. https://www.ncbi.nlm.nih.gov/pubmed/30504855, publisher: Nature Publishing Group UK.
42. Lafayette L, Wiebelt B. Spartan and NEMO: Two HPC-Cloud Hybrid Implementations. 2017 IEEE 13th International Conference on e-Science (e-Science) 2017 Oct;p. 458–459.
43. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Research 2015;25(7):1043–1055.
44. Chaumeil PA, Mussig AJ, Hugenholtz P, Parks DH. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. Bioinformatics 2019;36(6):1925–1927. https://academic.oup.com/bioinformatics/article-abstract/36/6/1925/5626182.
45. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nature Methods 2018 Jul;15(7):475–476. https://www.nature.com/articles/s41592-018-0046-7, number: 7 Publisher: Nature Publishing Group.
46. Inc A. Conda — Conda documentation. Online resource 2023;https://docs.conda.io/en/latest/.
47. Jackson M, Kavoussanakis K, Wallace EWJ. Using prototyping to choose a bioinformatics workflow management system. PLOS Computational Biology 2021 Feb;17(2):e1008622. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008622, publisher: Public Library of Science.
48. Lai S, Pan S, Sun C, Coelho LP, Chen WH, Zhao XM. metaMIC: reference-free misassembly identification and correction of de novo metagenomic assemblies. Genome Biology 2022 Nov;23(1):242. https://doi.org/10.1186/s13059-022-02810-y.
49. Quince C, Nurk S, Raguideau S, James R, Soyer OS, Summers JK, et al. STRONG: metagenomics strain resolution on assembly graphs. Genome Biology 2021 Jul;22(1):214. https://doi.org/10.1186/s13059-021-02419-7.
50. Pandolfo M, Telatin A, Lazzari G, Adriaenssens EM, Vitulo N. MetaPhage: an Automated Pipeline for Analyzing, Annotating, and Classifying Bacteriophages in Metagenomics Sequencing Data. mSystems 2022 Sep;7(5):e00741–22. https://journals.asm.org/doi/10.1128/msystems.00741-22, publisher: Ameri-

can Society for Microbiology.
51. Karlicki M, Antonowicz S, Karnkowska A. Tiara: deep learning-based classification system for eukaryotic sequences. Bioinformatics 2022 Jan;38(2):344–350. https://doi.org/10.1093/bioinformatics/btab672.
52. Pronk L, Medema M. Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structur; 2021.
53. Shaban B, Quiroga MdM, Turnbull R, Tescari E, Lê Cao KA, Verbruggen H. MetaGenePipe: An Automated, Portable Pipeline for Contig-based Functional and Taxonomic Analysis. Journal of Open Source Software 2023 Feb;https://joss.theoj.org/papers/c9c52942084258507eeb1693b83153ba.

## Supplementary Material

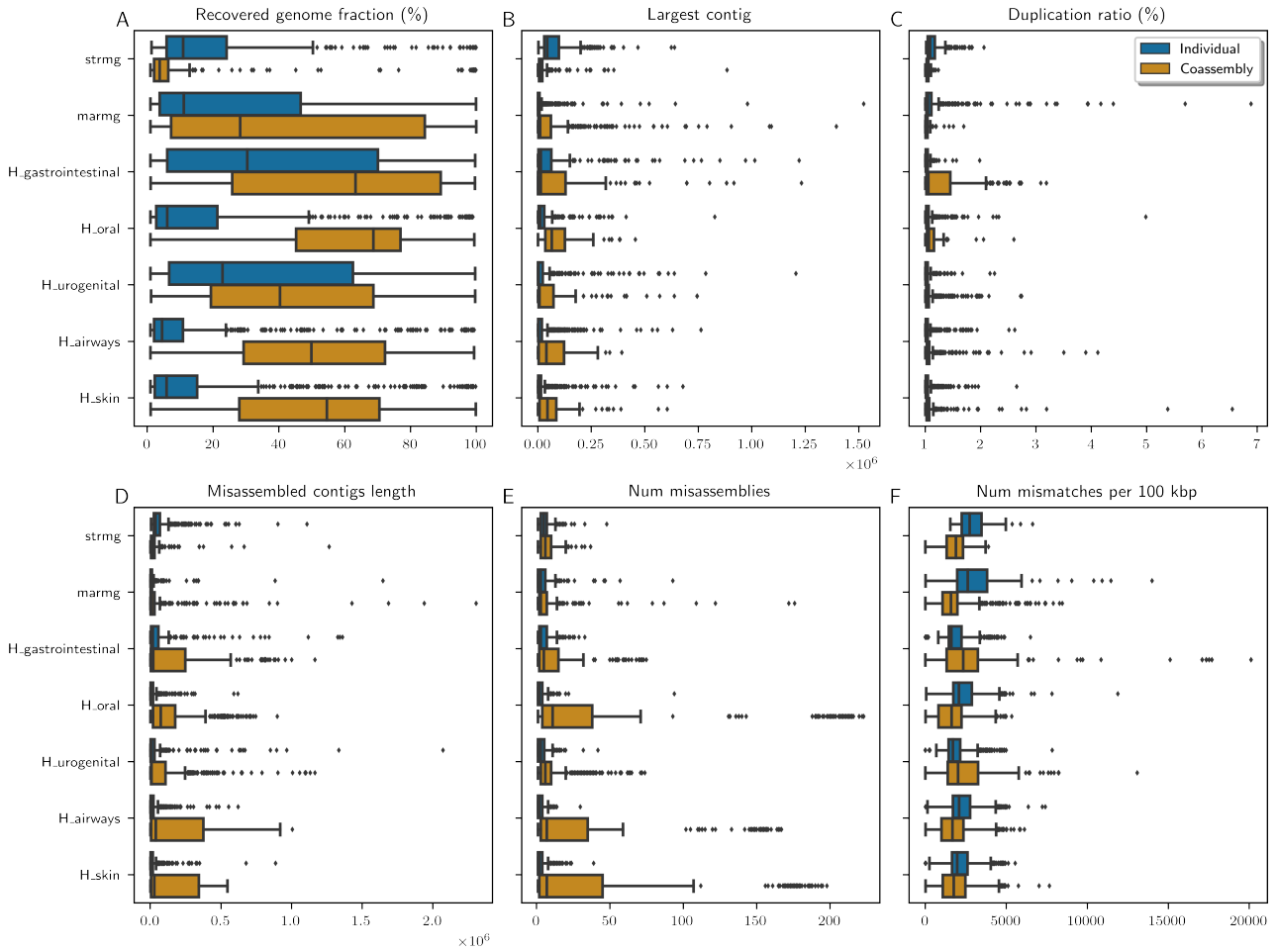


***Figure S1.*** **Differences between assembly strategies across datasets.** Same data as Fig 5, but including outliers.

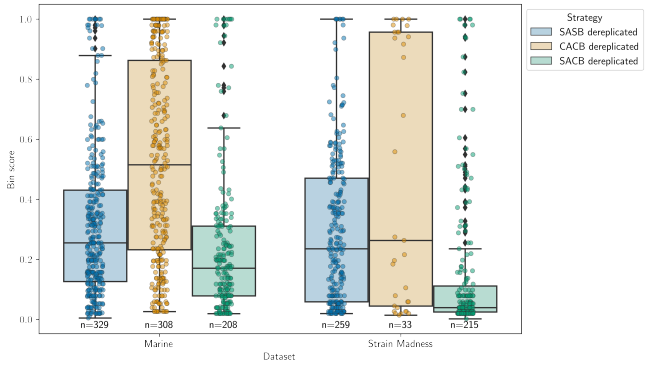


***Figure S2.*** **Boxplot of bin scores across different strategies.** Each data point is a genome bin, and Y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colours represent different strategies. Numbers underneath each bar show the number of data points for that bar. Bins sets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

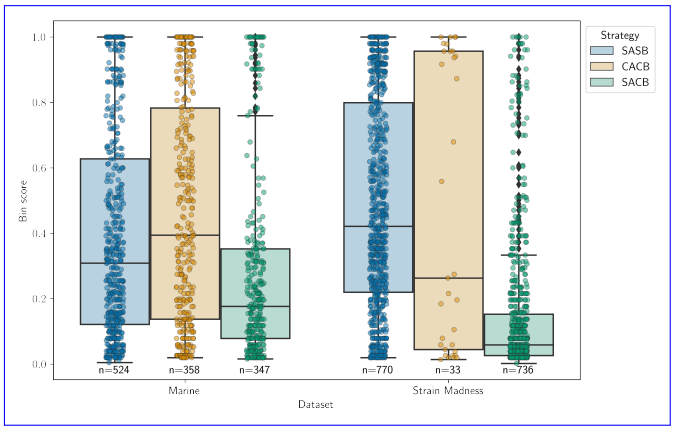


***Figure S3.*** **Boxplot of bin scores across different strategies for non-dereplicated data.** Same as **Figure S2**, but with non-dereplicated data. Each data point is a genome bin, and Y-axis depicts bin scores from 0 to 1. Columns separate datasets, and colours represent different strategies. Numbers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

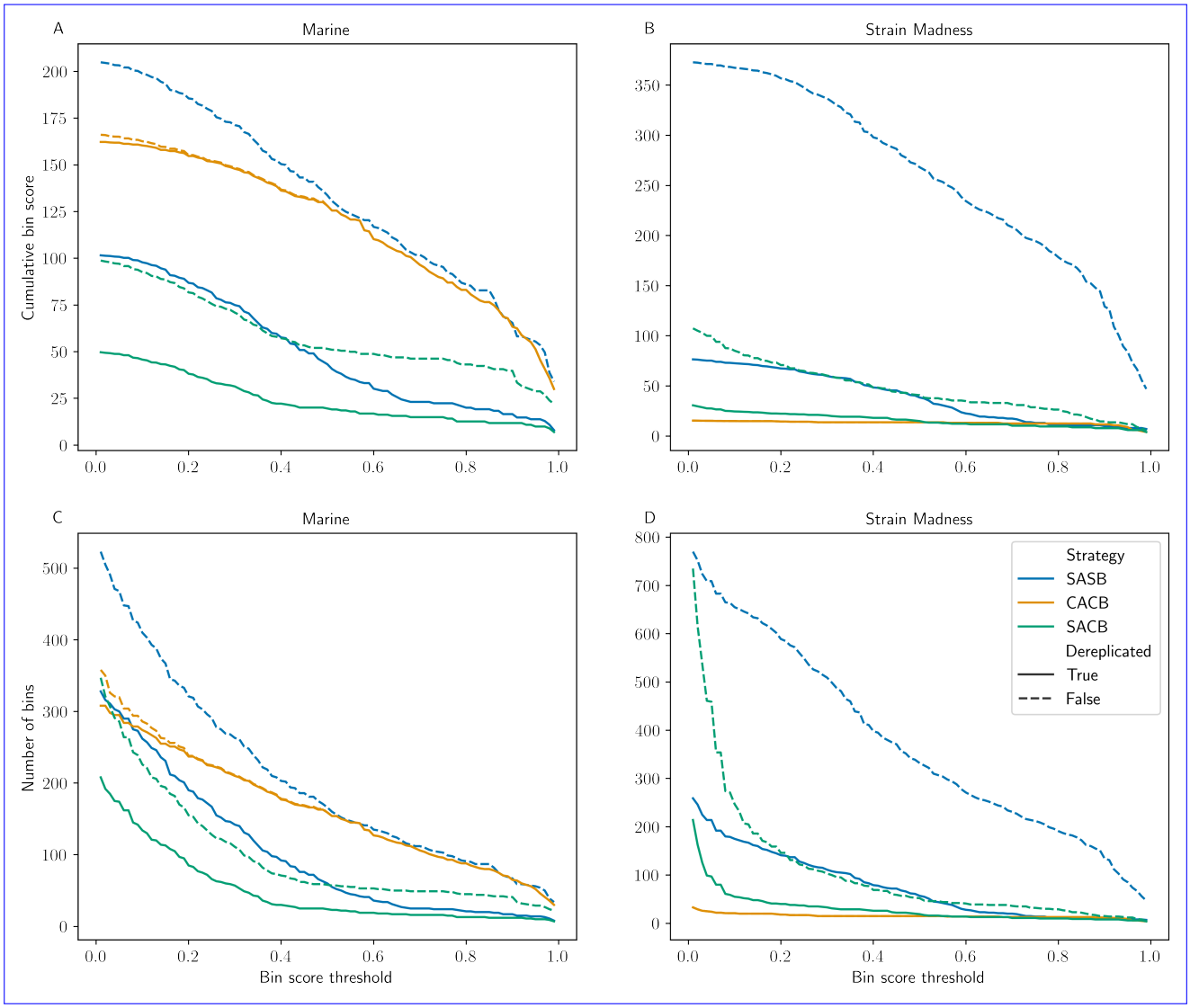


***Figure S4.*** **Cumulative bin score and number of bins between binning strategies for the Marine and Strain Madness datasets.** Solid lines show the same data as Fig 6, and dashed lines show data based on bins prior to dereplication with dRep.

***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].

| Bin ID | Bin score $S_b$ | SCG completeness | SCG redundancy | FastANI reference | GTDB Classification |
|---|---|---|---|---|---|
| metabat2.1221 | 1 | 100 | 0 | GCF_004793475.1 | Bacteroides sp002491635 |
| concoct.122 | 1 | 100 | 0 | GCF_024397795.1 | Lactobacillus intestinalis |
| vamb.S1C5590 | 1 | 100 | 0 | GCF_000614185.1 | Phocaeicola sartorii |
| metabat2.4898 | 1 | 100 | 0 | GCF_000969835.1 | Parabacteroides goldsteinii |
| concoct.156 | 0.98039216 | 98 | 0 | GCF_003030305.1 | Cutibacterium acnes |
| concoct.148 | 0.97843137 | 100 | 2 | GCF_001436695.1 | Lactobacillus taiwanensis |
| concoct.92 | 0.95686275 | 100 | 4 | GCF_014863545.1 | Paenibacillus lautus_A |
| concoct.121 | 0.95686275 | 100 | 4 | GCF_000012845.1 | Parabacteroides distasonis |
| metabat2.328 | 0.95686275 | 100 | 4 | GCF_000016825.1 | Limosilactobacillus reuteri |
| vamb.S1C971 | 0.94117647 | 94 | 0 | GCF_000392875.1 | Enterococcus faecalis |
| metabat2.3846 | 0.93678431 | 98 | 4 | GCA_009911065.1 | Ventrimonas sp009911065 |
| concoct.136 | 0.91668667 | 96 | 4 | GCF_001027105.1 | Staphylococcus aureus |
| metabat2.1266 | 0.87258904 | 96 | 8 | GCF_001544255.1 | Enterococcus_B faecium |
| concoct.115 | 0.67941176 | 71 | 2 | GCF_016758115.1 | Lactococcus sp002492185 |
| concoct.58 | 0.55843137 | 59 | 2 | GCF_013394695.1 | Streptococcus sp013394695 |
| metabat2.4512 | 0.2745098 | 27 | 0 | GCF_001729805.1 | Enterobacter roggenkampii |
| metabat2.2064 | 0.26315789 | 26 | 0 | GCF_000742135.1 | Klebsiella pneumoniae |
| metabat2.1951 | 0.21568627 | 22 | 0 | GCF_001457635.1 | Streptococcus pneumoniae |
| metabat2.3969 | 0.19607843 | 20 | 0 | GCF_011064845.1 | Citrobacter freundii |
| metabat2.1470 | 0.18421053 | 18 | 0 | GCF_000215745.1 | Klebsiella aerogenes |
| concoct.22 | 0.10526316 | 11 | 0 | GCF_001729745.1 | Enterobacter hormaechei_A |
| concoct.103_sub | 0.07894737 | 8 | 0 | | Unclassified Bacteria |
| concoct.97_sub | 0.05882353 | 6 | 0 | | Citrobacter |
| concoct.124_sub | 0.05882353 | 6 | 0 | | Unclassified Bacteria |
| concoct.27_sub | 0.04473684 | 16 | 3 | | Enterobacter |
| concoct.91_sub | 0.03921569 | 4 | 0 | GCF_001729745.1 | Enterobacter hormaechei_A |
| concoct.159 | 0.02631579 | 3 | 0 | | Unclassified Bacteria |
| concoct.64_sub | 0.02631579 | 3 | 0 | | Unclassified Bacteria |
| vamb.S1C21648 | 0.02631579 | 3 | 0 | | Unclassified |
| metabat2.3037_sub | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| concoct.13 | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| vamb.S1C7072 | 0.01960784 | 2 | 0 | | Unclassified Bacteria |
| concoct.35_sub | 0.01417112 | 86 | 53 | | Klebsiella |