A benchmark of the GraphMB metagenomic binning tool

Patrícia Costa and André Lamúrias

NOVA LINCS, NOVA School of Science and Technology
pg.costa@campus.fct.unl.pt
https://nova-lincs.di.fct.unl.pt/

Abstract. Metagenomic genome binning is a crucial step in reconstructing individual microbial genomes from complex environmental sequencing data. GraphMB is a recent tool that leverages Graph Neural Networks (GNNs) and Variational Autoencoders (VAEs) to integrate sequence features with assembly graph topology for improved binning accuracy. This study presents an extensive benchmarking of GraphMB across synthetic, semi-synthetic and real-world datasets encompassing diverse environments, sequencing and assembly technologies and binning strategies. We evaluate GraphMB's performance using standard bin quality metrics and compared against established binners. Our results show that GraphMB consistently produces a competitive or superior number of high-quality (HQ) and medium-quality (MQ) genome bins, particularly in long-read and co-assembly scenarios. We further assess the impact of optional input features and find that their benefits vary by dataset. Runtime performance scales linearly with dataset size. Overall, GraphMB demonstrates versatility and robustness across varied metagenomic conditions, though improvements in graph integration and negative pair selection could further enhance performance.

Keywords: metagenomics · genome binning · benchmarking · bioinformatics.

1 Introduction

The increasing availability of metagenomic sequencing data has significantly expanded our ability to study microbial communities in environments such as oceans, soil, and the human gut. These datasets consist of DNA fragments—strings of varying length—originating from hundreds or thousands of microorganisms. However, the raw data, without further analysis, offer little insight into the underlying microbial genomes.

A fundamental task in metagenomics is *genome binning*—the process of grouping assembled DNA fragments that likely originate from the same microbial genome. This enables the reconstruction of individual genomes, referred to as *metagenome-assembled genomes (MAGs)*, and supports analyses such as identifying the types of organisms present, understanding what functions they can

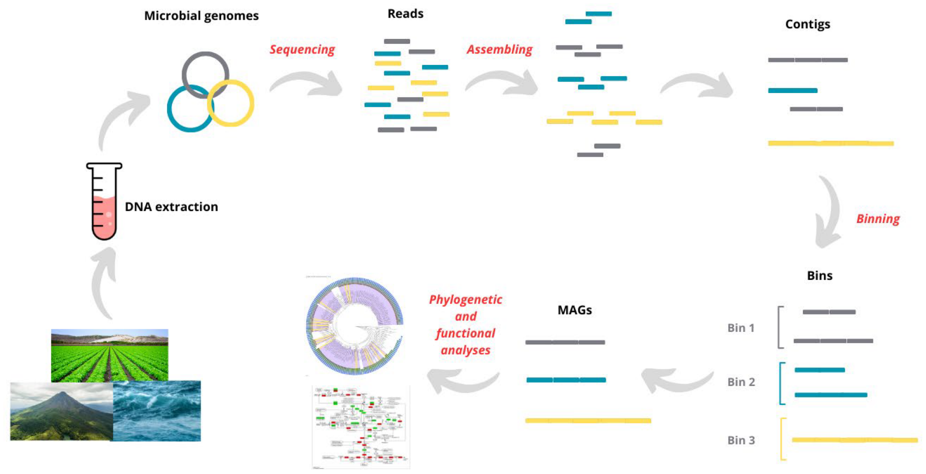

Fig. 1. Workflow for MAG recovery. DNA is extracted from environmental samples, sequenced, assembled into contigs, and binned into draft genomes (MAGs), which are then analyzed taxonomically and functionally. Adapted from [2], licensed under CC BY 4.0.

perform, and discovering new species[1]. Figure 1 provides an overview of this process, from sample collection and sequencing to MAG recovery and analysis.

Traditional binning approaches rely on sequence composition (e.g., k-mer profiles) and abundance patterns across samples. While effective in many cases, they often struggle with fragmented assemblies, low-abundance organisms, or closely related strains [3]. To address these challenges, recent tools have proposed incorporating *assembly graph information*—a representation of contig overlaps generated during the assembly process (Figure 1). This graph captures the structural context of contigs and can provide valuable information for more accurate binning.

GraphMB is a recent tool that exemplifies this direction. It combines contig features (such as k-mer composition and abundance patterns) with *graph neural networks (GNNs)* and *variational autoencoders (VAEs)* to learn informative embeddings of contigs by considering both their individual features and their connectivity in the assembly graph [9]. These embeddings are then clustered to generate genome bins. Although promising, GraphMB has been evaluated only on a limited set of datasets, raising questions about its robustness and versatility when applied to diverse scenarios.

This project aims to investigate the following question:

Is GraphMB versatile enough to handle the diversity of current metagenomic datasets, including different sequencing technologies, environments, and experimental designs?

To answer this, we conduct a detailed benchmarking study of GraphMB using real, synthetic, and semi-synthetic datasets. We assess the quality of the resulting genome bins, the impact of different assembly strategies, and the role of optional inputs such as SCGs and assembly graphs, as well as the tool's runtime performance.

Our main contributions are:

– A systematic evaluation of GraphMB across a diverse set of datasets, including different sequencing technologies, assemblies and binning strategies.
– A study of how optional inputs (e.g., depth profiles, SCGs, assembly graphs) influence binning results.
– An empirical analysis of runtime performance under GPU acceleration.

2 Domain Background

Metagenomics is the study of genetic material recovered directly from environmental samples, without the need to culture the organisms in the laboratory. This is especially valuable for studying microbial communities, where a large portion of organisms are difficult or impossible to grow in isolation.

Sequencing is the process of reading the DNA molecules from a sample and converting them into digital form. Modern sequencing platforms output millions of small DNA fragments called **reads**, which are strings of characters from the alphabet {A, C, G, T}, each character representing a nucleotide. These reads are typically stored in standardized file formats such as FASTA or FASTQ; an example of the FASTA format is shown in Figure 2.

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTCTTTTCTTATCATTGACATTTAAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCGGGCTCCGGCCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGTGCTCTTCTGCAAAAGTAGCAAAATGTTCCACTCCTAAGAGTGGACTTCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTCACGACATCCACGCTTGGGAAAG
TCCGTACCCGCGCCTGGAGCGCTTAAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

Fig. 2. Example of a sequencing read stored in the FASTA format. Each read consists of a header line (beginning with >) followed by the nucleotide sequence. Adapted from [4].

To reconstruct longer genomic regions, these reads are computationally assembled into larger sequences called **contigs** using *assemblers*. Contigs are uninterrupted stretches of sequence that likely come from the same genome, but the source organism is not known at this stage. The set of all contigs produced from a metagenomic sample is called a **metagenome assembly**. In addition to the assembled contigs themselves, one can also estimate how many sequencing reads map to each contig across one or multiple samples. This measure, referred to as **coverage depth** (or simply **depth**, sometimes also called **abundance** or **co-abundance**), provides information about the relative abundance of each contig in the sample(s), and is often used as a feature for genome binning.

Some assemblers (e.g., metaFlye [5], metaSPAdes [6]) also produce an **assembly graph**, which represents how sequencing reads overlap and are connected. These graphs are built using graph-based algorithms to help piece together contigs from individual reads. However, not all regions of the genome can be unambiguously resolved—due to repeats, sequencing errors, or low coverage—so some connections remain uncertain. As a result, the assembly graph includes edges between contigs that reflect these unresolved relationships. This structural information can be useful for downstream analysis, such as genome binning.

Another useful input for binning is the presence of **single-copy marker genes (SCGs)**—genes that are typically found exactly once in a genome and are conserved across related organisms. Identifying which contigs contain which SCGs provides important clues for separating contigs from different genomes. For example, if two contigs contain many of the same SCGs, it is unlikely they belong to the same genome, and they can be used to train binning models with negative examples.

Genome binning is the process of grouping contigs that likely originate from the same organism into bins, which are used to reconstruct *metagenome-assembled genomes (MAGs).* These bins can then be taxonomically classified and functionally annotated, enabling insights into the microbial community.

3 Related Work

3.1 Sequencing Technologies and Their Impact on Binning

Genome binning is fundamentally influenced by the characteristics of the sequencing data used in metagenomic studies. The sequencing process generates DNA reads from environmental samples, which vary in length, error profile, and coverage depending on the technology. These properties affect how accurately contigs are assembled and, subsequently, how well they can be binned into genomes.

Illumina sequencing[15] produces short reads (typically 100–300 bp) with very low error rates. These reads provide high-quality coverage and are well-suited for accurate taxonomic profiling and abundance estimation. However, due to their short length, they often result in fragmented assemblies, making it harder to correctly bin contigs, especially in complex communities with closely related strains.

Pacific Biosciences (PacBio)[16] and **Oxford Nanopore Technologies (ONT)**[17] are long-read sequencing platforms. PacBio reads can range from 10–30 kbp, while ONT reads can exceed 100 kbp. These longer reads improve assembly continuity and can resolve repeat regions and structural variations more effectively than short reads. However, they tend to have higher error rates, particularly ONT reads, which can complicate downstream analyses.

In metagenomics, combining technologies (hybrid assemblies) is a common strategy. Short reads provide base-level accuracy, while long reads offer continuity. For genome binning, long-read assemblies tend to produce longer, more informative contigs that are easier to cluster into accurate genome bins.

Different sequencing platforms, therefore, not only influence the quality and contiguity of the assembly but also the input features (like k-mer composition and coverage depth) used by binning algorithms. Long-read technologies are particularly beneficial for binners that leverage structural information, such as assembly graphs, as seen with tools like GraphMB.

3.2 Binning Strategies: Single, Multi, and Co-Assembly

Metagenomic binning strategies can differ not only in the algorithms they employ but also in how they use sequencing data across multiple samples. Three common strategies are [14]:

- **Single-Assembly Binning:** Each sample is assembled independently, and binning is performed using only its own contigs and depth profiles. This approach preserves sample-specific features but may fail to capture shared low-abundance genomes.
- **Multi-Assembly Binning:** Each sample is still assembled individually, but the binning process uses depth profiles from multiple samples to improve co-abundance estimation. This allows better discrimination of contigs from different organisms based on shared abundance patterns.
- **Co-Assembly Binning:** Reads from multiple samples are combined into a single assembly before binning. This increases the chance of assembling complete genomes, especially for low-abundance species, but may merge closely related strains, complicating binning.

These approaches present trade-offs between strain resolution, completeness, and the risk of introducing *chimeric contigs*—artificial contigs formed by incorrectly joining sequences from different organisms during assembly, which can lead to errors in binning and downstream analyses.

3.3 Assemblers Used in This Study

We tested GraphMB on assemblies produced by four widely used tools: **metaFlye**[5], **metaSPAdes**[6], **MEGAHIT**[7], and **OPERA-MS**[8]. These assemblers perform the critical task of *assembly*, which is responsible for generating contigs from sequencing reads. The first three are capable of outputting assembly graphs, although in this study only graphs generated by metaFlye were used. Regarding sequencing input, MEGAHIT and metaSPAdes are designed for short-read data (e.g., Illumina), while metaFlye targets long-read technologies such as Oxford Nanopore (ONT) and PacBio. OPERA-MS is a hybrid assembler that combines short and long reads, but does not produce an assembly graph.

3.4 Binning Tools Compared in This Study

Table 1 summarizes the genome binning tools evaluated in our benchmark. These tools were selected for their widespread use and diversity of approaches, which include probabilistic modeling, variational autoencoders, and graph-based learning. We also include *GraphMB*, the method under evaluation in this work.

Table 1. Overview of binning tools included in our benchmark.

Tool	**Description**	**Year**	**Ref.**
GraphMB	Graph-based method that uses Graph Neural Networks (GNNs) and Variational Autoencoders (VAEs) to integrate sequence features with assembly graph connectivity.	2022	[9]
CONCOCT	Bins contigs based on Gaussian mixture models using k-mer composition and coverage across multiple samples.	2014	[10]
MaxBin 2	Uses an Expectation-Maximization algorithm with sequence composition and abundance to assign contigs to bins.	2016	[11]
MetaBAT 2	Employs probabilistic distances between contigs based on tetranucleotide frequency and abundance for adaptive binning.	2019	[12]
VAMB	Learns contig embeddings using a Variational Autoencoder trained on k-mer and abundance profiles, followed by clustering.	2021	[13]

3.5 Evaluation Metrics and Tools

To assess the quality of the bins produced by GraphMB and other binning tools, we adopted standard metrics that reflect both the completeness and specificity of genome reconstruction.

Completeness quantifies how much of a reference genome is recovered within a bin. Higher completeness indicates that most of the genome's sequences have been captured.

Contamination estimates the extent to which a bin includes sequences from other genomes. It is typically inferred based on the number of duplicated single-copy marker genes.

For real and semi-synthetic datasets that lack ground truth genome labels, we define **Purity** as $1 -$ Contamination, providing an approximate measure of bin specificity. For fully synthetic datasets, where true genome assignments are available, purity can be computed directly by comparing predicted bins to the known references.

We also categorize bins by quality based on these metrics:

– **Medium Quality (MQ)** bins have completeness greater than 50% and contamination below 10%.

– **High Quality (HQ)** bins are defined as those with completeness greater than 90% and contamination below 5%. Although some definitions of HQ bins require additional features (e.g., rRNA or tRNA genes), we adopt this simplified definition to avoid introducing additional layers of analysis.

While completeness, contamination, and purity are valuable metrics—particularly for tool developers—the number of HQ bins is often the most relevant from a practical perspective, as it reflects how many near-complete genomes can be reliably recovered. Therefore, the more HQ bins recovered, the better.

To compute these metrics, we employed two tools depending on the dataset type:

AMBER [18] was used for synthetic datasets with known ground truth labels. It directly compares predicted bins to reference genomes and reports metrics such as completeness, purity, and the number of high-quality bins, allowing for rigorous benchmarking in controlled conditions.

CheckM2 [19] was applied to real and semi-synthetic datasets without ground truth. It estimates completeness and contamination using machine learning models trained on lineage-specific SCGs. While less precise than ground truth comparisons, CheckM2 enables quality assessment in practical, real-world scenarios.

This combination of metrics and tools provides a robust framework for evaluating binning performance across both synthetic and natural metagenomic samples.

4 Experimental Design

4.1 Datasets Used for Benchmarking

To evaluate the performance and versatility of GraphMB, we selected a diverse set of metagenomic datasets encompassing synthetic, semi-synthetic, and real-world scenarios (the assemblies used for these datasets are summarized in Table 2). This variety ensures a comprehensive assessment of the tool across different levels of taxonomic complexity, sequencing strategies, and ecological contexts.

From the CAMI (Critical Assessment of Metagenome Interpretation) benchmarking challenges, we used three datasets. The **CAMI 1 high-complexity dataset** [20] is the simplest of the CAMI datasets used in this study—it simulates a community with high species and strain diversity, but includes only a single co-assembly based on short reads and is less heterogeneous in terms of experimental design.

In contrast, the **CAMI 2 marine** and **plant-associated** datasets [21] are significantly more complex and realistic. They include a wide variety of reference genomes and were assembled using multiple strategies, including single-sample and co-assembly approaches. These datasets also incorporate sequencing data from different technologies (e.g., Illumina, PacBio) and include assemblies generated by multiple assembler tools submitted by participants in the CAMI challenge. This makes them ideal for assessing a tool's robustness across diverse conditions.

We also included the **MetaHIT dataset** [13], a semi-synthetic dataset based on human gut microbiota. It was constructed by combining reads from known bacterial isolates and assembling them with minimal sequencing errors (referred as error-free in Table 2), offering a realistic yet controlled benchmark with available reference labels.

To test GraphMB on real, complex microbial communities, we used three additional datasets. The **AalE dataset**, from a wastewater treatment plant (WWTP), and the **Soil dataset**, representing a highly diverse and heterogeneous soil microbiome, were both introduced in the original GraphMB study [9]. These datasets present unique challenges due to the lack of complete reference genomes and the variability in sequencing depth and community composition.

Finally, we included the **Cow Rumen dataset**, a real dataset sequenced using Oxford Nanopore Technologies (ONT), which provides long reads and presents a low-bias, strain-resolved view of a complex microbial community [5]. This dataset adds further diversity to the benchmark by incorporating long-read sequencing data from an agricultural environment.

This selection of datasets enables a robust evaluation of GraphMB under a wide range of conditions, from controlled synthetic environments to ecologically rich real-world samples.

4.2 Optional Input Features and Their Impact

GraphMB supports three optional input features that can influence its binning performance: **coverage depth**, **assembly graphs**, and **single-copy marker gene (SCG) annotations**. These inputs are explained in more detail in section 2. Depth profiles capture co-abundance patterns across samples, assembly graphs offer structural context between contigs, and SCGs enable biologically informed constraints during model training. While not required, these features can enhance bin quality in complex scenarios, though they may also increase runtime (especially markers).

4.3 Negative Pair Filtering Based on SCG Overlap

An experimental feature was added to GraphMB to control the number of shared single-copy genes (SCGs) between contigs when generating negative training pairs. Two new arguments were introduced: one specifying the **minimum** number of SCGs required for two contigs to be considered a negative pair, and another specifying the **maximum**. While this approach yielded improved binning results in a few cases, the outcomes were inconsistent across datasets, and no universally optimal threshold values were identified.

Despite its limited impact on accuracy, this feature proved useful for reducing memory consumption in large datasets. By filtering out a substantial number of highly redundant negative pairs, it allowed GraphMB to run more efficiently on memory-constrained systems.

4.4 Computational Efficiency Evaluation

To assess GraphMB's computational efficiency, we recorded the total execution time of each run, including both training and clustering phases. All experiments were conducted with GPU acceleration enabled. The hardware used consisted of an **AMD EPYC 9124** 16-core processor, **256 GiB of system memory**, and an **NVIDIA L4 GPU**. These conditions provide a realistic baseline for evaluating the tool's runtime performance across datasets of varying size and complexity.

5 Results and Discussion

In this section, we analyze the performance of GraphMB across a variety of datasets. Our goal is to evaluate both the versatility and robustness of GraphMB under diverse experimental conditions.

All results presented in this section were obtained after applying a filtering step to the binning outputs: bins with a total length below 200 kbp were excluded. This threshold helps remove spurious, incomplete bins that can distort quality metrics and comparisons. The same filtering was applied to GraphMB results and to the external results used for other binners.

It is important to note that only the GraphMB results were produced as part of this study. Assemblies and binning outputs for other tools (e.g., MetaBAT 2, VAMB, MaxBin 2) were obtained from publicly available benchmarks and evaluation platforms[5, 9, 13, 21]. As such, with the exception of the AalE, Soil and Cow Rumen datasets, we did not have access to the corresponding assembly graph files for these assemblies, and therefore they were not tested with graph-based features.

Table 2 summarizes the main characteristics of most of the assemblies used in our experiments. These factors are important to consider when interpreting GraphMB's binning performance and runtime behavior, as they can significantly affect both quality and computational requirements.

5.1 Execution Time and Efficiency

Figure 3 analyzes GraphMB's runtime behavior. Plot **3A** shows that execution time increases linearly with the number of contigs, with runs using SCGs being consistently slower. Despite the added overhead, the use of markers can yield a higher number of HQ and MQ bins in most cases. Notably, the CAMI High (s12) dataset demonstrates both high efficiency and output quality, whereas complex real datasets like Soil (r3) yield lower binning throughput, as seen in plots **3B** and **3C**.

5.2 Binning Performance and Ranking Comparison

Figure 4 summarizes the performance of GraphMB compared to other widely used binners (CONCOCT, MaxBin 2, MetaBAT 2 and VAMB) (Table 1) across

Table 2. Summary of synthetic (code sx), real and semi-synthetic (code rx) datasets and their assembly metrics. **GA** - Gold Standard Assembly

Code	**Name**	**Co-Assembly**	**Sequencer**	**Assembler**	**Length (Gbp)**	**N50 (kbp)**	**# Contigs**	**# Samples**
s1	Marine	NO	Illumina	GA	0.415	5	125197	1
s2	Marine	YES	Illumina	GA	2.079	29	350238	10
s3	Marine	YES	Illumina	MEGAHIT	0.953	5	303974	10
s4	Marine	NO	PacBio	GA	1.035	8	168481	1
s5	Marine	YES	PacBio	GA	2.595	83	149860	10
s6	Plant	NO	Illumina	GA	0.207	31	35258	1
s7	Plant	YES	Illumina	GA	1.487	26	311881	21
s8	Plant	NO	PacBio	GA	0.584	6	119814	1
s9	Plant	YES	PacBio	GA	2.600	19	367373	21
s10	Plant	NO	ONT	GA	0.241	25	41392	1
s11	Plant	YES	ONT	GA	1.087	41	163877	21
s12	CAMI-High	YES	Illumina	GA	2.803	249	42038	5
r1	AalE	YES	ONT	metaFlye	1.906	79	45843	4
r2	MetaHIT	YES	Illumina	error-free	1.144	8	195601	264
r3	Soil	NO	ONT	metaFlye	1.919	93	47062	1
r4	Cow Rumen	YES	PacBio	metaFlye	1.254	40	52653	23

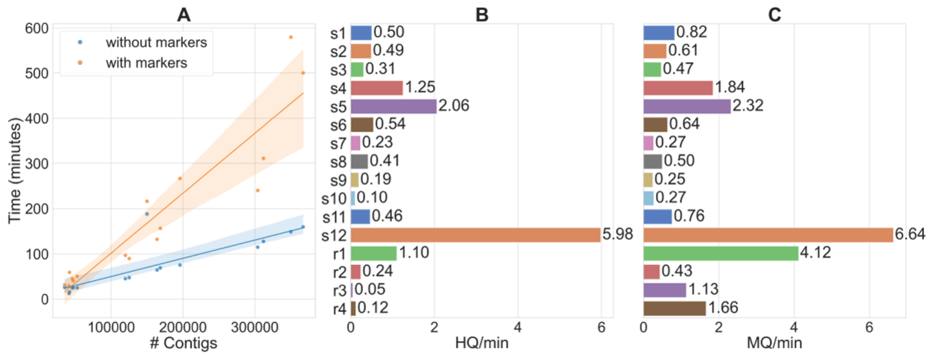

Fig. 3. Plot **A** illustrates how GraphMB's execution time scales with the number of contigs in the input assembly, comparing runs with and without the use of a marker gene (SCG) file. The accompanying bar plots depict the number of High Quality (HQ) (**B**) and Medium Quality (MQ) (**C**) bins obtained per minute of execution time of the best run. In those best runs, all datasets, with the exception of soil (r3), were using SCGs. The labels correspond to the code in Table 2.

four datasets. These specific datasets (s2, s7, s12, and r2) were selected because public benchmark results for the other binners were available for them. GraphMB demonstrates competitive results in most metrics, particularly in the number of High Quality (HQ) bins. More notably, it scored considerably better in completeness and the proportion of HQ bins for datasets s12 and r2.

To consolidate performance across different metrics, we calculated an overall ranking score for each binner (Figure 5). For each dataset, tools were ranked by their average completeness, purity, HQ bin percentage, and total HQ bins; these ranks were summed to obtain a final score (lower is better). GraphMB ranked 1st in every dataset except the more complex plant-associed one (s7).

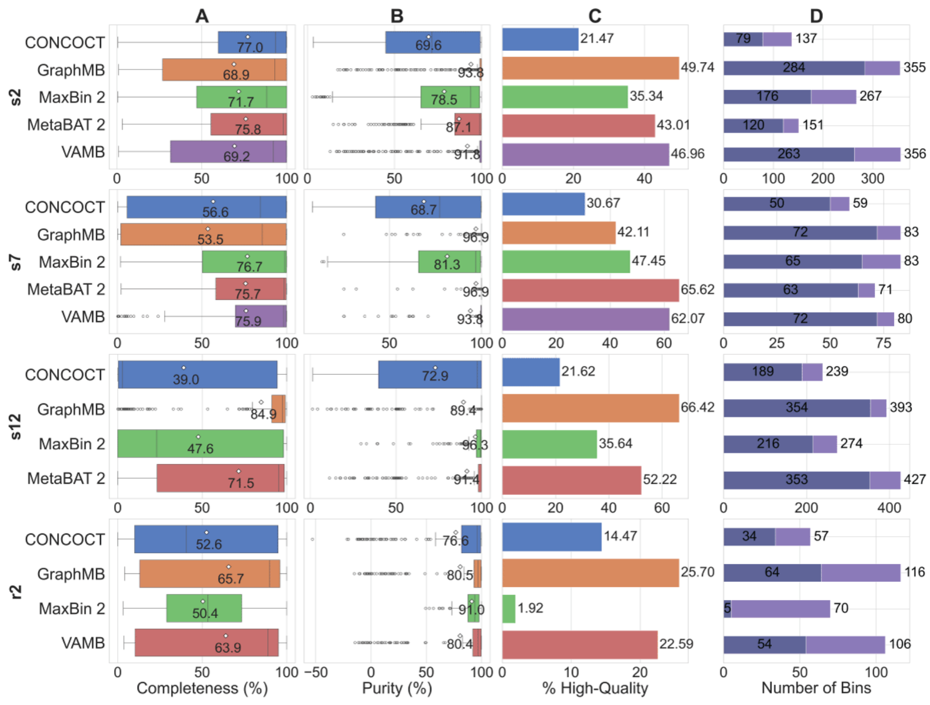

Fig. 4. Binning performance comparison across datasets s2, s7, s12 and r2 (see Table 2 for the corresponding code). **A** - bin completeness distribution (the diamond symbol and number indicate the mean), **B** - bin purity distribution (like in **A**,the diamond symbol and number indicate the mean), **C** - percentage of HQ bins present in the bins produced (after filtering), **D** - number of HQ bins (dark color) and MQ bins (light color).

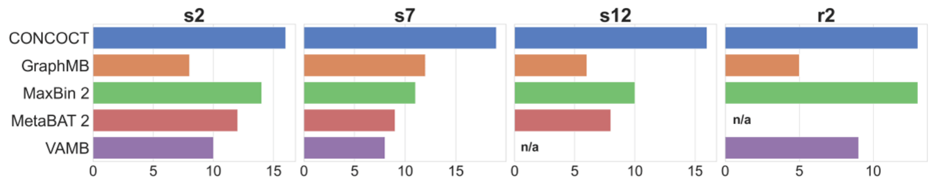

Fig. 5. Overall ranking scores of each binner across datasets s2, s7, s12 and r2 (see Table 2 for the corresponding code). The smaller the score the better.

5.3 Impact of Binning Strategy on GraphMB Performance

Figure 6 shows the number of HQ and MQ bins obtained across various binning strategies and sequencing technologies (see sections 3.1 and 3.2). Co-assembly consistently yielded the highest number of bins, particularly when depth information was included. Interestingly, `single_no_d` often outperformed `single` (for HQ bins), suggesting that depth profiles derived from a single sample may introduce noise rather than useful signal.

In terms of sequencing technologies, PacBio-based assemblies produced the best results overall, followed by Illumina. ONT-based assemblies consistently yielded the lowest number of high-quality bins, which may be attributed to the higher error rates traditionally associated with ONT reads—despite recent improvements in accuracy.

Overall, when multiple samples are available, co-assembly with depth profiles appears to be the most effective strategy. In contrast, for scenarios with only a single sample, performing a single assembly without using depth information seems to be the safer choice.

5.4 Impact of Assembly Strategy and Assembler on Binning Performance

Figure 7 examines how the choice of assembly affects binning performance. In Plot **7A**, we compare binners using two assemblies derived from the same CAMI Marine dataset: the gold standard (s2) and a real assembly produced with MEGAHIT (s3). As expected, performance is lower on the MEGAHIT assembly due to its imperfections. However, the relative drop varies across binners. MetaBAT 2 and CONCOCT show a moderate decrease in the number of HQ and MQ bins, while GraphMB and VAMB suffer a larger decline, indicating that some methods are more robust to imperfect assemblies.

Plots **7B** and **7C** focus on GraphMB alone, comparing its performance across different assemblers (see subsection 3.3) without optional input features. These assemblies were submitted by participants of the CAMI 2 challenge, providing a diverse set of real-world assembly strategies. metaSPAdes seem to have the best overall performance. metaFlye performs particularly well for the Plant dataset. In contrast, the Marine assembly, generated using Flye (not metaFlye), shows lower performance—likely due to Flye not being tailored for metagenomics.

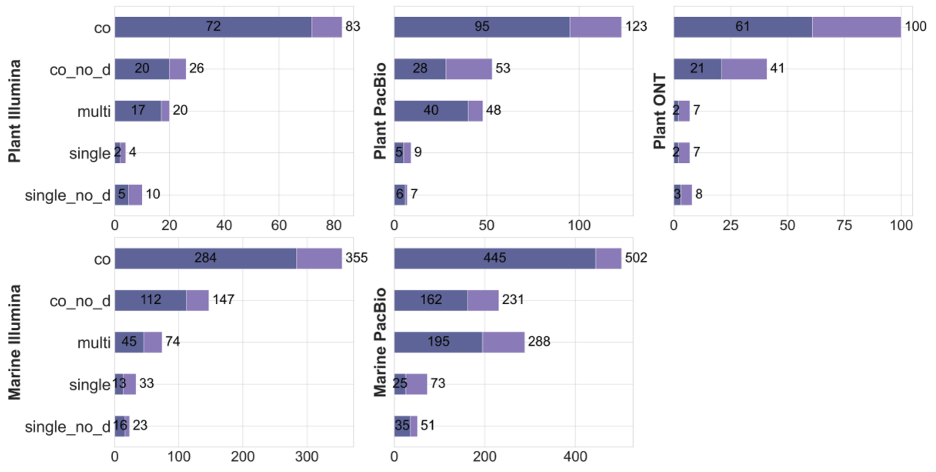

Fig. 6. Number of HQ and MQ bins, produced by GraphMB, across different datasets and binning strategies (as referred in subsection 3.2). **co** - co-assembly, **co_no_d** - assembly from multiple samples but with no coverage depth information, **multi** - multi-assembly, **single** - single-assembly, **single_no_d** - assembly from a single sample but with no coverage depth information.

These results emphasize the combined influence of both sequencing technology and assembler choice on downstream binning quality.

5.5 Impact of Optional Input Features on Real Datasets

Figure 8 shows how different combinations of optional inputs affect GraphMB's performance on real datasets. In general, providing more auxiliary information can improve results, but the optimal combination varies by dataset.

If we consider the best result as the one yielding the highest number of HQ bins (using MQ bins as a tiebreaker), then the most effective configuration differs across datasets. The Cow Rumen dataset performed best when all optional features were used (**gmd**), AalE achieved its highest results using only SCGs and depth profiles (**md**), and surprisingly, Soil achieved its best performance without any optional inputs (**f**).

Interestingly, the use of the assembly graph (**g**) appears to decrease performance in most scenarios. It's worth noting that these results are based solely on graphs generated by the `metaFlye` assembler; graphs from other assemblers may yield different outcomes.

6 Conclusion

In this study, we conducted an extensive benchmarking of GraphMB, a graph-based genome binning tool, across a diverse collection of synthetic, semi-synthetic,

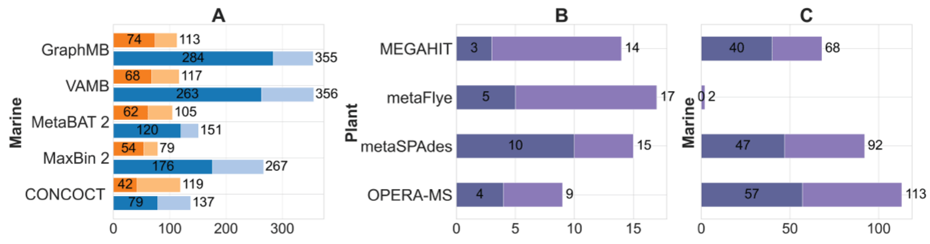

Fig. 7. Plot **A** compares the number of HQ and MQ bins produced by different binners from datasets s2 (blue, gold standard assembly) and s3 (orange, MEGAHIT assembly) (see Table 2 for codes), which are different assemblies from the same samples. Plots **B** and **C** compare the number of HQ and MQ bins produced by GraphMB from assemblies from different assemblers across multiple samples, all evaluated without optional input features (e.g., SCGs or depth profiles). The MEGAHIT and metaSPAdes assemblies were generated from Illumina short reads. The metaFlye Plant assembly was produced from ONT long reads, while the Marine assembly uses Flye (not specifically designed for metagenomics) with PacBio long reads. OPERA-MS combines both short and long reads in a hybrid assembly approach.

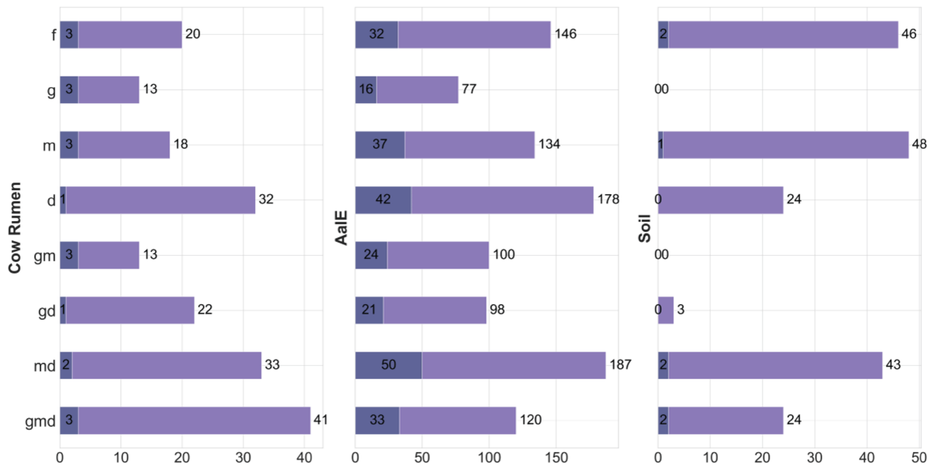

Fig. 8. Number of HQ and MQ bins produced by GraphMB across real datasets, using different combinations of optional input features. Each bar is labeled with a letter code representing the features used: **f** – no optional features, **g** – assembly graph, **m** – marker genes (SCGs), **d** – coverage depth.

and real metagenomic datasets. The results highlight GraphMB's strong performance in recovering high- and medium-quality bins, particularly in co-assembly settings where depth information is available. Its ability to integrate graph-based structural information and deep learning techniques gives it a notable edge in complex scenarios, often outperforming traditional binners in datasets with long reads or high strain diversity.

GraphMB's main advantages include its flexibility in handling different sequencing technologies, its competitive binning accuracy, and its support for optional features that can enhance results in certain conditions. In terms of efficiency, it scales linearly with assembly size and remains practical for large datasets, especially when optimized configurations are used.

However, some weaknesses remain. The quality of the assembly graph strongly affects performance, and in some cases (e.g., metaFlye-generated graphs), the inclusion of this feature slightly degrades results. Furthermore, the tool is computationally demanding when using marker gene information.

Overall, GraphMB is a versatile and promising tool for metagenomic genome recovery. Future work may explore optimizing its negative pair selection and extending compatibility with a broader range of assemblers, including different types of assembly graph.

References

1. Quince, C., Walker, A., Simpson, J., et al.: Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology* **35**, 833–844 (2017). https://doi.org/10.1038/nbt.3935
2. Mirete, S., Sánchez-Costa, M., Díaz-Rullo, J., González de Figueras, C., Martínez-Rodríguez, P., González-Pastor, J.E.: Metagenome-Assembled Genomes (MAGs): Advances, Challenges, and Ecological Insights. *Microorganisms* **13**(5), 985 (2025). https://doi.org/10.3390/microorganisms13050985
3. Sczyrba, A., Hofmann, P., Belmann, P., et al.: Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nature Methods* **14**, 1063–1071 (2017). https://doi.org/10.1038/nmeth.4458
4. CompGenomR: Computational Genomics with R, https://compgenomr.github.io/book/fasta-and-fastq-formats.html, last accessed 2025/05/19
5. Kolmogorov, M., Bickhart, D.M., Behsaz, B., Gurevich, A., Rayko, M., Shin, S.B., Kuhn, K., Yuan, J., Polevikov, E., Smith, T.P.L., Pevzner, P.A.: metaFlye: scalable long-read metagenome assembly using repeat graphs. Nat. Methods **17**, 1103–1110 (2020). https://doi.org/10.1038/s41592-020-00971-x
6. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A.: metaSPAdes: a new versatile metagenomic assembler. Genome Res. **27**(5), 824–834 (2017). https://doi.org/10.1101/gr.213959.116
7. Li, D., Liu, C.-M., Luo, R., Sadakane, K., Lam, T.-W.: MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics **31**(10), 1674–1676 (2015). https://doi.org/10.1093/bioinformatics/btv033

8. Bertrand, D., Shaw, J., Kalathiyappan, M. et al.: Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. Nat Biotechnol **37**, 937–944 (2019). https://doi.org/10.1038/s41587-019-0191-2
9. Lamurias, A., Sereika, M., Albertsen, M., Hose, K., Nielsen, T.D.: Metagenomic binning with assembly graph embeddings. *Bioinformatics* **38**(19), 4481–4487 (2022). https://doi.org/10.1093/bioinformatics/btac557
10. Alneberg, J., Bjarnason, B.S., de Bruijn, I., Schirmer, M., Quick, J., Ijaz, U.Z., Lahti, L., Loman, N.J., Andersson, A.F., Quince, C.: Binning metagenomic contigs by coverage and composition. *Nature Methods* (2014). https://doi.org/10.1038/nmeth.3103
11. Wu, Y.-W., Simmons, B.A., Singer, S.W.: MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics* **32**(4), 605–607 (2016). https://doi.org/10.1093/bioinformatics/btv638
12. Kang, D.D., Li, F., Kirton, E., Thomas, A., Egan, R., An, H., Wang, Z.: MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**, e7359 (2019). https://doi.org/10.7717/peerj.7359
13. Nissen, J.N., Johansen, J., Allesøe, R.L., et al.: Improved metagenome binning and assembly using deep variational autoencoders. *Nature Biotechnology* **39**, 555–560 (2021). https://doi.org/10.1038/s41587-020-00777-4
14. Han, H., Wang, Z., Zhu, S.: Benchmarking metagenomic binning tools on real datasets across sequencing platforms and binning modes. Nat. Commun. **16**, 2865 (2025). https://doi.org/10.1038/s41467-025-57957-6
15. Margraf, R.L., Durtschi, J.D., Dames, S., Pattison, D.C., Stephens, J.E., Mao, R., Voelkerding, K.V.: Multi-sample pooling and Illumina Genome Analyzer sequencing methods to determine gene sequence variation for database development. J. Biomol. Tech. **21**(3), 126–140 (2010)
16. Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., et al.: Real-time DNA sequencing from single polymerase molecules. Science **323**(5910), 133–138 (2009). https://doi.org/10.1126/science.1162986
17. Jain, M., Olsen, H.E., Paten, B., Akeson, M.: The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biology **17**, 239 (2016). https://doi.org/10.1186/s13059-016-1103-0
18. Meyer, F., Hofmann, P., Belmann, P., Garrido-Oter, R., Fritz, A., Sczyrba, A., McHardy, A.C.: AMBER: Assessment of Metagenome BinnERs. GigaScience **7**, giy069 (2018). https://doi.org/10.1093/gigascience/giy069
19. Chklovski, A., Parks, D.H., Woodcroft, B.J., et al.: CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. Nat Methods **20**, 1203–1212 (2023). https://doi.org/10.1038/s41592-023-01940-w
20. Sczyrba, A., Hofmann, P., Belmann, P., et al.: Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. Nat Methods **14**, 1063–1071 (2017). https://doi.org/10.1038/nmeth.4458
21. Meyer, F., Fritz, A., Deng, Z.L., et al.: Critical Assessment of Metagenome Interpretation: the second round of challenges. Nat Methods **19**, 429–440 (2022). https://doi.org/10.1038/s41592-022-01431-4