Bioinformatics, 2023, **39**, i21–i29
https://doi.org/10.1093/bioinformatics/btad209
ISMB/ECCB 2023

OXFORD

SemiBin2: self-supervised contrastive learning leads to better MAGs for short- and long-read sequencing

Shaojun Pan [1,2], **Xing-Ming Zhao** [1,2,3,4,*], **Luis Pedro Coelho** [1,2,*]

[1]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China
[2]Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence, Ministry of Education, Shanghai 200433, China
[3]MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China
[4]Zhangjiang Fudan International Innovation Center, Shanghai 201203, China

*Corresponding author. E-mail: xmzhao@fudan.edu.cn (X.-M.Z.); luispedro@big-data-biology.org (L.P.C.)

Abstract

Motivation: Metagenomic binning methods to reconstruct metagenome-assembled genomes (MAGs) from environmental samples have been widely used in large-scale metagenomic studies. The recently proposed semi-supervised binning method, SemiBin, achieved state-of-the-art binning results in several environments. However, this required annotating contigs, a computationally costly and potentially biased process.

Results: We propose SemiBin2, which uses self-supervised learning to learn feature embeddings from the contigs. In simulated and real datasets, we show that self-supervised learning achieves better results than the semi-supervised learning used in SemiBin1 and that SemiBin2 outperforms other state-of-the-art binners. Compared to SemiBin1, SemiBin2 can reconstruct 8.3–21.5% more high-quality bins and requires only 25% of the running time and 11% of peak memory usage in real short-read sequencing samples. To extend SemiBin2 to long-read data, we also propose ensemble-based DBSCAN clustering algorithm, resulting in 13.1–26.3% more high-quality genomes than the second best binner for long-read data.

Availability and implementation: SemiBin2 is available as open source software at https://github.com/BigDataBiology/SemiBin/ and the analysis scripts used in the study can be found at https://github.com/BigDataBiology/SemiBin2_benchmark.

1 Introduction

Microorganisms live in all environments on Earth and play essential roles in human health, agriculture, food, climate change, and other processes (Cavicchioli et al. 2019). Considering the difficulty of culturing microorganisms, metagenome sequencing is widely used to study microorganisms (Quince et al. 2017). Due to the low cost of short-read sequencing, large compendia of metagenome-assembled genomes (MAGs) have been built, expanding the known diversity of bacteria in human-, animal-associated, and environmental habitats (Tully et al. 2018; Almeida et al. 2019; Stewart et al. 2019; Coelho et al. 2022; Zeng et al. 2022). Despite the success of short-read sequencing, it often fails to reconstruct repeated elements (Koren and Phillippy 2015; Tørresen et al. 2019). Recently, long-read sequencing technologies, such as PacBio and Oxford Nanopore which address this limitation, have started to become popular (Bickhart et al. 2022; Feng et al. 2022; Sereika et al. 2022).

Whether using short- or long-read sequencing, assembling large contiguous sequences (contigs) from individual reads is the first step in recovering MAGs. Metagenomic binning is a clustering problem, namely grouping together contigs inferred to originate from the same organism to reconstruct MAGs (Parks et al. 2017). Several binning methods have been proposed. Most existing unsupervised metagenomic binners, such as Canopy (Nielsen et al. 2014), SolidBin (Wang et al. 2019), MetaBAT2 (Kang et al. 2019), MaxBin2 (Wu et al. 2016), and MetaDecoder (Liu et al. 2022) reconstruct bins using *k*-mer frequencies and abundance information. Recently, deep learning has been applied to this problem. VAMB (Nissen et al. 2021) uses deep variational autoencoders to encode *k*-mer and abundance features prior to clustering. SemiBin (henceforth, SemiBin1) implements a semi-supervised approach to learn an embedding by contrastive learning with information from reference genomes, and it achieved state-of-the-art binning results across several habitats, and different binning modes (Pan et al. 2022).

Nevertheless, semi-supervised learning has two drawbacks: (i) SemiBin1 requires using a contig annotation tool such as MMseqs2 (Steinegger and Söding 2017; Mirdita et al. 2021) to generate cannot-link constraints, which significantly increases the running time and peak memory usage of the binning process (Pan et al. 2022); (ii) limitations of the reference genome databases lead to bias (some genomes cannot be annotated and will not be covered in the cannot-link constraints) (Pan et al. 2022) (see Supplementary Table S2). Binning can be performed per-sample or using multiple samples at once, a setting termed multi-sample (Nissen et al. 2021). For single-sample binning, embedding models can be pretrained from large collections of samples, which can alleviate the annotation bias problem and, given the fact that models can be reused, the computational costs are amortized. On the other hand, for multi-sample binning, models need to be relearned for each binning task (as they depend on the number of samples used). So here we focus on improving multi-sample binning, which was shown to lead to the highest number of recovered high-quality MAGs (Nissen et al. 2021; Pan et al. 2022). In particular, we propose a self-supervised binning method that does not require reference genome annotation.

Another limitation of existing binners is that most of them are not optimized for long-read sequencing data. Even though they can be used, assemblies from long-read sequencing are significantly different from short-read sequencing (see Supplementary Table S1) and results will be sub-optimal. Thus, SemiBin2 proposes an ensemble-based clustering algorithm to extend to long-read data. We compared it to the other binners proposed for long-read sequencing data, such as LRBinner (Wickramarachchi et al. 2020; Wickramarachchi and Lin 2021) and the recently proposed GraphMB (Lamurias et al. 2022), and show that it outperforms them.

We have shown that SemiBin2 obtains state-of-the-art binning results on the short- and long-read sequencing data and needs much less computational resources (less running time and peak memory usage) than SemiBin1.

2 Materials and methods

2.1 The SemiBin2 algorithm

We developed SemiBin2, a self-supervised contrastive deep learning-based contig-level binning tool for short- and long-read metagenomic data (see Fig. 1). As in SemiBin1, SemiBin2 uses must-link and cannot-link constraints to learn a feature embedding prior to clustering (Pan et al. 2022). The intuition is that, compared to the original feature space, in the embedded space, contigs from the same genome will be closer together while contigs from different genomes will be further apart. However, SemiBin2 improves on SemiBin1 by using self-supervised learning to obtain cannot-link constraints: randomly sampled pairs of contigs are assumed to contain a cannot-link between them.

The clustering approach used depends on the type of data. For short-read data, the same community detection-based method employed in SemiBin1 is used (Rosvall and Bergstrom 2008). For long-read data, however, a novel ensemble method is used (see below), based on DBSCAN (Ester et al. 1996).

2.2 Preprocessing

Every contig is represented by its tetramer frequencies and its estimated abundance (the average number of reads per base mapping to the contig). The abundance is calculated using BEDTools (version 2.29.1, genomecov command) (Quinlan and Hall 2010) after mapping the short reads to the contigs being binned. Depending on the numbers of samples and sequencing technology, SemiBin2 uses different ways to process the abundance values. Assuming the original abundance value is a and the number of samples is N, SemiBin2 processes the abundance as follows:

$$f(a) = \begin{cases} log(a) & \text{if } N < 5 \text{ and long} - \text{read data} \\ \dfrac{a}{100 \times \left\lceil \frac{a_{mean}}{100} \right\rceil} & \text{if } 5 \leq N \leq 20 \\ \dfrac{a}{\sum_i a_i} & \text{if } N > 20 \end{cases}. \quad (1)$$

a_{mean} is the average of all abundance values and a_i is the abundance value of sample i used in binning. The inputs to the deep learning model are the k-mer frequencies and preprocessed abundance values.

2.2.1 Self-supervised contrastive learning

SemiBin2 uses the same approach as SemiBin1 to generate must-link constraints, namely simulating the break up of longer contigs. For the generation of cannot-link constraints, SemiBin1 used taxonomic annotations which carried large computational costs (including high memory requirements which limited the accessibility of the method). SemiBin2 uses a self-supervised approach, randomly sampling pairs of contigs and treating them as cannot-link pairs.

To control the training time and the ratio between must-link and cannot-link constraints, SemiBin2 limits the number of cannot-link constraints used in training to $\min(m, 4000000)$, where m is the number of must-link constraints.

Then, SemiBin2 uses a deep siamese neural network (Chopra et al. 2005) to implement a contrastive learning method and learns a better embedding from the must-link and cannot-link constraints. The inputs to the neural network are the features computed from the contigs, while its outputs are the fixed dimensional embedding in $\mathbf{R}^{100}$. The first two layers of the network are followed by a batch normalization (Ioffe and Szegedy 2015) layer, a leaky rectified linear unit (Maas et al. 2013) layer and a dropout layer (Srivastava et al. 2014):

$$H(x) = \text{Dropout}(\text{Leaky\_ReLU}(BN(Wx + b))), \quad (2)$$

where $H(x)$ is the output of the neural network, x is the inputs, W, b is the parameter of the neural network.

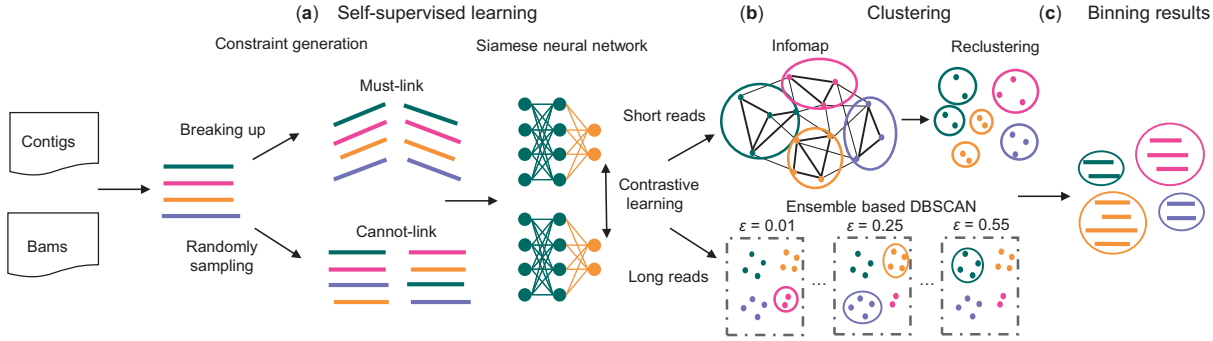

Figure 1. Overview of the SemiBin2 pipeline. (a) Self-supervised learning, including two steps: constraint generation and the siamese neural network. Generating must-link constraints is done by breaking up longer contigs and cannot-link constraints by random sampling. Then, a deep siamese neural network is used to learn a better embedding from the inputs. (b) For short-reads, the Infomap algorithm is used to obtain preliminary bins from the sparse graph generated from the embeddings, followed by weighted k-means to recluster bins whose the mean number of single-copy marker genes is >1. For long-reads, SemiBin2 runs DBSCAN with different values of the ε parameter with embeddings as inputs and integrates the results based on single-copy marker genes. (c) Output the final binning results larger than a user-definable threshold (default 200 kb).

SemiBin2 uses a supervised contrastive loss to classify the must-link (positive label) and cannot-link (negative label) constraints. Intuitively, the distance in the embedded space between elements with a must-link constraint is minimized, while keeping elements with a cannot-link constraint apart. This is unlike SemiBin1, which combines this loss with an unsupervised reconstruction loss. In particular, the contrastive loss function used is:

$$L = \frac{1}{|\mathbf{M} \cup \mathbf{C}|} \sum_{(x_1, x_2) \in \mathbf{M} \cup \mathbf{C}} \Big[y \cdot d\Big(H(x_1), H(x_2)\Big)^2 + (1-y) \cdot \max\{1 - d\Big(H(x_1), H(x_2)\Big), 0\}^2 \Big], \tag{3}$$

where x_1, x_2 are input features, $\mathbf{M}$ denotes all must-link constraints, $\mathbf{C}$ denotes all cannot-link constraints, d denotes the Euclidean distance, and y is the label such that 1 denotes x_1 and x_2 contain a must-link constraint between them while 0 denotes they contain a cannot-link constraint. The loss function is minimized with Adam optimization algorithm (Kingma and Ba 2014).

As was the case with SemiBin1, when binning a single sample (or many samples independently), models can be built once from either the samples to be binned or an external dataset and reused many times. For multi-sample binning, however, models need to be learned for each input. In that case, a model is learned from the same input contigs that will subsequently be clustered.

2.2.2 Clustering of short- and long-read data

After obtaining the contig embeddings from the deep learning model, a clustering algorithm is used to obtain the final bins. For short-read data, SemiBin2 uses the same clustering method used in SemiBin1 (Pan et al. 2022). Briefly, a graph is built based on the contig embeddings and abundance features using contigs as nodes, and similarity between contigs as the weight of each edge. For each node, only the edges with the highest weights are kept (the number is controlled by the parameter *max_edges*). Infomap (Rosvall and Bergstrom 2008) (an information-theory-based community detection algorithm) is used to generate preliminary bins from the sparse graph. If there are bins whose mean number of single-copy marker genes (Wu et al. 2014) is >1, these bins are reclustered with the weighted *k*-means method to get the final results.

However, preliminary testing showed that this approach is not suitable for long-read data as assemblies from long-read data have different properties compared to assemblies from short-read data. In particular, long-read contigs are fewer and much longer (see Supplementary Table S1). This results in some genomes consisting of a small number of contigs (even a single contig). The existing tools are mostly optimized for short-read data and do not work very well when applied to long-read data.

Therefore, for long-read data, SemiBin2 employs an ensemble-based DBSCAN algorithm to bin contigs. DBSCAN (Ester et al. 1996; Schubert et al. 2017) is a clustering method that identifies regions of space that are densely populated. Namely, it uses a user-tunable parameter (ε) to define the maximum distance at which two points are considered to be connected. The resulting graph is then processed to extract subgraphs that fulfill a set of connection criteria that attempts to capture the notion of a dense region of space. A smaller ε value will lead to a sparser connection graph and hence smaller clusters; while a larger ε value will lead to larger clusters. SemiBin2 uses the implementation of DBSCAN in scikit-learn (Pedregosa et al. 2011) and runs DBSCAN with ε value equals to 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, and 0.55. Then SemiBin2 integrates the results of these runs based on the single-copy marker genes that have been used in other tools (Lamurias et al. 2022; Liu et al. 2022). In particular, SemiBin2 uses 107 single-copy marker genes (Wu et al. 2014) to estimate the completeness, contamination, and F1-score of every bin.

Assume we find G instances of N single-copy marker genes (i.e. G counts the total number of sequences matching *any* single-copy gene, while N counts the number of different single-copy genes found). Then,

$$\begin{aligned} \text{completeness} &= \frac{N}{107} \\ \text{contamination} &= \frac{G-N}{G} \\ \text{F1}-\text{score} &= \frac{2 \times \text{completeness} \times (1 - \text{contamination})}{\text{completeness} + (1 - \text{contamination})} \end{aligned} \tag{4}$$

Based on these metrics, SemiBin2 uses a greedy algorithm to select the final bins: at each step, the best bin is selected and all its contigs are removed from further consideration until no more bins can be found that fulfill the minimal quality criteria (see Algorithm 1).

2.3 Data used for benchmarking

For benchmarking, we used 5 simulated datasets from the CAMI II challenges, real data from 4 short-read sequencing projects and from 3 long-read sequencing projects. CAMI II datasets comprise 5 human-associated environments: airways (10 samples), gastrointestinal (10 samples), oral (10 samples), skin (10 samples), and urogenital (9 samples). Every

Algorithm 1. Integrate all runs of DBSCAN

Input: Results of every run of DBSCAN: R_1, R_2, R_3 ... R_n;
A function best_bin(contamination) to return the bin with best F1-score from the bins with contamination smaller than predefined value (contamination).
If not found, return None.
Output: Reconstructed bins
for cont = 0.1, 0.2, 0.3, 0.4, 0.5, 1 **do**
  while not all bins are empty **do**
    bin = best_bin(contamination)
    if bin is not None **then**
      output the returned bin;
      remove the contigs in the bin from all other bins
    else
      break (go to next value of cont)
    end if
  end while
end for

environment has short-read simulated datasets and PacBio (long-read) simulated datasets.

We used the same 4 short-read sequencing projects and the same assemblies previously used to evaluate SemiBin1: human gut (82 samples) (Wirbel et al. 2019), dog gut (129 samples) (Coelho et al. 2018), ocean (109 samples) (Sunagawa et al. 2015), and soil (101 samples) (Olm et al. 2017). For long-read datasets, we used three projects: PRJCA007414 (human gut, 3 PacBio-HiFi samples, 3 Nanopore R9.4 samples), PRJNA595610 (sheep gut, 2 PacBio-HiFi samples) (Bickhart et al. 2022), and PRJEB48021 (activated sludge from an anaerobic digester, 1 PacBio-HiFi sample, 2 NanoPore R9.4.1 samples, and 1 Nanopore R10.4 sample) (Sereika et al. 2022) (see Supplementary Table S3). Long-reads were assembled with Flye (Kolmogorov et al. 2020) (version 2.9-b1768, *–pacbio-hifi* for assembling PacBio-HiFi reads, *–nano-raw* for assembling Nanopore reads from PRJCA007414 and *–nano-hq* for assembling Nanopore reads from PRJEB48021). For correction of the assemblies from Nanopore reads, we used the tools used in the original studies. For the correction of assemblies from PRJCA007414, we used Pilon (Walker et al. 2014) (version 1.24) and for assemblies from Nanopore R9.4.1 from PRJEB48021, we runned Medaka (version 1.7.1, *-m r941_min_sup_g507*) and Racon (Vaser et al. 2017) (version 1.5.0, used two times, one round using long-reads and one round using short-reads). For short-reads, we used Bowtie2 (Langmead and Salzberg 2012) to generate the mapping (BAM) files and for long reads, we used Minimap2 (Li 2018) (version 2.24-r1122, *-x map-hifi* for PacBio-HiFi reads and *-x map-ont* for Nanopore reads).

The simulated CAMI II datasets can be downloaded from https://data.cami-challenge.org/participate. The short-read datasets used in the study can be found in SemiBin1 (Sunagawa et al. 2015; Olm et al. 2017; Coelho et al. 2018; Wirbel et al. 2019; Pan et al. 2022). The long-read datasets used are publicly available in the NGDC with the study accession PRJCA007414 and in the ENA with the study accessions PRJNA595610 and PRJEB48021.

2.4 Methods used in benchmarking

For short-read datasets with multi-sample binning, we compared to VAMB (version 3.0.7, *-m 2000*) (Nissen et al. 2021) and SemiBin1 (version 1.0.0) (Pan et al. 2022), existing binners supporting for multi-sample binning. For long-read datasets, we compared to MetaBAT2 (version 2.15, *–percentIdentity 50*) (Kang et al. 2019), VAMB (version 3.0.7, *-m 2000*) (Nissen et al. 2021), SemiBin1 (version 1.0.0, *–environment human_gut* for PRJCA007414) (Pan et al. 2022), GraphMB (version 0.1.4) (Lamurias et al. 2022), MetaDecoder (version 1.0.13) (Liu et al. 2022), LRBinner (version 2.1) (Wickramarachchi and Lin 2021), CONCOCT (version 1.1.0) (Alneberg et al. 2014), and MetaBinner (version 1.4.4) (Wang et al. 2023).

Parameters for model training (number of must-link constraints, number of cannot-link constraints) and short-read clustering (*max_edges*, used to control the sparsity of the graph for clustering) are set to defaults, but we previously showed that results are robust to these parameters (Pan et al. 2022).

To benchmark the value of the embedding specifically, we performed an ablation study whereby we removed the self-supervised learning step and performed clustering based on the original inputs (a setting we called *NoSemi*).

2.5 Evaluation metrics

We used two metrics to evaluate a bin, completeness (the fraction of the original genome that is captured by the bin) and contamination (the fraction of the bin which does not belong to the original genome).

For simulated datasets, we used gold standard assemblies for binning provided by the CAMI II challenge. We used AMBER (version 2.0.2) to evaluate the results of the simulated datasets (completeness and contamination).

For real datasets, as labels are not known, we evaluated the results using CheckM (version 1.1.9, using lineage_wf workflow) (Parks et al. 2015) and GUNC (version 1.0.5). For long-read datasets, as some tools perform binning based on a set of single-copy marker genes which overlaps with the genes used by CheckM for evaluation, this may overestimate the quality of the outputs. Thus, we used CheckM2 (Chklovski et al. 2022) (version 0.1.3) which is based on machine learning to estimate the completeness and contamination for long-read datasets.

For simulated datasets, we defined high-quality bins as those with completeness >90% and contamination <5%. For short-read datasets, we defined high-quality bins as those with completeness >90%, contamination <5%, and passing the chimeric detection of GUNC. For long-read datasets, we termed the high-quality bins defined before as near-complete bins and defined high-quality bins as those with completeness >90%, contamination <5%, passing the chimeric detection of GUNC and having at least one 23S, 16S, 5S rRNA genes, and 18 distinct tRNAs (Bowers et al. 2017). We used Barrnap (version 0.9, https://github.com/tseemann/barrnap) and tRNAscan-SE (version 2.0.9) (Chan and Lowe 2019) to detect these genes. A high-quality bin can make sure that the bin capture enough biological information with few contamination from other genomes. We used the number of high-quality bins to evaluate the performance of the binners which is widely in other studies (Nissen et al. 2021; Lamurias et al. 2022; Liu et al. 2022; Pan et al. 2022).

3 Results

3.1 Self-supervised learning reduces resource usage and improves results

When comparing the cannot-link constraints generated from taxonomic annotation to random sampling on simulated data (where the ground truth is known), we found that taxonomic annotation leads to more accurate constraints. On the other hand, random sampling could cover more genomes (see Supplementary Table S2). Deep learning can be robust to noise (Rolnick et al. 2017), and more genomes covered can provide more information to the model. Thus, it is an empirical question which approach results in better binning.

Compared to semi-supervised learning (used in SemiBin1), self-supervised learning could achieve similar or better results in most of the simulated datasets (see Fig. 2, Supplementary Fig. S1). As simulated datasets are less complex than real-world datasets (Pan et al. 2022) and better represented in databases, cannot-link constraints from contig annotations can have high accuracy and coverage, and SemiBin1 and SemiBin2 return comparable results. However, in complex real data, self-supervised learning results in a large improvement in the number of returned high-quality bins (see Fig. 5).

In the SemiBin1 workflow, contig annotation, which uses MMseqs2, is the most time-consuming and memory intensive

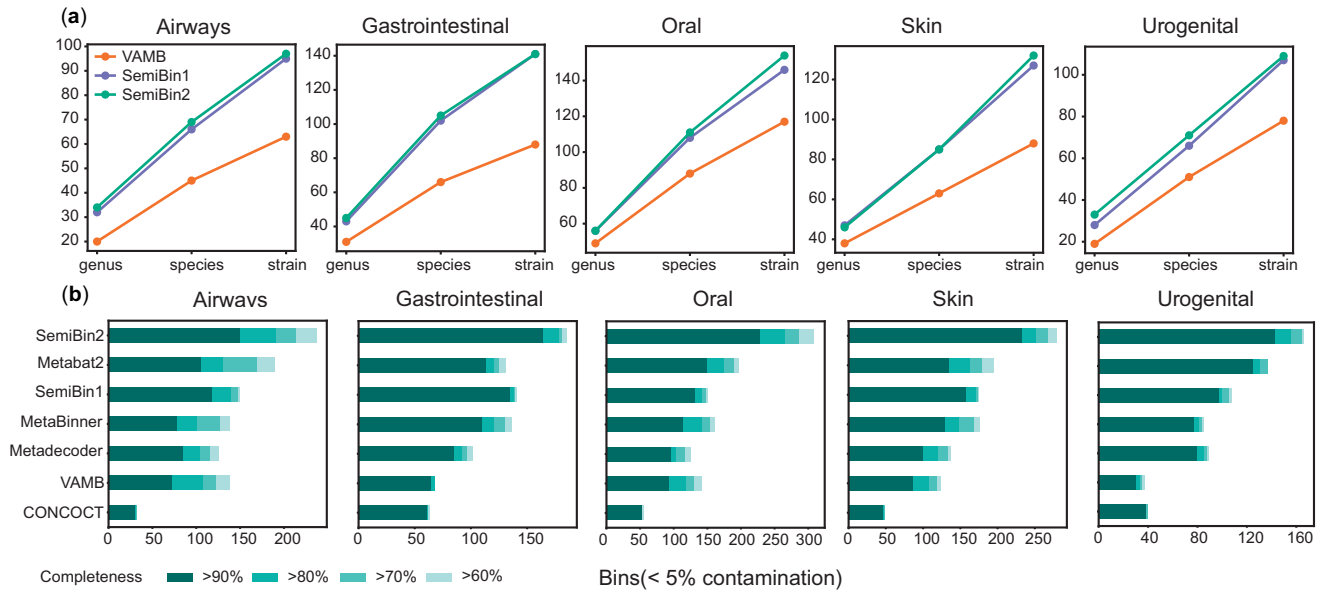

Figure 2. SemiBin2 outperforms other binners with short- and long-read sequencing CAMI II datasets. (a) SemiBin2 outperformed VAMB and got similar results to SemiBin1 in CAMI II short-read sequencing datasets. Shown is the number of distinct genera, species and strains (see Section 2). (b) SemiBin2 reconstructed many more high-quality bins than other binners in CAMI II long-read sequencing datasets. Shown are the numbers of reconstructed genomes with different completeness and contamination <5%.

step. Self-supervised learning avoids this step and makes SemiBin2 more efficient than SemiBin1 (see Supplementary Table S4).

To further demonstrate the value of self-supervised constrastive learning, we also compared SemiBin2 to the same pipeline without the deep learning step (termed NoSemi). In the five environments of CAMI II short-read sequencing datasets, SemiBin2 could reconstruct average 21.6% more distinct high-quality strains, 18.4% more distinct high-quality species, and 17.1% more distinct high-quality genera compared to NoSemi (see Fig. 3). For long-read sequencing datasets, SemiBin2 could reconstruct average 31.5% more high-quality bins, showing the effectiveness of self-supervised learning.

3.2 SemiBin2 outperformed other binners in CAMI II simulated datasets

We compared SemiBin2 to widely used and recently proposed binners on the five environments with short- and long-read datasets from CAMI II. For short-read sequencing datasets, because of the low cost of this technology, there are enough samples to allow us to run binning in multi-sample binning mode, which has been shown to reconstruct the most high-quality bins (Nissen et al. 2021; Pan et al. 2022). We compared SemiBin2 with SemiBin1 (Pan et al. 2022) and VAMB (Nissen et al. 2021), which are the only two existing binners supporting multi-sample binning. For the long-read datasets, we compared to MetaBAT2 (Kang et al. 2019), MetaDecoder (Liu et al. 2022), VAMB (Nissen et al. 2021), SemiBin1 (Pan et al. 2022), CONCOCT (Alneberg et al. 2014), and MetaBinner (Wang et al. 2023). We did not include GraphMB (Lamurias et al. 2022) and LRBinner (Wickramarachchi and Lin 2021) in this comparison because we used gold standard assemblies for binning in simulated datasets, and we could not obtain the assembly graph, which GraphMB requires as input and LRBinner cannot be run with co-assembly binning (we compared SemiBin2 to GraphMB and LRBinner in another simulated dataset, see Supplementary Fig. S5).

In the short-read datasets, SemiBin2 reconstructed on average 44.8% more distinct high-quality genera (range 14.3–73.7%), 42.5% more distinct high-quality species (range 26.1–59.1%), and 47.1% more distinct high-quality strains (range 31.6–60.2%) compared to VAMB (see Fig. 2). When compared to SemiBin1, SemiBin2 performed similarly or better, showing the effectiveness of self-supervised contrastive learning and avoiding the time and memory usage required for contig annotations.

For long-read datasets, we proposed an ensemble-based DBSCAN clustering algorithm (see Section 2) instead of the community detection approach used for short-read datasets. The ensemble-based DBSCAN clustering algorithm runs DBSCAN with different ε values and integrates them using single-copy marker genes (see Section 2). To show that the ensemble step could improve binning results, we compared SemiBin2 (ensemble-based DBSCAN algorithm) to binning with running DBSCAN with a single ε value (see Fig. 4). In the airways, gastrointestinal, oral, skin and urogenital environments, SemiBin2 could reconstruct 78.6%, 37.8%, 51.7%, 66.4%, and 25.4% more high-quality bins compared to the best result of single DBSCAN running, indicating the ensemble learning could effectively integrate different runs and improve binning results. Using HDBSCAN (Campello et al. 2013), another density-based clustering method, did not improve results compared to DBSCAN (see Supplementary Fig. S4).

Then, we compared SemiBin2 to other binners (see Fig. 2). SemiBin2 outperformed other tools in all environments and reconstructed 27.1%, 21.5%, 52.7%, 47.5%, and 14.4% more high-quality bins than the second-best binner in these five environments.

3.3 SemiBin2 outperformed other binners in short- and long-read real datasets

We compared SemiBin2 to VAMB (Nissen et al. 2021) and SemiBin1 (Pan et al. 2022) with multi-sample binning in short-read sequencing datasets and to MetaBAT2 (Kang et al.

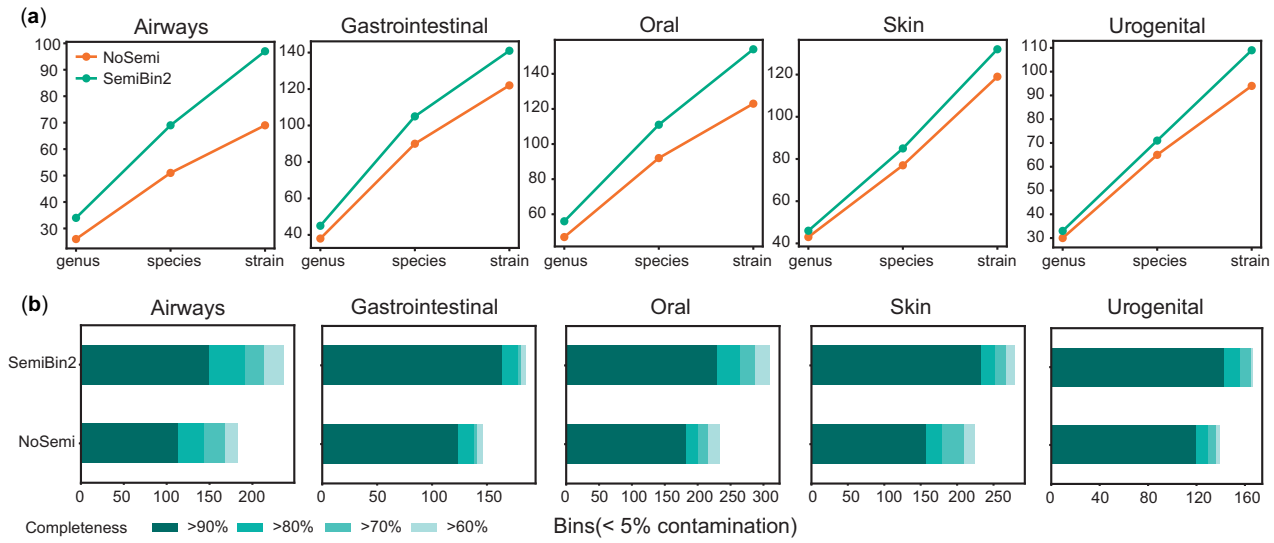

Figure 3. Self-supervised learning improveds binning results in CAMI II datasets. To evaluate the impact of self-supervised learning, we compare the same pipeline with (SemiBin2) or without (NoSemi) the deep learning step on the simulated CAMI II datasets. (a) SemiBin2 reconstructed more distinct high-quality genera, species and strains in the five environments from CAMI II compared to the NoSemi version. Shown are the number of distinct genera, species and strains (see Section 2). (b) SemiBin2 reconstructed more high-quality bins than the NoSemi version. Shown are the numbers of reconstructed genomes with different completeness and contamination <5%.

2019), MetaDecoder (Liu et al. 2022), VAMB (Nissen et al. 2021), SemiBin1 (Pan et al. 2022), GraphMB (Lamurias et al. 2022), and LRBinner (Wickramarachchi and Lin 2021) in long-read sequencing datasets. For short-read sequencing, we used datasets from four environments: human gut (Wirbel et al. 2019), dog gut (Coelho et al. 2018), ocean (Sunagawa et al. 2015), and soil (Olm et al. 2017). For long-read sequencing, we chose three long-read sequencing studies: human gut (3 PacBio-HiFi samples, 3 Nanopore R9.4 samples), sheep gut (2 PacBio-HiFi samples) (Bickhart et al. 2022), and activated sludge (1 PacBio-HiFi sample, 2 Nanopore R9.4.1 samples and 1 Nanopore R10.4 sample) (Sereika et al. 2022). These studies cover different technologies for long-read sequencing that can be used to evaluate SemiBin2 in different situations.

In real data, the true labels of the contigs (e.g. which genomes each contig belongs to) are unknown. Thus, we used automated tools [CheckM (Parks et al. 2015; Chklovski et al. 2022) and GUNC (Orakov et al. 2021)] to evaluate the outputs (see Section 2).

SemiBin2 could reconstruct the largest number of high-quality bins for short-read datasets with multi-sample binning. In the four environments, SemiBin2 reconstructed 1678, 3776, 631, and 254 high-quality bins (see Fig. 5). Compared to VAMB, SemiBin2 generated 27.3% (360), 21.5% (669), 44.7% (195), and 229.9% (177) more high-quality bins. Compared to SemiBin1, SemiBin2 reconstructed 8.3% (129), 9.5% (328), 10.7% (61), and 21.5% (45) more high-quality bins. On a sample-by-sample basis, SemiBin2 significantly outperformed VAMB [$P = 9.7 \cdot 10^{-14}$ ($n = 82$), $P = 4.7 \cdot 10^{-22}$ ($n = 129$), $P = 2.1 \cdot 10^{-10}$ ($n = 109$), and $P = 1.3 \cdot 10^{-12}$ ($n = 101$)] and SemiBin1 [$P = 1.5 \cdot 10^{-05}$ ($n = 82$), $P = 4.1 \cdot 10^{-18}$ ($n = 129$), $P = 8.7 \cdot 10^{-05}$ ($n = 109$), and $P = 5.8 \cdot 10^{-04}$ ($n = 101$), *P*-values were computed using Wilcoxon signed rank test, two-sided null hypothesis] (see Fig. 5).

When applying them to short-read datasets, the only difference between SemiBin1 and SemiBin2 is the use of semi-supervised vs. self-supervised learning. The results showed

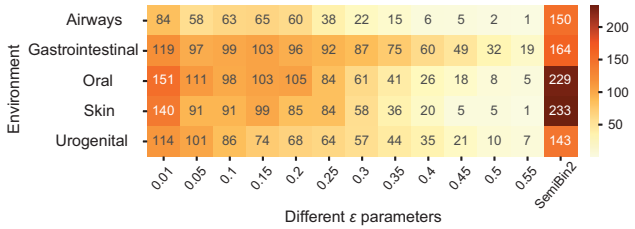

Figure 4. Ensemble-based DBSCAN algorithm improves binning results for long-read data. We compared SemiBin2 (ensemble-based DBSCAN algorithm) to the results of a single DBSCAN run with different ε values in CAMI II long-read datasets. Shown is the number of high-quality bins and *SemiBin2* denotes the result of the ensemble-based DBSCAN algorithm.

that self-supervised learning could learn a better embedding in complex real environments possibly because randomly generated cannot-link constraints cover more genomes (see Supplementary Table S2) and provide more information to the model.

One limitation of SemiBin1 is that it needs to run taxonomic annotation (MMseqs2 by default) to generate the cannot-link constraints. This step requires a lot of computational resources (running time and memory usage). SemiBin2 with self-supervised learning both improves the binning results and addresses this limitation. Taking advantage of self-supervised learning, SemiBin2 needs only *circa* 25% running time on a GPU (the gain when using a CPU is smaller, but still close to 50%; see Supplementary Table S4). More importantly, using either a CPU or a GPU, SemiBin2 requires only 11% of the peak memory usage of the SemiBin1 (see Supplementary Table S4). Therefore, applying SemiBin2 with multi-sample binning to large-scale metagenomic analysis will be much more efficient.

For long-read datasets, SemiBin2 also reconstructed the most near-complete bins (see Section 2). In the human gut, sheep gut, and activated sludge projects with samples from different long-read sequencing technologies, SemiBin2 generated 13.2%, 28.1%, and 14.8% more near-complete bins compared to the second-best binner (see Fig. 6,

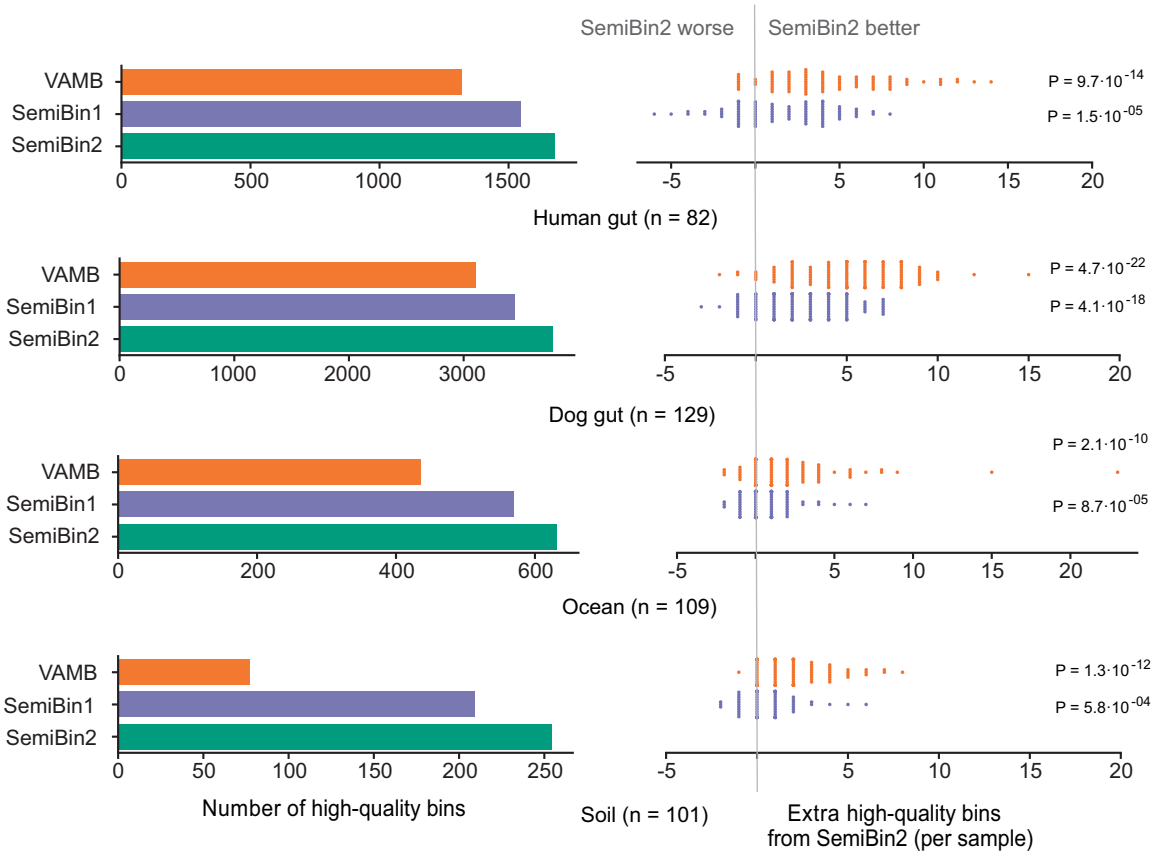

Figure 5. SemiBin2 outperformed other binners in short-read sequencing real datasets with multi-sample binning. SemiBin2 produced more high-quality bins than VAMB and SemiBin1 in the four real datasets with multi-sample binning (left) and reconstructed more high-quality bins in most of the samples (right). *P*-values shown are computed using a two-sided Wilcoxon signed-rank test based on the counts of each sample.

Supplementary Fig. S2). We also benchmarked these binners by evaluating the high-quality bins with 23S, 16S, 5S rRNA genes, and tRNAs (see Section 2). When considering high-quality bins, SemiBin2 still performed the best and could reconstruct 15.6%, 26.3%, and 13.1% more high-quality bins. Overall, SemiBin2 could outperform other binners in all situations for long-read datasets.

4 Discussion and conclusions

During the development of SemiBin1 (Pan et al. 2022), we found that contig annotations with GTDB taxonomy using MMSeqs2 performed better than annotations with the NCBI taxonomy using CAT (von Meijenfeldt et al. 2019). Initially, our expectation was that continued improvement to reference genome databases and taxonomic prediction tools would lead to better binning results. However, there will always be novel species or strains in the environment (Coelho et al. 2022), and taxonomic prediction remains challenging and requires long running times. Thus, we attempted using random sampling to generate cannot-link constraints. Although using in these randomly-sampled links results in more noise compared to those from contig annotation (see Supplementary Table S2), deep learning can be robust to noise (Rolnick et al. 2017), and the observed error rates were low. Empirically, cannot-link constraints generated by random sampling cover more genomes in the environment (including novel strains that the annotation algorithms cannot accurately identify), which leads to SemiBin2 getting better results than SemiBin1.

A similar idea to generate negative inputs has been used in CoCoNet (Arisdakessian et al. 2021) for viral metagenome binning, which also showed that the proportion of mislabeled contigs from the same genome is negligible. CoCoNet splits the contigs into fragments of size 1024 bp to generate the cannot-link constraints, which is too short for metagenomic bacterial binning (contigs with this length are removed by most binning tools). SemiBin2 uses the whole contig for cannot-link constraints to learn a better embedding for clustering.

To a first approximation, the chance that a randomly sampled pair of contigs is actually derived from the same genome decreases with the inverse of the number of genomes present in a sample. While we expect that most samples will contain enough genomes that this rate will be low, it is possible that for very low complexity samples, particularly those from mock or otherwise constructed communities, this error will be too large. In these situations, one can fall back to using a pretrained model as proposed in SemiBin1 (Pan et al. 2022). Models pretrained from short-read samples can be used for long-read datasets (see Supplementary Fig. S3).

In recent long-read sequencing studies, MetaBAT2 (Kang et al. 2019), MaxBin2 (Wu et al. 2016), and VAMB (Nissen et al. 2021) have been used (Sereika et al. 2022). However, these tools are optimized for short-read datasets. For this question, SemiBin2 uses an ensemble-based DBSCAN clustering algorithm for long-read datasets. SemiBin1 performed poorly in long-read datasets, as its community detection algorithm was not designed for long-read sequencing data.

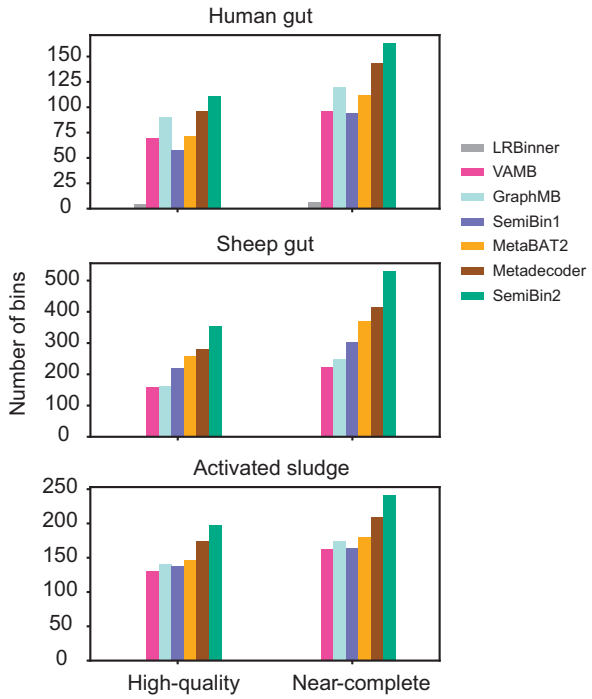

Figure 6. SemiBin2 outperformed other binners in long-read sequencing real datasets. SemiBin2 reconstructed more high-quality and near-complete genomes than other binners in three long-read sequencing projects. Near-complete bins: completeness >90% , contamination <5% and passing the chimeric detection of GUNC; high-quality bins: completeness >90%, contamination <5%, passing the chimeric detection of GUNC and having at least one 23S, 16S, 5S rRNA genes, and 18 distinct tRNAs.

This study proposes SemiBin2, a metagenomic binning method based on self-supervised learning. SemiBin2 outperforms other binners in both short- and long-read datasets (simulated and real). Compared to SemiBin1, by taking advantage of self-supervised learning, SemiBin2 requires much less running time and peak memory usage. Looking forward, there are other sources of information, such as the assembly graph (as used by GraphMB) and information from metaHiC related to physically linked contigs that can be considered to further improve binning results.

In addition to the algorithmic improvements discussed above, since the release of SemiBin1, we have also improved the tool in other ways. For example, SemiBin2 supports CRAM (Hsi-Yang Fritz et al. 2011) files, has more options for ORF finding, produces more statistics on its outputs, enables the user to better control output formats (e.g. filenames, and compression options) and—at the request of users—added support for re-using contig abundance estimates from MetaBAT2. In order to make it easier to run the tool as part of a pipeline, modules for nf-core (Ewels et al. 2020), Galaxy (Galaxy Community 2022) (available at the European Galaxy server, UseGalaxy.eu), and NGLess (Coelho et al. 2019) are now available. We have also fixed issues reported by users and improved error handling and reporting. Overall, SemiBin2 is more robust, easier to use and more flexible than SemiBin1, while returning better results at lower computational cost.

Acknowledgements

We thank Senying Lai (Fudan University) for her helpful comments on a previous version of this manuscript. We thank Bérénice Batut (University of Freiburg) for designing and implementing the Galaxy wrapper. Members of the Zhao and Coelho groups are thanked for their comments throughout the development of this work. Users of SemiBin are thanked for their suggestions and bug reports.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This work was supported by the National Natural Science Foundation of China [31950410544 to L.P.C. and T2225015, 61932008 to X.-M.Z.]; the Shanghai Municipal Science and Technology Major Project [2018SHZDZX01 to X.-M.Z. and L.P.C.]; and the National Key R&D Program of China [2020YFA0712403 to X.-M.Z.].

References

Almeida A, Mitchell AL, Boland M *et al.* A new genomic blueprint of the human gut microbiota. *Nature* 2019;**568**:499–504.

Alneberg J, Bjarnason BS, de Bruijn I *et al.* Binning metagenomic contigs by coverage and composition. *Nat Methods* 2014;**11**:1144–6.

Arisdakessian CG, Nigro OD, Steward GF *et al.* CoCoNet: an efficient deep learning tool for viral metagenome binning. *Bioinformatics* 2021;**37**:2803–10.

Bickhart DM, Kolmogorov M, Tseng E *et al.* Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. *Nat Biotechnol* 2022;**40**:711–9.

Bowers RM, Kyrpides NC, Stepanauskas R *et al.*; Genome Standards Consortium. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nat Biotechnol* 2017;**35**:725–31.

Campello RJ, Moulavi D, Sander J. Density-based clustering based on hierarchical density estimates. In: *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013Gold Coast, Australia, April 14–17, 2013, Proceedings, Part II 17*. Berlin Heidelberg: Springer, 2013, 160–172.

Cavicchioli R, Ripple WJ, Timmis KN *et al.* Scientists' warning to humanity: microorganisms and climate change. *Nat Rev Microbiol* 2019;**17**:569–86.

Chan PP, Lowe TM. tRNAscan-SE: searching for tRNA genes in genomic sequences. In: Kollmar, M. (eds) *Gene Prediction. Methods in Molecular Biology*, vol 1962. New York, NY: Springer. https://doi.org/10.1007/978-1-4939-9173-0_1.

Chklovski A, Parks DH, Woodcroft BJ *et al.* CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. *bioRxiv*, 2022, preprint: https://doi.org/10.1101/2022.07.11.499243.

Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*. Vol. 1. 2005, 539–46.

Coelho LP, Kultima JR, Costea PI *et al.* Similarity of the dog and human gut microbiomes in gene content and response to diet. *Microbiome* 2018;**6**:1–11.

Coelho LP, Alves R, Monteiro P *et al.* NG-meta-profiler: fast processing of metagenomes using NGLess, a domain-specific language. *Microbiome* 2019;**7**:84.

Coelho LP, Alves R, Del Río ÁR *et al.* Towards the biogeography of prokaryotic genes. *Nature* 2022;**601**:252–6.

Ester M, Kriegel HP, Sander J *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Portland, Oregon, USA*. Vol. **96**. 1996, 226–231.

Ewels PA, Peltzer A, Fillinger S *et al.* The NF-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol* 2020; **38**:276–8.

Feng X, Cheng H, Portik D *et al.* Metagenome assembly of high-fidelity long reads with hifiasm-meta. *Nat Methods* 2022;**19**:671–4.

Galaxy Community. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Res* 2022;**50**:W345–51.

Hsi-Yang Fritz M, Leinonen R, Cochrane G *et al.* Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Res* 2011;**21**:734–40.

Ioffe S, Szegedy C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning, Lille, France*, 2015, 448–456.

Kang DD, Li F, Kirton E *et al.* MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* 2019;**7**:e7359.

Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations, (ICLR) 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Kolmogorov M, Bickhart DM, Behsaz B *et al.* metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nat Methods* 2020; **17**:1103–10.

Koren S, Phillippy AM. One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Curr Opin Microbiol* 2015;**23**:110–20.

Lamurias A, Sereika M, Albertsen M *et al.* Metagenomic binning with assembly graph embeddings. *Bioinformatics* 2022;**38**:4481–7.

Langmead B, Salzberg SL. Fast gapped-read alignment with bowtie 2. *Nat Methods* 2012;**9**:357–9.

Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018;**34**:3094–100.

Liu C-C, Dong S-S, Chen J-B *et al.* MetaDecoder: a novel method for clustering metagenomic contigs. *Microbiome* 2022;**10**:1–16.

Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: *International Conference on Machine Learning, Atlanta, GA, USA*. Vol. **30**. 2013, 3.

Mirdita M, Steinegger M, Breitwieser F *et al.* Fast and sensitive taxonomic assignment to metagenomic contigs. *Bioinformatics* 2021;**37**: 3029–31.

Nielsen HB, Almeida M, Juncker AS *et al.*; MetaHIT Consortium. Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nat Biotechnol* 2014;**32**:822–8.

Nissen JN, Johansen J, Allesøe RL *et al.* Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotechnol* 2021;**39**:555–60.

Olm MR, Butterfield CN, Copeland A *et al.* The source and evolutionary history of a microbial contaminant identified through soil metagenomic analysis. *MBio* 2017;**8**:e01969–16.

Orakov A, Fullam A, Coelho LP *et al.* GUNC: detection of chimerism and contamination in prokaryotic genomes. *Genome Biol* 2021;**22**: 1–19.

Pan S, Zhu C, Zhao X-M *et al.* A deep siamese neural network improves metagenome-assembled genomes in microbiome datasets across different environments. *Nat Commun* 2022;**13**:2326.

Parks DH, Imelfort M, Skennerton CT *et al.* CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* 2015;**25**:1043–55.

Parks DH, Rinke C, Chuvochina M *et al.* Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nat Microbiol* 2017;**2**:1533–42.

Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.

Quince C, Walker AW, Simpson JT *et al.* Shotgun metagenomics, from sampling to analysis. *Nat Biotechnol* 2017;**35**:833–44.

Quinlan AR, Hall IM. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 2010;**26**:841–2.

Rolnick D, Veit A, Belongie S *et al.* Deep learning is robust to massive label noise. *arXiv, arXiv:1705.10694*, 2017, preprint: https://arxiv.org/abs/1705.10694.

Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci USA* 2008;**105**: 1118–23.

Schubert E, Sander J, Ester M *et al.* DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans Database Syst* 2017;**42**:1–21.

Sereika M, Kirkegaard RH, Karst SM *et al.* Oxford nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nat Methods* 2022;**19**:823–6.

Srivastava N *et al.* Dropout: a simple way to prevent neural networks from overfitting. *The J Mach Learn Res* 2014;**15**:1929–58.

Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* 2017; **35**:1026–8.

Stewart RD, Auffret MD, Warr A *et al.* Compendium of 4,941 rumen metagenome-assembled genomes for rumen microbiome biology and enzyme discovery. *Nat Biotechnol* 2019;**37**:953–61.

Sunagawa S, Coelho LP, Chaffron S *et al.*; Tara Oceans coordinators. Structure and function of the global ocean microbiome. *Science* 2015;**348**:1261359.

Tørresen OK, Star B, Mier P *et al.* Tandem repeats lead to sequence assembly errors and impose multi-level challenges for genome and protein databases. *Nucleic Acids Res* 2019;**47**:10994–1006.

Tully BJ, Graham ED, Heidelberg JF *et al.* The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. *Sci Data* 2018;**5**:1–8.

Vaser R, Sović I, Nagarajan N *et al.* Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res* 2017;**27**: 737–46.

von Meijenfeldt FAB, Arkhipova K, Cambuy DD *et al.* Robust taxonomic classification of uncharted microbial sequences and bins with CAT and BAT. *Genome Biol* 2019;**20**:1–14.

Walker BJ, Abeel T, Shea T *et al.* Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS One* 2014;**9**:e112963.

Wang Z, Wang Z, Lu YY *et al.* SolidBin: improving metagenome binning with semi-supervised normalized cut. *Bioinformatics* 2019;**35**: 4229–38.

Wang Z, Huang P, You R *et al.* Metabinner: a high-performance and stand-alone ensemble binning method to recover individual genomes from complex microbial communities. *Genome Biol* 2023;**24**:1.

Wickramarachchi A, Lin Y. Binning long reads in metagenomics datasets using composition and coverage information. *Algorithms Mol Biol* 2022;**17**:14.

Wickramarachchi A, Mallawaarachchi V, Rajan V *et al.* MetaBCC-LR: metagenomics binning by coverage and composition for long reads. *Bioinformatics* 2020;**36**:i3–11.

Wirbel J, Pyl PT, Kartal E *et al.* Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer. *Nat Med* 2019;**25**:679–89.

Wu Y-W, Tang Y-H, Tringe SG *et al.* MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome* 2014;**2**:1–18.

Wu Y-W, Simmons BA, Singer SW *et al.* MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics* 2016;**32**:605–7.

Zeng S, Patangia D, Almeida A *et al.* A compendium of 32,277 metagenome-assembled genomes and over 80 million genes from the early-life human gut microbiome. *Nat Commun* 2022;**13**:5139.