A graph-based Gaussian Mixture Variational Autoencoder improves metagenome binning for short contigs

Lu Zhang (ericluzhang@hkbu.edu.hk)
Hong Kong Baptist University https://orcid.org/0000-0002-2794-7371

Zhenmiao Zhang
Hong Kong Baptist University https://orcid.org/0000-0003-3748-1664

Mingxing Rao
Hong Kong Baptist University

Yufen Huang
BGI Genomics

Article

Keywords: Contig binning, Metagenome-assembled genome, Gaussian mixture variational autoencoder, Machine learning, Graph embedding

Posted Date: September 13th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3308172/v1

Additional Declarations: There is **NO** Competing Interest.

A graph-based Gaussian Mixture Variational Autoencoder improves metagenome binning for short contigs

Zhenmiao Zhang[1†], Mingxing Rao[1†], Yufen Huang[2], Lu Zhang[1,3*]

[1]Department of Computer Science, Faculty of Science, Hong Kong Baptist University, Hong Kong SAR, China

[2]BGI Research, Shenzhen 518083, China

[3]Institute for Research and Continuing Education, Hong Kong Baptist University, Shenzhen, China

†These authors contributed equally to this work

*To whom correspondence should be addressed: E-mail: ericluzhang@hkbu.edu.hk

Abstract

Grouping contigs from the same genome together through metagenome binning is crucial in reconstructing metagenome-assembled genomes (MAGs). While existing metagenome binning methods have successfully produced near-complete MAGs for long contigs with lengths exceeding 2Kb, they typically neglect short contigs in binning due to their inability to process the unstable conventional sequence composition features of short contigs.

We developed DeepMetaBin, a contig binning method that utilizes a graph-based Gaussian Mixture Variational Autoencoder (GMVAE) to group short (1Kb-2Kb) and long contigs (>2Kb) together. DeepMetaBin starts by generating a k-nearest neighbor graph (k-NN) for contigs based on their sequence composition features and removes noise nodes based on preliminary clustering and label propagation. Next, edges in the refined k-NN are incorporated as "must-links" to stabilize the embedding of short contigs in GMVAE. Finally, Gaussian Mixture Model is used to group the contigs into super clusters in the low-dimensional space, followed by involving single-copy genes as "cannot-links" to facilitate the division of these super clusters using constraint k-means.

We extensively evaluated DeepMetaBin by comparing its performance with six existing metagenome binning tools using simulated and real microbial communities. DeepMetaBin achieved the highest F1 scores for most of the single-sample and multi-sample metagenome binning tasks. It addressed the current practice in the research community of omitting short contigs during modeling and was the only tool that achieved state-of-art performance on both F1 scores and the number of near-complete MAGs.

Keywords: Contig binning; Metagenome-assembled genome; Gaussian mixture variational autoencoder; Machine learning; Graph embedding

Introduction

Due to the practical difficulties in isolating and culturing some microbes in the laboratory[1-3], metagenomic sequencing, which takes whole environmental samples as sequencing input without the need for microbial isolation, has become a popular method that provides rich information for both culturable and unculturable microbes[4]. The typical workflow to reconstruct metagenome-assembled genomes (MAGs) from sequencing reads takes two computational steps: the metagenome assembly step that generates non-overlapping, continuous sequences (contigs) from the reads[5], and the metagenome binning step that groups the contigs predicted to belong to the same taxonomy together, forming the MAGs[5].

Read depths and tetranucleotide frequencies (TNFs) are two types of sequence composition features used for metagenome binning. The former is obtained by aligning reads to the contigs and calculating the average depth of the contigs; the latter is a vector of the frequencies of all 4-mers (subsequences of length 4) for each contig. Many metagenome binning tools utilize these features, such as MaxBin[6], MetaBAT2[7], SolidBin-naive[8], and VAMB[9]. MaxBin applies expectation maximization on probabilistic modeling of read depths and TNFs. MetaBAT2 builds a similarity graph using scores obtained from read depths and TNFs and performs graph partitioning using label propagation. SolidBin-naive is a reference-free mode of SolidBin that utilizes a normalized-cut-based spectral clustering on the graph built from read depths and TNFs. VAMB compresses the dimension of read depths and TNFs to obtain low-dimensional latent embedding and uses a density-based latent clustering algorithm to group the contigs.

Some binning tools use single-copy genes in addition to read depths and TNFs to improve metagenome binning such as MaxBin, SolidBin-naive, MetaCoAG[10], and MetaDecoder[11]. Single-copy genes by definition have only one copy per microbial genome, so contigs with the same single-copy gene must belong to different genomes and should not be clustered together. Another type of binning tool utilizes the taxonomic annotation or alignment information from reference genome databases for performing semi-supervised contig binning. Examples of such tools include COCACOLA[12], SolidBin-coalign mode[8], and SemiBin[13]. However, the performance of these binning tools depends on the quality of reference genomes.

A substantial fraction of the total assembly length is contained within contigs with lengths lower than 2Kb (e.g. 25.6% of the total length for the metaSPAdes[14] assembly from Sharon[15] dataset; **Methods**). However, current binning methods using read depths and TNFs are challenged by short contigs (1Kb-2Kb), which renders the read depths and TNFs of short contigs unreliable[16].

Existing contig binning tools can be divided into two categories according to their performance on short contigs and near-complete MAGs (NCMAGs; completeness > 90 and contamination <5; **Methods**). The first category of binning tools (such as MaxBin and SolidBin-naïve) can cluster short contigs with a minimum length of 1Kb, but they are unable to generate a competitive number of NCMAGs compared to the other state-of-the-art tools. The second category of binning tools (such as VAMB and MetaDecoder) focuses on improving the number of NCMAGs but the performance relies on statistically sufficiently powered sequence composition features, so they discard the short contigs by default to increase the stability of their computational models. VAMB and MetaDecoder only group contigs exceeding 2Kb and 2.5Kb by default, respectively. The F1 score (**Methods**) reflects a balanced evaluation on both short and long contigs and was widely used in previous binning evaluations[8, 16]. This category of contig binning tools discarding short contigs would result in a low F1 score[16]. To cope with this problem, there are binning refinement tools (such as GraphBin[16]) developed to cluster short contigs based on the binning results of the initial binning tools and the assembly graph, but their performance on NCMAGs has not yet been evaluated.

In this work, we describe DeepMetaBin, which implements a graph-based Gaussian mixture variational autoencoder (GMVAE) to accurately group both short (1Kb-2Kb) and long contigs (>2Kb) into MAGs. DeepMetaBin starts by generating a k-nearest neighbor graph (k-NN) for contigs based on their sequence composition features and removes noise nodes based on preliminary clustering and label propagation. This k-NN graph provides accurate connections between short and long contigs if they are from the same microbes. Next, edges in the refined k-NN are incorporated as “must-links” to stabilize the embedding of short contigs in GMVAE. During the training, we restrain the contigs with connections in the k-NN to be close in the latent embedding of GMVAE (**Methods**). In this way, short contigs are binned with both the unstable features of themselves and the stable features of long contigs propagated from the connections in the k-NN graph. This can increase the stability of the binning tool for dealing with short contigs. Finally, Gaussian Mixture Model (GMM) is used to group the contigs into super clusters in the low-dimensional space followed by imposing single-copy genes as “cannot-links” to facilitate the division of these super clusters using constraint k-means. We extensively benchmarked DeepMetaBin against the existing unsupervised contig binning tools MaxBin, MetaBAT2, SolidBin-naive (referred to as SolidBin), MetaCOAG, MetaDecoder, VAMB, and GraphBin. The binning tools were evaluated on CAMI I[17] and CAMI II[18] simulation datasets, and the Sharon[15] real-world microbiome datasets. DeepMetaBin obtained the highest F1 score on a majority of datasets and created a comparable number of NCMAGs with the state-of-the-art metagenome binning tools.

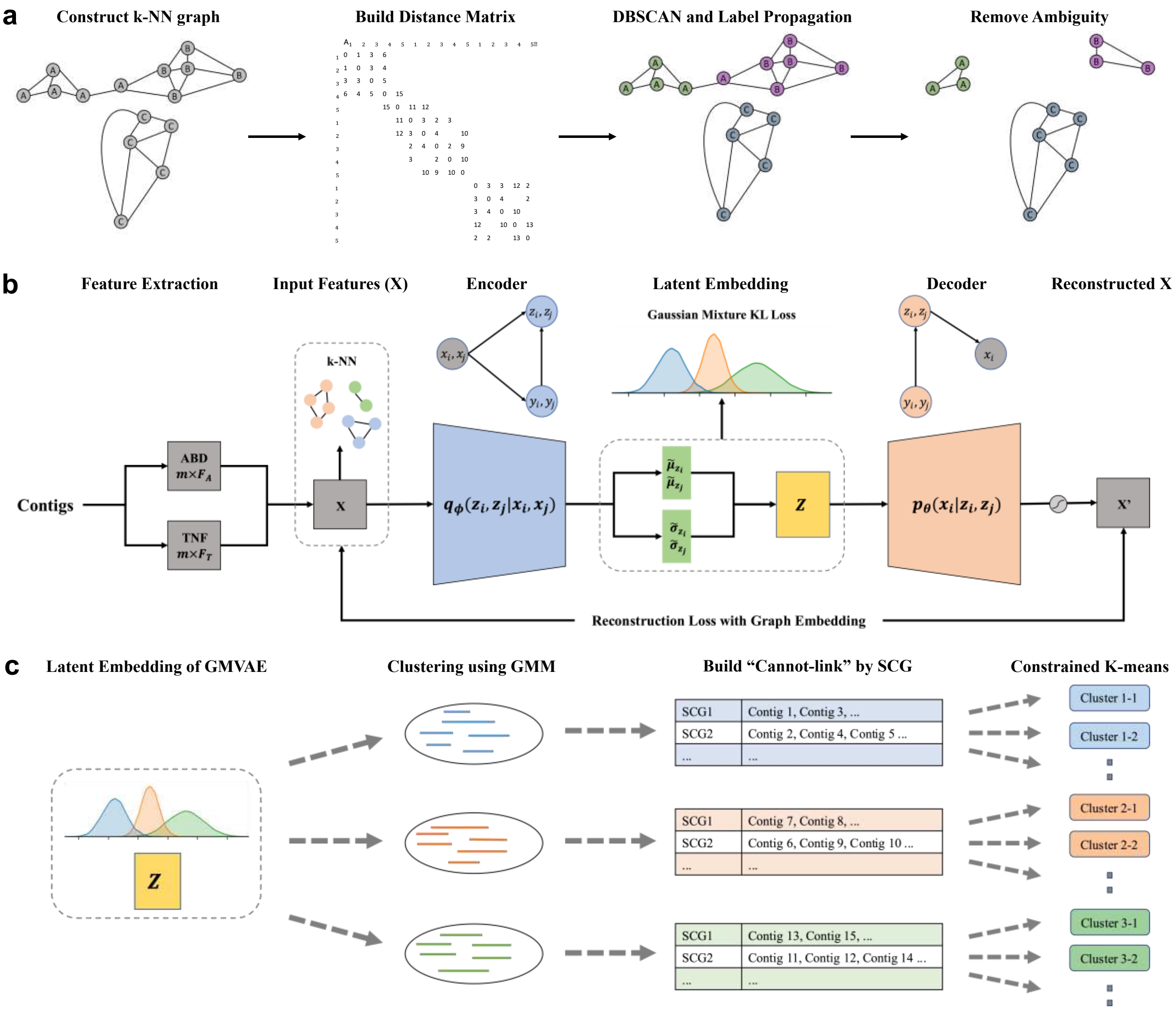

Figure 1. Overview of DeepMetaBin (a) Workflow to generate a refined k-NN graph. (b) Architecture of the Gaussian Mixture Variational Autoencoder (GMVAE) of DeepMetaBin. (c) Latent clustering on the embedding of GMVAE using Gaussian Mixture Model (GMM), and further clustering with "cannot-links" created from single-copy genes (SCG) using constrained k-means.

Results

Overview of DeepMetaBin

The workflow of DeepMetaBin contains three steps (**Figure 1**). In the first step, DeepMetaBin concatenates read depth and TNF features of contigs and constructs a k-NN graph for contigs based on the feature distances (**Methods**). To refine the k-NN graph, we extract the distance matrix from the weights of edges in the k-NN and use DBSCAN to generate seed clusters (**Figure 1 a; Methods**). The seed cluster labels are propagated on the k-NN graph, and nodes with ambiguous labels are removed (**Figure 1 a; Methods**). The second step of DeepMetaBin applies a graph-based GMVAE that models the latent embedding of contigs as a Gaussian Mixture distribution (**Figure 1 b; Methods**). The refined k-NN graph is involved as a "must-link" constraint in the loss function to minimize the embedding distance of linked nodes in the k-NN graph (**Figure 1 b; Methods**). For each super cluster generated by the GMM on contig embedding, DeepMetaBin creates "cannot-links" between contigs if they have shared single-copy genes, and uses a constraint k-means algorithm to divide these super cluster into the final MAGs (**Figure 1 c; Methods**).

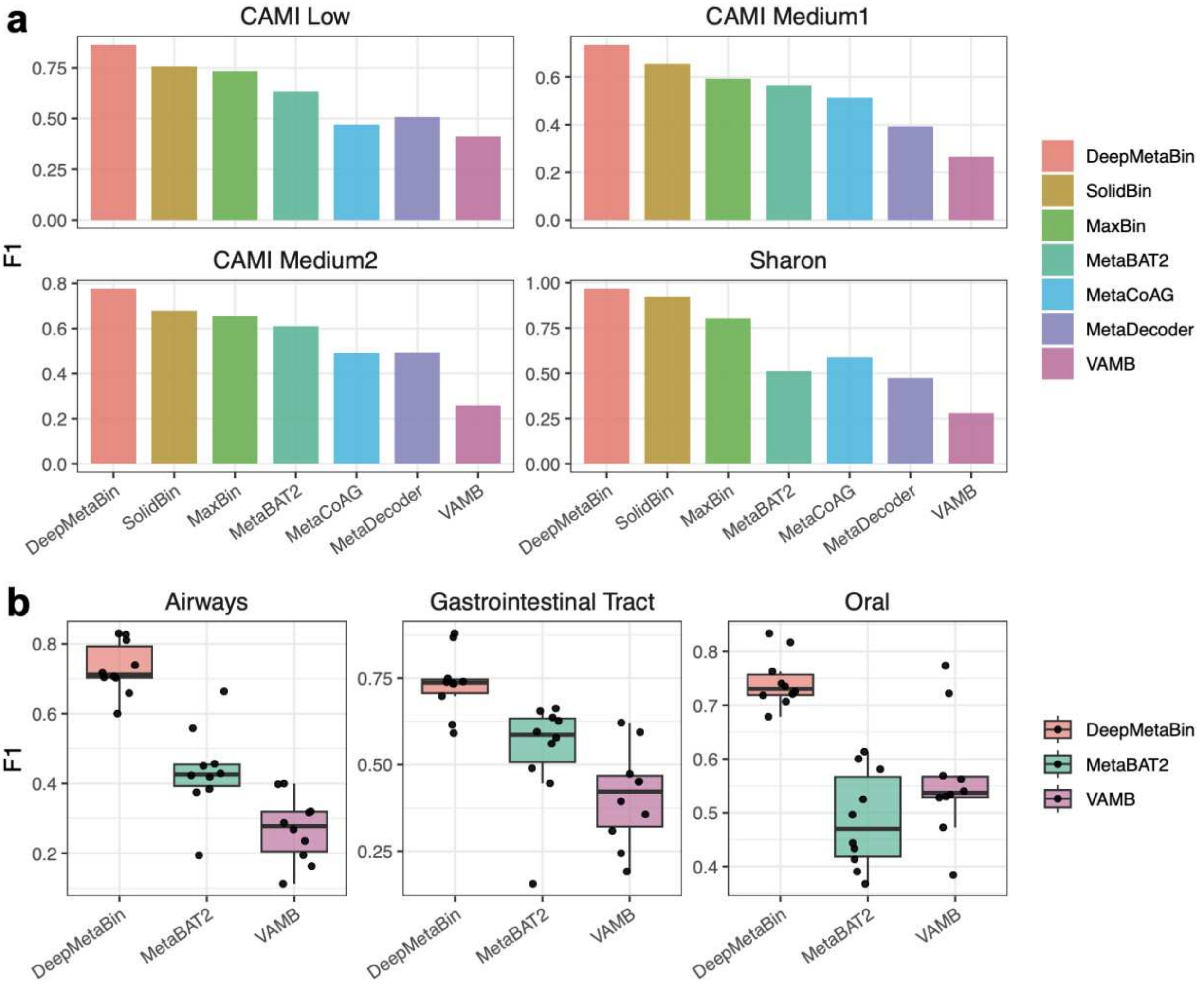

Figure 2. Comparison of overall F1 scores obtained by different binning tools on all the benchmarked datasets.

DeepMetaBin achieved the highest F1 scores than the other tools

DeepMetaBin was developed with both single-sample and multi-sample modes (**Methods**). We benchmarked single-sample mode of DeepMetaBin against MaxBin, MetaBAT2, SolidBin, MetaCOAG, MetaDecoder, and VAMB on CAMI I low complexity (CAMI Low)[17], CAMI I medium complexity 1 (CAMI Medium 1)[17], CAMI 1 medium complexity 2 (CAMI Medium 2)[17] and the Sharon dataset[15]. The multi-sample mode of DeepMetaBin was compared with VAMB and MetaBAT2 (**Methods**) on CAMI II Airways (Airways), CAMI II Gastrointestinal Tract (Gastrointestinal Tract), and CAMI II Oral (Oral) datasets[18], each containing 10 samples.

To validate the binning performance considering both long and short contigs, we obtained high-confidence taxonomy labels as the ground truth labels for the contigs longer than 1Kb and calculated the overall F1 score for the binning results (**Methods**). For single-sample binning, DeepMetaBin achieved the highest F1 scores among all the benchmarked contig binning tools, which were on average 1.55 times higher on CAMI Low, 1.62 times higher on CAMI Medium 1, 1.62 times higher on CAMI Medium 2, and 1.88 times higher on Sharon than the other tools (**Figure 2 a**).

DeepMetaBin also obtained much higher F1 scores than VAMB and MetaBAT2 on the multi-sample binning task (**Methods**). DeepMetaBin generated average F1 scores of 0.73, 0.73, and 0.74 for the samples in Airways, Gastrointestinal Tract, and Oral, respectively. The numbers are much higher than those of VAMB (Airways = 0.27, Gastrointestinal Tract = 0.41, and Oral = 0.56; **Figure 2 b**) and MetaBAT2 (Airways = 0.44, Gastrointestinal Tract = 0.54, and Oral = 0.49; **Figure 2 b**). The F1 scores of DeepMetaBin were also significantly higher than the F1 scores of VAMB (Wilcoxon rank-sum test p-value: Airways = 1.95e-3, Gastrointestinal Tract = 1.95e-3, Oral = 5.86e-3; **Figure 2 b**) and MetaBAT2 (Wilcoxon rank-sum test p-value: Airways = 1.95e-3, Gastrointestinal Tract = 1.95e-3, Oral = 1.95e-3; **Figure 2 b**).

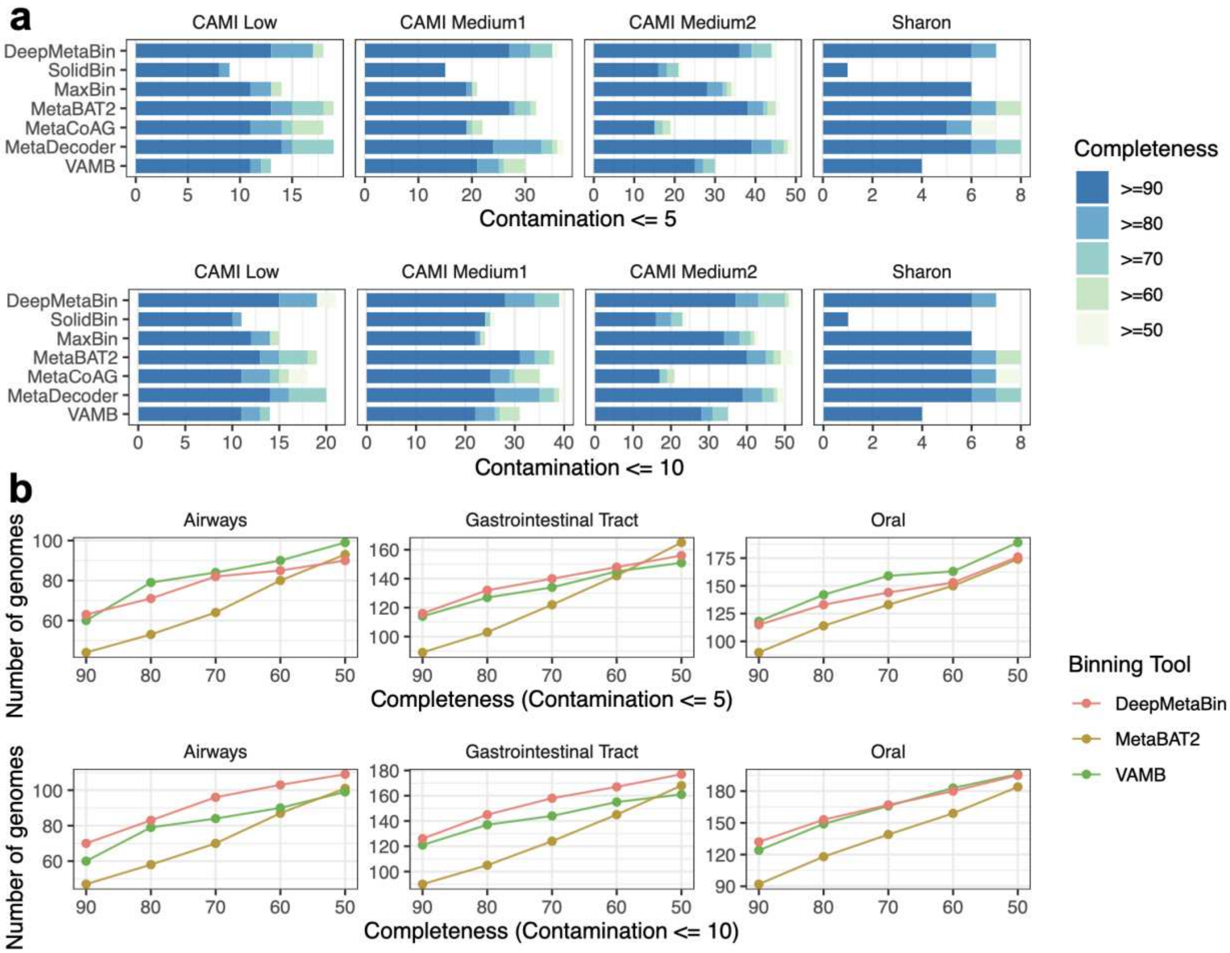

Figure 3. Comparison of the number of MAGs obtained by different metagenome binning tools satisfying a range of quality thresholds of completeness and contamination.

MAG quality evaluation on different binning tools

We evaluated MAG quality by counting the number of MAGs satisfying different levels of single-copy gene completeness (from >=90% to >=50%) and contamination (<=5% or <= 10%) (**Figure 3**; **Methods**). For single-sample contig binning, DeepMetaBin, MetaBAT2, and MetaDecoder obtained a comparable number of MAGs for different quality levels on all the single-sample datasets (**Figure 3 a**). The performance of the other four metagenome binning tools (SolidBin, MaxBin, MetaCoAG, and VAMB) generated a lower number of MAGs for different quality levels (**Figure 3 a**). On multi-sample contig binning, DeepMetaBin produced a slightly higher number of MAGs than VAMB and MetaBAT2, for completeness ranging from >=90% to >=50% and contamination of <=10% on the Airways and Gastrointestinal Tract datasets (**Figure 3 b**). These results indicate that DeepMetaBin generates MAG numbers that are comparable to state-of-the-art contig binning tools on both single-sample binning and multi-sample binning tasks.

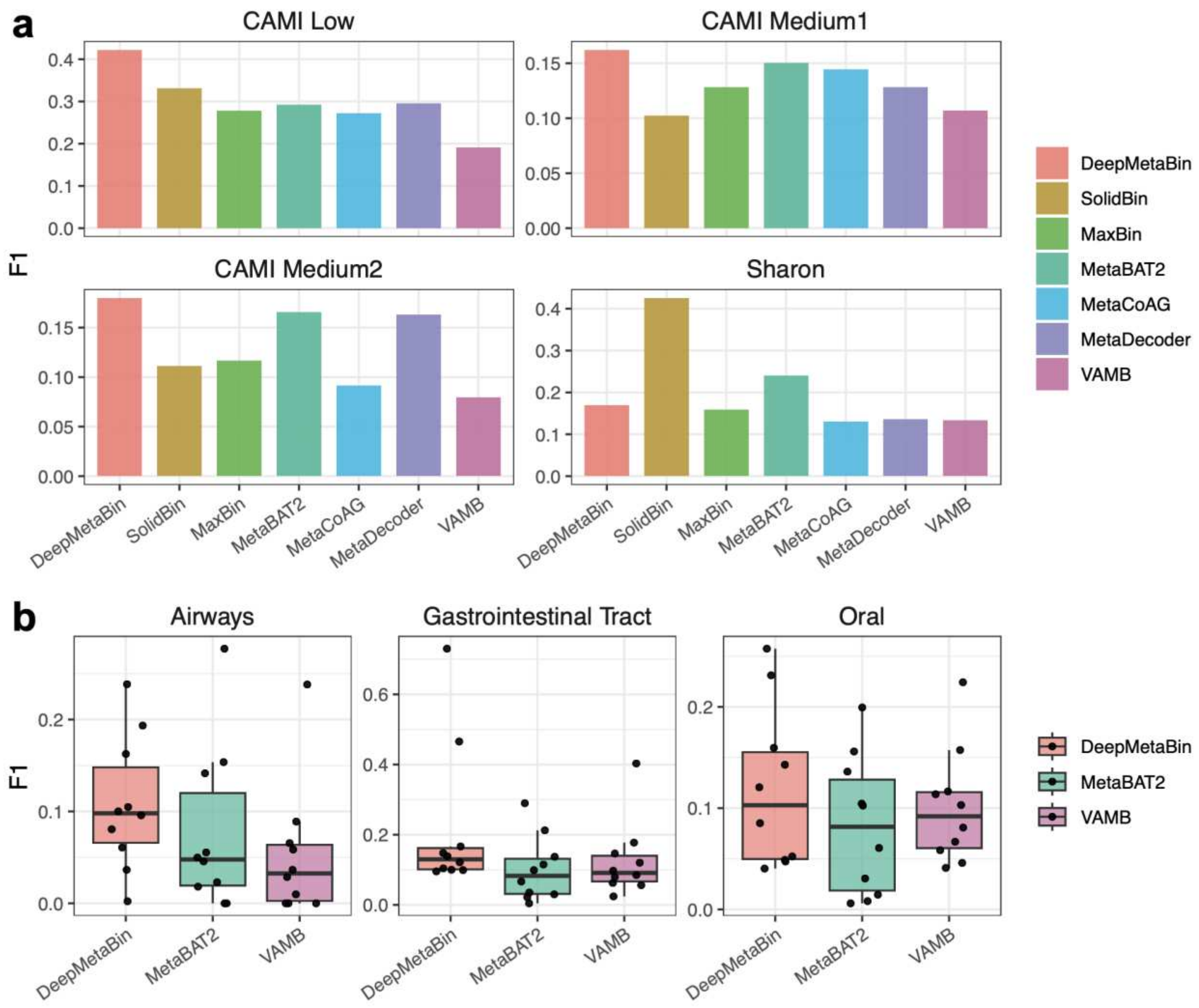

Figure 4. Comparison of F1 scores of NCMAGs obtained by different binning tools on all benchmarked datasets.

Comparison of F1 scores for NCMAGs

We further investigated F1 scores on NCMAGs to further evaluate the performance of DeepMetaBin. DeepMetaBin had superior F1 scores compared with all the other binning tools on CAMI Low (1.57 times on average), CAMI Medium 1 (1.30 times on average), and CAMI Medium 2 (1.59 times on average; **Figure 4 a**). On the Sharon dataset, Solidbin achieved the highest F1 score, but it only produced one NCMAG (number of NCMAGs: MaxBin = 6, MetaBAT2 = 6, VAMB = 4, SolidBin = 1, MetaCoAG = 5, MetaDecoder =6, DeepMetaBin = 6; **Figure 3 a**). MetaBAT2 also had a slightly better F1 score than DeepMetaBin on the Sharon dataset, but the performance of MetaBAT2 F1 scores was not consistent on all datasets (**Figure 4 a**). Except for the preceding two instances, DeepMetaBin achieved higher F1 scores than all the other metagenome binning tools on the Sharon dataset (**Figure 4 a**).

For multi-sample binning tasks, DeepMetaBin generated the highest F1 scores. It obtained an average F1 score of 0.11, 0.22, and 0.12 for the samples in the Airways, Gastrointestinal Tract, and Oral, respectively. These numbers were much higher than those obtained by VAMB (Airways = 0.05, Gastrointestinal Tract = 0.13, and Oral = 0.10; **Figure 4 b**) and MetaBAT2 (Airways = 0.08, Gastrointestinal Tract = 0.10, and Oral = 0.08; **Figure 4 b**). DeepMetaBin also generated significantly higher F1 scores than VAMB on Airways (Wilcoxon rank-sum test p-value: 1.95e-3; **Figure 4 b**), and MetaBAT2 on Oral (Wilcoxon rank-sum test p-value: 3.71e-2; **Figure 4 b**).

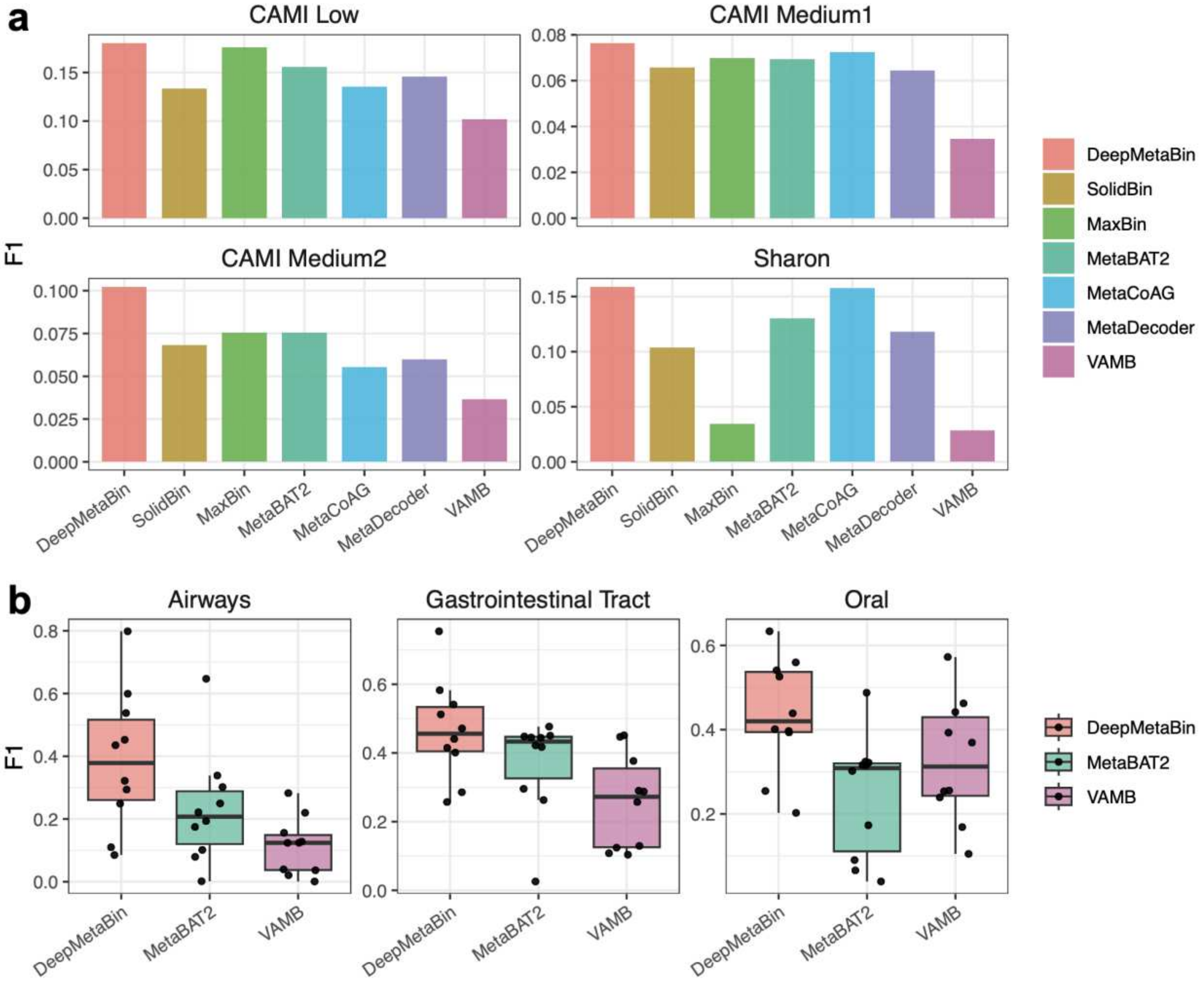

Figure 5. Comparison of F1 scores of medium-quality MAGs obtained by different binning tools on all benchmarked datasets.

Comparison of F1 scores for medium-quality MAGs

We next assessed the F1 scores for the medium-quality MAGs (**Methods**). DeepMetaBin generated medium-quality MAGs with the highest F1 score for all the single-sample datasets. It obtained on average 1.31, 1.30, 1.76, and 2.55 times higher F1 scores than the other binning tools on CAMI Low, CAMI Medium1, CAMI Medium2, and Sharon dataset, respectively (**Figure 5 a**). DeepMetaBin achieved the best F1 scores across the board for the medium-quality MAGs on all multi-sample datasets. DeepMetaBin obtained F1 scores averages of 0.39, 0.47, and 0.43 for the corresponding samples in Airways, Gastrointestinal Tract, and Oral, respectively (**Figure 5 b**). These F1 scores are much higher than those generated from the medium-quality MAGs of VAMB (Airways = 0.11, Gastrointestinal Tract = 0.26, and Oral = 0.33; **Figure 5 b**) and MetaBAT2 (Airways = 0.23, Gastrointestinal Tract = 0.37, and Oral = 0.24; **Figure 5 b**). DeepMetaBin also generated significantly higher F1 scores for medium-quality bin samples than VAMB on all three multi-sample datasets (Wilcoxon rank-sum test p-value: Airways = 1.95e-3, Gastrointestinal Tract = 9.77e-3, Oral = 1.37e-2; **Figure 5 b**) and MetaBAT2 on Airways and Oral (Wilcoxon rank-sum test p-value: Airways = 5.86e-3, Oral = 3.91e-3e-3; **Figure 5 b**).

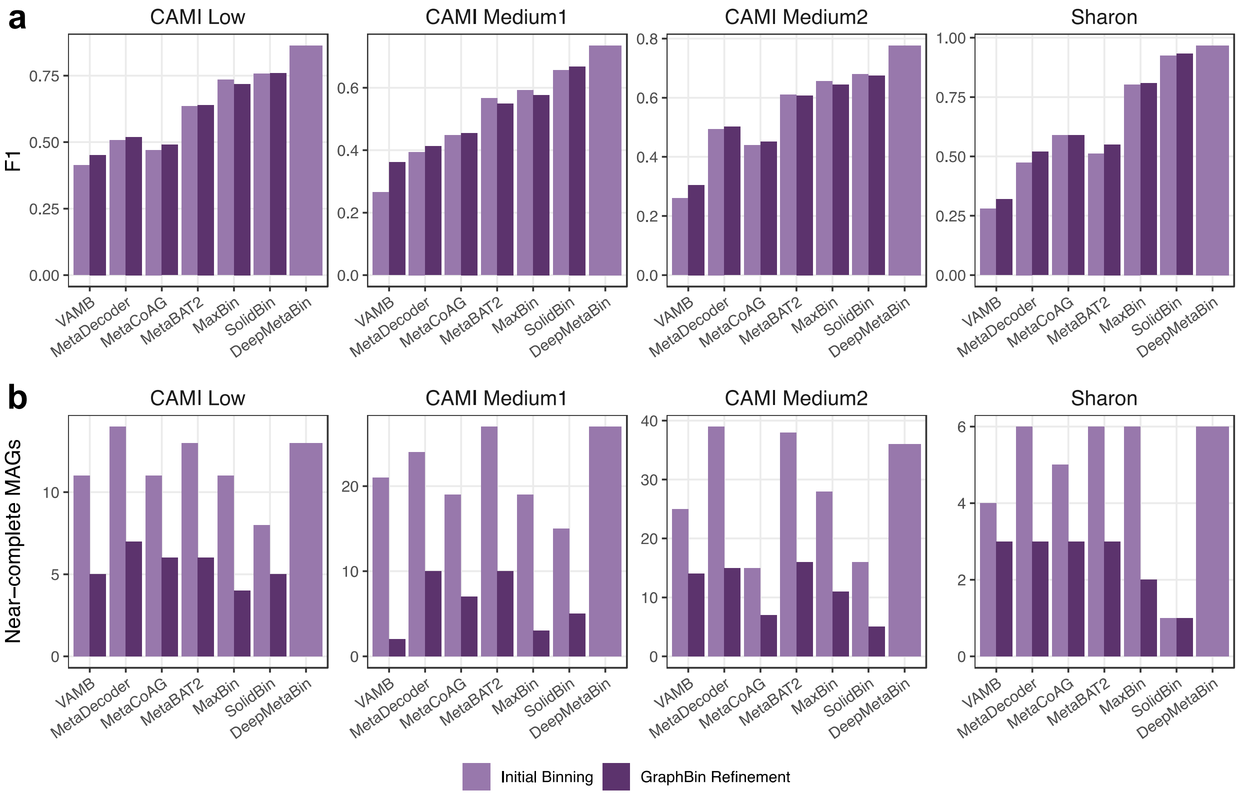

Figure 6. Comparison of F1 scores and the number of near-complete MAGs obtained by DeepMetaBin, and the other binning tools refined by GraphBin2.

DeepMetaBin outperforms GraphBin2 on both F1 scores and the number of NCMAGs

We compared DeepMetaBin with GraphBin2, a binning refinement tool that was developed to increase the F1 score of single-sample contig binning. We refined the initial binning results of VAMB, MetaDecoder, MetaCoAG, MetaBAT2, MaxBin, and SolidBin with GraphBin2 on the four single-sample datasets, and compared the results with DeepMetaBin. GraphBin2 marginally improved the F1 scores for most of the initial binning tools. The F1 scores remained much lower than those of DeepMetaBin. DeepMetaBin generated on average 1.50, 1.52, 1.58, and 1.75 times higher F1 scores than GraphBin2 refined binning results of all other tools on CAMI Low, CAMI Medium 1, CAMI Medium 2, and the Sharon dataset, respectively (**Figure 6 a**).

The F1 score increases brought about by GraphBin2 refining were at the cost of a substantially reduced number of NCMAGs compared to the initial binning tools (except for SolidBin on the Sharon dataset; **Figure 6 b**). GraphBin2 refining led to a loss of 35, 88, 93, and 13 NCMAGs from all six tools on CAMI Low, CAMI Medium 1, CAMI Medium 2, and the Sharon dataset, respectively (**Figure 6 b**). In contrast, the high F1 scores of DeepMetaBin came with only a slight decrease in the number of NCMAGs. DeepMetaBin produced only 1, 0, 3, and 0 less NCMAGs than the highest number of NCMAGs generated by all other binning tools on CAMI Low, CAMI Medium 1, CAMI Medium 2, and the Sharon dataset, respectively (**Figure 6 b**).

Discussion

Metagenome binning is a crucial step to generate MAGs from metagenomic sequencing data. The goal of metagenome binning is to cluster the contigs of the same microbial genome together in order to reconstruct the MAG. Recently developed metagenome binning tools, such as VAMB and MetaDecoder (**Figure 3**), performs well at creating NCMAGs, but they lose a significant fraction of short contigs during binning. Conventional metagenome binning tools, such as SolidBin and MaxBin, perform better binning on short contigs and obtain higher F1 scores, but they fail to produce state-of-the-art results in the number of NCMAG (**Figure 3**). To improve on the current compromise between high number of NCMAG and high F1 scores, we introduce DeepMetaBin, a graph-based GMVAE model with the ability to cluster long contigs (>2Kb) and short contigs (1Kb-2Kb) simultaneously. We decided to use a minimum contig length of 1Kb for DeepMetaBin because, in our experience, the sequence composition features below this threshold remain insufficiently informative for contig binning.

DeepMetaBin consists of three steps. Firstly, DeepMetaBin builds a k-NN graph on the contigs and uses DBSCAN and label propagation to refine the graph. This step generates accurate links between contigs. We investigated the graph on the CAMI Low dataset and found that 90.34% of the edges in the k-NN graph were connecting two contigs that belonged to the same microbial genome (**Supplementary Figure 1**). The k-NN graph provides the "must-links" for binning both the short (1Kb-2Kb) and long contigs (>2Kb). Secondly, DeepMetaBin trains a GMVAE and adds a loss function to require the contigs that have links in the k-NN graph to have a short distance in the latent embedding of GMVAE. In this way, GMVAE exploits the advantages of both the sequence composition features of long contigs and the accuracy of the k-NN graph to cluster short contigs. Thirdly, DeepMetaBin generates super clusters of contigs in the latent embedding of GMVAE using GMM, and further group contigs in each super cluster using constrained k-means with the constraint being that contigs sharing the same single-copy gene should not be clustered together.

For single-sample binning, we benchmarked DeepMetaBin against popular metagenome binning tools on CAMI Low, CAMI Medium 1, and CAMI Mdeium2 and Sharon datasets. For multi-sample binning, we evaluated the binning tools on three commonly used benchmarking datasets (CAMI II Airways. CAMI II Gastrointestinal Tract, and CAMI II Oral). The CAMI high complexity dataset is another commonly used dataset that doubles as a multi-sample dataset with 5 samples, but we excluded the CAMI high complexity dataset[17] from our analysis because it has insufficient read depth for each sample (5.4x) and thus cannot produce a reasonable assembly without co-assembly. On all the benchmarked datasets, DeepMetaBin achieved high F1 scores and a comparable number of NCMAGs.

We noticed some binning tools incorporate assembly graph to improve metagenome binning because it can provide the connectivity between contigs, such as MetaCoAG and GraphBin2. However, the contig connectivity offered by the assembly graph would largely depends on the quality of metagenome assembly and sequencing data. We observed both MetaCoAG and GraphBin2 could not achieve stable performance of NCMAGs on different datasets (**Figure 3 and Figure 6**). With refined k-NN, DeepMetaBin was the only contig binning tool that could achieve state-of-art performance on both the F1 score and the number of NCMAGs on all the benchmarked datasets. In addition, we found the graph-based binning methods may require additional computational resources to process links between contigs (**Supplementary Note 1**) and DeepMetaBin was observed more efficient than the other tools processing graph information.

Methods

Constructing accurate k-NN graphs using read depths and TNFs

We extracted read depths and TNF features from contigs using the same approach described in a previous study[9]. The two types of features are concatenated into a single contig feature vector for each contig. We constructed the initial k-NN graph on the sequence composition features using Euclidean distances. Given two contigs i and j, the Euclidean distance between these two contigs was calculated as $d_{ij} = \sqrt{\sum\left(f_i - f_j\right)^2}$. To construct the initial k-NN graph, we added edges between two contigs if one contig falls in the k $(k = 3)$ nearest neighbors of the other contig.

The initial k-NN graph may involve edges that mis-linked the contigs from different microbes. To refine the k-NN graph, we clustered the contigs in the k-NN graph and remove nodes whose neighbors are in different clusters. We denote E as the edge set in the k-NN graph, and we then first extracted a distance matrix D from the k-NN graph as follows

$$D_{ij} = \begin{cases} d_{ij} & if\ (i,j) \in E \\ +\infty & otherwise \end{cases}$$

We use DBSCAN with the distance matrix D to cluster the contigs initially. Some contigs will be unclustered after DBSCAN, so we perform label propagation[19] to propagate the clustering labels from DBSCAN to all contigs in the k-NN graph. Finally, we removed all the noise edges in the k-NN graph to guarantee all neighbors of a contig have the same labels with it.

Gaussian Mixture Variational Autoencoder (GMVAE)

We assume the contig feature vectors (x) are derived from latent variables of Gaussian mixture distribution. Specifically, we model the generative process of x as follows:

$$p(y) \sim uniform\left(\frac{1}{K}\right) \quad (1)$$

$$p(z|y_k = 1) \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right) \quad (2)$$

$$p_\theta\ (x|z) \sim \mathcal{N}(\mu_x, \sigma_x^2) \quad (3)$$

where y is an indicator vector sampled from a uniform distribution. The function of y is to select one Gaussian from the K Gaussian distributions to generate the latent variable z, and $y_k = 1$ denotes it selects the k^{th} Gaussian distribution parameterized by μ_k and σ_k^2, where the latent variable z is sampled. Then x is generated from another Gaussian distribution parameterized by

μ_x and σ_x^2, which are determined by a batch-normalized multilayer perceptron given z, with trainable parameter θ. We use μ_x to represent the latent embedding of x.

The inference process to infer latent variables y and z is modeled as follows:

$$q_{\phi_1}(y|x) = Multinomial\big(g_1(x; \phi_1)\big) \qquad (4)$$

$$q_{\phi_2}(z|x, y_k = 1) = \mathcal{N}(\tilde{\mu}_z, \tilde{\sigma}_z^2) \qquad (5)$$

where $\begin{bmatrix} \tilde{\mu}_z \\ \tilde{\sigma}_z^2 \end{bmatrix} = g_2(x, y; \phi_2)$; g_1 and g_2 denote neural networks with parameters ϕ_1 and ϕ_2. The detailed parameters in GMVAE are described in **Supplementary Figure 2**. The goal is to find the parameters that can maximize the log-likelihood of evidence $\ln \mathrm{p}_\theta(x_i)$, and we can maximize its evidence lower bond:

$$\max_{\phi_1, \phi_2, \theta} \ln \mathrm{p}_\theta(x_i) = ln \int_z \sum_y p_\theta(x_i, y, z) dz \geq E_{q_\phi(z, y|x_i)} \left[ln \frac{p_\theta(x_i, y, z)}{q_\phi(z, y|x_i)} \right] = L(\theta, \phi_1, \phi_2, x_i) \qquad (6)$$

The evidence lower bond can be further simplified as

$$L(\theta, \phi_1, \phi_2, x_i) = E_{q_\phi(z, y|x_i\,)} \left[ln p_\theta(x_i|z) + ln \frac{p_\theta(z|y)}{q_{\phi_2}(z|x_i, y)} + ln \frac{p_\theta(y)}{q_{\phi_1}(y|x_i)} \right] \qquad (7)$$

Equation (7) can be interpreted as the sum of the reconstruction loss, Gaussian loss, and categorical loss, respectively.

GMVAE with k-NN graph embedding

We aim to minimize the distance between two x_i and x_j in the latent embedding if they have an edge in the k-NN graph. We used a regularized optimization function of (6) with graph constraint can be written as

$$\max_{\phi_1, \phi_2, \theta} \sum_{i=1}^{N} \left(\ln \mathrm{p}_\theta(x_i) - \sum_{j \in \mathcal{N}(x_i)} w_{ij} JS\left(q_\phi(z, y|x_i), q_\phi\left(z, y|x_j\right) \right) \right) \quad (8)$$

where N denotes the number of contigs; $\mathcal{N}(x_i)$ denotes the neighbor nodes of x_i in the k-NN graph; $JS(\ldots)$ denotes the Jenson-Shannon divergence. We defined w_{ij} using Gaussian kernel with Softmax function as follows

$$w_{ij} = softmax\left(\exp\left(-\frac{\|x_i - x_j\|_2^2}{2 s_i^2} \right) \right), \quad j \in \mathcal{N}(x_i) \quad (9)$$

The weight w_{ij} is inversely proportional to the Euclidean distance of contig i and j. The softmax is used to constrain the weights of the neighbor edges of contig i sum up to one.

The equation (8) can be further transformed as

$$\max_{\phi_1,\phi_2,\theta} \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}(x_i)} w_{ij}\left(L(\theta,\phi_1,\phi_2,x_i)+L(\theta,\phi_1,\phi_2,x_i,x_j)\right) \quad (10)$$

where

$$L(\theta,\phi_1,\phi_2,x_i,x_j) = E_{q_\phi(z,y|x_j)}\left[ln\frac{p_\theta(x_i,y,z)}{q_\phi(z,y|x_j)}\right]$$

$$= E_{q_\phi(z,y|x_j\,)}\left[lnp_\theta(x_i|z) + ln\frac{p_\theta(z|y)}{q_{\phi_2}(z|x_j,y)} + ln\frac{p_\theta(y)}{q_{\phi_1}(y|x_j)}\right] \quad (11)$$

Considering (11), it can be seen from equation (10) that x_i not only reconstructs itself but also reconstructs its neighbor x_j in the k-NN graph. In this way, the features of the neighboring contigs of contig i in the k-NN graph will be integrated into the latent embedding of contig i. We used mini-batch gradient descent to determine $\phi_1,\phi_2,\theta,\{\mu_k,\sigma_k\}_{k=1}^{K}$ that optimizes (10).

Clustering in the latent embedding of GMVAE

GMVAE assumes that the latent distribution of data is a Gaussian Mixture Distribution. Therefore we first use a GMM to cluster the contigs in the latent embedding of GMVAE. Furthermore, to separate the genomes with similar sequence composition features that cannot be told apart by GMVAE, we perform a secondary clustering on highly contaminated bins. Two contigs of the same genome cannot share the same single-copy gene (SCG), so the contig pairs sharing SCG are considered as “cannot-links”. With “cannot-links”, we applied a constrained k-means algorithm[20] on the clusters with high contamination generated by GMM to further separate genomes. We use SCG to determine the number of clusters in the contigs. For the primary clustering with GMM, we set the number of clusters to 7/8 of the number of clusters estimated by SCG, since some genomes can have similar sequence composition features that would be naturally clustered into the same cluster in this step. For the secondary clustering with constrained k-means, we use the exact cluster number estimated by SCG.

Metagenome binning and evaluation

For single-sample binning, the reads are assembled into contigs by metaSPAdes (v3.15.0)[14]. We compared DeepMetaBin with SolidBin (v1.3), MaxBin (v2.2.7), MetaBAT2 (v2.12.1), MetaCoAG (v1.0), MetaDecoder (v1.0.11), and VAMB (v3.0.3). For multi-sample binning, we assembled each sample into contigs independently using metaSPAdes (v3.15.0). We benchmarked DeepMetaBin with VAMB (v3.0.3) and MetaBAT2 (v2.12.1) on the multi-sample datasets. VAMB supports multi-sample binning by default. We used the same binning strategy proposed in[21] to run MetaBAT2 in multi-sample mode. All binning tools were run with default parameters. CPU time, real time, and maximum RSS were reported by "/usr/bin/time -v".

For the CAMI datasets, we aligned the contigs to the reference genomes using minimap2 (v2.17)[22] with parameter "-ax asm5". We filtered out all secondary alignments and assigned the aligned genomes as the true labels of the contigs. If a contig could be mapped to more than one genome, its label was removed since the true genome of the contig is then ambiguous. Since reference genomes were not available for the Sharon dataset, we used kraken2 (v2.1.2)[23] with its standard database to annotate the contigs while setting the "--confidence" as 0.9.

We counted the number of contigs belonging to the most frequent species in each MAG and calculated the fraction of those contigs in all the MAGs as the precision score. For each annotated species, we also counted the number of contigs belonging to the most frequent MAG in this annotated species and calculated the fraction of those contigs in all the annotated species as the recall score. The F1 score was obtained by $2 \times precision \times recall/(precision + recall)$.

We evaluated the completeness and contamination of the MAGs using CheckM (v1.1.2)[24]. MAGs with over 90% completeness and less than 5% contamination are defined as NCMAGs, while those with completeness over 60% and contamination less than 20% are defined as medium-quality MAGs.

Availability of data and materials

The code of DeepMetaBin is available at https://github.com/mx-ethan-rao/deepmetabin. The CAMI Low dataset was downloaded at "1st CAMI Challenge Dataset 1 CAMI_low" from https://data.cami-challenge.org/participate. CAMI Medium 1 and CAMI Medium 2 datasets were obtained at "1st CAMI Challenge Dataset 2 CAMI_medium" from https://data.cami-challenge.org/participate. CAMI II Airways, CAMI II Gastrointestinal Tract, and CAMI II Oral were downloaded from "2nd CAMI Toy Human Microbiome Project Dataset" at https://data.cami-challenge.org/participate. The reads of the Sharon dataset were downloaded in the NCBI sequence read archive ID SRX144807.

Competing interests

The authors declare no competing interests.

Ethics approval and consent to participate

Not applicable

Funding

This research was partially supported by Shenzhen, Shenzhen 518000, China (BGIRSZ20220014), the Hong Kong Research Grant Council Early Career Scheme (HKBU 22201419), HKBU Start-up Grant Tier 2 (RC-SGT2/19-20/SCI/007), HKBU IRCMS (No. IRCMS/19-20/D02), the Guangdong Basic and Applied Basic Research Foundation (No. 2021A1515012226) and Shenzhen Science and Technology Innovation Commission (SZSTI) - Shenzhen Virtual University Park (SZVUP) Special Fund Project (No. 2021Szvup135).

Authors' contributions

LZ conceived the study. ZMZ, LZ and MXR designed the algorithms in DeepMetaBin. LZ and ZMZ conceived the experiments. MXR implemented DeepMetaBin. MXR and ZMZ conducted the experiments. ZMZ, LZ and MXR analyzed the results. ZMZ and MXR wrote the manuscript. LZ and YFH revised the manuscript. All authors reviewed the manuscript.

References

1. Berg, G. et al. Microbiome definition re-visited: old concepts and new challenges. *Microbiome* **8**, 103 (2020).
2. Bharti, R. & Grimm, D.G. Current challenges and best-practice protocols for microbiome analysis. *Brief Bioinform* **22**, 178-193 (2021).
3. Quince, C., Walker, A.W., Simpson, J.T., Loman, N.J. & Segata, N. Shotgun metagenomics, from sampling to analysis. *Nat Biotechnol* **35**, 833-844 (2017).
4. Yang, C. et al. A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data. *Comput Struct Biotechnol J* **19**, 6301-6314 (2021).
5. Ayling, M., Clark, M.D. & Leggett, R.M. New approaches for metagenome assembly with short reads. *Brief Bioinform* **21**, 584-594 (2020).
6. 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**, 605-607 (2016).
7. Kang, D.D. et al. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**, e7359 (2019).
8. Wang, Z., Wang, Z., Lu, Y.Y., Sun, F. & Zhu, S. SolidBin: improving metagenome binning with semi-supervised normalized cut. *Bioinformatics* **35**, 4229-4238 (2019).
9. Nissen, J.N. et al. Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotechnol* **39**, 555-560 (2021).
10. Mallawaarachchi, V. & Lin, Y. in Research in Computational Molecular Biology: 26th Annual International Conference, RECOMB 2022, San Diego, CA, USA, May 22–25, 2022, Proceedings 70-85 (Springer, 2022).

11. Liu, C.C. et al. MetaDecoder: a novel method for clustering metagenomic contigs. *Microbiome* **10**, 46 (2022).

12. Lu, Y.Y., Chen, T., Fuhrman, J.A. & Sun, F. COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. *Bioinformatics* **33**, 791-798 (2017).

13. Pan, S., Zhu, C., Zhao, X.-M. & Coelho, L.P. A deep siamese neural network improves metagenome-assembled genomes in microbiome datasets across different environments. *Nature Communications* **13**, 2326 (2022).

14. Nurk, S., Meleshko, D., Korobeynikov, A. & Pevzner, P.A. metaSPAdes: a new versatile metagenomic assembler. *Genome research* **27**, 824-834 (2017).

15. Sharon, I. et al. Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome Res* **23**, 111-120 (2013).

16. Mallawaarachchi, V., Wickramarachchi, A. & Lin, Y. GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics* **36**, 3307-3313 (2020).

17. Sczyrba, A. et al. Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. *Nat Methods* **14**, 1063-1071 (2017).

18. Meyer, F. et al. Critical Assessment of Metagenome Interpretation: the second round of challenges. *Nat Methods* **19**, 429-440 (2022).

19. ZhuЃ, X. & GhahramaniЃн, Z. Learning from labeled and unlabeled data with label propagation. *ProQuest Number: INFORMATION TO ALL USERS* (2002).

20. Baumann, P. & Hochbaum, D.S. A k-means algorithm for clustering with soft must-link and cannot-link constraints. (2021).

21. Mattock, J. & Watson, M. A comparison of single-coverage and multi-coverage metagenomic binning reveals extensive hidden contamination. *Nat Methods* **20**, 1170-1173 (2023).

22. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094-3100 (2018).

23. Wood, D.E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome biology* **20**, 1-13 (2019).

24. Parks, D.H., Imelfort, M., Skennerton, C.T., Hugenholtz, P. & Tyson, G.W. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome research* **25**, 1043-1055 (2015).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryTextandFigures.pdf
- nrsoftwarepolicy.pdf
- nrreportingsummary.pdf