nature portfolio

Peer Review File

LorBin: Efficient binning of long-read metagenomes by multiscale adaptive clustering and evaluation

Corresponding Author: Professor Gaofei Jiang

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
Major Revisions:

* While an accuracy is used, it is calculated by matching with the best assignment. In order to convince the reader, ARI or Purity/Completeness should be shown, like the CAMI2 competition computed.

* The paper would benefit from further discussion on how LorBin's computational efficiency compares with other methods. If this
computational efficiency is a strength of LorBin, it is best to quantify it.

* It is unclear what the different shapes of t-sne clusters mean. What does it mean to have a concave cluster? why do the number of concave clusters increase? Wouldn't the convexity and concavity shape depend on the t-sne parameters being used, since t-sne is an arbitrary transformation of data? Why is studying the shape of clusters so important here?

* It is unclear whether the cluster quality assessment model is pretrained. Include more detail on the training process of the reclustering decision model.

* Also, justification of the architecture would help: Why the crossnet/deepnet architecture. Please explain why the clustering algorithms are needed in the architecture and justify how DBSCAN and BIRCH complement each other in the various steps. What led to the use of both the clustering algorithms and the autoencoders — show why you chose the VAE and DBSCAN and Birch to integrate into LorBin. More rationale is needed for utilizing a deep-learning model for quality assessment could strengthen the theoretical base for this paper, as some researchers may be hesitant due to the relative lack of interpretability of NN solutions. Write a more fleshed out introduction/background section that describes some of the key aspects of approaches that were tested against in benchmarking. This would provide the reader with further insight into your inspirations and how your design decisions are different from the other algorithms.

* Explain how the system performs with some of the different parts being altered/removed as to understand which components of the system are most integral. For example, if the reclustering decision model were to be replaced with a simple approach weighing the completeness and contamination values against some threshold, how would this impact performance.

Minor Revisions:

* In line 337 and equation 7, it looks like output of the deepnet module may have inconsistent labeling - sometimes it seems to be referred to as C_j, and other times as D_j.

* "first" misspelled/typo'd in fig. 3

* (270-271) "The output of the μ layer is the mean of the latent, while the output is the variance of the latent." I believe they meantt to put "the output of σ is the variance"

* while well known to the assembly field, justify/put a reference for using tetranucleotide features

* Fig. 3 caption says “The analysis of cluster shapes is shown in Supplementary Figure 8.” but it’s really in Supplementary Figure 12.

(Remarks on code availability)
I have not installed it, but it looks reasonable.

Reviewer #3

(Remarks to the Author)
Wei Xue et al. introduces a binner binner, LorBin. It achieves state of the art performance by combining known techniques from existing, modern binners:
* Common binning features (k-kmer composition and abundance) integrated with a VAE
* Multi-step reclustering informed by single-copy genes
* Being optimized for accurate, long reads such as HiFi reads

With a novel adaptive and hybrid clustering scheme that handles deferentially abundant genomes better than previous binners.

Overall, I find the manuscript well written, and the claims of accuracy are convincingly supported by extensive testing and optimization, as well as excellent ablation studies.
The work shown in this manuscript is a valuable contribution to the field and I would be excited to be able to run LorBin myself.
While I would like to see this work published, I do find the manuscript is lacking in some areas and have some issues with the manuscript as-is:

Major points:

You claim that LorBin is reference-free, but it cannot be reference-free if it uses a set of reference single-copy genes (SCGs). From skimming the source code, these SCGs are found with a HMM search using SCG signatures taken from a database. You could argue that SCGs generalize to large swaths of the tree of life than e.g. sequence alignment, and therefore LorBin is not dependent on the precise species/genus represented by a bin being present in databases, but it's wrong to call it reference-free.
This matters, because as demonstrated in e.g. the CheckM2 paper, the completion of genomes from clades are systematically under- or overestimated using fixed SCGs (because they may lack some of them, or may contain multiple copies).

In my opinion, applying t-SNE on a VAE embedding makes little sense and is bound to give misleading results. The VAE prior, enforced by the regularization loss ("KLoss", Eq 3 in the manuscript), is that the data is distributed as a multivariate unit Gaussian around the origin. Therefore, to reduce KLoss the VAE will attempt to embed the contigs in an N-dimensional ball around the origin, to the extent allowed by the RLoss.
This implies that the VAE will attempt to spread the data into as many dimensions as available. And this in turn means that one cannot assume the data lies in some lower-dimensional manifold that can be approximated by t-SNE. On the contrary, one can assume the data is intrinsically high-dimensional and can't be reduced to 2 dimensions!
For this reason, I don't think any of the analysis on the t-SNE projection (lines 540-555) are meaningful, and they may even be misleading.
I suggest the authors either remove this analysis, or else support its validity e.g. by doing a PCA of the embedding and showing that the top two components explain the majority of the total variance, and that the data therefore lies on a two-dimensional plane.

The measurement of performance on the simulated dataset is quite nonstandard. Typically, other papers presenting new binners count the number of high quality bins, or the mean bin recall or precision or bin F1 distribution. There are also tools such as BinBencher or AMBER to compute various bin statistics which may be useful (disclaimer: I authored BinBencher, so it would not be appropriate for me to suggest it specifically). While I have no reason to doubt that LorBin would also perform well as measured using these more classical metrics, I think it would strengthen the paper to also report them.

Computational efficiency is an important metric of new methods, since slow or memory-hungry methods may make some methods prohibitive in practice.
It would be better if the authors presented resource use (e.g. RAM and CPU/GPU time) for LorBin along with at least a few other methods, e.g. the faster MetaBAT2 or VAMB, and the slower COMEBIN.
If possible, it would also be informative to show how runtimes scale with the number of contigs. Can I run LorBin on a dataset with a million contigs?

The discussion section does a good job of summarizing the content above it, but there is little perspectives about what the success of LorBin implies for binning. How does the authors attribute the performance of LorBin compared to existing methods which use some of the same techniques as LorBin (e.g. Vamb uses a VAE, SemiBin2 uses a SCG-guided clustering which strikes me as similar to LorBin's, COMEBIN is also optimized for long reads)? What novelties from LorBin could other authors in the future use for the next generation binners? What work is left to be done?
I believe this is important for the field, because to my count, more than 20 binners have been published the last decade, each claiming superior performance. If the field is to actually progress, it's important that authors explain how they advanced the status quo and how future binners may advance it still further.

Minor points
* On line 118, the abbreviation hmBin should be explained.
* Line 150: It's not clear if this means that LorBin has a coefficient 0.36 (etc) higher than the second-best binner tested? Or is this the actual silhuette coefficient of LorBin? If so, what is the coefficient of the other binners?
* Line 160: Mention that this is according to CheckM2. In some papers, this information could perhaps be mentioned in the methods section only, but here, I think it's crucial to mention how this result was obtained.
* Line 430: "trains" should be "strains"

It is my policy to sign reviews,
Jakob Nybo Nissen, University of Copenhagen

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The authors answered most of our questions. Most edits were done with detail, and we thank the authors.

I still would like more biological meaning put on concave and convex clusters. Why are genome 1, 2, and 4 more complex? Can you discuss or at least hypothesize into why the different genomes made these clusters — since I would like know the biological significance of these different shapes of clusterings. Can you investigate more about this? Is there any computation->biological intuition?

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
Thank you for your thorough response.
All major points have been addressed to my satisfaction, except one which I believe still requires changes.

I'll address your responses to major issues raised by the reviewers point by point:

1. Claiming LorBin is reference free
A: The authors satisfyingly resphrased LorBin as unsupervised.

2. Use of non-standard accuracy measures
A: The authors have included the more common ARI and F1 metrics, which is satisfying. I believe there are some methodological issues with ARI scores for metagenomic binning - see e.g. https://viralinstruction.com/BinBencherBackend.jl/dev/considerations/#ari. Nonetheless, I think this reservation of mine is minor, somewhat subjective, and should not block the publication of LorBin.

3. Lack of measurements of computational requirements
A: Authors have added supplementary tables with the requested information.

4. Lack of discussion about how exactly LorBin advances the field
A: The authors have addressed this satisfactory by rewriting the discussion section.

5. t-SNE is inappropriate to represent a VAE embedding
A: In my opinion, this has not been satisfactory addressed. Reviewer 1 asked a similar question, and the authors show in Response Figure 2 to reviewer 1 that the number of convex and concave clusters only changes slightly with t-SNE parameters. However, a) the fact that it changes, despite convexity being a mathematically rigorous definition does seem to

prove that the t-SNE distorts the cluster shape, and b) the figure does not show whether the same clusters remain convex or concave across all 20 t-SNE settings - it could be that they switch, but with a fixed probability.
To address the same question in my review, the authors provide Response Figure 2 for reviewer 3, which shows that convexity change between PCA and t-SNE, and also differ as t-SNE parameters change. The authors does show that the convexity count is highly correlated when measured on PCA and t-SNE embeddings. Like above, they do not show that the same clusters remain convex/concave under PCA and t-SNE, and between t-SNE settings.
In response figure 3, the authors show that, when manually adding convex or concave clusters, the number of convex and concave clusters detected with t-SNE correlates with the true number.
However, only about half the number of truly convex clusters are detected in the t-SNE embedding, and approximately 20% more concave clusters are detected than what is manually added to the embedding. Further, they do not provide evidence that it's the same clusters that are actually convex/concave that are detected as such.

I conclude that the use of t-SNE remains unwarranted. The authors show, at best, that there is strong correlation between the true number of convex/concave clusters and the measured number in a simulated setting, and that the number of measured convex clusters differ by less than 20% between PCA and t-SNE and between different t-SNE runs.
But this is not the same as showing that clusters that are convex in high dimensions are also convex in t-SNE - it is, in fact proof, that at least some of the time, t-SNE changes the shape of the clusters.

I suggest the authors measure convexity directly in the high-dimensional space, if possible, or else simply forgo the mentioning of cluster shapes altogether, since it is not crucial to the paper's findings.

(Remarks on code availability)

Version 2:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
Thank you for elaborating.
I won't push it any further for studying how biology shapes the clusters, But perhaps this can be an interesting follow-up study.I am satisfied.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
Thank you for your response.
This round of revision concerned itself with the appropriateness of low-dimensional visualisation of a high-dimensional space, and in particular, whether the notion of "convexity" is preserved when reducing from higher- to lower dimensions.
I remain unconvinced that it's a good idea to attempt to reason about high-dimensional shapes as 2D projections, especially when there is good a priori reasons to believe that the overall shape of sequences is inherently high dimensional and therefore cannot be reduced.
However, I don't believe this disagreement is sufficiently important to prevent the publication of this paper. After all, it's not a critique of the methodology or even of an accuracy measurement, but only a critique of a visualisation meant to explain how LorBin works.

(Remarks on code availability)

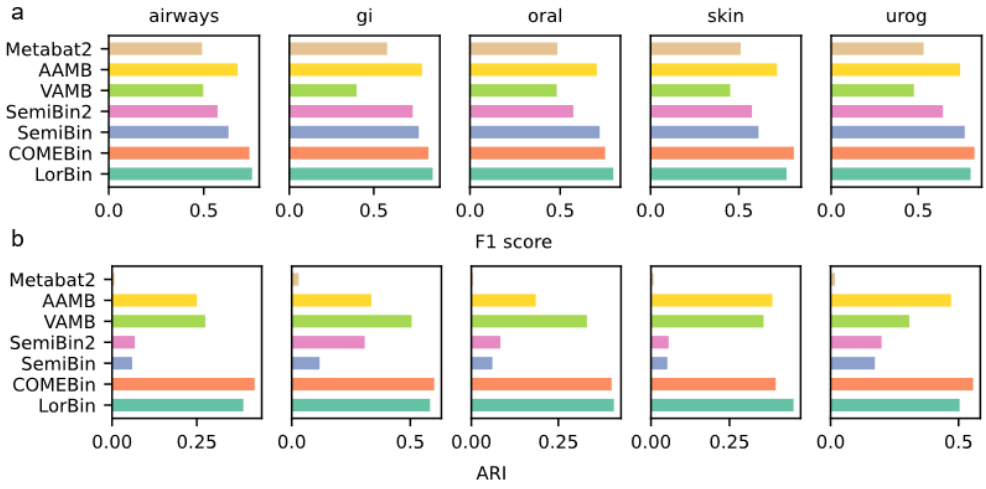
Response Letter to Reviewer Comments

Comments from Reviewer #1:

Major Revisions:

1. While an accuracy is used, it is calculated by matching with the best assignment. In order to convince the reader, ARI or Purity/Completeness should be shown, like the CAMI2 competition computed.

Reply: Thank you for your suggestion. We agree with you and to follow your suggestions, we have included more convincing indicators now, and we used a new supplementary figure with the ARI and F1 scores, which combines purity and completeness, as shown in the current Supplementary Figure 8 and the main text.



Supplementary Figure 8. The F1 score and the adjusted Rand index (ARI) of the results from different binners on five datasets. (a) F1 scores of the results from different binners on five datasets. The F1 score is calculated by completeness and contamination. (b) Adjusted Rand index (ARI) of the results from different binners on five datasets.

Supplementary Figure 7b). We also compared the adjusted Rand index (ARI) and F1 score of the binning performance across different binners. LorBin achieved the highest F1 score in the airway, gastrointestinal, and oral environments and the highest ARI in the oral and skin environments (Supplementary Figure 8). Overall,

2. The paper would benefit from further discussion on how LorBin's computational efficiency compares with other methods. If this computational efficiency is a strength of LorBin, it is best to quantify it.

Reply: Thank you for your suggestion. The computational efficiency includes two phases: 1) from contigs to the latent space and 2) the clustering algorithm. We performed two new experiments to further compare the overall efficiency of the binning method on four human intestinal datasets (SRR13128012, SRR13128013, SRR13128014, and ERR9765783) using only a GPU or CPU. In the first phase, LorBin was the most efficient of the three different encoders (Supplementary Table S1, Supplementary Table S2). For the entire binning process, LorBin ranked fourth using CPU (Supplementary Table S3), and third using GPU (Supplementary Table S4). Compared with the SOTA methods, LorBin is 2.3-25.9 times faster than SemiBin2 and COMEBin. The fastest method is MetaBat2 since it is a nonneural network. Please find our new section and the screenshot on our changes.

In Additional file

Datafram	Description
Table S1	Supplementary Table 1: Comparison of the encoder training time cost using the CPU mode on the human intestinal simulation datasets
Table S2	Supplementary Table 2: Comparison of the training time cost of the encoder used in GPU mode on the human intestinal simulation dat
Table S3	Supplementary Table 3: Comparison of the time cost of binners using the CPU mode on the human intestinal simulation datasets.
Table S4	Supplementary Table 4: Comparison of the time cost of binners using the GPU mode on the human intestinal simulation datasets.
Table S5	Supplementary Table 5: Comparison of the memory cost of binners using the GPU mode on the human intestinal simulation datasets.
Table S6	Supplementary Table 6: Statistical data of all media-and-high quality MAGs.

Supplementary Table 0

CPU/s	SRR13128912	SRR13128913	SRR13128914	ERR9765783
Metabat2	3.6	3.1	3.1	6.6
VAMB	175.1	124.0	163.2	142.7
AAMB	187.6	168.1	172.6	174.6
SemiBin	2583.7	1945.3	1577.4	2080.0
SemiBin2	2333.0	1912.7	1632.6	2128.1
COMBin	4442.4	3297.5	4404.6	3725.7
LorBin	232.5	202.6	170.2	550.9

Supplementary Table S3

GPU/s	SRR13128912	SRR13128913	SRR13128914	ERR9765783
VAMB	45.3	40.5	40.8	40.3
AAMB	131.5	126.9	115.5	115.9
SemiBin	1774.9	1474.2	1206.5	1674.1
SemiBin2	1735.7	1429.4	1194.7	1676.2
COMEBin	532.2	552.0	549.3	1094.5
LorBin	154.6	134.6	115.6	482.6

Supplementary Table S4

	SRR13128012	SRR13128013	SRR13128014	ERR9765783
LorBin	4.35GB	4.32GB	4.32GB	4.41GB
SemiBin	9.10GB	8.45GB	7.75GB	8.24GB
Semibin2	8.87GB	8.42GB	7.80GB	8.22GB
VAMB	4.31GB	4.17GB	4.18GB	4.18GB
AAMB	4.17GB	4.17GB	4.17GB	4.17GB
COMEBin	4.24GB	4.24GB	4.24GB	4.24GB
MetaBAT2	0.09GB	0.09GB	0.09GB	0.17GB

Supplementary Table S5

In Result section

LorBin optimization in binning performance

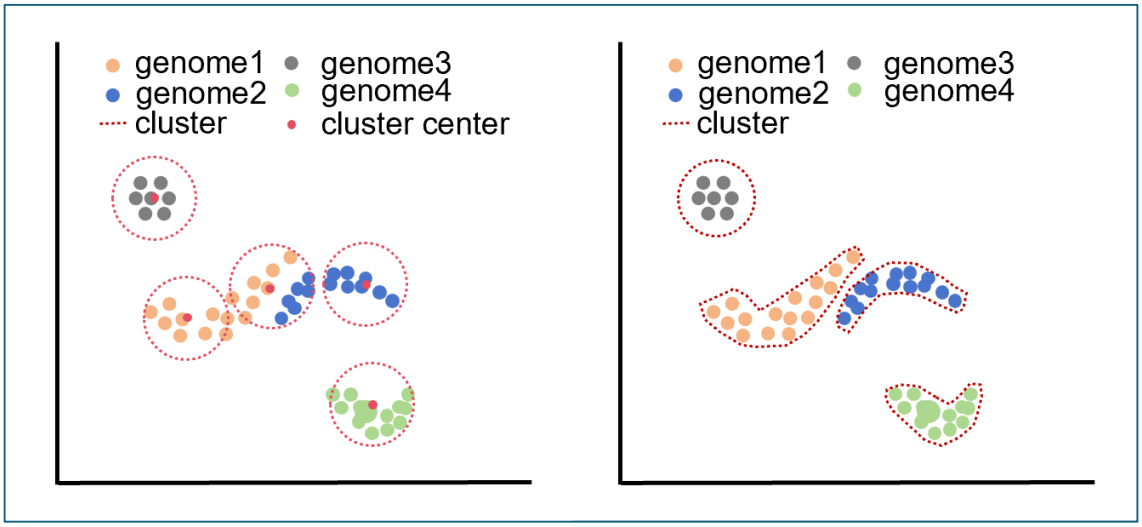
Metagenome binning comprises two default processes: feature extraction and clustering. LorBin comparison experiments were conducted to reduce resource usage and improve binning results (Methods). For feature extraction, we found that the VAE was more efficient at extracting the embedded features of contigs than other two encoders either using CPUs (Supplementary Figure 2 and Supplementary Table 1) or GPUs (Supplementary Figure 2 and Supplementary Table 2). Compared with SOTA methods, LorBin was 2.3-25.9 times faster than SemiBin2 and COMEBin under normal memory consumption (Supplementary Table 3 - Table 5). In the clustering stage, DBSCAN and BIRCH were determined as the two best approaches out of 12 different clustering algorithms for the embedded features (Supplementary Figure 1). To further improve the binning performance, we

3. It is unclear what the different shapes of t-sne clusters mean. What does it mean to have a concave cluster? why do the number of concave clusters increase? Wouldn't the convexity and concavity shape depend on the t-sne parameters being used, since t-sne is an arbitrary transformation of data? Why is studying the shape of clusters so important here?

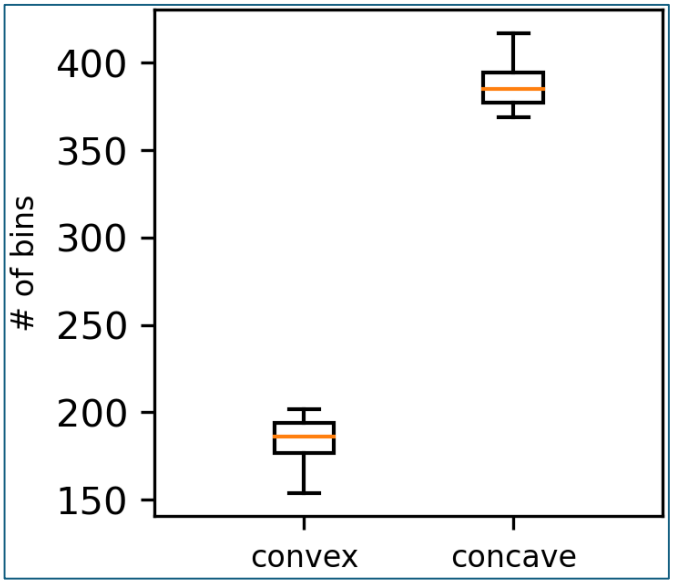
Reply: We appreciate your insightful comments. Convex clusters are regularly shaped (genome 3 in Response Figure 1), whereas concave clusters tend to be more complex[1] (genome 1, genome 2, and genome 4 in Response Figure 1). Owing to the intricate shapes of concave clusters, specialized processing techniques are needed[1–3]. In the biological context, variations in bacterial genome features lead to these differing shapes (Figure 3e), thus impacting their proximity in the latent space[4]. Contigs from closed genomes within the latent space often result in misclassification when the cluster shapes are not adequately considered (genome 1 and genome 2 in Response Figure 1).

In the second stage of clustering, the number of concave clusters decreases (Supplementary Figure 14a), whereas their proportion increases compared to the first stage (Supplementary Figure 14b). This finding demonstrates that our multiscale DBSCAN can effectively address clusters with different shapes and complexities, which is consistent with the results from the clustering algorithm ablation experiment (Supplementary Figure 1). These findings highlight the robustness of the two-stage clustering algorithm in processing both concave and convex clusters effectively.

T-SNE, a widely used dimensionality reduction method in bioinformatics[5,6], was applied to show the ability of LorBin to address clusters with different concavities and convexities. Indeed, T-SNE is out of our model. By generating 20 random parameter sets for T-SNE and repeating cluster shape detection, we found that the proportion of convex and concave clusters in the overall population remained relatively stable (Respond Figure 2). While the T-SNE parameters influence cluster shape detection, the effect is limited. The overall distribution of cluster characteristics remains observable. t-SNE analyses demonstrated that LorBin is applicable to both convex and concave clusters (Supplementary Figure 14).



Response Figure 1. Potential outcomes of clustering without accounting for convex and concave clusters: When two genomes form concave clusters that are close to each other in the latent space (*e.g.*, genome 1 and genome 2 on the left plot), ignoring their shapes can lead to incorrect results. The right plot shows the correct clustering taking into account the shape of the clusters. Specifically, contigs from different genomes may be grouped into the same cluster because clustering is based solely on the proximity of the center points. This approach increases bin contamination. In this scenario, orange dots represent contigs from genome 1, blue dots represent contigs from genome 2, yellow dots represent contigs from genome 3, and light green dots represent contigs from genome 4. The red dotted line indicates the clustering boundary that considers only convex clusters, and the red dots mark the cluster centers. Notably, genomes 1, 2, and 4 exhibit concave shapes, whereas genome 3 displays a convex shape.



Response Figure 2. Convexity and concavity shapes of different t-SNE parameters. Twenty randomly generated t-SNE parameter sets were used to repeat cluster shape detection. The box plot on the left shows the distribution of the number of convex clusters across the 20 experiments, whereas the box plot on the right shows the distribution of concave clusters over the same set of experiments.

4. It is unclear whether the cluster quality assessment model is pretrained. Include more detail on the training process of the reclustering decision model.

Reply: Thank you for your suggestions. Both the cluster quality assessment model and the reclustering decision model are pretrained and we describe them in detail in lines 424-437 of method. To construct the pretraining dataset, we extracted 3,000 bins from 10 simulated samples, maintaining a 1:2 ratio of high- to medium-quality bins to low-quality bins. These bins were derived from the combined results of MetaBAT2, SemiBin, and SemiBin2. For the cluster quality assessment model, the F1 score was calculated based on completeness and contamination and used as the true label during training. For the reclustering decision model, high- and medium-quality bins were labelled positive examples (not requiring reclustering), whereas low-quality bins were labelled negative examples (requiring second-stage clustering). The data for each model were split into five equal parts. One part was used as the testing set, while the remaining four parts formed the training set, ensuring that there was no overlap between the training and testing datasets. This process was repeated five times to ensure robust evaluation. The revised description about model pretraining can be found in the screenshot below.

Model pretraining

Both the cluster quality assessment models and the reclustering decision models were pretrained. The cluster quality assessment model uses the mean square error as the loss function, and the reclustering decision model uses a sigmoid function as the loss function. We extracted 3000 bins as the pretraining dataset from 10 simulated samples at a ratio of 1:2 between high- and medium-quality bins and low-quality bins. These bins in equal proportions were obtained from MetaBAT2[21], SemiBin[8], and SemiBin2[10], and the F1 score was calculated via the completeness and contamination obtained via CheckM2 (version 1.0.2)[31]. The F1 score is used as the true label to train the cluster quality assessment model. We recorded high- and medium-quality bins as positive examples, which did not need to be reclustered, and low-quality bins as negative examples. These labelled data were used to train the reclustering decision models. For each training model, the data were divided into five equal parts. One part was selected as the testing set, while the remaining four parts were used as the training set, with no overlap between the training sets and testing sets. This process was repeated five times. The highest accuracy reclustering decision model, and the minimum mean square error cluster quality assessment model were applied to LorBin.

5. Also, justification of the architecture would help: Why the crossnet/deepnet architecture. Please explain why the clustering algorithms are needed in the architecture and justify how DBSCAN and BIRCH complement each other in the various steps. What led to the use of both the clustering algorithms and the autoencoders — show why you chose the VAE and DBSCAN and Birch to integrate into LorBin. More rationale is needed for utilizing a deep-learning model for quality assessment could strengthen the theoretical base for this paper, as some researchers may be hesitant due to the relative lack of interpretability of NN solutions. Write a more fleshed out introduction/background section that describes some of the key aspects of approaches that were tested against in benchmarking. This would provide the reader with further insight into your inspirations and how your design decisions are different from the other algorithms.

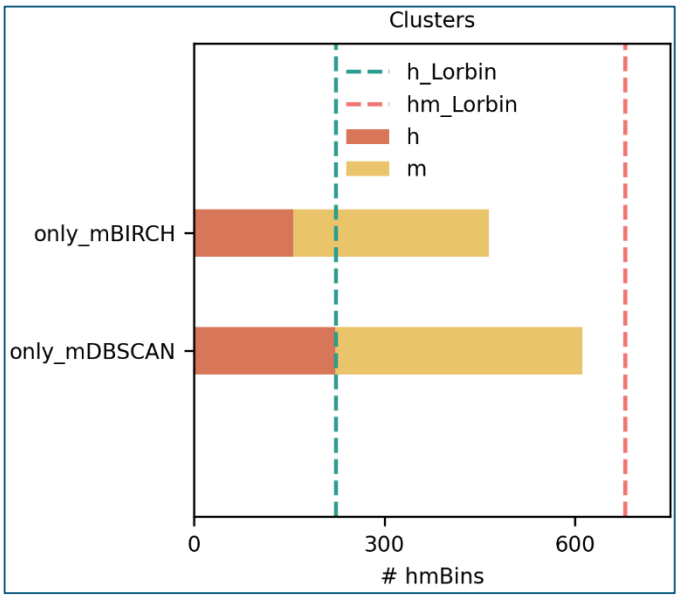
Reply: Thank you for your suggestions. Encoders and clustering algorithms are critical components of the binning process. To optimize the performance, we systematically evaluated three encoder algorithms (Supplementary Figure 2) and 12 clustering algorithms (Supplementary Figure 1) for feature extraction and clustering through ablation experiments. Based on these results, we selected the optimal encoder, VAE (Supplementary Table S1, Supplementary Table S2, Supplementary Figure 2), along with DBSCAN and BIRCH as the clustering algorithms (Supplementary Figure 1).

The clustering process is structured in two stages, with DBSCAN and BIRCH complementing each other. In the first stage, multi-DBSCAN identifies high-quality clusters, whereas in the second stage, multi-BIRCH clusters contigs that were not successfully grouped in the first stage. Compared with a single clustering algorithm, this two-stage clustering approach yields more high- and medium-quality bins (Respond Figure 3).

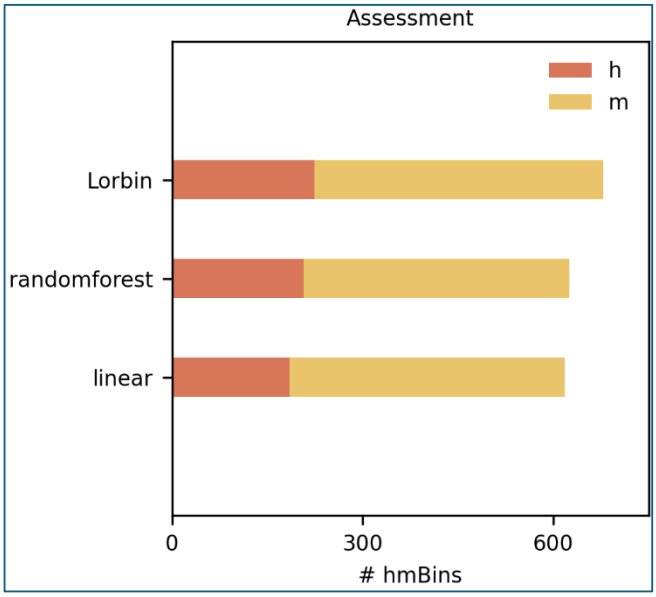
The cluster quality assessment model and reclustering decision model, built with a crossnet/deepnet architecture, serve as the bridge between the two clustering stages. This architecture efficiently captures

effective feature interactions, learns highly nonlinear relationships, and eliminates the need for manual feature engineering[7,8]. The cluster quality assessment model in LorBin outperforms simpler approaches such as linear regression and random forest (Respond Figure 4). Additionally, the emergence of other tools[9] that leverage deep learning models for quality assessment highlights the growing importance of advanced methodologies in this field.

To enhance interpretability, we applied SHAP analysis[10], which has been used[11–14] to the reclustering decision model and the quality assessment model (Supplementary Figure 6). The analysis revealed that the key indicators—predicted completeness and contamination align with established criteria for evaluating high- and medium-quality bins[15–17]. Based on your suggestion, we have also expanded the introduction and background sections to provide a more comprehensive discussion of the approaches benchmarked in our study.



Response Figure 3. The number of high- and medium-quality bins (hmbins) obtained by LorBin with two separate clustering algorithms. The original two-stage multiscale adaptive DBSCAN and BIRCH clustering methods were replaced with single multiscale adaptive DBSCAN and BIRCH algorithms, respectively. The orange and yellow bars indicate the number of high-quality medium-quality bins, and the green and orange dotted lines represent the number of high-quality bins and the total number of hmbins obtained by LorBin, respectively.



Response Figure 4. The number of high- and medium-quality bins used for different cluster quality assessment models. The orange and yellow bars represent the numbers of high-quality and medium-quality bins, respectively.

In Result section

evaluation-decision models (Figure 1bc). Compared with single clustering binners, the two-stage clustering approach yielded more high- and medium-quality bins (hmBins) (Supplementary Figure 3) due the complementarity effect between two clustering stages (Supplementary Figure 4). In addition, the iterative process using the quality assessment model generates more confident preliminary bins by analysing the boundaries, overlaps, and shapes of clusters. The reclustering decision model enabled the complementarity of high-quality bins by determining the free contigs and those in low-completeness bins to be reclustered (Supplementary Figure 5). This assessment-decision model was found to be more effective than linear regression and random forest models (Supplementary Figure 5bc). To explore the important features for

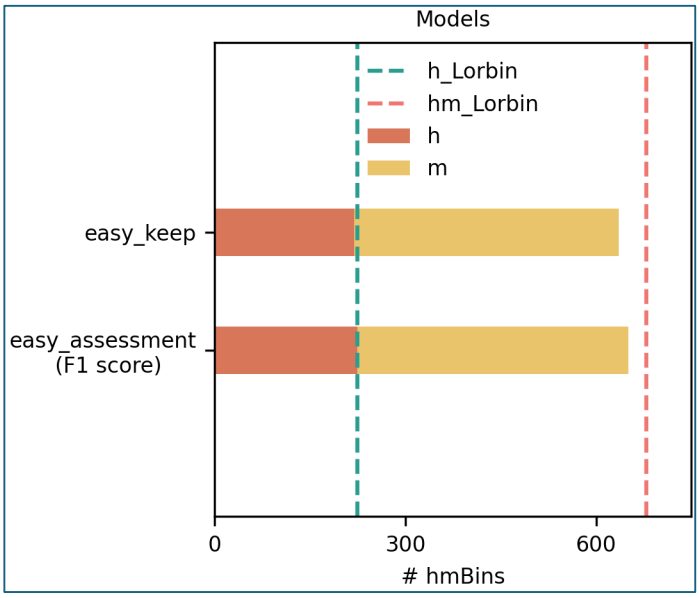
In Introduction section

Current metagenomic binners struggle to handle long-read data. The continuity and rich information of long reads enable the assembly of low-abundance genomes and improve the assembly to be more continuous with fewer stitching errors. These factors make existing short-read binners unsuitable for long-read binning due to the differences in the properties of assemblies between short- and long-read sequencing. The most advanced binning tools improve the quality of binning by optimizing the critical steps of feature extraction and contig clustering. For instance, VAMB first employs a deep variational autoencoder to encode the k-mer and abundance features[6], and was improved by AAMB via an adversarial autoencoder[7]. SemiBin[8] uses a siamese neural network with a pretrained model, whereas COMEBin[9] applies data augmentation and contrastive multi-view representation learning to embed contigs. SemiBin2[10] utilized a self-supervised contrastive learning to extract feature embeddings from short-read contigs and was extended to long-read data by incorporating a DBSCAN clustering algorithm. These methods span from encoders[6,8,10,11] to algorithms of clustering[9,10,12,13] or their integrated models[14–16].

6. Explain how the system performs with some of the different parts being altered/removed as to understand which components of the system are most integral. For example, if the reclustering decision model were to be

replaced with a simple approach weighing the completeness and contamination values against some threshold, how would this impact performance?

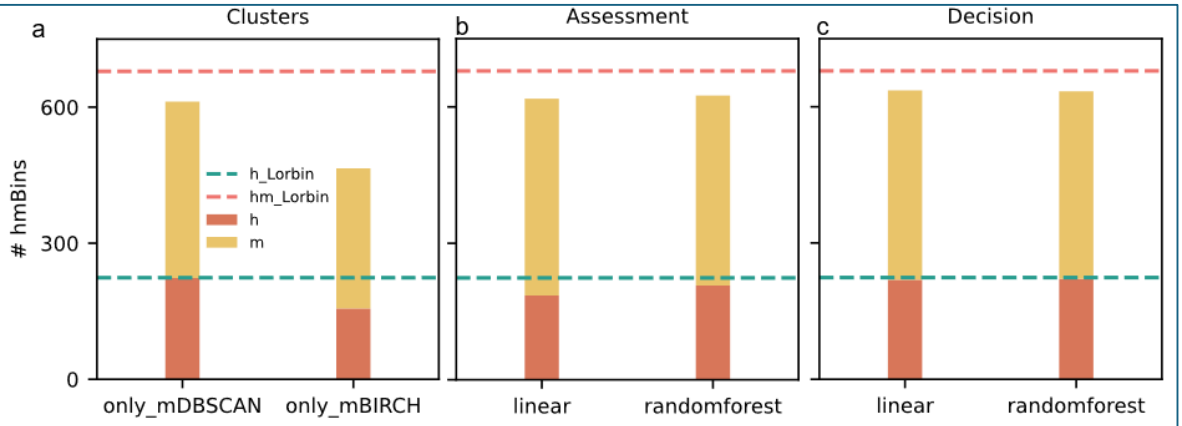
Reply: Thank you for the suggestions. To clarify how DBSCAN and BIRCH complement each other, we replaced the two-stage clustering algorithm with either DBSCAN or BIRCH (Respond Figure 3). To demonstrate how the cluster quality assessment model performs, we removed the model and relied solely on the F1 score, calculated from 107 single-copy markers, to assess cluster quality. This simpler approach combines completeness and contamination. For the reclustering decision model, we removed it and used completeness and contamination values from the 107 single-copy markers, comparing them to a threshold (completeness > 0.5, contamination < 0.1) to determine whether a cluster should be retained. In tests on 104 real metagenomic datasets, the raw LorBin method achieved the highest number of high- and medium-quality bins (Respond Figure 5). We summarized the replacement of clustering algorithms, cluster quality assessment models and reclustering decision models and then incorporated the results into the main text.



Response Figure 5. The number of high- and medium-quality bins (hmbins) generated by LorBin with modified models. The cluster quality assessment model was removed and replaced with the F1 score, whereas the reclustering model was removed and substituted with a simple retention strategy based on completeness and contamination. Orange and bar represent the number of high-quality and medium-quality bins, respectively. The green and orange dotted lines show the number of high-quality bins and the total number of hmbins achieved by LorBin, respectively.

In Result section

evaluation-decision models (Figure 1bc). Compared with single clustering binners, the two-stage clustering approach yielded more high- and medium-quality bins (hmBins) (Supplementary Figure 3) due the complementarity effect between two clustering stages (Supplementary Figure 4). In addition, the iterative process using the quality assessment model generates more confident preliminary bins by analysing the boundaries, overlaps, and shapes of clusters. The reclustering decision model enabled the complementarity of high-quality bins by determining the free contigs and those in low-completeness bins to be reclustered (Supplementary Figure 5). This assessment-decision model was found to be more effective than linear regression and random forest models (Supplementary Figure 5bc). To explore the important features for



Supplementary Figure 3. The number of high- and medium-quality bins (hmbins) obtained by LorBin, compared with some other parts, was altered. (a) The number of high- and medium-quality bins (hmbins) obtained by LorBin with two separate clustering algorithms. The original two-stage multiscale adaptive DBSCAN and BIRCH clustering methods were replaced with single multiscale adaptive DBSCAN and BIRCH algorithms, respectively. (b) The number of high- and medium-quality bins (hmbins) generated by LorBin with modified cluster quality assessment models. The cluster quality assessment model was removed and replaced with linear regression or random forest regression. (c) The number of high- and medium-quality bins using different reclustering decision models. The orange bar represents the number of high-quality bins; the yellow bar represents the medium-quality bins; the green dotted lines represent the number of high-quality bins; and the orange dotted lines represent the total number of high- and medium-quality bins obtained by LorBin.

Minor Revisions:

1. In line 337 and equation 7, it looks like output of the deepnet module may have inconsistent labeling - sometimes it seems to be referred to as C_j, and other times as D_j.

Reply: Thank you for your careful reading. We apologize for the mistake and have corrected it.

$$D_{j+1} = f\left(W_{d,j}D_j + b_j\right) \quad (6)$$

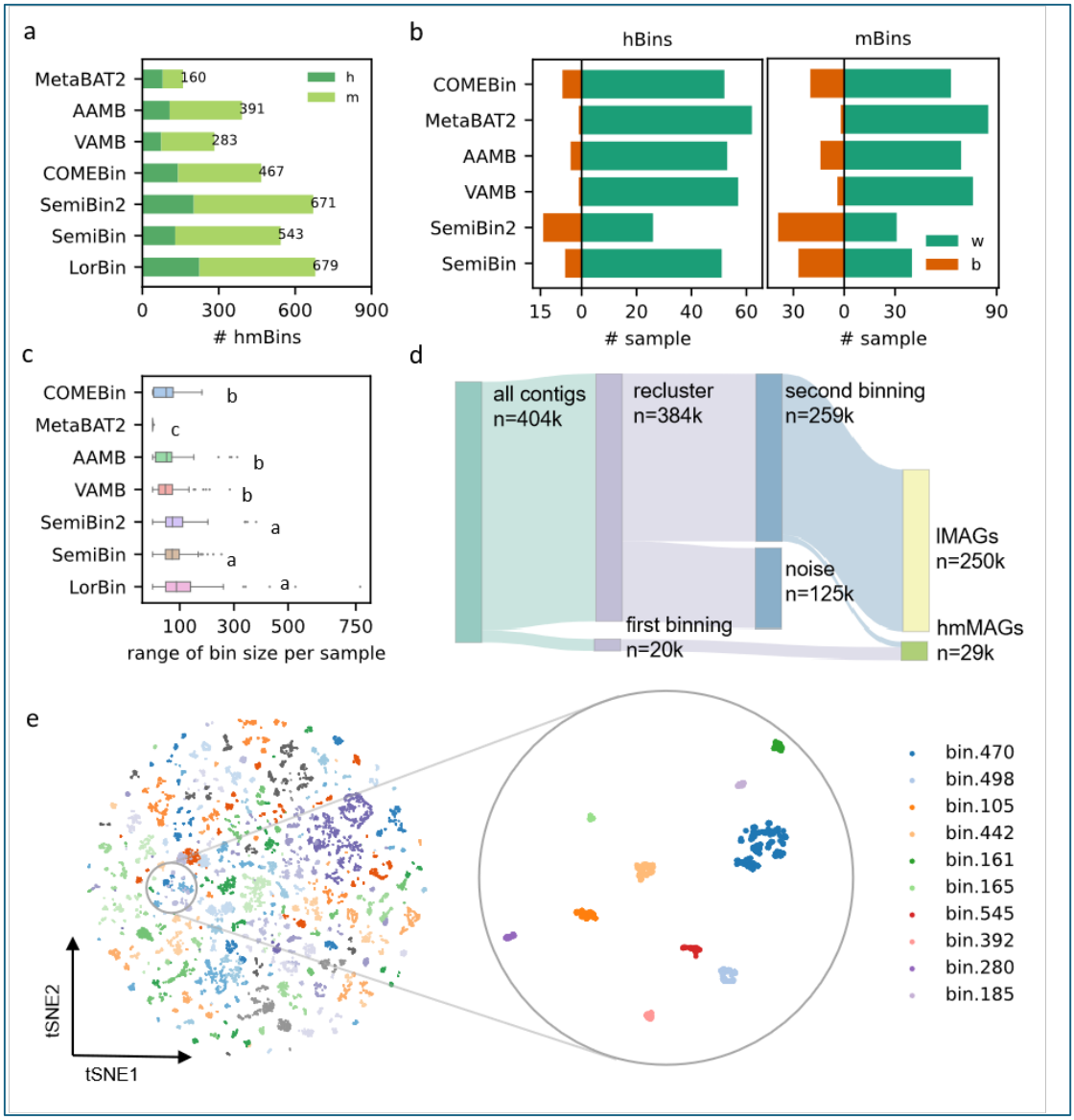
where $j = 1,2, \ldots, J\text{-}1$, D_j and D_{j+1} are the outputs of the jth and $(j+1)$th DeepNet layers, respectively, and $D_1 = X_m$. f is the ReLU activation function, which adds nonlinear components to the model and strengthens the learning capabilities. b_j and $W_{d,j}$ are the bias and weight of the jth layer, respectively. D_J is the final output of DeepNet.

The outputs of CrossNet and DeepNet are fused as D'_c and then sent to the fully connected layer to obtain D_c as the input of the multihead attention mechanism. The input vector is linearly transformed to obtain a matrix of $query(Q)$, $key(K)$, and $value(V)$.

$$D'_c = Concat\left(C_L, D_J\right) \quad (7)$$

2. “first” misspelled/typo’d in fig. 3

Reply: Thank you for your correction. We have redrawn Figure 3.



3. (270-271) "The output of the μ layer is the mean of the latent, while the output is the variance of the latent." I believe they meant to put "the output of σ is the variance"

Reply: Thank you for your correction; we apologize for this mistake.

VAE embedding

A variational autoencoder is constructed to embed contigs. The input layer dimension of VAE[6] is *S*+*T*, and the input matrix is $X \in \mathbb{R}^{N\times(S+T)}$. The hidden layer of the encoder consists of two fully connected layers, each using batch normalization and dropout ($p = 0.2$). The output of the last layer is passed to two different, fully connected layers of length N_z, termed the μ and σ layers. The output of the μ layer is the mean of the latent, whereas the output of the σ layer is the variance of the latent. Then, the latent layer, z, is of length N_z and is

4. while well known to the assembly field, justify/put a reference for using tetranucleotide features

Reply: Thank you for your suggestions. We have added references there.

The *k*-mer frequency is denoted as $F \in \mathbb{R}^{N\times T}$. The *T*-dimensional composition vector X^c is generated from the *k*-mer frequency. LorBin sets $k = 4$[6–8,10,11] and $T = 136$ (treating *k*-mers and their reverse complements as equivalents[6–10,29]).

$$x_i^c = \frac{F_i + 10^{-5}}{\sum_{j=1}^{T} F_{i,j} + 10^{-5}} \quad (1)$$

6. Nissen, J. N. *et al.* Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotechnol* **39**, 555–560 (2021).

7. Líndez, P. P. *et al.* Adversarial and variational autoencoders improve metagenomic binning. *Commun Biol* **6**, 1073 (2023).

8. 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. *Nat Commun* **13**, 2326 (2022).

9. Wang, Z. *et al.* Effective binning of metagenomic contigs using contrastive multi-view representation learning. *Nat Commun* **15**, 585 (2024).

10. Pan, S., Zhao, X.-M. & Coelho, L. P. SemiBin2: self-supervised contrastive learning leads to better MAGs for short- and long-read sequencing. *Bioinformatics* **39**, i21–i29 (2023).

11. 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).

5. Fig. 3 caption says "The analysis of cluster shapes is shown in Supplementary Figure 8." but it's really in Supplementary Figure 12.

Reply: Thank you for your correction. We have added new supplementary figures, and corrected the Figure numbers.

reconstruction of partial hmBins, which constituted the result. (e) Visualization of the embedding of hmBins reconstructed by LorBin. The analysis of cluster density is shown in Supplementary Figure 13. The analysis of cluster shapes is shown in Supplementary Figure 14.

Reference:

1. Bhargav, S. A Review of Clustering Methods forming Non-Convex clusters with, Missing and Noisy Data. *International Journal of Computer Sciences and Engineering* **4**, (2016).

2. Francetič, M., Nagode, M. & Nastav, B. Hierarchical clustering with concave data sets. *Adv Meth Stat* **2**, (2005).

3. Li, P. & Niggemann, O. Improving clustering based anomaly detection with concave hull: An application in fault diagnosis of wind turbines. in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* 463–466 (2016). doi:10.1109/INDIN.2016.7819205.

4. Ondov, B. D. *et al.* Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol* **17**, 132 (2016).

5. Wan, X. *et al.* Integrating spatial and single-cell transcriptomics data using deep generative models with SpatialScope. *Nat Commun* **14**, 7848 (2023).

6. Chen, V. *et al.* Applying interpretable machine learning in computational biology—pitfalls, recommendations and opportunities for new developments. *Nat Methods* **21**, 1454–1461 (2024).

7. Wang, R., Fu, B., Fu, G. & Wang, M. Deep & Cross Network for Ad Click Predictions. Preprint at http://arxiv.org/abs/1708.05123 (2017).

8. Guo, H., Tang, R., Ye, Y., Li, Z. & He, X. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. Preprint at http://arxiv.org/abs/1703.04247 (2017).
9. Chklovski, A., Parks, D. H., Woodcroft, B. J. & Tyson, G. W. CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. *Nat Methods* **20**, 1203–1212 (2023).
10. Lundberg, S. M. & Lee, S.-I. A Unified Approach to Interpreting Model Predictions. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 4765–4774 (Curran Associates, Inc., 2017).
11. Erfanian, N. *et al.* Deep learning applications in single-cell genomics and transcriptomics data analysis. *Biomedicine & Pharmacotherapy* **165**, 115077 (2023).
12. Brendel, M. *et al.* Application of Deep Learning on Single-Cell RNA Sequencing Data Analysis: A Review. *Genomics, Proteomics & Bioinformatics* **20**, 814–835 (2022).
13. Lin, X., Tian, T., Wei, Z. & Hakonarson, H. Clustering of single-cell multi-omics data with a multimodal deep learning method. *Nat Commun* **13**, 7705 (2022).
14. Johnsen, P. V., Riemer-Sørensen, S., DeWan, A. T., Cahill, M. E. & Langaas, M. A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values. *BMC Bioinformatics* **22**, 230 (2021).
15. Pan, S., Zhao, X.-M. & Coelho, L. P. SemiBin2: self-supervised contrastive learning leads to better MAGs for short- and long-read sequencing. *Bioinformatics* **39**, i21–i29 (2023).
16. 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. *Nat Commun* **13**, 2326 (2022).
17. 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 Res.* **25**, 1043–1055 (2015).
18. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9**, (2008).
19. Arora, S., Hu, W. & Kothari, P. K. An analysis of the t-sne algorithm for data visualization. in *Conference on learning theory* 1455–1462 (PMLR, 2018).
20. Cai, T. T. & Ma, R. Theoretical foundations of t-sne for visualizing high-dimensional clustered data. *Journal of Machine Learning Research* **23**, 1–54 (2022).
21. Choi, Y., Li, R. & Quon, G. siVAE: interpretable deep generative models for single-cell transcriptomes. *Genome Biol* **24**, 29 (2023).
22. Grønbech, C. H. *et al.* scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).
23. Pan, W., Long, F. & Pan, J. ScInfoVAE: interpretable dimensional reduction of single cell transcription data with variational autoencoders and extended mutual information regularization. *BioData Mining* **16**, 17 (2023).
24. He, Y., Tan, H., Luo, W., Feng, S. & Fan, J. MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data. *Frontiers of Computer Science* **8**, 83–99 (2014).
25. Xiao, W. & Hu, J. A survey of parallel clustering algorithms based on spark. *Scientific Programming* **2020**, 8884926 (2020).
26. Monath, N. *et al.* Scalable Hierarchical Agglomerative Clustering. in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* 1245–1255 (ACM, Virtual Event Singapore, 2021). doi:10.1145/3447548.3467404.

Comments from Reviewer #3:

Wei Xue et al. introduce a binner binner, LorBin. It achieves state of the art performance by combining known techniques from existing, modern binners:
* Common binning features (k-kmer composition and abundance) integrated with a VAE
* Multi-step reclustering informed by single-copy genes
* Being optimized for accurate, long reads such as HiFi reads
With a novel adaptive and hybrid clustering scheme that handles deferentially abundant genomes better than previous binners.
Overall, I find the manuscript well written, and the claims of accuracy are convincingly supported by extensive testing and optimization, as well as excellent ablation studies.
The work shown in this manuscript is a valuable contribution to the field and I would be excited to be able to run LorBin myself.
While I would like to see this work published, I do find the manuscript is lacking in some areas and have some issues with the manuscript as-is:

Reply: We appreciate your thoughtful and constructive feedback on the manuscript. Your suggestions help us improve the clarity of our work. Below, we provide a detailed response to your comments, incorporating your suggestions and clarifying the points raised. We have also revised the manuscript accordingly.

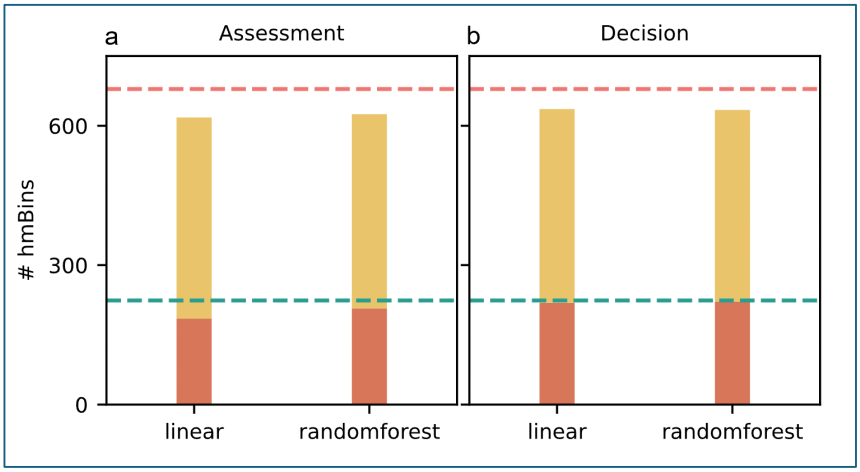
Major points:You claim that LorBin is reference-free, but it cannot be reference-free if it uses a set of reference single-copy genes (SCGs). From skimming the source code, these SCGs are found with a HMM search using SCG signatures taken from a database. You could argue that SCGs generalize to large swaths of the tree of life than e.g. sequence alignment, and therefore LorBin is not dependent on the precise species/genus represented by a bin being present in databases, but it's wrong to call it reference-free. This matters, because as demonstrated in e.g. the CheckM2 paper, the completion of genomes from clades are systematically under- or overestimated using fixed SCGs (because they may lack some of them, or may contain multiple copies).

Reply: Thank you for pointing out this distinction. We agree that the term "reference-free" was not accurate enough in the context of LorBin. To make it more accurate for the "reference-free", we would like to describe that LorBin does not rely on labelled data or supervised techniques during the training phase of the variational autoencoder (VAE). However, you are right that SCGs were used to guide clustering at the stage of cluster assessment. To clarify, we have revised the manuscript to avoid the term "reference-free" and instead describe LorBin as a method that minimizes reliance on reference genomes during the training phase while using SCGs for clustering refinement.

Abstract

Long-read sequencing has transformed metagenomics and improved the quality of metagenome-assembled genomes (MAGs). However, current binning methods struggle with identifying unknown species and managing imbalanced species distributions. Here, we present LorBin, an unsupervised binner specially designed to reconstruct MAGs in natural microbiomes. LorBin deploys a two-stage multiscale adaptive DBSCAN and BIRCH clustering with evaluation decision models using single-copy genes to maximize MAG recovery. LorBin outperforms six competing binners in both simulated and real microbiomes, including oral, gut, and marine samples. LorBin generated 15–189% more high- quality MAGs with high serendipity and identified 2.4–17 times more novel taxa than state-of-the-art binning methods. Together, LorBin is a promising long-read metagenomic binner for accessing species-rich samples containing unknown taxa and is efficient at retrieving more complete genomes from imbalanced natural microbiomes.

Regarding the use of SCGs, we acknowledge the limitations highlighted in the CheckM2 paper, where fixed SCGs can lead to systematic biases in genome completion estimates. To mitigate this, LorBin employs an adaptive assessment-decision model that dynamically corrects for biases introduced by SCGs. Our experiments demonstrate that this approach yields more high-quality bins than other methods that rely solely on fixed SCGs (SemiBin2). We have included additional results in the supplementary materials to support this claim (see Response Figure 1).

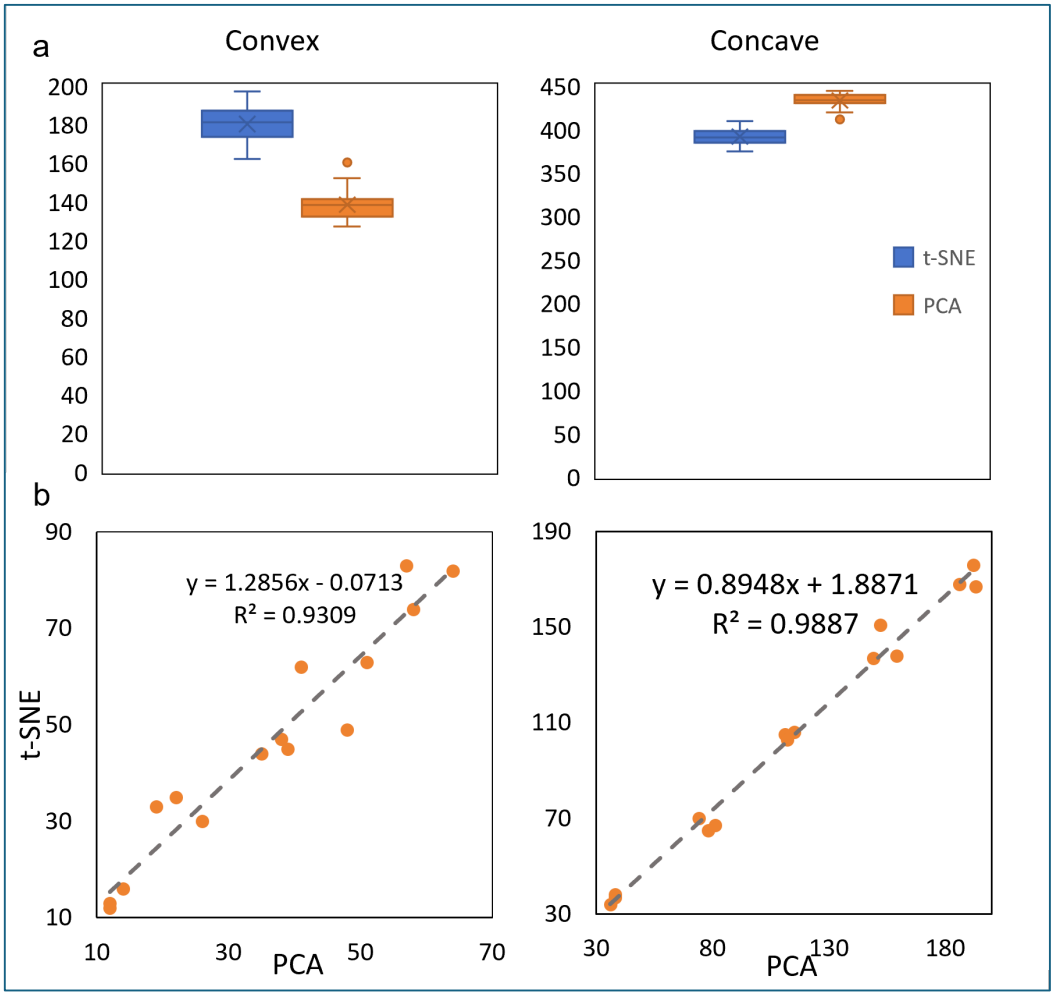


Response Figure 1. LorBin produced more high- and medium-quality bins (hmbins) than some other assessment-decision methods easily using SCGs. (a) The number of high- and medium-quality bins (hmbins) generated by different cluster quality assessment models such as linear regression or random forest regression. (b) The number of high- and medium-quality bins using different reclustering decision models such as linear regression or random forest regression. The orange bar represents the number of high-quality bins; the yellow bar represents the medium-quality bins; the green dotted lines represent the number of high-quality bins; and the orange dotted lines represent the total number of high- and medium-quality bins obtained by LorBin.

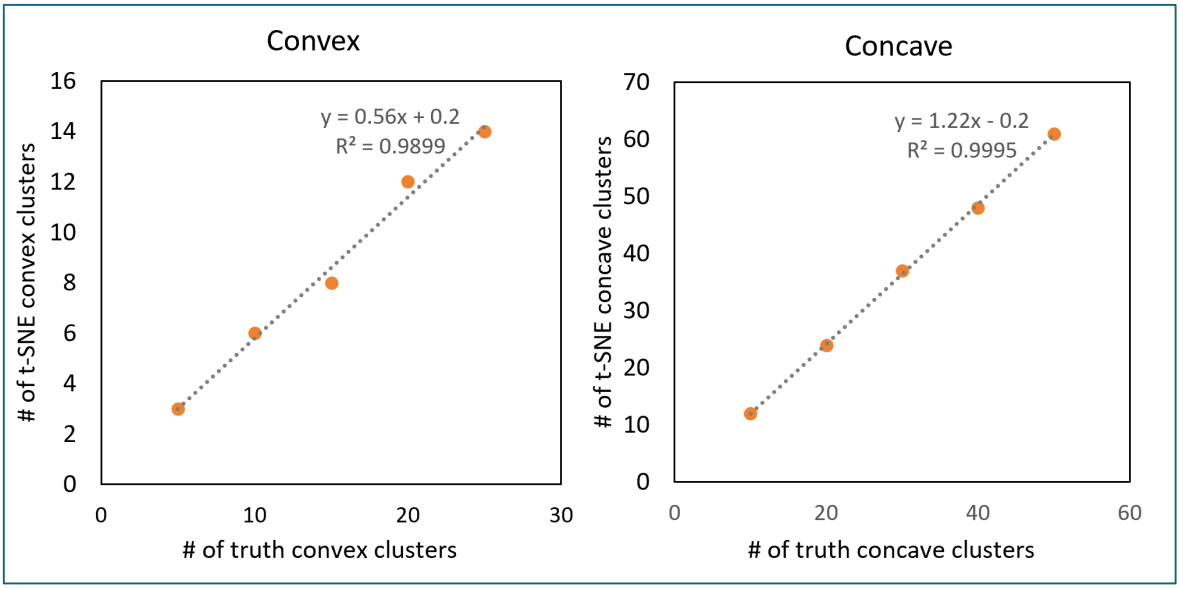
evaluation-decision models (Figure 1bc). Compared with single clustering binners, the two-stage clustering approach yielded more high- and medium-quality bins (hmBins) (Supplementary Figure 3) due the complementarity effect between two clustering stages (Supplementary Figure 4). In addition, the iterative process using the quality assessment model generates more confident preliminary bins by analysing the boundaries, overlaps, and shapes of clusters. The reclustering decision model enabled the complementarity of high-quality bins by determining the free contigs and those in low-completeness bins to be reclustered (Supplementary Figure 5). This assessment-decision model was found to be more effective than linear regression and random forest models (Supplementary Figure 5bc). To explore the important features for

In my opinion, applying t-SNE on a VAE embedding makes little sense and is bound to give misleading results. The VAE prior, enforced by the regularization loss ("KLoss", Eq 3 in the manuscript), is that the data is distributed as a multivariate unit Gaussian around the origin. Therefore, to reduce KLoss the VAE will attempt to embed the contigs in an N-dimensional ball around the origin, to the extent allowed by the RLoss. This implies that the VAE will attempt to spread the data into as many dimensions as available. And this in turn means that one cannot assume the data lies in some lower-dimensional manifold that can be approximated by t-SNE. On the contrary, one can assume the data is intrinsically high-dimensional and can't be reduced to 2 dimensions! For this reason, I don't think any of the analysis on the t-SNE projection (lines 540-555) are meaningful, and they may even be misleading. I suggest the authors either remove this analysis, or else support its validity e.g. by doing a PCA of the embedding and showing that the top two components explain the majority of the total variance, and that the data therefore lies on a two-dimensional plane.

Reply: We appreciate your insightful critique regarding the use of t-SNE on VAE embeddings. While we agree that the VAE latent space is inherently high-dimensional, we argue that t-SNE can still provide meaningful visualizations of reduced lower-dimensional structures to capture the original high-dimensional data in current practical applications[18–20]. We also noticed the impact of KL divergence on embedding, and we set KL divergence a weight at 0.5 in the loss function. Under appropriate weight settings, using t-SNE to visualize VAE embedding to show data features in high-dimensional space is a common technology[21–23]. To improve the clarity, we have performed additional experiments via PCA to validate the stability of the latent space structure. In 104 human gut samples, the number of convex and concave clusters obtained by PCA was slightly different from t-SNE (Response Figure 2a). We randomly selected numbers, and use PCA and t-SNE to visualize them. The results indicate that the two variables are highly positively correlated between PCA and t-SNE (Response Figure 2b). In addition, we use Guassion distribution and noise points to generate a batch of original high-dimensional clusters with ground truth of convexity and concavity at rate of 1:2, and then used t-SNE to visualize them to provide a benchmark for judgement of their convexity and concavity. It is undeniable that some information is lost during dimensionality reduction, and that t-SNE tends to underestimate the number of convex clusters. However, as shown in Response Figure 3, there are positive correlations between the number of convex clusters identified by t-SNE and the actual number of convex clusters, as well as between the number of concave clusters identified by t-SNE and the true number of concave clusters. In our manuscript, the t-SNE analysis is independent of LorBin and is used solely to illustrate the importance of considering data imbalance and the varying shapes of clusters, which have been largely overlooked by existing metagenomic binning tools.



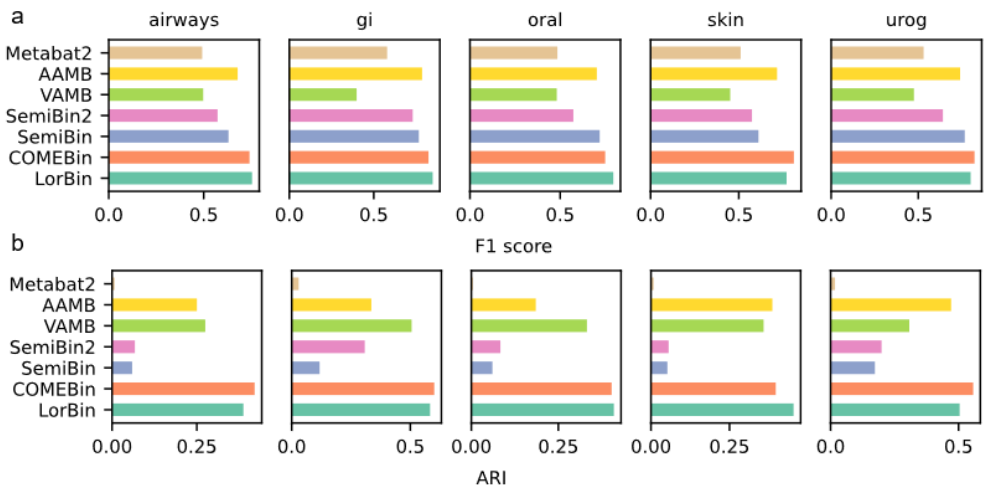
Response Figure 2. Convexity and concavity shapes of t-SNE and PCA. (a) In 104 human intestine samples, twenty randomly generated t-SNE parameter sets were used to repeat cluster shape detection. The left box plot shows the distribution of the number of convex clusters across the 20 experiments, whereas the other box plot shows the distribution of concave clusters over the same set of experiments. (b) Correlation between PCA and t-SNE for concavity. Different numbers of bins from 104 human intestine samples were selected and visualized via PCA and t-SNE. The left plot shows the correlation between PCA and the t-sne of the number of convex clusters. The other is the correlation between PCA and the t-sne of the number of concave clusters.



Response Figure 3. Correlations between the shapes of t-SNE and ground truth shapes in high-dimensional space. (a) Correlations between the convex shapes of t-SNE and the ground truth convex shapes in high-dimensional space. (b) Correlations between concave shapes of t-SNE and ground truth concave shapes in high-dimensional space.

The measurement of performance on the simulated dataset is quite nonstandard. Typically, other papers presenting new binners count the number of high quality bins, or the mean bin recall or precision or bin F1 distribution. There are also tools such as BinBencher or AMBER to compute various bin statistics which may be useful (disclaimer: I authored BinBencher, so it would not be appropriate for me to suggest it specifically). While I have no reason to doubt that LorBin would also perform well as measured using these more classical metrics, I think it would strengthen the paper to also report them.

Reply: Thank you for this suggestion. We agree that including more standard performance metrics would strengthen the paper. Now, the adjusted rand index (ARI) and F1 score are included as metrics to evaluate the performance of LorBin. These metrics provide a more comprehensive assessment of binning accuracy and are consistent with those used in other studies. The results are included in Supplementary Figure 8 and discussed in the main text.



Supplementary Figure 8. The F1 score and the adjusted Rand index (ARI) of the results from different binners on five datasets. (a) F1 scores of the results from different binners on five datasets. The F1 score is calculated by completeness and contamination. (b) Adjusted Rand index (ARI) of the results from different binners on five datasets.

Supplementary Figure 7b). We also compared the adjusted Rand index (ARI) and F1 score of the binning performance across different binners. LorBin achieved the highest F1 score in the airway, gastrointestinal, and oral environments and the highest ARI in the oral and skin environments (Supplementary Figure 8). Overall, LorBin outperforms the available cutting-edge binners in recovering complete genomes from synthetic long-read metagenomes.

Computational efficiency is an important metric of new methods, since slow or memory-hungry methods may make some methods prohibitive in practice. It would be better if the authors presented resource use (e.g. RAM and CPU/GPU time) for LorBin along with at least a few other methods, e.g. the faster MetaBAT2 or VAMB, and the slower COMEBIN. If possible, it would also be informative to show how runtimes scale with the number of contigs. Can I run LorBin on a dataset with a million contigs?

Reply: We appreciate your suggestion. We have conducted additional experiments to compare the computational efficiency of LorBin with that of other SOTA binners, including MetaBAT2, VAMB, and COMEBin. Only GPUs or CPUs were used for this experiment on four human intestinal datasets (SRR13128012, SRR13128013, SRR13128014, and ERR9765783). The results show that LorBin is efficient, particularly when a GPU is used, and scales well with the number of contigs. We have included these results in Supplementary Tables S1-S5 and added a discussion on the scalability of LorBin. The time and space complexity serve as fundamental metrics for evaluating algorithmic efficiency. We conducted a comprehensive complexity analysis of LorBin, demonstrating that its computational and memory requirements scale optimally with input size. LorBin's time complexity, including encoder training and clustering, is $O(T \cdot n \sum_{l=1}^{L} d_{l-1} d_l + n \cdot (M_a + M_r) + nlogn)$, where n is the number of contigs, L is the number of VAE layers, d_l is the output dimensional number of the i-th layer, and T is the number of epochs. LorBin's space complexity, including the VAE model, VAE embedding and a k-distance matrix, is $O\left(\sum_{l=1}^{L} d_{l-1} d_l + n \times d_{latent} + M_a + M_r + n\right)$, where d_{latent} is the number of dimensions of VAE

embedding. The cost of model training is often greater during the whole process, and the time and memory consumed by LorBin are linearly correlated with the number of contigs through controlled scalability experiments (Supplementary Figure 17). For large datasets of millions of contigs, we recommend using a computer with at least 45GB of RAM and GPU or deploying the distributed-computing scheme to improve efficiency. The use of GPUs accelerates encoder training. Through marginal analysis, the efficiency of LorBin using a machine with a 32-core CPU, 64GB of RAM and 0.3*NVIDIA A800 GPU (with 24GB of VRAM) is 76 times faster than that of using only CPUs. Distributed computing is mainly used to accelerate clustering algorithms, and the theoretical acceleration ratio is close to linear (the accelerated clustering algorithm's time complexity is $O(n\log n/p)$, where p represents the number of parameters for distributed computing)[24–26]. We have added a discussion on this topic in the manuscript and provided guidelines for optimizing LorBin's performance on large datasets. Thank you for seeing the screenshots from additonal file, method section, discussion section and supplementary information below and checking them.

In Additional file

CPU/s	SRR13128912	SRR13128913	SRR13128914	ERR9765783
Metabat2	3.6	3.1	3.1	6.6
VAMB	175.1	124.0	163.2	142.7
AAMB	187.6	168.1	172.6	174.6
SemiBin	2583.7	1945.3	1577.4	2080.0
SemiBin2	2333.0	1912.7	1632.6	2128.1
COMBin	4442.4	3297.5	4404.6	3725.7
LorBin	232.5	202.6	170.2	550.9

Supplementary Table S2

GPU/s	SRR13128912	SRR13128913	SRR13128914	ERR9765783
VAMB	45.3	40.5	40.8	40.3
AAMB	131.5	126.9	115.5	115.9
SemiBin	1774.9	1474.2	1206.5	1674.1
SemiBin2	1735.7	1429.4	1194.7	1676.2
COMEBin	532.2	552.0	549.3	1094.5
LorBin	154.6	134.6	115.6	482.6

Supplementary Table S3

GPU/s	SRR13128912	SRR13128913	SRR13128914	ERR9765783
VAMB	45.3	40.5	40.8	40.3
AAMB	131.5	126.9	115.5	115.9
SemiBin	1774.9	1474.2	1206.5	1674.1
SemiBin2	1735.7	1429.4	1194.7	1676.2
COMEBin	532.2	552.0	549.3	1094.5
LorBin	154.6	134.6	115.6	482.6

Supplementary Table S4

	SRR13128012	SRR13128013	SRR13128014	ERR9765783
LorBin	4.35GB	4.32GB	4.32GB	4.41GB
SemiBin	9.10GB	8.45GB	7.75GB	8.24GB
Semibin2	8.87GB	8.42gb	7.80GB	8.22GB
VAMB	4.31GB	4.17GB	4.18GB	4.18GB
AAMB	4.17GB	4.17GB	4.17GB	4.17GB
COMEBin	4.24GB	4.24GB	4.24GB	4.24GB
MetaBAT2	0.09GB	0.09GB	0.09GB	0.17GB

Supplementary Table S5

In Method section

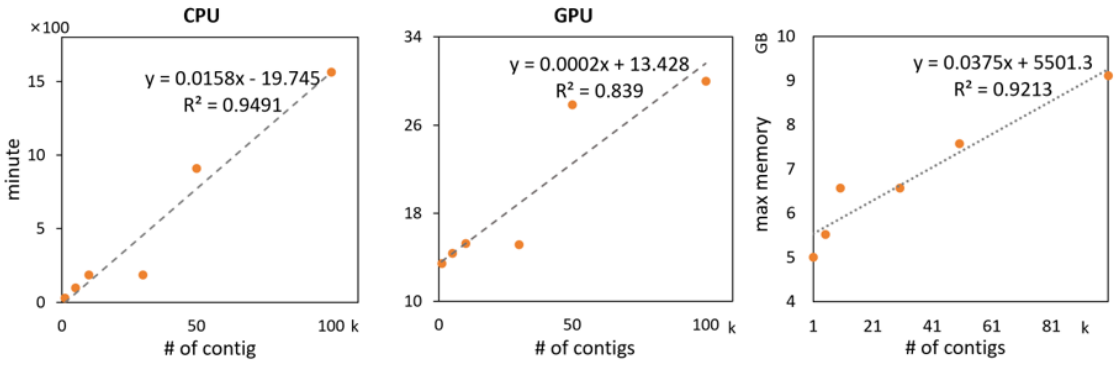
Time and space complexity analysis

We conducted a comprehensive time and space complexity analysis of LorBin. The time complexity of LorBin, including encoder training and clustering, is $O(T \cdot n \sum_{l=1}^{L} v_{l-1} v_l + n \cdot M_a + n \cdot M_r + n \cdot logn)$, where L is the number of VAE layers, v_l is the output dimensional number of the i-th layer, T is the number of epochs, M_a is the parameter number of the cluster assessment model and M_r is the parameter number of re-clustering decision model. The memory complexity of LorBin, including the VAE model, VAE embedding, and a k-distance matrix, is $O(\sum_{l=1}^{L} d_{l-1} d_l + M_a + M_r + n \times d_{latent} + n)$, where d_{latent} indicates the dimension number of VAE embedding.

In Discussion section

a more holistic binning assessment. Although LorBin is optimized for large-scale datasets, metagenomic binning remains computationally demanding. In the current metagenomics era, it is important for bioinformatic tools to process millions of contigs. We found that GPU acceleration can achieve a 76-fold speedup (Supplementary Figure 17). For ultra-large input datasets of millions of long-reads, applying the distributed computing or implementing more efficient encoding strategies that could further increase the scalability. Another important

In Supplementary information



Supplementary Figure 17. Time complexity and memory complexity of LorBin through controlled scalability contigs. (a) LorBin's consumption is near-linear (R^2=0.9491) across inputs spanning 10^3–10^5 contigs when only the CPU is used. (b) LorBin's consumption time is near-linear (R^2=0.9491) when the GPU is used. (c) LorBin's consumption memory is near-liner (R^2=0.9213). LorBin was executed on a machine with a 32-core CPU, 64GB of RAM, and 0.3*NVIDIA A800 GPU (with 24GB of VRAM).

The discussion section does a good job of summarizing the content above it, but there is little perspectives about what the success of LorBin implies for binning. How does the authors attribute the performance of LorBin compared to existing methods which use some of the same techniques as LorBin (e.g. Vamb uses a VAE, SemiBin2 uses a SCG-guided clustering which strikes me as similar to LorBin's, COMEBIN is also optimized for long reads)? What novelties from LorBin could other authors in the future use for the next generation binners? What work is left to be done?I believe this is important for the field, because to my count, more than 20 binners have been published the last decade, each claiming superior performance. If the field is to actually progress, it's important that authors explain how they advanced the status quo and how future binners may advance it still further.

Reply: We thank you for this insightful suggestion. We have expanded the discussion section to provide a more detailed perspective on how LorBin advances the field of metagenomic binning. Specifically, we include more discussion about the novelty of LorBin. To date, no Binners have analysed the distributional imbalance and shape of clusters. To this end, LorBin introduces a two-stage adaptive clustering approach that addresses the challenges of data imbalance and heterogeneous cluster shapes. LorBin expands phylogenetic coverage by recovering more unique and novel taxa. This highlights the power of LorBin in exploring microbial dark matter and accessing poorly described or unknown lineages. We systematically compared different encoders and clustering algorithms. We find that the proper clustering algorithm more significantly improves the binning quality, and LorBin combines efficient VAE and SCG-guided clustering for both efficiency and quality reasons. Compared with a single clustering algorithm, this two-stage clustering approach yields more high- and medium-quality bins (Supplementary Figure 3a). The assessment-decision model is used to connect two-stage clustering, which improves both the read/contig utilization and confidence in bin quality. In addition, LorBin is efficient in terms of runtime. In feature extraction and learning processes, LorBin requires both reference genomes and labelled data, whereas in the clustering process, LorBin deploys adaptive clustering to automatically determine parameters instead of traversing each of them.

We introduce LorBin, a novel algorithm tailored for the efficient binning of long-read metagenomes. LorBin brings several substantial advances: 1) More high-quality bins are recovered from both rare and low-abundant species. They are overlooked by existing methods due to compositional imbalance. LorBin addresses this issue directly and improves community resolution. 2) It expands phylogenetic coverage by recovering more unique and novel taxa. This highlights the power of LorBin in exploring microbial dark matter and accessing poorly described or unknown lineages. 3) It improves the binning quality and reliability. LorBin can identify and correct misclassified bins via assessment and reclustering steps. This improves both the read/contig utilization and confidence in bin quality. 4) It is efficient in runtime. In feature extraction and learning processes, LorBin requires nether reference genomes nor labelled data, whereas in clustering process, LorBin deploys the adaptive clustering to automatically determine parameters instead of traversing each of them. Those make LorBin run faster than most supervised or semi-supervised binners. Furthermore, LorBin outperforms the SOTA binners consistently across real-world datasets originating from diverse environments from the human gut to soil to marine samples. These advances make LorBin a robust and promising binner for metagenomic studies.

For long read binning, while COMEBin is not as good as SemiBin2, the discussion section of the COMEBin paper suggests, it is optimized for long reads.

The current and further directions of metagenomic binning are as follows.

We are happy to see the growing international interest in metagenomic binning, with over 20 available tools supporting microbiome research. This finding highlights the importance of metagenomic binning in advancing our understanding of microbial functions and ecology among biologists and bioinformaticians. However, compared with other domains, such as drug–protein interactions, (meta)genomic assembly, single-cell RNA analysis, and gene mining, binning-related studies still lag behind. We believe that this gap presents an opportunity for collaborative innovation and knowledge exchange to drive the field forward.

In the future, we must consider the interplay between sequencing technology and binning. If long-read sequencing advances to capture ultralong DNA (e.g., full genomes of each species in a microbiome), the need for assembly and binning could diminish. However, overcoming significant challenges, such as extracting unbroken DNA, preparing ultrafine sequencing libraries, and improving the read length, enzyme stability, and speed of sequencing instruments, is necessary. Owing to these limitations, binning methods, particularly tools for long-read sequencing, will remain essential for the foreseeable future.

To advance binning, we must revisit its foundational steps: from reads to contigs to bins. There are several important areas for improvement from our point of view. 1) Long sequence processing: developing efficient methods to handle long-read data and extract meaningful features. 2) Novel encoders for DNA feature extraction: innovating beyond traditional k-mer-based approaches to capture more genomic information. 3) New assemblers: developing assemblers that better handle the complexity and diversity of metagenomic data with uneven species distributions. 4) Efficient clustering algorithms for heteromorphic clusters: Design robust clustering algorithms to address unbalanced data and diverse cluster shapes. 5) Specific binners for exogenous genomes such as plasmids, prophages, and mobile genetic elements (MGEs). These genomes often exhibit distinct GC contents, abundances, and core genome characteristics, making them challenging to cluster. 6) Inter interpretability: AI-based binning has become increasingly complex. To increase the reliability of the AI model, it is crucial to understand how

models make binning decisions. 7) Efficient runtime: for ultra-large datasets of millions of long-reads, distributed computing or more efficient encoding strategies could further increase the scalability.

By focusing on these areas, we can overcome current limitations and develop binning tools that better reflect the realistic nature of microbiomes. We look forward to collaborating with the community to address these challenges and advance the field of metagenomic binning. Thank you for your insightful comments, and we are excited to continue this exchange of ideas. We have elaborated on those in the discussion section.

261 Despite these advancements, several challenges remain to be addressed in future iterations. One key limitation
262 is the reliance on single-copy genes as the primary metric for assessing bin quality during reclustering. While
263 this approach is efficient, integrating additional context-aware metrics could further improve bin accuracy. For
264 example, coverage correlation coefficients and k-mer co-occurrence patterns have been applied to improve
265 clustering[15]. Future developments could explore ensemble models that combine multiple genomic features for
266 a more holistic binning assessment. Although LorBin is optimized for large-scale datasets, metagenomic binning
267 remains computationally demanding. In the current metagenomics era, it is important for bioinformatic tools
268 to process millions of contigs. We found that GPU acceleration can achieve a 76-fold speedup (Supplementary
269 Figure 17). For ultra-large input datasets of millions of long-reads, applying the distributed computing or
270 implementing more efficient encoding strategies that could further increase the scalability. Another important
271 challenge is the binning of mobile genetic elements (MGEs). For instance, plasmids and prophages often exhibit
272 distinct sequence characteristics from the chromosomal DNA[27,28]. The development of binning algorithms for
273 MGEs could improve their recovery and contribute to a better understanding of horizontal gene transfer and
274 antimicrobial resistance (Supplementary Figure 18). Interpretability remains another key focus for future
275 research, particularly for AI-based binners. To increase reliability of AI model, it is crucial to understand how
276 models make binning decisions. Thus, developing methods to analyse the feature importance of k-mer
277 frequencies, sequence compositions, or evolutionary signals could provide deeper insights into binning
278 decisions. This approach might increase the confidence of LorBin predictions and provide strong biological
279 insights beyond binning performance metrics. Additionally, expanding LorBin to metatranscriptomic and
280 functional annotation pipelines could facilitate a deeper exploration of microbial interactions, gene expression
281 and metabolic potential.

Minor points

* On line 118, the abbreviation hmBin should be explained.

Reply: Thank you for your correction. We have made corrections in the manuscript.

122 evaluation-decision models (Figure 1bc). Compared with single clustering binners, the two-stage clustering
123 approach yielded more high- and medium-quality bins (hmBins) (Supplementary Figure 3) due the

* Line 150: It's not clear if this means that LorBin has a coefficient 0.36 (etc) higher than the second-best binner tested?

Or is this the actual silhuette coefficient of LorBin? If so, what is the coefficient of the other binners?

Reply: Thanks for your suggestion. We are sorry for the confusion. The original manuscript refers to the absolute value of LorBin's silhouette coefficient, and we indeed ignored the comparison between

LorBin's silhouette coefficient and the silhouette coefficients of other binners. We have changed the statement here to the value of LorBin being higher than the suboptimal value.

reduction in downsampling percentages on the raw metagenomes (Supplementary Figure 9c). Compared with SOTA binners, LorBin was effective at binning each perturbed sample with the highest silhouette coefficients, which was 0.32 ± 0.11 (airways), 0.19 ± 0.13 (gastrointestinal), 0.38 ± 0.10 (oral), 0.32 ± 0.10 (skin), and 0.19 ± 0.13 (urogenital) times higher than the second-best method (Supplementary Figure 9c). The area under the silhouette coefficient curve of LorBin (avg. 0.20 ± 0.01) was consistently larger than that of the competing binners in five different habitats (Figure 2c). The areas of the competing binners ranged from 0.10 ± 0.01 to 0.27 (Figure 2c). Overall, LorBin outperforms the SOTA binners in addressing imbalanced microbiomes.

* Line 160: Mention that this is according to CheckM2. In some papers, this information could perhaps be mentioned in the methods section only, but here, I think it's crucial to mention how this result was obtained.

Reply: Your suggestion is professional, we have mention it.

We further benchmarked the ability of LorBin to process the real long-read PacBio HiFi metagenomes in 104 human intestine samples (Figure 3). LorBin reconstructed 224 high-quality bins (hBins) and 455 medium-quality bins (mBins) according to CheckM2. On average, it achieved 106.1% ± 30.0% higher number for hBins and 115.6%

* Line 430: "trains" should be "strains"

Reply: Thank you for your correction.

The human intestinal simulation datasets used for encoder ablation experiments include four samples, namely, SRR13128012, SRR13128013, SRR13128014, and ERR9765783. SRR13128012, SRR13128013, and SRR13128014 are PacBio HiFi samples containing 21 strains from PRJNA680590[37]. ERR9765783 is a PacBio HiFi simulation sample that includes 76 strains from PRJEB52977[38].

Reference:

1. Bhargav, S. A Review of Clustering Methods forming Non-Convex clusters with, Missing and Noisy Data. *International Journal of Computer Sciences and Engineering* **4**, (2016).
2. Francetič, M., Nagode, M. & Nastav, B. Hierarchical clustering with concave data sets. *Adv Meth Stat* **2**, (2005).
3. Li, P. & Niggemann, O. Improving clustering based anomaly detection with concave hull: An application in fault diagnosis of wind turbines. in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* 463–466 (2016). doi:10.1109/INDIN.2016.7819205.
4. Ondov, B. D. *et al.* Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol* **17**, 132 (2016).
5. Wan, X. *et al.* Integrating spatial and single-cell transcriptomics data using deep generative models with SpatialScope. *Nat Commun* **14**, 7848 (2023).
6. Chen, V. *et al.* Applying interpretable machine learning in computational biology—pitfalls, recommendations and opportunities for new developments. *Nat Methods* **21**, 1454–1461 (2024).
7. Wang, R., Fu, B., Fu, G. & Wang, M. Deep & Cross Network for Ad Click Predictions. Preprint at http://arxiv.org/abs/1708.05123 (2017).
8. Guo, H., Tang, R., Ye, Y., Li, Z. & He, X. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. Preprint at http://arxiv.org/abs/1703.04247 (2017).
9. Chklovski, A., Parks, D. H., Woodcroft, B. J. & Tyson, G. W. CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. *Nat Methods* **20**, 1203–1212 (2023).
10. Lundberg, S. M. & Lee, S.-I. A Unified Approach to Interpreting Model Predictions. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 4765–4774 (Curran Associates, Inc., 2017).
11. Erfanian, N. *et al.* Deep learning applications in single-cell genomics and transcriptomics data analysis. *Biomedicine & Pharmacotherapy* **165**, 115077 (2023).
12. Brendel, M. *et al.* Application of Deep Learning on Single-Cell RNA Sequencing Data Analysis: A Review. *Genomics, Proteomics & Bioinformatics* **20**, 814–835 (2022).
13. Lin, X., Tian, T., Wei, Z. & Hakonarson, H. Clustering of single-cell multi-omics data with a multimodal deep learning method. *Nat Commun* **13**, 7705 (2022).
14. Johnsen, P. V., Riemer-Sørensen, S., DeWan, A. T., Cahill, M. E. & Langaas, M. A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values. *BMC Bioinformatics* **22**, 230 (2021).
15. Pan, S., Zhao, X.-M. & Coelho, L. P. SemiBin2: self-supervised contrastive learning leads to better MAGs for short- and long-read sequencing. *Bioinformatics* **39**, i21–i29 (2023).
16. 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. *Nat Commun* **13**, 2326 (2022).
17. 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 Res.* **25**, 1043–1055 (2015).
18. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9**, (2008).
19. Arora, S., Hu, W. & Kothari, P. K. An analysis of the t-sne algorithm for data visualization. in *Conference on learning theory* 1455–1462 (PMLR, 2018).
20. Cai, T. T. & Ma, R. Theoretical foundations of t-sne for visualizing high-dimensional clustered data. *Journal of Machine Learning Research* **23**, 1–54 (2022).
21. Choi, Y., Li, R. & Quon, G. siVAE: interpretable deep generative models for single-cell transcriptomes. *Genome Biol* **24**, 29 (2023).
22. Grønbech, C. H. *et al.* scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).
23. Pan, W., Long, F. & Pan, J. ScInfoVAE: interpretable dimensional reduction of single cell transcription data with variational autoencoders and extended mutual information regularization. *BioData Mining* **16**, 17 (2023).
24. He, Y., Tan, H., Luo, W., Feng, S. & Fan, J. MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data. *Frontiers of Computer Science* **8**, 83–99 (2014).

25. Xiao, W. & Hu, J. A survey of parallel clustering algorithms based on spark. *Scientific Programming* **2020**, 8884926 (2020).
26. Monath, N. *et al.* Scalable Hierarchical Agglomerative Clustering. in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* 1245–1255 (ACM, Virtual Event Singapore, 2021). doi:10.1145/3447548.3467404.

REVIEWER COMMENTS

Reviewer #1 (Remarks to the Author):

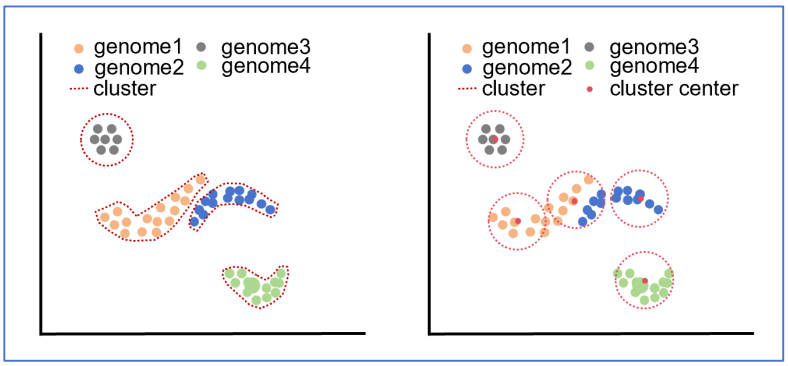
The authors answered most of our questions. Most edits were done with detail, and we thank the authors.

Response: Thank you for your fruitful suggestions, which improved the quality of our manuscript.

I still would like more biological meaning put on concave and convex clusters. Why are genome 1, 2, and 4 more complex? Can you discuss or at least hypothesize into why the different genomes made these clusters — since I would like know the biological significance of these different shapes of clusterings. Can you investigate more about this? Is there any computation->biological intuition?

Response: Thank you for your interest and valuable comments. We appreciate the suggestion to further discuss the computational and biological meaning of concave versus convex clusters.

First, the computational complexity of different genomes in the original Response Figure 1 is shown below. This figure is a conceptual diagram comparing two clustering outcomes: the left panel represents the theoretical "true" distribution of metagenomic contigs in 2D space, considering the shape of each genome. The right panel, by contrast, shows the clustering result assuming only central proximity (centroid-based) logic, such as K-means clustering [1]. This simplification can cause nearby but structurally distinct genomes to be merged or misclustered in the right panel. Eventually, high contamination will lead to low-quality metagenomic bins. In this schematic, genomes 1, 2, and 4 exhibit non-convex (crescent-like) distributions, whereas genome 3 is convex. Genomes 1, 2, and 4 with concave structures are misjudged by K-means because they are all treated as convex structures. Therefore, we selected DBSCAN and BIRCH, which perform better in nonconvex clustering, as the core clustering algorithms of LorBin binning to handle complex cluster shapes.



Response Figure 1. Conceptual outcomes of binning with or without accounting for convex and concave clusters. The quality of binning is reduced when two different genomes form concave clusters close to each other in the latent space (*e.g.*, genome 1 and genome 2). The left plot shows the theoretical clustering considering the shape of the clusters. Specifically, contigs from different genomes may be grouped into the same cluster (bin) based on the proximity of the center points (right plot). This approach increases the contamination of metagenomic bins (MAGs). The colorful dots denote contigs from different genomes (1 to 4). The red dotted line indicates the clustering boundary that considers only convex clusters, and the red dots mark the cluster centers. Genomes 1, 2, and 4 exhibit non-convex distributions, and genome 3 has a convex distribution.

Second, this difference in geometry of clusters might be not purely mathematical. This information may reflect some biological properties of the genomes or their corresponding contigs. Several potential factors may be associated with the shape of clusters, such as the GC content, contig length, and bin size. Biological interpretability analysis is an interesting project for our future research. We still would like to share and discuss this topic. For example, high GC contents cause sequencing biases and result in uneven coverage or k-mer distributions. These irregular patterns can distort their representation in feature space, making contigs toward nonconvex structures. Concave clusters may result from genomes that are poorly assembled or fragmented, often owing to sequencing difficulty (e.g., repetitive regions or low coverage). Short contigs have less reliable k-mers and coverage signals, leading to noisier embedding. To verify the biological meaning of cluster shapes, we propose collecting many concave and convex clusters and systematically comparing their biological features, such as GC content, N50, contig length, strain-level heterogeneity, and repetitive sequence density. Logistic regression, SHapley Additive exPlanations (SHAP), or other feature attribution methods could be used to identify which genomic attributes most strongly influence cluster shape in latent space. Thus, we only include some pieces of discussion in the main text at this time.

Discussion

and antimicrobial resistance (Supplementary Figure 17). Interpretability remains another key focus of future research, particularly for the structural characteristics of bins and AI-based binners. The shape of clusters in latent space may be associated with genomic features such as GC content, sequence complexity, and contig length or others. Non-convex clusters often indicate genomes with biased compositions or uneven assemblies. Understanding this biological significance can increase the accuracy of metagenomic binning. To increase the reliability of AI models, it is crucial to understand how models make binning decisions. Thus, developing

Reference:

1. Zhao, W.-L., Deng, C.-H. & Ngo, C.-W. k-means: A revisit. Neurocomputing 291, 195–206 (2018).

Reviewer #3 (Remarks to the Author):

Thank you for your thorough response.

All major points have been addressed to my satisfaction, except one which I believe still requires changes.

Response: Thank you. Our manuscript has been improved significantly based on your suggestions.

I'll address your responses to major issues raised by the reviewers point by point:

1. Claiming LorBin is reference free

A: The authors satisfyingly resphrased LorBin as unsupervised.

Response: Thank you.

2. Use of non-standard accuracy measures

A: The authors have included the more common ARI and F1 metrics, which is satisfying. I believe there are some methodological issues with ARI scores for metagenomic binning - see e.g. https://viralinstruction.com/BinBencherBackend.jl/dev/considerations/#ari. Nonetheless, I think this reservation of mine is minor, somewhat subjective, and should not block the publication of LorBin.

Response: Thank you for your reply and suggestions. We will include this in future research.

3. Lack of measurements of computational requirements

A: Authors have added supplementary tables with the requested information.

Response: Thank you.

4. Lack of discussion about how exactly LorBin advances the field

A: The authors have addressed this satisfactory by rewriting the discussion section.

Response: Thank you.

5. t-SNE is inappropriate to represent a VAE embedding

A: In my opinion, this has not been satisfactory addressed. Reviewer 1 asked a similar question, and the authors show in Response Figure 2 to reviewer 1 that the number of convex and concave clusters only changes slightly with t-SNE parameters. However, a) the fact that it changes, despite convexity being a mathematically rigorous definition does seem to prove that the t-SNE distorts the cluster shape, and b) the figure does not show whether the same clusters remain convex or concave across all 20 t-SNE settings - it could be that they switch, but with a fixed probability.

To address the same question in my review, the authors provide Response Figure 2 for reviewer 3, which shows that convexity change between PCA and t-SNE, and also differ as t-SNE parameters change. The authors does show that the convexity count is highly correlated when measured on PCA and t-SNE embeddings. Like above, they do not show that the same clusters remain convex/concave under PCA and t-SNE, and between t-SNE settings. In response figure 3, the authors show that, when manually adding convex or concave clusters, the number of convex and concave clusters detected with t-SNE correlates with the true number. However, only about half the number of truly convex clusters are detected in the t-SNE embedding, and approximately 20% more concave clusters are detected than what is manually added to the embedding. Further, they do not provide evidence that it's the same clusters that are actually convex/concave that are detected as such.

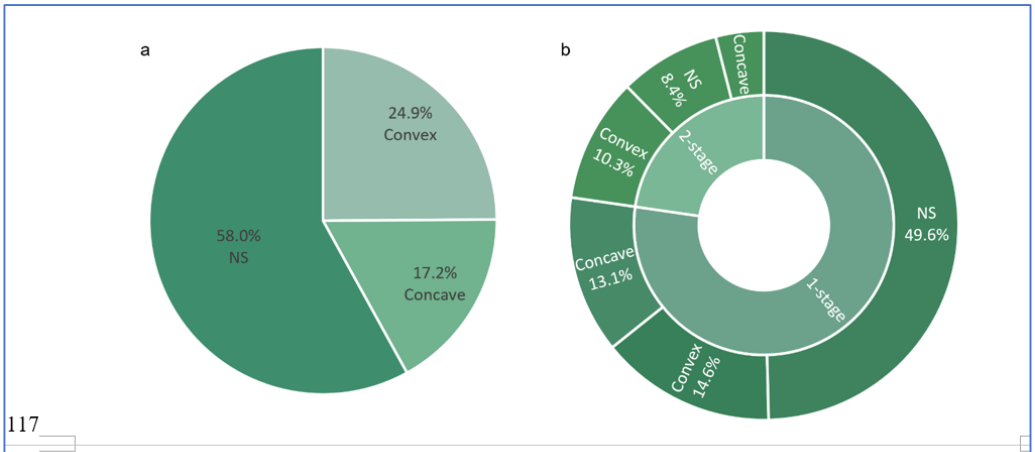
I conclude that the use of t-SNE remains unwarranted. The authors show, at best, that there is strong correlation between the true number of convex/concave clusters and the

measured number in a simulated setting, and that the number of measured convex clusters differ by less than 20% between PCA and t-SNE and between different t-SNE runs.
But this is not the same as showing that clusters that are convex in high dimensions are also convex in t-SNE - it is, in fact proof, that at least some of the time, t-SNE changes the shape of the clusters. I suggest the authors measure convexity directly in the high-dimensional space, if possible, or else simply forgo the mentioning of cluster shapes altogether, since it is not crucial to the paper's findings.

Response: Thank you for your insightful and professional comments. We acknowledge that t-SNE is a nonlinear dimensionality reduction method. This tool is not very appropriate for preserving global geometry and may distort cluster shape, including convexity. Our original intention in using t-SNE was not to claim that it faithfully reflects high-dimensional cluster structure. Although we can actually forgo it as you suggested, t-SNE provides intuitive visualization to help communicate the performance of LorBin on complex data distributions, particularly for non-computational biologists.

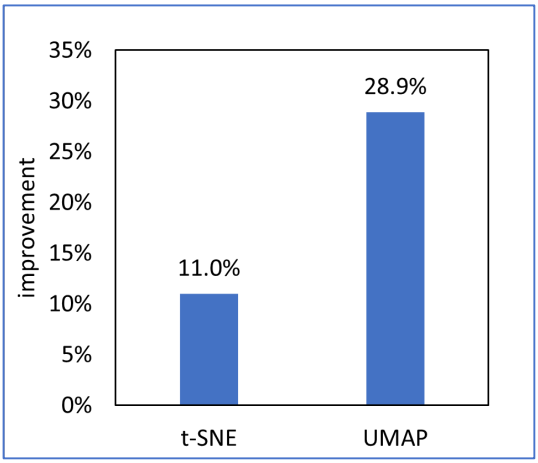
To address this concern, we took two complementary steps: first, we developed a high-dimensional approximation method to assess cluster convexity directly in the original feature space; second, we used this approximation to evaluate and improve the reliability of two-dimensional visualizations, ultimately leading us to replace t-SNE with UMAP.

1- We further attempted to approximate convexity in high-dimensional space by developing a deep learning-based classifier[1,2]. This classifier was trained on simulated convex and concave clusters. Specifically, we generated a synthetic dataset of 10,000 labelled clusters (50% convex, 50% concave), trained a transformer-based neural network[3], and achieved an average validation accuracy of 0.898 ± 0.010. This classifier was then used to predict convexity in 104 human gut metagenomic samples, revealing that ~25% of sufficiently large bins were convex and that ~17% were concave. Notably, 57% of the clusters had sizes ≤ 33 contigs and could not be classified due to dimensionality constraints[4,5]. Please find this result in the current Supplementary Figure 13 in the revision.



Supplementary Figure 13. Cluster shape characteristics analysis of 104 real samples from LorBin. The shape characteristics of clusters play a crucial role in selecting the appropriate clustering algorithm. We analyzed the shapes of bins extracted from high- and medium-quality bins via LorBin, categorizing them into convex clusters, concave clusters, and bins with a cluster size less than 33. (a) NS: the clusters with bin size smaller than 33, not involving in convexity classification for their bin size less or equal to the dimension of the space; convex: convex clusters; concave: concave clusters. (b) The inner ring represents the clustering stage where high- and medium-quality bins originate. The outer ring represents the shape of the clusters contained in each stage. The percentages on the outer ring are based on its parent ring. 1-stage: the first stage of two-stage clustering alogrithms; 2-stage: the second stage of two-stage clustering alogrithms.

2- We then used this classifier to guide a quantitative evaluation of dimensionality reduction techniques. Specifically, we performed a grid search over 1,000 parameter combinations for both t-SNE[6] and UMAP[7]. For each 2D embedding, we computed cluster shapes via ConvexHull[8] and compared the convexity assignments to those predicted by our high-dimensional classifier. This agreement rate served as a metric of shape preservation. After parameter optimization, the cluster shape consistency according to UMAP can reach 70.8% (n_neighbors=25, min_dist=0.6, random_state=42) on the 104 human intestinal samples. This value of 28.9% improvement over its default value is significantly higher than the maximum consistency achieved by t-SNE (only 11.0% improvement). Based on this evidence, we replaced t-SNE with UMAP in Figure 3e and clarified in the text that the visualization is intended only to show the handling of complex data distributions by LorBin, not to infer high-dimensional geometry.

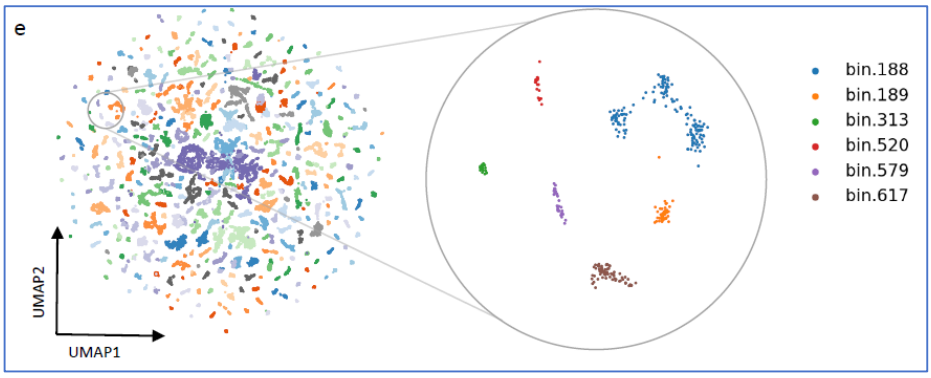


Response Figure 2. Shape consistency between high-dimensional space (via a convexity classifier) and 2D projections (via ConvexHull). Parameter optimization improved the consistency by 11.0% for t-SNE and 28.9% for UMAP.

These changes strengthen the manuscript by removing any overinterpretation of 2D projections

and ensuring that visualizations better reflect the underlying data structure. Importantly, the core conclusions of our work do not rely on t-SNE or UMAP but rather on the quantitative performance of LorBin in reconstructing genomes from complex metagenomes. The improved UMAP visual representation is intended to aid interpretability for broader audiences, particularly experimental biologists.

We updated Figure 3e in the Results and Methods sections.



Results

180 To demonstrate the capacity of LorBin to handle clusters differing in shape, we developed a transformer
181 classifier to assess the convexity of the embedded features of each bin. Among the obtained hmBins, 24.9%
182 and 17.2% were predicted to be convex and concave clusters, respectively (Supplementary Figure 13a).
183 Additionally, many of the unclassified clusters (~58%) with fewer than 33 contigs (bin size < 33) were generated
184 during the first stage of the LorBin workflow (Supplementary Figure 13b). We visualized their embedded
185 features via UMAP [24] (see Methods) to intuitively illustrate the complex shapes of bins in real datasets. The
186 UMAP embedding showed that bin.313 formed a convex cluster (i.e., approximated by a convex polygon),
187 whereas bin.188 exhibited a clearly concave structure (Figure 3e). In addition to human gut samples, LorBin

Methods

Convexity classifier

To directly evaluate the convexity of the embedded features of each bin, we design a neural classifier comprising a transformer encoder[57,58] and a classification head with two fully connected layers. The transformer architecture includes a self-attention layer and a feed-forward neural network, with the following parameters: an input size of 32, an embedding dimension of 64, two attention heads per layer, a feed-forward layer size of 128, and a maximum token length of 512. A synthetic dataset containing 10,000 clusters was generated, with a 1:1 ratio of convex to concave clusters. The convex clusters included hyperspheres and hyperellipsoids, whereas the concave clusters consisted of high-dimensional rings, crescents, and other nonconvex geometries. The classifier was trained for 100 epochs via the Adam optimizer (learning rate = 0.001) with fivefold cross-validation (training:validation ratio = 4:1), yielding a validation accuracy of 0.898 ± 0.010.

Cluster shape analysis

To demonstrate the adaptability of LorBin to clusters of different shapes, we visualized hmBins obtained from 104 real human intestinal samples. We used the convexity classifier with the best accuracy, which reached 0.910, to predict convexity on 104 real human intestinal samples. Bins containing fewer than 33 contigs (i.e., less than or equal to the ambient dimensionality of the space) were excluded from classification, as reliable convexity detection in such low-sample regimes is not feasible[59,60]. For clusters containing more than 512 contigs, we performed ten rounds of downsampling to 512 randomly selected contigs. A cluster was considered convex if it was classified as convex in at least 7 out of 10 replicates. To generate a two-dimensional visual representation consistent with a high-dimensional convexity structure, we use uniform manifold approximation and projection (UMAP). UMAP preserves both local and global topologies[24]. We used a grid search to confirm the parameters of UMAP and used the agreement rate between the convexity predicted in high-dimensional space and that inferred in 2D as the target objective. In 2D space, cluster convexity was determined via ConvexHull-based detection implemented via the scipy package[61]. The agreement rate was calculated as:

$$agreement\ rate = \frac{\sum_{i=1}^{N} y_1^i = y_2^i}{N} \times 100\%$$

where y_1^i and y_2^i represent the convexity classification of cluster i in high-dimensional space and 2D space, respectively, and N is the total number of clusters evaluated. The UMAP configuration that yielded the highest agreement rate (70.8%) was used for final visualization.

Reference:

1. Gm, H., Gourisaria, M. K., Pandey, M. & Rautaray, S. S. A comprehensive survey and analysis of generative models in machine learning. Computer Science Review 38, 100285 (2020).
2. Deep learning | Nature. https://www.nature.com/articles/nature14539.
3. Han, K. et al. A Survey on Vision Transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 87–110 (2023).
4. Pattern Recognition and Machine Learning.
5. Berg, M. de. Computational Geometry: Algorithms and Applications. (Springer Science & Business Media, 2000).
6. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. Journal of machine learning research 9, (2008).
7. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Preprint at https://doi.org/10.48550/arXiv.1802.03426 (2020).
8. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272 (2020).

REVIEWERS' COMMENTS

Reviewer #1 (Remarks to the Author):
Thank you for elaborating.
I won't push it any further for studying how biology shapes the clusters, But perhaps this can be an interesting follow-up study.I am satisfied.
Response: Thank you for your fruitful suggestions to improve the quality of our manuscript.

Reviewer #3 (Remarks to the Author):
Thank you for your response.
This round of revision concerned itself with the appropriateness of low-dimensional visualisation of a high-dimensional space, and in particular, whether the notion of "convexity" is preserved when reducing from higher- to lower dimensions.
I remain unconvinced that it's a good idea to attempt to reason about high-dimensional shapes as 2D projections, especially when there is good a priori reasons to believe that the overall shape of sequences is inherently high dimensional and therefore cannot be reduced.
However, I don't believe this disagreement is sufficiently important to prevent the publication of this paper. After all, it's not a critique of the methodology or even of an accuracy measurement, but only a critique of a visualisation meant to explain how LorBin works.
Response: Thank you. Our manuscript has been improved significantly based on your suggestions.