nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00908-5

A multi-modal deep language model for contaminant removal from metagenome-assembled genomes

Received: 8 October 2023

Accepted: 5 September 2024

Published online: 7 October 2024

Check for updates

Bohao Zou[1], Jingjing Wang[1], Yi Ding[1], Zhenmiao Zhang[1], Yufen Huang[2], Xiaodong Fang[2,3], Ka Chun Cheung[4], Simon See[4] & Lu Zhang[1,5]

Metagenome-assembled genomes (MAGs) offer valuable insights into the exploration of microbial dark matter using metagenomic sequencing data. However, there is growing concern that contamination in MAGs may substantially affect the results of downstream analysis. Current MAG decontamination tools primarily rely on marker genes and do not fully use the contextual information of genomic sequences. To overcome this limitation, we introduce Deepurify for MAG decontamination. Deepurify uses a multi-modal deep language model with contrastive learning to match microbial genomic sequences with their taxonomic lineages. It allocates contigs within a MAG to a MAG-separated tree and applies a tree traversal algorithm to partition MAGs into sub-MAGs, with the goal of maximizing the number of high- and medium-quality sub-MAGs. Here we show that Deepurify outperformed MDMclearer and MAGpurify on simulated data, CAMI datasets and real-world datasets with varying complexities. Deepurify increased the number of high-quality MAGs by 20.0% in soil, 45.1% in ocean, 45.5% in plants, 33.8% in freshwater and 28.5% in human faecal metagenomic sequencing datasets.

Short-read metagenomic sequencing has become popular for studying uncultured microbial genomes[1–4], However, it is challenging to generate complete microbial genomes with short-read assemblers[5–7]. Several contig binning tools[8–11] have been developed to group contigs with similar abundances and sequence contexts into metagenome-assembled genomes (MAGs) that represent microbial genomes. Some studies[12–14] claim that the quality of these MAGs is comparable to genomes from microbial isolates. However, there is growing concern that contamination may severely affect the qualities of MAGs[15]. MAG contamination refers to the presence of contigs from different microbes in the same MAG, resulting in chimeric MAGs that compromise the reliability of downstream ecological and evolutionary analyses. Bowers et al.[16] recommend eliminating the MAGs with more than 10% contamination, but this recommendation may miss microbes in MAGs with high completeness and marginal contamination. In our preliminary study, we found that a considerable number of MAGs including high-abundance MAGs, would be removed due to their marginal contamination levels (slightly higher than 10%) (Supplementary Note 1). This could lead to the loss of a substantial number of MAGs for downstream analysis. Several tools[17–20] have been developed to identify and remove potentially contaminated contigs from chimeric MAGs based on marker genes and the sequence characteristics of known species. Two previous pipelines[17,18] are no longer actively supported and have not gained widespread acceptance in the community. In addition to these two tools, MAGpurify[19] and MDMcleaner[20] were recently developed to remove contaminated contigs using different strategies. MAGpurify uses phylogenetic or clade-specific marker genes and analyses the guanine-cytosine (GC) contents and tetranucleotide frequencies of

[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong, China. [2]BGI Research, Shenzhen, China. [3]BGI Research, Sanya, China. [4]NVIDIA AI Technology Center, NVIDIA, Hong Kong, China. [5]Institute for Research and Continuing Education, Hong Kong Baptist University, Hong Kong, China. e-mail: ericluzhang@hkbu.edu.hk

https://doi.org/10.1038/s42256-024-00908-5

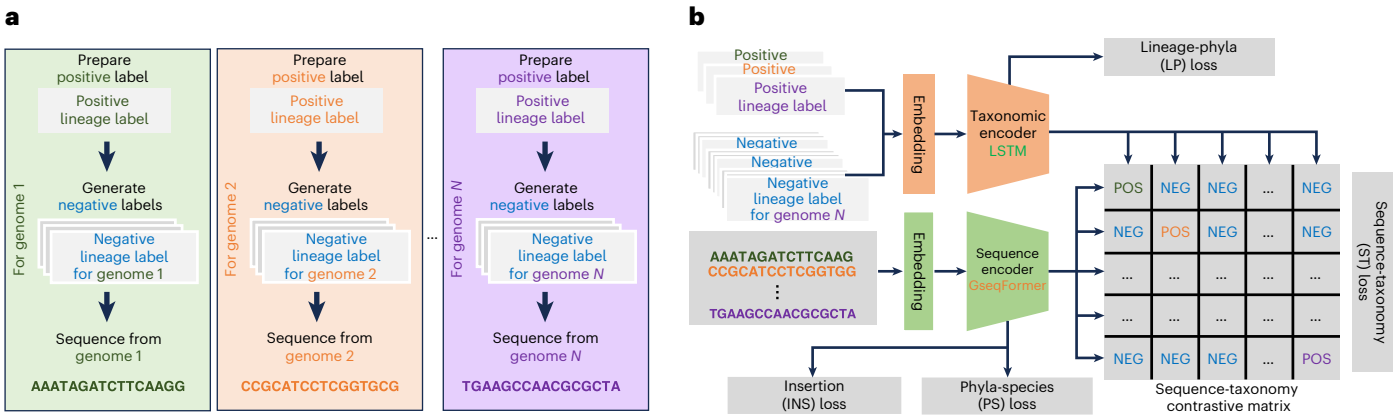

Fig. 1 | Deepurify training procedure. a, For data preparation, the representative microbial genomes were split into small segments. The true (positive) and fake (negative) taxonomic lineages for each genome were also collected. **b**, For model training, the LSTM was used to encode positive (POS) and negative (NEG) lineages. Genomic sequences were encoded using GseqFormer. A sequence-taxonomy contrastive matrix was built based on cosine similarities between embeddings of sequences and taxonomic lineages.

contigs to reduce MAG contamination. MDMcleaner uses marker genes (coding, 16S and 23S rRNA genes) to predict the taxonomic lineages of contigs. Contig taxonomies are determined by the taxonomic lowest common ancestor (LCA) of the involved marker genes. Any contigs with different taxonomic annotations from the dominant taxon of the MAG are removed.

Although MAGpurify and MDMcleaner have shown promising results, several issues hinder their widespread application in various scenarios. First, these methods require aligning marker genes or contigs to reference genomes, making them inapplicable to new microorganisms that are substantially different from those already known. Additionally, the alignment process is time consuming, even with optimized genome databases. Second, multiple factors[21] can reduce the performance of alignment-based tools in phylogenetic analysis. These factors include sequence misalignment, false-orthologous assignment, gene duplication or loss events, horizontal gene transfer, the presence of homoplasy and so on. Third, alignment-based tools do not consider various genomic alterations, such as genomic variations, alterations in gene order and genome rearrangements. Considering these genomic alternations could improve the resolution and reliability of distinguishing the genomic sequences from different species[22]. This evidence provides invaluable insights that can only be obtained through microbial whole genome sequences. Finally, in our preliminary study, we observed that most MAG contamination occurred at the genus and species levels (Supplementary Note 2). Unfortunately, both MAGpurify and MDMcleaner showed poor performance with these low taxonomic ranks (Supplementary Note 3).

In this study, we developed Deepurify, a multi-modal deep language model, for high-resolution and generalized MAG decontamination. During the training procedure, Deepurify used two encoders, GseqFormer (Methods) and long short-term memory (LSTM), to generate embeddings of genomic sequences and their source genomes' taxonomic lineages, respectively. These embeddings were then used in contrastive learning to establish relationships between these two types of modality (Methods and Fig. 1). In the decontamination procedure, Deepurify first assessed the taxonomic similarities of contigs in a MAG on the basis of their predicted taxonomic lineages (Fig. 2a). The predicted taxonomic lineages were used to construct a MAG-separated tree, where each node included contigs with the same taxonomic lineages at specific taxa. The contigs from each node were grouped into sub-MAGs on the basis of their sequence embeddings and annotated single copy genes (SCGs) (Methods and Fig. 2c). We implemented a tree traversal algorithm to select sub-MAGs that aim to maximize the total number of high- and medium-quality MAGs from the tree (Methods and Fig. 2d). Additionally, we implemented an iterative decontamination strategy called Deepurify_Iter to facilitate progressive decontamination on the MAGs from multiple binning tools (Supplementary Fig. 9). For simulated data, we observed that Deepurify outperformed the two state-of-the-art tools, MAGpurify and MDMcleaner in MAG decontamination (Fig. 3). Deepurify also demonstrated outstanding generalization capabilities, accurately identifying contaminated contigs even if their source genomes were absent from the training set (Fig. 4). For the Critical Assessment of Metagenome Interpretation (CAMI) I (ref. 23) and real-world metagenomic sequencing datasets, we used GUNC[24] to evaluate the contamination levels of MAGs after decontamination. For the CAMI I datasets, we applied MAGpurify, MDMcleaner, Deepurify, and Deepurify_Iter to MAGs generated by three contig binning tools: CONCOCT[9], MetaBAT2 (ref. 11) and SemiBin2 (ref. 25). The results showed that Deepurify and Deepurify_Iter substantially reduced MAGs' contamination, surpassing both MAGpurify and MDMcleaner for all binning tools (Table 1). Furthermore, we applied Deepurify_Iter to real-world metagenomic sequencing datasets with varying complexities, including samples from soil, ocean, plants, freshwater and human faeces. Our findings demonstrated that Deepurify_Iter substantially enhanced the quality of MAGs from all these samples (Table 1). We also observed that the performance of Deepurify_Iter remained robust even when dealing with the MAGs from highly complex ecosystems (Supplementary Note 4).

Results

Architecture of Deepurify and decontamination workflow

Deepurify is a multi-modal deep language model that was specifically developed to reduce the contamination of MAGs. Its architecture is similar to that of CLIP[26], a well-established multi-modal model incorporating two encoders: GseqFormer, which is designed to encode genomic sequences, and LSTM, which encodes taxonomic lineages (Methods). For model training, we downloaded 9,782 representative microbial genomes with complete genome taxonomy database (GTDB) taxonomic lineages for model training (Methods). During training, we used contrastive learning to enable Deepurify to distinguish between positive (real) and negative (fake) taxonomic lineages for a given sequence (Fig. 1). We encouraged the positive taxonomic lineages to have higher similarities than the given sequences compared with the negative ones. During the decontamination procedure (Fig. 2), Deepurify first assessed the taxonomic similarities of contigs by comparing the embeddings of their predicted taxonomic lineages (Fig. 2a). It then constructed a MAG-separated tree and assigned contigs from a MAG to

https://doi.org/10.1038/s42256-024-00908-5

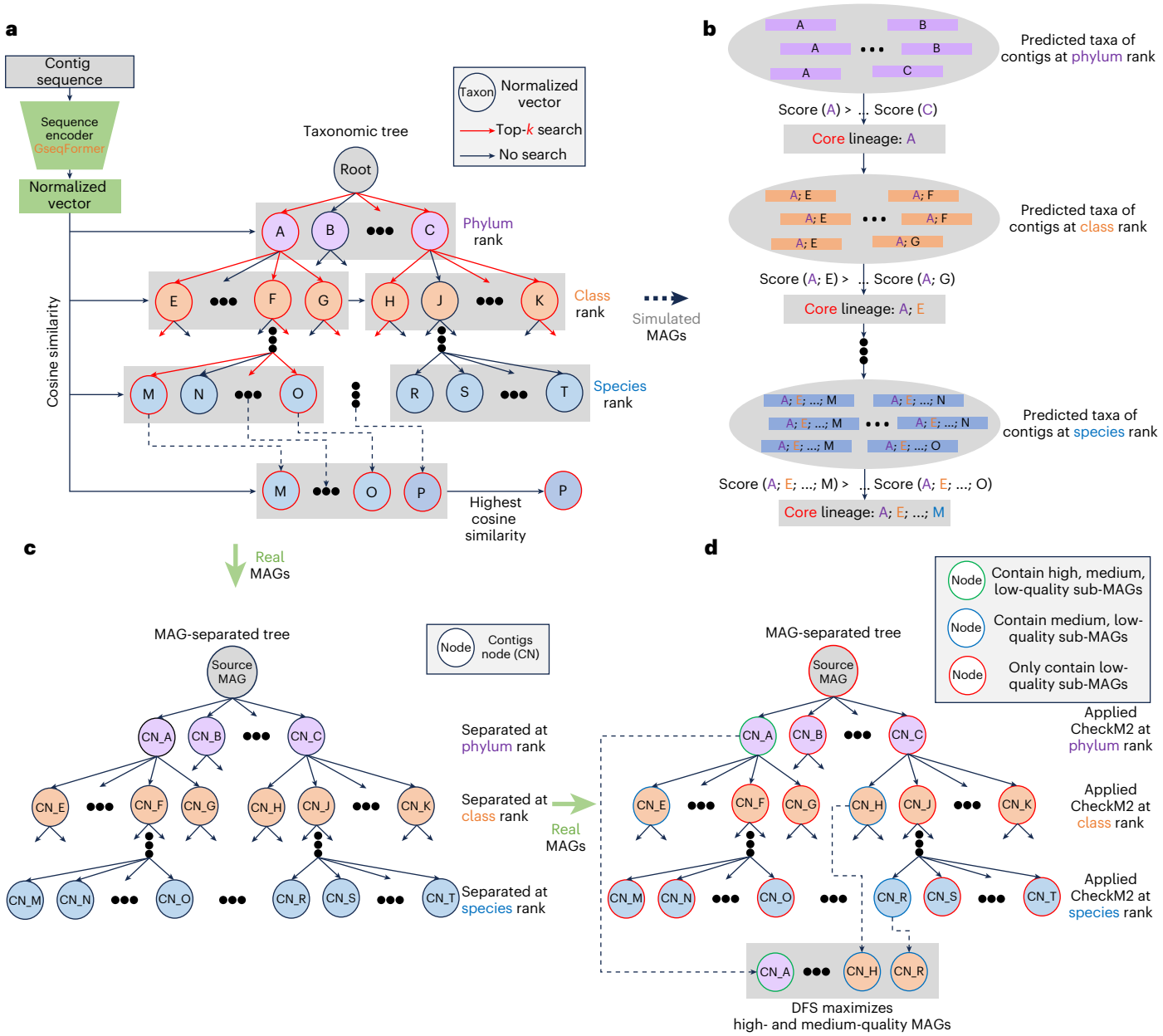

Fig. 2 | The workflow of Deepurify for MAG decontamination. a, Deepurify calculates taxonomic similarities between sequences and/or contigs on the basis of predicted taxonomic lineages. Deepurify first uses a top-*k* search in the taxonomic tree to generate the sequence's candidate annotations for each taxonomic rank and selects the taxonomic lineage with the highest similarity to the sequence. **b**, Deepurify determines the core lineage of a simulated MAG. The taxon with the highest score is incorporated into the core lineage of the MAG at each rank, given the previously established lineage. In the simulation testing sets, Deepurify removes the contigs that do not belong to the core lineage of the MAG. **c**, Deepurify constructs a MAG-separated tree to select contigs for CAMI and real-world metagenomic sequencing datasets. This tree partitions a MAG on the basis of its predicted taxonomic lineage. Each node contains contigs sharing the same taxon at a specific rank. Deepurify uses SCGs for each node to estimate the number of clusters and build 'cannot link' pairs for contigs. The COP-*K* means approach is applied to each node to group contigs into sub-MAGs based on their sequence embeddings. **d**, Deepurify uses a depth-first search (DFS) algorithm on the MAG-separated tree to maximize the total number of high- and medium-quality sub-MAGs.

the same node if they shared the same taxonomic lineages at specific taxa. We used COP-*K*-means[27] to group the contigs in each node into several sub-MAGs based on sequence embeddings and annotated SCGs (Fig. 2c). Deepurify applied CheckM2 (ref. 28) to each sub-MAG and used a depth-first search algorithm to traverse the MAG-separated tree and maximize the total number of high- and medium-quality sub-MAGs (Methods and Fig. 2d). Deepurify also supports iterative decontamination for ensemble binning. This optional module allows Deepurify to accept integrated results from multiple contig binning tools and iteratively select the high-quality MAGs (Methods and Supplementary Fig. 9).

Generation of training sets and test sets with chimeric MAGs

We generated two sets of high-quality representative microbial genomes: (1) a complete set (GS_c) consisting of 9,782 genomes (Methods), and (2) a partially complete set (GS_p) consisting of 94 randomly selected genomes from GS_c. The corresponding two training sets were constructed by cutting the genomes included in GS_c (Tr_c) and GS_p (Tr_p) (Methods). We simulated chimeric MAGs primarily consisting of core contigs (proportion 80–95%) and contaminated contigs (proportion 5–20%). The source genomes for these chimeric MAGs were core genomes (for core contigs) and contaminated genomes (for contaminated contigs). Two test sets with chimeric MAGs, SIM_1 and SIM_2, were

https://doi.org/10.1038/s42256-024-00908-5

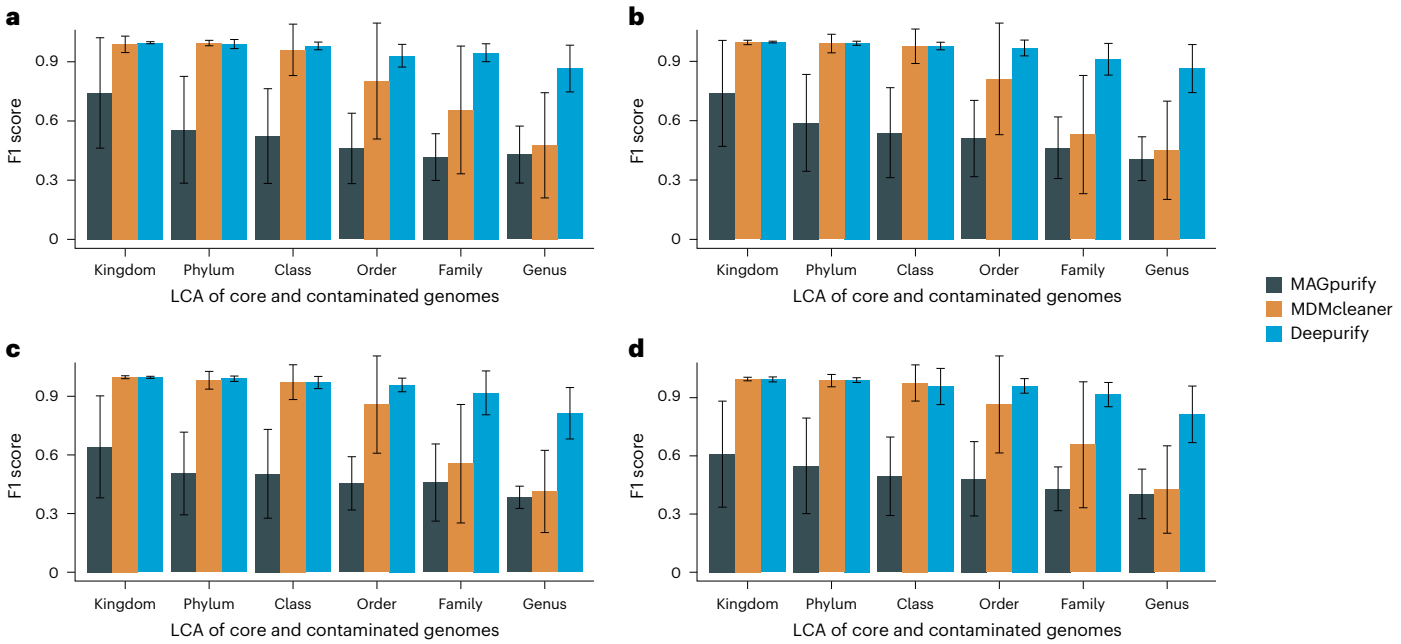

Fig. 3 | The averaged balanced macro F1 score across various contamination rates. a, 0.05. **b**, 0.1. **c**, 0.15. **d**, 0.2. Contamination rates along with the LCA of core and contaminated genomes at different taxonomic ranks, for MAGpurify, MDMcleaner and Deepurify (error bars represent standard deviation). The error bars, including mean and standard deviation values, are shown for $n = 50$ independent experiments.

generated to evaluate Deepurify's capability to distinguish core and contaminated contigs within a chimeric MAG (Methods). We trained Deepurify using Tr_c (Tr_p) and evaluated its performance on SIM_1 (SIM_2). SIM_1 was used to assess the performance of involved tools (MAGpurify, MDMcleaner and Deepurify) when both core and contaminated genomes were used to generate Tr_c. Thus, all source genomes used to generate SIM_1 were also observed in Tr_c. In SIM_2, the core and contaminated genomes might not be used to generate Tr_p. Therefore, SIM_2 included five different subtest sets: SIM_2^1, in which both core and contaminated genomes were used to generate Tr_p; SIM_2^2, in which only core genomes were used to generate Tr_p; SIM_2^3, in which only contaminated genomes were used to generate Tr_p; SIM_2^4, in which neither core nor contaminated genomes were used to generate Tr_p, with their taxonomic LCAs (LCA(c, t), where c is core genomes and t is contaminated genomes) were randomly chosen from kingdom to genus and SIM_2^5, in which neither core nor contaminated genomes were used to generate Tr_p, where LCA(c, t) was specifically assigned from Kingdom to Genus. Apart from the core and contaminated genomes, any remaining genomes from the children's nodes of LCA(c, t) must not be used to generate Tr_p.

Superior decontamination performance of Deepurify on SIM_1

We applied MAGpurify, MDMcleaner and Deepurify to process chimeric MAGs from SIM_1. Deepurify substantially outperformed MAGpurify across all taxonomic ranks of LCA(c, t) and contamination proportions (Fig. 3). Compared with MAGpurify, Deepurify increased the overall averaged F1 score by 42.14% (LCA(c, t) = kingdom), 80.16% (LCA(c, t) = phylum), 88.34% (LCA(c, t) = class), 100.21% (LCA(c, t) = order), 108.53% (LCA(c, t) = family) and 106.52% (LCA(c, t) = genus) across different contamination proportions (from 5 to 20%). While Deepurify and MDMcleaner delivered a comparable performance at the higher taxonomic levels of LCA(c, t), at lower levels, Deepurify showed marked improvements, achieving an average F1 score boost of 14.38% at the class level, 53.90% at the family level and 89.92% at the genus level, suggesting that Deepurify could be highly efficient for real-world metagenomic sequencing data because most of LCA(c, t) was found to exist at the family and genus levels (Supplementary Note 2). The F1 scores of MAGpurify, Deepurify and MDMcleaner (Fig. 3) declined as the taxonomic rank of LCA(c, t) became lower. This might be due to the higher proportion of homologous sequences between the core and contaminated genomes at lower levels of LCA(c, t).

Furthermore, the standard deviations (s.d.) of the F1 scores for Deepurify were considerably lower than those of MAGpurify and MDMcleaner, suggesting that Deepurify is more robust than the other two tools regardless of the taxonomic ranks of LCA(c, t). The s.d. of F1 scores for MAGpurify decreased steadily with lower taxonomic levels of LCA(c, t), whereas the s.d.s for Deepurify and MDMcleaner showed an increasing trend at these levels. This observation indicates that when the taxonomic rank of LCA(c, t) is low, MAGpurify tends to be more conservative in removing contaminated contigs while Deepurify and MDMcleaner proactively detect and eliminate contaminated contigs. When comparing different contamination rates, we noted a decrease in the performance of MAGpurify as the contamination rate increased. In contrast, Deepurify and MDMcleaner were more stable and robust regardless of the contamination rate, with Deepurify being more effective than MDMcleaner.

Strong generalization of Deepurify for new microbes

We could not evaluate MAGpurify and MDMcleaner on SIM_2 because both tools cannot deal with new microbes due to the lack of database rebuilding interfaces. We applied Deepurify on SIM_2 and found that its F1 scores were only marginally reduced regardless of the absence of core or contamination genomes in the training set Tr_p (Fig. 4). Deepurify demonstrated its best performance on SIM_2^1, followed by SIM_2^2 and SIM_2^5 with the lowest performance. This is because all or most of the contig source genomes in SIM_2^1 and SIM_2^2 were used to generate Tr_p, respectively. In SIM_2^5, the source genomes used were mostly different from those used to generate Tr_p. Nevertheless, we observed that the performance of Deepurify on SIM_2^3, SIM_2^4 and SIM_2^5 was comparable when LCA(c, t) was higher than family. This suggested that Deepurify performed well even if either core or contaminated genomes were absent from the training set. In addition, we observed a substantial decrease in the performance of Deepurify as LCA(c, t) reached the Genus level. These results indicated that addressing contamination at

https://doi.org/10.1038/s42256-024-00908-5

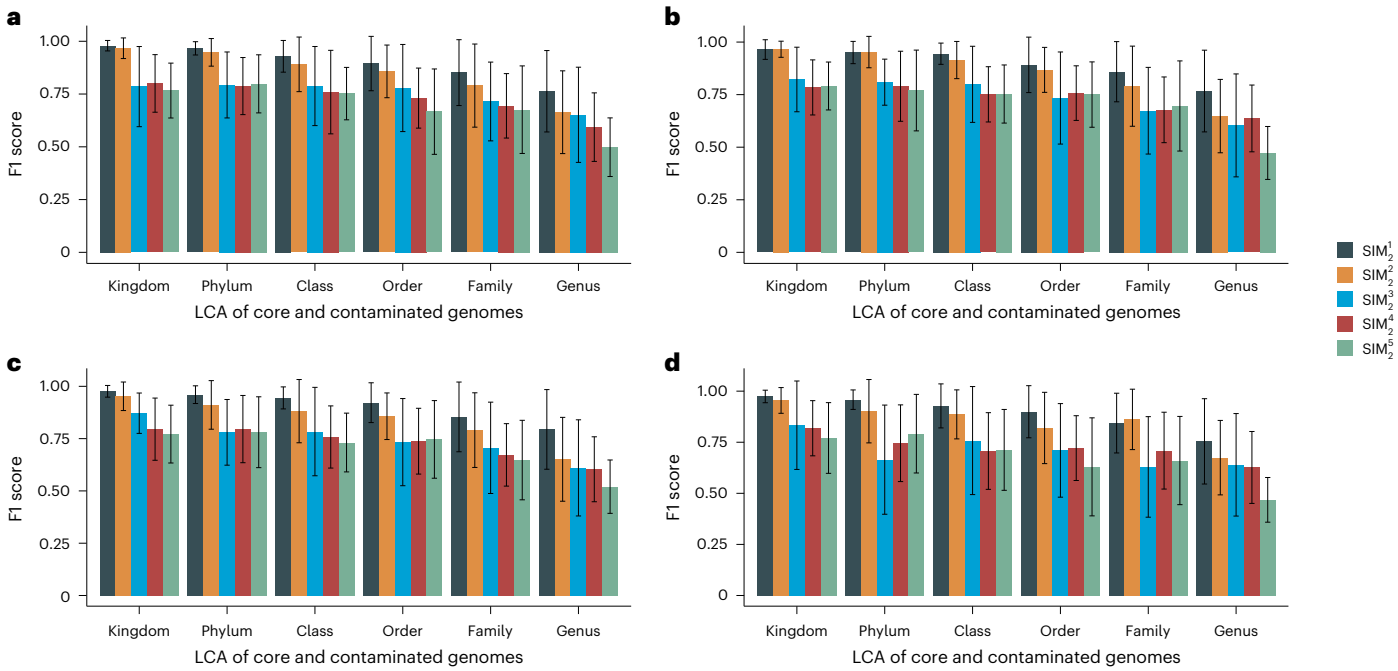

Fig. 4 | The averaged balanced macro F1 score across various contamination rates. a, 0.05. **b**, 0.1. **c**, 0.15. **d**, 0.2. Contamination rates along with the LCA of core and contaminated genomes at different taxonomic ranks, for SIM$_2^1$, SIM$_2^2$, SIM$_2^3$, SIM$_2^4$, SIM$_2^5$ (error bars represent standard deviation). The error bars, including mean and standard deviation values, are shown for $n = 50$ independent experiments.

the lower taxonomic ranks of LCA(c, t) was challenging due to the high prevalence of homologous sequences.

Improving MAGs' quality from binning tools on CAMI I data

We compared Deepurify, MAGpurify and MDMcleaner on the MAGs generated by CONCOCT, MetaBAT2 and SemiBin2 on CAMI I data, which encompassed four microbial community complexities. We evaluated the performance of MAG decontamination using two criteria: (1) the number of medium-quality (N_m) and high-quality (N_h) MAGs after evaluation by GUNC, and (2) the quality score (QS), which is used to measure overall MAG quality (Methods). Deepurify outperformed the other two tools across the MAGs generated by all three contig binning tools and four CAMI I datasets with varying complexities (Table 1). On average, Deepurify generated 4.23 and 8.28 times more high- and medium-quality MAGs compared to MDMcleaner. MAGpurify had a negligible effect on MAG decontamination. Deepurify showed a remarkable improvement in QS (QSI is the difference between the QS before and after decontamination), outperforming MDMcleaner by 3.30 times on average, whereas MAGpurify had a negligible QSI. MDMcleaner and MAGpurify occasionally failed to improve MAG qualities, while Deepurify consistently enhanced MAG qualities.

We performed ensemble binning by integrating MAGs produced by CONCOCT, MetaBAT2 and SemiBin2, which resulted in a higher number of qualified MAGs, with adequate completeness and marginal contamination (Methods). As expected, Deepurify achieved the highest number of high-quality MAGs (on average 57.50) than did MAGpurify (on average 50.00), MDMcleaner (on average 53.25) and Deepurify on individual binning tools (on average CONCOCT 51.00; MetaBAT2 48.25; SemiBin2 51.75). On applying an iterative decontamination strategy (Deepurify_Iter, Methods) on all four CAMI I datasets, we found that this approach further improved the number of high-quality MAGs (60.25 on average) than did the original Deepurify (57.50 on average).

Superior performance of Deepurify on real-world data

We evaluated the performance of Deepurify_Iter on real-world metagenomic sequencing data to examine its effectiveness in increasing the number of high-quality MAGs. The real-world metagenomic sequencing datasets used in this analysis were obtained from various complex environments, including soil (seven samples)[29], ocean (11 samples)[30], plants (three samples)[31,32], freshwater (three samples)[33–35] and human faecal samples (227 samples)[36]. Deepurify_Iter substantially outperformed MAGpurify and MDMcleaner in all datasets as shown in Table 1. The decontamination results for MAGpurify, MDMcleaner and Deepurify_Iter, using soil, ocean, plants, freshwater and human faecal samples are detailed in Supplementary Tables 5–9. MAGpurify had a detrimental effect on MAG quality across all datasets. Compared with MDMcleaner, Deepurify_Iter consistently generated a higher number of high-quality MAGs in all datasets (soil 24 versus 21; ocean 148 versus 117; plant 48 versus 34; freshwater 99 versus 76; human faeces 4,306 versus 3,441). Moreover, Deepurify_Iter exhibited a notable advantage over MDMcleaner in terms of QSI, showing an average of 4.02-fold across all datasets. In comparison to the original binning results, Deepurify_Iter produced 20.0% more high-quality MAGs for soil, 45.1% for ocean, 45.5% for plants, 33.8% for freshwater and 28.5% for human faeces datasets. Additionally, we used a generalized additive model to fit a curve demonstrating the correlation between completeness and contamination with all MAGs in the human faecal dataset. Our analysis showed that Deepurify_Iter achieved the lowest area under the curve (Fig. 5; original 506.1; MAGpurify 475.6; MDMcleaner 439.6; Deepurify_Iter 280.0). These findings indicate that Deepurify is capable of generating MAGs with the lowest contamination while maintaining completeness similar to that of other tools. We extended our assessment of Deepurify's performance by applying it to the outputs of individual binning tools (CONCOCT, MetaBAT2 and SemiBin2) on real-world metagenomic sequencing datasets. Our findings indicate that Deepurify improves the quality of MAGs generated by these tools (Supplementary Note 18).

Discussion

Genome assembly using short-read metagenomic sequencing data has become a common method for studying microbial dark matter in complex environments. However, single contigs only capture a segment of a full microbial genome. Therefore, contig binning is necessary to

https://doi.org/10.1038/s42256-024-00908-5

Table 1 | The number of high- and medium-quality MAGs that passed GUNC criterion on contamination, along with the quality scores (1k=1,000) across CAMI I and five real-world datasets

CONCOCT Pass GUNC	CAMI I high			CAMI I medium 1			CAMI I medium 2			CAMI I low		
	High	Medium	QS (k)	High	Medium	QS (k)	High	Medium	QS (k)	High	Medium	QS (k)
Before decontamination	22	33	4.24	23	4	2.32	25	9	2.89	8	5	1.03
MAGpurify	22	33	4.24	23	4	2.32	25	9	2.89	8	5	1.03
MDMcleaner	36	42	6.05	25	6	2.71	29	13	3.56	10	6	1.26
Deepurify	**126**	**184**	**22.65**	**28**	**22**	**3.88**	**36**	**27**	**4.96**	**14**	**10**	**1.84**
MetaBAT2 Pass GUNC												
Before decontamination	108	98	16.31	20	10	2.56	36	6	3.71	**13**	7	1.66
MAGpurify	108	98	16.31	20	10	2.56	36	6	3.71	**13**	7	1.66
MDMcleaner	113	121	18.21	**22**	10	2.76	35	14	4.21	12	8	1.64
Deepurify	**120**	**241**	**25.52**	**22**	**24**	**3.55**	**38**	**23**	**4.95**	**13**	**14**	**2.09**
SemiBin2 Pass GUNC												
Before decontamination	106	136	18.48	25	13	3.26	**39**	9	4.22	14	6	1.68
MAGpurify	106	136	18.47	25	13	3.26	**39**	9	4.22	14	6	1.68
MDMcleaner	121	144	20.53	25	17	3.55	37	11	4.21	13	7	1.66
Deepurify	**127**	**250**	**26.99**	**26**	**32**	**4.46**	**39**	**22**	**5.00**	**15**	**9**	**1.95**
Ensemble binning Pass GUNC												
Before decontamination	118	127	18.84	30	6	3.19	**39**	12	4.38	15	4	1.65
MAGpurify	116	128	18.80	30	6	3.19	**39**	12	4.38	15	4	1.64
MDMcleaner	134	157	21.98	29	5	3.05	35	**16**	4.31	15	**5**	**1.70**
Deepurify	142	**167**	23.32	33	6	3.44	**39**	13	4.49	16	3	**1.70**
Deepurify_Iter	**151**	166	**24.25**	**34**	**7**	**3.54**	**39**	13	**4.53**	**17**	2	**1.70**

Ensemble binning Pass GUNC	Soil (seven samples)			Ocean (11 samples)			Plants (three samples)			Freshwater (three samples)			Human faeces (227 samples)		
	High	Medium	QS (k)	High	Medium	QS (k)	High	Medium	QS (k)	High	Medium	QS (k)	High	Medium	QS (k)
Before decontamination	20	104	6.36	102	282	24.29	33	58	6.20	74	170	16.98	3,351	2,998	502.85
MAGpurify	19	104	6.33	76	309	23.96	27	66	6.16	63	201	16.70	2,933	**3,326**	493.54
MDMcleaner	21	135	7.76	117	322	29.11	34	84	7.18	76	180	17.90	3,440	3,209	525.57
Deepurify_Iter	**24**	**203**	**9.81**	**148**	**968**	**48.34**	**48**	**152**	**9.75**	**99**	**295**	**23.28**	**4,306**	3,044	**589.33**

'Ensemble binning' refers to the integration of MAGs from CONCOCT, MetaBAT2 and SemiBin2. The highest values have been highlighted in bold.

group contigs with similar sequence characteristics and abundances to represent microbial genomes. A recent study[15] highlighted that MAG contamination is a challenge during contig binning in metagenome assemblies. Tools such as MAGpurify and MDMcleaner have been developed to address this issue by removing contaminated contigs from MAGs. However, these tools have some limitations. They struggle to differentiate between contigs if the LCA of the core and contaminated genomes belong to the same family or genus. They also face challenges with contigs from source genomes that are absent in their reference databases. Moreover, these tools focus primarily on genes and overlook genomic variations such as gene order and genome rearrangements.

To overcome these challenges, we have developed Deepurify, a new multi-modal deep language model that can clarify the relationship between microbial genomes and their taxonomic lineages. Deepurify uses microbial whole genomic sequences, enabling it to distinguish between closely related species in the training set. This approach allows Deepurify to handle contigs without annotated genes and the source genomes of contigs that are absent from the training dataset. Deepurify has demonstrated superior decontamination performance, especially when processing contigs that contain a considerable number of homologous sequences (Supplementary Note 3). Furthermore, Deepurify can substantially speed up the MAG decontamination process through graphics processing unit acceleration, enabling efficient scaling across a large number of MAGs. The most time-consuming component of Deepurify_Iter is the iterative binning module, which requires several iterations of contig binning (Supplementary Note 15). In practice, we found three iterations were typically sufficient to generate all possible high-quality MAGs.

Deepurify implemented a clustering method based on a MAG-separated tree to select contigs within each MAG, with the aim of maximizing the total number of high- and medium-quality MAGs. To test the effectiveness of this approach, we conducted two ablation studies: the first retained only the contigs from the predominant taxonomic lineage in a MAG, and the second involved a greedy algorithm designed to iteratively eliminate the contig with the largest distance from others until QS of MAGs would not be increased. The results showed that our strategy outperformed both of these alternative approaches (Supplementary Note 16).

https://doi.org/10.1038/s42256-024-00908-5

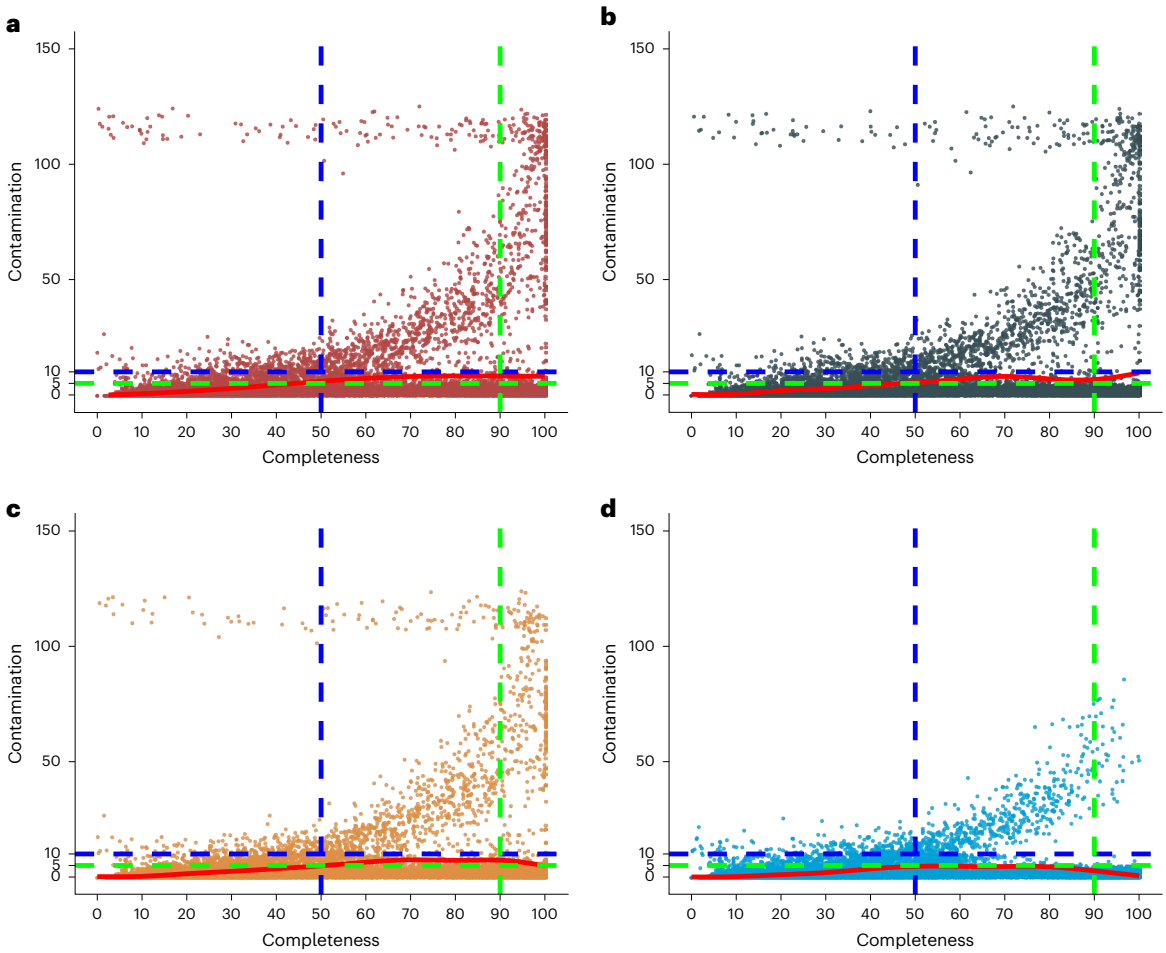

Fig. 5 | Distribution of the completeness and contamination of MAGs before and after decontamination on 227 human faecal samples by CheckM2. **a**, Before decontamination. **b**, MAGpurify. **c**, MDMcleaner. **d**, Deepurify_Iter. The red curve was generated using the generalized additive model on completeness and contamination of MAGs. The blue (green) dashed lines represent the thresholds of medium-quality (high-quality) MAGs.

The complexity of the microbial community may be a crucial factor associated with Deepurify's performance. In our experiments, we found that Deepurify maintained consistent performance on CAMI I datasets with both low and high complexity (Table 1). We used Nonpareil[37] to quantify the community complexity of the real-world metagenomic sequencing datasets (Supplementary Note 4). Deepurify improved both the overall QS and the number of high-quality MAGs for all samples, and its performance was not substantially affected by community complexity. Another key factor that may affect Deepurify's performance is the quality of the metagenome assembly. Metagenome assemblies from high-complexity communities may show low contig continuity (for example, low contig N50), which can result in inaccurate predicted taxonomic lineages for these shorter contigs. Conversely, high-complexity communities may produce a greater number of eligible MAGs that can be improved to medium- or high-quality MAGs through Deepurify.

Note that Deepurify is unable to handle large misassemblies in contigs, such as chimeric contigs and translocations. Deepurify failed to meet the contamination thresholds for some MAGs due to the presence of more than one SCG in the same contigs. Chimeric contigs thus remain a challenge for Deepurify as they can substantially alter the sequence context, thereby affecting the accurate assessment of taxonomic similarity among contigs within a MAG. To mitigate the influence of these misassemblies, we recommend that users apply assembly error correction tools such as metaMIC[38] before using Deepurify.

Deepurify relies exclusively on sequence context without considering contig covariance abundance across samples, which may lead to the unintentional removal of strain-specific functional genes in contigs whose sequence context does not closely resemble the reference genomes in the training set. While these strain-specific genes may not substantially affect the overall completeness or contamination of purified MAGs, their removal could result in the loss of unique strain-specific functions, potentially influencing downstream functional analyses. To mitigate this issue, re-binning can be performed after Deepurify's purification process. By using the contigs from a purified MAG as anchors, contig binning tools can be reapplied to include other contigs with strain-specific genes in the MAG if they have strong support from covariance abundance and tetranucleotide frequencies.

Our experiments demonstrated the remarkable efficacy of Deepurify in decontaminating MAGs from short-read metagenomic assemblies. We believe that Deepurify could also be applied to contigs from long-read assemblies due to its two distinct advantages: (1) contigs from long-read assemblies are substantially longer than those from short-read assemblies, providing Deepurify with a substantially enriched sequence context, thereby enhancing its capacity for decontamination, and (2) single-base substitutions and insertion errors are frequently observed in long-read assemblies[39]. We specifically focused on addressing these issues in the training procedure of Deepurify (Supplementary Note 5), unlike MAGpurify and MDMcleaner, which typically do not typically consider sequencing errors.

Methods

Preparing and processing representative microbial genomes

We obtained representative microbial genomes and their National Center for Biotechnology (NCBI) taxonomic lineages from the proGenomes v.2.1 database[40]. In cases where multiple genomes had the same taxonomic lineages, we retained only the largest genomes. We excluded microbial genomes that lacked phylum annotations or belonged to the phyla with fewer than 15 genomes. We further annotated the remaining genomes using GTDB-Tk[41] and removed genomes with incomplete GTDB's taxonomic lineages from phylum to species (Supplementary Note 6). The genome was considered annotated as a particular species (g_{ref} represents the corresponding genome in the GTDB database) if two conditions were met: (1) its average nucleotide identity with g_{ref} was at least 95% and (2) genome coverage of g_{ref} was at least 65%. In this manner, we obtained 9,782 representative microbial genomes with complete GTDB taxonomic lineage annotations.

Constructing training sets

During model training, Deepurify would classify identical sequences with the same taxonomic labels, even if they originate from different species. This scenario can introduce noise and result in false-negative classifications, which would affect the performance of Deepurify. To address this issue, we used MMSeqs2 to cluster similar sequences and selected only representative sequences from each cluster to minimize redundancy. The sequences from the genomes in GS_c and GS_p were segmented into 8,192 bp fragments, each with a 512 bp overlap. The 'minimum coverage' parameter in MMSeqs2 was set to 0.5.

Simulating chimeric MAGs in test sets

For SIM_1, we randomly selected two genomes from different lineages (SP_1 and SP_2) in GS_c and simulated 200 contigs with lengths between 1,000 and 8,192 base pairs (bp) and varying proportions of contigs from SP_2 (5, 10, 15 and 20%) for each MAG. The LCAs of taxonomic lineages of SP_1 and SP_2 were traversed from kingdom to genus. We generated 50 MAGs for each mixture proportion and on each taxonomic rank of LCA. We followed a similar chimeric MAG simulation procedure for SIM_2, with the only difference being that SP_1 and SP_2 could be selected from either GS_p or $GS_c - GS_p$.

Generating MAGs for real-world metagenomic sequencing data

We assembled metagenomic sequencing data from the soil samples and two samples (SRR14308228, SRR14308230) of plants using MegaHit[42] with default parameters due to the complex nature of the data. The other datasets were assembled by metaSPAdes[5] with default parameters. The contigs were grouped into MAGs using CONCOCT (contig length >1 kilobp (kbp)), MetaBAT2 (contig length >1.5 kbp) and SemiBin2 (contig length >1 kbp). CONCOCT was set with the parameter 'iterations = 200'. MetaBAT2 was run with default parameters, and SemiBin2 was run without recluster. All other parameters were set to default.

Defining MAG quality

MAGs were typically classified into three categories based on their completeness and contamination. MAGs were considered high quality if their completeness was ≥90% and contamination ≤5%. MAGs were considered medium quality if their 50% ≤ completeness < 90% and 5% < contamination ≤ 10%. MAGs that did not meet the criteria for high or medium quality were categorized as low-quality MAGs.

Metrics for performance evaluation

For simulated data, we applied a balanced macro F1 score to evaluate the performance of MAG decontamination, which mitigated the influence of unbalanced numbers of contigs in simulated chimeric MAGs. For the MAGs generated from CAMI I and the real-world datasets, we adopted two criteria to evaluate the improvement in MAG qualities: (1) the numbers of high- and medium-quality MAGs after quality control on contamination by GUNC, and (2) the QS for each dataset.

$$\mathrm{QS} = \sum_{i}^{n} (\mathrm{CN}_i - 5 \times \mathrm{CT}_i) \tag{1}$$

where n denotes the total number of high- and medium-quality MAGs passing the GUNC criterion on contamination, and CN_i and CT_i are the completeness and contamination scores of the MAG i that satisfy the GUNC criterion on contamination.

Architecture of Deepurify

The encoders for genomic sequences and taxonomic lineages. During the training procedure, Deepurify used GseqFormer and LSTM to encode genomic sequences and taxonomic lineages of their source genomes into 1,024-dimensional space. The overall architecture of Deepurify is shown in Fig. 1b.

Architecture of GseqFormer for genomic sequence embedding. A genomic sequence was represented as a unified embedded matrix by concatenating the sequence representations with one-hot, 3-mers and 4-mers (Supplementary Note 7). We developed GseqFormer to encode the sequence-embedded matrix in high-dimensional space. It was built on the structure of UniFormer[43], which had the advantage of transformer and convolutional neural networks. However, considering the limited modelling ability of UniFormer, we replaced the attention module of UniFormer with a new gated self-attention module, modified from Evoformer[44] (Supplementary Note 8). To handle long sequences (up to 1,000 bp), we incorporated EfficientNet[45] to compress the length of the sequence embedding into 512 tokens, allowing for input sequences of up to 8,192 bp. Furthermore, we implemented various techniques[46–48] for efficient training and improving model robustness (Supplementary Note 9). Supplementary Fig. 4 provides more details on the architecture of GseqFormer.

Architecture of LSTM for taxonomic lineage embedding. The taxonomic lineage of GTDB ($T_i = [t_{p_i}, t_{c_i}, t_{o_i}, t_{f_i}, t_{g_i}, t_{s_i}]$) of a sequence ($s_i$) was considered as a sentence that concatenated taxon (t_{k_i}) at different taxonomic ranks ($k_i = \{p_i, c_i, o_i, f_i, g_i, s_i\}$), including phylum ($p_i$), class ($c_i$), order ($o_i$), family ($f_i$), genus ($g_i$) and species ($s_i$). This taxonomic sentence was encoded by a five-layer LSTM model.

Training procedure of Deepurify

Contrastive learning. We leveraged contrastive learning to enable Deepurify to discriminate positive (true taxonomic lineage, T_{k_i}) and multiple negative (fake taxonomic lineages, T_{k_j}) labels for a given sequence s_i. During training, we randomly selected k_i and created fake taxonomic lineages ($T_{k_j} = [\leqslant t_{k_j}]$, the prefix of T_j before k_j rank) from the taxonomic tree for any other sequence s_j (Supplementary Note 10).

We applied four loss functions in contrastive learning: (1) sequence-taxonomy (ST) loss, (2) lineage-phyla (LP) loss, (3) insertion (INS) loss and (4) phyla-species (PS) loss. Deepurify's primary objective was to optimize sequence-taxonomy loss, which aims to maximize the cosine similarity of θ_{s_i} (normalized sequence embedding of s_i) with $\theta_{T_{k_i}}$ (normalized taxonomic lineage embedding of T_{k_i}) over $\theta_{T_{k_j}}$. The sequence-taxonomy loss (L_{ST}) is defined as follows:

$$L_{\mathrm{ST}} = -\left[\left(1 - P\left(\theta_{s_i}, \theta_{T_{k_i}}\right) \log\left(P\left(\theta_{s_i}, \theta_{T_{k_i}}\right)\right)\right] \tag{2}$$

$$P\left(\theta_{s_i}, \theta_{T_{k_i}}\right) = \frac{\exp\left(d\left(\theta_{s_i}, \theta_{T_{k_i}}\right)/\tau\right)}{\sum_{j=1}^{J} \exp\left(d\left(\theta_{s_i}, \theta_{T_{k_j}}\right)/\tau\right) + \exp\left(d\left(\theta_{s_i}, \theta_{T_{k_i}}\right)/\tau\right)} \tag{3}$$

where $d(\theta_{s_i}, \theta_{T_{k_i}}) = \theta_{s_i}^T \theta_{T_{k_i}}$, τ is a learnable parameter and J is the number of negative labels used in contrastive learning. The numbers of species

are different across phyla, resulting in unbalanced sequences involved in the training set. To mitigate this problem, we applied an oversampling strategy (Supplementary Note 11) and the focal loss[49].

Lineage-phyla loss (LP) aims to establish a taxonomic encoder to minimize the distance between $\theta_{T_{k_i}}$ and the phylum that s_i belongs to ($\theta_{t_{p_i}}$).

$$L_{\mathrm{LP}} = \mathrm{ReLU}\left(\alpha - d\left(\theta_{T_{k_i}}, \theta_{t_{p_i}}\right)\right) + \mathrm{ReLU}\left(d\left(\theta_{T_{k_i}}, \theta_{t_{p_j}}\right) - \beta\right) \quad (4)$$

where α and β are between 0 and 1, which control the cosine similarities of $d(\theta_{T_{k_i}}, \theta_{t_{p_i}})$ and $d(\theta_{T_{k_i}}, \theta_{t_{p_j}})$, and ReLU stands for rectified linear unit.

The insertion (INS) loss aims to enable GseqFormer to accept sequences with new insertions.

$$L_{\mathrm{INS}} = -\left[Y_{\mathrm{ins}}\log(P_{\mathrm{ins}}(\theta_{s_i})) + (1 - Y_{\mathrm{ins}})\log(1 - P_{\mathrm{ins}}(\theta_{s_i}))\right] \quad (5)$$

where $P_{\mathrm{ins}}(\theta_{s_i})$ is the predicted probability of s_i including insertions, and $Y_{\mathrm{ins}} = 1$ indicates s_i including insertions.

Phyla-species loss is used to examine the taxonomic inference of Deepurify on phylum and species ranks.

$$L_{\mathrm{PS}} = -\left[\sum_{h=1}^{H} Y_h \log(P_h(\theta_{s_i})) + \sum_{e=1}^{E} Y_e \log(P_e(\theta_{s_i}))\right] \quad (6)$$

where H and E are the number of phyla and species in the taxonomic tree; $Y_h = 1$ and $Y_e = 1$ if s_i belongs to the phylum h and species e and $P_h(\theta_{s_i})$ is the predicted probability of s_i belonging to the phylum h and $P_e(\theta_{s_i})$ is the predicted probability of s_i belonging to the species e.

Therefore, the training loss function of Deepurify can be defined as follows:

$$L = \gamma L_{\mathrm{ST}} + L_{\mathrm{LP}} + L_{\mathrm{INS}} + L_{\mathrm{PS}} \quad (7)$$

We set $\gamma = 2$ in our experiments to emphasize the importance of L_{ST} in model training. The settings of other hyper-parameters were similar to those in UniFormer[43] (Supplementary Note 12).

During training, we sampled the contig-sized sequences from the sequences in $\mathrm{Tr_c}$ and $\mathrm{Tr_p}$, with sequence lengths ranging from 1,000 to 8,192 bp, following a uniform distribution. We randomly incorporated insertions and single nucleotide variants (Supplementary Note 5) into these sampled sequences to reduce the impact of sequencing errors and enhance model generalization capabilities.

MAG decontamination

Quantifying sequence taxonomic similarity. The similarity between sequences is positively correlated with the similarity of their predicted taxonomic lineages. For sequence s_i, GseqFormer calculates the $P(\theta_{s_i}, \theta_{T_{j,k}}), j = 1 \ldots n$ for every taxon j at taxonomic rank k, where $T_{j,k} = [<t_{k,j}]$, and n is the total number of taxa in rank k. Deepurify selects the three candidate taxa with the highest values. The calculation of $P(\theta_{s_i}, \theta_{T_{j,k}})$ is similar to equation (3). For rank $k+1$, Deepurify only searches for those nodes whose parents have been selected in rank k. This top-k searching strategy results in a number of paths, ω, from the root to the species rank ($T_j, j = 1 \ldots \omega$). Deepurify then calculates $P(\theta_{s_i}, \theta_{T_j}), j = 1 \ldots \omega$ to select the best path.

Detecting contaminated contigs in simulated MAGs. For simulated data, a contig with low taxonomic similarities to the others in a MAG is more likely to be contaminated. We identified contigs as contaminants when their predicted taxonomic lineages did not match the most common lineage (core lineage) within the MAG (Fig. 2b). To facilitate this process, we collected the taxonomic lineages of all contigs in a MAG and developed a method to score and determine the core lineage. The $\mathrm{Score}_{j,k}$ was calculated for taxon j at rank k as follows,

$$\mathrm{Score}_{j,k} = \lambda \frac{1}{n_i} \sum_{i}^{n_i} P(\theta_{s_i}, \theta_{T_{j,k}}) + \mu V_{j,k} + \nu L_{j,k} \quad (8)$$

Here, n_i is the number of contigs with predicted lineage j at rank k. $V_{j,k}$ and $L_{j,k}$ denote the proportions of contigs and their total length in a MAG with the taxonomic lineage of $T_{j,k}$, respectively. $T_{j,k}$ with the highest score is then selected as the core lineage at rank k. This selection was performed for each rank, such that the selected core lineage at rank $k+1$ should be the offspring of the one at rank k.

Optimizing contig use in MAGs. For metagenomic sequencing data, Deepurify used a MAG-separated tree to divide contigs within a MAG. This method maximized the use of contigs within a MAG, thereby potentially increasing the number of medium- and high-quality MAGs. The construction of the MAG-separated tree was based on the predicted taxonomic lineages of all contigs in a MAG (Fig. 2c). The MAG-separated tree comprised six levels, ranging from phylum to species with each node containing the contigs that shared the same annotation at a particular rank. We used SCGs to estimate the number of clusters within each node of the MAG-separated tree (Supplementary Note 13). Additionally, SCGs enabled the establishment of 'cannot link' pairs between contigs. Two contigs with the same SCG would be identified as a 'cannot link' pair. We then grouped the contigs in each node into sub-MAGs using COP-*K*-means, which was capable of considering both contig embeddings produced by GseqFormer and the pairwise 'cannot link' constraints. We considered only those sub-MAGs with a total contig length of at least 200 kbp and evaluated their qualities using CheckM2. For node i in the MAG-separated tree, Deepurify calculated the number of high-quality sub-MAGs (#HQ(i)), the number of median-quality sub-MAGs (#MQ(i)) and the summation of QS (ΣQS(i)) for all sub-MAGs. Deepurify updated the contig set of node i to the contig sets of its children (C_i) if one of the following criteria was satisfied: (1) #HQ(i) < #HQ(C_i); (2) #HQ(i) = #HQ(C_i) and #MQ(i) < #MQ(C_i) and (3) #HQ(i) = #HQ(C_i), #MQ(i) = #MQ(C_i) and ΣQS(i) < ΣQS(C_i). Deepurify repeated this procedure from the leaves to the root of the MAG-separated tree by postorder traversal (Fig. 2d and Supplementary Note 14).

Applying Deepurify to ensemble contig binning

We used MAGs generated by CONCOCT, MetaBAT2 and SemiBin2 for both CAMI I and real metagenomic sequencing datasets. For each iteration, we used Deepurify_Iter and selected only high-quality MAGs. The remaining contigs were grouped by the three binning tools again. We iteratively repeated this process until either no further high-quality MAGs could be identified or after three iterations. Finally, we collected all MAGs from each iteration and used dRep[50] to compare and remove duplicated MAGs (Supplementary Fig. 9).

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The microbial representative genomes and their associated taxonomic lineages were downloaded from the proGenomes v.2.1 database. The GTDB r202 was used to annotate the reference genomes. The $\mathrm{SIM_1}$ test set is available via Zenodo at https://zenodo.org/record/8343498 (ref. 51). The $\mathrm{SIM_2}$ test set is available via Zenodo at https://zenodo.org/records/11608439 (ref. 52). The CAMI I short reads were downloaded from the 1st CAMI Challenge Dataset 1 CAMI_low, 1st CAMI Challenge Dataset 2 CAMI_medium and 1st CAMI Challenge Dataset 3 CAMI_high from https://data.cami-challenge.org/participate. The NCBI SRA accessions of seven soil samples are SRR25158210, SRR25158221, SRR25158244, SRR25158253, SRR25158281, SRR25158363 and SRR25158536; those of the three freshwater samples are ERR4195020, ERR9631077 and SRR26420192; those of the three plant samples are SRR10968246, SRR14308228 and SRR14308230. The 11 ocean samples

are from ref. 30. The human faecal metagenomic sequencing reads of the IBS-D cohort were downloaded from China National GeneBank with accession number CNPO000334.

Code availability

The source code is freely available at https://github.com/ericcombiolab/Deepurify/ (ref. 53) under an MIT licence. The versions of the software used in the study are provided in Supplementary Note 17.

References

1. Bernard, G., Pathmanathan, J. S., Lannes, R., Lopez, P. & Bapteste, E. Microbial dark matter investigations: how microbial studies transform biological knowledge and empirically sketch a logic of scientific discovery. *Genome Biol. Evol.* **10**, 707–715 (2018).
2. Dam, H. T., Vollmers, J., Sobol, M. S., Cabezas, A. & Kaster, A.-K. Targeted cell sorting combined with single cell genomics captures low abundant microbial dark matter with higher sensitivity than metagenomics. *Front. Microbiol.* **11**, 1377 (2020).
3. Kaster, A.-K. & Sobol, M. S. Microbial single-cell omics: the crux of the matter. *Appl. Microbiol. Biotechnol.* **104**, 8209–8220 (2020).
4. Pratscher, J., Vollmers, J., Wiegand, S., Dumont, M. G. & Kaster, A.-K. Unravelling the identity, metabolic potential and global biogeography of the atmospheric methane-oxidizing upland soil cluster α. *Environ. Microbiol.* **20**, 1016–1029 (2018).
5. Nurk, S., Meleshko, D., Korobeynikov, A. & Pevzner, P. A. metaspades: a new versatile metagenomic assembler. *Genome Res.* **27**, 824–834 (2017).
6. Liang, K.-C. & Sakakibara, Y. Metavelvet-dl: a metavelvet deep learning extension for de novo metagenome assembly. *BMC Bioinforma.* **22**, 427 (2021).
7. Kolmogorov, M. et al. metaflye: scalable long-read metagenome assembly using repeat graphs. *Nat. Methods* **17**, 1103–1110 (2020).
8. Nissen, J. N. et al. Improved metagenome binning and assembly using deep variational autoencoders. *Nat. Biotechnol.* **39**, 555–560 (2021).
9. Alneberg, J. et al. Binning metagenomic contigs by coverage and composition. *Nat. Methods* **11**, 1144–1146 (2014).
10. Wu, Y.-W., Tang, Y.-H., Tringe, S. G., Simmons, B. A. & Singer, S. W. Maxbin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome* **2**, 26 (2014).
11. Kang, D. D. et al. Metabat 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**, e7359 (2019).
12. Vollmers, J., Wiegand, S. & Kaster, A.-K. Comparing and evaluating metagenome assembly tools from a microbiologist's perspective-not only size matters! *PLoS ONE* **12**, e0169662 (2017).
13. Nayfach, S. et al. A genomic catalog of earth's microbiomes. *Nat. Biotechnol.* **39**, 499–509 (2021).
14. Almeida, A. et al. A new genomic blueprint of the human gut microbiota. *Nature* **568**, 499–504 (2019).
15. Jennifer Mattock, M. W. A comparison of single-coverage and multi-coverage metagenomic binning reveals extensive hidden contamination. *Nat. Methods* **20**, 1170–1173 (2023).
16. Bowers, R. M. et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nat. Biotechnol.* **35**, 725–731 (2017).
17. Rinke, C. et al. Insights into the phylogeny and coding potential of microbial dark matter. *Nature* **499**, 431–437 (2013).
18. Parks, D. H. et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nat. Microbiol.* **2**, 1533–1542 (2017).
19. Nayfach, S., Shi, Z. J., Seshadri, R., Pollard, K. S. & Kyrpides, N. C. New insights from uncultivated genomes of the global human gut microbiome. *Nature* **568**, 505–510 (2019).
20. Vollmers, J., Wiegand, S., Lenk, F. & Kaster, A.-K. How clear is our current view on microbial dark matter? (Re-) Assessing public MAG & SAG datasets with MDMcleaner. *Nucleic Acids Res.* **50**, e76–e76 (2022).
21. Drillon, G., Champeimont, R., Oteri, F., Fischer, G. & Carbone, A. Phylogenetic reconstruction based on synteny block and gene adjacencies. *Mol. Biol. Evol.* **37**, 2747–2762 (2020).
22. Periwal, V. & Scaria, V. Insights into structural variations and genome rearrangements in prokaryotic genomes. *Bioinformatics* **31**, 1–9 (2015).
23. Sczyrba, A. et al. Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nat. Methods* **14**, 1063–1071 (2017).
24. Orakov, A. et al. GUNC: detection of chimerism and contamination in prokaryotic genomes. *Genome Biol.* **22**, 178 (2021).
25. 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).
26. Radford, A. et al. Learning transferable visual models from natural language supervision. In *Proc. 38th International Conference on Machine Learning* **139**, 8748–8763 (PMLR, 2021).
27. Wagstaff, K. et al. Constrained *k*-means clustering with background knowledge. In *Proc. 18th International Conference on Machine Learning* **1**, 577–584 (Morgan Kaufmann, 2001).
28. 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).
29. Ma, B. et al. A genomic catalogue of soil microbiomes boosts mining of biodiversity and genetic resources. *Nat. Commun.* **14**, 7318 (2023).
30. Duncan, A. et al. Metagenome-assembled genomes of phytoplankton microbiomes from the arctic and atlantic oceans. *Microbiome* **10**, 67 (2022).
31. Faist, H. et al. Potato root-associated microbiomes adapt to combined water and nutrient limitation and have a plant genotype-specific role for plant stress mitigation. *Environ. Microbiome* **18**, 18 (2023).
32. Tláskal, V. et al. Metagenomes, metatranscriptomes and microbiomes of naturally decomposing deadwood. *Sci. Data* **8**, 198 (2021).
33. Buck, M. et al. Comprehensive dataset of shotgun metagenomes from oxygen stratified freshwater lakes and ponds. *Sci. Data* **8**, 131 (2021).
34. Kavagutti, V. S. et al. High-resolution metagenomic reconstruction of the freshwater spring bloom. *Microbiome* **11**, 15 (2023).
35. Maestre-Carballa, L., Navarro-López, V. & Martinez-Garcia, M. City-scale monitoring of antibiotic resistance genes by digital pcr and metagenomics. *Environ. Microbiome* **19**, 16 (2024).
36. Zhao, L. et al. A clostridia-rich microbiota enhances bile acid excretion in diarrhea-predominant irritable bowel syndrome. *J. Clin. Invest.* **130**, 438–450 (2020).
37. Rodriguez-R, L. M. & Konstantinidis, K. T. Nonpareil: a redundancy-based approach to assess the level of coverage in metagenomic datasets. *Bioinformatics* **30**, 629–635 (2014).
38. Lai, S. et al. metamic: reference-free misassembly identification and correction of de novo metagenomic assemblies. *Genome Biol.* **23**, 242 (2022).
39. Derakhshani, H., Bernier, S. P., Marko, V. A. & Surette, M. G. Completion of draft bacterial genomes by long-read sequencing of synthetic genomic pools. *BMC Genomics* **21**, 519 (2020).
40. Mende, D. R. et al. progenomes2: an improved database for accurate and consistent habitat, taxonomic and functional annotations of prokaryotic genomes. *Nucleic Acids Res.* **48**, D621–D625 (2020).

41. Chaumeil, P.-A., Mussig, A. J., Hugenholtz, P. & Parks, D. H. GTDB-Tk v2: memory friendly classification with the genome taxonomy database. *Bioinformatics* **38**, 5315–5316 (2022).
42. Li, D., Liu, C.-M., Luo, R., Sadakane, K. & Lam, T.-W. Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. *Bioinformatics* **31**, 1674–1676 (2015).
43. Li, K. et al. Uniformer: unified transformer for efficient spatiotemporal representation learning. Preprint at https://doi.org/10.48550/arXiv.2201.04676 (2022).
44. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
45. Tan, M. & Le, Q. Efficientnet: rethinking model scaling for convolutional neural networks. In *Proc. 36th International Conference on Machine Learning* **97**, 6105–6114 (PMLR, 2019).
46. Li, C., Zhou, A. & Yao, A. Omni-dimensional dynamic convolution. Preprint at https://doi.org/10.48550/arXiv.2209.07947 (2022).
47. Guo, M.-H., Lu, C.-Z., Liu, Z.-N., Cheng, M.-M. & Hu, S.-M. Visual attention network. *Comput. Vis. Media* **9**, 733–752 (2023).
48. Wang, H. et al. Deepnet: scaling transformers to 1,000 layers. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 6761–6774 (2024).
49. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. Preprint at https://doi.org/10.48550/arXiv.1708.02002 (2018).
50. Olm, M. R., Brown, C. T., Brooks, B. & Banfield, J. F. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *ISME J.* **11**, 2864–2868 (2017).
51. Zou, B. Deepurify: a multi-modal deep language model to remove contamination from metagenome-assembled genomes. Simulation 1, v.1. *Zenodo* https://doi.org/10.5281/zenodo.8343497 (2023).
52. Zou, B. Deepurify: a multi-modal deep language model to remove contamination from metagenome-assembled genomes. Simulation 2, v.2. *Zenodo* https://doi.org/10.5281/zenodo.8343505 (2024).
53. Zou, B. A deep multi-modal deep language model for contaminant removal from metagenome-assembled genomes (code). *Zenodo* https://doi.org/10.5281/zenodo.11919065 (2024).

Acknowledgements

The design of the study and the collection, analysis and interpretation of the data were partially supported by the Young Collaborative Research grant (no. C2004-23Y), HMRF (grant no. 11221026), the open project of BGI-Shenzhen, Shenzhen 518000, China (grant no. BGIRSZ20220014) and HKBU Start-up Grant Tier 2 (grant no. RC-SGT2/19-20/SCI/007). We also thank the BGI Research-Shenzhen, the Research Committee of Hong Kong Baptist University, and the Interdisciplinary Research Clusters Matching Scheme for their kind support of this project.

Author contributions

L.Z. conceived the study. B.Z. designed and implemented the Deepurify algorithms. L.Z. and B.Z. conceived the experiments. B.Z., Y.D. and Z.Z. conducted the experiments. B.Z. and J.W. analysed the results. B.Z. and L.Z. wrote the paper. Y.H. and X.F. revised the paper and supported the project. K.C.C. and S.S. contributed computational resources. All authors reviewed the paper.

Competing interests

K.C.C. and S.S. are the employees of Nvidia Corporation (NVIDIA). The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00908-5.

Correspondence and requests for materials should be addressed to Lu Zhang.

Peer review information *Nature Machine Intelligence* thanks Antonio Pedro Camargo and Luis Coehlo for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Lu Zhang

Last updated by author(s): Aug 30, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	No software was used for data collection.
Data analysis	Our study used MAGpurify (v2.1.2) and MDMcleaner (v0.8.7) for MAG decontamination. The development of Deepurify was developed using Python (v3.8.18) along with PyTorch (v2.0.0 + cu118). The SCGs calling was executed using Prodigal (v2.6.3) and HMMER (v3.3.2). The computation of the balanced macro F1-score was performed using Scikit-Learn (v1.2.0). The evaluation of MAG quality was carried out using CheckM2 (v1.0.1). For binning, we employed CONCOCT (v1.1.0), MetaBAT2 (v2.15), and SemiBin2 (v2.1.0). The annotation of MAGs was executed using GTDB-Tk (v1.4.0). In this study, metaSPAdes (v3.15.0) and MegaHit (v1.2.9) were applied for assembly. We applied MMseqs2 (v14.7e284) to cluster sequences. The GitHub link for this study is `https://github.com/ericcombiolab/Deepurify/' .

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The microbial representative genomes and their associated taxonomic lineages were downloaded from the proGenomes v2.1 database.
The genome taxonomy datatbase (GTDB) r202 was used to annotate the reference genomes.
The simulation 1 data have been uploaded to https://zenodo.org/record/8343498.
The simulation 2 data have been uploaded to https://zenodo.org/records/11608439.
The CAMI I short-reads were downloaded from `1st CAMI Challenge Dataset 1 CAMI_low', `1st CAMI Challenge Dataset 2 CAMI_medium' and `1st CAMI Challenge Dataset 3 CAMI_high' from https://data.cami-challenge.org/participate/.
The NCBI SRA accessions of 7 soil samples are SRR25158210, SRR25158221, SRR25158244, SRR25158253, SRR25158281, SRR25158363, and SRR25158536;
The NCBI SRA accessions of the 3 freshwater samples are ERR4195020, ERR9631077, and SRR26420192;
The NCBI SRA accessions of the 3 plant samples are SRR10968246, SRR14308228, and SRR14308230.
The JGI Project Id of 11 ocean samples are 1021520, 1021523, 1021526, 1102218, 1102220, 1102222, 1102224, 1102232, 1102234, 1125692, 1125694.
The human fecal metagenomic sequencing reads of the IBS-D cohort were downloaded from China National GeneBank (CNGB) with accession number CNPO000334.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No calculation of sample sizes were made. The sample sizes for CAMI were predetermined by the public datasets. We conducted at least three replications for each scenario to demonstrate the effectiveness of our method in practical applications. We used the available public datasets: CAMI_low (n= 1), CAMI_medium (n = 2), CAMI_high (n=5, merged into 1 file), soil (n= 7), plant (n=3), freshwater (n=3), ocean (n=11), and human feces (n=227).
Data exclusions	N/A
Replication	Findings are deterministic with given data.
Randomization	For all experiments, the participants were randomly chosen.
Blinding	The Investigators were not blinded to allocation during all experiments and all outcome assessment. No different treatments were given to different participants during experiments and outcome assessment. Therefore, blinding was not relevant to our study

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging