Deepurify: a multi-modal deep language model to remove contamination from metagenome-assembled genomes

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], and Lu Zhang[1,5*]

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

[2]BGI Research, Shenzhen 518083, China

[3]BGI Research, Sanya 572025, China

[4]NVIDIA AI Technology Center, NVIDIA

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

[*]To whom correspondence should be addressed: ericluzhang@hkbu.edu.hk

Abstract

Metagenome-assembled genomes (MAGs) offer valuable insights into the exploration of microbial dark matter using metagenomic sequencing data. However, there is a growing concern that contamination in MAGs may significantly impact the downstream analysis results. Existing MAG decontamination methods heavily rely on marker genes but do not fully leverage genomic sequences. To address the limitations, we have introduced a novel decontamination approach named Deepurify, which utilizes a multi-modal deep language model employing contrastive learning to learn taxonomic similarities of genomic sequences. Deepurify utilizes inferred taxonomic lineages to guide the allocation of contigs into a MAG-separated tree and employs a tree traversal strategy for maximizing the total number of medium- and high-quality MAGs. Extensive experiments were conducted on two simulated datasets, CAMI I, and human gut metagenomic sequencing data. These results demonstrate that Deepurify significantly outperforms other decontamination methods.

Introduction

Short-read metagenomic sequencing has gained popularity in investigating unculturable microbial genomes [1, 2, 3, 4], but single contigs assembled by short-reads often lead to fragmented and incomplete microbial genomes [5, 6, 7]. Several contig binning tools [8, 9, 10, 11] have been developed to group contigs into metagenome-assembled genomes (MAGs) based on their abundances and sequence contexts to represent microbial genomes. Several studies [12, 13, 14] claimed the qualities of those MAGs were comparable to the genomes from microbial isolates, but there has been a growing concern that contamination may seriously impact the qualities of MAGs [15]. MAG contamination refers to a mixture of contigs from different microbes in the same MAG and those chimeric MAGs would substantially reduce the reliability of downstream ecological and evolutionary analyses. Bowers et al. [16] suggested eliminating the MAGs with more than 10% contamination, but many microbes from MAGs with marginal contamination would be missed. In our preliminary study, we observed a considerable number of MAGs would be removed due to their marginal contamination values, even for some high-abundance MAGs (**Supplementary Note** 1). This may result in the loss of a significant number of MAGs for subsequent downstream analysis.

Several tools [17, 18, 19, 20] have been developed to identify and remove the potentially contaminated contigs from chimeric MAGs based on marker genes and the sequence characteristics from known species. Two pipelines [17, 18] published several years ago are no longer actively supported and have not been widely accepted by the community. More recent and actively supported tools are MAGpurify [19] and MDMcleaner [20]. MAGpurify was recently developed for MAG decontamination using

three sources of information: the phylogenetic or clade-specific marker genes, the GC contents, and tetranucleotide frequencies of contigs. MDMcleaner utilizes marker genes (coding, 16S, and 23S rRNA genes) to predict the taxonomic classification of contigs. The contig taxonomies are determined by the taxonomic Least Common Ancestor (LCA) of the involved marker genes and any contigs that have different annotations with the dominating taxon of the MAG would be removed.

Although MAGpurify and MDMcleaner show promising results, they suffer from several issues that hinder their widespread applications in various scenarios. First, both of them need to align marker genes/contigs to the reference databases and this approach is inapplicable to novel microorganisms. As previously observed, it has been noted that the reference genomes available in RefSeq (117,030 as of March 11, 2022) only account for less than 5.319% of all species [21]. In addition, the alignment is time-consuming even if the built-in databases are optimized (**Supplementary Note** 2). Second, previous study [22] has pointed out that many factors have the potential to reduce the performance of alignment-based tools on phylogenetic analysis, such as sequence misalignment, false-orthologous assignment, gene duplication or loss events, horizontal gene transfer, and the presence of homoplasy, etc. Third, various genomic alterations, including genomic variations, alterations in gene order, and genome rearrangements, among others, have been identified as factors enhancing the resolution and reliability of differentiating the genomic sequences from different species [23]. These forms of evidence can offer invaluable insights that are uniquely attainable through whole genome sequences. Fourth, we found a majority of contamination in MAGs occurred at the genus and species levels (**Supplementary Note** 3) and both MAGpurify and MDMcleaner demonstrated poor performance at these low taxonomic ranks (**Supplementary Note** 4).

In this study, we developed Deepurify for MAG decontamination with high resolution and generalization using a multi-modality deep language model. In the training procedure, Deepurify developed two distinct encoders, a genomic sequence encoder (GseqFormer, **Methods**) and a taxonomic encoder (Long short-term memory, LSTM) to encode genomic sequences and their source genomes' taxonomic lineages, respectively. Next, Deepurify learned their relationships in different taxonomic ranks using contrastive training (Figure 1). In the decontamination process, Deepurify initially quantified the taxonomic similarities of contigs by assigning taxonomic lineages to them (Figure 2 **a**). It then used these lineages to construct a MAG-separated tree, partitioning the MAG into distinct sections, each containing contigs with the same lineage (Figure 2 **c**). This approach optimized contig utilization within the MAG, avoiding immediate removal of contaminated contigs. It was especially effective for MAGs with high contamination rates. Lastly, a tree traversal algorithm was devised to maximize the count of medium- and high-quality MAGs within the MAG-separated tree (Figure 2 **d**).

We observed that Deepurify outperformed two state-of-the-art tools MAGpurify and MDMcleaner

in simulated, CAMI I challenge (high, medium1, medium2, and low) [24] and human gut metagenomic sequencing data [25, 26]. For simulated data, chimeric MAGs were created by mixing the sequences from two microbial genomes at various taxonomic ranks, with contamination rates from 5% to 20%. Deepurify achieved balanced macro F1-scores almost twice as high as that of MAGpurify across all taxonomic ranks and 1.5 times higher than that of MDMcleaner at the genus and species ranks (Figure 3, **Supplementary Table** 3) on average. Additionally, Deepurify demonstrated outstanding generalization capabilities, where it achieved excellent accuracy in identifying contaminated contigs even if their source genomes were absent from the training set (Figure 4, **Supplementary Table** 4). For CAMI I and a human gut metagenomic sequencing dataset, $S1$ [25], we applied Deepurify to the results of four mainstream contig binning tools (VAMB [8], CONCOCT [9], MetaBAT2 [11], and MaxBin [10]), and the results showed that it could substantially improve MAG quality, surpassing both the MAGpurify and MDMcleaner for all binning tools. Next, we applied Deepurify to a large metagenomic sequencing dataset derived from a diarrhea-predominant Irritable Bowel Syndrome (IBS-D) cohort, including 290 patients and 89 healthy controls [26]. We found Deepurify could rescue 70.12% highly contaminated MAGs (completeness $\geq$ 50% and contamination $\geq$ 25%) to medium- (completeness $\geq$ 50% and contamination $\leq$ 10%) and high-quality (completeness $\geq$ 90% and contamination $\leq$ 5%) MAGs. The corresponding percentages of MAGpurify and MDMcleaner were only 1.4% and 0.7%, respectively. Moreover, we compared the annotation of these MAGs before and after MAG decontamination and identified five new species (**Supplementary Table** 5) and one new genus (**Supplementary Table** 6). Among them, one of the species demonstrated a suggestive association with IBS-D.

Results

Deepurify architecture and decontamination workflow

Deepurify was a multi-modal deep language model developed specifically to remove contaminated contigs from a MAG. Figure 1 **b** and Figure 2 depict the fundamental architecture and decontamination workflow of Deepurify. Its architecture resembles that of CLIP [27], a well-established multi-modal model incorporating two encoders designed to process data from two modalities: 1). GseqFormer, for encoding genomic sequences, and 2). LSTM, for encoding taxonomic lineages (**Methods**). During training, we utilized contrastive learning to empower Deepurify to distinguish between real (positive) and fake (negative) taxonomic lineages of a sequence (Figure 1). This distinction is based on the cosine similarity between normalized encoded sequences and both positive and negative normalized encoded lineage vectors. Positive encoded lineages should exhibit higher cosine similarity with encoded sequences compared to negative ones. During the decontamination process (Figure 2), Deepurify first

assessed the taxonomic similarities of contigs by computing cosine similarity scores between the contigs and lineages in the taxonomic tree. Subsequently, it assigned the lineages to the contigs based on the highest similarity (Figure 2 **a**). Deepurify devised a scheme involving the construction of a MAG-separated tree to maximize the effective utilization of contigs, without directly discarding contaminated ones within a MAG (Figure 2 **c**). This scheme was especially valuable in MAGs with high contamination rates. The MAG-separated tree partitioned contigs within the MAG into distinct branches according to their predicted taxonomic lineages across multiple taxonomic ranks. Each node in the tree contains contigs sharing the same taxon at that rank. Deepurify identified and applied single-copy genes (SCGs) to each node to prevent duplication of SCGs within it. Finally, Deepurify applied CheckM [28] to each node of the tree and employed a depth-first search (DFS) algorithm to traverse the MAG-separated tree to maximize the count of high- and medium-quality MAGs (**Methods**; Figure 2 **d**).

Development of simulated testing sets

We generated two simulated testing sets, SIM_1 and SIM_2, to evaluate Deepurify's capability in distinguishing between core and contaminated contigs within a chimeric MAG. The SIM_1 testing set assessed Deepurify's decontamination performance when the source genomes of both core and contaminated contigs were part of the training set. Conversely, the SIM_2 testing set evaluated its performance when the source genomes of the contigs were either included or excluded from the training set. A simulated chimeric MAG primarily consisted of core contigs, with a minority being contaminated. We referred to the source genomes of core contigs as "core" genomes and the source genomes of contaminated contigs as "contaminated" genomes.

In our chimeric MAG simulation, we simulated contamination occurring at different taxonomic ranks by randomly selecting core and contaminated genomes from two species at varying taxonomic distances on the taxonomic tree. The LCAs of these two species' lineages ranged from kingdom to genus (lineages differ starting from phylum to species). Each simulated MAG consisted of 200 contigs, with lengths distributed uniformly between 1,000 bps and 8,192 bps. We generated 50 simulated MAGs for different contamination proportions (5%, 10%, 15%, and 20%) at each taxonomic rank of LCA.

The test set of SIM_1 was generated using the genomes that were all included in its training set GS_c (**Methods**). For SIM_2, its training set GS_p (**Methods**) lacked either core or contaminated genomes, resulting in four scenarios for simulation: 1. both core and contaminated genomes included in the GS_p (SIM_2^1); 2. only core genomes included in the GS_p (SIM_2^2); 3. only contaminated genomes included in the GS_p (SIM_2^3); 4. both core and contaminated genomes were not included the GS_p (SIM_2^4). To address the imbalance issue between the number of core and contaminated contigs in a simulated MAG, we utilized a balanced macro F1-score to evaluate the performance of MAGpurify, MDMcleaner,

and Deepurify.

Deepurify has superior purification performance on SIM_1

We applied MAGpurify, MDMcleaner, and Deepurify to SIM_1 testing set and we observed that Deepurify outperformed MAGpurify significantly across all taxonomic ranks and contamination proportions (Figure 3, **Supplementary Table** 3). Compared to MAGpurify, Deepurify increased the overall averaged F1-score by 45.18% (phylum), 76.75% (class), 80.53% (order), 89.75% (family), 90.51% (genus), and 78.02% (species) across different contamination proportions. We observed that Deepurify and MDMcleaner performed comparably when the lineages of core and contaminated genomes differed at higher taxonomic ranks such as phylum, class, and order. However, Deepurify exhibited significant improvements when the differences in lineages began at the family, genus, and species ranks, with an overall average F1-score increase of 8.45% (family), 40.54% (genus), and 63.72% (species) compared to MDMcleaner. This fact suggested Deepurify could be more efficient to be applied in real metagenomic sequencing data, as most of the MAG contamination was found to exist at the genus and species (**Supplementary Note** 3). We noticed that the F1-scores of MAGpurify, Deepurify, and MDMcleaner decreased as the taxonomic ranks became lower. This could be due to the higher proportion of homologous sequences between the core and contaminated genomes at the genus and species taxonomic ranks.

We also observed the standard deviations (SD) of the F1-scores of Deepurify were considerably lower than those of MAGpurify and MDMcleaner suggesting Deepurify was more robust regardless of the sources of contamination. On the one hand, the SD of F1-scores of MAGpurify were consistently reduced at taxonomic ranks from high to low, revealing it is more conservative to remove contigs at low taxonomic ranks. Consequently, it may not effectively remove contaminated contigs when contamination occurs at these lower taxonomic ranks. On the other hand, the SD of F1-scores of MDMcleaner were the highest at genus and species ranks, indicating that it was not stable in accurately distinguishing between genomes with homologous sequences. Furthermore, we observed an opposite trend between the contamination rates and the average F1-score of MAGpurify. This indicates that MAGpurify was not able to eliminate contaminated contigs at high rates of contamination efficiently. Although MDMcleaner's performance remained relatively stable across different contamination rates, it experienced a significant decline as taxonomic ranks decreased. Deepurify emerged as the most efficient and robust model across all tested conditions.

Deepurify has strong generalization ability for novel microbes

As MAGpurify and MDMcleaner did not provide any interface to allow users to rebuild their databases, we could not evaluate them on the SIM_2 testing set. We applied Deepurify on SIM_2 and we found that the F1-scores of Deepurify were only marginally reduced regardless of core or contamination genomes absent from the training sets (Figure 4, **Supplementary Table** 4). We used the performance of Deepurify on SIM_2^1 (all genomes were included in the training set) as the baseline. In SIM_2^2 (contaminated genomes excluded in the training set), the F1-score reduction from phylum to species rank was the smallest, with only a 1.07% decrease at the phylum rank and a 17.4% decrease at the species rank. Conversely, SIM_2^4 (both core and contaminated genomes were excluded in the training set) exhibited the greatest reduction, with a 19.65% decrease in F1-score at the phylum rank and a 24.48% decrease at the species rank. The observations aligned with our expectations since understanding the sequences' pattern of the core genomes was essential for the purification of MAGs. Furthermore, we noted a slight decrease of merely 1.07% and 6.81% in the F1-scores for SIM_2^2 when the lineages were different from phylum to family. Nonetheless, a substantial disparity of 11.84% for genus and 17.14% for species was observed. This finding indicates that Deepurify exhibited greater efficacy in removing contamination when it occurs at higher ranks, irrespective of their inclusion in the training set. In contrast, addressing contamination at lower taxonomic ranks proved to be more challenging due to the increased presence of homologous sequences.

The impact of homologous sequences and contig length on MAG decontamination

For a simulated MAG, we defined the contigs as derived from homologous sequences if they could be aligned to both core and contaminated genomes (**Methods**). In the test set of SIM_1, we identified contigs from homologous sequences at various taxonomic ranks: 142 at phylum, 832 at class, 3,015 at order, 4,429 at family, 8,048 at genus, and 17,169 at species. The number of contigs from homologous sequences increased from phylum to species, which could explain the reason for the performance declination of MAGpurify, MDMcleaner, and Deepurify if contamination derived from the LCAs of genomes at low taxonomic ranks.

Furthermore, we categorized the contigs based on their lengths (intervals of 1,000 bps) to assess the influence of contigs' length on the performance of Deepurify. Deepurify showed better performance on long contigs compared to short ones (**Supplementary Figure** 8, **Supplementary Table** 7) probably because long contigs could provide more information on the genomic context of their source genomes.

Deepurify improves the qualities of MAGs from different contig binning tools

We applied MAGpurify, MDMcleaner, and Deepurify to the MAGs generated by MaxBin, MetaBAT2, VAMB, and CONCOCT to examine if they could increase the number of medium- and high-quality MAGs. The contigs of CAMI I and human gut metagenomic sequencing ($S1$) were downloaded from our previous study [25] (**Methods**). We used two criteria to evaluate the performance of MAG decontamination: 1. the increased number of medium- (INM_{mq}) and high-quality MAGs (INM_{hq}); 2. the improved quality score (IQS), which measures the overall MAG quality improvement (**Methods**).

Deepurify consistently outperformed the other two tools in nearly all datasets and contig binning methods (Figure 5, **Supplementary Table** 8). On average, across all datasets and binning methods, Deepurify exhibited 2.87-fold (1.33-fold) and 5.15-fold (4.16-fold) higher mean value for the INM_{hq} and INM_{mq} compared to MAGpurify (MDMcleaner). Interestingly, the INM_{hq} and INM_{mq} values of Deepurify for all binning tools were commonly positive except for VAMB on the high-complexity community in CAMI I (VAMB does not work well on a single sample) and the values for MAGpurify and MDMclean were more frequently to be observed negatively. In Figure 6, we depict the completeness and contamination rates of MAGs, before and after purification with MAGpurify, MDMcleaner, and Deepurify, using data from CAMI I and $S1$. We also employ a generalized additive model to create a smooth curve, effectively capturing the contamination trends within these MAG datasets. It was observed that Deepurify consistently outperformed the others by having the smallest areas under the curve.

Deepurify demonstrated remarkable performance superiority over MAGpurify (IQS: 29.21-fold on average for all cases) and MDMcleaner (IQS: 1.82-fold on average for all cases), especially on the binning results of CONCOCT, VAMB, and MetaBAT2 (Figure 7, **Supplementary Table** 9). These observations suggested that Deepurify was more effective in improving contig binning performance than other tools as many low-quality MAGs were able to be upgraded to medium- or high-quality MAGs.

Deepurify outperforms other purification tools on real-world data

We further applied Deepurify to the human gut metagenomic sequencing data from 290 IBS-D patients and 89 healthy controls [26]. The sequencing data were assembled by metaSPAdes [5] followed by contig binning using MetaBAT2 (**Methods**), which generated 4,887 high-quality and 5,943 medium-quality MAGs. We selected 713 MAGs with high contamination (completeness $\geq$ 50% and contamination $\geq$ 25%) to evaluate the efficacy of Deepurify on MAG decontamination. Our examination revealed

that MAGpurify and MDMcleaner could enhance the quality of only a small fraction of these highly contaminated MAGs. Specifically, MAGpurify improved 1.4% of them to high- and medium-quality MAGs ($INM_{hq} = 1$, $INM_{mq} = 9$), while MDMcleaner improved 0.7% of them to high- and medium-quality MAGs ($INM_{hq} = 1$, $INM_{mq} = 4$). Deepurify demonstrated a remarkable ability for MAG decontamination, as it was able to rescue a significant proportion of these MAGs, 70.12% of them ($INM_{hq} = 3$, $INM_{mq} = 497$). Deepurify demonstrated a significantly elevated IQS at 248994.46, surpassing both MAGpurify, which has an IQS of 17772.28, and MDMcleaner, with an IQS of 14466.47. The contamination rates of these MAGs were mostly reduced to below 10% after undergoing MAG decontamination using Deepurify whereas the values obtained from the other tools were considerably higher (Figure 8 **a**).

Deepurify identified novel IBS-D association signals

We examined all high-quality (4,931) and medium-quality (6,539) MAGs obtained from the IBS-D cohort after Deepurify decontamination to identify novel association signals. We utilized GTDB-TK [29] (**Methods**) to annotate these MAGs both before and after Deepurify's purification process. Upon comparing the MAG annotation results, we identified five new species (**Supplementary Table** 5) and one new genus (**Supplementary Table** 6). We performed an association analysis of IBS-D on the 678 MAGs, which were initially categorized as low-quality but were reclassified as medium- or high-quality after decontamination (**Methods**). This analysis identified several suggestive signals (P-value < 0.05) including one novel species (s__Collinsella sp900541055), and two confirmed species (Alistipes [30] and Ruminococcus gnavus [31, 32]) that were known to be associated with IBS-D. Lastly, we showed the completeness and contamination rates for all MAGs in the IBS-D cohort before and after purification by Deepurify in Figure 8 **b**. This plot demonstrated Deepurify's remarkable ability to purify contaminated contigs in MAGs.

Discussion

Utilizing genome assembly with short-read metagenomic sequencing data has become a prevalent method to decipher microbial compositions in complex environments. However, each assembled metagenomic contig only partially represents a microbial genome. It is therefore crucial to perform contig binning to obtain contig sets with similar genomic characteristics and abundances, which then represent MAGs that originate from the same microbe. As was highlighted in a recent paper [15], MAG contamination is a significant stumbling block during contig binning on single sample assembly. Decontamination tools, such as MAGpurify and MDMcleaner, have been developed to address

the challenge of eliminating contaminated contigs from MAGs. Nonetheless, these tools demonstrate several limitations. Most notably, they are ineffective in distinguishing contigs from each other if their core and contaminated genomes belong to the same family or genus. Furthermore, these tools are unable to process contigs whose source genomes are absent from their built-in databases. And thirdly, they mainly focus on genes, with genomic variations such as gene order and genome rearrangements being left out of consideration.

To address these limitations, we developed Deepurify, a novel tool that uses deep language models to learn the relationship between microbial genomes and their taxonomic lineages. Deepurify models all nucleotides in sequences and can learn local genomic alternations between adjacent species in the training set. This approach allows Deepurify to handle contigs without known genes. Deepurify has a superior decontamination capacity, particularly if the contigs share a high proportion of homologous sequences (**Supplementary Note** 4). It also outperformed the existing tools if the source genomes of contigs were not included in the training dataset. Deepurify could significantly speed up the MAG decontamination procedure with GPU acceleration (**Supplementary Note** 2), which allows scaling of decontamination to large numbers of MAGs. The primary runtime bottleneck for Deepify lies in the duration required for running CheckM, which is nearly twice as long as inferring lineages for the contigs within MAGs. The efficiency of Deepurify's execution could be significantly improved with a method to expedite the CheckM runtime.

Deepurify adopts a unique approach to optimize the utilization of contigs within a MAG. Instead of adopting the common practice of directly discarding contaminated contigs, Deepurify constructs a MAG-separated tree for filtering. This innovative strategy proves especially advantageous in scenarios where MAGs exhibit a substantial degree of contamination, typically exceeding a contamination rate of 100%. Deepurify has the ability to resolve a highly contaminated MAG into two separate MAGs, typically falling within the high- or medium-quality range. On occasion, it may yield three or more MAGs that hold potential for further utilization.

Our experiments demonstrated the remarkable efficacy of Deepurify in decontaminating MAGs from short-read assembly. We hold a strong belief that its applicability extends to contigs derived from long-read assemblies, accompanied by two distinct advantages: Firstly, contigs derived from long-read assemblies are significantly longer than those from short-read assemblies. It offers Deepurify a substantially enriched sequence context, thereby enhancing its capacity for decontamination. Secondly, single-base substitutions and indel errors are frequently observed in long-read assemblies [33], which we placed emphasis on during the development of Deepurify's training procedure (**Supplementary Note** 6). It is worth mentioning that contemporary decontamination tools do not typically consider sequence noise.

On the other hand, it is important to note that Deepurify cannot deal with overly large misassemblies in contigs (such as chimeric contigs, translocations, etc.). We observed that for some MAGs Deepurify failed to achieve its specified decontamination standard because a designated single-copy gene was detected multiple times. Chimeric contigs may therefore remain a challenge to Deepurify since they could substantially influence the local context of sequences, which may adversely impact the quantification of taxonomic similarity between contigs in a MAG. To mitigate the influence of such misassemblies, we recommend that users apply assembly error correction tools such as metaMIC [34] prior to using Deepurify.

Methods

Preparing and processing microbial reference genomes

We downloaded microbial representative genomes and their taxonomic lineages from proGenomes v2.1 database [35] to generate two training sets GS_c, GS_p for model training and two simulated testing sets SIM_1 and SIM_2 for evaluating. We excluded the microbial genomes without phylum annotations or if the phyla they belonged to had less than 15 species. For microbes with only phylum and species annotations, all other taxonomic ranks inbetween were annotated as "Unclassified".

Training sets construction

After data preprocessing, we generated two training sets for SIM_1 and SIM_2: 1. a complete reference genome training set (GS_c) consisting of the genomic sequences from 10,332 species belonging to 37 phyla (**Supplementary Table** 10); 2. a partial reference genome training set (GS_p) by randomly selecting 112 species, which come from 12 phyla (**Supplementary Table** 11) in GS_c. GS_p was used to evaluate the performance of Deepurify when either core or contaminated genomes were not included in the training set.

During the training stage, we sampled the contig-sized sequences from the genomes in GS_c and GS_p. The sequence lengths ranged from 1,000 bps to 8,192 bps, following a pre-defined contig length distribution learned from a real metagenomic assembly exercise (**Supplementary Note** 5). We randomly incorporated into these sequences insertions, deletions, and single nucleotide variants (**Supplementary Note** 6) in order to reduce the impact of sequencing errors and enhance model generalization capabilities.

MAG generation for SIM_1 and SIM_2

We simulated chimeric MAGs for use in SIM_1 and SIM_2 sets for evaluation: 1. For SIM_1, all source genomes of contigs were included in GS_c; 2. For SIM_2, some source genomes of contigs might be absent from GS_p. For SIM_1, we randomly selected the genomes of two distinct lineages (SP_1 and SP_2) in GS_c and simulated 200 contigs from them with lengths between 1,000 bps and 8,192 bps and with varying proportions of contigs from SP_2 (5%, 10%, 15%, and 20%) for each MAG. The lineage LCAs of SP_1 and SP_2 were traversed from kingdom to genus (lineages differ starting from phylum to species). We generated 50 MAGs for each mixture proportion and on each taxonomic rank of LCA. For SIM_2, we followed a similar chimeric MAG simulation procedure as we did for SIM_1, the only difference being that SP_1 and SP_2 may be extracted from either GS_p or from $GS_c - GS_p$. There are four permuted scenarios for SIM_2: 1. both core and contaminated genomes are included in the GS_p (SIM_2^1); 2. only core genomes are included in the GS_p (SIM_2^2); 3. only contaminated genomes are included in the GS_p (SIM_2^3); 4. neither core nor contaminated genomes are included in the GS_p (SIM_2^4).

Generate MAGs from contig binning tools

We downloaded the metagenomic sequencing datasets from CAMI I with low, medium (two datasets), and high complexity and from a human stool sample (S_1) [25]. The contigs of these datasets were assembled by metaSPAdes with default parameters. Contigs were grouped as MAG using VAMB (contig length $>$ 1kbps), CONCOCT (contig length $>$ 1kbps), MaxBin (contig length $>$ 1kbps), and MetaBAT2 (contig length $>$ 1.5kbps). We only kept MAGs from VAMB with a completeness of at least 50% to exclude the MAGs with few contigs (e.g. $<$ 3 contigs).

MAG quality definitions

MAGs are typically classified into distinct quality categories based on their degrees of completeness and contamination. High-quality MAGs are defined by completeness levels equal to or exceeding 90% and contamination levels at or below 5%. Medium-quality MAGs are characterized by completeness levels of 50% or higher, with contamination levels below or equal to 10%. MAGs failing to meet the high or medium-quality criteria are categorized as low-quality.

IBS-D real-world validation study

We applied metaSPAdes with default parameters to assemble short-read metagenomic sequencing data from 290 IBS-D patients and 89 healthy controls. The contigs longer than 1.5kb were grouped into

MAGs by MetaBAT2. We evaluated MAGpurify, MDMcleaner, and Deepurify on the MAGs with completeness $\geq 50\%$ and contamination $\geq 25\%$.

Microbial taxonomic annotation

We used GTDB-Tk [29] to annotate and allocate MAGs to the taxonomic tree. A MAG would be annotated as a particular species (g_{ref} represents its genome) if 1. its average nucleotide identity with g_{ref} is no less than 95% and 2. its alignment fraction against g_{ref} is no less than 65%.

Identification of homologous sequences

We conducted BLASTN alignments between the core contigs and the contaminated genomes, as well as between the contaminated contigs and the core genomes. Contigs with an E-value less than $1e^{-6}$ were considered aligned and would be categorized as sequences derived from homologous sequences.

Metrics for performance evaluation

For simulated chimeric MAGs, we applied a balanced macro F1-score to evaluate the performance of MAG decontamination to mitigate the influence of unbalanced numbers of contigs from SP_1 and SP_2. For the binned MAGs that were generated from CAMI I, S_1 and the IBS-D cohort, we adopted two criteria to evaluate the improvement of MAG qualities: 1. total increased number of high- (completeness $\geq 90\%$ and contamination $\leq 5\%$) and medium-quality (completeness $\geq 50\%$ and contamination $\leq 10\%$) MAGs; 2. increased quality score (IQS)

$$\begin{aligned} IQS &= \sum_{i}^{n_p} QS_{p,i} - \sum_{j}^{n_q} QS_{q,i} \\ &= \sum_{i}^{n_p} (CN_{p,i} - 5 \times CT_{p,i}) - \sum_{j}^{n_q} (CN_{q,j} - 5 \times CT_{q,j}) \end{aligned} \tag{1}$$

where n_p and n_q denote the total number of high- and medium-quality MAGs after and before MAG decontamination, respectively. $CN_{p,i}$ ($CN_{q,j}$) and $CT_{p,i}$ ($CT_{q,j}$) are the completeness and contamination values of the MAGs after (before) MAG decontamination.

Architecture of Deepurify

Genomic sequence and taxonomic lineage encoders

Deepurify utilizes GseqFormer and LSTM to encode genomic sequences and taxonomic lineages into 1024-dimensional space. The fundamental architecture of Deepurify is illustrated in Figure 1 **b**.

GseqFormer: Genomic sequence encoder. The genomic sequences were 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 [36], which takes advantage of transformer and convolutional neural networks (CNNs). We substituted the attention module of UniFormer with a new gated self-attention module, which was modified from Evoformer [37] (**Supplementary Note** 8). Because UniFormer has a limitation in modeling long sequences (¿1,000 bps), we adopted EfficientNet [38] to compress the input sequences into 512 tokens. This strategy enables the maximum lengths of input sequences up to 8,192 bps. Additionally, we incorporated a variety of *tricks* [39, 40, 41] into EfficientNet for efficient training and improved model robustness (**Supplementary Note** 9).

LSTM: Taxonomic lineage encoder. The taxonomic lineage ($T_i = [t_{p_i}, t_{c_i}, t_{o_i}, t_{f_i}, t_{g_i}, t_{s_i}]$) of a sequence ($s_i$) at species rank was considered as a sentence that concatenates taxon (t_{k_i}) at different taxonomic ranks (k_i), spanning from phylum to species. The taxonomic sentence would be encoded by a 5-layer LSTM model.

Deepurify training procedure

Contrastive training

For a sequence s_i, we represented its normalized encoded vector as θ_{s_i} and the normalized encoded vector of its taxonomic lineage at the species rank as θ_{T_i}.The prefix of T_i before k_i rank denotes as $T_{k_i} = [\leqslant t_{k_i}]$. We leveraged contrastive training to enable Deepurify to discriminate true (positive label, T_{k_i}) and multiple fake (negative labels, T_{k_j}) taxonomic lineages for a given sequence s_i. During training, we randomly selected k_i and created fake taxonomic lineages (T_{k_j}) from the taxonomic tree for contrastive, making sure they were distinct from T_{k_i} (**Supplementary Note** 10).

We applied four loss functions in contrastive training, including 1. sequence-taxonomy (ST) loss, 2. lineage-phyla (LP) loss, 3. indel loss, and 4. phyla-rank (PR) loss. Deepurify's primary objective is to optimize ST loss, which aims to make θ_{s_i} have higher cosine similarity with $\theta_{T_{k_i}}$ than with $\theta_{T_{k_j}}$. The ST loss (L_{ST}) is defined as:

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

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

where $d(\theta_{s_i}, \theta_{T_{k_i}}) = \theta_{s_i}^T \theta_{T_{k_i}}$, τ is a learnable parameter, J is the number of negative labels used in contrastive training. The numbers of species are different across phyla, which leads to unbalanced

sequences in the training set. We applied an oversampling strategy (**Supplementary Note** 11) and the focal loss [42] to mitigate this problem.

The goal of LP loss (L_{LP}) is to establish a taxonomic encoder to minimize the distance between $\theta_{T_{k_i}}$ and the phylum that s_i belongs to ($\theta_{t_{p_i}}$).

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

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

The aim of indel loss was to enable GseqFormer to accept the sequences with insertions and infer masked sequences.

$$L_{INDEL} = -[Y_{ins} log(P_{ins}(\theta_{s_i})) + (1 - Y_{ins}) log(1 - P_{ins}(\theta_{s_i})) + \frac{1}{M|V|} \sum_{m=1}^{M} \sum_{q=1}^{|V|} Y_{del}^{m,q} log(P_{del}^{m,q})] \tag{5}$$

where $P_{ins}(\theta_{s_i})$ is the predicted probability of s_i including insertions, and $Y_{ins} = 1$ indicates s_i including insertions. M is the number of masked positions in s_i and each position has six candidate values ($V = \{A, T, C, G, N, padding\}$). $P_{del}^{m,q}$ is the predicted probability of m-th masked position equals to V_q ($V_q \in V$). $Y_{del}^{m,q} = 1$ if the m-th masked position is V_q.

PR loss was used to examine the taxonomic inference of Deepurify on phylum rank.

$$L_{PR} = -[\sum_{c=1}^{C} Y_c log(P(\theta_{s_i}))] \tag{6}$$

where C is the number of phyla in the taxonomic tree, $Y_c = 1$ if s_i belongs to the phylum c, $P(\theta_{s_i})$ is the predicted probability of s_i belongs to the phylum c.

Therefore, the final training loss function of Deepurify is defined as follows:

$$L = \gamma L_{ST} + L_{LP} + L_{INDEL} + L_{PR} \tag{7}$$

We set $\gamma = 2$ in our experiments to emphasize the importance of L_{ST} in Deepurify training. The settings of other hyper-parameters were similar to UniFormer [36] (**Supplementary Note** 12).

Deepurify MAG decontamination procedure

Quantifying sequence taxonomic similarity

Deepurify utilized GseqFormer to encode genomic sequences and then to quantify their taxonomic similarities. This is achieved by identifying the taxonomic lineage from the taxonomic tree that exhibited the highest similarity with the genomic sequences (Figure 2 **a**). The degree of similarity between the sequences is positively correlated with the similarity of their predicted taxonomic lineages. For sequence s_i, GseqFormer would calculate the $P(\theta_{s_i}, \theta_{T_{j,k}}), j = 1...n$ for every taxon j at taxonomic rank k, where $T_{j,k} = [< t_k, j]$, n is the total number of taxa in rank k. We then selected the three candidate taxa with the highest values. The calculation of $P(\theta_{s_i}, \theta_{T_{j,k}})$ is similar to Eq.(3). For rank $k+1$, Deepurify would only search for the nodes, whose parents have been selected in rank k. This top-k searching strategy would result in a number of paths, ω, from the root to the species rank $(T_j, j = 1...\omega)$. We then calculate $P(\theta_{s_i}, \theta_{T_j}), j = 1...\omega$ to select the best path.

Detecting contaminated contigs in simulated MAGs

On simulated data, a contig with low taxonomic similarity to others in a MAG is more likely to be contaminated. Consequently, contigs were classified as contaminants if their predicted lineages differed from the predominant ones (Figure 2 **b**). We collected the predicted taxonomic lineages of contigs in a MAG and implemented an approach to determine the predominant one. The $Score_{j,k}$ was calculated for taxon j at rank k,

$$Score_{j,k} = \lambda \frac{1}{n_i} \Sigma_i^{n_i} P(\theta_{s_i}, \theta_{T_{j,k}}) + \mu V_{j,k} + \nu L_{j,k} \tag{8}$$

where n_i is the number of contigs that have predicted annotation 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. We would select $T_{j,k}$ with the highest value as the predominant lineage in the MAG at rank k. The selection would be performed for each rank, where the selected predominant lineage at rank $k+1$ should be the offspring of the one at rank k. At rank k, the contigs were identified as contaminants if their predicted lineages were different from the predominant ones.

Optimizing contig utilization in MAGs

On real data, Deepurify divides the contigs from a MAG to maximize the number of medium- and high-quality MAGs using the MAG-separated tree. The MAG-separated tree is constructed based on the predicted taxonomic lineage for the contigs in a MAG (Figure 2 **c**). Each node includes the contigs with the same annotation at rank k. We collected single-copy genes (SCGs) from the databases of SolidBin [43] and bacteria and archaea domains in CheckM [28]. We used Prodigal [44] to predict

genes on contigs and aligned them with SCGs by HMMER (http://hmmer.org). We removed contigs to eliminate duplicated SCGs within each node (**Supplementary Note** 13). This procedure may result in multiple candidate contig divisions for a node. To enhance computational efficiency, Deepurify discarded the divisions if 1. more than 45% of the original SCGs were removed and 2. the total lengths of involved contigs were less than 550kb (**Supplementary Note** 14). We applied CheckM to each division and selected the best one to represent a node based on quality score (QS). Its quality (high-, medium- or low-quality) was also annotated by CheckM. Deepurify utilized post-order traversal to traverse the MAG-separated tree to maximize the total number of medium- and high-quality MAGs (Figure 2 **d**, **Supplementary Note** 15).

Data availability

The microbial representative genomes and their associated taxonomic lineages were downloaded from the proGenomes v2.1 database. The SIM_1 was uploaded to https://zenodo.org/record/8343498. The SIM_2 was uploaded to https://zenodo.org/record/8343506. 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' with the following link: https://data.cami-challenge.org /participate/. The Illumina short-reads, 10x linked-reads, and long-reads of $S1$ data were downloaded from NCBI SRA accessions SRR19505636. The fecal metagenomic sequencing reads of the IBS-D cohort were downloaded from China National GeneBank (CNGB) with accession number CNPO000334.

Code availability

The source code used in the manuscript is freely available under an MIT license at https://github.com/zoubohao/Deepurify_Project. The versions of the software used in the study were provided in the **Supplementary Note** 16.

Funding

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

Authors' contributions

LZ conceived the study. BHZ designed and implemented the Deepurify algorithms. LZ and BHZ conceived the experiments. BHZ, YD, and ZMZ conducted the experiments. BHZ and JJW analyzed the results. BHZ drew and analyzed the plots. BHZ and LZ wrote the manuscript. KCC and SS contributed computational resources.

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 biology and evolution* **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. *Frontiers in microbiology* **11**, 1377 (2020).

[3] Kaster, A.-K. & Sobol, M. S. Microbial single-cell omics: the crux of the matter. *Applied microbiology and biotechnology* **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. *Environmental microbiology* **20**, 1016–1029 (2018).

[5] Nurk, S., Meleshko, D., Korobeynikov, A. & Pevzner, P. A. metaspades: a new versatile metagenomic assembler. *Genome research* **27**, 824–834 (2017).

[6] Liang, K.-c. & Sakakibara, Y. Metavelvet-dl: a metavelvet deep learning extension for de novo metagenome assembly. *BMC bioinformatics* **22**, 1–21 (2021).

[7] Kolmogorov, M. *et al.* metaflye: scalable long-read metagenome assembly using repeat graphs. *Nature Methods* **17**, 1103–1110 (2020).

[8] Nissen, J. N. *et al.* Improved metagenome binning and assembly using deep variational autoencoders. *Nature biotechnology* **39**, 555–560 (2021).

[9] Alneberg, J. *et al.* Binning metagenomic contigs by coverage and composition. *Nature 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**, 1–18 (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. *Nature biotechnology* **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. *Nature 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. *Nature biotechnology* **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. *Nature microbiology* **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 Research* (2022).

[21] Mock, F., Kretschmer, F., Kriese, A., Böcker, S. & Marz, M. Taxonomic classification of dna sequences beyond sequence similarity using deep neural networks. *Proceedings of the National Academy of Sciences* **119**, e2122636119 (2022).

[22] Drillon, G., Champeimont, R., Oteri, F., Fischer, G. & Carbone, A. Phylogenetic Reconstruction Based on Synteny Block and Gene Adjacencies. *Molecular Biology and Evolution* **37**, 2747–2762

(2020). URL https://doi.org/10.1093/molbev/msaa114. https://academic.oup.com/mbe/article-pdf/37/9/2747/33866566/msaa114.pdf.

[23] Periwal, V. & Scaria, V. Insights into structural variations and genome rearrangements in prokaryotic genomes. *Bioinformatics* **31**, 1–9 (2015).

[24] Sczyrba, A. *et al.* Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods* **14**, 1063–1071 (2017).

[25] Zhang, Z., Yang, C., Fang, X. & Zhang, L. Benchmarking de novo assembly methods on metagenomic sequencing data. *bioRxiv* (2022).

[26] Zhao, L. *et al.* A clostridia-rich microbiota enhances bile acid excretion in diarrhea-predominant irritable bowel syndrome. *The Journal of clinical investigation* **130**, 438–450 (2020).

[27] Radford, A. *et al.* Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763 (PMLR, 2021).

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

[29] 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).

[30] Wei, W. *et al.* Altered metabolism of bile acids correlates with clinical parameters and the gut microbiota in patients with diarrhea-predominant irritable bowel syndrome. *World Journal of Gastroenterology* **26**, 7153 (2020).

[31] Williams, B. B. *et al.* Discovery and characterization of gut microbiota decarboxylases that can produce the neurotransmitter tryptamine. *Cell host & microbe* **16**, 495–503 (2014).

[32] Mars, R. A. *et al.* Longitudinal multi-omics reveals subset-specific mechanisms underlying irritable bowel syndrome. *Cell* **182**, 1460–1473 (2020).

[33] 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**, 1–11 (2020).

[34] Lai, S. *et al.* metamic: reference-free misassembly identification and correction of de novo metagenomic assemblies. *Genome Biology* **23**, 242 (2022).

[35] Mende, D. R. *et al.* progenomes2: an improved database for accurate and consistent habitat, taxonomic and functional annotations of prokaryotic genomes. *Nucleic Acids Research* **48**, D621–D625 (2020).

[36] Li, K. *et al.* Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676* (2022).

[37] Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).

[38] Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114 (PMLR, 2019).

[39] Li, C., Zhou, A. & Yao, A. Omni-dimensional dynamic convolution. In *International Conference on Learning Representations* (2021).

[40] Guo, M.-H., Lu, C.-Z., Liu, Z.-N., Cheng, M.-M. & Hu, S.-M. Visual attention network. *arXiv preprint arXiv:2202.09741* (2022).

[41] Wang, H. *et al.* Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555* (2022).

[42] Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988 (2017).

[43] 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).

[44] Hyatt, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics* **11**, 1–11 (2010).

[45] Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with kraken 2. *Genome biology* **20**, 1–13 (2019).

[46] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520 (2018).

[47] Nguyen, P. M., Le, T., Nguyen, H. T., Tran, V. & Nguyen, M. L. Phrasetransformer: an incorporation of local context information into sequence-to-sequence semantic parsing. *Applied Intelligence* **53**, 15889–15908 (2023).

[48] Iandola, F. N. *et al.* Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360* (2016).

[49] Robinson, J., Chuang, C.-Y., Sra, S. & Jegelka, S. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[50] Huang, G., Sun, Y., Liu, Z., Sedra, D. & Weinberger, K. Q. Deep networks with stochastic depth. In *European conference on computer vision*, 646–661 (Springer, 2016).

[51] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**, 1929–1958 (2014).

[52] Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[53] Loshchilov, I. & Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

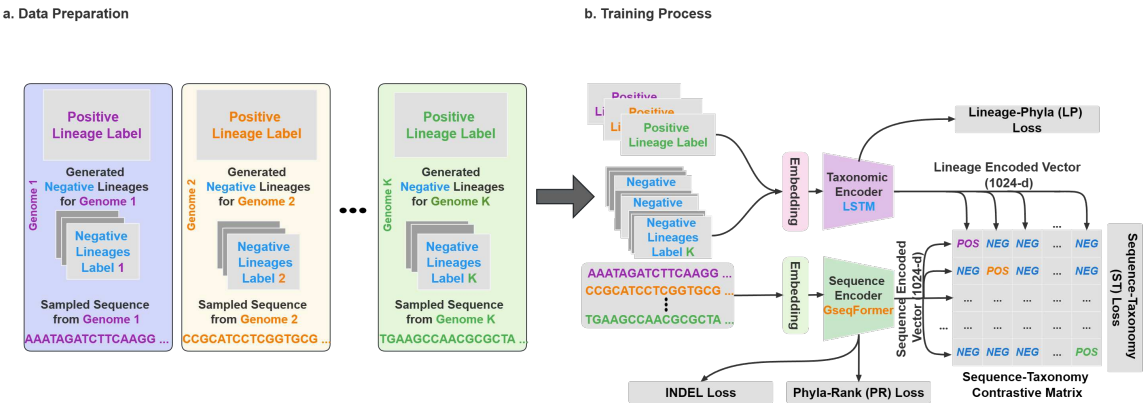

Figure 1: The Deepurify training procedure consisted of two phases: data preparation and the training process. (**a**). In data preparation, Deepurify used the positive lineage label of each genome, generated multiple negative lineage labels for each genome, and sampled an appropriate-length sequence from the corresponding genome. (**b**). During training, the taxonomic encoder encoded positive and negative lineage labels. Sequences were encoded using GseqFormer. A sequence-taxonomy contrastive matrix was built based on calculating the cosine similarity between encoded sequences and lineages. The cosine similarity between the positive label and the sequence is anticipated to surpass that between negative labels and the sequence. Therefore, the ST loss accounted for the majority of the training losses, whereas the other losses facilitated the training process and improved the model's robustness.

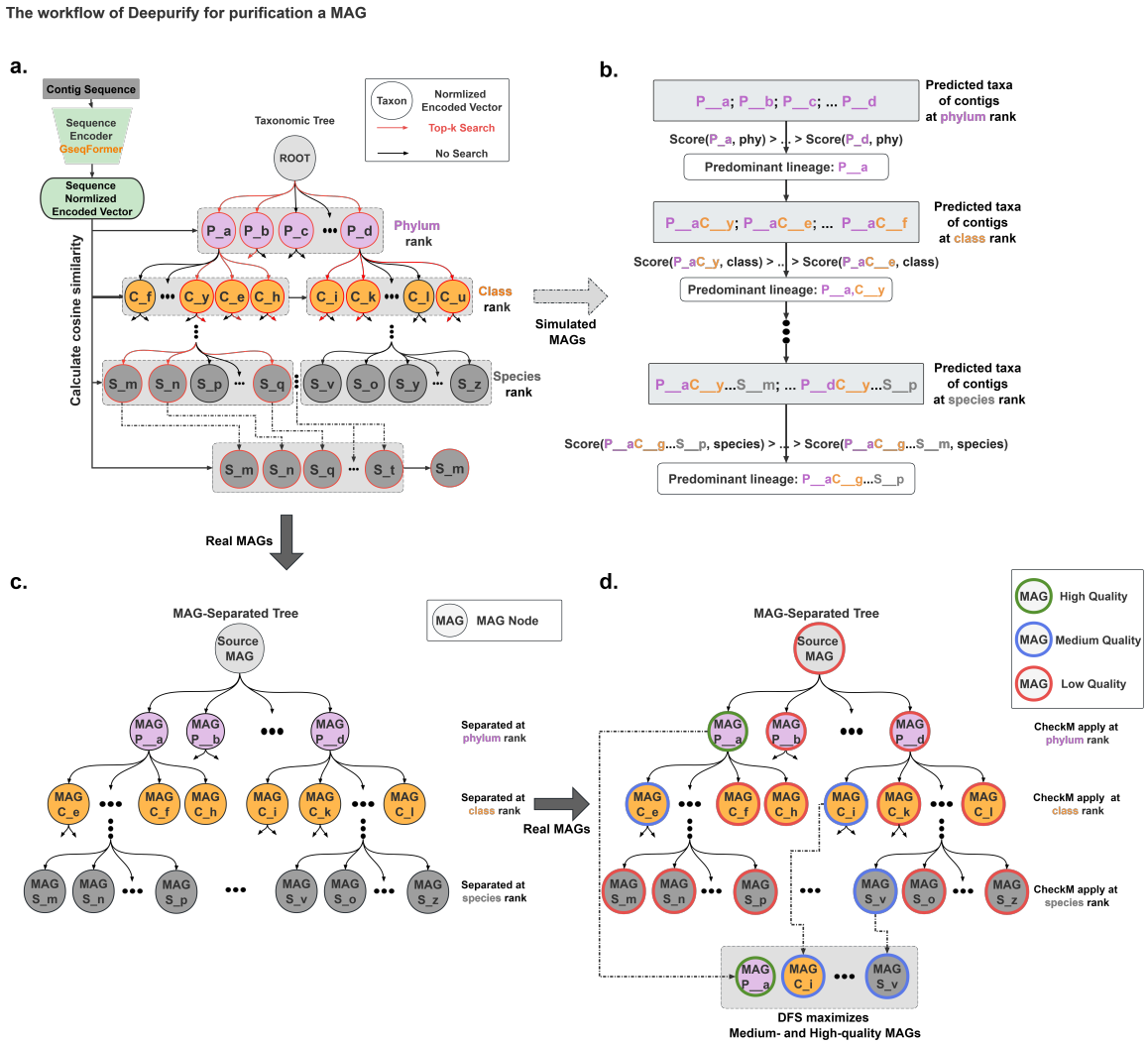

Figure 2: The purification workflow of Deepurify. (**a**). Deepurify assesses taxonomic similarities among sequences through the assignment of taxonomic lineages. It employs a top-k search approach within the taxonomic tree to identify candidate lineages, subsequently selecting the lineage with the highest similarity to the sequences. (**b**). Deepurify applies a scoring function to the lineage of contigs to determine the predominant lineage of contigs in the MAG. The taxon with the highest score is chosen as the predominant lineage at different ranks. This process crosses ranks from phylum to species, ensuring the predominant lineage is consistent and coherent. (**c**). For optimal contig utilization within a MAG without dropping contaminated contigs directly, Deepurify constructs a MAG-separated tree. This tree partitions the MAG based on predicted lineage. Each node contains contigs sharing the same taxon at that rank. To prevent duplicate single-copy genes (SCGs), Deepurify applies SCGs to each node. (**d**). Deepurify employs a depth-first search (DFS) algorithm on the MAG-separated tree to maximize the total number of high- and medium-quality MAGs.

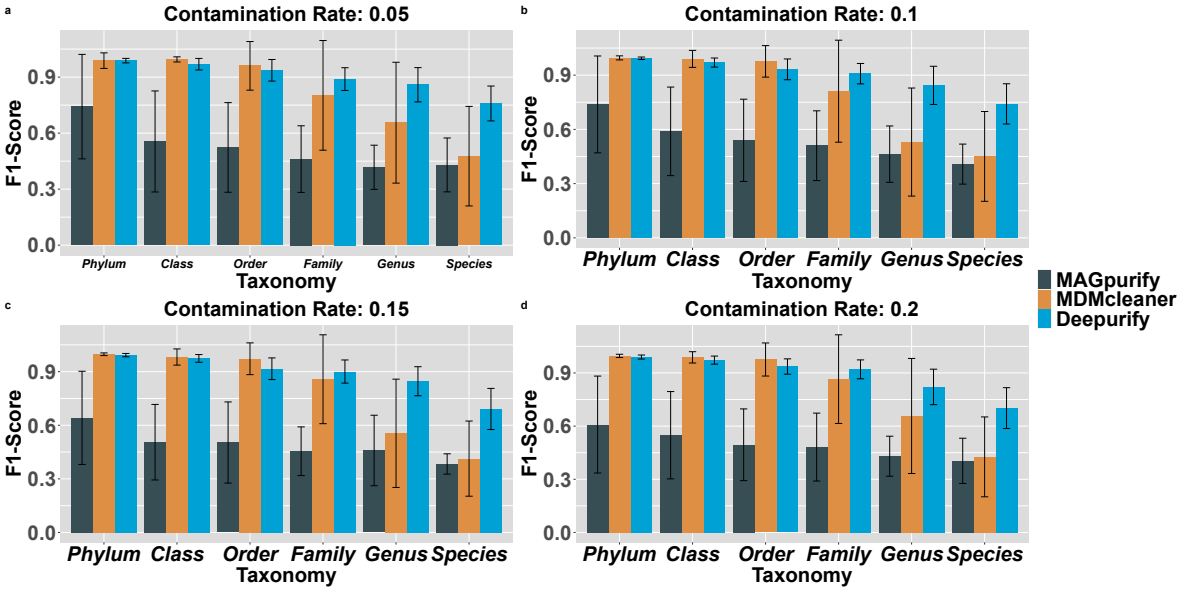

Figure 3: The averaged macro F1-score at various contamination ratios and taxonomic ranks for MAGpurify, MDMcleaner, and Deepurify. The error bars represent standard deviations.

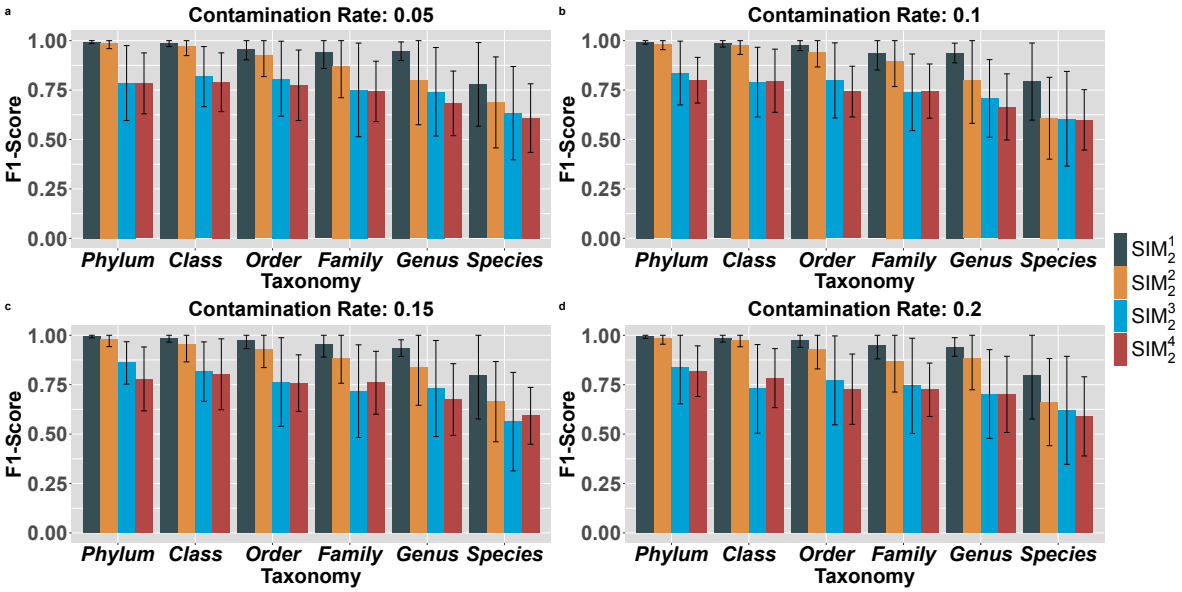

Figure 4: The averaged macro F1-score calculated for SIM_2^1, SIM_2^2, SIM_2^3, SIM_2^4 at different contamination ratios and taxonomic ranks for Deepurify.

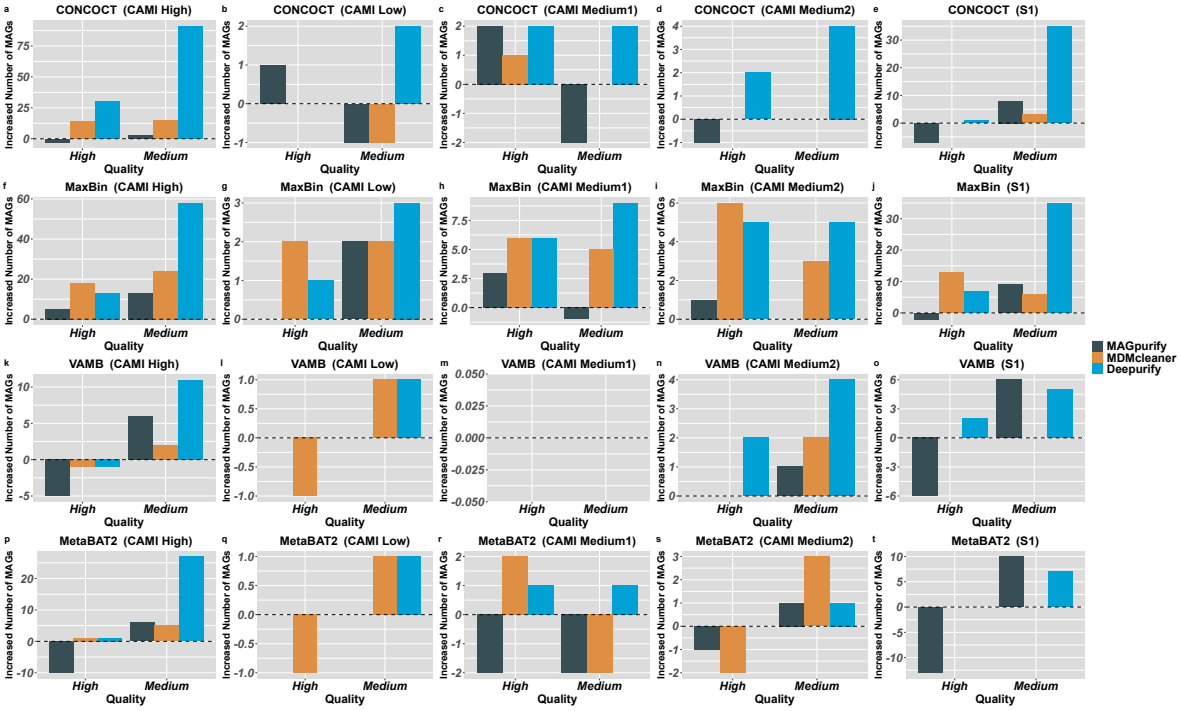

Figure 5: The increased number of MAGs (INM) for CAMI I and $S1$ datasets with different binning methods (CONCOCT, MaxBin, VAMB, MetaBAT2) for MAGpurify, MDMcleaner, and Deepurify.

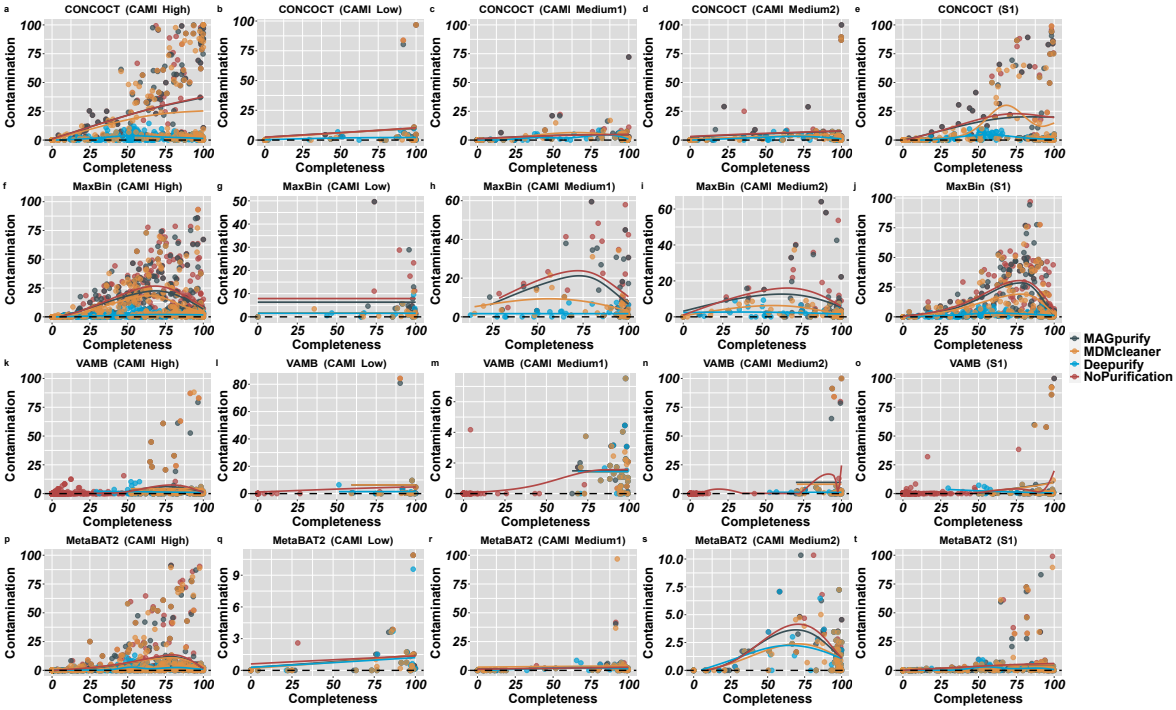

Figure 6: The correlation between the completeness and contamination levels of MAGs both before and after purification using MAGpurify (grey), MDMcleaner (orange), and Deepurify (blue) in the CAMI I and $S1$ datasets. These datasets were initially binned using CONCOCT, MetaBAT2, VAMB, and MaxBin. A Generalized Additive Model (GAM) was applied to construct a smooth curve that represents the contamination trends exhibited by MAGs in these instances. These plots serve to illustrate the superior purification performance of Deepurify when used on MAGs with high contamination levels.

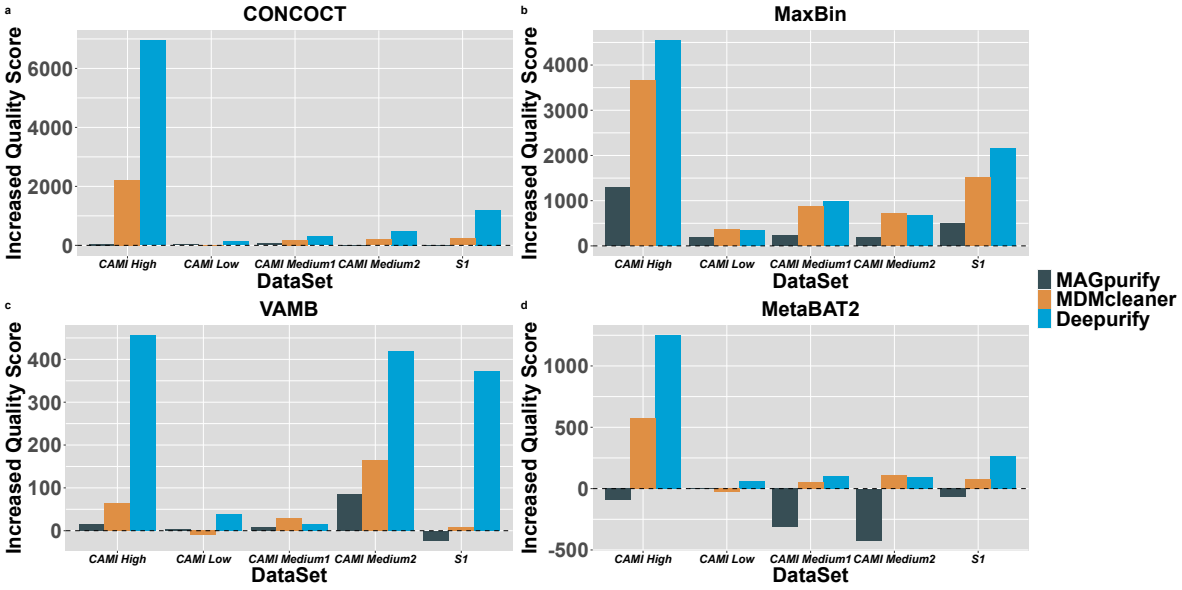

Figure 7: The increased quality scores (IQS) for the CAMI I and $S1$ datasets binned with MaxBin, CONCOCT, VAMB, and MetaBAT2 reveal that Deepurify's IQS is substantially higher than that of MAGpurify and MDMcleaner in almost all cases.

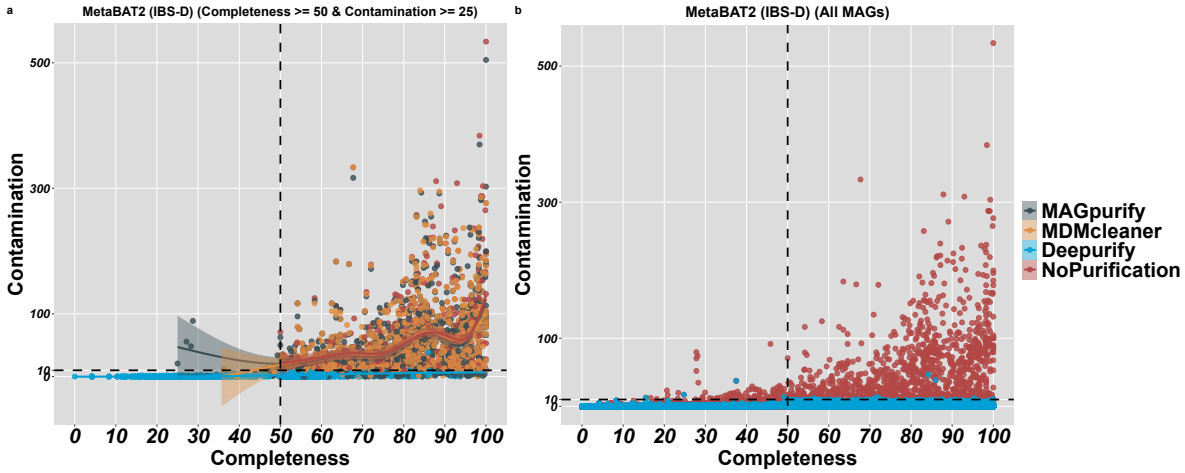

Figure 8: The correlation between completeness and contamination of MAGs before and after purification. In (**a**), we employed MAGpurify (grey), MDMcleaner (orange), and Deepurify (blue) to filter the contamination of MAGs with completeness greater than 50% and contamination exceeding 25%. A Generalized Additive Model (GAM) was applied to construct a smooth curve that effectively captured the contamination trends exhibited by MAGs in these instances. In (**b**), Deepurify (blue) was utilized for all MAGs within the IBS-D cohort. Notably, Deepurify exhibits the capacity to rescue a significant proportion of MAGs with high contamination rates ($> 10\%$).