binny: an automated binning algorithm to recover high-quality genomes from complex metagenomic datasets

Oskar Hickl[1], Pedro Queirós[2], Paul Wilmes[2], Patrick May[1,†] and Anna Heintz-Buschart[3,†,*],

1 Bioinformatics Core, Luxembourg Centre for Systems Biomedicine, University of Luxembourg
2 Systems Ecology, Luxembourg Centre for Systems Biomedicine, University of Luxembourg
3 Biosystems Data Analysis, Swammerdam Institute for Life Sciences, Faculty of Science, University of Amsterdam

*** Correspondence: a.u.s.heintzbuschart@uva.nl**
† Equal contributions.

Abstract

The reconstruction of genomes is a critical step in genome-resolved metagenomics as well as for multi-omic data integration from microbial communities. Here, we present *binny*, a binning tool that produces high-quality metagenome-assembled genomes from both contiguous and highly fragmented genomes. Based on established metrics, *binny* outperforms existing state-of-the-art binning methods and finds unique genomes that could not be detected by other methods.

binny uses k-mer-composition and coverage by metagenomic reads for iterative, non-linear dimension reduction of genomic signatures as well as subsequent automated contig clustering with cluster assessment using lineage-specific marker gene sets.

When compared to five widely used binning algorithms, *binny* recovers the most near-complete (>95% pure, >90% complete) and high-quality (>90% pure, >70% complete) genomes from simulated data sets from the Critical Assessment of Metagenome Interpretation (CAMI) initiative, as well as from a real-world benchmark comprised of metagenomes from various environments. *binny* is implemented as Snakemake workflow and available from `https://github.com/a-h-b/binny`.

Introduction

High-throughput shotgun sequencing has become the standard to investigate metagenomes [40,45]. Metagenome-assembled genomes (MAGs) allow the linking of the genetic information at species or strain level: In the absence of cultured isolates, MAGs form an important point of reference. Thereby, study-specific MAGs have led to the discovery of previously uncharacterised microbial taxa [57] and deepened insights into microbial physiology and ecology [14, 50]. In addition, large system-wide collections, which have been assembled recently, e.g. for the human microbiome [3] and several environmental systems [39], provide researchers with a common resource for short-read annotation. These collections also provide researchers with an overview of the pangenomic potential of microbial taxa of interest [23, 52]. In addition to facilitating the interpretation of metagenomic data, genome-resolution also provides an anchor for the integration of functional omics [18, 19].

However, obtaining high-quality MAGs is still challenging [13]. Most approaches start from assembled contigs, which are then binned by clustering, e.g. expectation-maximization clustering [4, 55] or graph-based clustering [22], of k-mer frequency or abundance profiles or both. Because of this, binning algorithms have to account for and re-evaluate issues with metagenomic assemblies, such as fragmentation of the assembly because of insufficient sequencing depth, repeat elements within genomes, and unresolved ambiguities between closely related genomes. In addition, the features based on which contigs are binned are not generally homogeneous over genomes: for example copy number, and thereby metagenomic coverage, may vary over the replicating genome; certain conserved genomic regions, but also newly acquired genetic material, can deviate in their k-mer frequency from the rest of the genome [13].

In the face of these challenges, the algorithms used to bin assembled metagenomic into congruent groups which form the basis for MAGs, can approximately be evaluated according to a set of criteria [35]. Most

importantly, MAGs should be as complete as possible and contain as little contamination as possible. In metagenomic datasets with defined compositions, such as those provided by the Critical Assessment of Metagenome Interpretation (CAMI) initiative [34, 48], the evaluation can be achieved by comparison with the reference genomes. For yet un-sequenced genomes, completeness and contamination can be assessed based on the presence and redundancy of genes that are expected to be present as single copies in many [38] or all [9] bacteria or archaea [47], or in specific lineages [42]. Contiguity and GC-skew provide further measures for highly complete genomes [13]. For reporting and storing MAGs in public repositories, the MIMAG standard has been proposed [7]. In addition to completeness and contamination based on protein-coding genes, this standard also takes into account the presence of tRNAs and rRNAs. The latter present particular challenges for assembly and binning methods alike [13]. Nevertheless, the recruitment of rRNA genes to MAGs would improve the association with existing MAG collections [3, 37] and rRNA-gene based databases [2], which are widely used for microbial ecology surveys. In addition to binning tools, methods that refine MAGs by complementing results from multiple binning methods have been developed [51, 53]. These generally improve the overall yield and quality of MAGs [56]. Finally, manual refinement is still recommended and is supported by multiple tools [6, 8, 13, 15, 16].

Here, we present *binny*, an automated binning method that was developed based on a semi-supervised binning strategy [18, 27]. *binny* is implemented as a reproducible Python-based workflow using Snakemake [26]. *binny* is based on iterative clustering of dimension-reduced k-mer and abundance profiles of metagenomic contigs. It evaluates clusters based on the presence of lineage-specific single copy marker genes [42]. We benchmarked *binny* against six CAMI [34, 48] data sets and compared the results to the most popular binning methods MetaBAT2 [22], MaxBin2 [55], CONCOCT [4], and the recently developed VAMB [41]. We evaluated the contribution of *binny* to automatic MAG refinement using MetaWRAP [53] and DAS tool [51]. Finally, we evaluated the MAGs returned by all approaches from real-world metagenomic datasets from a wide range of ecosystems. We report that *binny* outperforms existing methods in terms of completeness and purity and improves combined refinement results. *binny* also returned most high-quality and near-complete MAGs from both highly fragmented and more contiguous metagenomes over a range of microbial ecosystems.

Materials and Methods

binny workflow

binny is implemented as a Snakemake [26] workflow (Figure 1). At the centre of the workflow is the binning algorithm written in Python, which uses iterative, nonlinear dimension reduction of genomic *k*-mer signatures and subsequent automated contig clustering with cluster assessment by lineage-specific marker gene sets. Preparatory processing steps include calculation of average depth of coverage, gene calling using Prokka [49], masking of rRNA gene and CRISPR regions on input contigs, and applying CheckM [42] marker gene identification by using Mantis [44] gene annotations.

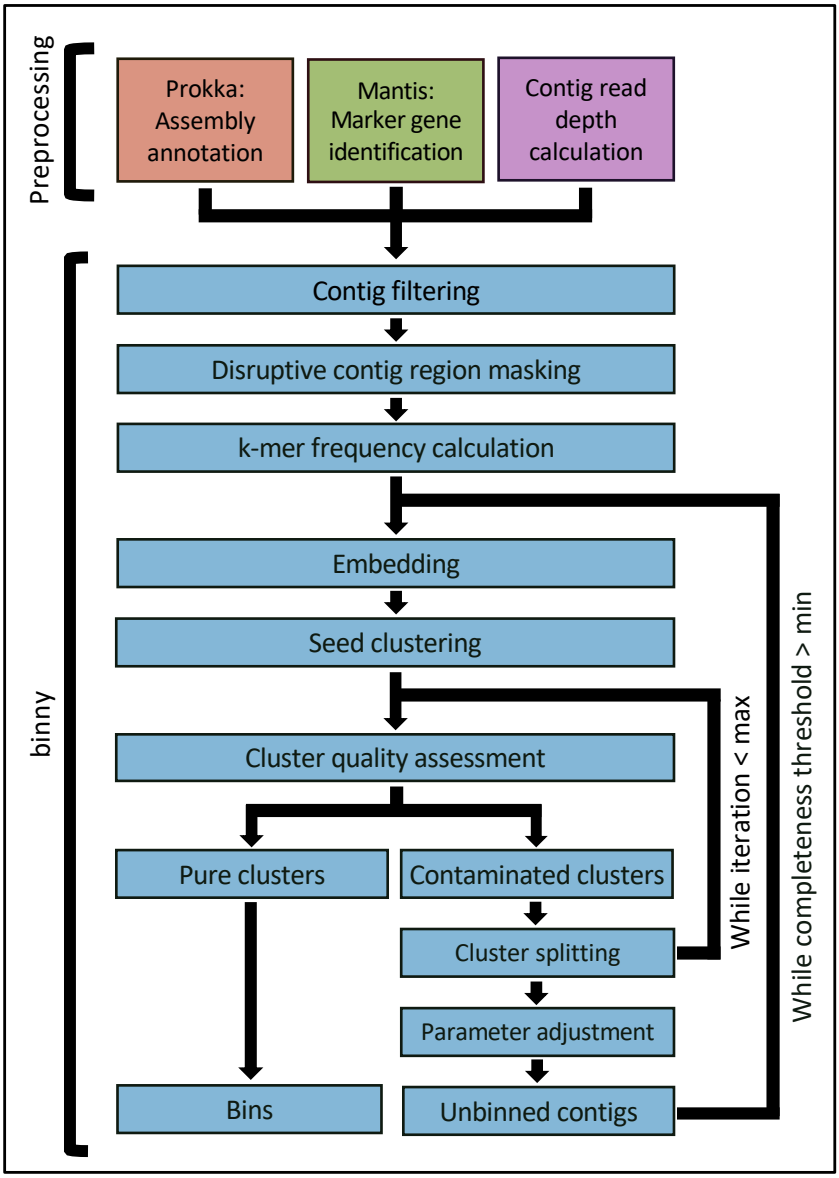

Figure 1. *binny* workflow. Overview of the Snakemake pipeline and of *binny*'s binning method. Preprocessing includes assembly annotaion using Prokka, CheckM marker gene detection using Mantis, and (optional) average contig read coverage calculation. *binny* filters out contigs shorter than the specified value, masks potentially disruptive contig regions before calculating k-mer frequencies for the chosen k-mer size(s). In its main routine, *binny* iteratively embeds the contig data into two dimensional space, forms clusters, assesses them with marker genes, and extracts high quality clusters as MAGs.

Overview *binny* operates in an iterative manner after processing of the annotated marker gene sets: *binny* clusters un-binned contigs. Each iteration consists of non-linear dimension reduction on the selected features (depth(s) of coverage and k-mer frequencies) and clustering of the contigs based on the resulting two-dimensional coordinates. Clusters are selected, if the contained marker gene sets indicate purity and completeness above defined thresholds. A new iteration is started on left-over un-binned contigs with dynamically adjusted parameters. Finally, clusters above the thresholds are output as MAGs.

Marker gene set processing *binny* generates a directed graph database of the CheckM [42] taxon-specific marker sets annotated per contig in NetworkX [17]. This allows for fast access to the hierarchical (lineage-based) information. Some marker sets are omitted, as they are very small and/or led to imprecise assessments in testing (Supplementary Table 1).

Filtering of short sequences Contigs below the minimum specified size (default 1000 bp) which do not contain marker genes are removed. This retains the maximum amount of information from an assembly, because only contigs are omitted which have low information content.

Masking of disruptive sequence regions Repetitive regions on a sequence could possibly skew the k-mer frequency significantly and, thus, adversely affect the binning process. To avoid this and still keep sequences intact, *binny* masks sequence elements/regions such as rRNAs and CRISPR regions so that they are ignored during the k-mer frequency calculation. CRISPR regions contain foreign genetic elements, which have k-mer frequencies that can deviate substantially from the rest of the genome, while rRNAs have highly conserved sequences whose k-mer profiles do not resemble the rest of a given genome.

Single contig genome recovery Because (near-)complete genomes represented by single contigs might not be distinguishable from noise or be clustered together with highly similar contigs of other genomes during the clustering step, their separation and subsequent recognition as pure and complete is hampered. Therefore, contigs with at least 40 different markers are extracted first and assessed. If they are at least 80% pure and complete (as *binny* tends to underestimate both metrics), they are kept as single MAGs and do not enter the iterative binning procedure.

Binning features *binny* uses two features of the contigs for dimensionality reduction and clustering: the k-mer frequencies ($k = 2$, 3, and 4: centered log ratio transformed) and average depth of read coverage (raw read counts of one or more samples). If available, multiple sources of depth of coverage information can be included in form of a file with tab-separated depth of coverage values per sample.

Dimensionality reduction *binny* uses the Fast Fourier Transform-accelerated Interpolation-based t-distributed Stochastic Neighbor Embedding (flt t-SNE) implementation of openTSNE [43] to reduce the dimensionality of all features to two. To reduce the computation time of t-SNE dimensionality reduction, Principal Component Analysis (PCA) is used beforehand to already lower the dimensionality of the initial feature matrix to either as many dimensions needed to explain 75% of the variation or at a maximum 75 dimensions. To improve the embedding quality, especially with large datasets, multiple strategies are used: i) A multi-scale kernel with perplexities 10 and 100 is used instead of a Gaussian model to balance out local and global structure, as described by Kobak and Berens [24]. ii) An early exaggeration of 1000 for the first 250 optimization iterations was chosen, to improve the embedding [32]. iii) To avoid the impreciseness of Euclidean distance measures in high dimensional space, Manhattan distance was chosen instead [1]). Default values were kept for all other openTSNE parameters.

Iterative clustering *binny* uses hierarchical density-based spatial clustering of applications with noise (HDBSCAN) [11] on the generated two dimensional embedding, in iterations. Clustering is performed with the following parameters:

The minimum cluster size is calculated with $ln(ncontigs)$, the minimum number of samples is set to 5, and the distance metric used is Manhattan.

For each cluster, completeness and purity are assessed (see below). If a cluster passes the completeness threshold (by default starting with 90% and then decreasing to a minimum of 80%) and has a purity above 85%, it is kept as a MAG. Otherwise, *binny* will attempt to split that contig cluster iteratively using

HDBSCAN with the same parameters as for the initial clustering but adding the raw depth(s) of coverage as additional dimension(s). Clustering is repeated until no new clusters are identified and/or the maximum number of iterations is reached. The default maximum of two iterations has proven to be effective to produce pure MAGs, while maintaining the impact of the iterative clustering on the runtime moderate. The purity threshold of 85% was chosen, because it was observed that *binny* tends to slightly underestimate the purity of clusters at high completeness levels. To prevent the selection of low purity clusters, the purity threshold is increased to 87.5% at subsequent stages of the iterative clustering, when the completeness threshold is below 90%.

Cluster assessment using marker gene sets Clusters are assessed by calculating the purity and completeness based on the CheckM marker grouping approach, where marker genes known to be co-located in genomes of a lineage are collapsed into marker sets [42]. *binny* calculates MAGcompleteness C as the number of marker sets of which one or more marker genes of that set are detected BMS divided by the total number of marker sets for a given lineage MSS:

$$C = \frac{BMS}{MSS}$$

Purity P of a MAG is defined as:

$$P = \frac{UM}{TM}$$

where NRM is the number of non-redundant marker genes in the MAG and TM the total number of marker genes in the MAG.

The taxonomic level and identity of the marker set is chosen dynamically: Assessment starts with completeness and purity of the domain-level marker sets and traverses the lineage down one taxonomic level at a time. For each level, completeness and purity are compared to the previous level. If the current marker set has a equal or higher completeness and at least 75% of the purity of the previously best fitting marker set, the current taxonomic level and assessment are retained. This means that *binny* emphasises completeness over purity, as the marker set with the highest completeness is least likely to be matching by chance.

Iterative embedding *binny* creates 2-dimensional embeddings of the un-binned contigs of each clustering and runs subsequent clustering iterations, for as long as it finds new MAGs that satisfy the purity and completeness thresholds. By default, the completeness threshold is decreased by 5% in every clustering iteration where no MAGs were found, down to the minimum completeness threshold (80% completeness). Once the minimum completeness threshold is reached, the purity threshold is increased to 87.5%. For each iteration, the early exaggeration will be increased by 10%. Therefore, by slightly changing the embedding, different clusters can be formed in each iteration. When no more MAGs are found at the minimum completeness value, *binny* runs one more round with an early exaggeration that is 10 times as high as the last value.

binny contig annotation

Contig depth calculation If not provided explicitly, the average depth of coverage calculation can be performed directly from given BAM files within the Snakemake workflow using BEDTools [46] *genomeCoverageBed* and an in-house script.

Coding sequence, rRNA gene, and CRISPR prediction by Prokka A modified Prokka [49] executable is run in *–metagenome* mode, to retrieve open reading frame (ORF) predictions from Prodigal [21], rRNA gene predictions from barrnap [49] and CRISPR region predictions from *minced* [5]. The modification improves speed by omitting the creation of a GenBank output and by the paralellisation of the Prodigal ORF prediction step. Additionally, it allows the output of partial coding sequences without start- and/or stop-codons, which are frequently encountered in fragmented assemblies. No functional annotations of the called coding sequences are performed. The gff output of Prokka is used in the subsequent steps.

Marker gene set annotation Taxon-specific marker gene sets are acquired from CheckM [42] `https://data.ace.uq.edu.au/public/CheckM_databases/` upon installation of *binny*, hidden Markov models (HMM) of marker genes not found in `taxon_marker_sets.tsv` are removed, and `checkm.hmm` is split into Pfam [36] and Tigrfam [30] parts. Mantis [44] is used to annotate coding sequences using the two HMM

sets. Because both resources are of different scope and quality, consensus generation weights of 1.0 and 0.5 are used for Pfam and Tigrfam models, respectively. Mantis' depth-first search algorithm is used for hit processing and the e-value threshold is set to 1×10^{-10}.

Parameter customization To optimize for their use case a user can chose to change the sizes and number of k-mers used, the minimum length to filter contigs by, as well as the minimum completeness and purity thresholds. Additionally, it is possible to chose between internal calculation of the average contig read depth or supply of a depth value file.

Requirements / dependencies *binny* is implemented as a Snakemake pipeline and an installation script is provided, which takes care of the installation of all necessary dependencies as well as other data required.

Benchmarking

Synthetic benchmark data *binny* performance was evaluated using data sets from the Critical Assessment of Metagenome Interpretation initiative [34, 48]. To benchmark against data of varying complexity, five short-read data sets with a total of 49 samples were chosen from the 2nd CAMI Toy Human Microbiome Project Dataset (`https://data.cami-challenge.org/participate`). Additionally, to test against a very large, high complexity data set, the five sample Toy Test Dataset High Complexity from the first CAMI challenge (`hhttps://openstack.cebitec.uni-bielefeld.de:8080/swift/v1/CAMI_I_TOY_HIGH`) was used.

To test the performance on co-assembled data, the pooled assemblies of each of the six CAMI datasets and the respective number of sample read files for each data set were used. Contig read depth per sample was calculated using *binny* and provided to all binning methods unless stated otherwise. Read files were de-interleaved (`https://gist.github.com/nathanhaigh/3521724#file-deinterleave_fastq-sh`) and mapped against the contigs using `bwa-mem` [29].

Selection of default values for minimum contig size

To find a high performance default minimum contig size value, *binny* was run on the 54 CAMI samples from the 6 data sets with the following four pairs of values for minimum contig size and minimum size with CheckM markers: i) 2000bp and 2000bp, ii) 1000bp and 1000bp, iii) 1000bp and 0bp, iv) 0bp and 0bp. Based on benchmark results iii) was chosen as default parameter, as it produced the highest number of NC MAGs (quality metrics explained in section: MAG quality standards) while other key metrics were similar between parameter sets (Supplementary Figure 1).

Real world benchmark data To assess the binning performance in different real-world scenarios with a variety of metagenome sizes, complexities and qualities, 105 metagenomes used in the MetaBAT2 publication [22] for benchmarking were chosen based on the availability of preprocessed read data at the Joint Genome Institute (JGI). The newest available assembly for the metagenomes and the respective preprocessed reads were retrieved from JGI (`https://jgi.doe.gov/`). The read data was processed in the same way as the CAMI data above. For a full list with all sample information see Supplementary Table 2.

Other binning and refinement methods The performance of *binny* was compared to four other state-of-the-art binning methods, and as input to two binning refinement tools. They were all run using the default settings, unless specified otherwise:
MaxBin2 (2.2.7) [55] was run by providing the contig read depth files using the `-abund` option, and with the `-verbose` option.
MetaBAT2 (2.2.15) [22] was provided the contig read depth files using the `-a` option, and the options `-cvExt`, `--saveCls`, as well as `-v`.
CONCOCT (1.1.0) [4] was run following the 'Basic Usage' section in the documentation (`https://concoct.readthedocs.io/en/latest/usage.html`)
VAMB (3.0.2) [41] was run with the default parameters and using the Snakemake pipeline as described in the documentation at `https://github.com/RasmussenLab/vamb/blob/master/README.md`. Because VAMB is designed to achieve optimal performance through the combination of the data of multiple samples, the samples from each of the six CAMI data sets were concatenated and run together, as described by the authors (README sections `Recommended workflow` and `Snakemake workflow`. For the real-world metagenomes,

samples sharing a JGI GOLD Study ID were run together. As VAMB could not be successfully run on some of the real-world samples using default values, or when trying with lower values of `-m` and `--minfasta`, the number of MAGs recovered was counted as zero for these samples. For a list of samples where this was the case see Supplementary Table 3.

DAS Tool (1.1.2) [51] was run using Diamond [10] as a search engine on the unfiltered binning method outputs.

MetaWRAP (1.2.2) [53] was set to output only contigs with less than 10% contamination and at least 70% completeness was provided the unfiltered binning method outputs.

Both refinement tools, DAS Tool and MetaWRAP, were run per sample using the data of all five binning methods and all binning methods except *binny*, to asses how many MAGs *binny* contributes in an ensemble approach.

MAG quality standards To match real world workflows, all binning outputs were assessed using CheckM (1.0.12) [42] and filtered to contain only MAGs with a purity > 90% and a completeness > 70%. The latter threshold was set in accordance with the CheckM publication, which suggests that CheckM results are reliable at completeness equal or larger than 70%. MAGs above these thresholds are subsequently called "high quality" (HQ) MAGs. MAGs with a purity > 95% and a completeness > 90% are called "near complete" (NC) MAGs, as defined by Bowers et al. [7].

Additionally, the minimum information about a metagenome-assembled genome (MIMAG) definition of high-quality MAGs was employed, requiring at least 18 unique tRNAs and three unique rRNAs to be present in the MAG in addition to a purity of > 95% and a completeness of > 90% [7].

Besides the recall in terms of bp of the assembly recovered, the read recruitment of MAGs was assessed. All reads mapping as primary mappings to contigs of a MAG were counted per sample and divided by the total read count (forward + reverse) using pysam (`https://github.com/pysam-developers/pysam`).

Assessment of benchmark results Results for the CAMI benchmark were processed using AMBER (2.0.3) [33], a genome reconstruction evaluation tool, with the following parameters, `-x "50,70,90"` and `-k "circular element"`.

To evaluate a MAG, AMBER selects the gold standard genome with the highest share of bps in that MAG as the reference. In contrast to CheckM, where purity and completeness refer to the amount of marker genes present or duplicated, within AMBER and using an available a gold standard, purity and completeness refer to the amount of bp of the reference genome recovered for completeness, and the share of bp of a given MAG with a given reference genome, respectively. Additionally, to assess one or multiple data sets taken together, AMBER defines overall completeness as '*Sum of base pairs coming from the most abundant genome in each predicted genome bin divided by the sum of base pairs in all predicted bins. ...*', and overall purity as '*Sum of base pairs coming from the most abundant genome in each predicted genome MAG divided by the sum of base pairs in all predicted bins. ...*'.

Purity and completeness values are reported as the per data set average, unless specified otherwise.

For the real-world benchmarks the average proportion of bp recovered or the number of MAGs recovered is reported together with the standard error of the mean (SEM).

Another metric used is the adjusted Rand index (ARI), which is a commonly used metric to measure how similar two datasets are by also applying a multiple testing correcting.

Trying to make the comparisons between different binning methods as fair and transparent as possible, we report all metrics derived from the CheckM-filtered binning results, unless specified otherwise.

To assess the intersections of MAGs formed by the different binning methods on multi-sample datasets, genomes were counted separately for each sample. To this end, the gold standard genome name was concatenated with the sample id to yield unique identifiers for each genome in each sample. All other figures were created using Python's *matplotlib* [20] and Seaborn [54] libraries, as well as UpSetPlot [28] for the upset plot. Remaining data analyses were performed and table outputs created using the Python *NumPy* and *pandas* libraries.

Results

Performance on synthetic data sets

To assess *binny*'s performance, six datasets from the Critical Assessment of Metagenome Interpretation (CAMI) initiative were chosen: the high complexity toy dataset of the first CAMI iteration to investigate how *binny* performs on very large, complex data sets, and the five toy human microbiome data sets of the second CAMI iteration to evaluate the performance on a wide range of microbiome sizes and complexities.

Over all six data sets (54 samples), *binny* recovered 33.0% (SEM 2.5%) of the reference genomes in the samples as NC MAGs (n=1490) and 39.0% (SEM 2.7%) as HQ MAGs (n=1926), with median recall values of 25.5% and 33.7%, respectively (Figure 2, Supplementary Table 4). In total, 41.1% of the reference genomes where recovered at a purity of 98.8% with an ARI of 0.983 (Supplementary Figure 2, Supplementary Table 5).

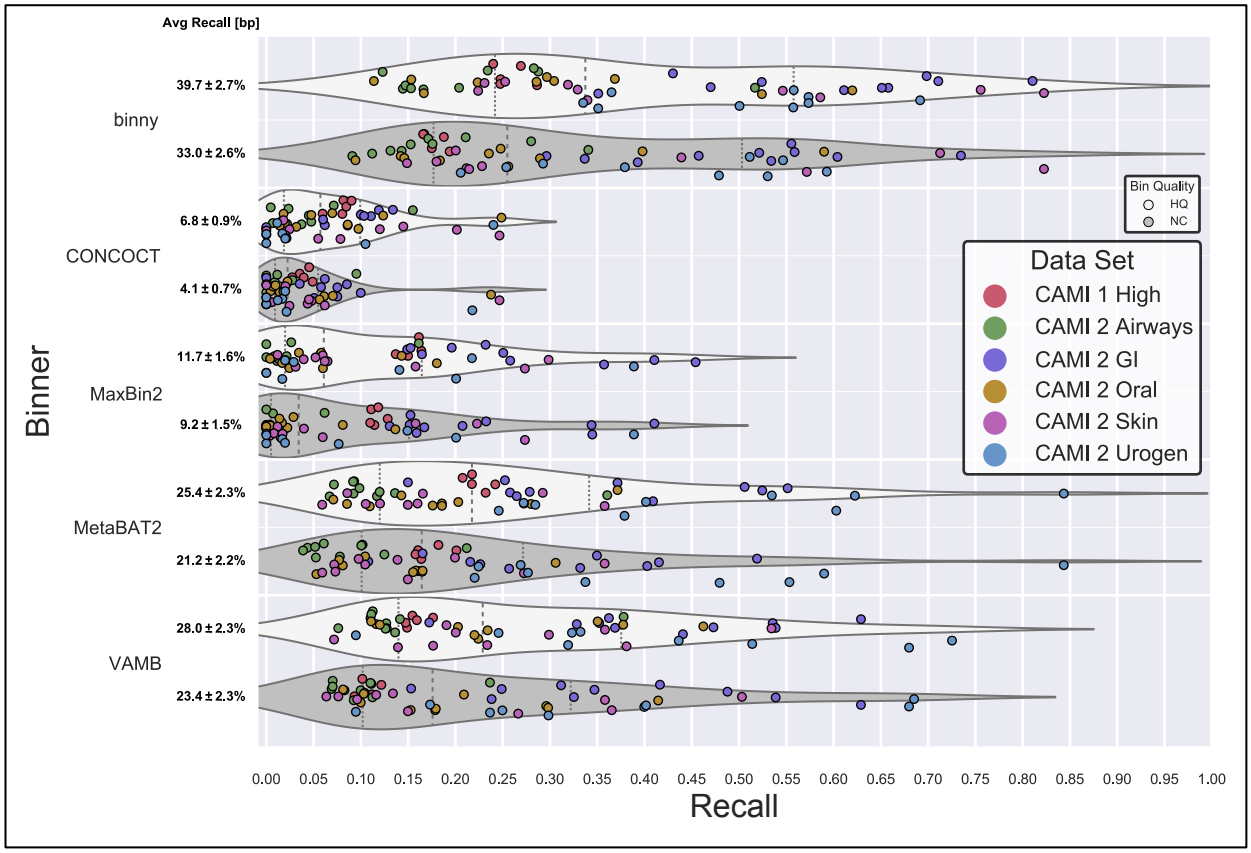

Figure 2. Performance of binning methods on CAMI data sets. Recall as HQ and NC MAGs per binning method per sample from the six CAMI data sets. The average recall is shown with the standard errors of the mean.

For the high complexity data set, *binny* recovered 27.7% of the total reference genomes with a purity of 98,0% , and an ARI of 0.974. In total 313 NC and 473 HC MAGs were recovered (Supplementary Figure 3, Supplementary Table 6).

The lowest recall was observed for the CAMI 2 Airways data set with 23.5%, a purity of 98,8% , and an ARI of 0.974 (Supplementary Figure 4), while the highest recall of 60.9%, with a purity of 98,9% , and an ARI of 0.982 (Supplementary Figure 5) was reached with the CAMI 2 Gastrointestinal (GI) data set. For the other three data sets, *binny* achieved the following respective recall, purity and ARI numbers: 51.6%, 98,8%, and 0.984 (CAMI 2 Urogenital); 45.8%, 98,8%, and 0.985 (CAMI 2 Skin); and 31.4%, 99,0%, and 0.986 (CAMI 2 Oral) (Supplementary Figure 6-8, Supplementary Table 6; for detailed metrics for MAGs and samples see Supplementary Tables 7 and 8, respectively.).

Read recruitment of the unfiltered *binny* output was, with 70%, on average higher than the recall based on the assembly length. The highest recruitment was achieved for the GI data set sample 12 with 89.7%, while the lowest was observed for the skin dataset sample 19 (40.5%). Notably, a significant proportion of

the reads recruited were mapped to single contig MAGs for the CAMI 2 datasets, while for the CAMI 1 datasets, only about a third of the reads recruited by binned contigs, mapped to single contig MAGs (Table 1, Supplementary Table 9).

Table 1. *binny* MAG read recruitment over 54 CAMI samples. scMAG: Single contig MAG, std: Standard deviation, 25/50/75%: Percentiles, min/max: Minimum and maximum values.

	reads mapping	scMAG reads mapping	read recall	scMAG read recall	sample total reads
mean	31,332,990	21,457,013	0.702	0.577	44,050,279
std	25,875,150	5,235,221	0.129	0.173	33,875,702
min	13,479,857	9,881,682	0.404	0.192	33,312,040
25%	20,133,589	17,735,923	0.604	0.491	33,330,381
50%	23,858,140	20,821,038	0.716	0.606	33,331,571
75%	28,445,584	25,475,766	0.797	0.667	33,332,082
max	115,413,829	33,881,406	0.897	0.886	149,120,056

Running *binny* with multiple depth files

When assessing the performance on co-assembled datasets, *binny* had a recall of 50.6% over the CAMI datasets with a purity of 98.9%. In total 967 NC MAGs were produced, 288 of which contained 10 or more contigs (Supplementary Figure 9-11). The highest recall was achieved for the CAMI 2 Gastrointestinal co-assembly with 73.5% and a purity of 99.0%, while the worst performance was observed for the CAMI 2 Airways dataset with a recall of 27.9% and a purity of 98.5% (Supplementary Table 10,11).

To test to which degree *binny* makes use of the information from the multiple read depth files per co-assembly, *binny* was additionally run with only one depth file per co-assembly. *binny* using all available depth files had a 4.7% higher recall at the same purity, leading to a recovery of 10% more NC MAGs (95) in total and 32% more NC MAGs (91) of contig sizes larger than 10 (Supplementary Figure 9-11, Supplementary Table 10,11).

Effect of masking potentially disruptive sequence regions

To test the effect of masking potentially disruptive sequences, we also ran *binny* on the six CAMI co-assembly data sets without the masking procedure. The unmasked run did not differ substantially from the one with the default settings regarding assembly recall and purity. While the purity remained unchanged, the recall was reduced by 0.3%, resulting in two fewer NC MAGs (Supplementary Table 12). The amount of MAGs recovered matching the MIMAG standard was reduced by 4 from 636 to 632.

Run time

For all experiments, *binny* was run on compute nodes equipped with Xeon E5-2680v4 CPUs allocating 14 cores and 56 GB of RAM. For the CAMI samples, the complete *binny* pipeline took on average 183 minutes to run, with a max of 714 minutes for sample 3 of the CAMI 1 high complexity data set. The Prokka annotations took on average 32 minutes, the Mantis annotations on average 45 minutes, and *binny* on average 105 minutes (Table 2).

binny outperformed state-of-the-art binning methods on synthetic data sets

Over all six CAMI datasets *binny* recovered per sample the highest portion of the assembly (bps) as HQ (39.7%) or NC (33.0%) MAGs, followed by VAMB (28.0%, 23.4%) and MetaBAT2 (25.4%, 21.2%). Additionally, *binny* showed the highest median MAG counts with 41.1%, 84.6% more NC and 21.7%, 75.0% more HQ MAGs than VAMB and MetaBAT2, respectively Figure 2, Supplementary Table 4).

Table 2. *binny* run-time statistics for the CAMI benchmark. Time is shown in minutes. scMAG: Single contig MAG, std: Standard deviation, 25/50/75%: Percentiles, min/max: Minimum and maximum values.

	total	**Prokka**	**Mantis**	**binny**
mean	183	32	45	105
std	163	26	26	116
min	29	6	10	10
25%	91	18	27	44
50%	134	23	37	64
75%	192	35	54	116
max	714	113	120	516

binny was the only binning method that resulted in high purity (98.3%) and high ARI (0.977) output over all datasets without additional CheckM filtering. Using CheckM filtering, *binny*'s purity and ARI were increased by 0.5% and 0.007, respectively, while the assembly recall was decreased by 0.9% (Supplementary Figure 2 b,c, Supplementary Table 5). The second best binning tool, VAMB, had a purity of 75.3% natively and an ARI of 0.675. After CheckM filtering, the purity and ARI of VAMB was the highest among binning methods (99.5% purity and an ARI of 0.994, respectively), but at the same time the recall was reduced from 56.7% to 28.5% (Supplementary Figure 2 b,c, Supplementary Table 5).

For detailed metrics on the MAGs and samples see Supplementary Tables 7 and 8, respectively.

binny also outperformed the other binning methods for the individual data sets, followed by MetaBAT2 for the CAMI 1 High Complexity data set Figure 2, Supplementary Figure 3-8, Supplementary Table 6).

Additionally, *binny* surpassed the second and third best performing binning methods of the single sample assembly benchmark, VAMB and MetaBAT2, on the co-assembly versions of the six data sets. It recovered 64.2% more NC MAGs of any contig number and still 22.0% more NC MAGs containing more than ten contigs than the second best performer, MetaBAT2 (Supplementary Figure 9-11, Supplementary Table 10,11).

Lastly, we assessed the amount of MAGs meeting the MIMAG high quality draft standard. *binny* recovered the most MAGs of that quality for each CAMI data set, recovering in total 38.9% more than the second best method (Table 3).

Table 3. MIMAG High Quality Draft MAGs recovered by binning methods. Rows represent values for the 6 CAMI data sets, the sum over all of them, and the number for the real-world benchmark data (IMG). Bold values show the highest count per data set, underlined values the second highest.

	High	**Airways**	**GI**	**Oral**	**Skin**	**Urogen**	**Total**	**IMG**
binny	**157**	**182**	**186**	**240**	**209**	**136**	**1110**	**490**
CONCOCT	17	10	19	37	18	6	107	142
MaxBin2	85	5	79	20	26	16	231	<u>422</u>
MetaBAT2	<u>144</u>	81	100	134	85	111	655	417
VAMB	107	<u>121</u>	<u>138</u>	<u>197</u>	<u>123</u>	<u>113</u>	<u>799</u>	406

binny recovered unique genomes as well as ones also found by other binning methods

To evaluate the performance of different binning tools, it is also of interest to see how much unique information is recovered by each individual binning method. *binny* yielded substantially more unique NC MAGs (402) than the other methods for the CAMI data sets, followed by VAMB (134) and MetaBAT2 (39). Additionally, the two largest sets of MAGs shared by two binning methods are both *binny* sharing MAGs with VAMB

(206) or MetaBAT2 (168), respectively (Figure 3). For the HQ genomes, similar results were observed: *binny* recovered the most unique MAGs and was present in almost all of the intersections with the largest numbers of genomes (Supplementary Figure 12).

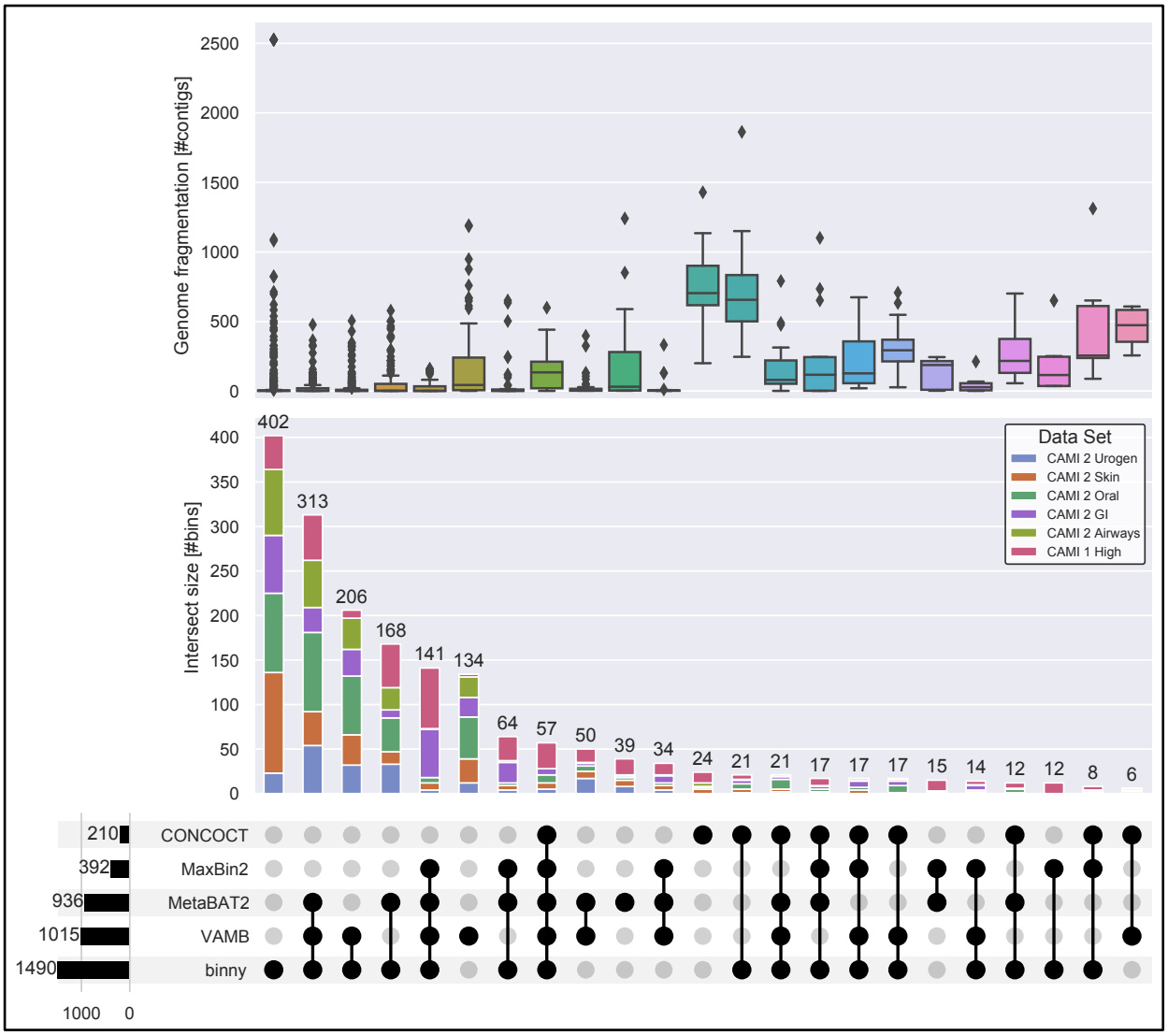

Figure 3. Intersections of recovered CAMI NC MAGs and reference genome fragmentation grade. Intersections of NC MAGs of 5 CheckM-filtered binning methods for 54 samples from 6 CAMI data sets. Upper panel: Reference genome fragmentation in number of contigs. Middle panel: Intersection size in number of NC bins with proportions of bins stemming from the 6 CAMI data sets. Lower panel: Number of bins per binning method on the left, intersection types in the centre.

binny produced high quality MAGs from contiguous as well as highly fragmented genomes

Next, we assessed the ability of different binning methods to recover genomes at different fragmentation levels. *binny* recovered more highly fragmented genomes (defined here as genomes with more than 500 contigs) than MaxBin2, MetaBAT2 and VAMB. Only CONCOCT, which on other metrics performed poorly, recovered more highly fragmented genomes than *binny*, while both methods shared a large portion of fragmented genomes (Figure 3).

Several of the CAMI samples contain a larger amounts of single-contig or almost contiguous genomes than is commonly observed in real-world samples. To evaluate *binny*'s performance on more realistically fragmented genomes, we considered the subset of genomes which consisted of more than ten contigs. Here,

binny also produced substantially more NC (28.4%) and HQ (24.5%) MAGs than VAMB overall, as well as the most unique MAGs found between *binny*, VAMB and MetaBAT2. While VAMB recovered more genomes for the CAMI 2 oral dataset, *binny* recovered substantially more genomes from the CAMI 1 high complexity data set (Supplementary Figure 13). Looking at the assembly recall, *binny* again showed the best performance and, while *binny* outperformed the other binning methods for the CAMI 1 data set, VAMB had a small lead over *binny* regarding the recall for the oral data set (Supplementary Figure 14).

binny recovered a larger number of high-quality MAGs than other binning methods for real-world assemblies from different environments.

When benchmarking binning tools with real-world data from a wide variety of environments, *binny* recovered on average the largest amount of the assembly (bp) as NC (16.7%) and HQ (24.8%) MAGs, producing 13.0% more NC and 22.5%, more HQ MAGs than the second best binning tool, VAMB. While MetaBAT2 had an comparable median assembly recovery as NC MAGs, *binny* showed substantially higher median value for HQ MAGs. As in the CAMI benchmarks, CONCOCT showed the lowest recall for both NC and HQ MAGs, while MaxBin2 performed comparatively better with this data than in the CAMI benchmark (Figure 4, Supplementary Table 13, 14). When counting the recovered MIMAG high quality genomes, *binny* produced 490 MAGs, 16% more than the second best binning method, MaxBin2 (422; Table 3).

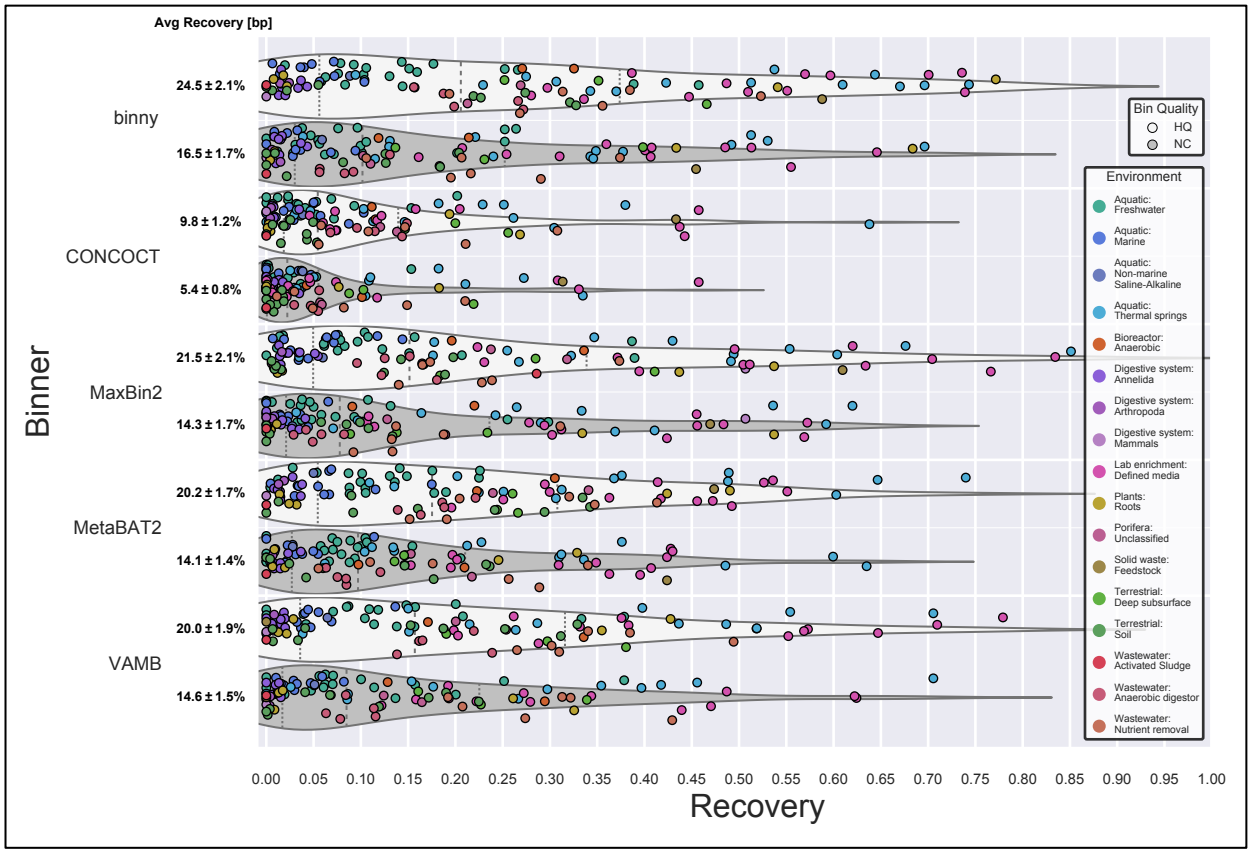

Figure 4. Performance of binning methods on real-world data sets from various environments. Assembly recovery as HQ and NC MAGs per binning method per sample from 105 real-world samples. The average recovery is shown with the standard errors of the mean.

binny improved ensemble binning/refinement approaches

To test if *binny* is able to improve refinements in combination with other binning methods, we ran the two most popular automatic refinement tools, DASTool and MetaWrap, on the 54 samples of the six CAMI data sets, combining all other tested binning methods either with or without *binny*.

When *binny* was excluded, a 8.9% and 9.6% lower recall was observed for DASTool (42.2%) and MetaWrap (38.9%), respectively (Supplementary Figure 15 b, c). *binny* on its own recovered more NC MAGs than both refinement tools without *binny* input. Including *binny*, MetaWrap was able to recover 15.6% more NC MAGs (1723) than *binny* on its own, while DASTool produced 12.6% more NC MAGs (1678) (Supplementary Figure 15 a, Supplementary Figure 16, Supplementary Table 15, 16). Including only MAGs with more than 10 contigs, the refiners without *binny* performed slightly better than *binny* on its own. As expected, also for the MAGs with more than 10 contigs, the runs including all five binning methods showed the highest performance overall, with MetaWrap recovering the most MAGs (Supplementary Figure 17). Of note, while MetaWrap produced almost no heavily contaminated MAGs, DASTool did output large numbers of MAGs with very low purity, despite showing over the entire CAMI benchmark data high purity (Supplementary Figure 15 d, Supplementary Table 15, 16)).

Discussion

binny implements a fully automated binning tool, recovering unique information in form of high quality, pure MAGs. *binny* combines *k*-mer-composition, read coverage, and lineage-specific marker gene sets for iterative, non-linear dimension reduction of genomic signatures and subsequent automated contig clustering with cluster assessment. The low dimensional embedding strategy to reduce large amounts of features has been used before for binning to aid the clustering of contigs [12, 27], as clustering algorithms perform better in fewer dimensions, because distance information becomes increasingly imprecise at higher dimensions and the chance of random correlation between features rises [25]. While there are already binning methods available that, to some extent, make use of marker genes ([55], [31]) and also lower dimensional embedding of contig features ([31]), *binny* uses a new and unique iterative clustering strategy. Importantly, it assesses clusters of contigs during its iterations, recognizing when further splitting of clusters is necessary. As this lowers the complexity of each clustering task, *binny* recovers genomes that might not be separable in a single clustering run. In combination with the ability to incorporate also short informative contigs, *binny* is able to deal with highly fragmented genomes as shown for the CAMI samples – of the tested binning methods only CONCOCT was able to do so as well. Additionally, *binny* performed also particularly well at recovering highly contiguous CAMI genomes. This can again be attributed to the ability to assess purity and completeness using the marker gene approach, here in particular for single-contig genomes. The marker gene approach in addition allows to bin also smaller contigs containing marker genes that would be discarded by most other binning methods due to their applied contig length thresholds. Although contigs below 1000 bp rarely made up more than 5% of the total recovered MAGs, in the cases where *binny* recruited more short contigs to MAGs, it did so with high precision (Supplementary Table 17).

Being able to iteratively extract high-quality MAGs, *binny* is also beneficial for large, complex communities as shown for the five CAMI 1 High complexity samples. The extraction of MAGs over multiple rounds reduces sample complexity and allows the recovery of genomes that would not have been binned in a single iteration.

binny performs well on co-assemblies and makes use of the additional information provided by contig depth of coverage data from additionally mapped samples. This is in line with previous studies observing additional discriminatory power of differential coverage depth compared to only sequence-based features [4, 22].

While the difference in performance from masking potentially disruptive sequences is minimal, we believe that it is a sound approach in principle. One key reason for the small impact might be the prominent role played by the contig read depth in the embedding and clustering, which could outweigh the masked *k*-mer profile. Masking reads mapping to the disruptive regions, also modifying the depth information might increase its effectiveness and could be implemented in future versions.

Conclusions

In conclusion, we demonstrate that *binny* outperforms currently available, state-of-the-art popular binning methods based on established evaluation metrics, recovering unique, high-quality MAGs from simple and complex samples alike, while being able to handle contiguous, as well as fragmented genomes. In consequence, *binny* can add valuable new unique information when using combinations of binning methods together with binning refinement approaches, enabling researchers to further improve the recovery of genomes from their metagenomes.

Additional file 1 — supplementary_figures.pdf

Contains supplementary figures 1-17.

Additional file 2 — supplementary_tables.xlsx

Contains supplementary tables 1-17 as Excel sheets.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

OH, PM, and AHB designed this study. OH and AHB created the application. OH performed all experiments. OH,PM and AHB wrote the manuscript; PQ and PW contributed to the review of the manuscript before submission. All authors read and approved the manuscript.

Acknowledgements

The authors would like to thank Francesco Delogu, Benoit Kunath, for scientific discussions. Development and data analysis were performed using the research cluster of the Faculty of Science at the University of Amsterdam and the HPC facilities of the University of Luxembourg, both of whose administrators are thanked for excellent support. We also thank Adrian Fritz of the CAMI team for support.

Funding

This work was supported by the Luxembourg National Research Fund (FNR) under grant PRIDE/11823097 and the European Research Council (ERC-CoG 863664).

Availability of data and material

The latest version of *binny* can be found at `https://github.com/a-h-b/binny`. The version used in this study and related data is available at `https://github.com/ohickl/binny_manuscript` and `https://doi.org/10.5281/zenodo.5779794`.

References

1. C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In G. Goos, J. Hartmanis, J. van Leeuwen, J. Van den Bussche, and V. Vianu, editors, *Database Theory — ICDT 2001*, volume 1973, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. Series Title: Lecture Notes in Computer Science.

2. A. Almeida, A. L. Mitchell, A. Tarkowska, and R. D. Finn. Benchmarking taxonomic assignments based on 16S rRNA gene profiling of the microbiota from commonly sampled environments. *GigaScience*, 7(5), May 2018.

3. A. Almeida, S. Nayfach, M. Boland, F. Strozzi, M. Beracochea, Z. J. Shi, K. S. Pollard, E. Sakharova, D. H. Parks, P. Hugenholtz, N. Segata, N. C. Kyrpides, and R. D. Finn. A unified catalog of 204,938 reference genomes from the human gut microbiome. *Nature Biotechnology*, 39(1):105–114, Jan. 2021.

4. J. Alneberg, B. S. Bjarnason, I. de Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson, and C. Quince. Binning metagenomic contigs by coverage and composition. *Nature Methods*, 11(11):1144–1146, Nov. 2014.

5. C. Bland, T. L. Ramsey, F. Sabree, M. Lowe, K. Brown, N. C. Kyrpides, and P. Hugenholtz. CRISPR recognition tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats. *BMC bioinformatics*, 8:209, June 2007.

6. T. L. Bornemann, S. P. Esser, T. L. Stach, T. Burg, and A. J. Probst. uBin – a manual refining tool for metagenomic bins designed for educational purposes. preprint, Genomics, July 2020.

7. R. M. Bowers, N. C. Kyrpides, R. Stepanauskas, M. Harmon-Smith, D. Doud, T. B. K. Reddy, F. Schulz, J. Jarett, A. R. Rivers, E. A. Eloe-Fadrosh, S. G. Tringe, N. N. Ivanova, A. Copeland, A. Clum, E. D. Becraft, R. R. Malmstrom, B. Birren, M. Podar, P. Bork, G. M. Weinstock, G. M. Garrity, J. A. Dodsworth, S. Yooseph, G. Sutton, F. O. Glöckner, J. A. Gilbert, W. C. Nelson, S. J. Hallam, S. P. Jungbluth, T. J. G. Ettema, S. Tighe, K. T. Konstantinidis, W.-T. Liu, B. J. Baker, T. Rattei, J. A. Eisen, B. Hedlund, K. D. McMahon, N. Fierer, R. Knight, R. Finn, G. Cochrane, I. Karsch-Mizrachi, G. W. Tyson, C. Rinke, Genome Standards Consortium, A. Lapidus, F. Meyer, P. Yilmaz, D. H. Parks, A. M. Eren, L. Schriml, J. F. Banfield, P. Hugenholtz, and T. Woyke. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nature Biotechnology*, 35(8):725–731, Aug. 2017.

8. B. Broeksema, M. Calusinska, F. McGee, K. Winter, F. Bongiovanni, X. Goux, P. Wilmes, P. Delfosse, and M. Ghoniem. ICoVeR - an interactive visualization tool for verification and refinement of metagenomic bins. *BMC bioinformatics*, 18(1):233, May 2017.

9. C. T. Brown, L. A. Hug, B. C. Thomas, I. Sharon, C. J. Castelle, A. Singh, M. J. Wilkins, K. C. Wrighton, K. H. Williams, and J. F. Banfield. Unusual biology across a group comprising more than 15% of domain Bacteria. *Nature*, 523(7559):208–211, July 2015.

10. B. Buchfink, C. Xie, and D. H. Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, Jan. 2015.

11. R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7819, pages 160–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Lecture Notes in Computer Science.

12. J. Ceballos, L. Ariza-Jiménez, and N. Pinel. Standardized approaches for assessing metagenomic contig binning performance from barnes-hut t-stochastic neighbor embeddings. In C. A. González Díaz, C. Chapa González, E. Laciar Leber, H. A. Vélez, N. P. Puente, D.-L. Flores, A. O. Andrade, H. A. Galván, F. Martínez, R. García, C. J. Trujillo, and A. R. Mejía, editors, *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 761–768, Cham, 2020. Springer International Publishing.

13. L.-X. Chen, K. Anantharaman, A. Shaiber, A. M. Eren, and J. F. Banfield. Accurate and complete genomes from metagenomes. *Genome Research*, 30(3):315–333, Mar. 2020.

14. T. O. Delmont, C. Quince, A. Shaiber, O. C. Esen, S. T. Lee, M. S. Rappé, S. L. McLellan, S. Luecker, and A. M. Eren. Nitrogen-fixing populations of Planctomycetes and Proteobacteria are abundant in surface ocean metagenomes. *Nature Microbiology*, 3(7):804–813, July 2018.

15. A. M. Eren, O. C. Esen, C. Quince, J. H. Vineis, H. G. Morrison, M. L. Sogin, and T. O. Delmont. Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ*, 3:e1319, 2015.

16. A. M. Eren, E. Kiefl, A. Shaiber, I. Veseli, S. E. Miller, M. S. Schechter, I. Fink, J. N. Pan, M. Yousef, E. C. Fogarty, F. Trigodet, A. R. Watson, O. C. Esen, R. M. Moore, Q. Clayssen, M. D. Lee, V. Kivenson, E. D. Graham, B. D. Merrill, A. Karkman, D. Blankenberg, J. M. Eppley, A. Sjödin, J. J. Scott, X. Vázquez-Campos, L. J. McKay, E. A. McDaniel, S. L. R. Stevens, R. E. Anderson, J. Fuessel, A. Fernandez-Guerra, L. Maignien, T. O. Delmont, and A. D. Willis. Community-led, integrated, reproducible multi-omics with anvi'o. *Nature Microbiology*, 6(1):3–6, Jan. 2021.

17. A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

18. A. Heintz-Buschart, P. May, C. C. Laczny, L. A. Lebrun, C. Bellora, A. Krishna, L. Wampach, J. G. Schneider, A. Hogan, C. de Beaufort, and P. Wilmes. Integrated multi-omics of the human gut microbiome in a case study of familial type 1 diabetes. *Nature Microbiology*, 2:16180, Oct. 2016.

19. M. Herold, S. Martínez Arbas, S. Narayanasamy, A. R. Sheik, L. A. K. Kleine-Borgmann, L. A. Lebrun, B. J. Kunath, H. Roume, I. Bessarab, R. B. H. Williams, J. D. Gillece, J. M. Schupp, P. S. Keim, C. Jäger, M. R. Hoopmann, R. L. Moritz, Y. Ye, S. Li, H. Tang, A. Heintz-Buschart, P. May, E. E. L. Muller, C. C. Laczny, and P. Wilmes. Integration of time-series meta-omics data reveals how microbial ecosystems respond to disturbance. *Nature Communications*, 11(1):5281, Oct. 2020.

20. J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

21. D. Hyatt, G.-L. Chen, P. F. Locascio, M. L. Land, F. W. Larimer, and L. J. Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, 11:119, Mar. 2010.

22. D. D. Kang, F. Li, E. Kirton, A. Thomas, R. Egan, H. An, and Z. Wang. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, 7:e7359, 2019.

23. N. Karcher, E. Nigro, M. Punčochář, A. Blanco-Míguez, M. Ciciani, P. Manghi, M. Zolfo, F. Cumbo, S. Manara, D. Golzato, A. Cereseto, M. Arumugam, T. P. N. Bui, H. L. P. Tytgat, M. Valles-Colomer, W. M. de Vos, and N. Segata. Genomic diversity and ecology of human-associated Akkermansia species in the gut microbiome revealed by extensive metagenomic assembly. *Genome Biology*, 22(1):209, July 2021.

24. D. Kobak and P. Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, Dec. 2019.

25. H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, Mar. 2009.

26. J. Köster and S. Rahmann. Snakemake-a scalable bioinformatics workflow engine. *Bioinformatics (Oxford, England)*, 34(20):3600, Oct. 2018.

27. C. C. Laczny, T. Sternal, V. Plugaru, P. Gawron, A. Atashpendar, H. H. Margossian, S. Coronado, L. v. der Maaten, N. Vlassis, and P. Wilmes. VizBin - an application for reference-independent visualization and human-augmented binning of metagenomic data. *Microbiome*, 3(1):1, 2015.

28. A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, Dec. 2014.

29. H. Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv:1303.3997 [q-bio]*, May 2013. arXiv: 1303.3997.

30. W. Li, K. R. O'Neill, D. H. Haft, M. DiCuccio, V. Chetvernin, A. Badretdin, G. Coulouris, F. Chitsaz, M. K. Derbyshire, A. S. Durkin, N. R. Gonzales, M. Gwadz, C. J. Lanczycki, J. S. Song, N. Thanki, J. Wang, R. A. Yamashita, M. Yang, C. Zheng, A. Marchler-Bauer, and F. Thibaud-Nissen. RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation. *Nucleic Acids Research*, 49(D1):D1020–D1028, Jan. 2021.

31. H.-H. Lin and Y.-C. Liao. Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. *Scientific Reports*, 6:24175, nov 2016.

32. G. C. Linderman and S. Steinerberger. Clustering with t-SNE, Provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332, Jan. 2019.

33. F. Meyer, P. Hofmann, P. Belmann, R. Garrido-Oter, A. Fritz, A. Sczyrba, and A. C. McHardy. AMBER: Assessment of Metagenome BinnERs. *GigaScience*, 7(6), June 2018.

34. F. Meyer, T.-R. Lesker, D. Koslicki, A. Fritz, A. Gurevich, A. E. Darling, A. Sczyrba, A. Bremges, and A. C. McHardy. Tutorial: assessing metagenomics software with the CAMI benchmarking toolkit. *Nature Protocols*, 16(4):1785–1801, Apr. 2021.

35. A. Meziti, L. M. Rodriguez-R, J. K. Hatt, A. Peña-Gonzalez, K. Levy, and K. T. Konstantinidis. The Reliability of Metagenome-Assembled Genomes (MAGs) in Representing Natural Populations: Insights from Comparing MAGs against Isolate Genomes Derived from the Same Fecal Sample. *Applied and Environmental Microbiology*, 87(6):e02593–20, Feb. 2021.

36. J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G. Salazar, E. L. L. Sonnhammer, S. C. E. Tosatto, L. Paladin, S. Raj, L. J. Richardson, R. D. Finn, and A. Bateman. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, Jan. 2021.

37. A. L. Mitchell, A. Almeida, M. Beracochea, M. Boland, J. Burgin, G. Cochrane, M. R. Crusoe, V. Kale, S. C. Potter, L. J. Richardson, E. Sakharova, M. Scheremetjew, A. Korobeynikov, A. Shlemov, O. Kunyavskaya, A. Lapidus, and R. D. Finn. MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*, page gkz1035, Nov. 2019.

38. S.-I. Na, Y. O. Kim, S.-H. Yoon, S.-M. Ha, I. Baek, and J. Chun. UBCG: Up-to-date bacterial core gene set and pipeline for phylogenomic tree reconstruction. *Journal of Microbiology (Seoul, Korea)*, 56(4):280–285, Apr. 2018.

39. S. Nayfach, S. Roux, R. Seshadri, D. Udwary, N. Varghese, F. Schulz, D. Wu, D. Paez-Espino, I.-M. Chen, M. Huntemann, K. Palaniappan, J. Ladau, S. Mukherjee, T. B. K. Reddy, T. Nielsen, E. Kirton, J. P. Faria, J. N. Edirisinghe, C. S. Henry, S. P. Jungbluth, D. Chivian, P. Dehal, E. M. Wood-Charlson, A. P. Arkin, S. G. Tringe, A. Visel, IMG/M Data Consortium, T. Woyke, N. J. Mouncey, N. N. Ivanova, N. C. Kyrpides, and E. A. Eloe-Fadrosh. A genomic catalog of Earth's microbiomes. *Nature Biotechnology*, 39(4):499–509, Apr. 2021.

40. F. N. New and I. L. Brito. What Is Metagenomics Teaching Us, and What Is Missed? *Annual Review of Microbiology*, 74:117–135, Sept. 2020.

41. J. N. Nissen, J. Johansen, R. L. Allesøe, C. K. Sønderby, J. J. A. Armenteros, C. H. Grønbech, L. J. Jensen, H. B. Nielsen, T. N. Petersen, O. Winther, and S. Rasmussen. Improved metagenome binning and assembly using deep variational autoencoders. *Nature Biotechnology*, 39(5):555–560, May 2021.

42. D. H. Parks, M. Imelfort, C. T. Skennerton, P. Hugenholtz, and G. W. Tyson. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, 25(7):1043–1055, July 2015.

43. P. G. Poličar, M. Stražar, and B. Zupan. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv*, 2019.

44. P. Queirós, F. Delogu, O. Hickl, P. May, and P. Wilmes. Mantis: flexible and consensus-driven genome annotation. *GigaScience*, 10(6):giab042, June 2021.

45. C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata. Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology*, 35(9):833–844, Sept. 2017.

46. A. R. Quinlan and I. M. Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, 26(6):841–842, 2010.

47. C. Rinke, P. Schwientek, A. Sczyrba, N. N. Ivanova, I. J. Anderson, J.-F. Cheng, A. Darling, S. Malfatti, B. K. Swan, E. A. Gies, J. A. Dodsworth, B. P. Hedlund, G. Tsiamis, S. M. Sievert, W.-T. Liu, J. A. Eisen, S. J. Hallam, N. C. Kyrpides, R. Stepanauskas, E. M. Rubin, P. Hugenholtz, and T. Woyke. Insights into the phylogeny and coding potential of microbial dark matter. *Nature*, 499(7459):431–437, July 2013.

48. A. Sczyrba, P. Hofmann, P. Belmann, D. Koslicki, S. Janssen, J. Dröge, I. Gregor, S. Majda, J. Fiedler, E. Dahms, A. Bremges, A. Fritz, R. Garrido-Oter, T. S. Jørgensen, N. Shapiro, P. D. Blood, A. Gurevich, Y. Bai, D. Turaev, M. Z. DeMaere, R. Chikhi, N. Nagarajan, C. Quince, F. Meyer, M. Balvočiūtė, L. H. Hansen, S. J. Sørensen, B. K. H. Chia, B. Denis, J. L. Froula, Z. Wang, R. Egan, D. Don Kang, J. J. Cook, C. Deltel, M. Beckstette, C. Lemaitre, P. Peterlongo, G. Rizk, D. Lavenier, Y.-W. Wu, S. W. Singer, C. Jain, M. Strous, H. Klingenberg, P. Meinicke, M. D. Barton, T. Lingner, H.-H. Lin, Y.-C. Liao, G. G. Z. Silva, D. A. Cuevas, R. A. Edwards, S. Saha, V. C. Piro, B. Y. Renard, M. Pop, H.-P. Klenk, M. Göker, N. C. Kyrpides, T. Woyke, J. A. Vorholt, P. Schulze-Lefert, E. M. Rubin, A. E. Darling, T. Rattei, and A. C. McHardy. Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. *Nature Methods*, 14(11):1063–1071, Nov. 2017.

49. T. Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics (Oxford, England)*, 30(14):2068–2069, July 2014.

50. L. Shen, Y. Liu, M. A. Allen, B. Xu, N. Wang, T. J. Williams, F. Wang, Y. Zhou, Q. Liu, and R. Cavicchioli. Linking genomic and physiological characteristics of psychrophilic arthrobacter to metagenomic data to explain global environmental distribution. *Microbiome*, 9(1):136, jun 2021.

51. C. M. K. Sieber, A. J. Probst, A. Sharrar, B. C. Thomas, M. Hess, S. G. Tringe, and J. F. Banfield. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nature Microbiology*, 3(7):836–843, July 2018.

52. A. Tett, K. D. Huang, F. Asnicar, H. Fehlner-Peach, E. Pasolli, N. Karcher, F. Armanini, P. Manghi, K. Bonham, M. Zolfo, F. De Filippis, C. Magnabosco, R. Bonneau, J. Lusingu, J. Amuasi, K. Reinhard, T. Rattei, F. Boulund, L. Engstrand, A. Zink, M. C. Collado, D. R. Littman, D. Eibach, D. Ercolini, O. Rota-Stabelli, C. Huttenhower, F. Maixner, and N. Segata. The Prevotella copri Complex Comprises Four Distinct Clades Underrepresented in Westernized Populations. *Cell Host & Microbe*, 26(5):666–679.e7, Nov. 2019.

53. G. V. Uritskiy, J. DiRuggiero, and J. Taylor. MetaWRAP-a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome*, 6(1):158, Sept. 2018.

54. M. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, Apr. 2021.

55. Y.-W. Wu, B. A. Simmons, and S. W. Singer. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, 32(4):605–607, Feb. 2016.

56. Y. Yue, H. Huang, Z. Qi, H.-M. Dou, X.-Y. Liu, T.-F. Han, Y. Chen, X.-J. Song, Y.-H. Zhang, and J. Tu. Evaluating metagenomics tools for genome binning with real metagenomic datasets and CAMI datasets. *BMC bioinformatics*, 21(1):334, July 2020.

57. K. Zaremba-Niedzwiedzka, E. F. Caceres, J. H. Saw, D. Bäckström, L. Juzokaite, E. Vancaester, K. W. Seitz, K. Anantharaman, P. Starnawski, K. U. Kjeldsen, M. B. Stott, T. Nunoura, J. F. Banfield, A. Schramm, B. J. Baker, A. Spang, and T. J. G. Ettema. Asgard archaea illuminate the origin of eukaryotic cellular complexity. *Nature*, 541(7637):353–358, Jan. 2017.