(GIGA)n SCIENCE

AMBER: Assessment of Metagenome BinnERs

Fernando Meyer[1,2], Peter Hofmann[1,2], Peter Belmann[1,2,3,4], Ruben Garrido-Oter[5,6], Adrian Fritz[1,2], Alexander Sczyrba[3,4] and Alice C. McHardy[1,2,*]

[1]Department of Computational Biology of Infection Research, Helmholtz Centre for Infection Research (HZI), Braunschweig, Germany; [2]Braunschweig Integrated Centre of Systems Biology (BRICS), Braunschweig, Germany; [3]Faculty of Technology, Bielefeld University, Bielefeld, Germany; [4]Center for Biotechnology, Bielefeld University, Bielefeld, Germany; [5]Department of Plant Microbe Interactions, Max Planck Institute for Plant Breeding Research, Cologne, Germany; [6]Cluster of Excellence on Plant Sciences (CEPLAS)

*Correspondence: Alice.McHardy@helmholtz-hzi.de.

Abstract

Reconstructing the genomes of microbial community members is key to the interpretation of shotgun metagenome samples. Genome binning programs deconvolute reads or assembled contigs of such samples into individual bins, but assessing their quality is difficult due to the lack of evaluation software and standardized metrics. We present AMBER, an evaluation package for the comparative assessment of genome reconstructions from metagenome benchmark data sets. It calculates the performance metrics and comparative visualizations used in the first benchmarking challenge of the Initiative for the Critical Assessment of Metagenome Interpretation (CAMI). As an application, we show the outputs of AMBER for eleven different binning programs on two CAMI benchmark data sets. AMBER is implemented in Python and available under the Apache 2.0 license on GitHub (https://github.com/CAMI-challenge/AMBER).

Keywords: binning, metagenomics, benchmarking, performance metrics, bioboxes

Introduction

Metagenomics allows studying microbial communities and their members by shotgun sequencing. Evolutionary divergence and abundances of these members can vary widely,

with genomes occasionally being very closely related to one another, representing strain-level diversity, or evolutionary far apart, whereas abundance can differ by several orders of magnitude. Genome binning software deconvolutes metagenomic reads or assembled sequences into bins representing genomes of the community members. A popular and performant approach in genome binning uses the covariation of read coverage and short k-mer composition of contigs with the same origin across co-assemblies of one or more related samples, though the presence of strain-level diversity substantially reduces bin quality [1].

Benchmarking methods for binning and other tasks in metagenomics, such as assembly and profiling, is crucial for both users and method developers. The former need to determine the most suitable programs and parametrizations for particular applications and data sets, and the latter need to compare their novel or improved method with existing ones. When lacking evaluation software or standardized metrics, both need to individually invest considerable effort in assessing methods. CAMI is a community-driven initiative aiming to tackle this problem by establishing evaluation standards and best practices, including the design of benchmark data sets and performance metrics [1,2]. Following community requirements and suggestions, the first CAMI challenge provided metagenome data sets of microbial communities with different organismal complexities, for which participants could submit their assembly, taxonomic and genomic binning, and taxonomic profiling results. These were subsequently evaluated, using metrics selected by the community [1]. Here, we describe AMBER (Assessment of Metagenome BinnERs), an evaluation package for the comparative assessment of genome binning reconstructions from metagenome benchmark data sets. It implements all metrics decided by the community to be most relevant for assessing the quality of genome reconstructions in the first CAMI challenge and is applicable to arbitrary benchmark data sets. AMBER automatically generates binning quality assessments outputs in flat files, as summary tables, rankings, and as visualizations in images and an interactive HTML page. It complements the popular CheckM software that assesses genome bin quality on real metagenome samples based on sets of single-copy marker genes[3].

(GIGA)n SCIENCE

Methods

Input

AMBER uses as input three types of files to assess binning quality for benchmark data sets: (1) a gold standard mapping of contigs or read IDs to underlying genomes of community members; (2) one or more files with predicted bin assignments for the sequences; and (3), a FASTA or FASTQ file with sequences. Benchmark metagenome sequence samples with a gold standard mapping can, for instance, be created with the CAMISIM metagenome simulator [4,5]. A gold standard mapping can also be obtained for sequences (reads or contigs), provided that reference genomes are available, by aligning the sequences to these genomes. Popular read aligners are, for example, Bowtie [6] and BWA [7]. MetaQUAST [8] can also be used for contig alignment while it evaluates metagenome assemblies. High confidence alignments can then be used as mappings of the sequences to the genomes. The input files (1) and (2) use the Bioboxes binning format [9,10]. AMBER also accepts as bin assignments individual FASTA files for each bin, as provided by MaxBin [11]. These can be converted to the Bioboxes format. Example files are provided in the AMBER GitHub repository [12].

Metrics and accompanying visualizations

AMBER uses the gold standard mapping to calculate a range of relevant metrics [1] for one or more genome binnings of a given data set. We give below a more formal definition of all metrics than in [1], together with an explanation of their biological meaning.

Assessing the quality of bins

The purity and completeness, both ranging from 0 to 1, are commonly used measures for quantifying bin assignment quality, usually in combination [13]. We provide formal definitions below. As predicted genome bins have no label, e.g. a taxonomic one, the first step in calculating genome purity and completeness is **mapping each predicted genome bin to an underlying genome**. For this, AMBER uses one of the following choices:

(1) A predicted genome bin is mapped to the most abundant genome in that bin in number of base pairs. More precisely, let X be the set of predicted genome bins and Y the set of underlying genomes. We define a mapping of the predicted genome bin $x \in X$ as $g(x) = y$, such that genome y maps to x and the overlap between x and y, in base pairs, is maximal among all $y \in Y$, i.e.

$$g(x) = \underset{y \in Y}{\arg\max} |x \cap y| . \qquad (1)$$

(2) A predicted genome bin is mapped to the genome whose largest fraction of base pairs has been assigned to the bin. In this case, we define a mapping $g'(x) = y$ as

$$g'(x) = \underset{y \in Y}{\arg\max} \frac{|x \cap y|}{|y|}. \qquad (2)$$

If more than a genome is completely included in the bin, i.e. $|x \cap y|/|y| = 1.0$ for more than a $y \in Y$, then the largest genome is mapped.

Using either option, each predicted genome bin is mapped to a single genome, but a genome can map to multiple bins or remain unmapped. Option 1 maps to each bin the genome that best represents the bin, since the majority of the base pairs in the bin belong to that genome, whereas option 2 maps to each bin the genome that best represents that genome, since most of the genome is contained in that specific bin. AMBER uses per default option 1. In the following, we use g^* to denote one of these mappings for simplicity whenever possible.

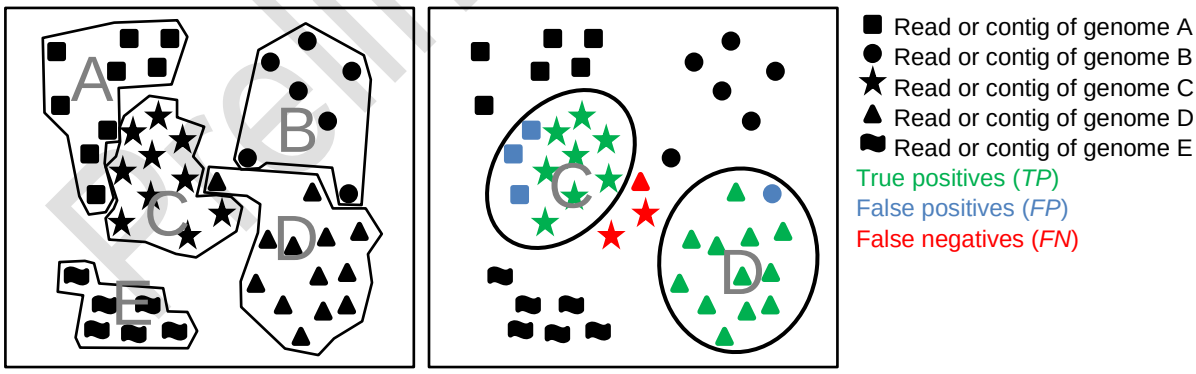

Figure 1: Schematic representation of establishing a bin-to-genome mapping for calculation of bin quality metrics. Reads and contigs of individual genomes are represented by different symbols and grouped by genome (left) or predicted genome bins (right). A bin-to-genome mapping is established using one of the criteria outlined in the text, with the upper bin mapping to genome C and the lower bin mapping to genome D. The mapping implies *TP*s, *FP*s and *FN*s for calculation of genome bin purity, completeness, contamination and overall sample assignment accuracy.

The **purity** $\boldsymbol{p}$, also known as precision, or specificity, quantifies the quality of genome bin predictions in terms of how trustworthy those assignments are. Specifically, the purity represents the ratio of base pairs originating from the mapped genome to all bin base pairs. For every predicted genome bin x,

$$p_x = \frac{TP_x}{TP_x + FP_x} \quad (3)$$

is determined, where the true positives TP_x are the number of base pairs that overlap with the mapped genome $g^*(x)$, i.e. $TP_x = |x \cap g^*(x)|$, and the false positives FP_x are the number of base pairs belonging to other genomes and incorrectly assigned to the bin. The sum $TP_x + FP_x$ corresponds to the size of bin x in base pairs. See Figure 1 for an example of predicted genome bins and respective true and false positives.

A related metric, the **contamination** $\boldsymbol{c}$**,** can be regarded as the opposite of purity and reflects the fraction of incorrect sequence data assigned to a bin (given a mapping to a certain genome). Usually, it suffices to consider either purity or contamination. It is defined for every predicted genome bin x as

$$c_x = 1 - p_x. \quad (4)$$

The **completeness** $\boldsymbol{r}$, also known as recall, or sensitivity, reflects how complete a predicted genome bin is with regard to the sequences of the mapped underlying genome. For every predicted genome bin x,

$$r_x = \frac{TP_x}{TP_x + FN_x} \quad (5)$$

is calculated, where the false negatives FN_x are the number of base pairs of the mapped genome $g^*(x)$ that were classified to another bin or left unassigned. The sum $TP_x + FN_x$ corresponds to the size of the mapped genome in base pairs.

Because multiple bins can map to the same genome, some bins might have a purity of 1.0 for a genome (if they exclusively contain its sequences), but the completeness for those bins sum up to at most 1.0 (if they include together all sequences of that genome). Genomes remaining unmapped are considered to have a completeness of zero and their purity is undefined.

As summary metrics, the **average purity** $\bar{\boldsymbol{p}}$ and **average completeness** $\bar{\boldsymbol{r}}$ of all predicted genome bins can be calculated, which are also known in computer science as the macro-

averaged precision and macro-averaged recall [14]. To these metrics, small bins contribute in the same way as large bins, differently from the sample-specific metrics discussed below. Specifically, the average purity $\bar{p}$ is the fraction of correctly assigned base pairs for all assignments to a given bin averaged over all predicted genome bins, where unmapped genomes are not considered. This value reflects how trustworthy the bin assignments are on average. Let $n_p = |X|$ be the number of predicted genome bins. Then $\bar{p}$ is calculated as

$$\bar{p} = \frac{1}{n_p} \sum_{x \in X} p_x \, . \qquad (6)$$

A related metric, the **average contamination $\bar{c}$** of a genome bin, is computed as

$$\bar{c} = 1 - \bar{p} \, . \qquad (7)$$

If very small bins are of little interest in quality evaluations, the **truncated average purity $\bar{p}_\alpha$** can be calculated, where the smallest predicted genome bins adding up to a specified percentage (the α percentile) of the data set are removed. For instance, the 99% truncated average purity can be calculated by sorting the bins according to their predicted size in base pairs and retaining all larger bins that fall into the 99% quantile, including (equally sized) bins that overlap the threshold. Let $S, S \subset X$, be the subset of predicted genome bins of X after applying the α percentile bin size threshold and $n_{p,\alpha} = |S|$. The truncated average purity $\bar{p}_\alpha$ is calculated as

$$\bar{p}_\alpha = \frac{1}{n_{p,\alpha}} \sum_{x \in S} p_x \, . \qquad (8)$$

AMBER also allows to exclude other subsets of bins, such as bins representing viruses or circular elements.

While the average purity is calculated by averaging over all predicted genome bins, the average completeness $\bar{r}$ is averaged over all genomes, including those not mapped to genome bins (for which completeness is zero). More formally, let Z be the set of unmapped genomes, i.e. $Z = \{y \in Y \mid \forall x \in X : g^*(x) \neq y\}$, and $n_r = |X| + |Z|$, i.e. the sum of the number of predicted genome bins and the number of unmapped genomes. Then $\bar{r}$ is calculated as

$$\bar{r} = \frac{1}{n_r} \sum_{x \in X} r_x \, . \qquad (9)$$

Assessing binnings of specific samples and in relation to bin sizes

Generally, it may not only be of interest how well a binning program does for individual bins, or all bins on average, irrespective of their sizes, but also how well it does overall for specific types of samples, where some genomes are more abundant than others. Binners may perform differently for abundant than for less abundant genomes, or for genomes of particular taxa, whose presences and abundances depend strongly on the sampled environment. To allow assessment of such questions, another set of related metrics exist, which either measure the binning performance for the entire sample, the binned portion of a sample, or to which bins contribute proportionally to their sizes.

To give large bins higher weight than small bins in performance determinations, the **average purity $\bar{p}_{bp}$** and **completeness $\bar{r}_{bp}$** per base pair can be calculated as

$$\bar{p}_{bp} = \frac{\sum_{x \in X} TP_x}{\sum_{x \in X} TP_x + FP_x} = \frac{\sum_{x \in X} \max_y |x \cap y|}{\sum_{x \in X} |x|} \qquad (10)$$

and

$$\bar{r}_{bp} = \frac{\sum_{y \in Y} \max_x |x \cap y|}{\sum_{y \in Y} |y|}. \qquad (11)$$

Equation (10) strictly uses the bin-to-genome mapping function g. Equation (11) computes the sum in base pairs of the intersection between each genome and the predicted genome bin that maximizes the intersection, averaged over all genomes. A genome that does not intersect with any bin results in an empty intersection. Binners achieving higher values of $\bar{p}_{bp}$ and $\bar{r}_{bp}$ than for $\bar{p}$ and $\bar{r}$ tend to do better for larger bins than for small ones, and for those with lower values it is the other way around.

The **accuracy a** measures the average assignment quality per base pair over the entire data set, including unassigned base pairs. It is calculated as

$$a = \frac{\sum_{x \in X} TP_x}{U + \sum_{x \in X} TP_x + FP_x}, \qquad (12)$$

where U is the number of base pairs that were left unassigned. Like the average purity and completeness per base pair, large bins contribute more strongly to this metric than small bins.

Genome binners generate groups or clusters of reads and contigs for a given data set. Instead

of calculating performance metrics established with a bin-to-genome mapping, another way to evaluate the quality of a clustering is to measure the similarity between the obtained and correct cluster partitions of the data set, corresponding here to the predicted genome bins and the gold standard contig or read genome assignments, respectively. This is accomplished with the Rand Index by comparing how pairs of items are clustered [15]. If two contigs or reads of the same genome are placed in the same predicted genome bin, these are here considered true positives TP. If two contigs or reads of different genomes are placed in different bins, these are considered true negatives TN. The Rand Index ranges from 0 to 1 and is the number of true pairs, $TP + TN$, divided by the total number of pairs. However, for a random clustering of the data set, the Rand Index would be larger than 0. The **Adjusted Rand Index** (ARI) corrects for this by subtracting the expected value for the Rand Index and normalizing the resulting value, such that the values still range from 0 to 1.

More formally, following [16], let m be the total number of base pairs assigned to any predicted genome bin and, $m_{x,y}$, the number of base pairs of genome y assigned to predicted genome bin x. The ARI is computed as

$$ARI = \frac{\sum_{x,y}\binom{m_{x,y}}{2} - \frac{\sum_x \binom{m_{x,.}}{2} \sum_y \binom{m_{.,y}}{2}}{\binom{m}{2}}}{\frac{1}{2}\left[\sum_x \binom{m_{x,.}}{2} + \sum_j \binom{m_{.,y}}{2}\right] - \frac{\sum_x \binom{m_{x,.}}{2} \sum_y \binom{m_{.,y}}{2}}{\binom{m}{2}}}, \quad (13)$$

where $m_{.,y} = \sum_x m_{x,y}$ and $m_{x,.} = \sum_y m_{x,y}$. That is, $m_{.,y}$ is the number of base pairs of genome y from all bin assignments and $m_{x,.}$ is the total number of base pairs in predicted genome bin x.

AMBER also provides ARI as a measure of assignment accuracy per sequence (contig or read) instead of per base pair by considering m to be the total number of sequences assigned to any bin and, $m_{x,y}$, the number of sequences of genome y assigned to bin x. The meaning of $m_{.,y}$ and $m_{x,.}$ changes accordingly.

Importantly, the ARI is mainly designed for assessing a clustering of an entire data set, but some genome binning programs exclude sequences from bin assignment, thus assigning only a subset of the sequences from a given data set. If this unassigned portion is included into the ARI calculation, the ARI becomes meaningless. AMBER, therefore, calculates the ARI only for the assigned portion of the data. For interpretation of these ARI values, the

percentage of assigned data should also be considered (provided by AMBER together in plots).

Output and visualization

AMBER combines the assessment of genome reconstructions from different binning programs or created with varying parameters for one program. The calculated metrics are provided as flat files, in several plots, and in an interactive HTML visualization. An example page is available at [17]. The plots visualize:

- (Truncated) purity $\bar{p}_\alpha$ per predicted genome bin vs. average completeness $\bar{r}$ per genome, with the standard error of the mean
- Average purity per base pair $\bar{p}_{bp}$ vs. average completeness per base pair $\bar{r}_{bp}$
- Adjusted Rand Index ARI vs. percentage of assigned data
- Purity p_x vs. completeness r_x and boxplots for all predicted bins
- Heatmaps for individual binnings representing base pair assignments to predicted bins vs. their true origins from the underlying genomes

Heatmaps are generated from binnings without requiring a mapping, where rows represent the predicted genome bins and, columns, the genomes. The last row includes all unassigned base pairs for every individual genome and, individual entries, the number of base pairs assigned to a bin from a particular genome. Hence, the sum of all entries in a row corresponds to the bin size and, the sum of all column entries, to the size of the underlying genome. To facilitate the visualization of the overall binning quality, rows and columns are sorted as follows: for each predicted bin in each row, a bin-to-genome mapping function (g, per default) determines the genome (column) that maps to the bin and the true positive base pairs for the bin. Predicted bins are then sorted by the number of true positives in descending order from top to bottom in the matrix and genomes are sorted from left to right in the same order of the bin-to-genome mappings for the predicted bins. In this way, true positives concentrate in the main diagonal starting at the upper left corner of the matrix.

AMBER also provides a summary table with the number of genomes recovered with less than a certain threshold (5% and 10% per default) of contamination and more than another threshold (50%, 70%, and 90% per default) of completeness. This is one of the main quality measures used by CheckM [3] and in e.g. [18] and [19]. In addition, a ranking of different binnings by the highest average purity, average completeness, or the sum of these two metrics is provided as a flat file.

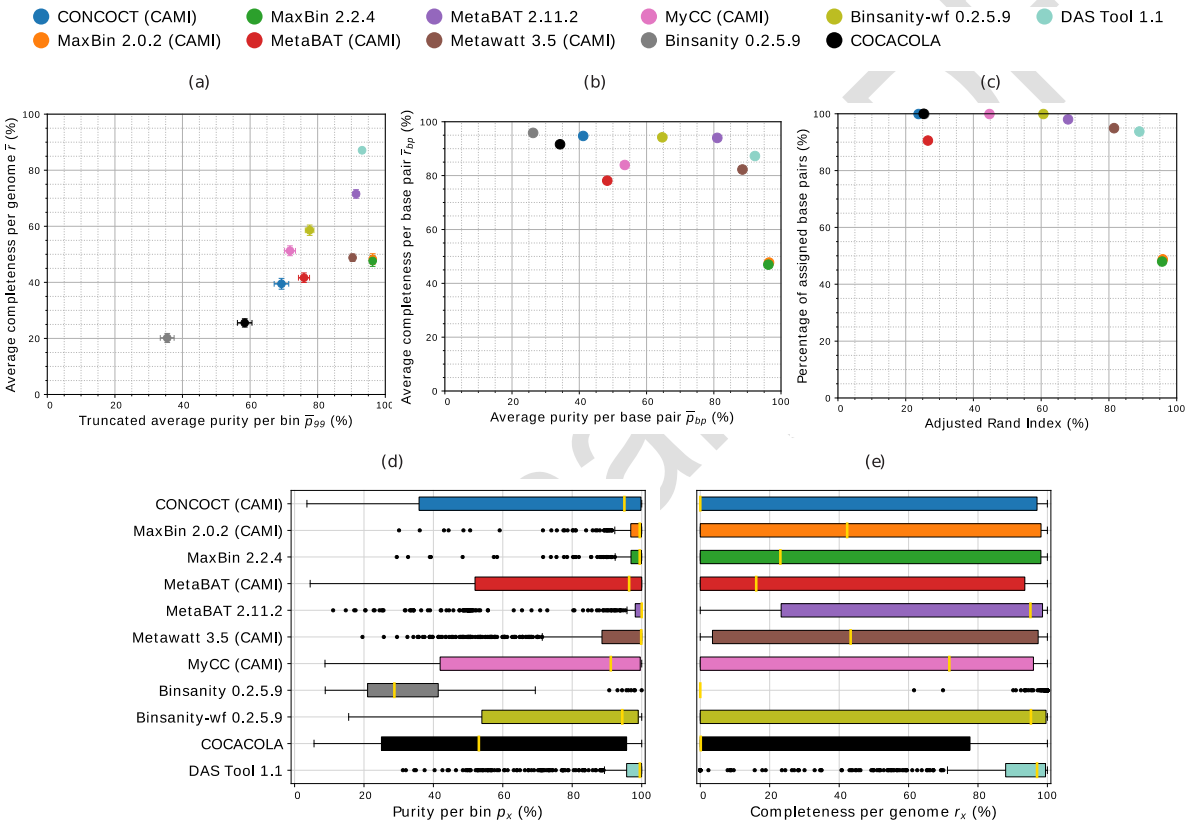

Figure 2: Assessment of genome bins reconstructed from CAMI's high complexity challenge data set by different binners. Binner versions participating in CAMI are indicated in the legend in parentheses. (a) Average purity per bin (x-axis), average completeness per genome (y-axis), and respective standard errors (bars). As in the CAMI challenge, we report $\bar{p}_{99}$ with 1% of the smallest bins predicted by each program removed. (b) Average purity per base pair (x-axis) and average completeness per base pair (y-axis). (c) Adjusted Rand Index per base pair (x-axis) and percentage of assigned base pairs (y-axis). (d-e) Boxplots of purity per bin and completeness per genome, respectively.

Results

To demonstrate an application of AMBER, we performed an evaluation of the genome binning submissions to the first CAMI challenge, together with predictions from four more programs and new program versions, on two of the three challenge data sets. These are simulated benchmark data sets representing a single sample data set from a low complexity microbial community with 40 genomes and a 5-sample time series data set of a high complexity microbial community with 596 genome members. Both data sets include bacteria, the high complexity sample also archaea, high copy circular elements (plasmids and viruses) and substantial strain-level diversity. The samples were sequenced with paired-end 150 bp Illumina reads to a size of 15 GB for each sample. The assessed binners were CONCOCT [16], MaxBin 2.0.2 [11], MetaBAT [20], Metawatt 3.5 [21], and MyCC [22]. We generated results with newer program versions of MetaBAT and MaxBin. Furthermore, we ran Binsanity, Binsanity-wf [23], COCACOLA [24], and DAS Tool 1.1 [25] on the data sets. DAS Tool combines predictions from multiple binners, aiming to produce consensus high-quality bins. We used as input for DAS Tool the predictions of all binners, except COCACOLA; for MaxBin and MetaBAT we used the results of the newer versions 2.2.4 and 2.11.2, respectively. The commands and parameters used with the programs are available in the Supplementary information.

On the low complexity data set, MaxBin 2.2.4, as its previous version 2.0.2, performed very well, as did the new MetaBAT version 2.11.2 and DAS Tool 1.1 (Figure 3, Supplementary Figure 1). Both MaxBin versions achieved the highest average purity per bin and, version 2.0.2, the highest completeness per genome on this data set. As in the evaluation of the first CAMI challenge, we report the truncated average purity, $\bar{p}_{99}$, with 1% of the smallest bins predicted by each program removed. These small bins are of little, practical interest for the analysis of individual bins and distort the average purity, since their purity is usually much lower than that of larger bins (Supplementary Table 2) and small and large bins contribute equally to this metric. On the high complexity data set, both MaxBin versions assigned less data than other programs, though with the highest purity (Figures 2, 3). MetaBAT 2.11.2 substantially improved over the previous version with all measures. Apart from DAS Tool 1.1, which created the most high quality bins from the predictions of the different binners,

MetaBAT 2.11.2 recovered the most high quality bins and showed the highest interquartile range in the purity and completeness boxplots for the high complexity data set. MetaBAT 2.11.2 and MaxBin 2.0.2 also recovered the most genomes with more than specified thresholds of completeness and contamination on the high and the low complexity data sets, respectively (Table 1, Supplementary Table 1). DAS Tool 1.1 could further improve on this measure, recovering the most genomes satisfying these conditions on both data sets. Overall, DAS Tool obtained high quality consensus bins, asserting itself as an option that can be used particularly when is not clear which binner performs best on a specific data set. As shown in [25], no single binner performs well on all ecosystems and, equivalently, there is no guarantee that the best performing binners on the analyzed data sets from the first CAMI challenge also perform best on other data sets. For more extensive information on program performances of multiple data sets, we refer the reader to [1] and future benchmarking challenges organized by CAMI [26]. Notably, some binners, such as CONCOCT, may require more than five samples for optimal performance. In general, the binning performance can also be influenced by parameter settings. These could possibly be fine-tuned to yield better results than the ones presented here. We chose to use default parameters or parameters suggested by the developers of the respective binners during the CAMI challenge (Supplementary information), reproducing a realistic scenario where such fine-tuning is difficult due to the lack of gold standard binnings. To thoroughly and fairly benchmark binners, the CAMI challenge encouraged multiple submissions of the same binner with different parameter settings. Although here we present results for binner versions released after the end of the challenge, with noticeable improvements of MetaBAT 2.11.2, the authors of MetaBAT claim that no data set specific fine-tuning was performed (direct communication). All results and evaluations are also available in the CAMI benchmarking portal[27].

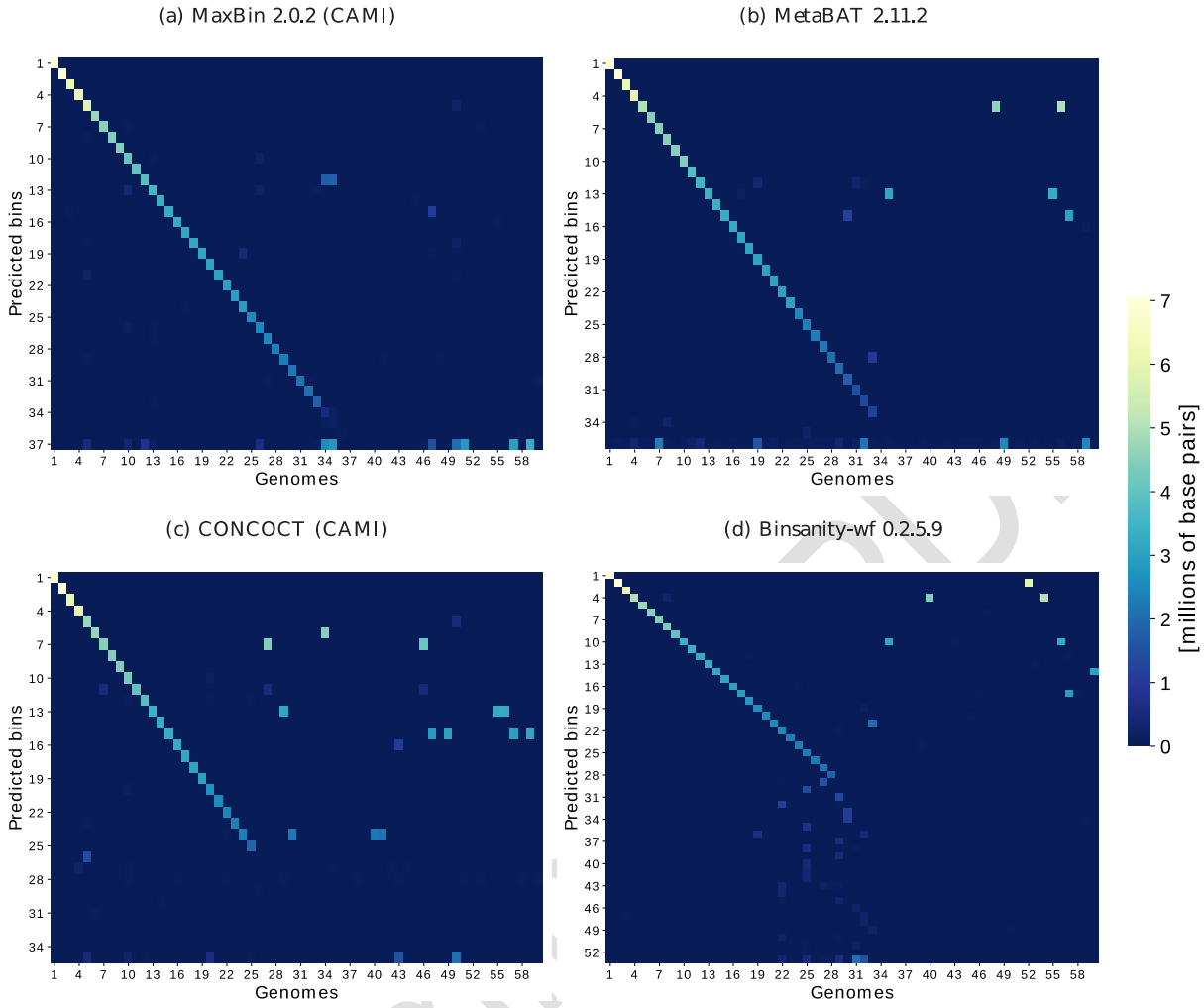

Figure 3: Heatmaps of confusion matrices for four different binning results for the low complexity data set of the first CAMI challenge representing the base pair assignments to predicted genome bins (y-axis) vs. their true origin from the underlying genomes or circular elements (x-axis). Rows and columns are sorted according to the number of true positives per predicted bin (see main text). Row scatter indicates a reduced average purity per base pair and thus underbinning (genomes assigned to one bin), whereas column scatter indicates a lower completeness per base pair and thus overbinning (many bins for one genome). The last row represents the unassigned bases per genome, allowing to assess the fraction of the sample left unassigned. These views allow a more detailed inspection of binning quality relating to the provided quality metrics (Supplementary Figure 1).

Table 1: Respective number of genomes recovered from CAMI's high complexity data set with less than 10% and 5% contamination and more than 50%, 70%, and 90% completeness.

Genome binner (% contamination)		Predicted bins (% completeness)		
		>50%	>70%	>90%
Gold standard		596	596	596
CONCOCT (CAMI)	<10%	129	129	123
	<5%	124	124	118
MaxBin 2.0.2 (CAMI)	<10%	277	274	244
	<5%	254	252	224
MaxBin 2.2.4	<10%	274	271	236
	<5%	249	247	216
MetaBAT (CAMI)	<10%	173	152	126
	<5%	159	140	118
MetaBAT 2.11.2	<10%	**427**	**417**	**361**
	<5%	**414**	**404**	**353**
Metawatt 3.5 (CAMI)	<10%	408	387	338
	<5%	396	376	330
MyCC (CAMI)	<10%	189	182	145
	<5%	166	159	127
Binsanity 0.2.5.9	<10%	9	9	9
	<5%	6	6	6
Binsanity-refine 0.2.5.9	<10%	206	204	192
	<5%	183	181	171
COCACOLA	<10%	88	87	75
	<5%	69	69	60
DAS Tool 1.1	<10%	**465**	**462**	**405**
	<5%	**428**	**425**	**376**

Conclusions

AMBER provides commonly used metrics for assessing the quality of metagenome binnings on benchmark data sets in several convenient output formats, allowing in-depth comparisons of binning results of different programs, software versions, or with varying parameter settings. As such, AMBER facilitates the assessment of genome binning programs on benchmark metagenome data sets, for bioinformaticians aiming to optimize data processing pipelines and method developers. The software is available as a standalone program [12], as a Docker image (automatically built with the provided Dockerfile), and in the CAMI

benchmarking portal [27]. We will continue to extend the metrics and visualizations according to community requirements and suggestions.

Availability of supporting source code and requirements

Project name: AMBER: Assessment of Metagenome BinnERs
Project home page: https://github.com/CAMI-challenge/AMBER
Research Resource Identifier: SCR_016151
Operating system(s): Platform independent
Programming language: Python 3.5
License: Apache 2.0

Availability of supporting data

An archive of the CAMI benchmark data sets [2] and snapshots of the code [28] are available in the GigaScience GigaDB repository.

Additional files

SupplementaryInformation.pdf

Acknowledgements

The authors thank Christopher Quince for contributing Python code, all genome binning software developers participating in the CAMI challenge for their feedback on most relevant metrics, all developers who helped us to run their binning software, and the Isaac Newton Institute in Cambridge for its hospitality under the program MTG.

Competing interests

The authors declare that they have no competing interests.

(GIGA)n SCIENCE

Funding

This work has been supported by Helmholtz society and the Cluster of Excellence in Plant Sciences (CEPLAS) funded by the German Research Foundation (DFG).

References

1. Sczyrba, A., Hofmann, P., Belmann, P. et al. Critical Assessment of Metagenome Interpretation – a benchmark of metagenomics software. Nature Methods, 14, 11 (2017), 1063-1071.
2. Belmann, P., Bremges, A., Dahms, E. et al. Benchmark data sets, software results and reference data for the first CAMI challenge. GigaScience Database (2017). http://dx.doi.org/10.5524/100344.
3. Parks, H. D., Imelfort, M., Skennerton, T., C., Hugenholtz, P., and Tyson, W. G. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Research, 25, 7 (2015), 1043-1055.
4. CAMISIM metagenome simulator. https://github.com/CAMI-challenge/CAMISIM. Accessed 27 Apr 2018.
5. Fritz, A., Hofmann, P., Majda, S. et al. CAMISIM: Simulating metagenomes and microbial communities. bioRxiv, 300970 (2018).
6. Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biology, 10, 3 (2009), R25-10.
7. Li, H. and Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics, 25, 14 (2009), 1754-1760.
8. Mikheenko, A., Saveliev, V., and Gurevich, A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics, 32, 7 (2016), 1088-1090.
9. Belmann, P., Dröge, J., Bremges, A., McHardy, A. C., Sczyrba, A., and Barton, M. D. Bioboxes: standardised containers for interchangeable bioinformatics software.

Gigascience, 4, 47 (2015).

10. Bioboxes binning format. https://github.com/bioboxes/rfc/tree/master/data-format. Accessed 27 Apr 2018.
11. Wu, Y. W., Simmons, B. A., and Singer, S. W. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics, 32 (2016), 605-607.
12. AMBER GitHub repository. https://github.com/CAMI-challenge/AMBER. Accessed 27 Apr 2018.
13. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 16 (2000), 412–424.
14. Tsoumakas, G., Katakis, I., and Vlahavas, I. Mining Multi-label Data. In Maimon, O. and Rokach, L., eds., Data Mining and Knowledge Discovery Handbook. Springer-Verlag, 2010.
15. Rand, W. M. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66, 336 (1971), 846-850.
16. Alneberg, J., Bjarnason, B. S., de Bruijn, I. et al. Binning metagenomic contigs by coverage and composition. Nature Methods, 11 (2014), 1144–1146.
17. AMBER example HTML visualization of calculated metrics. https://cami-challenge.github.io/AMBER/. Accessed 30 May 2018.
18. National Center for Biotechnology Information (NCBI)[Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information (1988).
19. Parks, D. H., Rinke, C., Chuvochina, M. et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nature Microbiology, 2 (2017), 1533–1542.
20. Kang, D. D., Froula, J., Egan, R., and Wang, Z. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ, 3 (2015), e1165.

21. Strous, M., Kraft, B., Bisdorf, R., and Tegetmeyer, H. E. The binning of metagenomic contigs for microbial physiology of mixed cultures. Frontiers in Microbiology, 3, 410 (2012).

22. Lin, H. H. and Liao, Y. C. Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. Scientific Reports, 6, 24175 (2016).

23. Graham, E. D., Heidelberg, J. F., and Tully, B. J. BinSanity: unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation. PeerJ, 5 (2016), e3035.

24. Lu, Y. Y., Chen, T., Fuhrman, J. A., and Sun, F. COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. Bioinformatics, 33, 6 (2017), 791-798.

25. Sieber, C. M. K., Probst, J., Sharrar, A., Thomas, B. C., Hess, M., Tringe, S. G., and Banfield, J. F. Recovery of genomes from metagenomes via a dereplication, aggregation, and scoring strategy. bioRxiv, 107789 (2017).

26. Critical Assessment of Metagenome Interpretation (CAMI). http://www.cami-challenge.org. Accessed 27 Apr 2018.

27. CAMI benchmarking portal. https://data.cami-challenge.org. Accessed 27 Apr 2018.

28. Meyer, F., Hofmann, P., Belmann, P., Garrido-Oter, R., Fritz, A., Sczyrba, A., and McHardy, A., C. Supporting data for "AMBER: Assessment of Metagenome BinnERs". GigaScience Database (2018). http://dx.doi.org/10.5524/100454.