Genome analysis

MAGScoT: a fast, lightweight and accurate bin-refinement tool

Malte Christoph Rühlemann 1,2,*, Eike Matthias Wacker 1, David Ellinghaus 1 and Andre Franke 1

1Institute of Clinical Molecular Biology, Kiel University, 24105 Kiel, Germany and 2Institute for Medical Microbiology and Hospital Epidemiology, Hannover Medical School, 30625 Hannover, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: Recovery of metagenome-assembled genomes (MAGs) from shotgun metagenomic data is an important task for the comprehensive analysis of microbial communities from variable sources. Single binning tools differ in their ability to leverage specific aspects in MAG reconstruction, the use of ensemble binning refinement tools is often time consuming and computational demand increases with community complexity. We introduce MAGScoT, a fast, lightweight and accurate implementation for the reconstruction of highest-quality MAGs from the output of multiple genome-binning tools.

Results: MAGScoT outperforms popular bin-refinement solutions in terms of quality and quantity of MAGs as well as computation time and resource consumption.

Availability and implementation: MAGScoT is available via GitHub (https://github.com/ikmb/MAGScoT) and as an easy-to-use Docker container (https://hub.docker.com/repository/docker/ikmb/magscot).

Contact: m.ruehlemann@ikmb.uni-kiel.de

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Reconstructing high-quality microbial genome sequences from complex metagenomic sequencing libraries is crucial for studying microbial traits and their influence on host health or disease, or their role in adaptation to changing environmental factors. A typical workflow for this task usually encompasses the assembly of larger contiguous sequences from short-read sequencing libraries, followed by binning of these contigs into larger sets called metagenome-assembled genomes (MAGs), and has been the basis for large-scale metagenomic surveys (Almeida et al., 2019; Pasolli et al., 2019).

Binning of contigs into MAGs can be achieved by one or more specific properties of the assembled contigs, i.e. similarity in coverage, k-mer or tetranucleotide frequencies, GC content or the presence of specific marker genes, paired with different types of statistical and/or machine learning approaches for clustering contigs into bins, such as expectation maximization or variational autoencoders (Alneberg et al., 2014; Kang et al., 2019; Nissen et al., 2021; Wu et al., 2016). The resulting MAGs are subsequently assessed for their quality, e.g. by the CheckM software (Parks et al., 2015), which uses clade-specific microbial marker genes to assess the completeness of a MAG, as well as contamination due to incorrect assembly or binning, and the presence of marker gene duplicates that may indicate the binning of multiple closely related genomes.

It was demonstrated that applying and combining multiple binning approaches can be useful to reconstruct more and better quality MAGs from metagenomic datasets (Sieber et al., 2018). Two popular algorithms for such a bin-refinement are DASTool and the bin-refinement module of the metaWRAP pipeline (Sieber et al., 2018; Uritskiy et al., 2018). Briefly, DASTool relies on an iterative dereplication, aggregation and scoring algorithm that selects the highest-quality genome from a collection of contig-to-bin mappings based on 51 bacterial- and 38 archaeal-specific single-copy marker genes. The metaWRAP refinement module, henceforth simply called metaWRAP, relies on the identification of overlaps between multiple bin sets, from which it constructs potentially higher quality hybrid bins. The CheckM software is then used to select the highest quality version of a genome from original and hybrid bins. MetaWRAP can use the output of up to three individual binning softwares at once; if more input sets are to be included, they must be distributed across multiple metaWRAP runs.

We introduce here the tool MAGScoT (=MAG Scoring Tool) which combines these two concepts into a single approach. MAGScoT relies on two sets of microbial single-copy marker genes from the Genome Taxonomy Database Toolkit (Chauvel et al., 2020), 120 bacterial and 53 archaeal, stored as HMM-profiles for fast annotation of amino acid sequences predicted from the assembled contigs (Hyatt et al., 2010; Johnson et al., 2010; Mistry et al., 2010).
et al., 2013). Presence profiles of marker genes are compared between the results of the individual binning algorithms, and new hybrid candidate bins are created if MAGs from binsets share a user-adjustable proportion of marker genes (default: 80%). Subsequently, all original bins from each binning software, and the newly constructed hybrid bins are scored in the same way as introduced by DASTool, with the option to weight completeness (default: a = 1) and penalize contamination (default: b = 0.5 and c = 0.5). MAGScoT is not limited in the number of binning algorithms outputs that can be used as input. Details on the MAGScoT algorithm can be found in Supplementary Methods section.

We evaluated the binning performance and computational demand of MAGScoT, DASTool and metaWRAP on two datasets. The first dataset was the simulated ‘marine’ dataset of the CAMI2 challenge (see Section 2) (Meyer et al., 2022). This complex dataset with known ground truth offered a comparison to a perfect binning output. The second dataset consisted of 50 gut metagenome samples known from the integrative Human Microbiome Project (HMP/HMP2, see Section 2 for selection criteria), which allowed us to compare performance in a real-world setting (Lloyd-Price et al., 2019). All three bin-refinement softwares were run on the same input data derived from four binning algorithms: MaxBIn2, MetaBAT2, CONCOCT and VAMB (Supplementary Tables S1 and S2). All original and refined bins were evaluated using CheckM.

Computational time and hardware demand were assessed on a single compute node in a high-performance computing environment in individual jobs restricted to 8 CPU cores and 80 GB of RAM (Table 1, Fig. 1, Supplementary Tables S3 and S4). The computational time was longest for metaWRAP in both datasets due to the iterative usage of the CheckM software to score the individual bins. Of note, RAM usage was also highest for metaWRAP, as CheckM relies on large reference trees for scoring based on clade-specific marker genes. MAGScoT showed the fastest performance of all tools in both datasets. Especially in the highly complex and fragmented synthetic ‘marine’ dataset, HMM-based annotation and the main MAGScoT algorithm were approx. 15-fold faster than DASTool for the equivalent steps, with both programs using the same predicted amino acid genes scales linearly with metagenome complexity; however, additional compute resources can speed-up these steps almost linearly.

For all three bin refinement algorithms, the completeness and contamination statistics showed very good values, with the respective median values clearly surpassing the thresholds for HQ MAGs and showing performance close to the gold standard for the simulated marine dataset (GSA2k, Table 1, right panels). The statistics clearly show the importance of refining the bins using the results of multiple binning algorithms to leverage their respective strengths, i.e. MaxBIn2 showed the highest completeness scores of all individual binners in the real metagenomic data, but at the expense of the highest contamination. Conversely, VAMB reconstructs bins with very low contamination at the cost of the lowest overall completeness. All CheckM scores for all original and refined bins can be found in Supplementary Tables S5 and S6.

In both the synthetic marine and the real-world gut metagenome datasets, MAGScoT produced the highest number of high quality MAGs (HQ, completeness > 90%, contamination < 10%, Table 1, Fig. 1, middle panels). The number of medium quality (MQ) MAGs (completeness > 50%, contamination < 10%) was highest for metaWRAP. It is likely that MAGScoT identifies fewer MQ MAGs since its default scoring cutoff is 0.5, which is calculated based on bin completeness and contamination (see Section 2), thus bins marginally above a completeness of 50%, even with low contamination up to 10%, are not reported as they fall below this threshold. Comparing the MAGScoT output to the gold standard for the simulated marine community, MAGScoT identified 425 of 464 (92.6%) possible HQ MAGs and 614 of 697 (88.1%) of MQ MAGs (Table 1). Disabling the bin merging steps in MAGScoT resulted in the lowest computation time of all bin-refinement software, accompanied by a slight decrease in the number of HQ MAGs but an increase in MQ MAGs (Table 1).

In summary, MAGScoT outperforms the two popular bin-refinement solutions metaWRAP and DASTool in terms of the quantity and quality of MAGs recovered, while being more economical with computational resources.

Table 1. Overview of bin-refinement performances in the marine and HMP2 gut metagenome datasets

|                      | Marine dataset from CAMI2 (n = 10) | HMP2 gut dataset (n = 50) |
|----------------------|-------------------------------------|---------------------------|
|                      | Gold standard (>2k bp)              | Gold standard (>2k bp)    |
| MAGScoT (no merge)   | MAGScoT (no merge)                 | MAGScoT (no merge)        |
| DASTool              | metaWRAP                            | DASTool                   |
|                      |                                     | metaWRAP                  |
| Total number of      |                                     |                           |
| Contigs              | 439 032                             | 256 456                   |
| Proteins             | 3 284 220                           | 2 038 229                 |
| Input bins           | 2726                                | 3385                      |
| Binned Mbp (%)       | 2542.0 (78.1%)                      | 2437.7 (74.9%)            |
|                      | 2374.0 (73.0%)                      | 2065.5 (63.5%)            |
|                      | 2155.0 (66.2%)                      | 2149.7 (100%)             |
|                      | 1658.2 (77.1%)                      | 1562.6 (72.7%)            |
|                      | 1342.6 (62.5%)                      |                           |
| Runtime              |                                     |                           |
| Total (min)          | 135                                 | 101                       |
| Preprocessing         | 79.1                                | 82                        |
| Annotate and score   | 55.9                                | 140                       |
| Total (per sample)   | 13.5                                | 56.5                      |
| CheckM               |                                     |                           |
| Comp.                |                                     |                           |
| Cont.                |                                     |                           |
| >90%                 | 464                                 | 251                       |
| >70%                 | 601                                 | 327                       |
| >50%                 | 697                                 | 368                       |
| >30%                 | 697                                 | 420                       |
| >20%                 | 697                                 | 423                       |
| >10%                 | 697                                 | 333                       |
| > 5%                 | 697                                 | 458                       |
|                      |                                     |                           |
| Note: Performance was evaluated by runtime on a single compute node in a high-performance computation environment with 8 CPUs and 80Gb of RAM and the quality and quantity of recovered MAGs based on CheckM scores for clade-specific completeness (Comp.) and contamination (Cont.). Values in bold font indicate the best observed value of the respective table row. |
| a) Total length of all binned contigs > 2 kbp. |
| b) Prodigal was used for identification of open reading frames as input for MAGScoT and DASTool, hence the equal times for preprocessing. |
2 Materials and methods

All scripts used for processing the data are available via GitHub: https://github.com/mruehlemann/MAGScoT_benchmarking_scripts

2.1 Dataset acquisition and processing

Short reads and gold standard assembly contig data of the simulated marine dataset were downloaded from the CAMI resource server (https://frl.publisso.de/data/frl:6425521/marine/short_read/). Gut metagenome short read data for the HMP2 gut dataset were acquired from the servers of the Inflammatory Bowel Disease Multi‘omics Database (https://ibdmdb.org/tunnel/public/HMP2/WGS/1818/rawfiles/). Here, data of the first 100 samples were downloaded and the 50 samples with the most reads were kept to ensure high enough read depth for the subsequent steps. Short read data was assembled using the Megahit software to acquire metagenomic contigs. From here, both datasets were treated equally. Contig data was filtered to retain contigs with a size of ≥2000 bp, as this was used as minimum contig size threshold for all binning softwares. For each sample, reads were mapped against the ‘own’ contigs using minimap2 (version 2.17-r941). In addition, for both datasets, a cross-sample contig catalog was created and also the reads of the respective datasets were mapped against this catalog using minimap2. The resulting mapping files (SAM format) were sorted and converted to binary (BAM) format using Samtools (version 1.9). The gqi_summarize_bam_contig_depths executable from the MetaBAT2 binning algorithm was used with default options to receive coverage information for the individual sample mappings and the mappings to the dataset catalogs. Per-sample contig sequences and mapping results were used for the MetaBAT2, MaxBin2 and CONCOCT binning softwares with default options. The cross-mappings to the dataset catalogs were used in VAMB with default options and using CPU-accelerated calculations. The binning outputs were used in the bin-refinement algorithms.

As ground truth data was available for the simulated marine dataset, best-case binning results were derived from these profiles. To have a fair comparison of binning tools and refiners to the ground truth with a focus on bacterial and archaeal genomes, the original metagenomic contigs were filtered to include only data from bacterial/archaeal genomes which were represented to at least 40% in contigs of ≥2000 bp, as this was the length cutoff used for the binning algorithms. In addition, only contigs ≥2000 bp were included in the ground truth bin quality assessment using CheckM. The resulting filtered gold standard assembly (gsa2k) together with the ground-truth contig-to-bin information was scored using MAGScoT and CheckM to generate the binning/refinement gold standard for comparison.

2.2 Binning algorithms

This section gives a brief summary of the four used as baseline for the bin refinement:

- MetaBAT2 (v2.12.1): MetaBAT2 relies on normalized tetranucleotide frequencies (TNFs) calculated from the input contigs to create an initial graph for graph-based clustering. Subsequently, similarity scores from TNFs and correlation between contig abundances are used iteratively to be incorporated into the graph structure. A label propagation algorithm is used to score connections and identify clusters of contigs.
- MaxBin2 (v2.2.7): In addition to TNFs and contig abundances MaxBin2 uses a set universal single-copy marker gene for the identification and clustering of metagenomic bins. The clustering is based on an Expectation-Maximization algorithm.
- CONCOCT (v.1.1.0): CONCOCT more generally uses k-mer frequencies and contig abundances for clustering, by default, the k-mer length in CONCOCT is 4, which makes it congruent to the aforementioned TNFs. One major difference, is CONCOCT suggests to cut contigs into pieces of 10kb, to identify and disentangle mis-assemblies. The clustering of the contigs into bins is done by a Gaussian mixture model fit with a variational Bayesian approximation.
- VAMB (v.3.0.3): VAMB is the most recent of the herein used binning algorithms. Similar to the other binning tools, VAMB relies on contig coverage and TNFs. The major difference to the other binning tools is the use of unsupervised deep learning via variational autoencoders and can be accelerated by the use of GPUs.

2.3 MAGScoT algorithm

The MAGScoT algorithm is implemented in R (v.4.0) and by default takes two input files: (i) the HMMER-based GTDDBk bacterial and archaeal single-copy marker gene (SCG) annotations (release 207) of amino acid sequences of open reading frames from the assembled contigs identified by prodigal (option: -p meta) and (ii) the contig-to-bin mappings of the individual binning algorithms.

In a first step, MAGScoT pairs annotated marker genes with their respective contigs. Subsequently, bins with a predefined minimum number of identified marker genes (default: 20) are scanned for overlap with all other bins based on shared genes which were annotated as marker genes. If this overlap surpasses a threshold (default: 90%), a hybrid bin of the two original bins is added to the total bin collection. This process can be performed iteratively in which also the newly formed hybrid bins are scanned for overlap with additional original bins. By default, this is done twice, as each iteration can add hundreds of new bins for the scoring and refinement process. The creation of hybrid bins can be skipped by the --skip_merge_bins flag when starting MAGScoT.

The second step is the scoring and refinement iteration. Here, each bin is scored using sets of SCGs, 53 for archaea and 120 for bacteria. The scoring function is the same as in DASTool:

\[
\text{score} = a \times \text{completeness} - b \times \text{duplicateSCGs} - c \times \text{multiplicateSCG}
\]

with

\[
\text{completeness} = \frac{\text{uniqueSCGs}}{\text{SCGs} \in \text{set}}
\]
duplicateSCGs = \frac{SCG\text{occurring} > 1}{uniqueSCGs}

\text{multiplySCGs} = \frac{\text{totalSCGs} - \text{uniqueSCGs} - \text{duplicateSCGs}}{\text{SCGs} \in \text{set}}

The scoring weights are \(a = 1, b = 0.5, c = 0.5\) by default, but can be adjusted by the user, e.g., to put more weight on completeness. The higher score from the archaeal and bacterial sets of SCGs is used as the bins score. Each round, the bin with the highest score is picked from the dataset. Depending on the number of candidate bins in the dataset, the iterative (re-)scoring of bins can be a time-consuming step. A major reason for the speed of MAGScoT is that it does not re-score all bins in the dataset in each iteration, but only those bins affected by the last selected candidate bin, i.e., sharing contigs with the selected candidate, which only represents a very small fraction of all bins. In addition, MAGScoT uses the information from which input dataset the previous winning bin originated, thus when two or more consecutive winning bins come from the same input set, the re-scoring step is also skipped. The re-scoring is never skipped in case a hybrid bin (see above) was selected, as this can affect multiple input sets at once. In addition, after each scoring, the reporting threshold \(t\) (default: 0.5) is used to remove bins from the dataset based on if their completeness does not surpass \(t\), as bins below this completeness will never reach the reporting threshold. The scoring and refinement process is repeated until there is no more bin surpassing \(t\).

### 2.4 Refinement benchmark

The same input generated by the individual binners was used as input for MAGScoT, DASTool and metaWRAP. DASTool was run with default options, however using the \(-p\) option to use the same amino acid sequences as used in MAGScoT/MAGScoT and to have a better estimate of the actual time DASTool spends in the iteration and scoring phase. As the bin-refinement module of metaWRAP accepts a maximum of three bin sets simultaneously as input, it was first run with output from MaxBin2 and VAMB. In a second run, the resulting refined bins were used as input together with the output from CONCOCT and MetaBAT2. This strategy is recommended by the metaWRAP developers (see https://github.com/bxlab/metaWRAP/blob/master/Usage_tutorial.md#step-5-consolidate-bin-sets-with-the-bin-refinement-module). After refinement, all original and refined bins were scored using CheckM. The CheckM scores for all bins can be found in Supplementary Tables S5 and S6 for the simulated marine and the real HMP2 gut metagenomes, respectively. Running times for all samples and all steps in the pre-processing and main bin-refinements software can be found in Supplementary Tables S3 and S4.

### Data availability

All data were required from publicly accessible repositories (marine synthetic community data: https://frl.publisso.de/data/ibdbmdb.org/tunnel/public/HMP2/WGS/1818/rawfiles). All filtered assemblies (contigs > 2 kb), prodigal-inferred amino acid sequences, HMM annotations and binning results (pre-refinement and refined for all included tools) are accessible via: https://zenodo.org/record/7084980.

### Funding

The project was funded by the German Research Foundation (DFG) Collaborative Research Center 1182: Origin and Function of Metaorganisms [SFB 1182; Project-ID 261376515, Projects A2 and INF] and DFG Research Unit 3042: miTarget—miTarget—The Microbiome as a Therapeutic Target in Inflammatory Bowel Diseases [RU 3042; Project-ID 426660215, Project: P1 (BA 68522-1) and INF (EL 8315-1)].

### Conflict of Interest: none declared.

### References

Almeida,A. et al. (2019) A new genomic blueprint of the human gut microbiota. Nature, 568, 499–504.

Alneberg,J. et al. (2014) Binning metagenomic contigs by coverage and composition. Nat. Methods, 11, 1144–1146.

Chaumeil,P.-A. et al. (2020) GTDB-Tk: a toolkit to classify genomes with the genome taxonomy database. Bioinformatics, 36, 1925–1927.

Hyatt,D. et al. (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics, 11, 119.

Johnson,L.S. et al. (2010) Hidden Markov model speed heuristic and iterative HMM search procedure. BMC Bioinformatics, 11, 431.

Kang,D.D. et al. (2019) MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ, 7, e7359.

Lloyd-Price,J. et al. (2019) Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. Nature, 569, 653–662.

Meyer,F. et al. (2022) Critical assessment of metagenome interpretation: the second round of challenges. Nat. Methods, 19, 429–440.

Mistry,J. et al. (2013) Challenges in homology search: HHMmer3 and convergent evolution of coiled-coil regions. Nucleic Acids Res., 41, e121.

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

Parks,D.H. et al. (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res., 25, 1043–1055.

Pascioli,E. et al. (2019) Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle. Cell, 176, 649.

Sieber,C.M.K. et al. (2018) Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. Nat. Microbiol., 3, 836–843.

Uritskiy,G.V. et al. (2018) MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. Microbiome, 6, 158.

Wu,Y.-W. et al. (2016) MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinform. Oxf. Engl., 32, 605–607.