Chapter 6

How to Obtain and Compare Metagenome-Assembled Genomes

Fabio Beltrame Sanchez, Suzana Eiko Sato Guima, and João Carlos Setubal

Abstract

Metagenome-assembled genomes, or MAGs, are genomes retrieved from metagenome datasets. In the vast majority of cases, MAGs are genomes from prokaryotic species that have not been isolated or cultivated in the lab. They, therefore, provide us with information on these species that are impossible to obtain otherwise, at least until new cultivation methods are devised. Thanks to improvements and cost reductions of DNA sequencing technologies and growing interest in microbial ecology, the rise in number of MAGs in genome repositories has been exponential. This chapter covers the basics of MAG retrieval and processing and provides a practical step-by-step guide using a real dataset and state-of-the-art tools for MAG analysis and comparison.

Keywords Metagenomics, Microbiome, Genomics, Bacteria

1 Introduction

Given an environment, such as the human skin or water in a pond, its microbiota is the collection of all microorganisms inhabiting it. We define the microbiome of that environment as the collection of genomes, genes, and gene products directly related to that microbiota. This is considered a genome-driven definition. For a broader definition, we refer the reader to the paper by Berg et al. [1]. We also refer the reader to a review article on the term "microbiome" [2].

Environments can be sampled, and the total DNA from a sample can be extracted, and this DNA can be sequenced (this is also known as the shotgun approach for microbiome sequencing). The set of DNA reads obtained from the sample in this way is called a metagenome dataset.

João Carlos Setubal et al. (eds.), *Comparative Genomics: Methods and Protocols*, Methods in Molecular Biology, vol. 2802, https://doi.org/10.1007/978-1-0716-3838-5_6,

Metagenome-assembled genomes, or MAGs, are genomes reconstructed from a metagenome dataset. For the last 10 years or so, MAGs have become an essential tool for the understanding of microbiomes. It has become almost routine for single papers to describe dozens, hundreds, or even thousands of MAGs from specific environments [3–5].

This chapter describes how MAGs can be obtained, analyzed, and compared. Even though metagenome datasets generally contain DNA from a variety of microorganisms (viruses, bacteria, archaea, protozoans), for simplicity of exposition the focus in this chapter is on bacterial genomes.

2 How MAGs Can Be Obtained

The steps for the reconstruction of MAGs are outlined in Fig. 1. Here, we give brief descriptions of each of those steps.

As defined above, a metagenome dataset is a collection of DNA reads that are the result of sequencing.

Preprocessing The reads must be checked for quality and the presence of artifacts (such as adapter sequences).

Assembly The next step is to assemble the reads. Two popular assembly tools for metagenome datasets are MEGAHIT [6] and metaSPAdes [7]. The result of this step is a collection of *contigs*. A contig is a DNA sequence that presumably corresponds to a contiguous section of a genome. If the contig was assembled from several reads, its length should be longer than any of the constituent reads. In the best possible case, a contig may correspond to an entire genome, assuming the genome to be just one chromosome, as is the case for many bacteria. But this scenario is unlikely for metagenome datasets. The most frequent outcome, especially for complex environments (environments that contain thousands of species), is a collection of thousands or hundreds of thousands of contigs. A comparison of different assembly tools for metagenome datasets was done by Vollmers et al. [8].

Binning This is the step in which we separate contigs into bins, with each bin representing one individual genome. The general idea is to determine the intrinsic properties of contig sequences, and then place in the same bin those contigs which have the same or almost identical properties. Many properties can be used to group contigs into bins, but the most common is the *k*-mer profile [9].

The binning process is vulnerable to false negative and false positive errors. In order to explain false negatives, it is important to mention the biological phenomenon of horizontal gene transfer, or

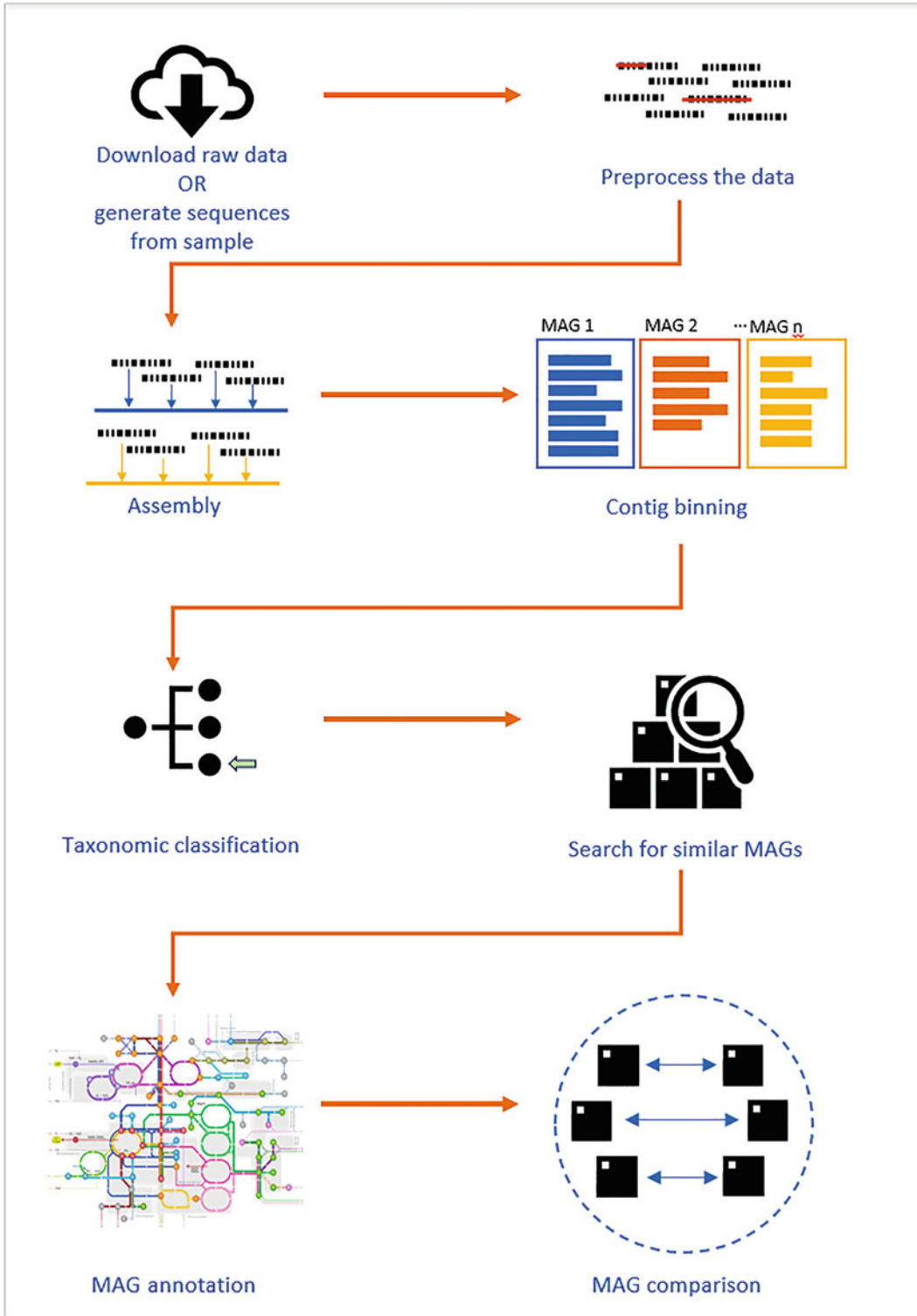

Fig. 1 Summary view of MAG generation and analysis steps

HGT [10]. In the context of this chapter, it is useful to distinguish between ancient and recent HGT events. In an ancient HGT event, foreign DNA that entered a genome had time to evolve and become more like the host genome. This means, for example, that the *k*-mer profile of the foreign DNA will be similar to the *k*-mer profile of the host genome, and therefore should not be a problem for binning. On the other hand, a recent HGT event means that not enough evolutionary time has elapsed for the foreign DNA to adapt to the host genome, in the case that the foreign DNA is "very different" from that of the host (e.g., when the host is a bacterium and the foreign DNA comes from a virus). In this case, the *k*-mer profile of the foreign DNA will likely be quite different from that of the host and will likely result in the binning program not placing the corresponding contig in the correct bin, and hence a false negative error will be created.

A false positive error occurs when a contig that does not belong to a bin is incorrectly placed in that bin. This may happen, for example, when the contig in question has properties (such as its *k*-mer profile) that are similar to the properties of other contigs that are already in the bin. A *k*-mer profile is not an absolute property that can distinguish any two organisms; two different organisms may share similar *k*-mer profiles purely by chance. This is rare, but it is not impossible.

An important question regarding the use of *k*-mer profiles to bin contigs is the minimum length required of a contig so that its *k*-mer profile will be sufficiently similar to the *k*-mer profile of its genome. Experiments show [11] that the accuracy of a *k*-mer profile for binning decreases with decreasing contig length. A rough guideline is that contigs that are shorter than 2500 bp [12] are, in general, not long enough to be properly binned using *k*-mer profiles. Another important consideration is the value of *k*. A small value (such as 4 or 5) helps minimize the number of zero counts, since nearly all tetramers or 5-mers will be found on any given genome (although at differing frequencies, of course). On the other hand, the probability of two unrelated genomes having similar *k*-mer profiles is higher for smaller values of *k*. A recent study recommends values in the range 17–19 for prokaryote genome comparison [13].

Once we have distributed the contigs into bins, we can say that each bin is a candidate MAG. In order to declare that a bin is in fact a MAG, we still have to apply quality criteria, as described in Subheading 3.

Pipelines for Obtaining MAGs

All the steps described above can be accomplished by a single pipeline, which can automatically run specific programs for each step. One such pipeline is MetaWRAP [14]. MetaWRAP is a pipeline containing several modules for reconstructing and analyzing

MAGs such as preprocessing, assembly, binning, bin refinement, taxonomic classification, and quantifying MAGs in samples. MetaWRAP is flexible, allowing the user to optionally execute some steps outside the pipeline. In addition to the basic process for reconstructing MAGs (preprocessing, assembly, and binning), MetaWRAP contains the bin refinement module, a step which combines the results of different binning tools, improving on the result of using only a single binning tool.

3 Quality Checking

3.1 Completeness and Contamination

The steps described in the previous section are all vulnerable to errors. Therefore, an essential additional step in determining MAGs is the evaluation of the genome quality of each bin. The quality criteria most commonly used are *completeness* and *contamination*.

In evaluating completeness, we seek to determine how close the set of contigs present in a bin is to the complete genome they are assumed to belong to. This means that completeness is measured as a percentage; the closer to 100%, the more complete is the MAG.

How can this be done, if we do not know what the real genome is? In the case of bacteria, the basic idea is to use knowledge about bacterial genes that are known to be present in all bacterial genomes (the *core* bacterial gene set). With the additional resource of having well-defined ortholog families for each of those genes (e.g., in terms of hidden Markov models, [15]), we can readily check for the presence of these genes in any given candidate MAG. For example, the software CheckM [16] defines the bacterial core gene set as containing 104 "marker" genes. So, for example, if we find 95 of these genes in a MAG known to be from a bacterium, we could state that this MAG is $95/104 = 91.3\%$ complete. This of course is an estimate; the real completeness value will in general be lower or greater than this value. (In reality, CheckM's estimates are more sophisticated than the simple description just given.)

In evaluating contamination, the core bacterial gene set is also used. If that set is assumed to contain only single-copy genes, then the presence of duplicates (or any number of occurrences greater than one) of these genes is evidence of contamination. Just like in the case of completeness, contamination is expressed as a percentage.

In explaining completeness and contamination we restricted the MAGs to be bacterial. The same reasoning applies to archaea; but, of course, we need to have a core archaeal gene set in order to measure completeness and contamination in MAGs that are classified as archaea.

Table 1
The MIMAG Standard for MAG quality

Category	Assembly quality	Completion	Contamination
Finished	Single contiguous sequence without gaps or ambiguities with a consensus error rate equivalent to Q50 or better	100%	0
High-quality draft	Multiple fragments where gaps span repetitive regions; presence of the 23S, 16S, and 5S rRNA genes and at least 18 tRNAs	>90%	<5%
Medium-quality draft	Many fragments with little to no review of assembly other than reporting of standard assembly statistics	>50%	<10%
Low-quality draft	Many fragments with little to no review of assembly other than reporting of standard assembly statistics	<50%	<10%

Adapted from Ref. [17]

3.2 MAG Quality Standards

Now that we have presented the essentials of MAG quality evaluation, there comes the next question: what quality values make a MAG "good enough"? We will answer this question by presenting a proposal for minimum quality expectations for MAGs published by Bowers et al. [17] and called the MIMAG standard. The proposal is summarized in Table 1.

In the literature, it is generally the case that the category of medium-quality draft or higher is adopted for reporting the overall number of MAGs.

4 MAG Annotation

Once we have MAGs that are preferably at least medium-quality draft, the next step is to annotate them. Annotation is basically the process of locating genes in the genome and inferring what their function is. There is nothing in MAG annotation that differs from isolate genome annotation. Therefore, to annotate a MAG, one can simply run an automated annotation pipeline. This can be done in a number of ways [18–20].

5 MAG Comparative Analysis

5.1 MAG Pairwise Comparison

After annotating a MAG, we are in a position to perform comparative analyses. The most basic comparison is one that will simply align the MAG with other genome sequences, both from isolates or other MAGs. This can be done with a number of tools; among them, we cite MUMmer [21] and FastANI [22].

MUMmer computes and displays an alignment between any two genomes. It is fast, aligning typical bacterial genomes in seconds on a standard laptop. FastANI aims to compare two genomes in terms of their average nucleotide identity (ANI), expressing the result as a percentage (i.e., it does not actually align the genomes being compared). FastANI is also very efficient and is a recommended tool for doing large-scale (i.e., hundreds or thousands) pairwise comparisons among bacterial genomes.

5.2 MAG Databases

With the explosion of MAGs in the literature, some research groups have created MAG databases. Note that, ideally, there should be a central MAG repository that would store and make available to researchers worldwide all MAGs that have been obtained by different groups, with frequent updates. This is the role that GenBank still plays with respect to isolate genomes. Unfortunately, there is no central repository for MAGs, which means that researchers interested in comparing their MAGs to other MAGs need to do searches against different databases. Here, we briefly describe two of these.

The Genomes from Earth's Microbiomes (GEM) catalog [23] contained, at the time of publication, 52,515 MAGs, assembled from 10,450 metagenome datasets obtained from samples of diverse microbial habitats and worldwide geographic locations. It is hosted by the Integrated Microbial Genomes and Microbiomes (IMG/M) platform [24] at the Joint Genome Institute. All MAGs from the GEM catalog meet or exceed the medium-quality level of the MIMAG standard (mean completeness = 83%; mean contamination = 1.3%). Of the total, 9,143 (17.4%) MAGs can be considered high quality.

The Unified Human Gastrointestinal Genome (UHGG) collection [25] contained, at the time of publication, 204,938 nonredundant genomes from 4,644 human gut prokaryotes. Not all of these genomes are MAGs, but the vast majority are. Before redundancy verification, the total number of genomes was 286,997, of which 276,349 (96.3%) were MAGs. The authors do not give the exact number of nonredundant MAGs in their catalog, but it can be assumed that it is close to 200,000. For the nonredundant set, the quality verification was as follows: at least 50% genome completeness and at most 5% contamination. The authors further combined these two measures using the formula $completeness - 5 \times contamination$, and required each genome in the set to have a value of at least 50. This catalog is hosted by the European Bioinformatics Institute at https://www.ebi.ac.uk/metagenomics/genomes.

5.3 Taxonomic Classification

One important motivation for MAG comparisons is the need to identify the organism to which a MAG belongs. This is called taxonomic classification. There are several programs and platforms that perform this task. In this chapter, we will focus on one popular tool called GTDB-Tk [26, 27].

GTDB-Tk is a software toolkit for taxonomic classifications of bacterial and archaeal genomes based on the Genome Taxonomy Database (GTDB) [28]. It achieves classification by placing genomes in reference trees. GTDB-Tk first uses Mash [29] and FastANI [22] against all representative genomes on the GTDB database. Then, if the genomes are not classified by similarity with Mash and FastANI, HMMER [30] is used to identify a set of bacterial and archeal marker genes. These genes are concatenated and aligned, and then placed in a reference tree using the tool pplacer to determine the closest taxon in the tree.

Although the methods for obtaining and doing quality control of MAGs are constantly being improved, the fact remains that a MAG does not, by definition, correspond to an isolate. This means that one has to be careful about the biological reality of any given MAG. Prompted by this consideration, Setubal [31] proposed that MAGs can be classified into three categories:

SMAGs (Species MAGs) MAGs for which a species can be assigned. This assignment assumes that there exists an isolate genome that has been correctly classified as belonging to species *S*, and that the ANI between the MAG and the isolate genome from *S* is at least 97% (some authors use 98%). It is good practice to also require that an alignment between the MAG and the *S* genome covers at least 80% of both genomes.

CHMAGs (or conserved hypothetical MAGs) These are MAGs that have ANI against another MAG (or another isolate genome for which there is no species assignment) satisfying the same requirements as that for a SMAG.

HMAGs (hypothetical MAGs) These are MAGs that cannot be classified as either SMAGs or CHMAGs. The use of the word hypothetical in the definition stresses the fact that we do not have strong evidence that the MAG indeed corresponds to the genome of an actual organism.

Most MAGs from nonhuman samples are HMAGs [32]. In the GEM catalog, at the time of publication, 12,556 nonredundant MAGs could be considered HMAGs, out of 18,028 nonredundant MAGs, that is, a fraction of nearly 70%. If one assumes that the MAG recovery methods, by and large, recover correct genomes, this high fraction can be interpreted to mean that most prokaryotes are still unknown (belonging the so-called microbial dark matter).

Over time, as more isolate genome and metagenome sequencing is done, the expectation is that many CHMAGs will become SMAGs and that many HMAGs will become either CHMAGs or even SMAGs.

5.4 MAGset and MAGcheck

The authors have developed two tools to facilitate MAG comparison. In this section, we describe them briefly.

MAGset accepts as input one MAG and a set of reference genomes. The reference genomes should belong to the same species as the MAG, which means that in principle MAGset is aimed at SMAGs (it can also be used for CHMAGs). MAGset provides the user with information about genomic regions and genes present in the MAG and absent in the reference genomes and vice versa. This is based on the concept of genomic region of interest (GRI). The definition of a GRI is that it is a genomic region of size at least 5 kbp that is present in the MAG and absent in all reference genomes (in which case we call it a *positive GRI*); or it is a genomic region with the same size constraint present in at least one reference genome but absent in the MAG (in which case we call it a *negative GRI*).

In addition to finding GRIs, the software offers a user-friendly interface, through which the user can make searches to better understand the differences between the MAG and the reference genomes. The search mechanism was based on the search mechanism provided by the platform VEuPathDB (https://veupathdb.org/veupathdb/app).

MAGcheck is an accessory module of MAGset. After negative GRIs have been found in a given comparison, MAGcheck can search among the reads that were used to obtain the MAG and determine whether any of the negative GRIs can actually be found, in full or in part, among the reads. If there are reads covering at least 80% of a negative GRI, the user can, if desired, try to improve the assembly of the MAG being studied. The functionality for this improvement is not provided by MAGcheck, since it requires a specific re-assembly of the MAG in question.

6 Practical Example

In this section, we offer a complete example of how to generate, evaluate, and compare MAGs based on a real-world public dataset.

6.1 Assumptions

In order to run the example, we assume the reader is familiar with Linux (and the bash shell) and has access to a server running Linux with at least 100 GB of RAM. The programs listed in Table 2 need to be installed on the server.

6.2 Sample Description

The metagenome dataset used in this example comes from 1163 fecal samples from premature infants [33]. Fecal samples are considered to contain primarily DNA of the gut microbiota.

Table 2
Programs used in the example

Name	Version used in this chapter	Download URL
SRA Toolkit	3.0.0	https://github.com/ncbi/sra-tools/wiki/01.-Downloading-SRA-Toolkit
MetaWRAP	1.3.0	https://github.com/bxlab/metaWRAP
GTDB-Tk	2.1.1	https://github.com/Ecogenomics/GTDBTk
FastANI	1.33	https://github.com/ParBLiSS/FastANI
PGAP	2022-12-13.build6494	https://github.com/ncbi/pgap
MAGset	1.5.0	https://github.com/LaboratorioBioinformatica/MAGset

6.3 Downloading the Data

The data is publicly available for download on the NCBI website (https://www.ncbi.nlm.nih.gov/) using the SRA Toolkit. We do this by running the following commands:

```
$ prefetch SRR3466404 --max-size 420000000000
$ fasterq-dump SRR3466404
```

where:

- `prefetch` is a tool that downloads the data from NCBI. Parameters are:
  - `SRR3466404` is the reference code of the dataset we are downloading. Generally, this code is available inside the article when the data is public.
  - `--max-size` indicates the maximum download size allowed (this parameter is necessary because this dataset is bigger than the default limit size).
- `fasterq-dump` is a tool to convert the original NCBI download format to the format we needed for the next steps (fastq). `SRR3466404` is the reference code of the data we downloaded.

After executing these commands, two metagenomic files (paired-end read files) will be available inside the folder:

- `SRR3466404_1.fastq`
- `SRR3466404_2.fastq`

6.4 Preprocessing

We want to ensure that all sequences are of high quality and without artifacts. Therefore, we preprocess the raw reads by trimming them using the `read_qc` module from MetaWRAP. This module preprocesses the reads based on the sequencing

quality assigned to each nucleotide. By default, MetaWRAP uses Trim Galore for quality trimming and Cutadapt for adapter removal. We use the following command line for the preprocessing step:

```
$ metawrap read_qc -1 SRR3466404_1.fastq -2
SRR3466404_2.fastq -t 4 --skip-bmtagger -o READ_QC
```

The parameters `-1` and `-2` refer to the forward and reverse raw reads respectively. By default, the `read_qc` module includes decontamination by aligning the reads to a host genome (e.g., the human genome) and by removing the reads that have more than a given similarity threshold. In this practical example, we use the `--skip-bmtagger` flag to skip this decontamination step, assuming that the raw reads from the downloaded dataset are not contaminated. After `read_qc` execution, two HTML reports are generated, one for the raw reads and another after quality trimming (respectively `pre-QC_report` and `post-QC_report`).

6.5 MAG Assembly Pipeline

The main wrapper function of MetaWRAP consists of separate modules for assembly, initial binning, and binning refinement. For assembly, we use the option MEGAHIT and execute the following command line:

```
$ metawrap assembly -1 READ_QC/*_1.fastq -2
READ_QC/*_2.fastq -m 100 -t 12 --megahit -o ASSEM-
BLY
```

Alternatively, metaSPAdes [7] can be used in the MetaWRAP pipeline. metaSPADES is an excellent assembler; it generates the best contig size statistics compared to other assemblers. However, it requires large RAM memory. MEGAHIT is a good option if the available RAM is less than, say, 16 GB.

After obtaining the contigs by assembling the reads, we execute the initial binning using three different tools: MaxBin2 [34], metaBAT2 [12], and CONCOCT [35]. MetaWRAP offers the option of executing the binning step with all three binners in one command line:

```
$ metawrap binning -o INITIAL_BINNING -t 12 -a AS-
SEMBLY/final_assembly.fasta --metabat2 --maxbin2
--concoct READ_QC/*.fastq
```

MetaWRAP includes a module that is able to combine the results of all three binning tools and yield a result that should be better than the one generated by any of these binners separately. This can be done by the following command:

```
$ metawrap bin_refinement -o BIN_REFINEMENT -t
12 -m 100 --quick -A INITIAL_BINNING/metabat2_bins/
-B INITIAL_BINNING/maxbin2_bins/ -C INITIAL_BIN-
NING/concoct_bins/ -c 50 -x 10
```

The bin sets from each binning tool are combined and compared against each other. The parameter `-c` refers to the minimum required completeness and `-x` to the maximum contamination allowed. In the end, we obtain a set of reconstructed MAGs and a report of their completeness, contamination, size, and other metrics, as given by CheckM [16]. The reports can be visualized using the following command line:

```
$ cat BIN_REFINEMENT/metawrap_50_10_bins.stats
```

In our example, we obtain 28 MAGs with the metrics shown in Table 3.

In general, MAGs will be a fraction of all the data contained in a metagenome dataset. This fraction will vary according to many factors, such as the minimum completeness threshold used to generate MAGs, microbial diversity in the sample, and sequencing depth. For the dataset we are using in this example, this fraction is 70%, if we simply count the number of contigs in MAGs relative to the total number of contigs, or 87%, if we count the number of sequence base pairs in MAGs relative to the total number of sequence base pairs in contigs (Table 4). This result suggests that the sample in the example is not particularly diverse, and the sequencing depth allowed us to capture a large fraction of this diversity in the MAGs. Note, however, that any comprehensive microbiome analysis of the studied environment (the human gut in this case) should take into account not only the MAGs but also contigs not included in MAGs. The analysis of such contigs is outside the scope of our example.

6.6 Taxonomic Classification

To obtain the taxonomic classification of the MAGs assembled in the previous step, we use GTDB-Tk [27]. We do this by running the following command:

```
$ gtdbtk classify_wf --min_af 0.8 --extension fa --
genome_dir /work/analysis/metawrap/BIN_REFINE-
MENT/metawrap_50_10_bins/ --out_dir gtdbtk_result
```

where:

- `classify_wf` is the default workflow to generate the taxonomic classification.
- `--min_af` is the minimum alignment necessary to assign the MAG to a species. Here, we are using 0.8 (80%), but the default value is 0.5 (50%).

Table 3
MAGs obtained

ID	Completeness (%)	Contamination (%)	N50 (bp)	Size (bp)
bin.1	95.20	1.19	9,277	2,071,822
bin.10	99.05	0.59	248,623	2,154,473
bin.11	98.71	0.46	151,207	4,092,388
bin.12	97.60	0.75	129,685	1,987,272
bin.13	96.45	1.74	15,654	2,110,144
bin.14	60.34	0.00	197,721	2,272,956
bin.15	97.12	0.60	97,498	5,098,176
bin.16	94.57	0.08	132,720	2,615,966
bin.17	98.87	0.00	117,227	2,752,442
bin.18	64.27	0.20	116,114	2,943,683
bin.19	97.58	1.81	55,583	3,330,466
bin.2	88.70	0.00	89,992	4,430,022
bin.20	82.85	1.32	4,748	5,912,272
bin.21	66.12	0.81	129,447	1,854,467
bin.22	96.78	0.13	87,935	2,325,206
bin.23	51.61	0.81	135,731	1,899,461
bin.24	90.29	2.30	5,036	2,126,201
bin.25	99.25	0.75	202,714	3,250,245
bin.26	94.15	0.15	229,507	2,253,828
bin.27	75.86	0.00	35,240	2,789,627
bin.28	51.07	4.27	1,815	2,202,426
bin.3	97.58	3.49	16,046	2,320,520
bin.4	55.08	1.64	2,084	1,517,918
bin.5	99.28	0.95	248,309	2,404,125
bin.6	99.90	2.03	99,790	3,855,070
bin.7	94.26	0.81	5,777	1,503,206
bin.8	100.00	0.00	237,187	2,485,664
bin.9	100.00	0.00	101,443	1,806,569

Table 4
Information about contigs

	All contigs of the assembly	Contigs not included in MAGs
Number of contigs	9,029	2,718
Length of shortest contig (bp)	1,000	1,000
Length of longest contig (bp)	674,977	478,425
Average length of contigs (bp)	9,687.58	4,084.82
Total size (bp)	87,469,161	11,102,546
N50 (bp)	64,981	8,610
L50 (bp)	301	195
% GC	43.51	41.34

- `--extension` is the extension of the MAG files; in our case, it is "`.fa`".
- `--genome_dir` is the folder in which MAG files can be found.
- `--out_dir` is the folder in which the GTDB-Tk results will be placed.

Once execution is complete, the result folder will contain one file called `gtdbtk.bac120.summary.tsv`. This file indicates the best classification obtained for each MAG. As this file is tab-delimited, to visualize the data, we suggest using a spreadsheet program. The most important columns for our objective are:

- **classification**: Indicates the classification level of the MAG. If one species name is filled after the "s__" content, the MAG was classified at the species level. Otherwise, if the column content finishes just with "s__", the MAG was not classified at the species level.
- **fastani_reference**: Indicates (for the MAGs classified at the species level) which reference genome was considered to classify the MAG at that level.
- **fastani_ani:** Indicates the ANI result between the MAG and the reference.

It is necessary to pay attention to the reference genome because sometimes the reference genome is another MAG (it is not an isolate genome). This situation may mean that the classification is not as reliable as one obtained by reference to an isolate genome. The results of GTDB-Tk are summarized in Table 5.

Table 5
Taxonomic classification using GTDB-Tk

ID	Classification level	Classification	Reference	ANI Reference (%)
bin.1	Species	s__Enterococcus_B faecium	GCF_001544255.1	99.32
bin.2	Species	s__Clostridium butyricum	GCF_006742065.1	98.85
bin.3	Species	s__Staphylococcus epidermidis	GCF_006742205.1	98.95
bin.4	Species	s__Cutibacterium acnes	GCF_003030305.1	99.1
bin.5	Species	s__Pauljensenia radingae_A	GCA_900106055.1	98.94
bin.6	Species	s__Anaerosporomusa sp900542835	GCA_900542835.1	99.97
bin.7	Species	s__Atopobium minutum	GCF_001437015.1	99.83
bin.8	Species	s__Cutibacterium avidum	GCF_000227295.1	99.08
bin.9	Species	s__Streptococcus lutetiensis	GCF_900475675.1	99.11
bin.10	Species	s__Varibaculum cambriense_A	GCA_000508625.1	97.46
bin.11	Species	s__Mixta calida	GCF_002953215.1	99.48
bin.12	Species	s__Veillonella parvula_A	GCF_902810435.1	96.92
bin.13	Species	s__Agathobacter rectalis	GCA_000020605.1	97.14
bin.14	Species	s__Clostridium baratii	GCF_000789395.1	98.71
bin.15	Species	s__Klebsiella pneumoniae	GCF_000742135.1	99.07
bin.16	Species	s__Staphylococcus aureus	GCF_001027105.1	99.81
bin.17	Species	s__Enterococcus faecalis	GCF_000392875.1	98,9
bin.18	Species	s__Escherichia coli	GCF_003697165.2	96.76
bin.19	Species	s__Clostridium_X cadaveris	GCF_000424205.1	99.71
bin.20	**Genus**	g__Hungatella	**N/A**	**N/A**
bin.21	**Genus**	g__Clostridium	**N/A**	**N/A**
bin.22	Species	s__Staphylococcus warneri	GCF_900636385.1	99.6
bin.23	Species	s__Clostridium paraputrificum	GCF_900447045.1	95.4
bin.24	Species	s__Corynebacterium aurimucosum_E	GCF_016127015.1	97.13
bin.25	Species	s__Enterococcus_D gallinarum	GCF_001544275.1	97.58
bin.26	Species	s__Dermabacter hominis	GCF_001570785.1	96.02
bin.27	Species	s__Clostridium sp900547475	GCA_900547475.1	95,18
bin.28	**Genus**	g__Clostridioides	**N/A**	**N/A**

6.7 Searching MAGs Against the GEM Database

In addition to obtaining a classification for our MAGs, it is of interest to determine whether the MAGs we recovered have been found in other environments. Some of this information is given already by the GTDB-Tk results, since a classification at the species level implies that there is at least one other genome of the same species that was isolated from some sample (or, rarely, another MAG, as explained above).

In this example, we will show how our MAGs can be searched against the GEM database (described in Subheading 5.2). Because our metagenome dataset comes from the human gut, it would make more sense to search against the UHGG collection (also described in Subheading 5.2). However, the UHGG database is too large for local processing; so, we chose to demonstrate the MAG database search step using GEM.

First of all, it is necessary to download the GEM database [23] from its repository:

https://portal.nersc.gov/GEM/genomes/

As we want to compare FASTA files, download the `fna.tar` file, and extract the contents in a folder of your preference. The commands are as follows:

```
$ curl -o fna.tar https://portal.nersc.gov/GEM/genomes/fna.tar
$ mkdir /work/databases/GEM
$ tar xvf fna.tar -C /work/databases/GEM/
```

After extracting the GEM database, generate one file with the list of all GEM genomes with the complete path, using the command:

```
$ find /work/databases/GEM/fna > list_fna_GEM.txt
```

In the next step, we compare our MAGs against all GEM genomes. As the GEM database has more than 52,000 genomes, it may be necessary to split the process to use less memory. Here, we will split it into ten batches of a maximum of 5500 lines each.

```
$ split -d -l5500 list_fna_GEM.txt input_GEM_mags_list/GEM_list_split_
```

This command will generate ten files inside the folder "`input_GEM_mags_list`", named as `GEM_list_split_{0..9}` (numbered from zero to nine).

After generating the files with the list of GEM genomes, we need to do the same process to generate a file with all assembled MAGs. The content of this file should be all genome file paths obtained in Subheading 6.5 (one file per line):

```
$ find /work/analysis/metawrap/BIN_REFINEMENT/me-
tawrap_50_10_bins/ > input_mags_from_reassembly.
txt
```

Please make sure to change the path where the MAGs are available at the command above if you are using a different folder structure.

In the next step, we use FastANI to compare the genomes. The following command compares our assembled genomes with the GEM genomes:

```
$ for i in input_GEM_mags_list/*; do out="$(base-
name -- $i)"; fastANI -t 16 --minFraction 0.8 --ql
input_mags_from_reassembly.txt --rl "$i" -o "re-
sult_01_$out.txt"; done;
```

where:

- `for i in input_GEM_mags_list/*` iterates between the files.
- `-t 16` is the threads quantity to execute the software.
- `minFraction 0.8` is the minimum alignment necessary to define that two genomes are similar enough to be reported. Here, we are using 0.8 (80%), but the default value is 0.2 (20%).
- `--ql input_mags_from_reassembly.txt` is the genome list to compare against the GEM genomes. In our case, it is the list of assembled genomes in previous steps.
- `--rl "$i"` is the GEM genome list files.
- `-o "result_01_$i.txt"` is the result file of the comparison.

The command above took about 3 hours on a server with 16 processors and 100 GB of RAM. The result of this command will be a list of files with the results found by FastANI, but we need to do some additional steps to get the results we are interested in:

```
$ cat result_01_*.txt > result_02_ani.txt
$ rm result_01_*.txt
$ awk '{ if ($3 >= 95) { print } }' result_02_ani.txt
> result_03_ani_filtered_by_greater_than_95.txt
$ sort -k1,1 -k3nr,3 result_03_ani_filtered_by_-
greater_than_95.txt | sort -k1,1 -u > result_04_i-
dentified_chmags.txt
$ sort input_mags_from_reassembly.txt > resul-
t_05_original_mags_from_article_sorted.txt
$ sort result_04_identified_chmags.txt > resul-
t_05_identified_chmags_sorted.txt
$ comm -23 result_05_original_mags_from_article_-
sorted.txt result_05_identified_chmags_sorted.txt
> result_05_identified_hmags.txt
```

```
$ rm result_05_original_mags_from_article_sorted.
txt result_05_identified_chmags_sorted.txt
```

where:

- `cat` merges all result files in one file.
- `rm` removes intermediary files.
- `awk` keeps results where the ANI is equal to or greater than 95%.
- `sort -k1,1 -k3nr,3` sorts the results by MAG name (first column) and by ANI result (third column).
- `sort -k1,1 -u` keeps just the better match for each MAG. This returned file contains the MAGs with matches against GEM database—Table 6).
- `sort` sorts the files (necessary for the next command).

Table 6
Best hits for MAGs against the GEM database

MAG	GEM best hit	ANI	Length (Mbp)	Contigs	Completeness	Contamin.	Quality
bin.1	3300029065_2	99.28	2.87	90	99.63	0	MQ
bin.3	3300013570_1	98.99	2.39	34	99.25	0	MQ
bin.4	3300006215_1	99.13	2.47	13	98.94	0.03	MQ
bin.5	3300014916_9	99.02	2.05	204	87.31	2.23	MQ
bin.8	3300013226_1	96.06	2.46	105	99.01	0.33	MQ
bin.9	3300010283_34	99.23	0.98	25	59.18	0	MQ
bin.10	3300029005_8	97.49	2.10	40	98.82	0.83	HQ
bin.11	3300014642_2	99.62	4.15	33	99.3	0.46	HQ
bin.12	3300006255_16	97.18	1.03	150	63.81	1.42	MQ
bin.13	3300008404_12	98.24	2.50	48	99.03	1.69	MQ
bin.15	3300011751_3	99.19	4.92	40	95.54	0.57	MQ
bin.16	3300007865_1	97.95	2.02	28	75.77	0.08	MQ
bin.17	3300014970_8	99.86	2.87	43	99.63	0	HQ
bin.18	3300029613_8	99.19	4.69	69	99.62	0.21	HQ
bin.22	3300027028_14	99.88	2.33	46	98.83	0.08	MQ
bin.23	3300014583_1	97.62	3.66	111	95.97	1.81	MQ
bin.24	3300012579_3	97.31	2.16	135	97.43	1.32	MQ
bin.25	3300014531_12	99.73	2.13	32	69.81	0	MQ

The columns "Length," "Contigs," "Completeness," "Contamin.," and "Quality" were obtained from https://portal.nersc.gov/GEM/genomes/genome_metadata.tsv

- `comm` creates a file containing MAG identifiers without hits against the GEM database.
- `rm` removes intermediary files.

The results obtained are shown in Table 6.

6.8 Merging GTDB-Tk and GEM Comparison Results

Now that we have results from both GTDB-Tk and GEM searches, we can compare these results (Table 7) and arrive at a classification of our MAGs in terms of SMAGs, CHMAGs, and HMAGs, as explained in Subheading 5.3. If the MAG has results in GTDB-Tk against an isolate genome, it will be considered a SMAG. If the MAG has results only in the GEM database, it will be considered a CHMAG. MAGs without results will be considered HMAGs. The results are shown in Table 8.

It is important to note that GTDB-Tk results sometimes are not isolate genomes, so it is necessary to check at the NCBI website whether the genome that is a hit is from an isolate. To check this, open the NCBI website (https://www.ncbi.nlm.nih.gov/) and search for the returned reference code. As an example, the reference code "GCF_001544255.1" was returned by GTDB-Tk as reference for the MAG "bin.1" (Table 5). At the NCBI website it is possible to verify that the hit genome was assembled from type material (in other words, it is an isolate genome).

In Table 7, we show the results of both GTDB-Tk and the GEM searches (best hit only). This table shows that a number of our MAGs did not have any hits in the GEM database. This shows that the underlying genome database used by GTDB-Tk is more comprehensive than GEM. On the other hand, there were two discrepancies among MAGs that did have hits. We now analyze each discrepancy.

- Bin.11: The discrepancy here can be explained by synonymy. Using the NCBI taxonomy we see that *Pantoea calida* is just a synonym for *Mixta calida*.
- Bin.5: Again in the NCBI taxonomy, we find that *Pauljensenia radingae* is not a valid name, but the look-up produces *Schaalia radingae* instead; this species is a member of the Actinomycetaceae family. Something similar happens with the GEM hit: *Actinomyces bhumii* seems to be a synonym for *Actinomyces ihuae*, which is also a member of the Actinomycetaceae family. So, which species should be assigned to Bin.5? Because GTDB-Tk is a taxonomic classification program, and the GEM database by definition contains only MAGs, we should choose the GTDB-Tk classification, *Schaalia radingae*.

Putting these results together, we now arrive at a categorization of our MAGs according to the three categories mentioned

Table 7
Comparison between GTDB-Tk classification and GEM best hit classification

MAG	GTDB-tk Classification	GEM best hit classification
bin.1	s__Enterococcus_B faecium	s__Enterococcus_B faecium
bin.2	s__Clostridium butyricum	N/A
bin.3	s__Staphylococcus epidermidis	s__Staphylococcus epidermidis
bin.4	s__Cutibacterium acnes	s__Cutibacterium acnes
bin.5	s__Pauljensenia radingae_A	s__Actinomyces bhumii
bin.6	s__Anaerosporomusa sp900542835	N/A
bin.7	s__Atopobium minutum	N/A
bin.8	s__Cutibacterium avidum	s__Cutibacterium avidum_A
bin.9	s__Streptococcus lutetiensis	s__Streptococcus lutetiensis
bin.10	s__Varibaculum cambriense_A	s__Varibaculum cambriense_A
bin.11	s__Mixta calida	s__Pantoea_B calida
bin.12	s__Veillonella parvula_A	s__Veillonella parvula
bin.13	s__Agathobacter rectalis	s__Agathobacter rectalis
bin.14	s__Clostridium baratii	N/A
bin.15	s__Klebsiella pneumoniae	s__Klebsiella pneumoniae
bin.16	s__Staphylococcus aureus	s__Staphylococcus aureus
bin.17	s__Enterococcus faecalis	s__Enterococcus faecalis
bin.18	s__Escherichia coli	s__Escherichia coli
bin.19	s__Clostridium_X cadaveris	N/A
bin.20	**g__Hungatella**	N/A
bin.21	**g__Clostridium**	N/A
bin.22	s__Staphylococcus warneri	s__Staphylococcus warneri
bin.23	s__Clostridium paraputrificum	s__Clostridium paraputrificum_A
bin.24	s__Corynebacterium aurimucosum_E	s__Corynebacterium tuberculostearicum
bin.25	s__Enterococcus_D gallinarum	s__Enterococcus_D gallinarum
bin.26	s__Dermabacter hominis	N/A
bin.27	s__Clostridium sp900547475	N/A
bin.28	**g__Clostridioides**	N/A

Highlighted in gray are the MAGs in which the GTDB-Tk classification at the species level does not match the GEM best hit

previously (Table 8). We see that nearly all MAGs are SMAGs, with a few exceptions. This was expected because the samples come from the human gut, possibly the best-studied "environmental niche." Had the samples come from, for example, soil or lake water, which are far less known from a microbiome perspective than the human gut, then the proportion would be reversed: most MAGs would be HMAGs, and only a few would be SMAGs or CHMAGs.

Table 8
Categorization of MAGs

MAG	Found in GTDB-tk?	Found in GEM Database?	Category
bin.1	Yes	Yes	SMAG
bin.2	Yes	**No**	SMAG
bin.3	Yes	Yes	SMAG
bin.4	Yes	Yes	SMAG
bin.5	Yes	Yes	SMAG
bin.6	**Yes (MAG)**	**No**	**CHMAG**
bin.7	Yes	**No**	SMAG
bin.8	Yes	Yes	SMAG
bin.9	Yes	Yes	SMAG
bin.10	**Yes (MAG)**	Yes	**CHMAG**
bin.11	Yes	Yes	SMAG
bin.12	Yes	Yes	SMAG
bin.13	Yes	Yes	SMAG
bin.14	Yes	**No**	SMAG
bin.15	Yes	Yes	SMAG
bin.16	Yes	Yes	SMAG
bin.17	Yes	Yes	SMAG
bin.18	Yes	Yes	SMAG
bin.19	Yes	**No**	SMAG
bin.20	**No**	**No**	**HMAG**
bin.21	**No**	**No**	**HMAG**
bin.22	Yes	Yes	SMAG
bin.23	Yes	Yes	SMAG
bin.24	Yes	Yes	SMAG
bin.25	Yes	Yes	SMAG
bin.26	Yes	**No**	SMAG
bin.27	**Yes (MAG)**	**No**	**CHMAG**
bin.28	**No**	**No**	**HMAG**

6.9 Annotating MAGs with PGAP

After obtaining and classifying MAGs, the next step is to annotate them, using PGAP. To execute the software, it is necessary to have three files:

- `genome_file` with the FASTA sequences (contigs).
- `pgap_submol.yaml`, which contains additional configurations to execute PGAP. The most important configuration in this file is the species name.
- `pgap_metadata.yaml`. which contains the path of the genome file and the path of the file `pgap_submol.yaml`.

Here, as example, we present the content required for the files to annotate the `bin.1.fa`, assembled in the previous steps.

Example of `pgap_submol.yaml` File:

```
organism:
 genus_species: 'Enterococcus faecium'
```

This is the minimal file possible, with just the mandatory field (the species or genus name). Additional fields can be filled out.

The species or genus name needs to be a valid NCBI taxonomy name (https://www.ncbi.nlm.nih.gov/taxonomy). The easiest way to obtain the correct name is looking up the reference returned by GTDB-Tk (Table 5). In the case of bin.1, the reference is GCF_001544255.1. Searching at the NCBI site using the reference code, it is possible to determine the species taxonomy for this reference.

Example of `pgap_metadata.yaml` File:

```
fasta:
 class: File
 location: bin.1.fa
submol:
 class: File
 location: pgap_submol.yaml
```

This file indicates the path of the other files necessary to execute PGAP. In this example, all the files are in the same folder.

Command line to execute PGAP:

```
$ pgap -r -o result/ -c 8 -m 32g --ignore-all-errors
pgap_metadata.yaml
```

where:

- `-r` allows PGAP to send a usage report to its server.
- `-o result/` is where the output will be saved.
- `-c 8` is the quantity of CPUs PGAP is allowed to use.
- `-m 32g` is the memory size PGAP is allowed to use.
- `--ignore-all-errors` indicates that PGAP should continue with the execution even if some errors occur.
- `pgap_metadata.yaml` is a file already described in this section.

The command above took about 4 hours on a server with 16 processors and 100 GB of RAM. The folder `result` will contain the annotated genome. The file of interest is `annot.gbk`, which contains all annotated genes and the original FASTA sequences.

6.10 Comparing and Analyzing MAGs with MAGset

To compare the genomes, we will use the software MAGset. To execute MAGset, the following files are necessary:

- MAG file (annotated)
- Reference genome file (annotated)
- Sample data files (raw data, the source fastq files where the MAGs were obtained)
- `configuration.properties`, the file with all configurations necessary to execute the software

Here, we will exemplify the use of MAGset with MAG bin.1. The annotated file for this MAG was generated in the previous step of this section.

To download the annotated reference genome file, we have two options:

First, using the **datasets** program, available at https://www.ncbi.nlm.nih.gov/datasets/docs/v2/download-and-install/ and executing the following command (where GCF_001544255.1 is the code of the reference given by GTDB-Tk for bin.1):

```
$ datasets download genome accession
GCF_001544255.1 --include gbff
```

Alternatively, we can use the NCBI site directly, in the genome page, to download the GBFF file.

Extract the reference genome and copy the MAG file to the same folder. As an example, we will use a folder with the path `/work/analysis/magset/genomes`.

The `configuration.properties` file should be like the following example:

```
title=enterococcus_faecium
```

```
genomes_folder=/work/analysis/magset/genomes/
output_folder=/work/analysis/magset/
mag_file=bin.1.gbff
reference_genome_files=GCF_001544255.1.gbff
num_threads=8
input_type=GBK
raw_reads_folder=/work/analysis/raw_data/
raw_reads_files_r1=SRR3466404_1.fastq
raw_reads_files_r2=SRR3466404_2.fastq
```

where:

- `title` is the result name for user reference; this title will be showed inside the html result.
- `genomes_folder` is where genomes were saved.
- `output_folder` is where the result will be saved.
- `mag_file` is the name of the MAG file, available inside the genomes_folder. Please copy the annotated genome obtained in Subheading 6.9 inside the `genomes_folder` using the name `bin.1.gbff`.
- `reference_genome_files` is the name of the reference genome file, available inside the genomes_folder. Please copy the downloaded reference genome file inside the `genomes_folder` using the name `GCF_001544255.1.gbff`.
- `num_threads` are the number of threads the software is allowed to use.
- `input_type` is the genome data format we are using; in this example, we are using GBK format.
- `raw_reads_folder` is the folder where the sample fastq files are saved.
- `raw_reads_files_r1` is the forward fastq file we used to assemble the MAG.
- `raw_reads_files_r2` is the reverse fastq file we used to assemble the MAG.

To execute MAGset, run the following command:

```
$ /work/apps/run-magset.sh configuration.properties
```

Please note that, in the command above, it is assumed that MAGset was installed in folder `/work/apps/`. After the end of the execution, inside the folder `result/html`, open the file `index.html` with a browser, and it will be possible to analyze the result of the comparison between the MAG and the reference genome (Fig. 2). The first page is a summary report, showing:

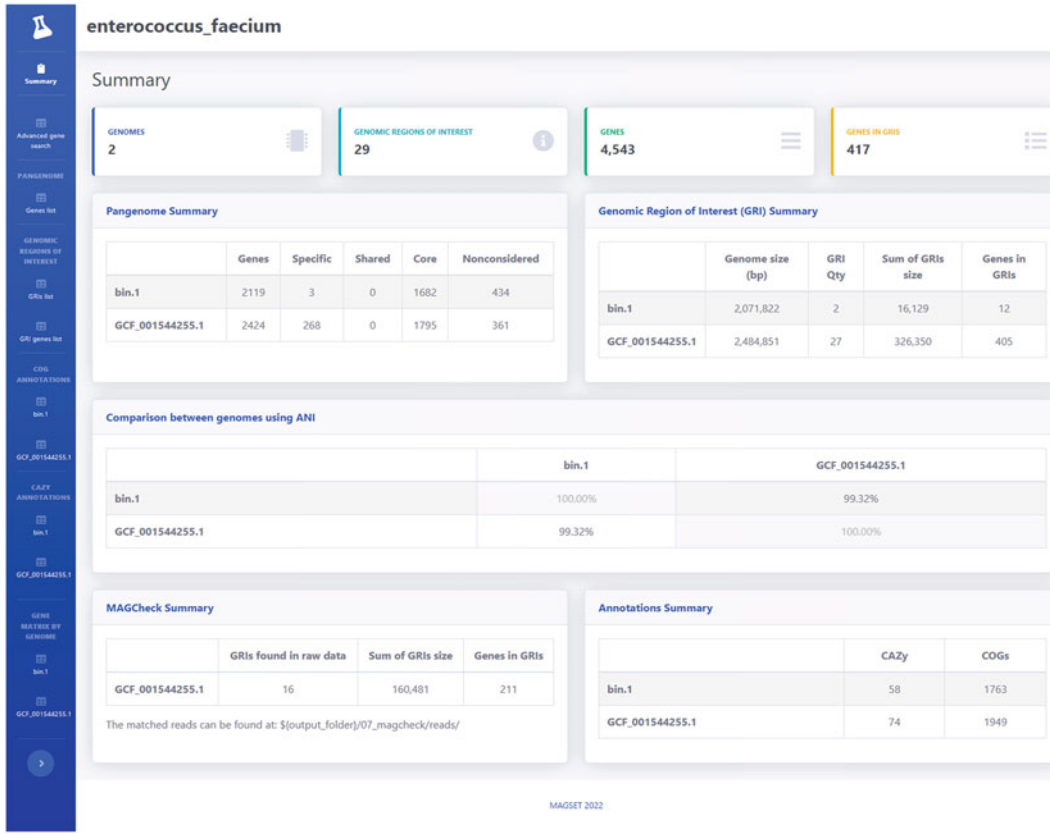

Fig. 2 MAGset screen, showing results comparing one MAG and one reference genome

- The pangenome of the genomes (specific and shared genes)
- Genomic regions of interest (regions that exist in one genome and do not exist in the other)
- ANI result between the genomes
- List of genomic regions of interest that could be available inside the sample data (raw data)
- Annotations summary (CAZY and COG annotations)

We now briefly discuss the results generated by MAGset. As seen in Table 5, our MAG (bin.1) was classified by GTDB-Tk as *Enterococcus faecium*. The reference genome is GCF_001544255.1. Their similarity in terms of ANI is given as 99.32% (see frame "Comparison between genomes using ANI" in Fig. 2); this value is consistent with the taxonomic classification. Note, however, that the sizes of the two genomes are substantially different: The reference genome has 2,484,851 bp, whereas our MAG has 2,071,822 bp, a difference greater than 400 kbp. To some extent this is expected, because MAGs almost always are incomplete. Indeed, Table 3 shows that CheckM estimates that bin.1 is 95.2% complete. If we were to use the reference genome

size as the true length of the genome to which our MAG corresponds, then its completeness would be only 83.4%, suggesting that CheckM's value may be an overestimate. (We note that CheckM2 became available after this chapter was completed [36]. We ran it on our MAGs and verified that nearly all completeness and contamination estimates changed, although usually by only a few percentage points. In the case of bin.1, the change in completeness was from 95.2% to 90.16%, supporting the hypothesis that CheckM's estimate was an overestimate for this MAG. However, on our 28 MAGs, we did not observe a consistent increase or decrease in the completeness estimates. In about half the MAGs there was a decrease, while there was an increase in the other half.)

Frame "Genomic Region of Interest (GRI) Summary" in Fig. 2 shows that 27 negative GRIs were found in GCF_001544255.1 and only two in bin.1. This large difference is again suggestive of the incompleteness of bin.1. Indeed, as will be seen in the next section, many of these 27 negative GRIs can be found in the raw data. On the other hand, the two positive GRIs may contain valuable information about genes that are specific to the strain that was retrieved from this metagenome dataset. Using the search mechanism provided by MAGset, a user can get detailed information on the gene contents of these two regions. In this particular case, one of the genes in one of these regions is annotated as coding for a group II intron reverse transcriptase/maturase, which is evidence that the GRI was the result of some genome insertion event, consistent with its absence from the reference genome.

6.11 Running MAGcheck

We use the MAGcheck module to verify whether negative GRIs of the MAG can be found in the metagenome dataset that we started with. This module is automatically run when MAGset is executed, by providing a file with the reads that were used to generate the MAG, as we did in the previous step (parameters `raw_reads_folder`, `raw_reads_files_r1`, and `raw_reads_files_r2` in the `configuration.properties` file). The results are output in the same HTML page where the MAGset results are presented, as menu item "GRIs list," or as a CSV file (`result/csv/gri_list.csv`).

For our example, we show the results in Table 9. Of the 27 negative GRIs, MAGcheck was able to find 16. This suggests that the binning of this metagenome dataset could be improved. If a user is particularly interested in refining the assembly of a MAG, the reads that correspond to the negative GRIs that were found in the raw data can be extracted, along with the reads that went into the original assembly of this MAG, and a new assembly can be attempted. It is our experience that the assemblies of specific MAGs can indeed be improved in this way. By improvement, we mean that size and completeness generally increase, while there may be a slight increase in contamination (as determined by CheckM).

Table 9
Negative genomic regions of interest (NGRI)

Id	Size (bp)	Genes Qty	Covered positions (%)	Found by MAGcheck
NGRI0001_01	8,344	5	99.99	true
NGRI0002_01	6,320	5	97.58	true
NGRI0003_01	11,570	13	66.93	false
NGRI0004_01	11,392	13	98.62	true
NGRI0005_01	17,111	30	100.00	true
NGRI0006_01	7,323	8	99.92	true
NGRI0007_01	7,926	5	79.98	false
NGRI0008_01	32,109	51	53.27	false
NGRI0009_01	5,018	7	99.78	true
NGRI0010_01	11,616	14	98.95	true
NGRI0011_01	16,865	14	24.67	false
NGRI0012_01	20,850	22	22.16	false
NGRI0013_01	13,722	17	57.10	false
NGRI0014_01	7,043	6	1.43	false
NGRI0015_01	5,205	9	99.83	true
NGRI0016_01	27,208	36	67.99	false
NGRI0017_01	6,286	10	100.00	true
NGRI0018_01	5,671	9	99.93	true
NGRI0019_01	9,194	7	98.81	true
NGRI0020_01	6,387	6	30.23	false
NGRI0021_01	10,915	11	52.67	false
NGRI0022_01	13,914	13	98.77	true
NGRI0023_01	5,243	5	99.64	true
NGRI0024_01	6,773	23	99.84	true
NGRI0025_01	22,037	36	93.32	true
NGRI0026_01	19,034	17	99.42	true
NGRI0027_01	11,274	13	20.76	false

Yellow lines indicate the NGRI was found by MAGcheck in the raw data

7 Conclusion

This chapter is an introduction to the concept of metagenome-assembled genome and the methods for obtaining and analyzing MAGs, and comparing them to other genomes, especially isolate genomes. Methods are continually improving, and new programs for some of the steps described here are constantly being developed and published. Nevertheless, we believe the analysis framework presented will be valid for some years. When new analysis programs become available, users should still be able to apply this same framework, replacing the particular method described here by a newer one.

Acknowledgments

This work was supported in part by CNPq grant 440230/2022-5, by a graduate student fellowship to SESG and by a CNPq senior researcher fellowship to JCS.

References

1. Berg G, Rybakova D, Fischer D, Cernava T, Verges MC, Charles T et al (2020) Microbiome definition re-visited: old concepts and new challenges. Microbiome 8(1):103
2. Setubal JC, Dias-Neto E (2022) Microbiome. Reference Module in Life Sciences, 2022. https://doi.org/10.1016/B978-0-12-822563-9.00081-0
3. Tully BJ, Graham ED, Heidelberg JF (2018) The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. Sci Data 5:170203
4. Zeng S, Patangia D, Almeida A, Zhou Z, Mu D, Paul Ross R et al (2022) A compendium of 32,277 metagenome-assembled genomes and over 80 million genes from the early-life human gut microbiome. Nat Commun 13(1): 5139
5. Stewart RD, Auffret MD, Warr A, Walker AW, Roehe R, Watson M (2019) Compendium of 4,941 rumen metagenome-assembled genomes for rumen microbiome biology and enzyme discovery. Nat Biotechnol 37(8): 953–961
6. Li D, Liu CM, Luo R, Sadakane K, Lam TW (2015) MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 31(10):1674–1676
7. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA (2017) metaSPAdes: a new versatile metagenomic assembler. Genome Res 27(5): 824–834
8. Vollmers J, Wiegand S, Kaster AK (2017) Comparing and evaluating metagenome assembly tools from a Microbiologist's perspective - not only size matters! PLoS One 12(1):e0169662
9. Yue Y, Huang H, Qi Z, Dou HM, Liu XY, Han TF et al (2020) Evaluating metagenomics tools for genome binning with real metagenomic datasets and CAMI datasets. BMC Bioinform 21(1):334
10. Arnold BJ, Huang IT, Hanage WP (2022) Horizontal gene transfer and adaptive evolution in bacteria. Nat Rev Microbiol 20(4): 206–218
11. Zhou F, Olman V, Xu Y (2008) Barcodes for genomes and applications. BMC Bioinform 9: 546
12. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H et al (2019) MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ 7:e7359
13. Bussi Y, Kapon R, Reich Z (2021) Large-scale *k*-mer-based analysis of the informational properties of genomes, comparative genomics and taxonomy. PLoS One 16(10):e0258693
14. Uritskiy GV, DiRuggiero J, Taylor J (2018) MetaWRAP-a flexible pipeline for genome-

resolved metagenomic data analysis. Microbiome 6(1):158

15. Haft DH, Selengut JD, Richter RA, Harkins D, Basu MK, Beck E (2013) TIGRFAMs and genome properties in 2013. Nucleic Acids Res 41(Database issue):D387-95
16. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res 25(7):1043–1055
17. Bowers RM, Kyrpides NC, Stepanauskas R, Harmon-Smith M, Doud D, Reddy TBK et al (2017) Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol 35(8): 725–731
18. Schwengers O, Jelonek L, Dieckmann MA, Beyvers S, Blom J, Goesmann A (2021) Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. Microb Genom 7(11):000685
19. Tatusova T, DiCuccio M, Badretdin A, Chetvernin V, Nawrocki EP, Zaslavsky L et al (2016) NCBI prokaryotic genome annotation pipeline. Nucleic Acids Res 44(14):6614–6624
20. Olson RD, Assaf R, Brettin T, Conrad N, Cucinell C, Davis JJ et al (2023) Introducing the Bacterial and Viral Bioinformatics Resource Center (BV-BRC): a resource combining PATRIC, IRD and ViPR. Nucleic Acids Res 51(D1):D678–DD89
21. Marcais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, Zimin A (2018) MUMmer4: A fast and versatile genome alignment system. PLoS Comput Biol 14(1): e1005944
22. Jain C, Rodriguez RL, Phillippy AM, Konstantinidis KT, Aluru S (2018) High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. Nat Commun 9(1):5114
23. Nayfach S, Roux S, Seshadri R, Udwary D, Varghese N, Schulz F et al (2021) A genomic catalog of Earth's microbiomes. Nat Biotechnol 39(4):499–509
24. Chen IA, Chu K, Palaniappan K, Ratner A, Huang J, Huntemann M et al (2023) The IMG/M data management and analysis system v.7: content updates and new features. Nucleic Acids Res 51(D1):D723–DD32
25. Almeida A, Nayfach S, Boland M, Strozzi F, Beracochea M, Shi ZJ et al (2021) A unified catalog of 204,938 reference genomes from the human gut microbiome. Nat Biotechnol 39(1):105–114
26. Chaumeil PA, Mussig AJ, Hugenholtz P, Parks DH (2019) GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. Bioinformatics 36(6):1925–1927
27. Chaumeil PA, Mussig AJ, Hugenholtz P, Parks DH (2022) GTDB-Tk v2: memory friendly classification with the genome taxonomy database. Bioinformatics 38(23):5315–5316
28. Parks DH, Chuvochina M, Waite DW, Rinke C, Skarshewski A, Chaumeil PA et al (2018) A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. Nat Biotechnol 36(10): 996–1004
29. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S et al (2016) Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol 17(1):132
30. Mistry J, Finn RD, Eddy SR, Bateman A, Punta M (2013) Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. Nucleic Acids Res 41(12): e121
31. Setubal JC (2021) Metagenome-assembled genomes: concepts, analogies, and challenges. Biophys Rev 13(6):905–909
32. Lloyd KG, Steen AD, Ladau J, Yin J, Crosby L (2018) Phylogenetically novel uncultured microbial cells dominate earth microbiomes. mSystems 3(5):e00055
33. Olm MR, Bhattacharya N, Crits-Christoph A, Firek BA, Baker R, Song YS et al (2019) Necrotizing enterocolitis is preceded by increased gut bacterial replication, Klebsiella, and fimbriae-encoding bacteria. Sci Adv 5(12): eaax5727
34. Wu YW, Simmons BA, Singer SW (2016) MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics 32(4):605–607
35. Alneberg J, Bjarnason BS, de Bruijn I, Schirmer M, Quick J, Ijaz UZ et al (2014) Binning metagenomic contigs by coverage and composition. Nat Methods 11(11): 1144–1146
36. Chklovski A, Parks DH, Woodcroft BJ, Tyson GW (2023) CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. Nat Methods 20(8):1203–1212