Cansdale and Chong *Microbiome (2024)12:226*
https://doi.org/10.1186/s40168-024-01949-z

Microbiome

METHODOLOGY

Open Access

MAGqual: a stand-alone pipeline to assess the quality of metagenome-assembled genomes

Annabel Cansdale[1*] and James P. J. Chong[1]

Abstract

Background Metagenomics, the whole genome sequencing of microbial communities, has provided insight into complex ecosystems. It has facilitated the discovery of novel microorganisms, explained community interactions and found applications in various fields. Advances in high-throughput and third-generation sequencing technologies have further fuelled its popularity. Nevertheless, managing the vast data produced and addressing variable dataset quality remain ongoing challenges. Another challenge arises from the number of assembly and binning strategies used across studies. Comparing datasets and analysis tools is complex as it requires the quantitative assessment of metagenome quality. The inherent limitations of metagenomic sequencing, which often involves sequencing complex communities, mean community members are challenging to interrogate with traditional culturing methods leading to many lacking reference sequences. MIMAG standards aim to provide a method to assess metagenome quality for comparison but have not been widely adopted.

Results To address the need for simple and quick metagenome quality assignation, here we introduce the pipeline MAGqual (Metagenome-Assembled Genome qualifier) and demonstrate its effectiveness at determining metagenomic dataset quality in the context of the MIMAG standards.

Conclusions The MAGqual pipeline offers an accessible way to evaluate metagenome quality and generate metadata on a large scale. MAGqual is built in Snakemake to ensure readability and scalability, and its open-source nature promotes accessibility, community development, and ease of updates. MAGqual is built in Snakemake, R, and Python and is available under the MIT license on GitHub at https://github.com/ac1513/MAGqual.

Keywords Metagenomics, Snakemake, Bioinformatics, Pipeline, MAGs, Microbiome, Metagenome-assembled genomes, Bioinformatics workflow

Background

Metagenomics, the analysis of whole genomes of microbial communities directly from environmental samples, has proved to be a revolutionary tool in microbiology. With applications in environmental, medical and biotechnology arenas, metagenomics has resulted in the discovery of many interesting species and even whole phyla that had remained uncharacterised because they are not easily manipulated in the lab or are unculturable [1, 2]. This has led to the elucidation of the real dynamics of more complex microbial communities [3].

Metagenomic sequencing of environmental samples has become increasingly popular in recent years, mainly due to the development of next-generation sequencing. Sequencing technologies have become higher throughput and lower in cost, which makes the

*Correspondence:
Annabel Cansdale
annabel.cansdale@york.ac.uk
[1] Centre of Excellence for Anaerobic Digestion, Department of Biology, University of York, Wentworth Way, Heslington, York YO10 5DD, UK

sequencing of entire microbial communities more feasible [4]. Due to the advantage metagenomic sequencing offers by removing the need for isolation and amplification of organisms, many mixed microbial communities once known as the "uncultured microbial majority" or "microbial dark matter" [5, 6] that were previously challenging to characterise [7] have been targeted by metagenomic sequencing. A result of metagenomics targeting little-characterised communities is that high-quality reference genomes do not exist for many microbes observed in metagenomic studies, either because they are being seen for the first time or are uncultured non-model organisms [7].

A typical metagenomics analysis pipeline would be as follows: raw reads from shotgun DNA sequencing of a microbial community would undergo quality control before being assembled using appropriate assembly software (including MetaSPAdes, MEGAHIT, IDBA-UD for short-read metagenomics and metaFlye, Canu for long-read metagenomics [8–12]), and then the metagenomic assembly would be binned using a variety of approaches to group the contigs associated with different organisms in the sequenced community into metagenome-assembled genomes (MAGs) [13].

Metagenomic-specific software employs many different assembly and binning strategies [13] as metagenomic studies have different challenges than single-organism genomic studies. A mixed community with organisms at different abundances makes both the assembly and binning of a metagenome challenging [8, 13, 14]. The risk of contamination of MAGs from closely related organisms is an additional challenge [13]. Therefore, it is important to have a method of determining overall metagenome and MAG quality.

The lack of reference genomes available for many organisms identified through metagenomics becomes an issue when comparing metagenome analysis methods and software due to a lack of ground truth. Benchmarking and quality assessment tools exist for metagenomic studies, such as AMBER and MetaQUAST [15, 16]; however, these require the organisms present in the dataset to be known and have appropriate reference genomes.

Determining MAG quality is important to indicate the quality of the initial analysis and highlight which MAGs are worthy of further investigation or deposition onto online databases. The Minimum Information about a Metagenome-Assembled Genome (MIMAG) [17] is a standard developed by the Genomics Standards Consortium (GSC) which outlines a framework for the classification of MAG quality (into either high-quality draft, medium-quality draft or low-quality draft) and recommends the reporting of specific metadata for each MAG. While this framework aids in the reproducibility of metagenomic studies, it has not yet received universal uptake.

Within the MIMAG standards, three criteria are used to determine overall MAG quality: genome completeness, contamination and assembly quality. When taxonomy is known and a reference genome is available for a MAG, these metrics are easier to determine. However, identifying appropriate references and the subsequent pairwise alignment of MAGs is often a manual and computationally intensive process and so is not an appropriate method for a large number of MAGs [13].

Due to the lack of a "ground truth" (i.e. a closely related reference strain) for many communities that are investigated using metagenomic sequencing and the computational power required to determine closely related organisms at scale, it is necessary to take a reference-free or de novo approach to determine the success of both metagenomic sequencing and binning [17]. One such approach determines the completeness and contamination of a genome (or in this case a metagenome-assembled genome) using marker genes, as exemplified by the popular software CheckM [18].

For many organisms identified through metagenomics, determining assembly quality is challenging as there is not a defined sequence to compare the MAG back to. For MAGs, determining assembly quality is suggested to be determined by the presence and completeness of encoded rRNA and tRNA genes within the metagenome bin [17].

Due to the abundance of metagenomic software available, community adoption of standards like MIMAG is important to increase reproducibility and reliability within and between datasets; however, currently, the MIMAG standards remain underutilised by many studies. Adopting these standards is an important aspect of the FAIR principles for scientific data, which emphasise findability, accessibility, interoperability and reusability [19]. The advances and increasing throughput of metagenomic sequencing have resulted in the generation of hundreds to thousands of MAGs per metagenome [14, 20]. Parsing the information required to determine the quality of these bins and isolating the higher-quality MAGs worthy of further analysis is a challenge.

Here, we introduce MAGqual (*M*etagenome-*A*ssembled *G*enome *Qual*ity), a pipeline implemented in Snakemake v7.30.1 [21], to automate MAG quality analysis at scale. MAGqual enables the user to pass in MAGs generated by metagenomic binning software and quickly assess the quality of these bins according to the MIMAG

standards. These bins are analysed to determine completeness and contamination (using CheckM v1.0.13 [18]) and the number of rRNA and tRNA genes (using Bakta v1.7.0 [22]) that each bin encodes. This information is used by bespoke code to determine the quality of each bin, in line with the MIMAG standards (with an additional "near-complete" category), and produces figures and a report that outlines the quality and other metrics of the input MAGs.

MAGqual enables users to automate the assignment of quality to their metagenome bins and quickly determine the success of their metagenomic analysis. This will hopefully improve the uptake of MIMAG standards across the metagenomics community and provide an easy way to benchmark new metagenomic binning software or analysis methods. MAGqual supports the FAIR principles by generating comparable metrics from metagenomic datasets collected by diverse methods and provides a visual measure of MIMAG and additional metagenomic statistics. Its open-access nature and simple Snakemake pipeline will enable timely updates as the metagenomic field moves forward. MAGqual is available from https://github.com/ac1513/MAGqual under an MIT license.

Methods

MAGqual pipeline (Fig. 1)

The MAGqual pipeline is built in Snakemake (v.7.30.1) [21]. Snakemake is a popular workflow management tool based on Python that enables a human-readable plug-and-play strategy for analysis pipeline design. This method of design results in a pipeline that is easier to understand, adapt and maintain. Furthermore, Snakemake integrates easily into high-performance computing clusters, making workflows highly scalable on many systems — which is key as datasets increase in size.

The MAGqual pipeline requires only the installation of Miniconda and Snakemake by the user. The installation of all remaining software is handled by the Snakemake pipeline using Conda environments. Additionally, MAGqual handles the installation of databases required by Bakta and CheckM. The light version of the Bakta database is downloaded to maximise speed and minimise storage space required; however, MAGqual allows the specification of a local Bakta and/or CheckM database if required. Two file types are required as input: first, a directory containing the metagenomic bins (MAGs) in FASTA format (with the file extension fasta, fna or fa) and second the

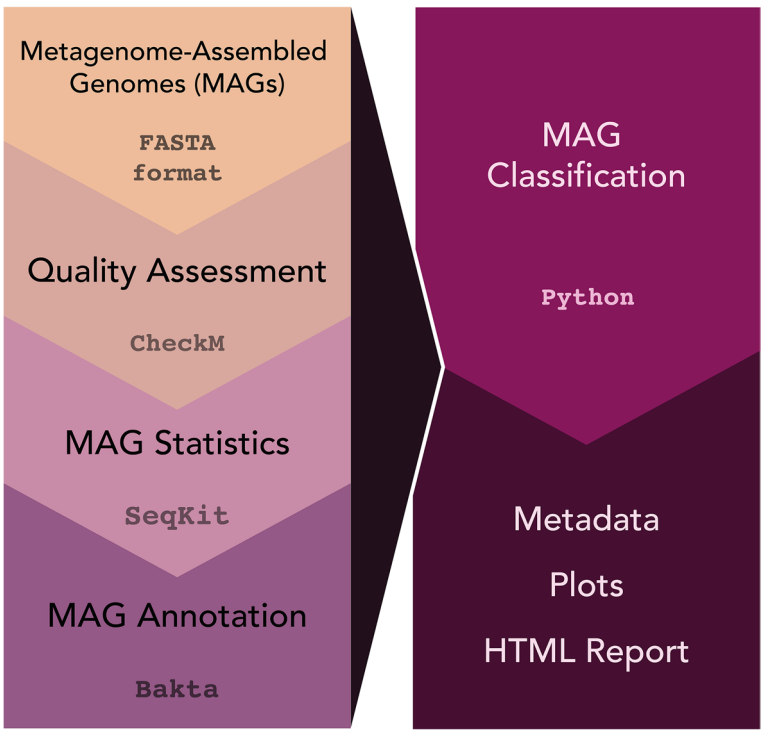

Fig. 1 The MAGqual pipeline. MAGs in FASTA format are run through CheckM for quality assessment, SeqKit to calculate basic statistics and Bakta for MAG annotation, and the output from these is classified according to the MIMAG standards using a bespoke Python script. This outputs a CSV file containing recommended metadata, the MAGs organised according to their classification and plots and a report are generated to visualise the classification and assembly statistics

Table 1 MAGqual command line options and their usage

Command line option	Usage
-a / --asm	*Required*: The location of the assembly in FASTA format used to generate the metagenome bins
-b / --bins	*Required*: The location of the directory containing all the metagenome bins for quality assignation
-p / --prefix	The prefix for the job. Default is MAGqual_YYYYMMDD
-j / --jobs	Links to the Snakemake flag -j, the number of cores to use or if using the cluster option and the number of jobs to run at once. Default is 1
--cluster	Optional: The type of cluster to run MAGqual on a HPC system (available options: slurm), not to be used if running MAGqual locally
--checkmdb	Optional: The location of a local install of the CheckM database
--baktadb	Optional: The location of a local install of the Bakta database. Note: Must be v.5 or above
-h / --help	Show help message

metagenomic assembly (in FASTA format) used to generate the metagenomic bins.

To remain accessible to those unfamiliar with Snakemake pipelines, MAGqual can be run using a Python wrapper with the basic command:

```
python MAGqual.py --asm assembly.fa --bins bins_dir/
```

The wrapper achieves full pipeline functionality without requiring users to edit configuration files. See Table 1 for the full command line options available to the user and their defaults.

MAGqual also retains full Snakemake functionality and can easily be run using the Snakemake architecture. The basic command for this is snakemake --use-conda -j 1 and the user is required to edit the config/config.yaml file to specify the location of the input files and has the option to further edit the command and the config/cluster.json file to add configuration options to run MAGqual on an HPC cluster. This is a useful option for those more familiar with Snakemake pipelines, as the pipeline can be further modified to run better on different infrastructures; however, this is not necessarily appropriate for every user.

MAG completeness and contamination

While MIMAG introduces metagenomic standards, it does not recommend a specific "industry-standard" method for calculating them. Both the completeness and contamination values vary depending on what set of initial single-copy marker genes is used, so the method used requires reporting. Here, we use the CheckM, which has become the de facto software used for these calculations in the years since the standards were published [23].

Table 2 MAGqual quality evaluation categories and the completeness, contamination and rRNA/tRNA requirements

Quality	Completeness	Contamination	rRNA/tRNA required
High	> 90%	≤ 5%	≥ 18 tRNA and 23S, 16S and 5S rRNA genes
Near complete	> 90%	≤ 5%	None
Medium	≥ 50%	≤ 10%	None
Low	< 50%	≤ 10%	None
Failed	-	≥ 10%	None

Assembly quality

In line with the MIMAG standards, the presence and completeness of tRNA and rRNA ribosomal genes are also determined. This is suggested as a method of assembly quality determination. Here, the MAG is run through Bakta for annotation and rapidly identifies rRNA and tRNA ribosomal genes. Previously, Prokka was the software of choice for fast microbial annotation [24]; however, Bakta improves the annotation of CDS compared to Prokka and remains actively supported so it was chosen for use in this pipeline [22]. To be classified as a high-quality draft MAG, a MAG must encode tRNAs for at least 18 of the 20 possible amino acids and the 5S, 16S and 23S rRNA genes [17].

Determining MAG quality

Using the results from CheckM and Bakta, MAGqual calculates overall MAG quality using Python (Table 2).

Along with the high, medium and low-quality standards introduced in the MIMAG standards [17], we include the "near-complete" quality draft MAG category introduced by Almeida et al. (2019) [25]. We define this "near-complete" quality (NCQ) draft MAG as > 90% complete, < 5% contaminated, but not encoding the necessary tRNA and rRNA to be classified as a highly complete draft MAG. This was determined to be an important addition to MAGqual due to documented problems around the assembly and annotation of rRNA/tRNA sequences [23], especially with metagenomes generated from short-read sequencing [20, 26]. As uncultured and previously undetermined organisms comprise a significant proportion of metagenomic communities, this enables some flexibility for any CDS annotation issues.

Table 3 Metadata and metrics reported by the MAGqual pipeline in CSV format

Metric	Reported
Assembly quality	High-quality draft, near complete, medium-quality draft, low-quality draft, failed
Completeness score	Percentage (%)
Contamination score	Percentage (%)
Completeness software	CheckM (version)
16S rRNA genes recovered	Yes/no
16S rRNA software	Bakta (version & database version)
tRNA extracted	No./20
tRNA software	Bakta (version & database version)
Completeness approach	Marker gene
Assembly statistics	-
Size	No. of bp
N50	No. of bp
Maximum contig length	No. of bp
Number of contigs	No

Once the MAGs have been assessed, a figure showing a breakdown of size, completeness and contamination scores and quality category for each bin is generated. This provides users with simple and quick evidence of the quality of their metagenome bins. A file containing the recommended metadata (see Table 3) is exported for each MAG to enable easy submission and analysis. An interactive HTML report is also generated to allow easy viewing of these plots and metadata. MAGqual will finally output multiple directories containing the MAGs split by overall quality category.

Reported metadata

Alongside determining MAG quality, MAGqual also generates metadata recommended by the MIMAG standards. The metadata categories can be seen in Table 3; MAGqual produces a CSV file with a line of corresponding metadata for each MAG run through the pipeline.

MAGqual report

Along with a metadata table, MAGqual generates an interactive HTML report generated using Python, RMarkdown and Plotly. This report produces numerous figures including the bases and contigs binned, completeness and contamination, N50 length, total length and tRNA completeness along with tables for MAG quality and metadata. This report is generated from all available MAGqual runs in the same directory, enabling a quick comparison of binning results between MAGqual runs.

Running the pipeline

To determine the runtime of the pipeline on minimal architecture, an 8-core 32 GB Linux instance (Canonical Ubuntu 22.04) on Oracle Cloud was used. Snakemake (v7.30.1) and Conda (v.23.5.2) were installed into this nascent environment, and MAGqual handled the installation and database downloads required.

Due to the requirement for over 40 GB of memory, CheckM was run for this test with the --reduced_tree option which lowers the memory required to 16 GB.

To reflect actual runtime, each benchmark was run from a clean environment; therefore, the Conda environments had to be remade and the databases re-downloaded to include the time required for these into each run.

To validate the MAGqual pipeline, we generated a dataset of 10, 100, 500 and 1000 MAGs from Parks et al. (2017) [20]. This study was chosen as these MAGs were previously assigned a quality with completeness and contamination scores. However, this paper predates the publication of the MIMAG standards so no high-quality MAGs were defined.

Comparison of binning tools

A small metagenomics dataset from a gut microbiome was procured from ENA project PRJEB44880 [27] corresponding to a nanopore metagenome assembly polished with Illumina short reads and seven samples of short-read Illumina raw sequencing data. Three popular metagenomic binning tools were chosen for comparison: CONCOCT (v1.1.0), MetaBAT2 (v2.12.1) and BinSanity (v0.5.4) [28–30]. All three tools use abundance information for binning, obtained by mapping Illumina short reads back to the assembly using BWA (v0.7.17) [31] to produce a BAM file for each sample.

The BAM files were then passed to each binning software using the minimum contig length of 1000 bp — apart from CONCOCT, which requires contigs < 10 kb, where the assembly had to be split and the raw reads remapped to each split assembly. The MAGs produced by these three binners were passed through the bin refinement tools MetaWrap (v1.3.2) and DAS Tool (v1.1.6) [32, 33]. DAS Tool was run as directed, with the flag --write_bins to produce the bins for comparison. MetaWrap was run using the bin_refinement module and the flags -c 0 (minimum completeness)—×100 (maximum contamination) to output all bins regardless of quality.

Results

Running the pipeline

MAGqual is quick and easy to run, with only one command required to initiate the pipeline, install dependencies and run each of the steps. The speed of the pipeline

Table 4 Time and memory requirements for the MAGqual pipeline when run with an increasing number of MAGs (10, 100, 500 and 1000) on an 8-core 32GB RAM instance. The real time, CPU time and maximum memory of the overall pipeline and also the individual steps of Bakta. CheckM and Python scripts are outlined below. As Bakta runs a separate job for each MAG, the average real time, average CPU time and average max memory are also included

		MAGqual overall		Bakta				CheckM		Scripts	
		Wall-clock time (HH:MM:SS)	CPU time (HH:MM:SS)	Total wall-clock time (HH:MM:SS)	Total CPU time (HH:MM:SS)	Average wall-clock time (HH:MM:SS)	Average CPU time (HH:MM:SS)	Wall time (HH:MM:SS)	CPU time (HH:MM:SS)	Wall time (HH:MM:SS)	CPU time (HH:MM:SS)
No. of MAGs	**10**	00:40:21	01:46:38	00:26:05	01:27:44	00:02:37	00:08:46	00:03:55	00:15:00	00:00:10	00:00:12
	100	04:41:35	16:05:00	04:13:00	16:40:26	00:02:32	00:10:00	00:25:28	02:51:03	00:00:09	00:00:09
	500	22:59:02	84:18:59	20:58:15	77:18:18	00:02:31	00:09:17	03:31:57	11:06:23	00:00:12	00:00:10
	1000	45:48:35	164:08:03	41:56:34	95:31:20	00:02:31	00:11:44	04:27:48	29:30:18	00:00:15	00:00:11

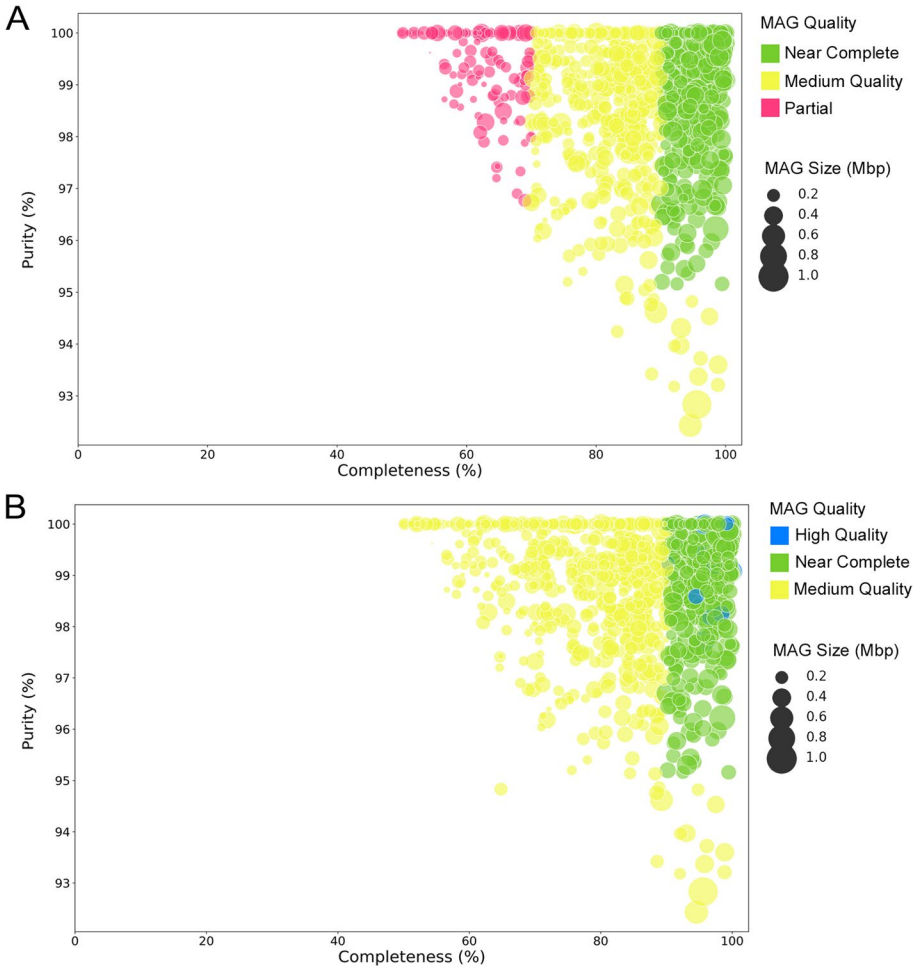

Fig. 2 **A** The completeness and purity (100-contamination score) of the 1000 MAGs from the metagenomic benchmarking dataset when using the completeness, contamination and quality metrics from the original paper. **B** The completeness and purity (100-contamination score) of the 1000 MAGs from the metagenomic benchmarking dataset using the MIMAG standards and the introduced near-complete category

depends on the size of the dataset being analysed and is overall limited by the speed of the programmes Bakta and CheckM.

As seen in Table 4, with the generated dataset of 10, 100, 500 and 1000 MAGs from [20], as the number of MAGs increases, the majority of runtime is assigned to Bakta, which, while only taking on average ~ 2.5 min per MAG, becomes significant when running 1000 MAGs.

While it was important to determine the minimal computational infrastructure required to run this pipeline, as the majority of metagenomic research is undertaken on HPCs, it is recommended to run Bakta jobs in parallel which is possible using the flag -j using both the Python wrapper and using the Snakemake infrastructure.

Table 5 The total number of MAGs and their respective qualities for each of the five binning tools evaluated with this dataset

	BinSanity	CONCOCT	MetaBAT2	DAS Tool	MetaWRAP
Total number of MAGs	**109**	**91**	**92**	**25**	**126**
High quality	8	8	5	10	9
Near complete	1	2	1	2	2
Medium quality	11	12	18	11	17
Low quality	83	61	66	2	98
Failed	6	8	2	0	0

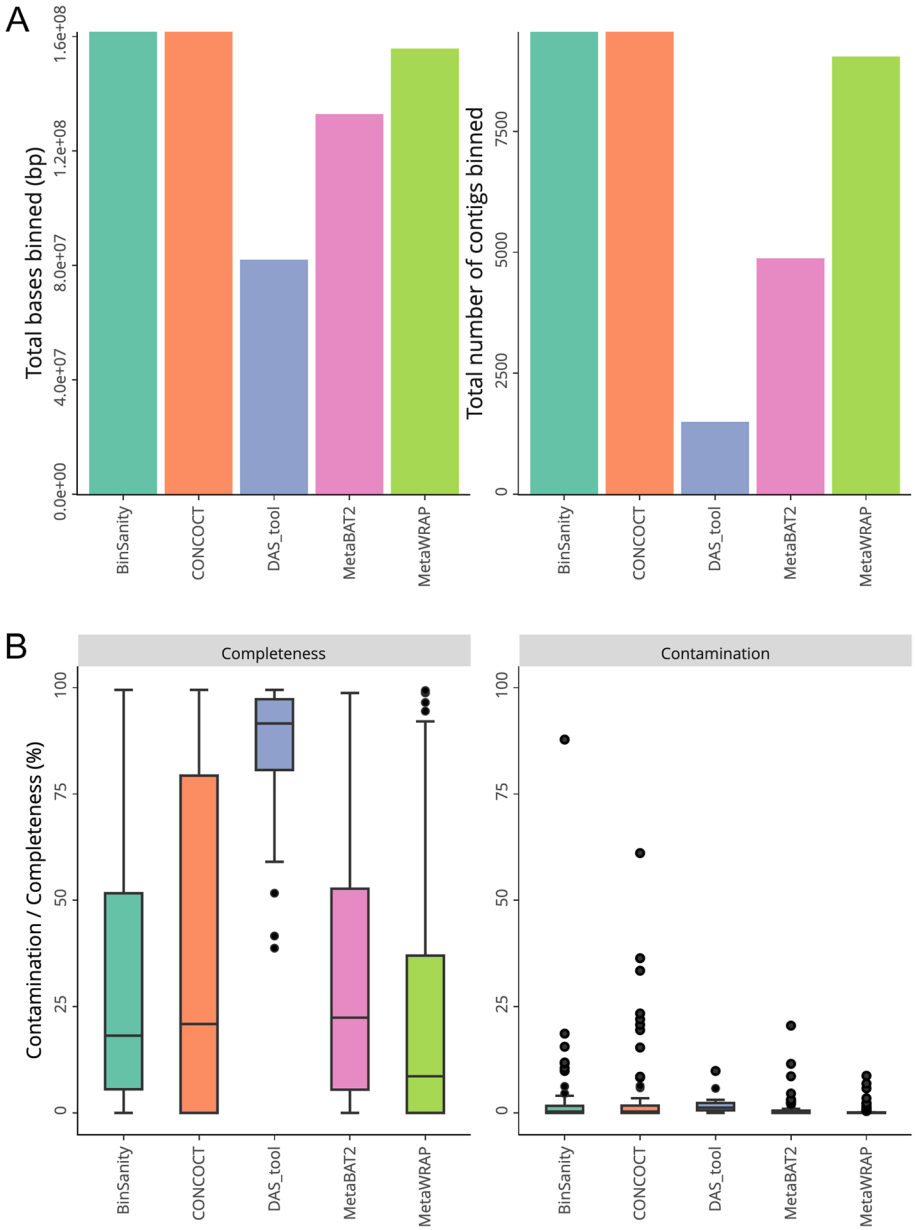

Fig. 3 **A** The total number of bases (bp) and contigs binned for each of the five binning tools examined with this dataset. **B** Boxplot showing the distribution of the completeness and contamination scores for each MAG generated by the five binning tools evaluated with this dataset

Validation of MAGqual

The 1000 MAGs used in the previous analysis were also used to compare the quality score assigned by Parks et al. (2017) [20] (Fig. 2A) to the new MAGqual quality score (Fig. 2B). The introduction of MIMAG quality scores changed the assignment of many MAGs from the original dataset, classifying 17 high-quality MAGs that were previously within the "near-complete" category.

Analysis of this example dataset highlights the importance of the near-complete category in the MAGqual pipeline. All 417 would have been assigned as medium-quality MAGs, and only 17 would be classed as high quality. In contrast, these MAGs are still > 90% complete and < 5% contaminated and potentially would benefit from further analysis. Including this category in the MAGqual pipeline will hopefully increase the uptake of metagenomic dataset benchmarking in studies. A small

number of MAGs have different completeness and purity scores; however, the majority remain the same.

To demonstrate the speed of MAGqual on an HPC, where most metagenomic research is undertaken, these 1000 MAGs were analysed using the MAGqual pipeline on a 64-core, 512 GB machine. MAGqual completed with a wall-clock time of 2:55:57 and a CPU time of 243:34:04.

Comparison of binning tools

To demonstrate the use of MAGqual as a bin comparison tool, a simple gut microbiome metagenome [27] was re-binned using three different metagenomic binning tools, CONCOCT, MetaBAT2 and BinSanity [28–30], and then refined using the pipeline from MetaWRAP and DAS Tool [32, 33]. MAGqual was used to analyse the bins generated using these five different tools. Table 5 shows that CONCOCT and MetaBAT2 both generated a similar number of bins (91 and 92, respectively); however, CONCOCT generated more high-quality bins (8) than MetaBAT2, as did BinSanity. DAS Tool and MetaWRAP both improved the overall quality of bins, indicating the benefits of a combined binning approach. However, DAS Tool produced a much lower number of bins overall, and DAS Tool binned substantially fewer contigs and bases overall (Fig. 3A) but produced higher-quality MAGs (more complete with low contamination, Fig. 3B), illustrating a potential trade-off between assigning more of the sequence data and improved bin quality that likely depends on different algorithmic approaches to binning philosophies. MAGqual enabled a rapid comparison of the MAGs created using these five methods so that users could select the most appropriate binning strategy for their research. Further plots from this analysis can be seen in the Supplementary HTML file.

Conclusions

As the size of metagenomic datasets continues to grow, researchers face new challenges in managing and analysing these data. Larger datasets require more sophisticated computational infrastructure, often involving high-performance computing clusters or cloud resources. Metagenome analysis is characterised by a wide array of methods and tools, each with its strengths and limitations, and researchers often choose different tools based on their specific research questions or the nature of their data. This diversity in tools can lead to variations in analysis outcomes. The adoption of MAGqual, a lightweight and user-friendly pipeline, offers a valuable solution to researchers, as it simplifies the binning evaluation process and can be quickly and efficiently applied to datasets of varying sizes generated by any analysis tool.

Building the MAGqual pipeline in Snakemake provides further advantages, including cluster execution, modularisation and simple pipeline updates. After testing and benchmarking using the Snakemake benchmarking function, new tools can be integrated into the MAGqual pipeline with ease, and as the pipeline is hosted on GitHub, any updates will be shared with the community promptly. For example, during the preparation of this manuscript, CheckM2 [34] was released and could be easily substituted into the pipeline.

One of the pipeline's primary objectives is to encourage wider adoption of the MIMAG (Minimum Information about a Metagenome-Assembled Genome) reporting standards to ultimately improve the consistency and quality of metagenomic research. MAGqual aids users in swiftly identifying data that merits further analysis. By identifying good quality MAGs, MAGqual provides a targeted approach to metagenome analysis, which can reduce both computational and storage costs, making metagenome analysis more accessible and cost-effective.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40168-024-01949-z.

Additional file 1: Supplementary figures: Fig. 1: Total number of bases and contigs binned for CONCOCT, DAS_tool, BinSanity, MetaWRAP, MetaBAT2. Figure 2: Distribution of the completeness and contamination of CONCOCT, DAS_tool, BinSanity, MetaWRAP, MetaBAT2. Figure 3: Distribution of the the N50 lengths (bp) of the bins generated by CONCOCT, DAS_tool, BinSanity, MetaWRAP, MetaBAT2. Supplementary table: Table 1: Overall MAG quality of CONCOCT, DAS_tool, BinSanity, MetaWRAP, MetaBAT2 and total number of MAGs generated by each method

Additional file 2.

Acknowledgements

We are grateful for computational support from the University of York High-Performance Computing service, Viking and the Research Computing team. We thank Sarah Forrester and Joseph McGrory for their critical reading of the manuscript.

Authors' contributions

AC conceived and designed the work, created the software used and drafted the work; JPJC contributed to the interpretation of data and revised the manuscript. All authors read and approved the final manuscript.

Funding

J. P. J. C. is an Oracle for Research Fellow. The Centre of Excellence for Anaerobic Digestion is supported in part by BBSRC grant BB/Y003314/1.

Data availability

All data generated or analysed during this study are included in this published article and its Supplementary information files. The MAGqual pipeline is available from https://github.com/ac1513/MAGqual under an MIT license. A snapshot of the MAGqual code was taken at time of publication and is available on Zenodo DOI: https://doi.org/10.5281/zenodo.13384336.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 19 December 2023 Accepted: 13 October 2024
Published online: 04 November 2024

References

1. Pelletier E, Kreimeyer A, Bocs S, Rouy Z, Gyapay G, Chouari R, et al. "Candidatus Cloacamonas acidaminovorans": genome sequence reconstruction provides a first glimpse of a new bacterial division. J Bacteriol. 2008;190:2572–9.
2. Zaremba-Niedzwiedzka K, Caceres EF, Saw JH, Bäckström D, Juzokaite L, Vancaester E, et al. Asgard archaea illuminate the origin of eukaryotic cellular complexity. Nature. 2017;541:353–8.
3. Van Goethem MW, Osborn AR, Bowen BP, Andeer PF, Swenson TL, Clum A, et al. Long-read metagenomics of soil communities reveals phylum-specific secondary metabolite dynamics. Commun Biol. 2021;4:1302.
4. Albertsen M. Long-read metagenomics paves the way toward a complete microbial tree of life. Nat Methods. 2023;20:30–1.
5. Rappé MS, Giovannoni SJ. The uncultured microbial majority. Annu Rev Microbiol. 2003;57:369–94.
6. Filée J, Tétart F, Suttle CA, Krisch HM. Marine T4-type bacteriophages, a ubiquitous component of the dark matter of the biosphere. Proc Natl Acad Sci U S A. 2005;102:12471–6.
7. Nayfach S, Roux S, Seshadri R, Udwary D, Varghese N, Schulz F, et al. A genomic catalog of Earth's microbiomes. Nat Biotechnol. 2021;39:499–509.
8. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. metaSPAdes: a new versatile metagenomic assembler. Genome Res. 2017;27:824–34.
9. Li D, Liu C-M, Luo R, Sadakane K, Lam T-W. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. 2015;31:1674–6.
10. Peng Y, Leung HCM, Yiu SM, Chin FYL. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics. 2012;28:1420–8.
11. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res. 2017;27:722–36.
12. Kolmogorov M, Bickhart DM, Behsaz B, Gurevich A, Rayko M, Shin SB, et al. metaFlye: scalable long-read metagenome assembly using repeat graphs. Nat Methods. 2020;17:1103–10.
13. Pérez-Cobas AE, Gomez-Valero L, Buchrieser C. Metagenomic approaches in microbial ecology: an update on whole-genome and marker gene sequencing analyses. Microb Genom. 2020;6:6.
14. Pasolli E, Asnicar F, Manara S, Zolfo M, Karcher N, Armanini F, et al. Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle. Cell. 2019;176:649–62.e20.
15. Meyer F, Hofmann P, Belmann P, Garrido-Oter R, Fritz A, Sczyrba A, et al. AMBER: Assessment of Metagenome BinnERs. Gigascience. 2018;7:7.
16. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics. 2016;32:1088–90.
17. Bowers RM, Kyrpides NC, Stepanauskas R, Harmon-Smith M, Doud D, Reddy TBK, et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol. 2017;35:725–31.
18. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015;25:1043–55.
19. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR guiding principles for scientific data management and stewardship. Sci Data. 2016;3:160018.
20. Parks DH, Rinke C, Chuvochina M, Chaumeil P-A, Woodcroft BJ, Evans PN, et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nat Microbiol. 2017;2:1533–42.
21. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with Snakemake. F1000Res. 2021;10:33.
22. Schwengers O, Jelonek L, Dieckmann MA, Beyvers S, Blom J, Goesmann A. Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. Microb Genom. 2021;7:7.
23. Yang C, Chowdhury D, Zhang Z, Cheung WK, Lu A, Bian Z, et al. A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data. Comput Struct Biotechnol J. 2021;19:6301–14.
24. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics. 2014;30:2068–9.
25. Almeida A, Mitchell AL, Boland M, Forster SC, Gloor GB, Tarkowska A, et al. A new genomic blueprint of the human gut microbiota. Nature. 2019;568:499–504.
26. Singleton CM, Petriglieri F, Kristensen JM, Kirkegaard RH, Michaelsen TY, Andersen MH, et al. Connecting structure to function with the recovery of over 1000 high-quality metagenome-assembled genomes from activated sludge using long-read sequencing. Nat Commun. 2021;12:2009.
27. Shahi F, Forrester S, Redeker K, Chong JPJ, Barlow G. Case report: the effect of intravenous and oral antibiotics on the gut microbiome and breath volatile organic compounds over one year. Wellcome Open Res. 2022;7:50.
28. Alneberg J, Bjarnason BS, de Bruijn I, Schirmer M, Quick J, Ijaz UZ, et al. Binning metagenomic contigs by coverage and composition. Nat Methods. 2014;11:1144–6.
29. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, et al. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ. 2019;7:e7359.
30. Graham ED, Heidelberg JF, Tully BJ. BinSanity: unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation. PeerJ. 2017;5:e3035.
31. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv [q-bio.GN]. 2013.
32. Uritskiy GV, DiRuggiero J, Taylor J. MetaWRAP-a flexible pipeline for genome-resolved metagenomic data analysis. Microbiome. 2018;6:158.
33. Sieber CMK, Probst AJ, Sharrar A, Thomas BC, Hess M, Tringe SG, et al. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. Nat Microbiol. 2018;3:836–43.
34. Chklovski A, Parks DH, Woodcroft BJ, Tyson GW. CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. Nat Methods. 2023;20:1203–12.

Publisher's Note