



## SOFTWARE TOOL ARTICLE

# Metagenome quality metrics and taxonomical annotation visualization through the integration of MAGFlow and BIgMAG [version 1; peer review: awaiting peer review]

Jeferyd Yepes-García <sup>1,2</sup>, Laurent Falquet<sup>1,2</sup>

<sup>1</sup>Department of Biology, University of Fribourg, Fribourg, Canton of Fribourg, 1700, Switzerland

<sup>2</sup>Swiss Institute of Bioinformatics, Lausanne, Vaud, 1015, Switzerland

**V1** First published: 17 Jun 2024, 13:640  
<https://doi.org/10.12688/f1000research.152290.1>

Latest published: 17 Jun 2024, 13:640  
<https://doi.org/10.12688/f1000research.152290.1>

## Open Peer Review

**Approval Status** AWAITING PEER REVIEW

Any reports and responses or comments on the article can be found at the end of the article.

## Abstract

### Background

Building Metagenome-Assembled Genomes (MAGs) from highly complex metagenomics datasets encompasses a series of steps covering from cleaning the sequences, assembling them to finally group them into bins. Along the process, multiple tools aimed to assess the quality and integrity of each MAG are implemented. Nonetheless, even when incorporated within end-to-end pipelines, the outputs of these pieces of software must be visualized and analyzed manually lacking integration in a complete framework.

### Methods

We developed a Nextflow pipeline (MAGFlow) for estimating the quality of MAGs through a wide variety of approaches (BUSCO, CheckM2, GUNC and QUASt), as well as for annotating taxonomically the metagenomes using GTDB-Tk2. MAGFlow is coupled to a Python-Dash application (BIgMAG) that displays the concatenated outcomes from the tools included by MAGFlow, highlighting the most important metrics in a single interactive environment along with a comparison/clustering of the input data.

### Results

By using MAGFlow/BIgMAG, the user will be able to benchmark the MAGs obtained through different workflows or establish the quality of the MAGs belonging to different samples following *the divide and rule*

methodology.

## Conclusions

MAGFlow/BIgMAG represents a unique tool that integrates state-of-the-art tools to study different quality metrics and extract visually as much information as possible from a wide range of genome features.

## Keywords

Metagenomics, Nextflow, pipeline, dashboard, data analysis.



This article is included in the [Bioinformatics gateway](#).

**Corresponding author:** Laurent Falquet ([laurent.falquet@unifr.ch](mailto:laurent.falquet@unifr.ch))

**Author roles:** **Yepes-García J:** Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Falquet L:** Conceptualization, Funding Acquisition, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

**Competing interests:** No competing interests were disclosed.

**Grant information:** MAGFlow/BIgMAG is being developed under the support of the Federal Commission for Scholarships for Foreign Students (FCS) through the program Swiss Government Excellence Scholarships.

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

**Copyright:** © 2024 Yepes-García J and Falquet L. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**How to cite this article:** Yepes-García J and Falquet L. **Metagenome quality metrics and taxonomical annotation visualization through the integration of MAGFlow and BIgMAG [version 1; peer review: awaiting peer review]** F1000Research 2024, **13**:640 <https://doi.org/10.12688/f1000research.152290.1>

**First published:** 17 Jun 2024, **13**:640 <https://doi.org/10.12688/f1000research.152290.1>

## Introduction

The generation of metagenomics data has increased exponentially within the last ten years, supported by the rapid evolution of next generation sequencing techniques for both short and long reads (Tremblay et al., 2022). Moreover, to perform the analysis of microbiome data the usual approach involves reconstructing genomes, commonly known as Metagenome-Assembled Genomes (MAGs), from fragmented sequences obtained during DNA extraction. The steps to achieve this goal enclose the cleaning and filtering out of low-quality reads, the assembly of the short sequences into longer and contiguous strands (contigs), plus clustering contigs into bins according to multiple genome-level features such as tetranucleotide frequency, similarity in coverage, GC content, among others (Yang et al., 2021).

Routinely, the recovered MAGs, independently of the selected workflow to assemble them, should be subject to quality measurements by one or several tools. Contiguity, completeness, and contamination are regular metrics used to classify the MAGs in different categories based on arbitrary criteria (Haryono et al., 2022). Common thresholds used to separate low (or simply bins), mid or high-quality MAGs are completeness and contamination values generated by tools such as CheckM or CheckM2, *e.g.*, high-quality MAGs depict levels of contamination below 5% and completeness above 90%. For this manuscript we will use MAGs as a reference to any MAG regardless the category they should be classified in.

Furthermore, given the complexity and abundance of species within some specific environments, *i.e.*, soil or sea sources (Guseva et al., 2022), more sophisticated tools have been developed to detect the presence of specific marker genes, chimerism and duplication. In addition, proper taxonomical annotation of the MAGs allows to report unique insights related to the composition of the community, contributing to ensure the quality of the assembly and the accuracy of the methods employed to bin the contigs (Cornet & Baurain, 2022).

Nevertheless, many pipelines or methodologies only include one or two tools to measure the quality of the MAGs (Kieser et al., 2020; Krakau et al., 2022; Uritskiy et al., 2018), and the visualization and/or analysis of the quality data relies entirely on the user who should be familiar with the type of generated files and how to display the information in a pleasant manner. In other cases, the users must develop manually their own methodology to perform this important step during the metagenome assembly, increasing the risk of lack of reproducibility and the difficulty to test and validate the results. This scenario sets interesting challenges that encases maximizing the scope of the quality assessment, the need of pipelines coupled to visualization tools and boosting reproducibility by wrapping the process with workflow managers (Wratten et al., 2021).

As a result, we designed a framework to measure the quality of the MAGs generated by different methodologies or belonging to different samples, as well as to visualize the metrics through a web-based interface. This framework carries out the analysis through a Nextflow pipeline (MAGFlow) that takes the MAGs as input to measure their completeness and contamination using machine learning through CheckM2 (Chklovski et al., 2023), determine the level of single-copy ortholog (SCO) completeness, duplication and fragmentation using BUSCO (Manni et al., 2021), estimate chimerism and contamination with GUNC (Orakov et al., 2021), perform a full taxonomical classification by GTDB-Tk2 (Chaumeil et al., 2022), and produce a full report of the assembly features by QAST (Gurevich et al., 2013). After merging the outcomes from these tools, a final *tsv* file is compiled by MAGFlow, and the user can use it to render an interactive web-based Dash application (BIgMAG).

## Methods

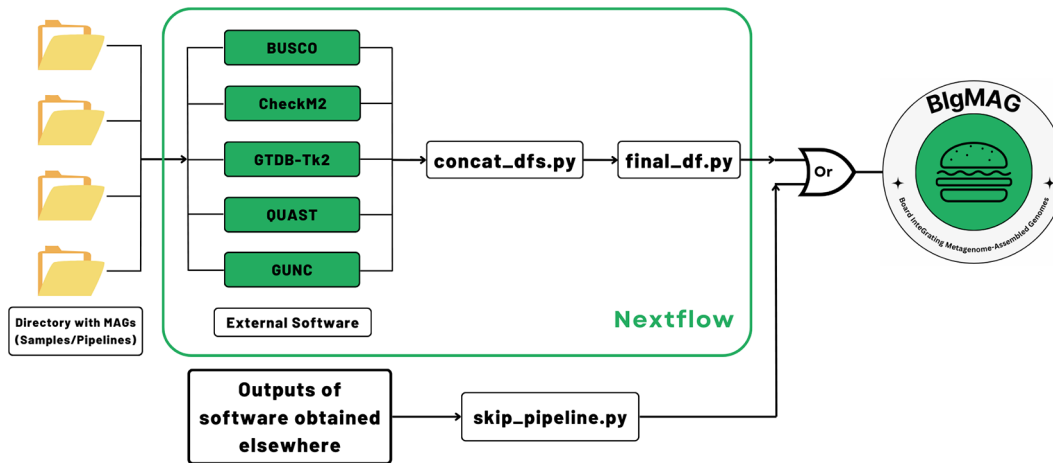
### Implementation

#### **MAGFlow**

MAGFlow is a pipeline designed to run in any environment, to be portable, easy to install, scalable to the available computational resources and ready to use in local or cloud-based infrastructures. In addition, MAGFlow can be customized through tuning parameters that are available for each tool encompassed in the workflow.

In order to assess the MAG quality and/or perform the taxonomical annotation, MAGFlow, wrapped by Nextflow (v23.04.0), runs BUSCO (v5.7.0), CheckM2 (v1.0.1), GUNC (v1.0.6), QAST (v5.2.0) and optionally GTDB-Tk2 (v2.3.2), along with a download of the latest Genome Taxonomy Database (GTDB, current release: 220), to produce a *tsv* file that concatenates the outputs from these pieces of software (Figure 1). The external tools enclosed by MAGFlow are Open Access, and they will remain in such status according to the license provided by their developers.

As input, the pipeline uses genomic files organized in corresponding folders, decompressed (*.fna*, *.fa*, *.fasta*) or compressed (*.gz*) obtained from a previous process of assembly/binning. Depending on the type of input, the workflow begins by decompressing the files and checking if there are empty files to be removed from the analysis; the original files



**Figure 1.** MAGFlow workflow to measure the metagenome quality by different tools, annotate the MAGs taxonomically and render an interactive dashboard using the output from each piece of software.

will not be modified. Afterwards, the tools mentioned above will be run in parallel according to the allocated resources for each job. The taxonomical annotation is optional for each user given the high demand it represents in terms of memory and storage. Once all the processes are finished, MAGFlow will merge the outcomes from each tool, to end up with a unique *final\_df.tsv* file, which is the main input for BIGMAG. Additionally, the pipeline will produce the regular output from each of the tools and HTML execution reports displaying the resource usage, success of each step and timestamps, allowing the user to explore them if required.

In case the user accounts with previously generated BUSCO, CheckM2, GTDB-Tk2, GUNC or QUAST files and their aim is to use them to explore the metrics using BIGMAG, it is possible to achieve this task by just running an additional script (*skip\_pipeline.py*). This script can recognize and process the files into the required *final\_df.tsv* by BIGMAG. Directions about how to run MAGFlow or use the *skip\_pipeline.py* script are available at <https://github.com/jeffe107/MAGFlow>.

### **BIGMAG**

BIGMAG consists of an interactive Dash application with Plotly as the rendering engine on a regular modern web browser. The layout is divided into 7 sections, 5 of them being plots dedicated exclusively to the tools executed by MAGFlow. In the case of CheckM2 and BUSCO, scatterplots are used to depict the behavior of genome completeness against contamination levels. Moreover, the results from QUAST and GUNC are displayed through boxplots that show the distribution of the data according to a user-selected parameter.

To process the outcomes from GTDB-Tk2, BIGMAG generates a presence/absence matrix representation of annotated taxa at the taxonomical level specified by the user, in a novel strategy to extract the core data without showing the complete phylogeny of the annotated MAGs. This approach is valuable in terms of the simplicity it achieves to display complex information, providing an alternative to the time-consuming branch-collapsing methodology that is usually followed through specialized software packages, *e.g.*, iTOL (Letunic & Bork, 2021), as shown by Bayer et al. (2020) to condensate the GTDB-Tk2 output.

Following the GTDB-Tk2 plot, as an attempt to compare the features among samples and cluster them according to their similarity using different parameters provided by BUSCO, CheckM2, GUNC and/or GTDB-Tk2, a cluster heatmap is displayed. To render this plot, average values are used, combined with the proportion of MAGs passing the GUNC test and/or the proportion of annotated MAGs by GTDB-Tk2.

At the bottom of the dashboard, the raw data in a format table is displayed with the possibility for the user of highlighting cells, deleting rows and filtering the columns with keywords.

Complementary to each plot, there is a series of callbacks that exploits the native interactivity of Plotly and Dash. These callbacks allow filtering out the MAGs according to contamination and completeness threshold levels in the CheckM2

plot, or based on complete SCO and duplicated SCO in the case of the BUSCO figure, selecting the parameter to display the data distribution of each sample in GUNC or QUASt plots and zooming in/out in the GTDB-Tk2 plot by selecting the number of samples to show. Also, it is possible to store all the figures individually directly from the dashboard with the in-built function to download the plots locally. More interactive details will be displayed when the cursor hovers each of the plot zones, i.e., on CheckM2 or BUSCO scatterplots, if GTDB-Tk2 data is found in the *final\_df.tsv* file, the taxonomical classification will be also included in the information depicted.

BIgMAG also features the possibility of being compiled as an HTML webpage to be displayed afterwards without requiring any of the processing components installed. However, given that this is not a native feature of Dash, the callbacks cannot be used while the dashboard is stored. Therefore, we developed an additional script (*app\_lite.py*), a lite version of BImAG, displaying the same layout as the regular version, although with an additional *Save to html* button below the heading section, and without the components triggering the callbacks. This lite version of BImAG can be customized and adjusted by using command line arguments (tutorial available at the repository <https://github.com/jeffe107/BIgMAG>).

### Operation

MAGFlow/BIgMAG is performed on UNIX-based operating systems, e.g., any Linux distribution or macOS; its operation on Windows 10 or 11 is also possible through Windows Subsystem for Linux (WSL2). Specifically, MAGFlow demands the installation of Nextflow ( $\geq 23.04.0$ ), which in turn requires Java JDK 17 or 19 (recommended version 17.0.3). In addition, it is necessary to account with a software management tool such as Conda ( $\geq 23.3.1$ ), Mamba ( $\geq 1.3.1$ ) or any container technology including Docker, Singularity, Apptainer, Podman or Charliecloud. After fulfilling these requirements, the user can start the analysis with MAGFlow by providing the directory path where the MAG files are stored or a .csv file indicating the different file paths. A terminal example command can be as follows:

```
nextflow run MAGFlow/main.nf -profile apptainer --files '~/samples/*' --outdir.
```

Or:

```
nextflow run MAGFlow/main.nf -profile apptainer --csv_file '~/samples.csv' --outdir.
```

The `-profile` argument must be set according to the software management tool the system user incorporates, and the path to store the output files is completely arbitrary to the user. Details in regard to the required MAG file structure, an example of the .csv datasheet, and other configuration options can be found at <https://github.com/jeffe107/MAGFlow>.

On the other hand, BImAG only requires a previous installation of Conda ( $\geq 23.3.1$ ), Mamba ( $\geq 1.3.1$ ) or pip, plus the availability of any modern browser such as Chrome ( $\geq v124.0.6367.62$ ) or Firefox ( $\geq v124.0.2$ ). Once the user satisfies these requisites, they can install the components and dependencies through the commands:

```
pip install -r BImAG/requirements.txt
```

Or:

```
conda create -n BImAG --file BImAG/requirements.txt
conda activate BImAG
```

As a result, the user can access the interactive dashboard by providing the path to the *final\_df.tsv* file generated by MAGFlow through the following terminal command:

```
BImAG/app.py -p 8050 './final_df.tsv'
```

The argument `-p` is included to display BImAG on the port of the user preference. The default value is 8050. Once the command is run, the prompt output will indicate the IP direction the user must type on the browser or copy and paste onto it, i.e., <http://127.0.0.1:8050/>.

## Results

With the aim of demonstrating the usability of MAGFlow/BIgMAG, we used it to benchmark the MAG features recovered from a mock community (ATCC MSA-1003<sup>TM</sup>, Table 1) by 6 different pipelines, namely ATLAS, DATMA, MetaWRAP, MUFFIN, nf-core/mag and SnakeMAGs (Kieser et al., 2020; Benavides et al., 2020; Uritskiy et al., 2018; van Damme et al., 2021; Krakau et al., 2022; Tadrent et al., 2023). The selected pipelines were performed using short reads only (Illumina), except for MUFFIN which was tested for hybrid assembly (Illumina and PacBio). Additionally, in order to establish the same starting point for all the pipelines Megahit (Li et al., 2016) was set as assembly software, as well as MetaBAT2 (Kang et al., 2019) was selected as the binning tool, excluding DATMA since this workflow follows a different strategy to recover the MAGs that groups first the reads using a specific approach called CLAME (Benavides et al., 2018), to assemble them in batches afterwards.

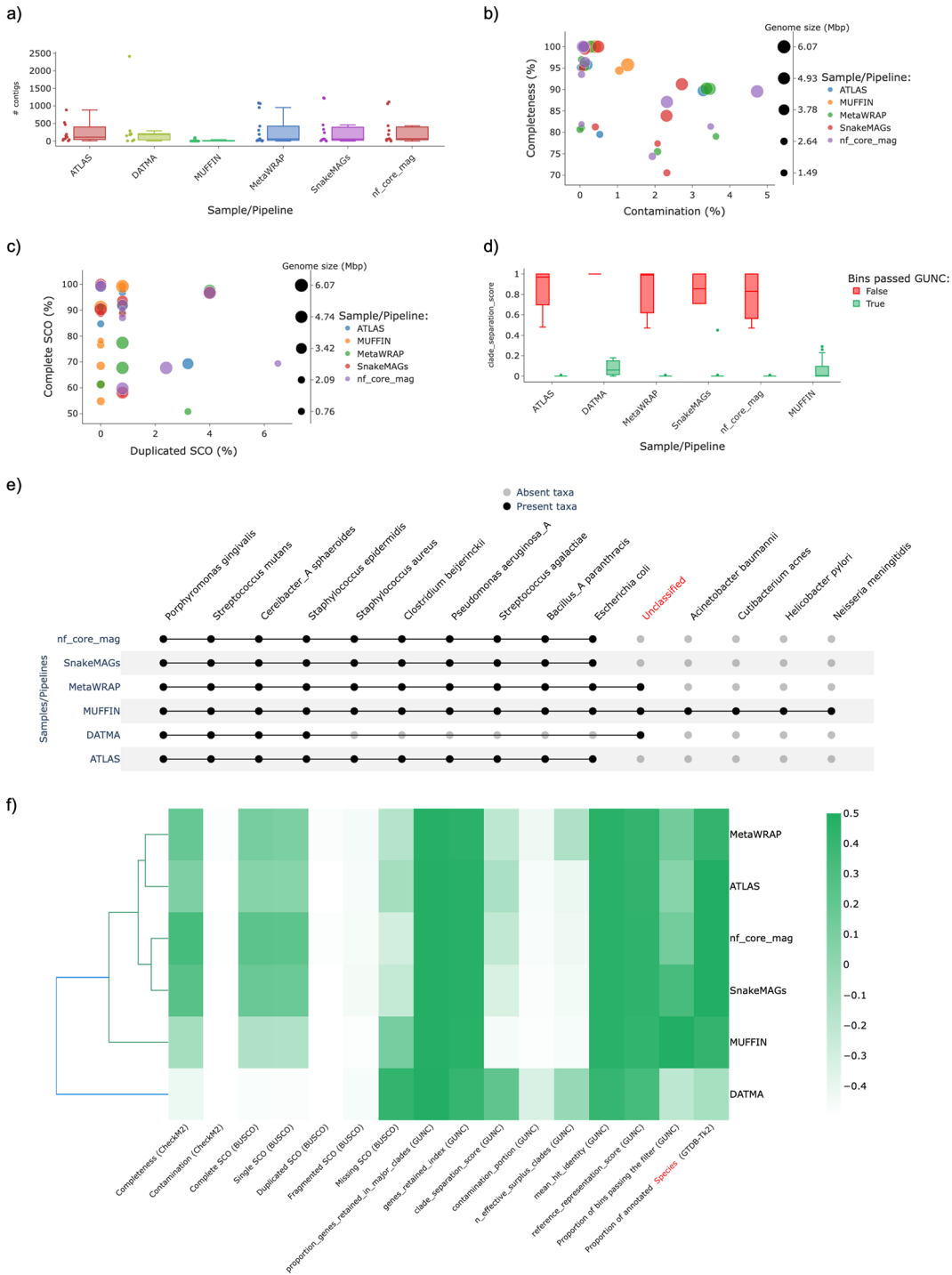
The results obtained by MAGFlow and displayed by BIgMAG for this experiment highlight the different behavior the pipelines in their default implementation can depict even if they account with the same tools to assemble the reads and group the contigs. For instance, MetaWRAP and ATLAS show a higher dispersion in terms of the number of contigs assigned per bin (Figure 2a), whilst nf-core/mag depicted a higher proportion of MAGs with higher quality (completeness > 70% and contamination < 5%) (Figure 2b). Likewise, nf-core/mag and MetaWRAP exhibit a trend to produce more MAGs with duplicated SCO with at least 50% of complete SCO (Figure 2c). Another important aspect to consider to compare the pipeline performance is the Clade Separation Score (CSS) calculated by GUNC, a metric that indicates the level of taxonomical chimerism in each contig of a MAG, where MetaWRAP, ATLAS, nf-core/mag and SnakeMAGs show the highest consistency and correspondence between the MAGs passing the GUNC test and the CSS value (Figure 2d). Regarding the taxonomical annotation at species level, it is noticeable that DATMA is not as effective to recover the genomes from the mock community as the rest of the workflows, remaining the main proportion of these unannotated (Figure 2e). Also, MUFFIN outcomes indicate an enhancement in the recovery of less abundant genomes by performing a hybrid assembly with short and long read technologies, since species such as *Acinetobacter baumannii*, *Cutibacterium acnes*, *Helicobacter pylori* and *Neisseria meningitidis* were only detected by this pipeline; this improvement by achieved hybrid assemblies has been proved by Tao et al. (2023) and Liew et al. (2024).

As a result, the observations afore-mentioned contribute to the pipeline grouping displayed in the cluster heatmap, in which the metrics of the outcomes from MetaWRAP seem to be closer to the values obtained with ATLAS, and more dissimilar from the cluster that encompasses nf-core/mag and SnakeMAGs; MUFFIN and DATMA represent the most different workflows in regard to the quality features depicted by the MAGs assembled with these pipelines (Figure 2f).

In terms of performance, MAGFlow takes approximately 18 minutes to analyze the MAGs obtained from the mock community on an HPC cluster with 128 CPUs and 500 GB of available memory using the local executor and the default configuration that can be found on the file *MAGFlow/conf/base.config* at the repository (<https://github.com/jeffe107/MAGFlow>). Downloading the GTDB-Tk2 database can take up to 4 hours, while retrieving the GUNC database takes ~1 hour depending on the bandwidth of the user Internet network.

**Table 1. Description and details of the samples used to test MAGFlow/BIgMAG.**

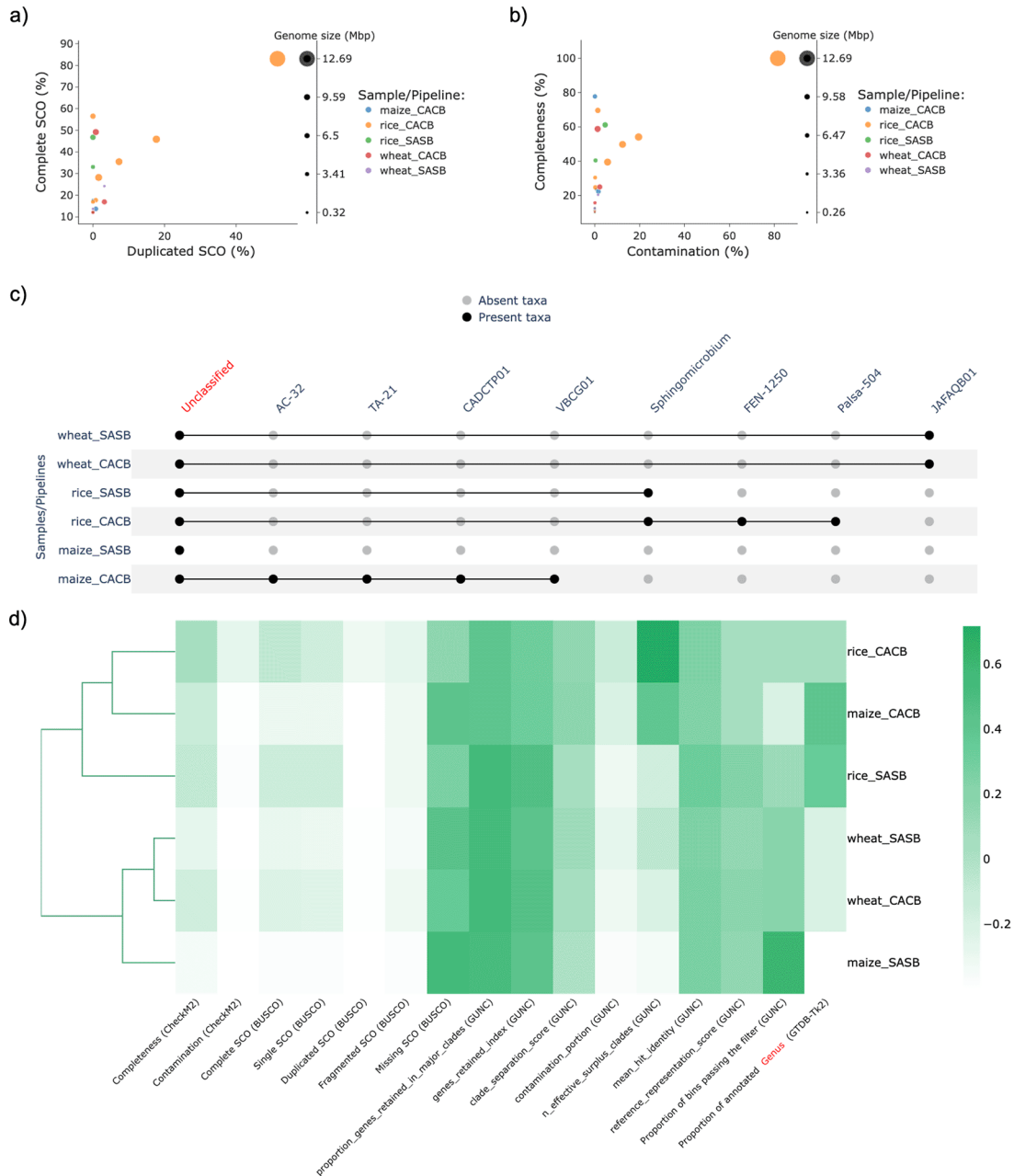
BioProject	Samples	Origin	Number of reads	Sequencing technology	Reference
PRJNA663614	SRR12687818	Rice soil	10.144.882	Illumina	Fernández-Baca et al., 2021
	SRR12687829		12.652.865		
	SRR12687830		8.286.935		
PRJNA448773	SRR7013867	Wheat soil	15.403.203		Li et al., 2020
	SRR7013874		16.440.929		
PRJNA645385	SRR12192848	Maize rhizosphere	9.483.602		Akinola et al., 2021
	SRR12192849		9.319.178		
	SRR12192850		11.889.426		
PRJNA510527	SRR8359173	Mock community	5.019.157		Portik et al., 2022
	SRR9328980		2.419.037		



**Figure 2.** Plots generated by BIGMAG depicting the outputs from different tools used to estimate the quality and annotate taxonomically the MAGs recovered from a sequenced mock community: *a)* distribution of the number of contigs per bin obtained by each pipeline, *b)* scatterplot of the completeness level against contamination portion of each MAG, *c)* data distribution of the CSS per pipeline filtered out by bins passing GUNC test, *d)* dispersion of the MAGs according to the presence of complete and duplicated SCO, *e)* presence/absence matrix representation depicting the presence/absence of annotated taxa in each pipeline at species level and *f)* cluster heatmap of the pipelines in function of their average values for different parameters.



In order to show another example of the utility of MAGFlow/BIgMAG, we used nf-core/mag to recover MAGs from several public metagenomics datasets generated from different crop rhizospheres or soil including rice, wheat and maize (Table 1). Additionally, two modes of assembly and binning were considered, namely SASB as single assembly/single binning and CACB as co-assembly/co-binning. Considering the intensive computational demands by the co-assembly process, only Megahit was used as assembly software and MetaBAT2 as the binning tool, allowing the same experimental setting for both SASB and CACB modes. Besides, the raw reads were previously cleaned using fastp (Chen et al., 2018),



**Figure 3.** Results of the experiment to explore the recovered MAGs by nf-core/mag from different soil/rhizosphere samples using MAGFlow/BIgMAG: *a)* dispersion of the MAGs according to the presence of complete and duplicated SCO, *b)* scatterplot of the completeness level against contamination portion of each bin/MAG, *c)* presence/absence matrix representation depicting the presence/absence of annotated taxa in each pipeline at genus level and *d)* cluster heatmap of the pipelines in function of their average values for different parameters. The names of the samples are represented by merging the origin of the sample (rice, wheat or maize) and the mode used by the pipeline to obtain the MAGs (SASB or CACB).



and a host removal was performed with bowtie2 (Langmead & Salzberg, 2012), indexing the proper host genome according to the type of crop the samples were obtained from.

Afterwards, the MAGs retrieved under each condition were used as input for MAGFlow/BIgMAG, and leveraging on these outcomes it is possible to notice how rice samples remarkably allowed to recover a higher number of MAGs, depicting greater quality in terms of percentage of complete SCO (Figure 3a), mainly for the CACB mode even though all the co-assemblies accounted with a similar number of reads (sequencing depth). This situation highlights how the sequencing depth is not the single factor affecting the assembly process and stresses the need to consider how additional conditions can influence the results such as the error rate of the sequencing method, the complexity of soil/rhizosphere samples, and the read length (Sims et al., 2014).

Moreover, this application of MAGFlow/BIgMAG underlines the advantages CACB permits by increasing both the quality and the number of MAGs obtained per type of sample. This is supported by the obtention of more mid-quality MAGs (completeness > 50% and contamination < 10%) (Figure 3b), as well as by the higher proportion of annotated MAGs at genus levels for all soil/rhizosphere samples treated under CACB mode (Figure 3c). These observations can even lead to cluster samples assembled under CACB in the same group when they belong to different matrices (Figure 3d). Similar outcomes have been published by Vosloo et al. (2021) during experiments set with this type of operational mode, fixing metaSPAdes as assembler and MetaBAT2 as binning software.

## Discussion

The strategy MAGFlow uses to perform the analysis attempts to follow the nf-core guidelines (P. A. Ewels et al., 2020), which allows robustness and reproducibility, and it has been tested using nf-test. This is supported by its Nextflow wrapping, along with its great performance features such as the possibility to run the software independently and in parallel, the chance to couple it to many other Nextflow modules or pipelines and the property to be scalable and adjusted to the available resources on the system user (Di Tommaso et al., 2017). MAGFlow can also be implemented through a wide variety of profiles such as Conda, Mamba, Docker, Singularity and Apptainer, which increases the scope of suitable systems to execute it. In addition, the pipeline can be launched in local environments, HPC clusters featuring executors as SLURM or SGE, or cloud-based solutions such as Azure Batch or AWS Batch (native Nextflow functionality not tested for MAGFlow).

When writing this report, there is another specific workflow aiming to run multiple tools to measure the quality of metagenomes called GENcontams.nf, which is part of the GEN-ERA toolbox (Cornet et al., 2022). Among the advantages of this package, we can enumerate the inclusion of tools that MAGFlow currently lacks, such as Physter, Kraken2 and EukCC, the wide control the user can apply over the parameters to run the software, as well as the integration with the GEN-ERA suite. Nonetheless, GENcontams.nf is written under Nextflow DSL1, which is no longer supported by the developers, it does not incorporate a visualization module, and it is not modularized, meaning that all of the analysis are performed in a single script, increasing the likely to fail, and limiting the possibility to couple it to other Nextflow pipelines. Besides, GENcontams.nf does not perform the taxonomical annotation with GTDB-Tk2, leaving this task to a different module within the GEN-ERA toolbox.

On the other hand, in terms of visualization and portability of the plots, MultiQC (Ewels et al., 2016) is the top-of-the-notch application that has eased the process of output integration from several pieces of software; however, out of the tools implemented in MAGFlow, MultiQC only accounts with support for QUAST and BUSCO data obtained from their execution elsewhere.

MAGFlow/BIgMAG attempts to fulfil these limitations in the field of metagenomics data analysis given the reproducibility and scalability of its workflow to assess the quality of the MAGs, the complementary functionalities it includes varying from taxonomic classification to assembly metrics passing through structural integrity, the easiness to analyze as many MAG folders as the user decides and the modularization to enclose the software tools in separate Nextflow modules. Furthermore, BIgMAG complements MAGFlow by generating high-quality automatic plots for all the encompassed tools, allowing a high degree of interactivity and customization according to the user needs. As a result, MAGFlow/BIgMAG represents a unique tool to our knowledge that integrates the execution of the software with a visualization module to extract MAG quality information and taxonomical features, making it interestingly useful when targeting comparisons among different metagenomics pipelines or tools to bin contigs. Also, MAGFlow/BIgMAG provides a convenient support during exploratory analyses that involve establishing general differences across samples as we showed with the examples presented in this paper.

Finally, following the principle of divide-and-rule proposed by [Lupo et al. \(2021\)](#), which suggests that genome quality and contamination should be assessed with as many different tools as possible, MAGFlow/BIgMAG scope will be continuously expanded through the inclusion of additional new tools, such as Kraken2, Physeter, EukCC, among others.

### Data availability

The raw sequencing data to test MAGFlow/BIgMAG were retrieved from the Sequence Read Archive (SRA), and they can be accessed through the following identifiers:

Sequence Read Archive: NextSeq500 of rice rhizosphere soil: Francis booting rep1. Accession number SRR12687830; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12687830> ([Fernández-Baca et al., 2021](#)).

Sequence Read Archive: NextSeq500 of rice rhizosphere soil: Francis booting rep2. Accession number SRR12687829; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12687829> ([Fernández-Baca et al., 2021](#)).

Sequence Read Archive: NextSeq500 of rice rhizosphere soil: Francis booting rep3. Accession number SRR12687818; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12687818> ([Fernández-Baca et al., 2021](#)).

Sequence Read Archive: AAFC-Pcyc\_02A\_Pos1\_Plot18\_noN\_noP\_2016-08. Accession number SRR7013867; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR7013867> ([Li et al., 2020](#)).

Sequence Read Archive: AAFC-Pcyc\_12A\_Pos1\_Plot73\_noN\_noP\_2016-08. Accession number SRR7013874; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR7013874> ([Li et al., 2020](#)).

Sequence Read Archive: Rhizosphere soil 1. Accession number SRR12192850; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12192850> ([Akinola et al., 2021](#)).

Sequence Read Archive: Rhizosphere soil 2. Accession number SRR12192849; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12192849> ([Akinola et al., 2021](#)).

Sequence Read Archive: Rhizosphere soil 3. Accession number SRR12192848; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR12192848> ([Akinola et al., 2021](#)).

Sequence Read Archive: Metagenome\_ID964\_ATCC. Accession number SRR8359173; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR8359173> ([Portik et al., 2022](#)).

Sequence Read Archive: WGS of ATCC MSA-1003 Mock Microbial Community with PacBio CCS on the Sequel II System. Accession number SRR9328980; <https://www.ncbi.nlm.nih.gov/sra/?term=SRR9328980> ([Portik et al., 2022](#)).

The bash scripts with all required configurations to run each pipeline as we performed in this paper, the MAGs recovered by each pipeline using the mock community or soil/rhizosphere samples, as well as the outputs generated by MAGFlow/BIgMAG during the analysis of these MAGs have been made available in a Zenodo repository.

Zenodo: Metagenome quality metrics and taxonomical annotation visualization through the integration of MAGFlow and BIgMAG (Sup. Material). <https://doi.org/10.5281/zenodo.11003957>.

This project contains the following underlying data:

- The recovered MAGs by 6 different metagenomics pipelines (ATLAS, DATMA, MetaWRAP, MUFFIN, nf-core/mag and SnakeMAGs) using a mock community as input (SRR8359173 and SRR9328980), complemented with the output from MAGFlow using these MAGs as input for their quality assessment and taxonomical annotation.
- The MAGs produced by nf-core/mag using rice/rhizosphere sequenced libraries (PRJNA663614, PRJNA448773 and PRJNA645385) in either single assembly/single binning or co-assembly/co-binning mode, complemented with the output from MAGFlow using these MAGs as input for their quality assessment and taxonomical annotation.

- Scripts, commands and configuration files to run the different pipelines (ATLAS, DATMA, MetaWRAP, MUFFIN, nf-core/mag and SnakeMAGs) and reproduce the experimental conditions.

Data are available under the terms of the [Creative Commons Attribution 4.0 International license \(CC-BY 4.0\)](#).

### Software availability

The MAGFlow/BIgMAG source code is freely available as GitHub repositories, and it can be used under an MIT license. A detailed tutorial to install the dependencies, configuration setting instructions, practical visualization tips, a demo video and example commands are documented on the repositories.

MAGFlow:

Source code: <https://github.com/jeffe107/MAGFlow>.

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.11178032>

BIgMAG:

Source code: <https://github.com/jeffe107/BIgMAG>.

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.11178078>

License: [MIT license](#).

### Acknowledgements

MAGFlow/BIgMAG is being developed under the support of the Federal Commission for Scholarships for Foreign Students (FCS) through the program Swiss Government Excellence Scholarships.

### References

- Akinola SA, Ayangbenro AS, Babalola OO: **Metagenomic insight into the community structure of maize-rhizosphere bacteria as predicted by different environmental factors and their functioning within plant proximity.** *Microorganisms*. 2021; **9**(7): 1419.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Bayer K, Busch K, Kenchington E, et al.: **Microbial Strategies for Survival in the Glass Sponge *Vazella pourtalesii*.** *MSystems*. 2020; **5**(4): e00473–e00420.  
[Publisher Full Text](#)
- Benavides A, Isaza JP, Niño-García JP, et al.: **CLAME: a new alignment-based binning algorithm allows the genomic description of a novel *Xanthomonadaceae* from the Colombian Andes.** *BMC Genomics*. 2018; **19**(Suppl 8): 858.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Benavides A, Sanchez F, Alzate JF, et al.: **DATMA: Distributed Automatic Metagenomic Assembly and annotation framework.** *PeerJ*. 2020; **8**: e9762.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chaumeil P-A, Mussig AJ, Hugenholtz P, et al.: **GTDB-Tk v2: memory friendly classification with the genome taxonomy database.** *Bioinformatics*. 2022; **38**(23): 5315–5316.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chen S, Zhou Y, Chen Y, et al.: **fastp: an ultra-fast all-in-one FASTQ preprocessor.** *Bioinformatics*. 2018; **34**(17): i884–i890.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chklovski A, Parks DH, Woodcroft BJ, et al.: **CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning.** *Nat. Methods*. 2023; **20**(8): 1203–1212.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Cornet L, Baurain D: **Contamination detection in genomic data: more is not enough.** *Genome Biol*. 2022; **23**(1): 1–15.  
[Publisher Full Text](#)
- Cornet L, Durieu B, Baert F, et al.: **The GEN-ERA toolbox: unified and reproducible workflows for research in microbial genomics.** *GigaScience*. 2022; **12**: 1–10.  
[Publisher Full Text](#)
- Di Tommaso P, Chatzou M, Floden EW, et al.: **Nextflow enables reproducible computational workflows.** *Nat. Biotechnol*. 2017; **35**(4): 316–319.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Ewels PA, Peltzer A, Fillinger S, et al.: **The nf-core framework for community-curated bioinformatics pipelines.** *Nat. Biotechnol*. 2020; **38**(3): 276–278.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Ewels P, Magnusson M, Lundin S, et al.: **MultiQC: summarize analysis results for multiple tools and samples in a single report.** *Bioinformatics*. 2016; **32**(19): 3047–3048.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Fernández-Baca CP, Rivers AR, Kim W, et al.: **Changes in rhizosphere soil microbial communities across plant developmental stages of high and low methane emitting rice genotypes.** *Soil Biol. Biochem*. 2021; **156**: 108233.  
[Publisher Full Text](#)
- Gurevich A, Saveliev V, Vyahhi N, et al.: **QUAST: quality assessment tool for genome assemblies.** *Bioinformatics*. 2013; **29**(8): 1072–1075.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Guseva K, Darcy S, Simon E, et al.: **From diversity to complexity: Microbial networks in soils.** *Soil Biol. Biochem*. 2022; **169**: 108604.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Haryono MAS, Law YY, Arumugam K, et al.: **Recovery of High Quality Metagenome-Assembled Genomes From Full-Scale Activated Sludge Microbial Communities in a Tropical Climate Using Longitudinal Metagenome Sampling.** *Front. Microbiol*. 2022; **13**: 869135.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

- Kang DD, Li F, Kirton E, *et al.*: **MetaBAT 2: An adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies.** *PeerJ*. 2019; **7**: e7359.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kieser S, Brown J, Zdobnov EM, *et al.*: **ATLAS: A Snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data.** *BMC Bioinformatics*. 2020; **21**(1): 1–8.  
[Publisher Full Text](#)
- Krakau S, Straub D, Gourel H, *et al.*: **nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning.** *NAR Genom. Bioinform.* 2022; **4**(1).  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Langmead B, Salzberg SL: **Fast gapped-read alignment with Bowtie 2.** *Nat. Methods*. 2012; **9**(4): 357–359.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Letunic I, Bork P: **Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation.** *Nucleic Acids Res.* 2021; **49**(W1): W293–W296.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Li D, Luo R, Liu CM, *et al.*: **MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices.** *Methods*. 2016; **102**: 3–11.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Li Y, Tremblay J, Bainard LD, *et al.*: **Long-term effects of nitrogen and phosphorus fertilization on soil microbial community structure and function under continuous wheat production.** *Environ. Microbiol.* 2020; **22**(3): 1066–1088.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Liew KJ, Shahar S, Shamsir MS, *et al.*: **Integrating multi-platform assembly to recover MAGs from hot spring biofilms: insights into microbial diversity, biofilm formation, and carbohydrate degradation.** *Environ. Microbiome*. 2024; **19**(1): 1–20.  
[Publisher Full Text](#)
- Lupo V, Van Vlierberghe M, Vanderschuren H, *et al.*: **Contamination in Reference Sequence Databases: Time for Divide-and-Rule Tactics.** *Front. Microbiol.* 2021; **12**.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Manni M, Berkeley MR, Seppely M, *et al.*: **BUSCO: Assessing Genomic Data Quality and Beyond.** *Curr. Protoc.* 2021; **1**(12): e323.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Orakov A, Fullam A, Coelho LP, *et al.*: **GUNC: detection of chimerism and contamination in prokaryotic genomes.** *Genome Biol.* 2021; **22**(1): 1–19.  
[Publisher Full Text](#)
- Portik DM, Brown CT, Pierce-Ward NT: **Evaluation of taxonomic classification and profiling methods for long-read shotgun metagenomic sequencing datasets.** *BMC Bioinformatics*. 2022; **23**(1): 1–39.  
[Publisher Full Text](#)
- Sims D, Sudbery I, Illott NE, *et al.*: **Sequencing depth and coverage: key considerations in genomic analyses.** *Nat. Rev. Genet.* 2014; **15**(2): 121–132.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Tadrent N, Dedeine F, Hervé V, *et al.*: **SnakeMAGs: a simple, efficient, flexible and scalable workflow to reconstruct prokaryotic genomes from metagenomes.** *F1000Res*. 2023; **11**: 1522.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Tao Y, Xun F, Zhao C, *et al.*: **Improved Assembly of Metagenome-Assembled Genomes and Viruses in Tibetan Saline Lake Sediment by HiFi Metagenomic Sequencing.** *Microbiology Spectrum*. 2023; **11**(1): 1–18.  
[Publisher Full Text](#)
- Tremblay J, Schreiber L, Greer CW: **High-resolution shotgun metagenomics: the more data, the better?** *Brief. Bioinform.* 2022; **23**(6): 1–16.  
[Publisher Full Text](#)
- Uritskiy GV, Diruggiero J, Taylor J: **MetaWRAP - A flexible pipeline for genome-resolved metagenomic data analysis.** *Microbiome*. 2018; **6**(1): 1–13.  
[Publisher Full Text](#)
- van Damme R, Hölzer M, Viehweger A, *et al.*: **Metagenomics workflow for hybrid assembly, differential coverage binning, metatranscriptomics and pathway analysis (MUFFIN).** *PLoS Comput. Biol.* 2021; **17**(2): 1–13.
- Vosloo S, Huo L, Anderson CL, *et al.*: **Evaluating de Novo Assembly and Binning Strategies for Time Series Drinking Water Metagenomes.** *Microbiol. Spectr.* 2021; **9**(3): e0143421.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Wratten L, Wilm A, Göke J: **Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers.** *Nat. Methods*. 2021; **18**(10): 1161–1168.  
[PubMed Abstract](#) | [Publisher Full Text](#)
- Yang C, Chowdhury D, Zhang Z, *et al.*: **A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data.** *Comput. Struct. Biotechnol. J.* 2021; **19**: 6301–6314.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact [research@f1000.com](mailto:research@f1000.com)

**F1000Research**