MetaGenePipe: An Automated, Portable Pipeline for Contig-based Functional and Taxonomic Analysis

Babak Shaban[1], **Maria del Mar Quiroga**[1¶], **Robert Turnbull**[1], **Edoardo Tescari**[1], **Kim-Anh Lê Cao**[2*], **and Heroen Verbruggen**[3*]

1 Melbourne Data Analytics Platform, The University of Melbourne **2** School of Mathematics and Statistics, Melbourne Integrative Genomics, The University of Melbourne **3** School of BioSciences, The University of Melbourne ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.04851

Software

- Review
- Repository
- Archive

Editor: Jacob Schreiber

Reviewers:

- @Ebedthan
- @rjorton

Submitted: 08 September 2022
Published: 20 February 2023

Summary

MetaGenePipe (MGP) is an efficient, flexible, portable, and scalable metagenomics pipeline that uses performant bioinformatics software suites and genomic databases to create an accurate taxonomic and functional characterization of the prokaryotic fraction of sequenced microbiomes. Written in the Workflow Definition Language (WDL), MGP produces output that can be explored and interpreted directly, or can be used for downstream analysis. MGP is a pipeline-development best practice tool that uses Singularity for containerization and includes a setup script that downloads the necessary databases for setup. The source code for MGP is freely available and distributed under the Apache 2.0 license.

The workflow uses MegaHIT for read assembly and the user can specify whether to co-assemble multiple samples or do sample-by-sample assembly. A BLAST search is carried out against a user-specified BLAST database.

Coding regions are predicted in contigs with Prodigal and taxonomic annotation of the predicted protein-coding genes is done using DIAMOND alignment against the Swiss-Prot database. Functional annotation uses HMMER searches against the KoalaFam HMM profiles. The main outputs of the workflow are tables with counts of taxonomic (organism) and functional hits against the reference databases, which can be easily employed for downstream statistical analysis. MGP's focus can be easily modified to find viruses, bacteria, plants, archaea, vertebrates, invertebrates, or fungi by choosing suitable reference databases.

Statement of need

MetaGenePipe (`MGP`) is a pipeline for characterizing the prokaryotic fraction of whole genome metagenomics shotgun sequencing data functionally and taxonomically.

`MGP` was designed to be used by computational microbiologists, written in WDL to make further customization accessible to researchers. `MGP` uses a Singularity container to overcome traditional portability obstacles and caters to a flexible research focus. For example, the default DIAMOND and BLAST databases can be replaced with any relevant databases owned by the researcher via an update in the configuration file. While `MGP` is focussed on prokaryotes, it can easily be adapted to eukaryotes or viruses by changing the prokaryotic gene prediction software, Prodigal, to eukaryotic gene prediction software such as GeneMark-EP+ (Brůna et al., 2020) or EuGene (Sallet et al., 2019), or a gene finding tool for viruses (González-Tortuero et al., 2021).

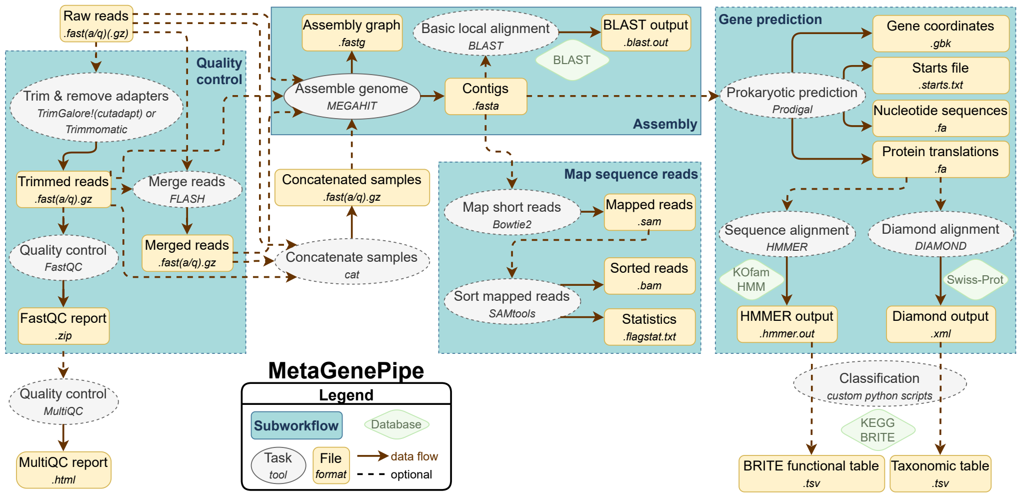

Figure 1: The MetaGenePipe Workflow

Workflow

MGP is written in the Workflow Definition Language (WDL), renowned for its human readable and writable syntax. Singularity (Kurtzer et al., 2017) is used to containerize the software required for MGP, and is stored in SylabsCloud for universal accessibility.

MGP is broken up into four sub-workflows: Quality Control (QC), Assembly, Map Sequence Reads, and Gene Prediction.

There are currently several assembly-based taxonomic software suites such as MG-RAST (Keegan et al., 2016), MMseqs2 (Mirdita et al., 2021), and a few that make use of a workflow language, including nf-core and Muffin written in Nextflow, and Atlas written in Snakemake (Di Tommaso et al., 2017; Kieser et al., 2020; Krakau et al., 2022; Mölder et al., 2021; Van Damme et al., 2021). The main advantage of MGP over MG-RAST is its ease of installation on local infrastructure, as it only requires running a setup script that downloads a supplied Singularity image from SylabsCloud, the latest version of Cromwell (Voss et al., 2022), Koalafam HMMER profiles (Takuya Aramaki et al., 2019), and the Swiss-Prot database (UniProt Consortium, 2018) which is converted to the DIAMOND aligner (Buchfink et al., 2015) format. This setup allows MGP to be used on a range of computing infrastructures across institutions.

QC sub-workflow

The quality control (QC) sub-workflow trims poor quality reads and any potential adapter sequence from the genomic samples using either Trimmomatic (Bolger et al., 2014) or TrimGalore (Krueger et al., 2021). There is also the option of lengthening the reads by merging overlapping paired-end reads using FLASH. This option can help overcome potential low-coverage regions encountered during the assembly process (Magoč & Salzberg, 2011). Visualizations of the sequence quality are obtained using FastQC (Andrews, 2010).

There is an optional extra standalone task to merge and analyze the FastQC outputs from each of the samples using MultiQC (Ewels et al., 2016).

Concatenate samples

This standalone and optional task can consolidate all forward and all reverse reads into single forward and reverse files. This step is intended to facilitate the co-assembly of the available sequences, which has been shown to provide more complete genomes with lower error rates when compared to multiassembly (Hofmeyr et al., 2020).

Assembly sub-workflow

For assembling contigs we chose MegaHIT (D. Li et al., 2015), which performs de-novo assembly of large and complex metagenomic samples in a time and cost-efficient manner (D. Li et al., 2015).

BLAST (Altschul et al., 1990, 1997; Camacho et al., 2009) is used to query the contigs created during assembly against a user-specified BLAST database (e.g., mito, nt, nr), downloaded from the NCBI ftp server ([https://ftp.ncbi.nlm.nih.gov/blast/db/]). The BLAST output is parsed to be easily searchable and also lists queries returning no hits. This informs researchers to further investigate potentially novel sequences. Additionally, the BLAST results can be used to filter contigs that belong to a taxon of interest that was not matched during the Swiss-Prot alignment stage. This can be useful for genomic binning or investigation of regions of interest.

Map reads sub-workflow

Mapping the raw reads back to the assembled contigs allows for the quantification of the relative abundance of contigs in a metagenomics dataset. This task is important for downstream genomic binning and metagenome statistics. The raw reads that have passed the QC stage are at this point aligned back to the contigs resulting from the assembly sub-workflow using the Bowtie2 Aligner (Langmead & Salzberg, 2012). A compressed binary file representing the alignment of sorted raw sequences to the assembly output in BAM format is created via SAMtools (H. Li et al., 2009). Mapping statistics for the alignment are created using the SAMtools flagstat function, which can be used to quantify relative abundance.

Gene prediction sub-workflow

The gene prediction sub-workflow uses Prodigal (PROkaryotic Dynamic programming Genfinding Algorithm) (Hyatt, Chen, LoCascio, et al., 2010) for predicting prokaryotic gene coding sequences and identifying the sites of translation initiation (Hyatt, Chen, Locascio, et al., 2010). Prodigal produces a Fasta file with the predicted amino acid (protein) coding sequences. These are then aligned to the Swiss-Prot database (Boutet et al., 2007) with the DIAMOND aligner, and the output is parsed to generate a table of taxonomic (organism) identifications. The reference database can be easily exchanged if Swiss-Prot is not suitable for the user's application.

The protein coding sequences are also aligned to the KoalaFam HMM profiles with HMMER (T. Aramaki et al., 2020), allowing assigning KEGG pathways and EC numbers to the proteins. Custom Python scripts are used to output counts at the A, B and C levels of the KEGG Brite hierarchical function classification (Kanehisa, 2019; Kanehisa et al., 2021; Kanehisa & Goto, 2000).

Outputs and interpretation

The principal outputs for downstream analysis are the assembly, the mapped reads, and tables of the functional and taxonomic (organisms) counts of the genes found in the contigs. For the assembly, outputs include the final contigs as well as the assembly graph which users can visualize (Wick et al., 2015) or use for downstream graph-based metagenome binning (Mallawaarachchi & Lin, 2022). The mapping results in SAM/BAM mapping files for each pair of read files, which can be used for downstream metagenome binning applications, or users could use these to obtain a list of contigs with read depth metrics by running these through the jgi_summarize_bam_contig_depths tool, available through MetaBAT (Kang et al., 2019). The taxonomic and functional tables of gene counts produced in the gene prediction sub-workflow are suitable for downstream statistical analysis. Several other potentially useful intermediate files are saved in each of the sub-workflows. A description of each of the output files and an example hierarchical structure of the output directory are available in the MGP documentation.

Resource usage and infrastructure requirements

MGP uses Unix's `time` tool to measure the resources used by each task, such as CPU usage, file size, elapsed time, and system time. This output can be visualized and used to inform resource requests when using a job scheduler on high-performance computing infrastructure. Table 1 shows indicative resource usage for processing paired-end samples of 25,000 reads each run on the University of Melbourne SPARTAN high-performance computing system consisting of Intel Xeon Gold 6154 3GHz CPUs. Running Cromwell on the head node took 2 minutes and 23 seconds and required a maximum memory of 840 MB.

MGP can be run locally on a laptop, a virtual machine, or in a high-performance computing setting.

Task	User time (mm:ss)	Threads	CPU utilization	Max memory (kbytes)
trimgalore	00:02.5	4	103%	23144
fastqc	00:05.78	2	120%	305268
multiqc	00:01.24	1	38%	84960
flash	00:00.17	2	46%	20388
concatenate	00:00.3	1	54%	20372
megahit	07:25.19	24	1770%	70928
blast	00:00.13	6	43%	22560
map reads	00:01.19	4	107%	92144
prodigal	00:00.07	1	39%	56796
diamond	00:10.23	18	535%	433768
hmmer	00:47.9	8	106%	59428
taxonomic classification	00:00.56	1	46%	74532

Table 1: The resource usage for processing two paired end samples of 25,000 reads each in MetaGenePipe.

Acknowledgements

We thank the members of the Verbruggen lab, Kshitij Tandon and Vinícius Salazar in particular, for sharing ideas, feedback, and testing the workflow. This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. The project benefited from funding by the Australian Research Council (DP200101613 to Heroen Verbruggen).

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, *25*(17), 3389–3402. https://doi.org/10.1093/nar/25.17.3389

Andrews, S. (2010). *FASTQC. A quality control tool for high throughput sequence data.* https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Aramaki, Takuya, Blanc-Mathieu, R., Endo, H., Ohkubo, K., Kanehisa, M., Goto, S., & Ogata, H. (2019). KofamKOALA: KEGG ortholog assignment based on profile HMM and adaptive score threshold. *Bioinformatics*, *36*(7), 2251–2252. https://doi.org/10.1093/bioinformatics/btz859

Aramaki, T., Blanc-Mathieu, R., Endo, H., Ohkubo, K., Kanehisa, M., Goto, S., & Ogata, H. (2020). KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold. *Bioinformatics*, *36*(7), 2251–2252. https://doi.org/10.1093/bioinformatics/btz859

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, *30*(15), 2114–2120. https://doi.org/10.1093/bioinformatics/btu170

Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., & Bairoch, A. (2007). UniProtKB/Swiss-Prot. *Methods Mol Biol*, *406*, 89–112. https://doi.org/10.1007/978-1-59745-535-0_4

Brůna, T., Lomsadze, A., & Borodovsky, M. (2020). GeneMark-EP+: eukaryotic gene prediction with self-training in the space of genes and proteins. *NAR Genomics and Bioinformatics*, *2*(2). https://doi.org/10.1093/nargab/lqaa026

Buchfink, B., Xie, C., & Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nat. Methods*, *12*(1), 59–60. https://doi.org/10.1038/nmeth.3176

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, *10*(1), 421. https://doi.org/10.1186/1471-2105-10-421

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*, *35*(4), 316–319. https://doi.org/10.1038/nbt.3820

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016). MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, *32*(19), 3047–3048. https://doi.org/10.1093/bioinformatics/btw354

González-Tortuero, E., Krishnamurthi, R., Allison, H. E., Goodhead, I. B., & James, C. E. (2021). Comparative analysis of gene prediction tools for viral genome annotation. *bioRxiv*. https://doi.org/10.1101/2021.12.11.472104

Hofmeyr, S., Egan, R., Georganas, E., Copeland, A. C., Riley, R., Clum, A., Eloe-Fadrosh, E., Roux, S., Goltsman, E., Buluç, A., Rokhsar, D., Oliker, L., & Yelick, K. (2020). Terabase-scale metagenome coassembly with MetaHipMer [Journal Article]. *Scientific Reports*, *10*(1), 10689. https://doi.org/10.1038/s41598-020-67416-5

Hyatt, D., Chen, G.-L., LoCascio, P. F., Land, M. L., Larimer, F. W., & Hauser, L. J. (2010). Prodigal: Prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, *11*(1). https://doi.org/10.1186/1471-2105-11-119

Hyatt, D., Chen, G.-L., Locascio, P. F., Land, M. L., Larimer, F. W., & Hauser, L. J. (2010). Prodigal: Prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, *11*(1), 119. https://doi.org/10.1186/1471-2105-11-119

Kanehisa, M. (2019). Toward understanding the origin and evolution of cellular organisms. *Protein Sci*, *28*(11), 1947–1951. https://doi.org/10.1002/pro.3715

Kanehisa, M., Furumichi, M., Sato, Y., Ishiguro-Watanabe, M., & Tanabe, M. (2021). KEGG: integrating viruses and cellular organisms. *Nucleic Acids Res*, *49*(D1), D545–D551. https://doi.org/10.1093/nar/gkaa970

Kanehisa, M., & Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, *28*(1), 27–30. https://doi.org/10.1093/nar/28.1.27

Kang, D. D., Li, F., Kirton, E., Thomas, A., Egan, R., An, H., & Wang, Z. (2019). MetaBAT 2: An adaptive binning algorithm for robust and efficient genomereconstruction from metagenome assemblies. [Journal Article]. *PeerJ*, *7*, e7359. https://doi.org/10.7717/peerj.7359

Keegan, K. P., Glass, E. M., & Meyer, F. (2016). MG-RAST, a metagenomics service for analysis of microbial community structure and function. *Microbial Environmental Genomics (MEG)*, 207–233. https://doi.org/10.1007/978-1-4939-3369-3_13

Kieser, S., Brown, J., Zdobnov, E. M., Trajkovski, M., & McCue, L. A. (2020). ATLAS: A snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data. *BMC Bioinformatics*, *21*(1). https://doi.org/10.1186/s12859-020-03585-4

Krakau, S., Straub, D., Gourlé, H., Gabernet, G., & Nahnsen, S. (2022). nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning. *NAR Genomics and Bioinformatics*, *4*(1). https://doi.org/10.1093/nargab/lqac007

Krueger, F., James, F., Ewels, P., Afyounian, E., & Schuster-Boeckler, B. (2021). *TrimGalore* (Version 0.6.7). Zenodo. https://doi.org/10.5281/zenodo.5127899

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PLOS ONE*, *12*(5), e0177459. https://doi.org/10.1371/journal.pone.0177459

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2 [Journal Article]. *Nature Methods*, *9*(4), 357–359. https://doi.org/10.1038/nmeth.1923

Li, D., Liu, C.-M., Luo, R., Sadakane, K., & Lam, T.-W. (2015). MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, *31*(10), 1674–1676. https://doi.org/10.1093/bioinformatics/btv033

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., & Subgroup, 1000. G. P. D. P. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Magoč, T., & Salzberg, S. L. (2011). FLASH: Fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, *27*(21), 2957–2963. https://doi.org/10.1093/bioinformatics/btr507

Mallawaarachchi, V., & Lin, Y. (2022). MetaCoAG: Binning metagenomic contigs via composition, coverage and assembly graphs. In I. Pe'er (Ed.), *Research in computational molecular biology* (pp. 70–85). Springer International Publishing. ISBN: 978-3-031-04749-7

Mirdita, M., Steinegger, M., Breitwieser, F., Söding, J., & Levy Karin, E. (2021). Fast and sensitive taxonomic assignment to metagenomic contigs. *Bioinformatics*, *37*(18), 3029–3031. https://doi.org/10.1093/bioinformatics/btab184

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., & Kanitz, A. et al. (2021). Sustainable data analysis with snakemake. *F1000Research*, *10*, 33. https://doi.org/10.12688/f1000research.29032.1

Sallet, E., Gouzy, J., & Schiex, T. (2019). EuGene: An automated integrative gene finder for eukaryotes and prokaryotes. In M. Kollmar (Ed.), *Gene prediction: Methods and protocols* (pp. 97–120). Springer New York. https://doi.org/10.1007/978-1-4939-9173-0_6

UniProt Consortium, T. (2018). UniProt: The universal protein knowledgebase. *Nucleic Acids Research*, *46*(5), 2699–2699. https://doi.org/10.1093/nar/gky092

Van Damme, R., Hölzer, M., Viehweger, A., Müller, B., Bongcam-Rudloff, E., & Brandt, C. (2021). Metagenomics workflow for hybrid assembly, differential coverage binning, metatranscriptomics and pathway analysis (MUFFIN). *PLOS Computational Biology*, *17*(2), e1008716. https://doi.org/10.1371/journal.pcbi.1008716

Voss, K., Van der Auwera, G., & Gentry, J. (2022). Full-stack genomics pipelining with GATK4 + WDL + cromwell. In *F1000research.com*. https://doi.org/10.7490/f1000research.1114634.1

Wick, R. R., Schultz, M. B., Zobel, J., & Holt, K. E. (2015). Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, *31*(20), 3350–3352. https://doi.org/10.1093/bioinformatics/btv383