`mga`: User-friendly `R` package for microbial genetic analysis of amplicon data

Erika Y. Lin[1,2], Laura Grieneisen[1,2], Michael J. Noonan[1,2,3]*

[1] Department of Biology, University of British Columbia Okanagan, Kelowna, British Columbia, Canada.

[2] Okanagan Institute for Biodiversity, Resilience, and Ecosystem Services, The University of British Columbia Okanagan, Kelowna, British Columbia, Canada.

[3] Department of Computer Science, Math, Physics, and Statistics, The University of British Columbia Okanagan, Kelowna, British Columbia, Canada.

*Corresponding Author: michael.noonan@ubc.ca; 1177 Research Road, Kelowna, BC, Canada V1V 1V7

Acknowledgements

This research was supported by the Irving K. Barber Faculty of Science from the University of British Columbia, Okanagan. A published bacterial dataset by Schloss et al. (2012) was also used to support the development of this package.

Data Availability

TBA

Conflict of Interest

The authors declare no competing interests.

Author Contributions

Erika Y. Lin and Michael J. Noonan conceived the ideas and designed methodology; Laura Grieneisen collected and provided the data; Laura Grieneisen and Michael J. Noonan reviewed the package structure and analyses; Erika Y. Lin analysed the data and led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

Abstract

Studying microbes is fundamental in advancing a wide array of scientific fields, from ecology to medicine and pharmaceutical research. With the increasing use of amplicon sequencing technology, new bioinformatics tools are needed to process and analyze microbial DNA sequencing data. Existing software for microbiome analysis, although extensive and established, are often either inflexible or difficult to learn for new users. Furthermore, to ensure reproducibility, researchers frequently build their workflows by coding in command-line programs like `R`, an open-source software made for quantitative analysis and graphical visualization. However, available `R` packages target different steps of the DNA analysis process and may be unintuitive for researchers with limited coding experience. To address this lack of rigorous but also easy-to-use software for microbiome analysis, we developed the `R` package `mga`: Microbial Genetic Analysis, that streamlines several key steps of genetic data processing. This openly accessible software package distinguishes genetic sequences, classifies taxonomy, builds evolutionary and co-occurrence networks, and calculates key measures of ecological community structure from filtered DNA sequences. To effectively handle genetic data for any microbial community analysis, the core function `mga()` was designed to be comprehensive, versatile, and user-friendly, collapsing thousands of lines of code into a few simple, yet flexible inputs. `mga` was tested for functionality and sensitivity with fungal and bacterial amplicon sequencing datasets, producing results consistent with the literature, and it is built to handle various microbial taxa targeted for amplicon sequencing. By facilitating downstream analyses, taxonomic filtering, and data visualization, `mga` supports reproducible and accessible microbiome research.

Keywords

Accessibility, Amplicon sequencing, Analytical software, Microbial genetics, Microbiome analysis, R, Reproducibility, User-friendly tools

Introduction

Modern advances in microbiology have garnered growing interest among new generations of researchers. The development of gene sequencing technologies and bioinformatic tools has been fundamental to overcoming limitations in investigating microbiota, supporting transformative research in ecology (Rotoni et al., 2022; Shade et al., 2012), evolution (Ward et al., 2021), environmental sciences (Castañeda Alejo et al., 2022), health sciences (McLean et al., 2022; Turnbaugh et al., 2007), medical research (Caldara et al., 2022; Yang et al., 2022), and more. However, while software progress, researchers may face challenges in deciding which tools to use for microbiome analysis, particularly during early career stages. Ideally, researchers should work with bioinformatic tools that are accessible, reproducible, practical, and suitable for their study area (Wen et al., 2023). Additionally, analytical tools should be easy to learn for new users. However, we currently lack data processing software that combine accessibility, reproducibility, and user experience.

Current established software include QIIME2 (Bolyen et al., 2019) and mothur (Schloss et al., 2009) , which are designed to be accessible, but have limited flexibility and contribute to the reproducibility crisis in research, due to hidden backend processes that inhibit replicability. In contrast, command-line programs using `R` (R Core Team, 2023) and Python, enable reproducible analyses via code availability and the capacity to select and tailor functions from software packages increases flexibility. However, flexible command-line tools are inherently inaccessible and unintuitive, particularly for researchers with little coding experience. Available `R` DNA analysis workflows are long and can therefore be computationally heavy and difficult to follow, even for experienced researchers. The last decade has seen the emergence of integrated `R` packages for comprehensive microbiome analysis, such as `phyloseq` (McMurdie & Holmes, 2013); however, these packages often lack versatility or functionality, including issues with data visualization (Wen et al., 2023).

To facilitate accessible research while maintaining reproducibility, we developed `mga`: Microbial Genetic Analysis, an `R` package that simplifies the complicated process of DNA sequence processing, effectively streamlining key components of the microbiome analysis workflow. By consolidating the benefits of widely used programs , our package is openly accessible, comprehensive, versatile, and user-friendly. Universal steps in sequence data processing primarily involve filtering and trimming sequence files, inferring operational taxonomic units (OTUs) or amplicon sequence variants (ASVs) within samples, assigning taxonomy , and reconstructing the phylogeny of sample communities (Callahan, Sankaran, et al., 2016). As such, the core function `mga()` distills this workflow into a short function with simple inputs. `mga()` merges trimmed sequence files from each sample to identify unique ASVs for taxonomic classification, phylogenetic tree construction, and generating co-occurrence networks The output is a `mga-class` object that stores all resulting data, including a table with computed measures of community diversity, facilitating sensitivity analyses on subjective user inputs and downstream community analyses. `mga()` employs the Bioconductor workflow by (Callahan, Sankaran, et al., 2016), which leverages the fine resolution of ASV inference over OTU clustering based on fixed but arbitrary dissimilarity thresholds (Callahan et al., 2017).

`mga()` works directly with raw and filtered genetic sequences in the text-based fastq format returned from sequencing services, requiring only that file names and paths be manually curated with the format:

"FOL_DER/SAMPLENAME_XXX.fastq…". The main elements needed for the function are the raw forward and reverse sequence files (`fnFs`, `fnRs`), the filtered forward and reverse sequence reads (`filtFs`, `filtRs`), and the reference library FASTA used for taxonomic classification (`refFasta`). Tabulated sample metadata can be attached to the returned outputs using the `metadata` argument. Additional arguments can further be specified to customize analyses, such as parameters for phylogeny and co-occurrence network construction. These arguments ensure that the function is flexible, while maintaining standard default settings that produce reasonable outcomes for users unfamiliar or unconcerned with the details of specific arguments.

```
mga(fastq.Fs = fnFs,  # forward raw fastq
    fastq.Rs = fnRs,  # reverse raw fastq
    filtFs = filtFs,  # forward filtered fastq files
```

```
    filtRs = filtRs,  # reverse filtered fastq files
    refFasta = silva.138.1S,  # 16S silva reference FASTA v138.1
    metadata = meta_data)  # sample metadata (optional)
```

We used fungal and bacterial data sets to test the performance of `mga()`, along with accompanying functions that support taxonomic filtering (`filter_mga()`) and data visualization (`plot.mga()`). A sensitivity analysis was also conducted to test the package for sensitivity to choice of taxonomic reference library. These analyses were performed in `R` (v4.3.0, R Core Team, 2023) and have been adapted into vignettes, available with supplementary materials on GitHub at: https://github.com/ErikaYLin/mga_Microbial-Genetic-Analysis. The package can be installed from GitHub at: https://github.com/ErikaYLin/mga.

Materials and Methods

The `mga()` function:

The core function `mga()` was adapted from Callahan, Sankaran, et al. (2016), using `phyloseq` and `dada2` to process filtered sequence reads for graphical analyses. `mga()` begins with sequence dereplication to remove redundant information, reducing computation time (Callahan, Sankaran, et al., 2016). The DADA2 method is employed to differentiate sequencing errors from genetic variation and to independently infer ASVs, eliminating substitution and indel errors (Callahan, Sankaran, et al., 2016). Forward-reverse sequence pairs from `dada2` (v1.28.0, Callahan, McMurdie, et al., 2016) processing are paired and chimeras are removed to build a table of ASVs, then fed into `assignTaxonomy()` from the `phyloseq` package (v1.44.0, McMurdie & Holmes, 2013) for taxonomic classification, using the provided reference library input. Any unidentified taxa are labelled as unclassified following a method by Hui (2021). By default, these unclassified taxa are all numbered individually, but can be kept using the `make.unique` argument.

The `DECIPHER` (v2.28.0, Wright, 2016) package is used for multiple sequence alignment for phylogenetic tree construction. . Each initial tree is created using the neighbour-joining method with `phangorn` (v2.11.1, Schliep, 2011), however clustering and tree arrangement can be customized using `tree.args`. Here we kept the default general time-reversible model. The full list of model and rearrangement types for tree construction can be found at `?phangorn::optim.pml()`.

The ASV table, taxonomic assignment, phylogenetic tree, and sample metadata are stored in a `phyloseq-class` object for easier handling in subsequent steps. All elements in the `phyloseq` object are agglomerated by species for downstream analyses, unless otherwise specified by the `group.taxa` argument. The `phyloseq` package is employed for measures of community structure: taxonomic richness; Shannon's H; Simpson's diversity index; and phylogenetic diversity , equivalent to Faith's phylogenetic diversity. Additionally, a co-occurrence network can be generated for each sample based on a threshold ecological distance between vertices (taxa or ASVs) that determines whether vertices are more likely to co-occur or not. We used default parameters and a maximum distance threshold of 0.35 for network construction. Network diagnostics are calculated using the `igraph` (v1.4.2, Csardi et al., 2006) package: the total number of vertices (v) and edges (e), the degree of each vertex, the network connectivity (e/v), and network connectance (e/v^2, Landi et al., 2018).

Package Performance & Sensitivity Analysis:

In developing `mga`, we used both bacterial and fungal DNA sequences. To test for the package's sensitivity to different reference libraries, we evaluated `mga()` for Illumina Miseq 2x250 16S amplicon sequences accessed from Callahan, Sankaran, et al. (2016), originally collected by Schloss et al. (2012) to study how the murine gut microbiome stabilizes post-weaning. Three 16S reference FASTAs were selected from two recognized databases for bacterial metabarcoding. We compared an older Silva library (16S Silva NR v132 training set, Callahan, 2018) and an amended version (16S Silva NR99 v138.1 training set with species,

McLaren & Callahan, 2021) to assess the importance of using updated libraries, and the 16S GreenGenes v13.8 training set (Callahan, 2016).

Upon inspecting the quality profiles for the first two sequences, forward reads sustained high average quality and were trimmed at position 240; reverse read quality decreased substantially around position 160 and thus, these were trimmed at position 160 (Callahan, Sankaran, et al., 2016). The first ten base pairs were also removed to avoid potential pathological issues as discussed by Callahan, Sankaran, et al. (2016). Raw fastq files were filtered and trimmed with `dada2` for a 2 maximum expected errors per read.

The `mga()` analysis was run separately for each reference library by inputting the raw and filtered sequence files, while maintaining default settings for all other arguments. We looped over each sample , extracting and merging results with metadata. Resulting phylogenetic diversity values were compared across the three reference libraries to determine differences attributable to library choice.

```
# List for storing results
SILVA.OLD <- list()

# Loop analysis for each file
for (i in 1:length(fnFs)){

  # Assign metadata input as a list
  meta_data <- list()
  for (j in 1:length(fnFs)){

    ID <- sapply(strsplit(fnFs[i], "_"), `[`, 2)
    ID <- sapply(strsplit(ID, "/"), `[`, 2)
    meta_data[[j]] <- as.list(meta[meta$sample.ID %in% ID,])
  }

  # Create new row for each output
  SILVA.OLD[[i]] <- mga::mga(fastq.Fs = fnFs[i],
                             fastq.Rs = fnRs[i],
                             filtFs = filtFs[i],
                             filtRs = filtRs[i],
                             refFasta = silva.132,
                             metadata = meta_data[[j]])
}
```

Furthermore, as analysis runtime may concern users, the computation time for `mga()` per additional sample was measured with the `tictoc` package (v1.2.1, Izrailev, 2024).

Taxonomic Filtering & Data Visualization:

The `mga` `filter_mga()` function prunes `mga-class` objects for select taxa of interest. This function operates similarly to `mga()` with an added matching and filtering process, facilitated by the `phyloseq` `prune_taxa()` function. The function requires an `mga-class` object and vector of taxa names to filter for. Alternatively, a two-column table may be provided: a "taxon" column of taxa names and "group" column with corresponding taxonomic ranks (Table A4). We employed `filter_mga()` to identify potential pathogens within a subset of 16S rNA gene lemur (*Eulemur rubriventer*) gut microbiome data (Grieneisen et al., 2024).

```
# Empty list to store filtered objects
lemur_filter <- list()
# Filter each object in output list from `mga()`
```

```
for (i in 1:length(LEMUR)) {
  lemur_filter[[i]] <- mga::drop_taxa(LEMUR[[i]],
                                      taxa = c("Citrobacter",
                                               "Escherichia-Shigella",
                                               "Achromobacter"))
}
```

The `mga` package also includes `plot.mga()`, a wrapper for `phyloseq` plotting, which helps visualize the phylogenetic trees and co-occurrence networks produced by `mga`. We used the second sample from the Schloss et al. (2012) results to generate figures for the phylogenetic tree (Fig A6) and the co-occurrence network (Fig A7). Graphical parameters for the tree and network were adjusted using arguments from `plot_tree()` and `plot_network()` in the `phyloseq` package.

Results

Performance & Outputs:

Overall, the package performed well when tested with two bacterial amplicon data sets targeted for the 16S rRNA gene for prokaryotes. The package could similarly be used in identifying fungi and other eukaryotes , provided the appropriate reference FASTA (e.g. UNITE general FASTA release, Abarenkov et al., 2022). The computation time was proportional to the number of samples processed (Fig A4).

The output for each sample is a `mga-class` list storing all data produced during sequence processing and analysis: a frequency table of all identified sequences, including chimeras (seqtabAll); a frequency table of unique ASVs (ASVtab); the corresponding taxonomy table (taxTab); models used to reconstruct phylogeny (phylo_tree); sample metadata if provided (sampledata), the `phyloseq` object (ps) storing `phyloseq`-formatted ASV table , taxonomy table , phylogenetic tree , and sample data ; the `phyloseq` object aggregated by chosen taxonomic level (ps.taxa); a data frame of diversity and network measures merged with metadata (results.samples); a co-occurrence network if specified (net); and a data frame tabulating vertex degrees (degree.samp). Computed diversity metrics and network diagnostics include Shannon's H, Simpson's diversity index, total count of taxa grouped by taxonomic level (rich), phylogenetic diversity (summed phylogeny branch lengths), total read counts for ASVs (reads), total ASV count, total vertex and edge count, network connectivity (average degree per vertex or interactions per species), and network connectance (proportion of edges present of all possible edges) (Landi et al., 2018). Below presents an example of the `mga-class` output for a single sample.

Table 1: Objects stored in an individual `mga` object.

seqtabAll	ASVtab	taxTab	phylo_tree	sampledata
ps	ps.species	results.samples	net	degree.samp

Sensitivity to Reference Libraries:

The results produced by `mga()` were similar across reference libraries, indicating that the package was insensitive to this subjective input. Specifically, although individual measures of diversity and network topology differed between libraries, the overall conclusions drawn from diversity metrics, such as phylogenetic diversity, were consistent regardless of the library used (Fig 2). The updated Silva v138.1 library produced phylogenetic diversity values more similar to the GreenGenes v13.8 library than to the older Silva v132 library.

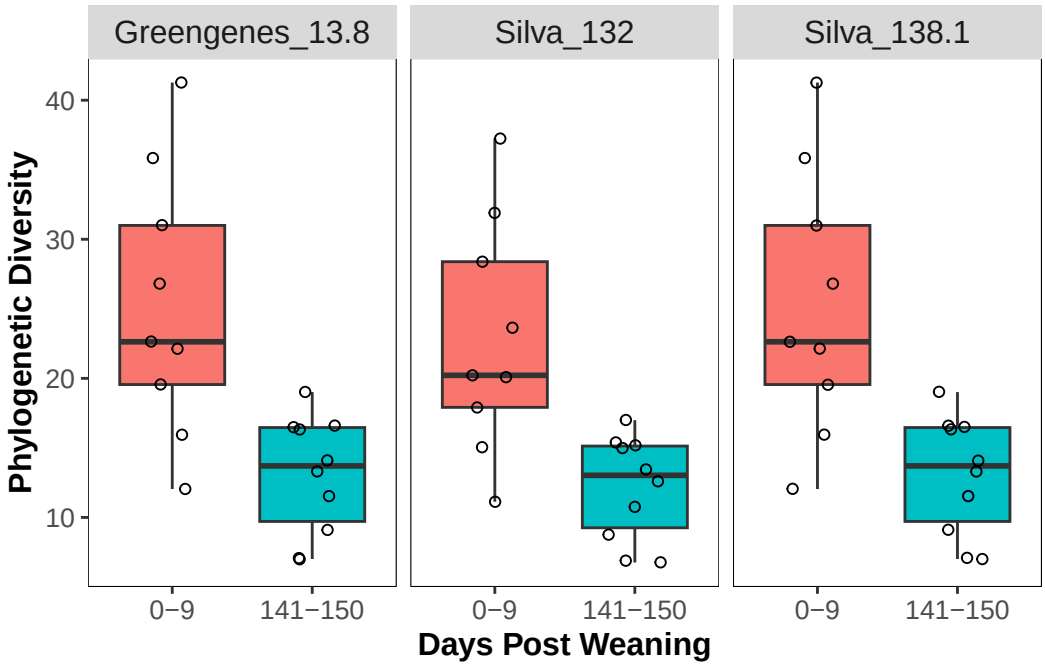

Figure 1: Boxplot of the phylogenetic diversity of murine gut bacteria immediately following weaning (0-9 days) and later (141-150 days). Results are compared across three taxonomic reference libraries.

Taxonomic Filtering & Data Visualization:

The `filter_mga()` function identified selected taxa, producing a list with an mga object pruned for pathogenic taxa and its pathogen-free complement. This allows the comparison of relative abundances of pathogenic to non-pathogenic taxa (Fig 3C), and the relative abundances across pathogenic taxa (Fig 3B). On average, *Escherichia shigella* genus, namely *E. coli*, was the most abundant pathogenic taxon found in lemur gut microbiome samples (Fig 3C).

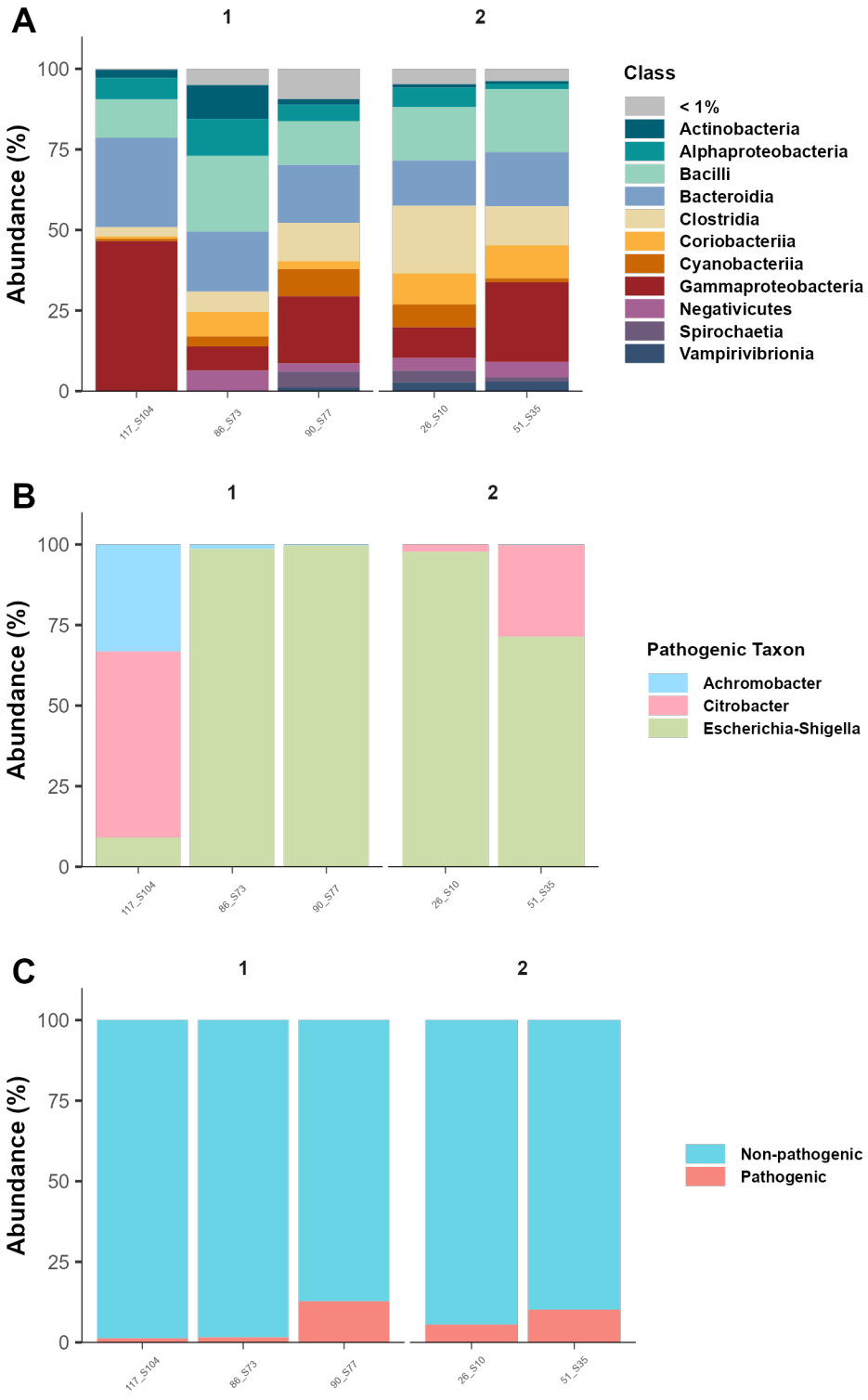

Figure 2: Bar plots comparing (A) the relative abundance of bacterial classes grouped lemur social group for five gut microbiome samples. (B) the relative abundance of identified pathogens of interest. (C) the relative abundance of pathogenic to non-pathogenic species.

Discussion

To facilitate microbial genetic analyses and sequence processing, we created `mga`, addressing the need for accessible and easily operable tools. By developing in `R`, `mga` was made available to an expanding body of biologists using command-line programs for statistical research, recognizing the importance of maintaining reproducibility and flexibility in their studies. Taking a step back from specialized but increasingly complex tools, `mga` contributes a user-friendly alternative to the pool of contemporary bioinformatics tools, addressing the three pillars of Open Science: transparency in research, accessibility and sharing, and inclusivity for researchers at all career stages (Center for Open Science, 2024).

When evaluating package functionality, available DADA2-formatted reference FASTAs, compiled by Callahan, McMurdie, et al. (2016), enabled `mga` to smoothly process 16S rRNA and ITS amplicon data. Other commonly targeted regions, such as 18S rRNA, can similarly be processed, but may require manually curating training FASTAs, as there is presently no available DADA2-formatted 18S reference library to the best of our knowledge. Beyond manual inputs, arguments can be specified to customize analyses, such as `group.taxa` for the taxonomic level at which diversity metrics are calculated and `phyloseq` objects are aggregated. Users can also nullify the argument to use individual ASVs for calculations and network construction.

The sensitivity analysis conducted on the choice of reference library and database showed comparable diversity values across libraries, indicating that the package was not sensitive to this subjective input. The findings that murine gut microbiota stabilize after weaning from initially more variable and diverse communities, were consistent with literature (Schloss et al., 2012), supporting package performance. This was desirable because conclusions should remain robust despite slight differences in reference training FASTAs across databases. We caution, however, that reference libraries update regularly due to our evolving understanding of microbial evolution. While taxonomic assignment was found to be insensitive to library choice, this does not discount the possibility for other subjective parameters in phylogenetic modelling and network construction to impact findings for specific cases and types of data. The package uses established standard settings for arguments, however, users have the capacity to perform sensitivity analyses on these arguments and parameters to ensure that reasonable results are obtained prior to follow-up work.

As presented in case studies, `mga` outputs are readily employable in downstream analyses. These include diversity and differential analyses to characterize microbial communities. Here, we have demonstrated how diversity metrics calculated as part of the `mga()` analysis, such as phylogenetic diversity, can be directly plotted to visualize preliminary results or fed into ensuing models for further investigation. All necessary components for subsequent analyses are stored together in the `mga-class` object returned by `mga()` and `filter_mga()`. The centralization of data allows for easy access and handling, such as with the package's integrated plotting function. `plot.mga()` facilitates the visualization of phylogenetic trees and co-occurrence networks, with aesthetic parameters that can be modified according to graphical arguments from the `phyloseq` `plot_tree()` and `plot_network()` functions, and `plot.phylo()` from the `ape` package [paradis2004ape]. Additionally, researchers commonly compare relative abundances within and across samples, as metabarcoding is subject to PCR amplification and sequencing bias (Krehenwinkel et al., 2017), making absolute read counts unreliable. We generated relative abundance bar plots from taxonomic classification and `filter_mga()` results (Fig3).

The `filter_mga()` functionfilters `mga` objects for taxa of interest, such as identifying putative pathogens in gut microbiota (Fig 3). Taxa can be kept or dropped, returning `mga` objects pruned for the taxa provided, with recalculated measures of diversity and network topology. As researchers may be concerned both with selected taxa and changes in community structure upon removal of selected taxa, users can return both outputs: `mga_keep` retaining selected taxa and `mga_drop` removing selected taxa. These results can be used to assess the occurrence of specific taxa within data. For the lemur gut microbiome subset, `filter_mga()` identified at least two of three pathogenic taxa per sample. Taxonomic filtering revealed that the presence of putative gut pathogens was minimal when compared to the relative abundance of non-pathogenic species identified from samples (Fig 3C).

Manual filtering and trimming of sequence files is required prior to inputting files as `mga()` arguments.

Due to variation in sequencing quality and error rates, it would be inappropriate to standardize this step of microbiome analysis. Moreover, primers should be removed to avoid issues with taxonomic classification. As such, we encourage following the workflow by Callahan, Sankaran, et al. (2016) to filter and trim sequence files based on quality profiles. As `dada2` and `phyloseq` are integral packages within `mga`, our package is able to smoothly handle any outputs and feed into functions from either package. With the integration of `phyloseq-class` objects, certain list elements produced from the core function can only be extracted using `phyloseq`-specific syntax. However, this format effectively organizes data for access to a wider range of downstream analyses, including those that make use of `phyloseq` syntax. Therefore, this limitation trades off with increased flexibility in follow-up work.

We seek to improve the efficiency of `mga()` in future updates by incorporating parallelization to reduce computation time. Moreover, while the package is publicly available on GitHub, it has undergone rigorous review to allow publication within CRAN, a community-vetted repository, which would facilitate installation from directly within `R`, recognizing that `mga` is well-designed by strict standards. Altogether, despite minor limitations, the `mga` package leverages existing `R` packages and workflows to form a comprehensive, yet easy-to-use bioinformatics tool for microbial genetic sequence processing. Not only does this package serve as a versatile, user-oriented software, helping to introduce new researchers to code-based analytical tools, the development of `mga` also acts as a call for greater accessibility in science and a framework for building user-friendly analytical software in all fields of research.

References

Abarenkov, K., Zirk, A., Piirman, T., Pöhönen, R., Ivanov, F., Nilsson, R. H., & Kõljalg, U. (2022). *UNITE general FASTA release for fungi*. UNITE Community. https://doi.org/10.15156/BIO/2483911

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C. C., Al-Ghalith, G. A., Alexander, H., Alm, E. J., Arumugam, M., Asnicar, F., et al. (2019). Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nature Biotechnology*, *37*(8), 852–857. https://doi.org/10.1038/s41587-019-0209-9

Caldara, M., Belgiovine, C., Secchi, E., & Rusconi, R. (2022). Environmental, microbiological, and immunological features of bacterial biofilms associated with implanted medical devices. *Clinical Microbiology Reviews*, *35*(2), e00221–20. https://doi.org/10.1128/cmr.00221-20

Callahan, B. J. (2016). The RDP and GreenGenes taxonomic training sets formatted for DADA2 [data set]. *Zenodo*, *158955*. https://doi.org/10.5281/zenodo.158955

Callahan, B. J. (2018). Silva taxonomic training data formatted for DADA2 (silva version 132) [data set]. *Zenodo*. https://doi.org/10.5281/zenodo.1172783

Callahan, B. J., McMurdie, P. J., & Holmes, S. P. (2017). Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *The ISME Journal*, *11*(12), 2639–2643. https://doi.org/10.1038/ismej.2017.119

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., & Holmes, S. P. (2016). DADA2: High-resolution sample inference from illumina amplicon data. *Nature Methods*, *13*(7), 581–583. https://doi.org/10.1038/nmeth.3869

Callahan, B. J., Sankaran, K., Fukuyama, J. A., McMurdie, P. J., & Holmes, S. P. (2016). Bioconductor workflow for microbiome data analysis: From raw reads to community analyses. *F1000Research*, *5*, 1492. https://doi.org/10.12688/f1000research.8986.2

Castañeda Alejo, S. M., Tejada Meza, K., Valderrama Valencia, M. R., Arenazas Rodríguez, A. J., & Málaga Espinoza, C. J. (2022). Tire ground rubber biodegradation by a consortium isolated from an aged tire. *Microorganisms*, *10*(7), 1414. https://doi.org/10.3390/microorganisms10071414

Center for Open Science. (2024). *Open Science*. https://www.cos.io/open-science; Center for Open Science.

Csardi, G., Nepusz, T., Traag, V., Horvát, S., Zanini, F., Noom, D., & Müller, K. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*, 1–9. https://doi.org/10.5281/zenodo.7682609

Grieneisen, L., Blekhman, R., & Tecot, S. (2024). *Temporal patterns of gut microbiota in lemurs (Eulemur rubriventer) living in intact and disturbed habitats.* [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.11105896

Hui, Y. (2021). *Tutorial for microbiome analysis in r*. https://www.yanh.org/2021/01/01/microbiome-r/; Yan Hui.

Izrailev, S. (2024). *Tictoc: Functions for timing r scripts, as well as implementations of "stack" and "StackList" structures*. https://cran.r-project.org/package=tictoc; CRAN.

Krehenwinkel, H., Wolf, M., Lim, J. Y., Rominger, A. J., Simison, W. B., & Gillespie, R. G. (2017). Estimating and mitigating amplification bias in qualitative and quantitative arthropod metabarcoding. *Scientific Reports*, *7*(1), 17668. https://doi.org/10.1038/s41598-017-17333-x

Landi, P., Minoarivelo, H. O., Brännström, Å., Hui, C., & Dieckmann, U. (2018). Complexity and stability of ecological networks: A review of the theory. *Population Ecology*, *60*(4), 319–345. https://doi.org/10.1007/s10144-018-0628-3

McLaren, M. R., & Callahan, B. J. (2021). Silva 138.1 prokaryotic SSU taxonomic training data formatted for DADA2 [data set]. *Zenodo*. https://doi.org/10.5281/zenodo.4587955

McLean, G., Kamil, J., Lee, B., Moore, P., Schulz, T. F., Muik, A., Sahin, U., Türeci, Ö., & Pather, S. (2022). The impact of evolving SARS-CoV-2 mutations and variants on COVID-19 vaccines. *MBio*, *13*(2), e02979–21. https://doi.org/10.1128/mbio.02979-21

McMurdie, P. J., & Holmes, S. (2013). Phyloseq: An r package for reproducible interactive analysis and graphics of microbiome census data. *PloS One*, *8*(4), e61217. https://doi.org/10.1371/journal.pone.0061217

R Core Team. (2023). *R: A language and environment for statistical computing (4.3.0)*. https://www.R-project.org; R Foundation for Statistical Computing.

Rotoni, C., Leite, M. F., Pijl, A., & Kuramae, E. E. (2022). Rhizosphere microbiome response to host genetic variability: A trade-off between bacterial and fungal community assembly. *FEMS Microbiology Ecology*, *98*(6), fiac061. https://doi.org/10.1093/femsec/fiac061

Schliep, K. P. (2011). Phangorn: Phylogenetic analysis in r. *Bioinformatics*, *27*(4), 592–593. https://doi.org/10.1093/bioinformatics/btq706

Schloss, P. D., Schubert, A. M., Zackular, J. P., Iverson, K. D., Young, V. B., & Petrosino, J. F. (2012). Stabilization of the murine gut microbiome following weaning. *Gut Microbes*, *3*(4), 383–393. https://doi.org/10.4161/gmic.21008

Schloss, P. D., Westcott, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., Lesniewski, R. A., Oakley, B. B., Parks, D. H., Robinson, C. J., et al. (2009). Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, *75*(23), 7537–7541. https://doi.org/10.1128/AEM.01541-09

Shade, A., Peter, H., Allison, S. D., Baho, D. L., Berga, M., Bürgmann, H., Huber, D. H., Langenheder, S., Lennon, J. T., Martiny, J. B., et al. (2012). Fundamentals of microbial community resistance and resilience. *Frontiers in Microbiology*, *3*, 417. https://doi.org/10.3389/fmicb.2012.00417

Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., & Gordon, J. I. (2007). The human microbiome project. *Nature*, *449*(7164), 804–810. https://doi.org/10.1038/nature06244

Ward, R. D., Stajich, J. E., Johansen, J. R., Huntemann, M., Clum, A., Foster, B., Foster, B., Roux, S., Palaniappan, K., Varghese, N., et al. (2021). Metagenome sequencing to explore phylogenomics of terrestrial cyanobacteria. *Microbiology Resource Announcements*, *10*(22), 10–1128. https://doi.org/10.1128/MRA.00258-21

Wen, T., Niu, G., Chen, T., Shen, Q., Yuan, J., & Liu, Y.-X. (2023). The best practice for microbiome analysis using r. *Protein & Cell*, *14*(10), 713–725. https://doi.org/10.1093/procel/pwad024

Wright, E. S. (2016). Using DECIPHER v2. 0 to analyze big biological sequence data in r. *R Journal*, *8*(1). https://doi.org/10.18129/B9.bioc.DECIPHER

Yang, J., Long, H., Hu, Y., Feng, Y., McNally, A., & Zong, Z. (2022). Klebsiella oxytoca complex: Update on taxonomy, antimicrobial resistance, and virulence. *Clinical Microbiology Reviews*, *35*(1), e00006–21. https://doi.org/10.1128/CMR.00006-21