mbctools: A User-Friendly Metabarcoding and Cross-Platform Pipeline for Analyzing Multiple Amplicon Sequencing Data across a Large Diversity of Organisms

Christian Barnabé[a†], Guilhem Sempéré[bc†*], Vincent Manzanilla[*], Joel Moo Millan[d] , Antoine Amblard-Rambert[d] and Etienne Waleckx[ad]

[a]Institut de Recherche pour le Développement, Université de Montpellier, UMR INTERTRYP IRD, CIRAD, University of Montpellier, France.

[b]Centre de Coopération Internationale en Recherche Agronomique pour le Développement, UMR INTERTRYP IRD, CIRAD, Campus international de Baillarguet, Montpellier, France

[c]South Green Bioinformatics Platform, Biodiversity, Montpellier, France

[d]Laboratorio de Parasitología, Centro de Investigaciones Regionales “Dr Hideyo Noguchi”, Universidad Autónoma de Yucatán, Mérida, México

†Co–first authors

*Corresponding authors: guilhem.sempere@cirad.fr, vincent.manzanilla@ird.fr

Abstract

We developed a python package called *mbctools*, designed to offer a cross-platform tool for processing amplicon data from various organisms in the context of metabarcoding studies. It can handle the most common tasks in metabarcoding pipelines like paired-end merging, primer trimming, quality filtering, zero-radius operational taxonomic units (ZOTU) denoising and filtering. This pipeline has the capability to process multiple genetic markers simultaneously. *mbctools* is a menu-driven program that eliminates the need for expertise in command-line skills and ensures documentation of each analysis for reproducibility purposes. In our workflow, VSEARCH is utilized for processing *fastq* files derived from amplicon data. To adapt to the diversity of projects in metabarcoding, we facilitate the reprocessing of datasets with the possibility to adjust parameters. *mbctools* can also be launched in headless mode, making it suited for integration into pipelines running on High-Performance Computing (HPC) environments. *mbctools* is available at: https://github.com/GuilhemSempere/mbctools, https://pypi.org/project/mbctools/.

Keywords: metabarcoding, amplicon, reproducibility, VSEARCH, pipeline, mbctools

Introduction

Over the past decade, environmental DNA (eDNA) metabarcoding has emerged as a robust and increasingly popular approach. Its simplicity of implementation enables biologists, local governments, and NGOs to increase their understanding of the spatial and temporal ecological networks (1–3). As illustrated in Figure 1, this method consists in identifying a targeted subset of genomes within bulk samples by massive sequencing of amplicons of taxonomically informative genetic markers, referred to as loci in *mbctools* (4,5). This approach has proven to be useful for diet profile analysis, air and water quality monitoring, biodiversity monitoring, and food quality control, beverage or ancient ecosystem composition (6–8). In our recent studies, we have employed this approach to investigate the transmission cycles of vector-borne diseases, using the midgut contents of vector insects as bulk samples, and simultaneously sequencing markers allowing for the identification of their genetic diversity, their diet composition (blood meal sources), their midgut microbiome composition, and pathogen diversity, as all these components and their interactions shape transmission cycles (9,10).

The increasing demand for efficient and versatile pipelines to analyze amplicon data has led to the development of over 30 programs and pipelines in recent years (11). However, these applications often present some challenges in terms of reproducibility and user-friendliness for non-expert users in the field of bioinformatics. In contrast, *mbctools* has been designed with a simplified interface, allowing scientists without advanced technical expertise in bioinformatics to easily install, navigate and utilize the pipeline. This accessibility ensures that non-expert users can effectively and autonomously analyze their data without experiencing difficulties in reproducibility or requiring extensive bioinformatics knowledge.

Among pipelines and algorithms used in the field of metabarcoding data processing, substantial variations in sensitivity and specificity are observed (12,13). From these benchmarking studies, VSEARCH appears appropriate for our applications. *mbctools*' main purpose is to facilitate the use of VSEARCH, which has demonstrated strong performance in handling large-scale next-generation sequencing (NGS) data (14), and in some aspects could be considered a "wrapper", hiding the complexity of command-line to the users. It thus allows them to easily process sequencing reads and obtain meaningful outputs, i.e, zero-radius operational taxonomic units (ZOTUs, (15), almost equivalent to an amplicon sequence variant, ASV)) along with abundance information.

mbctools process demultiplexed metabarcoding data obtained from one or multiple genetic markers to distribute cleaned and filtered sequences into ZOTUs sorted per marker, which can be used for taxonomic assignment or further analyzes such as phylogeny (Figure 1, 2). The software can be used only with command lines making it compatible with high-performance computing (HPC) job schedulers.

Prerequisites and dependencies

Input data types - sequencing tech & lab

mbctools enables the analysis of amplicon data from multiple markers, obtained through NGS sequencing. It can handle both single-end reads and paired-end reads, provided that the latter can be merged. In the eventuality where the merging of paired-end reads is prevented by either large amplicon lengths (leading to insufficient overlap zone) or the presence of low complexity sequences, *mbctools* is able to utilize only forward reads for analysis. The pipeline can accommodate a potentially unlimited number of samples and amplicons per sample.

Running environment

mbctools is a versatile cross-platform python package running seamlessly on Windows, Unix, or MacOS systems. The installation of Python3.7 (or higher) and VSEARCH (version 2.19.0 or higher) (14) is required to run the pipeline. Additionally, for Windows users, Powershell script execution must be enabled. The package, available from the Python Package Index (PyPI) (https://pypi.org/project/mbctools/), can be installed and upgraded using the command line tool pip (*pip install mbctools*).

Before starting the analyses, the package requires mandatory directories and files to be pre-arranged as illustrated in Figure 3:

- A default directory, named 'fastq', should contain all the fastq files corresponding to the R1 (and R2, when working with paired-ends sequencing) reads. These files are expected to be demultiplexed, meaning that each fastq file should contain reads from a single sample.
- A default directory named ‘refs’ that contains one reference file per genetic marker in the multi-FASTA format. Each of them should contain one or more unaligned reference sequences, from the same strand and without gaps. Providing multiple reference sequences for a given marker improves diversity support and thus leads to a better assignment accuracy. In addition, the ‘read orientation fixing’ step runs faster when using several reference sequences. Filenames must end with the extension '.fas' and must be called by the name of the corresponding locus (Figure 3).
- ‘lociPE.txt’ provides the list of markers (one per line) corresponding to paired-end R1/R2 reads.
- ‘lociSE.txt’ provides the list of loci (one per line) corresponding to single-end reads and for which only R1 reads will be analyzed.

- ‘samples.txt’ includes a list of sample names, where each name appears on a separate line.
- ‘*fastq*R1.txt’ contains the names of the fastq R1 files corresponding to the samples.
- ‘*fastq*R2.txt’ contains the names of the fastq R2 files corresponding to the samples.

The contents of the last three files must be sorted in the same order, i.e. line (i) of 'samples.txt' shall correspond to line (i) of both ‘*fastq*R1.txt’ and line (i) of ‘*fastq*R2.txt’.

mbctools shall be launched from within the working directory in a shell console, typically by typing *mbctools* if installed as a package.

Software features

Interface overview

The program presents a series of menus, each containing two to four options as illustrated in Figures 4 to 8. On startup, the program displays the main menu shown in Figure 4. This menu presents descriptive keywords that enable users to easily navigate through the available menus and sections. From each menu, the user is required to select one single option at a time.

To analyze a new dataset, it is necessary to perform a complete initial analysis by selecting the first option from both the first and second menu. The *initial analysis* processes the data through a succession of steps such as merging, sample-level dereplication, sequence denoising, chimera detection and removal, affiliation of sequences to loci, and sequence re-orientation (menu 1, Figure 5). Once those are completed, primers are removed, and the denoising process could be refined (menu 2, Figure 6). Finally, centroid sequences associated with each locus can be exported to fasta files (menu 3, Figure 7). The above represents the complete cycle of data processing. Additionally (menu 4, Figure 8), *mbctools* software offers

the possibility to export results in a format that is compatible with the metaXplor platform (16). The entire process is described in more detail below.

When option 1 (*initial analysis*) is selected from menu 1, the menu shown on Figure 5 allows to run the *initial analysis* (section *1*), namely read, merging, sample-level dereplication, sequence clustering (denoising), chimera detection, affiliation of sequences to loci, and sequence re-orientation. In the same menu, four other sections (*1a, 1b, 1c, 1d*) provide means to re-process some particular loci and/or samples with adjusted parameters. Option *1e* may be used to assess the read quality for each sample, and write results to files with suffix “*.quality.txt” file in the *outputs* directory.

When the initial analysis is completed, menu 2 should be used to remove primer sequences, and to exclude denoised sequences with an abundance considered too low for being relevant (Figure 6). This operation may be launched either in batch mode, or sample by sample in case different thresholds need to be applied.

In menu 3, the program groups sample files for each locus, creating a new file per locus with the suffix "*_allseq_select.fasta*" (Figure 7). During this process, *mbctools* can, as an option, dereplicate sequences at the project level, ensuring that each entry in the fasta file is unique. In such cases, an additional tabulated file is generated to maintain a record of sequence abundance across the samples. Optionally generated files are then named with the following suffixes: “*_allseq_select_derep.fasta”, “*_allseq_select_derep.tsv”.

Finally, menu 4 converts *mbctools* outputs into a format suitable for importing into metaXplor (Figure 8), a web-oriented platform for storing, sharing, exploring and manipulating metagenomic datasets. metaXplor significantly contributes to enhancing the FAIR (Findable, Accessible, Interoperable, and Reusable)(17) characteristics of this specific data type by

serving as a robust solution for centralized data management, intuitive visualization, and long-term accessibility.

Main pipeline description

From fastq files to relevant amplicon sequences (menu 1)

The filenames in the examples given in this section follow a number of conventions:

- *occurrences of the string "(paired|single)" actually represent either "paired" or "single" depending on the selection made by the user*
- *occurrences of the character string "(L1|L2)" actually represent the name of the locus selected by the user*
- *occurrences of the character string "(Sample1|Sample2)" actually represent the name of the sample selected by the user*
- *dir_fastq: directory containing the fastq files*
- *fastqR1: name of the fastq file containing the R1 reads*
- *fastqR2: name of the fastq file containing the R2 reads*

The workflow that applies to the sequencing data is detailed below in seven steps:

Step 1: conversion of R1 / single-end fastq file into a fasta file.

Using VSEARCH, this command converts the single-end fastq file into a multi-fasta file.

Underlying VSEARCH command:

vsearch --fastq_filter dir_fastq/fastaqR1 –fastaout ../tmp_files/(Sample1|Sample2)_singleEnd.fa

Step 2: merging

When the length of the expected amplicon allows a sufficient overlap, this step merges R1/R2 paired-end *fastq* files into a *fasta* file. When the merge reads are lower than 300bp, it removes the read tails (*--fastq_allowmergestagger*).

Underlying VSEARCH command:

vsearch --fastq_mergepairs dir_fastq/fastqR1 --reverse dir_fastq/fastqR2 --fastaout ./tmp_files/(Sample1|Sample2)_pairedEnd.fa --fastq_allowmergestagger --relabel sample=(Sample1|Sample2)_merged.

Step 3: dereplication

This step aggregates identical sequences into a unique sequence from both strands for R1/R2 paired-end merged sequences and single-end R1 sequences (if it applies) into a *fasta* file where we keep track of sequence abundance in the header.

Underlying VSEARCH command:

vsearch --fastx_uniques ../tmp_files/(Sample1|Sample2)_(paired|single)End.fa --fastaout ../tmp_files/(Sample1|Sample2)_(paired|single)End_derep.fas --sizeout --strand both

Step 4: Denoising

Denoising of amplicons following the UNOISE version3 algorithm by Robert Edgar (15), and clustering similar sequences around a centroid (i.e. the sequence with higher abundance in its cluster).

Underlying VSEARCH command:

vsearch --cluster_unoise ../tmp_files/(Sample1|Sample2)_(paired|single)End_derep.fas --sizein --centroids ../tmp_files/(Sample1|Sample2)_(paired|single)End_cluster.fas --strand both --minsize int --sizeout --unoise_alph int --minseqlength int

Step 5: chimera detection and removal

Detection and removal of potential chimeras in denoised sequences, using the UCHIME2 algorithm (18).

Underlying VSEARCH command:

vsearch --uchime3_dsenovo ../tmp_files/(Sample1|Sample2)_(paired|single)End_cluster.fas --nonchimeras ../tmp_files/(Sample1|Sample2)_(paired|single)End_cluster_OK.fas

Step 6: affiliation

To assign each denoised sequence to a specific investigated marker, a comparison is conducted between reference sequences of the targeted markers (specified by the "--db" argument) and the cluster-centroid sequences from each sample. This comparison is performed using a global pairwise alignment method, allowing to determine the affiliation of each denoised sequence to its respective marker.

Underlying VSEARCH command:

vsearch --usearch_global ../tmp_files/(Sample1|Sample2)_(paired|single)End_cluster_OK.fas --db ../refs/(L1|L2).fas --matched ../loci/(L1|L2)/(Sample1|Sample2)_(paired|single)End.fas --id 70 --strand both

Step 7: orientation

When necessary, the orientation of remaining sequences is corrected to be the same as the reference sequences.

Underlying VSEARCH command:

vsearch --orient ../loci/(L1|L2)/(Sample1|Sample2)_(paired|single)End.fas --db ../refs/(L1|L2).fas –fastaout ../loci/(L1|L2)/(Sample1|Sample2)_(paired|single)End_orient.fas

The essential inputs for options under menu 1 involve directing the system to all the required files and supplying a set of parameters (some of which having default values), to customize

the processing steps performed by VSEARCH. These inputs are prompted from the shell environment.

- Full path of the folder containing *fastq* files (default, *./fastq*) where "." refers to the current directory
- Name of the file listing R1 *fastq* (plain, or gzipped if installed VSEARCH version supports it) filenames (default, *fastqR1.txt*). *NB: file lists are all expected to provide one element per line*
- Name of the file listing R2 *fastq* (plain, or gzipped if installed VSEARCH version supports it) filenames (default, *fastqR2.txt*).
  NB: Target file must exist but may be empty to process only single-end or unmerged R1 reads
- Name of the file listing the names of the loci expected to be found in paired-end reads (default, *lociPE.txt*).
  NB: Target file must exist but may be empty to process only single-end or unmerged R1 reads
- Name of the file listing the names of the loci expected to be found in single-end reads (default, *lociSE.txt*)
  NB: Target file must exist but may be empty to process only merged paired-end reads
- Name of the file listing sample names (default, *samples.txt*)
- Minimum abundance (VSEARCH parameter named *--minsize)* for retaining denoised sequences (default, *8*)
- Minimum length (VSEARCH parameter named *--minseqlength*) of retained sequences for any locus (default, *100*)

- Value for the alpha parameter (VSEARCH parameter named *--unoise_alpha*) used in the UNOISE [REF] clustering algorithm (default, *2*)
- Identity percentage parameter (VSEARCH parameter named *--id*) to match the denoised sequences against local references (default, *70%*)

Note: Options 1a to 1d allow refining the 4 last parameters (the ones used for tuning VSEARCH behavior) in order to apply particular settings on all or some of the samples and loci. Previously dereplicated sequences are then used again for re-processing selected data from the denoising/clustering step #4.

All the specified parameters and their values are automatically stored in the following configuration file: *outputs/parameters_option_1.cfg*. Once this file has been created, it can be stored separately and reused to initiate the main analysis directly, eliminating the need to re-enter each parameter individually. This can be accomplished as follows by passing its full path as an argument to mbctools, as follows: *mbctools outputs/parameters_option_1.cfg*. Using the software in this manner transforms it into a purely command-line tool, eliminating the necessity for direct interaction with the interface and facilitating its integration into HPC pipelines.

Primer removal and abundance filtering (menu 2)

This section aims at cleaning up amplicon sequences by removing primers from their ends, and ignoring sequences whose abundance is considered too rare to be relevant. The following user inputs are requested for options under menu 2:

- Locus to work with (no default, to choose among those featured in the project)
- Length of both right and left primers to trim from the final selected centroid sequences (no default)

- Minimum abundance threshold to use for the selected locus, e.g. entering 5 will ignore denoised sequences covering under 5% of the sample's total reads
- The name of the sample to work with (only applies to options 2c and 2d since previous ones process all samples)

Compiling selected sequences by locus (menu 3)

Once option 2 has been executed, users may want to gather all samples' relevant ZOTUsinto one file per locus, typically for further analyses such as phylogeny.

The only expected user input is then the locus for which to group all samples' selected sequences into a single multi-fasta file called *locusX_allseq_select.fasta*.

Execution outputs

During the execution of basic analyses, several directories are automatically generated. These include the "loci" directory, which consists of subdirectories named after the corresponding reference files for each locus. Another directory named "scripts" is created to store intermediate VSEARCH invocation scripts specific to the current project, generated based on the selected parameters. Additionally, a directory named "tmp_files" is generated to hold all intermediate temporary files produced during various procedures. Lastly, an "outputs" directory is created, containing quality checking statistics, execution logs, and settings files related to the tasks initiated from the different menus.

For each sample, up to six output files generated in the corresponding subfolders of the *loci* directory:

- *(Sample1|Sample2)_(paired|single)End.fas* corresponding to outputs of options from menu 1,

- *(Sample1|Sample2)_(paired|single)End_orient.fas*, equivalent of the latter but with all sequences oriented in the same strand,
- *(Sample1|Sample2)_paired(single)End_select.fas*, corresponding to sequences retained after applying a particular abundance threshold (options from menu 2).

When running concatenation option (#3) the above-mentioned additional file called *locusX_allseq_select.fasta* is generated for each locus, grouping all finally retained sequences for that locus, and which can be used directly in phylogenetic programs.

Exporting analysis results into metaXplor format

Menu 4 guides users into generating a zip archive ready to import into metaXplor:

- option 4a dereplicates centroid sequences at the project level, keeping track of sample abundance via a .tsv file
- option 4b, providing that BLASTn (16,19) has been run appropriately on the above, converts its results into a metaXplor assignment file
- option 4c takes a tabulated sample metadata file and turns it into the appropriate format
- option 4d compresses files generated from previous steps into a single zip archive, ready for import into metaXplor

While processing data, the system keeps track of VSEARCH execution statistics, logging information and configuration parameters in the *outputs* folder, in files named after the executed procedure (e.g. for the main analysis #1, *Stats_option_1.txt*, , *res1.log* and *parameters_option_1.cfg* respectively).

Test data

The test data were generated in a study on Chagas disease, a zoonotic parasitic illness caused by *Trypanosoma cruzi*. This disease is primarily transmitted to mammals through the feces and urine of triatomines, commonly referred to as kissing bugs (Hemiptera: Reduviidae). *Triatoma dimidiata* is the main vector in the Yucatan Peninsula in Mexico. We used a metabarcoding approach to simultaneously identify the blood meals of the vector in a sylvatic ecotope, characterize the gut microbiome composition, and assess the genetic diversity of *T. cruz*i. In 2019, the owner of an isolated rural dwelling collected thirty-three samples of T. dimidiata individuals from a conserved sylvatic site within the Balam Kú State Reserve, located in Campeche, Mexico (Figure 9). We extracted the DNA and amplified the samples of *T. dimidiata* using six amplicons (12S, 16S, COII, GPI, ITS2, ND1, MINEX). Then we combined and sequenced the amplicons on an Illumina MiSeq. On average, we obtained 222,244 reads per sample. The reads were processed with the *mbctools* package and metaXplor web interface (16). In *mbctools*, we retained the reads with a minimum length of 100bp, clusters with a minimum size of eight and used an alpha value of two. The taxonomic identification of the ZOTU was performed using blast (20). The exported CSV file with the metadata served as input for metaXplor. The analysis is directly available on the web application under the dataset 'mbctools_review' (21) (Figure 9).

We identified an average of 33 ZOTUs per sample, and only 9.61% of the reads were discarded during the analysis (Table 1). After analyses we found thirteen blood-meal sources (the most prevalent being *Homo sapiens*, *Sus scrofa*, *Mus musculus*), six bacterial Orders (Bacillales, Oceanospirillales and Micrococcales being the most abundant), two *T. cruzi* DTUs (TcI and TcIV), as well as two *T. dimidiata* genetic subgroups (ITS-2 Haplogroups 2 and 3) (Figure 9). The test data input and output files can be accessed at the following DOI address on DataSuds: https://doi.org/10.23708/W3TODJ.

Conclusion

mbctools facilitates the processing of amplicon datasets for metabarcoding studies by having menus that guide the users through various steps of the data analysis. *mbctools* offers the possibility to reprocess specific loci and samples with customized parameters, providing the possibility to adapt to the diversity of metabarcoding datasets. In the end, it generates *fasta* files composed of the filtered ZOTU ready for the taxonomic assignment step. Given the accuracy variations that can occur during this particular step, we leave the user to customize the taxonomic assignment step (12). To increase the data management, the visualization, and the long-term accessibility, *mbctools* offers a file conversion compatible with the metaXplor web application (16). Although we initially developed this pipeline for reproducibility purposes across our teams about infectious diseases, the pipeline's strength resides in its capacity to handle diverse data and be tailored to different research needs. *mbctools* is currently under development and can be adjusted to improve its applicability.

Authors' contributions

C.B and G.S designed and conceived the software based on E.W.'s specifications. V.M. did the Python packaging. E.W, G.S, V.M and C.B wrote the article. A.A-R. provided the data to test mbctools. All authors read and approved the final manuscript.

Funding

This work received financial support from CONACYT (National Council of Science and Technology, Mexico) Basic Science (Project ID: CB2015-258752) and National Problems (Project ID: PN2015-893) grants awarded to EW, as well as from IRD (French National Research Institute for Sustainable Development).

Data, scripts, code, and supplementary information availability

mbctools is freely accessible on github https://github.com/GuilhemSempere/mbctools and on PyPI https://pypi.org/project/mbctools/.

Conflict of interest disclosure

The authors declare that they comply with the PCI rule of having no financial conflicts of interest in relation to the content of the article.

References

1. Batovska J, Lynch S, Cogan N, Brown K, Darbro J, Kho E, et al. Effective mosquito and arbovirus surveillance using metabarcoding. Mol Ecol Resour. 2018;18(1):32–40.
2. Holdaway RJ, Wood JR, Dickie IA, Orwin KH, Bellingham PJ, Richardson SJ, et al. Using DNA metabarcoding to assess New Zealand's terrestrial biodiversity. N Z J Ecol. 2017;41(2):251–62.
3. Compson ZG, McClenaghan B, Singer GA, Fahner NA, Hajibabaei M. Metabarcoding from microbes to mammals: comprehensive bioassessment on a global scale. Front Ecol Evol. 2020;8:581835.
4. Taberlet P, Coissac E, Pompanon F, Brochmann C, Willerslev E. Towards next-generation biodiversity assessment using DNA metabarcoding. Mol Ecol. 2012 Apr;21(8):2045–50.
5. Cristescu ME. From barcoding single individuals to metabarcoding biological communities: towards an integrative approach to the study of global biodiversity. Trends Ecol Evol. 2014;29(10):566–71.
6. Thomsen PF, Willerslev E. Environmental DNA–An emerging tool in conservation for monitoring past and present biodiversity. Biol Conserv. 2015;183:4–18.
7. Leray M, Knowlton N. DNA barcoding and metabarcoding of standardized samples reveal patterns of marine benthic diversity. Proc Natl Acad Sci. 2015;112(7):2076–81.
8. Raclariu AC, Paltinean R, Vlase L, Labarre A, Manzanilla V, Ichim MC, et al. Comparative authentication of *Hypericum perforatum* herbal products using DNA metabarcoding, TLC and HPLC-MS. Sci Rep. 2017;7(1):1291.
9. Dumonteil E, Ramirez-Sierra MJ, Pérez-Carrillo S, Teh-Poot C, Herrera C, Gourbière S, et al. Detailed ecological associations of triatomines revealed by metabarcoding and next-generation sequencing: implications for triatomine behavior and Trypanosoma cruzi transmission cycles. Sci Rep. 2018;8(1):4140.
10. Hernández-Andrade A, Moo-Millan J, Cigarroa-Toledo N, Ramos-Ligonio A, Herrera C, Bucheton B, et al. Metabarcoding: a powerful yet still underestimated approach for the comprehensive study of vector-borne pathogen transmission cycles and their dynamics. In: Vector-Borne Diseases-Recent Developments in Epidemiology and Control. IntechOpen; 2019.
11. Hakimzadeh A, Asbun AA, Albanese D, Bernard M, Buchner D, Callahan B, et al. A pile of pipelines: an overview of the bioinformatics software for metabarcoding data analyses. 2023;
12. Mathon L, Valentini A, Guérin P, Normandeau E, Noel C, Lionnet C, et al. Benchmarking bioinformatic tools for fast and accurate eDNA metabarcoding species identification. Mol Ecol Resour. 2021;21(7):2565–79.
13. Bailet B, Apothéloz-Perret-Gentil L, Baričević A, Chonova T, Franc A, Frigerio JM, et al. Diatom DNA metabarcoding for ecological assessment: Comparison among bioinformatics pipelines used in six European countries reveals the need for standardization. Sci Total Environ. 2020;745:140948.
14. Rognes T, Flouri T, Nichols B, Quince C, Mahe F. VSEARCH: a versatile open source tool for metagenomics. PeerJ. 2016/10/27 ed. 2016;4:e2584.
15. Edgar RC. UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing. BioRxiv. 2016;081257.
16. Sempéré G, Pétel A, Abbé M, Lefeuvre P, Roumagnac P, Mahé F, et al. metaXplor: an interactive viral and microbial metagenomic data manager. GigaScience. 2021;10(2):giab001.
17. Wilkinson MD, Dumontier M, Aalbersberg IjJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data.

2016;3(1):1–9.
18. Edgar RC. UCHIME2: improved chimera prediction for amplicon sequencing. BioRxiv. 2016;074252.
19. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
20. Johnson M, Zaretskaya I, Raytselis Y, Merezhuk Y, McGinnis S, Madden TL. NCBI BLAST: a better web interface. Nucleic Acids Res. 2008;36(suppl_2):W5–9.
21. metaXplor: an interactive viral and microbial metagenomic data manager [Internet]. 2024 [cited 2024 Feb 5]. metaXplor: mbctools data set. Available from: https://metaxplor.cirad.fr/metaXplor/main.jsp?module=mbctools_review

Figures and Tables

Figure 1: Metabarcoding workflow example and area of operation of mbctools

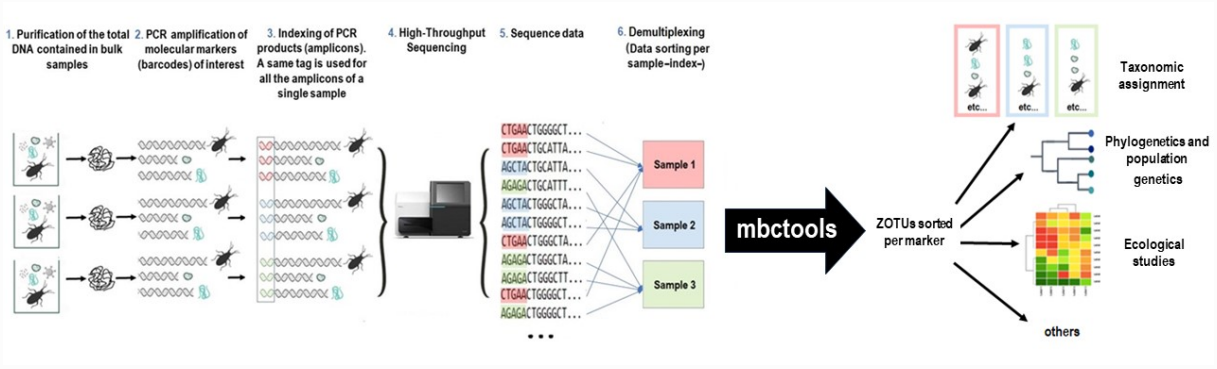


Figure 2: Overview of the *mbctools* workflow, where each station icon represents a distinct step in the process. In the initial analysis (1), data processing begins with the conversion of single-end reads from fastq to fasta (Step 1) and the merging for paired-end reads (Step 2). This is followed by dereplication (Step 3), denoising (Step 4), chimera removal (Step 5), sequence affiliation to loci (Step 6), and finally, orientation (Step 7). Step 1, denoising, can be re-run with various options (1a, 1b, 1c, 1d). Following this, there is the primer removal and denoised sequence abundance filtering (Step 2). The last step in the analysis involves obtaining per-locus fasta files containing retained ZOTUs (Step 3). The menu labeled as "4" comes into play after taxonomic assignment, preparing the file for metaXplor for visualization and ensuring long-term data accessibility.

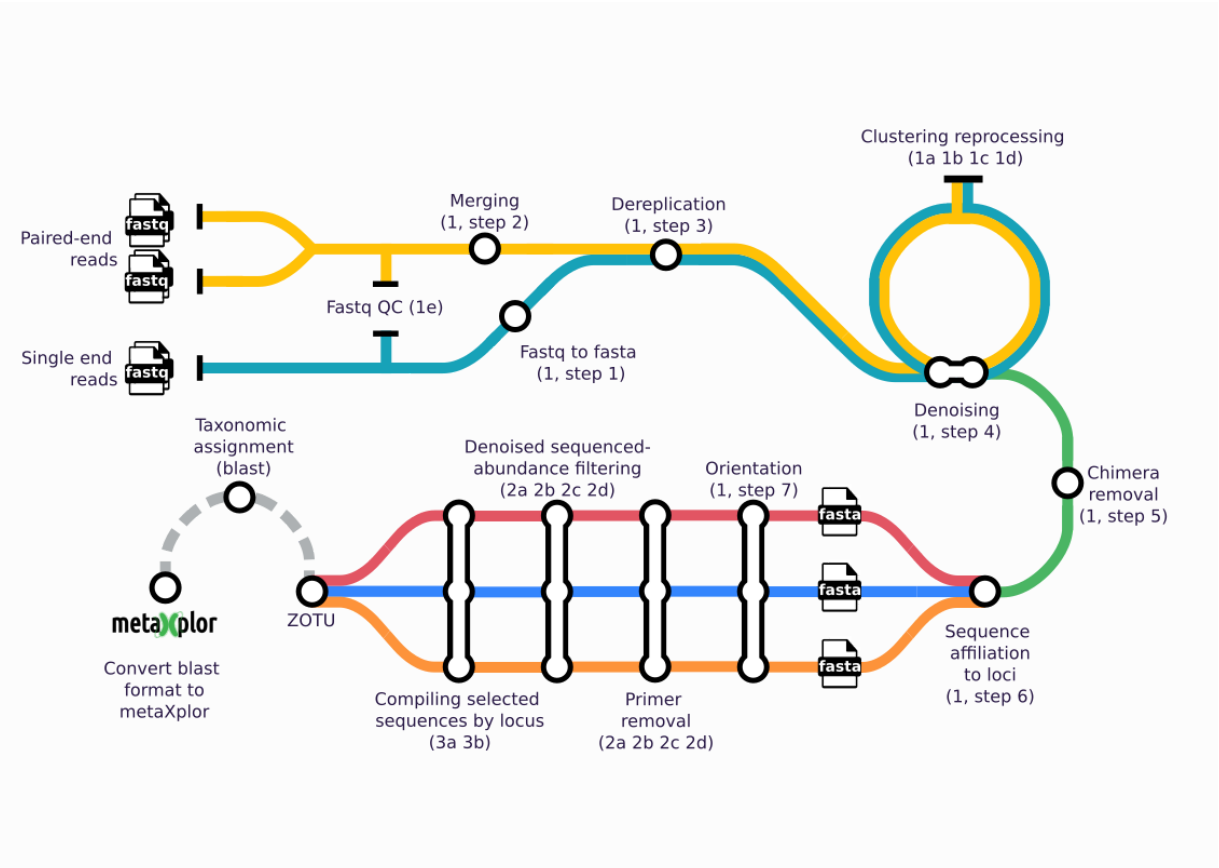


Figure 3: Pre-required files and directories

```
/YOUR_PROJECT_PATH/
├──────fastq
│  └──────sample1.R1.fastq
│  └──────sample1.R2.fastq
│  └──────sample2.R1.fastq
│  └──────sample2.R2.fastq
│  └──────...
├──────refs
│  └──────locus1.fas
│  └──────locus2.fas
│  └──────...
├──────samples.txt
├──────fastqR1.txt
├──────fastqR2.txt
├──────lociPE.txt
├──────lociSE.txt
```

Figure 4: Contents of the main menu, explaining navigation conventions and providing access to the four main sections.

```
----------------------------- mbctools - MAIN MENU -----------------------------

Please cite this software as follows:
mbctools:       A User-Friendly Metabarcoding and Cross-Platform Pipeline for Analyzing
                Multiple Amplicon Sequencing Data across a Large Diversity of Organisms
Christian Barnabé, Guilhem Sempéré, Vincent Manzanilla and Etienne Waleckx.
https://github.com/GuilhemSempere/mbctools


NAVIGATION CONVENTIONS:
Entering 'back' returns to the program upper level, if any
Entering 'home' returns to this main menu
Entering 'exit' leaves the program
Validating without typing anything applies the default value, if any


We recommend executing procedures in the provided order:

1 -> INITIAL ANALYSIS (mandatory): read merging, sample-level dereplication, sequence clustering,
        chimera detection, affiliation of sequences to loci, and sequence re-orientation

2 -> PRIMER REMOVAL AND SELECTION OF MINIMUM SEQUENCE ABUNDANCE LEVELS ACCORDING TO USER-DEFINED THRESHOLDS

3 -> GENERATION OF A UNIQUE SEQUENCE FILE FOR EACH LOCUS (comprising all samples' data)

4 -> EXPORTING ANALYSIS RESULTS INTO metaXplor FORMAT
```

Figure 5: Contents of section 1 menu, initial analyses. In menu 1, only option 1, INITIAL ANALYSIS, is mandatory as it executes all steps required to proceed with menu 2. Options 1a to 1d may be used after option 1 in case the user wants to refine some parameters. The analysis 1e is optional and provides information about the reads' quality.

```
--- MENU 1: BASIC ANALYSIS - only option 1 is strictly mandatory ---

1  -> INITIAL ANALYSIS (mandatory): read merging, sample-level dereplication, sequence clustering,
        chimera detection, affiliation of sequences to loci, and sequence re-orientation

1a -> Re-analyze all loci, from the clustering step, modifying parameters

1b -> Re-analyze only one locus of paired-end amplicon (merged reads), modifying parameters

1c -> Re-analyze only one locus of single-end amplicon (R1 only), modifying parameters

1d -> Re-analyse a given sample, modifying parameters

1e -> Optional quality checking of fastq files (slow)
```

Figure 6: Menu 2 combines two procedures that need to be launched separately for each locus: primer removal and application of an abundance threshold for retaining centroid sequences.

```
--- MENU 2: PRIMER REMOVAL, SELECTION OF MINIMUM SEQUENCE ABUNDANCE LEVELS ACCORDING TO USER-DEFINED THRESHOLDS ---

2a -> Apply the SAME size threshold for ALL SAMPLES for the loci based on PAIRED-END reads (R1/R2 merged)
        i.e. you want to keep only sequences whose abundance is greater than x% of the total number of
        sequences for a given sample. This threshold of x% can be chosen for each locus.

2b -> Apply the SAME size threshold for ALL SAMPLES for the loci based on SINGLE-END reads (R1 only)
        same as option 2a but only using the R1 reads instead of merged ones.

2c -> Apply a SPECIFIC size threshold for a given sample, for the loci based on PAIRED-END reads (R1/R2 merged)
        i.e. you want to modulate the threshold of x% by locus but also by sample within a particular locus.

2d -> Apply a SPECIFIC size threshold for a given sample, for the loci based on SINGLE-END reads (R1 only)
        same as option 2c but only using the R1 sequences instead of merged ones.
```

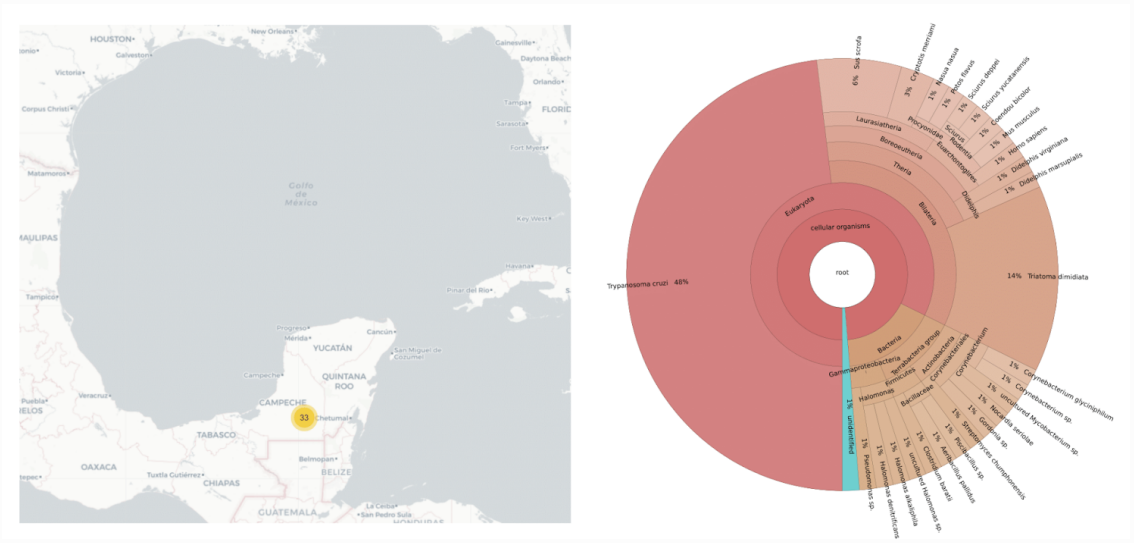
Figure 7: Contents of section 3 menu, allowing to obtain per-locus fasta files. Menu 3 provides means to generate fasta files containing the retained sequences, either for all loci at once, or for a given locus.

```
--- MENU 3: GENERATION OF A UNIQUE SEQUENCE FILE FOR EACH LOCUS (comprising all samples' data) ---

3a -> Process all loci at once

3b -> Process loci one by one
```

Figure 8: Contents of section 4 menu, guiding users in obtaining necessary files to create a metaXplor import archive (i.e., sample metadata, sequence fasta and abundance, assignment to reference taxa) in order to summarize all project results.

```
--- MENU 4: EXPORTING ANALYSIS RESULTS INTO metaXplor FORMAT ---

4a -> Generate sequence files
        Compiles all sequences selected for all loci into a single fasta
        Outputs a .tsv file indicating samples weights for each sequence

4b -> Generate assignment file
        Converts blastn results (obtained from blasting above-mentioned fasta file) from 'Hit table (text)'
        (format #7) into metaXplor format

4c -> Build metaXplor-format sample metadata file from provided tabulated file

4d -> Compresses all metaXplor files into a final, ready to import, zip archive
```

Figure 9: Graphic output from the metaXplor web interface. On the left the geographic origin of the thirty three samples. On the left side the taxonomic proportion of each identified organism (21).



Annex1: VSEARCH parameters mentioned in the article

--*centroids*: Output cluster centroid sequences to filename, in fasta format. The centroid is the sequence that seeded the cluster (i.e. the first sequence of the cluster).

--*db ‘filename’*: indicates the reference file to use for global pairwise alignment.

--*fastq_allowmergestagger*: this option allows to merge staggered read pairs. Staggered pairs are pairs where the 3’ end of the reverse read has an overhang to the left of the 5’ end of the forward read. This situation can occur when a very short fragment is sequenced. The 3’ overhang of the reverse read is not included in the merged sequence.

--*id real*: sets the minimum pairwise sequence identity to use in the command; *value in range [0 - 1.0]*.

--*matched ‘filename’*: writes sequences matching database target to *filename*, in *fasta* format.

--*minseqlength int*: retains length sequences > integer.

--*minsize int*: retains abundances > integer; default is 8.

--*nonchimeras* ‘filename’: outputs the *fasta* file ‘filename’ containing all non-chimeric sequences

--*orient ‘filename’*: orients the sequences in a same direction in the given file

--*sizein*: takes into account abundance annotations of the headers of input file

--*sizeout*: adds abundance annotations to output *fasta* file headers

--*strand both*: checks both strands when comparing sequences

--*uchime3_denovo*: improved detection of chimeric sequences

--*unoise_alph int*: sets the *alpha* parameter used in the --*cluster_unoise* command; default is 2

Table 1: statistics concerning the processing of the test dataset with the number of reads per sample and the attribution to each marker.

Sample	Reads per sample	Merging	Number of cluster	Number of cluster with no chimera	Cluster per maker							
					12S	16S	COII	GPI	ITS2	ND1	MINEX	MINEX_SE
CON001	215,137	86.96%	11	11	0	15	0	0	2	3	6	0
CON002	268,415	92.29%	17	13	18	12	0	0	2	0	0	0
CON003	242,823	86.36%	87	87	0	9	0	1	2	0	12	0
CON004	205,358	93.44%	3	3	0	11	0	0	3	0	0	0
CON005	149,945	94.08%	7	7	1	6	0	0	2	0	0	0
CON006	180,763	93.11%	2	2	0	11	0	0	3	0	0	0
CON007	7,898	89.10%	2	2	1	0	0	0	2	0	0	0
CON008	200,589	92.40%	3	3	0	19	0	0	2	0	0	0
CON009	261,914	92.26%	2	2	0	15	0	0	2	0	0	0
CON010	283,721	93.25%	139	126	16	16	0	0	2	0	0	0
CON011	186,936	91.12%	1	1	0	16	0	0	2	0	0	0
CON012	194,776	83.26%	61	60	14	10	2	1	2	0	24	4
CON013	189,441	93.03%	14	14	4	8	0	0	4	0	5	0
CON014	151,880	90.86%	25	25	0	8	0	0	30	0	0	0
CON015	221,302	91.74%	1	1	0	14	0	0	2	0	0	0
CON016	139,546	86.99%	46	46	2	6	2	1	10	0	14	3
CON017	228,038	88.12%	3	3	0	11	0	0	2	0	0	0
CON018	258,481	90.12%	28	27	6	12	0	0	15	0	0	0
CON019	251,403	79.04%	77	77	1	14	4	2	10	0	3	0
CON020	223,972	92.83%	24	24	2	1	0	0	4	0	0	0
CON021	309,618	93.46%	8	8	0	15	0	0	14	0	0	0
CON022	197,125	84.79%	83	83	11	8	2	2	4	0	5	0
CON023	193,857	75.58%	53	53	1	7	2	1	5	5	15	0
CON024	190,310	91.32%	62	62	13	20	0	0	2	0	0	0
CON025	273,603	91.04%	44	37	29	15	0	0	2	0	0	0
CON026	284,204	94.50%	2	2	1	17	0	0	2	0	0	0
CON027	436,822	92.25%	5	5	1	34	0	0	2	0	0	0
CON028	243,235	93.85%	63	63	1	12	0	0	4	0	0	0
CON029	219,907	94.86%	13	13	4	11	0	0	2	0	0	0
CON030	306,738	94.56%	17	17	0	10	0	0	26	0	0	0
CON031	172,815	93.98%	87	87	1	11	0	0	2	0	0	0
CON032	206,078	90.69%	34	33	1	13	0	0	24	0	0	0
CON033	237,429	91.58%	92	92	37	14	0	0	2	0	0	0
Average	222,245	90.39%	34	33	5	12	0	0	6	0	3	0