ANERIS

Operational Sensing Life Technologies for Marine Ecosystems

Deliverable 2.4 – SLIM 1.0 code and documentation

Lead Beneficiary: NORCE

Author/s: Tristan Cordier (NORCE)

20/12/2024

Funded by
the European Union

Prepared under contract from the European Commission

Grant agreement No. 101094924

EU Horizon Europe Research and Innovation action

Project acronym:	**ANERIS**
Project full title:	**operAtional seNsing lifE technologies for maRIne ecosystemS**
Start of the project:	January 2023
Duration:	48 months
Project coordinator:	Jaume Piera

Deliverable title:	SLIM 2.0 code and documentation
Deliverable n°:	D2.4
Nature of the deliverable:	Other - Documentation
Dissemination level:	PU - Public

WP responsible:	WP2
Lead beneficiary:	NORCE

Citation: Cordier, T (2024). *SLIM 1.0 code and documentation*. Deliverable D2.4 EU Horizon Europe
ANERIS Project, Grant agreement No. 101094924

Due date of deliverable:	24
Actual submission date:	24

Deliverable status:

Version	Status	Date	Author(s)
1.0	Draft	02 December 2024	Tristan Cordier (NORCE)
1.1	Review	16 December	Christor Arvanitidis (LifeWatch)
1.1	Review	20 December 2024	Jaume Piera (CSIC)
1.2	Final	20 December 2024	Tristan Cordier (NORCE)

Table of Contents

Summary

The SLIM application has been designed to facilitate the deployment and execution of a bioinformatic pipeline. This document presents the first update of the SLIM application planned within ANERIS. SLIM will undergo iterative enhancements throughout the project's life cycle in order to implement new functionalities to process raw high-throughput sequencing data and produce biodiversity assessments. These new functionalities will be implemented as processing modules that wrap state of the art bioinformatics software, to be chained using a user-friendly Graphical User Interface. Within this deliverable, we delineate the objectives of SLIM, its technical architecture, and present the phases of development of the SLIM application.

List of Abbreviations

ASVs - Amplicon Sequence Variants

CC - Creative Commons

COI - Cytochrome Oxidase 1 gene.

CPU - Central Processing Units

GNU - General Public License

GUI - Graphical User Interface

HTS - High-Throughput Sequencing

ITS - Internal Transcribed Spacer

LCA - Last Common Ancestor

OTUs - Operational Taxonomic Units

PCR - Polymerase Chain Reaction

RAM - Random-Access Memory

SSU - Small Sub Unit of the ribosomal rRNA gene.

1. Introduction and objective

High-throughput sequencing of environmental samples, i.e. environmental genomics, is a fast and cost-effective method for studying biodiversity. It has become a widely used tool in various fields, including monitoring terrestrial and marine ecosystems, analyzing animal diets, and assessing environmental health. It is also applied to study ancient environments using sedimentary ancient DNA in both marine, freshwater and terrestrial ecosystems.

HTS generates raw data that must be processed through a series of bioinformatics tools, i.e. a pipeline. Such a pipeline usually includes some quality filtering steps, clustering sequencing reads into biological meaningful “units” (e.g., OTUs, ASVs), that can correspond to different biological organization (from sub-species to species, from genus to family level) and taxonomic and/or functional annotations. This processed data then forms the basis for biodiversity assessment, ecological interpretation and/or any other hypothesis testing.

Current state-of-the-art bioinformatics tools require at least basic command-line knowledge to be used, making the processing of raw sequencing data potentially challenging for non-experts and end-users. SLIM (Dufresne et al., 2019) is an open-source web application that simplifies the creation and execution of sequencing data processing pipelines through an intuitive GUI, but its capabilities are currently restricted to short amplicons data (i.e. metabarcoding).

In ANERIS, our objective is to extend the capabilities of SLIM to integrate new modules able to process long-reads amplicons, and metagenomics. The versatility and scalability of SLIM (in its v2.0) will thus be largely improved.

1. Technical details

1.1. Container technologies

Container technologies provide a reproducible way to package software applications and their dependencies together into isolated, portable environments called containers. These containers ensure that applications run consistently across different computing environments. Several container technologies exist, and SLIM was initially developed using Docker. One limitation of Docker is that it needs admin credentials (i.e. ‘sudo’) to be built and managed, which is not ideal in terms of security. To circumvent this, we moved SLIM to Podman, which does not require admin credentials to build and manage containers.

1.2. The Node.js framework

Node.js is a JavaScript runtime, an open source server framework that has been designed for the building of scalable network applications, by handling asynchronous and parallel events. The core of the SLIM application is built to scale and schedule jobs on computing nodes or personal computers. Depending on the resources available (CPU, RAM), one can set how SLIM will scale. By default, a job takes up to 8 CPU cores. The number of available cores on the

machine determines how many jobs can run in parallel. For example, with 1-8 available cores, one job can run; with 16 cores, two jobs can run, and so on. If additional jobs are being submitted, these will be queued and executed once more resources become available.

1.3. Web interface

1.3.1. graphical user interface

SLIM relies on a web server which is accessible via any web browser. The landing page allows users to upload sequencing and other required files, and select the module from a drop-down list (Figure 1).

***Figure 1**: Landing page of SLIM, with the first section dedicated to select the files to be uploaded, and the drop-down list of modules to be chained.*

1.3.2. wildcard and file pointing

Modules are interconnected using input/output file names. To simplify sample selection during analysis, wildcards (i.e. '*' meaning "all") are automatically generated during demultiplexing or using the wildcard-creator module. These wildcards, derived from compressed multiplexed library files and the tag-to-sample file, are used by the application to schedule the modules execution steps. Users cannot manually type wildcards in file names. Instead, the GUI uses autocompletion in module input fields to suggest the files to the user for handy selection.

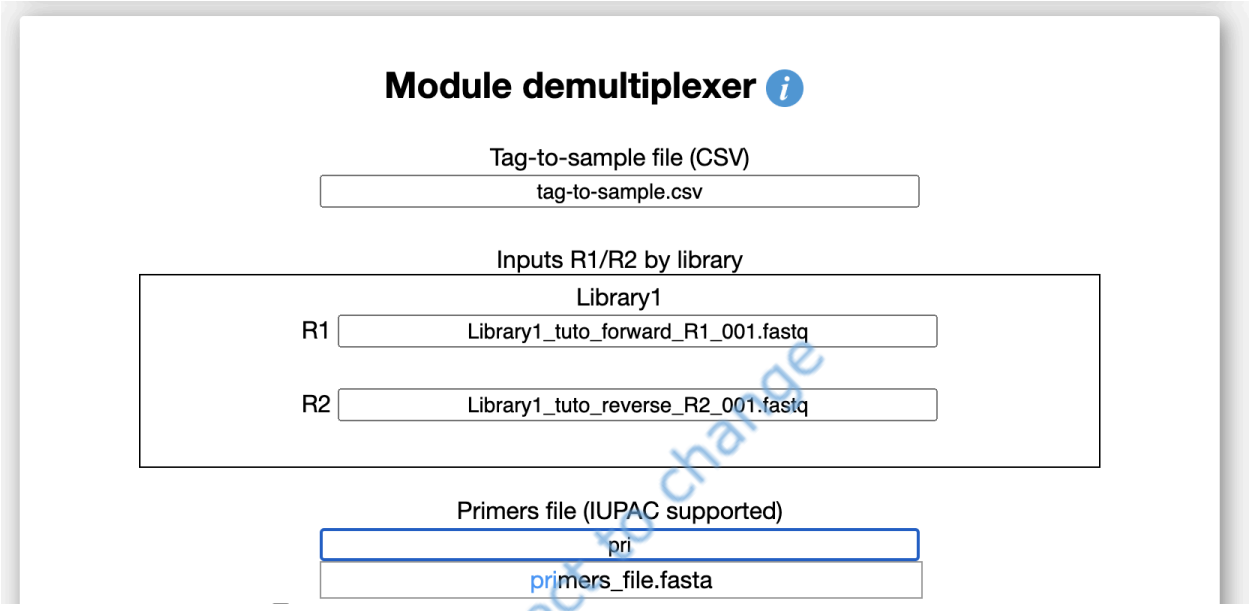


Figure 2: *Autocompletion feature in the GUI. Starting to type a filename will highlight suggestions, that can then be picked for fast setup of the workflow.*

For processing pipelines involving already-demultiplexed libraries or conventional illumina multiplexing strategies (where ".fastq" files correspond to individual samples), the wildcard is not generated automatically since there is no demultiplexing step using a tag-to-sample file. Instead, the wildcard-creator module has to be used to generate the required wildcard patterns that can later be fetched within input fields of modules.

1.3.3. mailing system

Once the processing pipeline has been set, the user has to fill an email address to execute the pipeline. When the execution starts, an email is sent to the user with a unique link to his job on the server. Once the job has ended, the user receives another email inviting him to download the results. Results are kept 24 hours on the server by default (this can be changed) and a reminder email is sent to the user a few hours before the job and the results are being deleted. This ensures an optimized usage of storage.

1.4. Reproducibility of processing pipelines

The SLIM application includes an easy system to facilitate analysis reproducibility. The GUI stores the succession of modules and their respective parameters into a small configuration file. To reproduce any SLIM execution, one needs the raw sequencing data, the version of SLIM that has been used to process the data and this configuration file. This configuration file is sent in the email to the user, but can also be manually saved.

1.5. Current software and capabilities

Table 1: *Current software implemented in SLIM.*

Software names (version)	Main function	Licence	Link	Reference
DTD (v1.0)	Demultiplexing	GNU Affero	https://github.com/yoann-dufresne/DoubleTagDemultiplexer	NA
PANDAseq (v2.11)	Pair-end joiner	GNU	https://github.com/neufeld/pandaseq	Masella et al., 2012
Casper (v0.8.2)	Pair-end joiner	CC	http://best.snu.ac.kr/casper	Kwon et al., 2014
vsearch (v2.8)	Paired-end joiner, chimera filtering, OTU clustering	GNU	https://github.com/torognes/vsearch	Rognes et al., 2016
DADA2 (v1.16)	ASV inference, pair-end joiner, chimera filtering	GNU	https://github.com/benjjneb/dada2	Callahan et al., 2016
swarm (v2.2.2 and v3.1.4)	OTU clustering (illumina)	GNU	https://github.com/torognes/swarm	Mahé et al., 2015 Mahé et al., 2021
ASHURE + OPTICS (v1.0.0)	OTU clustering (nanopore)	GNU	https://github.com/BBaloglu/ASHURE	Baloğlu et al., 2021
IDTAXA (v2.16)	Taxonomic annotation	GPL-3	http://www2.decipher.codes/index.html	Murali et al., 2018
LULU (v1)	Post-clustering matrix curation	GNU	https://github.com/tobiasgf/lulu	Frøslev et al., 2017
MSI (v0.3.7)	Reads polishing (nanopore)	GNU	https://github.com/nunofonseca/msi	Egeter et al., 2022

Below is a plain description of the current modules implemented in SLIM:

Demultiplexing / sample grouping

- Double Tag Demultiplexing (DTD)
  By adding specific nucleotides tails to PCR amplification primers, it is possible to pool (or multiplex) multiple samples together in a sequencing library. This approach dramatically reduces sequencing costs, allowing upscaling experiments and routinely analyzes hundreds to thousands of samples. This module allows the retrieval of individual samples from multiplexed illumina sequencing libraries.

- Wildcard creator
  This module allows to manually group the samples, i.e. prepare groups of files that need to be processed together, outside of common use cases (see “wildcard and file pointing” section).

Paired-end joiners

These modules allow the merging of pair-ends sequencing reads, that each contain a fraction of the amplicon, into full-length reads. Although very similar in their approach, differences exist in their implementation.

- Pandaseq joining
- Casper
- Mergepair vsearch

Chimeras filtering

- Chimera vsearch
  This module filters pervasive PCR artefacts, namely chimeras. Chimeras occur when the extension of an amplicon is stopped, and the partial product functions as a primer in the next PCR cycle. The partial product anneals to the wrong template and continues to extend, thereby synthesizing a single DNA fragment sourced from two different template DNA.

ASVs inference / OTUs clustering

- DADA2
  This module takes pairs of fastq files, i.e. before pair-end joiners, and uses a density-based denoising approach, by fitting error models to the raw sequencing data, to infer ASVs that can later be taxonomically annotated. The module also includes the chimera filtering step, and produces an ASVs to samples matrix, populated with reads counts. DADA2 has become a standard approach for metabarcoding data.

- vsearch
  This module implements the vsearch toolkit to produce OTUs from full-length, chimera-free, sequencing reads. It uses a sequence similarity-based approach to group reads together into OTUs. This approach has been mostly discontinued in modern metabarcoding pipeline, because it relies on a fixed similarity threshold (typically 97%), not accounting for possible biological variation of marker genes that exists between taxonomic groups.
- swarm v2
  This module implements the swarm clustering tool to produce OTUs from full-length, chimera-free, sequencing reads. It uses a sequence density-based approach to group reads together into OTUs. This approach assumes that abundant reads are genuine biological sequences, and group neighbouring sequences that stem from PCR or sequencing errors, in an iterative manner. As opposed to the fixed clustering threshold approach, this approach allows accounting for variable cluster sizes, to produce more biologically meaningful OTUs.
- swarm v3
  This module implements the latest version of swarm, which has been optimized to reduce computation time and resources, while achieving the same results.
- OPTICS
  This module implements the OPTICS clustering algorithm to produce OTUs from full-length, chimera-free, sequencing reads. It also uses a density-based approach to group the reads.

Taxonomic assignment

These modules are used to make taxonomic annotations to the OTUs/ASVs, based on their match with reference sequences in curated databases. Such reference databases include SSU of the ribosomal rRNA gene, e.g. PR2 (Guillou et al., 2013), SILVA (Quast et al., 2013), or the COI gene, e.g. MIDORI (Leray et al., 2022), or the ITS marker for fungi, e.g. UNITE (Kõljag et al., 2005). One can use a single ".fasta" file as input, and get a list of taxonomic annotations from each individual sequence of the fasta, or OTUs/ASVs table, in which case the taxonomic annotations are appended at the side of the OTUs/ASVs table. This annotated table therefore contains the main information to be analyzed for biodiversity assessment.

Two main approaches are implemented, one based on similarity cut-off and a LCA algorithm (vsearch), and another one based on k-mer classification trees (IDTAXA).

- fasta - vsearch
- fasta - IDTAXA
- OTU/ASV table - vsearch
- OTU/ASV table - IDTAXA

Post-processing and matrix curation

- LULU post clustering
  This tool can be used to group OTUs/ASVs sequences that are similar and that co-occur throughout the samples. The assumption is that such OTUs/ASVs may represent the same organism, although with a different DNA signature (because of PCR, sequencing errors, or because of multiple variants of the gene within a genomes or population).

Nanopore pipelines

- ASHURE
  This software has been developed to process nanopore metabarcoding data based on rolling circle amplification. Resulting reads contain repeats of their originating template (i.e. concatemers). These concatemers can be realigned to generate more error-free consensus reads. The module takes a list of nanopore fastq files produced using the rolling circle amplification protocol.

- MSI
  This software has been developed to process nanopore metabarcoding data. In SLIM, we include only some of its steps, namely the polishing of reads (grouping reads into similar chunks and producing consensus, i.e. more error-free reads) and primers trimming. The module takes a list of nanopore fastq files.

Miscellaneous / Utils

- Dereplication
  This module is used within some other module in SLIM, but can be called individually to “dereplicate” reads, i.e. collapsing identical reads while keeping the reads counts in each sequence headers of a “.fasta” file, for further processing.

- ASV/OTU filtering
  This module is used to filter out rare OTUs, based on their low reads count throughout the matrix. These sequences are sometimes considered artefactual, although some may be genuine. It also reduces the dimensionality of the data (more variables than observation), which can reduce computation times when analysing beta-diversity patterns.

- FASTA merging
  This module is used within other modules to merge multiple fasta files. It can also be called individually.

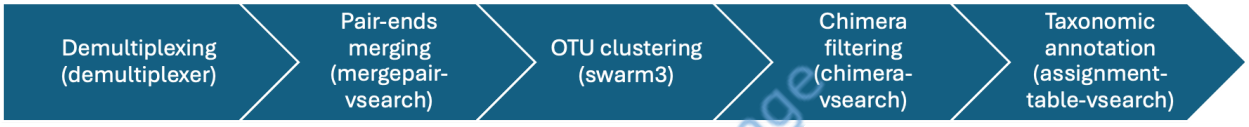
- FASTQ to FASTA
  This module converts fastq files into fasta files.

- CHOPPER
  This module is used to filter nanopore fastq files, by discarding low quality reads and/or trimming the edges of sequences that often contain poor-quality base calling scores in nanopore data.

2. Toy datasets and suggested modules chaining

Although the documentation specifies the file formats and module capabilities, we provide toy datasets and exemple processing pipelines for different types of HTS data, in order to allow one to inspect the files format, and suggested module chaining. In a nutshell:

For illumina HTS data:



For nanopore HTS data:

3. Future development

New capabilities.

In the next phase of the project, we will integrate processing modules to process shotgun metagenomics data into SLIM. A selection of candidate softwares for this task has been identified, although the list has yet to be narrowed. We expect to have selected the software within the first half of 2025. Modules development will follow this step and we expect to have implemented metagenomics capabilities in SLIM by the end of 2025.

Mailing system.

The initial version of SLIM included a mailing system to follow up a submitted job. The solution was using the gmail mailing service, which has now discontinued automated service. The mailing service has therefore been hidden in the application, until we identify a new technical solution to reinstate it.

Acknowledgements

The authors acknowledge the financial support provided by the European Union's Horizon Europe research and innovation programme under Grant Agreement No. [101094924], which has enabled the successful implementation of the ANERIS project. This funding has been essential for advancing our research and development activities in marine biodiversity assessment.

References

Baloğlu, B., Chen, Z., Elbrecht, V., Braukmann, T., MacDonald, S., & Steinke, D. (2021). A workflow for accurate metabarcoding using nanopore MinION sequencing. Methods in Ecology and Evolution, 12(5), 794–804. https://doi.org/10.1111/2041-210X.13561

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. http://dx.doi.org/10.1038/nmeth.3869

Egeter, B., Veríssimo, J., Lopes-Lima, M., Chaves, C., Pinto, J., Riccardi, N., Beja, P., & Fonseca, N. A. (2022). Speeding up the detection of invasive bivalve species using environmental DNA: A Nanopore and Illumina sequencing comparison. Molecular Ecology Resources, 22(6), 2232–2247. https://doi.org/10.1111/1755-0998.13610

Dufresne, Y., Lejzerowicz, F., Perret-Gentil, L. A., Pawlowski, J., & Cordier, T. (2019). SLIM: A flexible web application for the reproducible processing of environmental DNA metabarcoding data. BMC Bioinformatics, 20(1), 1–6. https://doi.org/10.1186/s12859-019-2663-2

Frøslev, T. G., Kjøller, R., Bruun, H. H., Ejrnæs, R., Brunbjerg, A. K., Pietroni, C., & Hansen, A. J. (2017). Algorithm for post-clustering curation of DNA amplicon data yields reliable biodiversity estimates. Nature Communications, 8(1). https://doi.org/10.1038/s41467-017-01312-x

Guillou, L., Bachar, D., Audic, S., Bass, D., Berney, C., Bittner, L., Boutte, C., Burgaud, G., de Vargas, C., Decelle, J., del Campo, J., Dolan, J. R., Dunthorn, M., Edvardsen, B., Holzmann, M., Kooistra, W. H. C. F., Lara, E., le Bescot, N., Logares, R., … Christen, R. (2013). The Protist Ribosomal reference database (PR2): a catalog of unicellular eukaryote Small Sub-Unit rRNA sequences with curated taxonomy. Nucleic Acids Research, 41(D1), D597–D604. https://doi.org/10.1093/nar/gks1160

Kõljalg, U., Larsson, K. H., Abarenkov, K., Nilsson, R. H., Alexander, I. J., Eberhardt, U., Erland, S., Hoiland, K., Kjoller, R., Larsson, E., Pennanen, T., Sen, R., Taylor, A. F. S., Tedersoo, L., Vralstad, T., & Ursing, B. M. (2005). UNITE: a database providing web-based methods for the molecular identification of ectomycorrhizal fungi. New Phytologist, 166(3), 1063–1068.

Kwon, S., Lee, B., & Yoon, S. (2014). CASPER: context-aware scheme for paired-end reads from high-throughput amplicon sequencing. BMC Bioinformatics, 15(Suppl 9), S10. https://doi.org/10.1186/1471-2105-15-S9-S10

Leray, M., Knowlton, N., & Machida, R. J. (2022). MIDORI2: A collection of quality controlled, preformatted, and regularly updated reference databases for taxonomic assignment of eukaryotic mitochondrial sequences. Environmental DNA. https://doi.org/10.1002/edn3.303

Mahé, F., Rognes, T., Quince, C., de Vargas, C., & Dunthorn, M. (2015). Swarm v2: highly-scalable and high-resolution amplicon clustering. PeerJ, 3, e1420. https://doi.org/10.7717/peerj.1420

Mahé, F., Czech, L., Stamatakis, A., Quince, C., de Vargas, C., Dunthorn, M., & Rognes, T. (2021). Swarm v3: towards tera-scale amplicon clustering. Bioinformatics. https://doi.org/10.1093/bioinformatics/btab493

Masella, A. P., Bartram, A. K., Truszkowski, J. M., Brown, D. G., & Neufeld, J. D. (2012). PANDAseq: paired-end assembler for illumina sequences. BMC Bioinformatics, 13(1), 31. https://doi.org/10.1186/1471-2105-13-31

Murali, A., Bhargava, A., & Wright, E. S. (2018). IDTAXA : a novel approach for accurate taxonomic classification of microbiome sequences. Microbiome, 6(140), 1–14. https://doi.org/10.1186/s40168-018-0521-5

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., & Glöckner, F. O. (2013). The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. Nucleic Acids Research, 41, D590–D596. https://doi.org/10.1093/nar/gks1219

Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). VSEARCH: a versatile open source tool for metagenomics. PeerJ, 4, e2584. https://doi.org/10.7717/peerj.2584