ANERIS

Operational Sensing Life Technologies for Marine Ecosystems

`

Deliverable 2.2 – Workflow for the analysis of genomic data

Lead Beneficiary: LifeWatch ERIC

Author/s: Joaquín López Lerida, Julio López Paneque, Marie-Catherine Bouquieaux, Pascal Hablützel, Hanneelor Heyderickx, Cristina Huertas, Christos Arvanitidis

16/12/2024

Funded by
the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the EU nor the EC can be held responsible for them.

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:	Workflow for the analysis of genomic data
Deliverable n°:	D2.2
Nature of the deliverable:	Other
Dissemination level:	Public
WP responsible:	WP2
Lead beneficiary:	LifeWatch ERIC
Citation:	Joaquín López Lerida, Julio López Paneque, Marie-Catherine Bouquieaux, Pascal Hablützel, Hanneelor Heynderickx, Cristina Huertas and Christos Arvanitidis (2024). Workflow for the analysis of genomic data. Deliverable D2.2.y EU Horizon Europe ANERIS Project, Grant agreement No. 101094924
Due date of deliverable:	Month n°24
Actual submission date:	Month n°24

Deliverable status:

Version	Status	Date	Author(s)
1.0	Draft	16 December 2024	López et al., LW ERIC and VLIZ
1.1	Review	18 December 2024	Berta Companys, CSIC
1.2	Review	18 December 2024	Tristan Cordier, NORCE
2.0	Final	20 December 2024	López et al., LW ERIC and VLIZ
2.0	Final Review	20 December 2024	Berta Companys, CSIC

The content of this deliverable does not necessarily reflect the official opinions of the European Commission or other institutions of the European Union

Table of Contents

Preface

This document serves as a deliverable for the ANERIS project, funded by the European Union's Horizon Europe Research and Innovation Action under Grant Agreement No. 101094924.

Deliverable D2.2: Workflow for the Analysis of Genomic Data outlines the successful development of workflows designed for the analysis of genomic data, focusing on the first Operational Marine Biology (OMB) product, **Intraspecific Variation**, developed within ANERIS Work Package 2 (WP2), Genomic Technologies.

This deliverable establishes the foundation for future works, which will describe the continued development and integration of workflows for additional OMB products. The second and third OMB product will focus on local and regional species diversity, and non-indigenous species, respectively.

Summary

This report provides a comprehensive overview of the development, integration, and deployment of the ANERIS OMB Product 1 workflow **Intraspecific Variation**, currently integrated into LifeWatch ERIC's infrastructure. It covers the complete process from initial development to production deployment, including testing, validation, and future development strategies.

This workflow was designed to enable scalable and reliable genomic data analysis, integrating seamlessly with MyLifeWatch platform (https://my.lifewatch.eu). This platform supports the workflow both as a standalone component and as a modular tool within the workflow editor, ensuring broad accessibility and adaptability.

The initial section of this report delves into the primary research question tackled by the OMB product Intraspecific Variation. It elucidates the scientific rationale for the development of this product and its role in advancing the understanding of genomic biodiversity via sophisticated analytical workflows.

Following this, the "Workflow Drawing" section provides a comprehensive depiction of the Intraspecific Variation workflow. It features detailed diagrams that sequentially illustrate the workflow's crucial elements, data inputs, and anticipated outputs. These visual aids ensure a clear comprehension of the processes involved, from data intake to the production of analytical results.

The core of the report, Integration of ANERIS OMB Product 1 Workflow into LifeWatch ERIC's Infrastructure, outlines the complete process of embedding the OMB Product 1 workflow within the LifeWatch ERIC infrastructure. This section meticulously describes the integration effort, addressing four main aspects: **Workflow Development and Integration, Testing and Validation, Production Deployment and Continuous Development and Feedback.**

By adhering to these clearly defined phases, the ANERIS OMB Product 1 workflow has been effectively developed and integrated, thereby bolstering genomic data analysis capabilities within the LifeWatch ERIC framework.

List of Abbreviations

ANERIS: OperAtional seNsing lifE technologies for maRIne ecosystemS

CSV: Comma-Separated Values

EBVs - Essential Biodiversity Variables

EOVs - Essential Ocean Variables

GPL: GNU General Public License

JSON: JavaScript Object Notation

LWE: LifeWatch ERIC

MSFD - Marine Strategy Framework Directive

OMB: Operational Marine Biology

WP2: Work Package 2

1. OMB Product 1 "Intraspecific Variation" Scientific Question

The ongoing biodiversity crisis may result in much of the ocean's biodiversity being lost or deeply altered without even being known. As the climate and anthropogenic-related impacts on marine systems accelerate, biodiversity knowledge integration is required urgently to evaluate and monitor marine ecosystems and to support suitable responses to underpin a sustainable future. Biological observations need to improve radically to enhance our understanding of marine ecosystems and biodiversity under long-term global change and multiple stressors. However, this is not trivial as biological properties are more difficult to measure and integrate, compared to physical or chemical parameters. ANERIS proposes to implement the concept of Operational Marine Biology (OMB), understood as a biodiversity information system for systematic and long-term routine measurements of the ocean and coastal life, and their rapid interpretation and dissemination. The achievement of the new Operational Marine Biology system is a key goal for the next decade and will enable a baseline of biological information related to Essential Biodiversity Variables (EBVs) and Essential Ocean Variables (EOVs). It will also deliver critical data for descriptors for Marine Policies, in particular the Marine Strategy Framework Directive (MSFD).

This workflow aims to describe the steps used to produce the results for the OMB products - intraspecific variation using metabarcoding data.

2. Workflow Drawing

2.1. Workflow Diagram

In a first step, DNA sequences belonging to the focal species are identified and moved into a species-specific FASTA file along with matching sequences from the reference (annotation) database using a combination of an R-script and shell code. Then the tool MAFFT is deployed to align all sequences in the file. A second R-script then calculates and draws haplotype networks, generates geographic distribution maps using coordinates from the metadata file, and calculates relevant summary statistics.

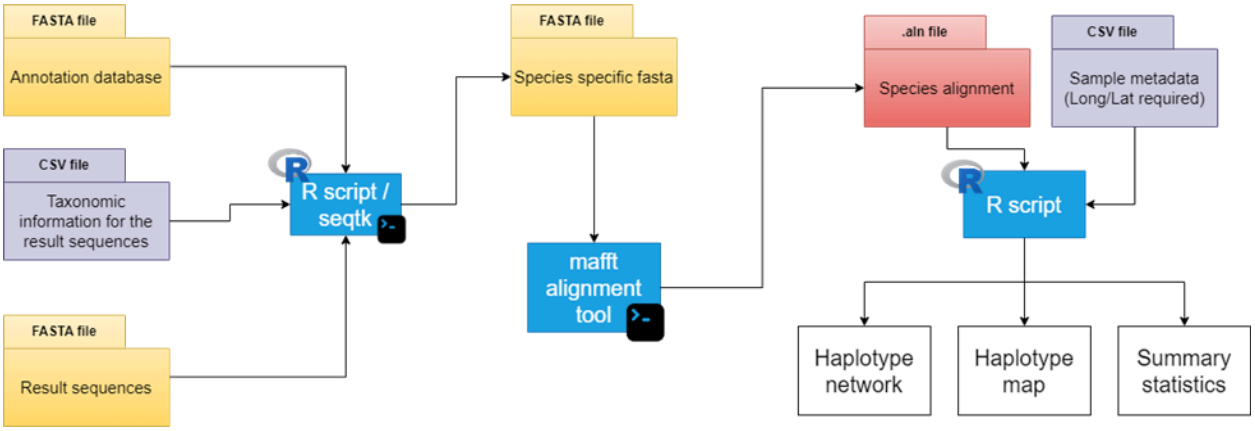

Figure 1. Workflow diagram showing the calculation of haplotype networks and associated results (geographic visualization and relevant summary statistics).

2.2. Step by Step Descriptions

STEP 1: Generate species specific FASTA file

Figure 2 shows the generation of one or more (usually many, depending on the dataset) species specific FASTA files. In the following lines the code and the necessary in- and output files are described.

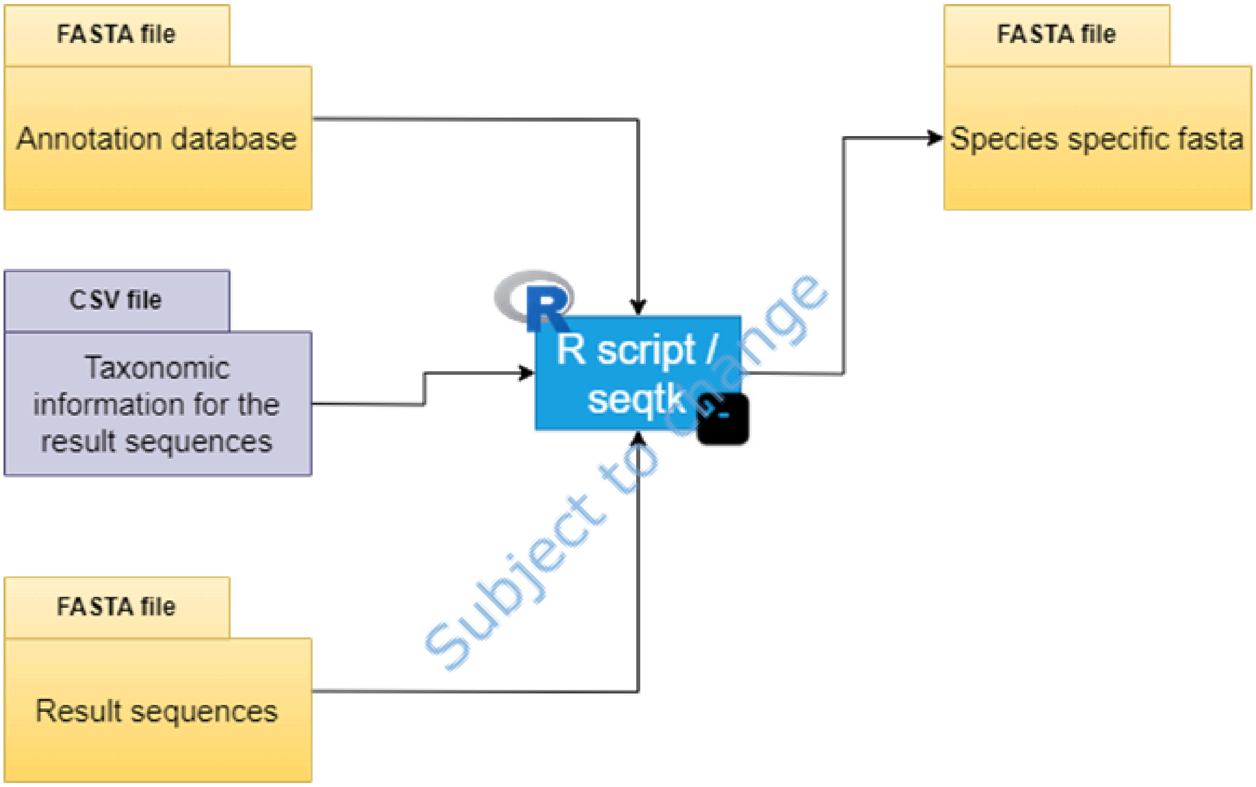

Figure 2. Workflow diagram showing the generation of species-specific individual sequence files from metabarcoding results.

INPUT1:

- Format: FASTA file
- Description: The annotation database FASTA should be the same database used to annotate the sequences of the samples. It consists of all the sequences of the database and their ID.

- Source: As for now, the workflow has been tested on the COI amplicon, and for this amplicon the database used was MIDORI2_UNIQ_NUC_GB257_CO1_BLAST. It can be downloaded here: https://www.reference-midori.info/

INPUT2:

- Format: CSV file

- Description: The sample CSV file is a result file from an annotation workflow. It contains the taxonomic information related to the sequences as well as information linking the sequences to its original sample. I must contain at least a column called “species”, with the species name associated with the sequence.

- Source: Result CSV was obtained with an annotation workflow.

INPUT3:

- Format: FASTA file

- Description: The FASTA file is a result file from an annotation workflow. The FASTA file contains all the sequences from all the samples used in the annotation workflow. Each sequence is associated with a specific ID. This ID must contain at the end an underscore followed by the name of the sample associated with that sequence (e.g.: sequenceID_samplename).

- Source: Result FASTA was obtained with an annotation workflow.

OUTPUT 1:

- Format: FASTA files

- Description: One FASTA file per species identified in the samples.

OUTPUT 2:

- Format: Temporary files, both in TXT and FASTA format

- Description: Temporary files are created during the workflow; they are not used after. They are kept in case a check needs to be done if an anomaly is detected. They consist of four different files per species: the list of sequences to be extracted from the result FASTA, a list of sequences to be extracted from the database, a FASTA file containing the sequences extracted from the result FASTA and a FASTA file containing the sequences extracted from the database.

SERVICE:

- Name: get_extract.R

- Type: R script

- Description: R script that can be run from the command line. It takes 5 arguments, the path to the 3 files described above (by default, a database FASTA file in not needed, and only the sample FASTA file), an output path for the results and the minimum of sequences desired (default is 2, as this is the minimum of sequences required for an alignment in the next step). Loops through the unique species present in the result csv and then extracts the sequences for each of those species from the result FASTA into a species specific FASTA. It does the same for the annotation database. To extract the sequence from both FASTA, a command line tool is called from within the R-script: *seqtk*. The IDs of the sequences are shortened in both cases (result/annotation database) to ease the identification later in the alignment. To do this, another command line tool is called from within the R-script: *awk*. Finally, both extractions are combined into one FASTA using the command tool line *cat* (again, called in the R-script). Here is an example of command that can be used:

    1. Rscript --vanilla get_extract.R -r COI.csv -f COI.fasta -d database.fasta -o ./result_extraction/ -s 5
    2. Rscript --vanilla get_extract.R --marker_result=COI.csv --marker_fasta=COI.fasta --database_fasta database.fasta --output=./result_extraction/ --number_sequences=5

- Availability: R-script not yet publicly available. It will be made available on LWE´s GitHub before the project ends. Seqtk is freely available online: https://github.com/lh3/seqtk .

- Author: Rscript: Marie-Catherine Bouquieaux, Pascal Hablützel; Seqtk: Heng Li

STEP 2: Alignment of DNA sequences

In this step, the MAFFT algorithm is used to align all sequences in the species specific FASTA files to each other.

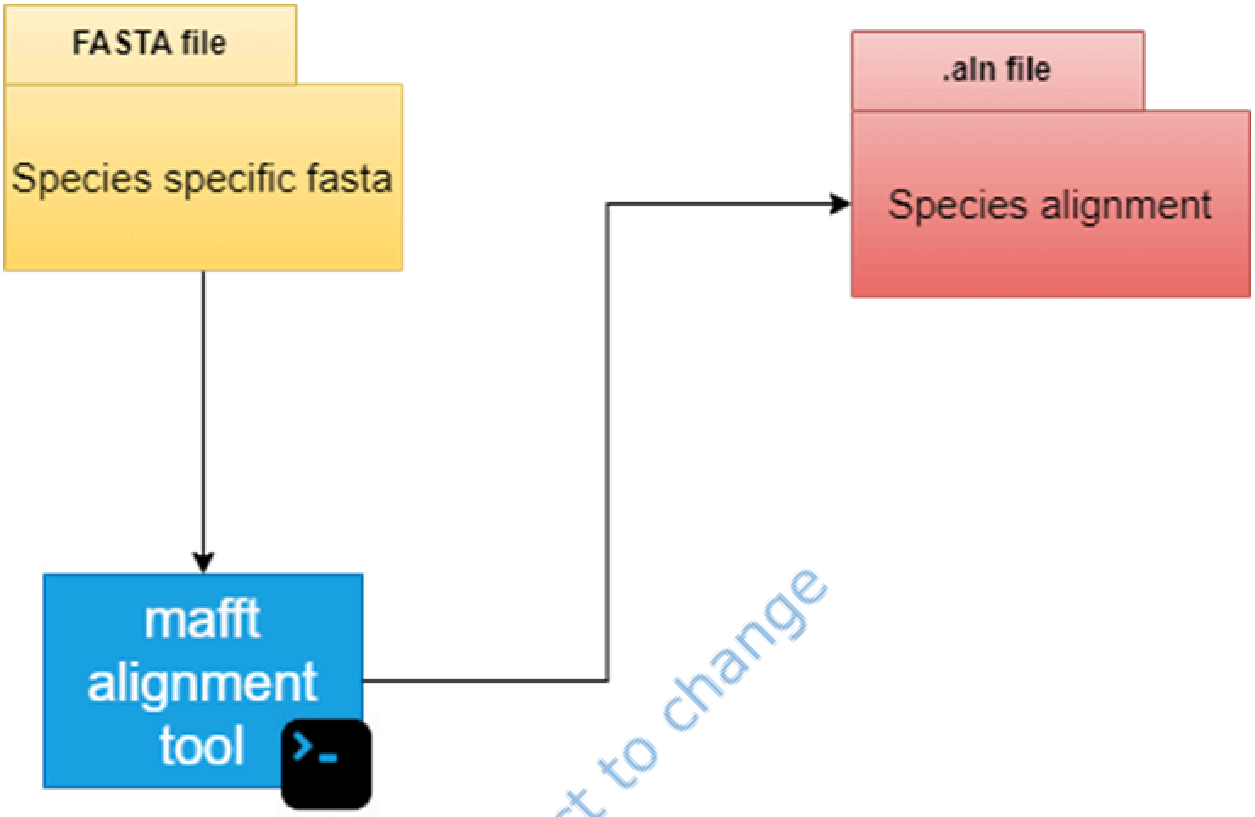

Figure 3. Deployment of the MAFFT algorithm to align sequences in the FASTA file to each other.

INPUT:

- Format: FASTA file
- Description: FASTA file containing all sequences, both from own results and an annotation database, for a specific species.
- Source: FASTA file from Step 1.

OUTPUT:

- Format: .aln file
- Description: One alignment file per species is created. It contains the result of the alignment calculated by the command tool *mafft*. To be used after in the next steps, a specific output format is selected: clustal format.

SERVICE:

- Name: MAFFT
- Type: Command line tool
- Description: MAFFT is a multiple sequence alignment program for Unix-like operating systems. It offers a range of multiple alignment methods. The following command was used in this workflow.
- Availability: The tool is freely available online: https://mafft.cbrc.jp/alignment/software/
- Author: Kazutaka Katoh, Kazuharu Misawa, Kei‐ichi Kuma, Takashi Miyata

STEP 3: Haplotype network, geographic maps, and summary statistics

In the last step, the intraspecific genetic variation of each species is visualized as a haplotype network. In addition, the origin of the sequences is shown on a geographic map and summary statistics are saved in a separate file.

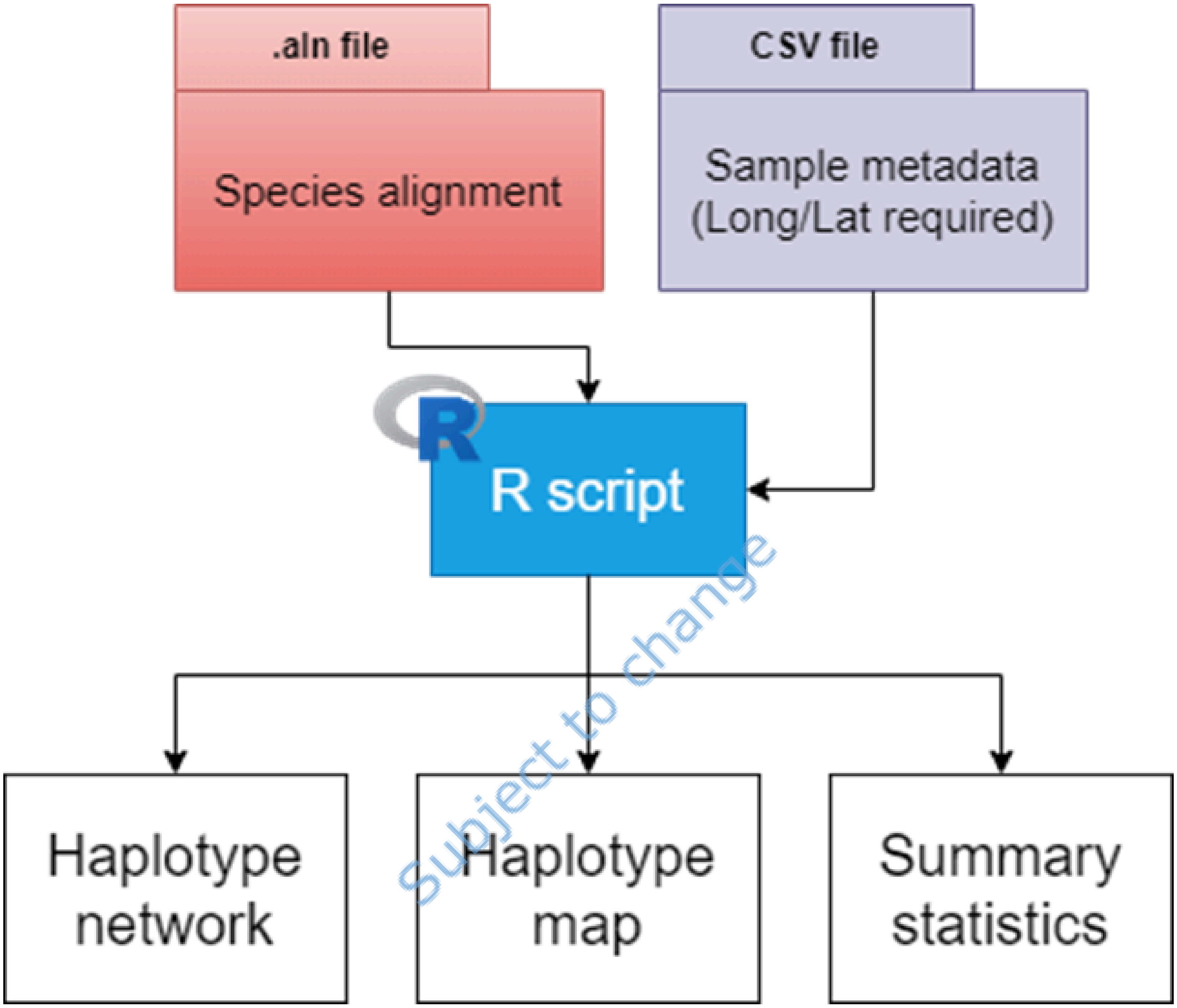

Figure 4. Workflow to generate haplotype networks from sequence alignment files as well as a map documenting the geographic origin of the DNA sequences and relevant summary statistics.

INPUT1:

- Format: .aln file
- Description: Alignment file, one per species of interest, with a CLUSTAL type of format.
- Source: Alignment file obtained from steps 2.

INPUT2:

- Format: .csv file

- Description: csv file containing the metadata related to the sequences used for the analysis. This file must contain the following column: sample_ID, Longitude, Latitude and locality where the sequences were taken.

- Source: From the users.

OUTPUT 1:

- Format: picture (PNG)

- Description: Picture of the haplotype network obtained. One picture per species of interest.

OUTPUT 2:

- Format: picture (PNG)

- Description: Map representing the haplotype detected on location.

OUTPUT 3:

- Format: text file

- Description: Summary statistics of the alignment and haplotype network.

SERVICE:

- Name: haplotype.R

- Type: R-script

- Description: R-script that produces several outputs from the alignment obtained in step 2. It can be run from the command line and takes 5 arguments: (a) the path to the alignment file and the sample metadata file; (b) the directory in which the results must be saved; (c) the length of the marker used and if sequences from a database were used during the alignment. For now, if database sequences are present, no map can be created as no Longitude/Latitude can be associated with them. It uses two important R-packages: *ape* and *pegas*. It produces a haplotype network, a map representing where the haplotype detected can be found and finally, a text file with some information regarding the alignment. The following command can be used:

1.
   ```
   Rscript --vanilla haplotype.R -s Acartia_tonsa.aln -m sample_location.csv -o ./haplotype_results/ -l 300 -d FALSE
   ```

2.
   ```
   Rscript --vanilla haplotype.R --sp_alignment=Acartia_tonsa.aln --metadata_sample=sample_location.csv --output=./haplotype_results/ --length_marker=300 --use_db=FALSE"
   ```

- Availability: R-script not yet publicly available. It will be open to public access in the future in the LifeWatch ERIC GitLab repository. At this moment it is available on demand.

- Author: Marie-Catherine Bouquieaux, Pascal Hablützel

3. Integration of ANERIS OMB Product 1 Workflow into LifeWatch ERIC's Infrastructure

The integration of the ANERIS OMB Product 1 workflow into the LifeWatch ERIC infrastructure represents a multifaceted process designed to ensure compatibility and composability with existing systems while preserving the integrity of the original code. The workflow was integrated into the MyLifeWatch platform - https://my.lifewatch.eu and is accessible both as a main workflow component and through the workflow editor as a wrapper. This document outlines the detailed steps taken to achieve this integration.

3.1 Objectives of the Integration

The primary goals of integrating the ANERIS OMB Product 1 workflow were twofold:

1. **Compatibility with LWE's Infrastructure**: Adapting the workflow to align with LWE standards for structured input, output, and execution processes.

2. **Preservation of Original Code**: Ensuring minimal modifications to the original code to maintain consistency and accuracy in results.

In addition to these goals, the integration process emphasized security, modularity, and adaptability to future improvements.

3.2 Creation of the Workflow Component Docker Image

The first step involved analyzing the original code provided by the workflow developers. This analysis focused on identifying and documenting the dependencies required for seamless execution. A Docker image was then created to encapsulate these dependencies as shown in Annex I.

- **Minimization of Image Size**: To enhance efficiency, the Docker image was designed to include only the essential components. Redundant data and unnecessary software were excluded to reduce the image size, thereby optimizing performance and minimizing potential security vulnerabilities.

- **Security Measures**: By isolating the required dependencies within the Docker environment, the risk of conflicts and vulnerabilities in the broader system was significantly reduced.

3.3 Development of the Annotation File

An essential component of the integration process was the creation of a JSON-based annotation file. This file serves as the semantic definition of the workflow and provides detailed metadata to ensure proper execution.

- **Structured Parameter Definition**: The annotation file included all necessary parameters, such as the input and output types, workflow name, description, current version, and license information.

- **Collaboration with Researchers**: Researchers provided the foundational information in the required format, enabling the creation of a robust annotation file tailored to LWE's standards.

3.4 Creation of the Main Execution Script

A Bash-based execution script was developed to orchestrate the workflow's operation within the Docker container. This script ensures a streamlined process for validation, execution, and output verification.

- **Input Validation**: The script performs checks to verify the presence and format of all input files before proceeding with execution.

- **Parameter Handling**: The script manages the workflow's sole parameter, use_database, which determines whether the workflow operates in database-dependent or independent mode.

- **Execution Logic**: Depending on the use_database parameter value, the script executes one of two workflow versions, ensuring flexibility and adherence to user requirements.

- **Output Verification**: After execution, the script validates all output files to confirm successful completion and proper formatting.

3.5 Local Debugging and Testing

To facilitate an agile development process, initial testing was conducted in LWE’s local environment. This environment replicates the cloud-based infrastructure and allows for iterative improvements.

- **Issue Resolution**: Minimal issues were identified during local testing and were promptly addressed in collaboration with researchers.
- **Use of Example Data**: Researchers provided example input and output files, which were invaluable for verifying workflow functionality and expected behavior.

3.6 Deployment and Testing in Development Environment

Following successful local testing, the workflow was migrated to LWE’s development environment for further validation.

- **GitLab Integration**: The workflow component was merged into LWE’s GitLab repository, enabling version control and automated testing.The code will be available during the lifetime of the project.
- **Docker Registry Deployment**: The Docker image was uploaded to LWE’s Docker registry, making it available only for deployment. At this moment this repository is private due to organizational reasons but it will be partially opened to public access in the future.
- **Web-Based Testing**: The component was tested in the MyLifeWatch development site (https://my.lifewatch.dev/) to ensure compatibility and functionality across different input and parameter configurations.

3.7 Deployment in Production Environment

After rigorous testing in the development environment, the workflow was promoted to the production environment.

- **Automated Deployment**: The workflow was automatically deployed to the production environment following successful integration testing.
- **Final Validation**: Additional tests were conducted to verify stability and reliability in a live setting.

3.8 Continuous Development and User Feedback

The workflow’s integration marks the beginning of an iterative improvement process driven by user feedback and performance metrics.

- **Feedback Integration**: Insights from users and researchers are continuously gathered to identify areas for enhancement.

- **Version Control**: Updates are managed through version tags, ensuring accessibility to previous workflow iterations and facilitating reproducibility.

4. Conclusions

This report details the development, testing, integration, and deployment of the ANERIS OMB Product 1 **Intraspecific Variation**, into the LifeWatch ERIC infrastructure, a critical step for advancing genomic data analysis in marine biology.

Integrating the Intraspecific Variation workflow into LifeWatch ERIC's established framework enhances researchers´ capacity to explore and understand marine biodiversity at a crucial time when global biodiversity faces significant threats from human activity. It offers a powerful tool that is both accessible and versatile.

It marks a significant milestone in the ANERIS project. It exemplifies how collaborative, funded research can lead to meaningful advancements in environmental science and policy, paving the way for substantial impacts on our understanding and preservation of marine life.

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

H. Li, Seqtk: a fast and lightweight tool for processing FASTA or FASTQ sequences, 2013. Available online at: https://github.com/lh3/seqtk.git

Kazutaka Katoh, Kazuharu Misawa, Kei‐ichi Kuma, Takashi Miyata, MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform, Nucleic Acids Research, Volume 30, Issue 14, 15 July 2002, Pages 3059–3066, https://doi.org/10.1093/nar/gkf436

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, 4, 894–907. https://doi.org/10.1002/edn3.303

Annex 1. Workflow code

Creation of the Workflow Component Docker Image

The workflow code was analyzed, and a Docker image was created to encapsulate all necessary dependencies. Efforts were made to minimize the image size, ensuring only essential components were included.

1. **Development of the Annotation File**

An annotation file was created to define parameters, inputs, outputs, and metadata. This structured JSON file ensures seamless integration and compatibility with the LifeWatch ERIC platform.

Annotation file content:

```
{
  "name": "AnerisIntraspecificVariation",
  "label": "ANERIS Intraspecific Variation",
  "description": "This is the Intraspecific Variation workflow of the ANERIS project.",
  "type": "DataAnalysing",
  "dockerImage": "lw-r-wrapper-aneris-omb-products",
  "parameters": [
   {
    "name": "use_database",
    "label": "Use Database",
    "description": "Whether or not to use the database",
    "defaultValue": "FALSE",
        "type": "Boolean"
      }
    ],
    "inputs": [
      {
        "type": "Fastq",
        "name": "database_sequences_fasta",
        "label": "Database Sequences Fasta",
        "path": "/mnt/inputs/database_sequences.fasta",
        "description": "The annotation database FASTA should be the same database used to
annotate the sequences of the samples. It consists of all the sequences of the database and
their id."
      },
      {
        "type": "TabularDataSet",
        "name": "marker_csv",
        "label": "Marker CSV",
        "path": "/mnt/inputs/marker.csv",
        "description": "The sample CSV file is a result file from an annotation workflow. It contains
the  taxonomic information related to the sequences as well as information linking the
sequences  to its original sample. I must contain at least a column called “species”, with the
species name associated with the sequence."
      },
      {
        "type": "Fastq",
        "name": "marker_sequences_fasta",
        "label": "Marker Sequences Fasta",
        "path": "/mnt/inputs/marker_sequences.fasta",
        "description": "The FASTA file is a result file from an annotation workflow. The FASTA file
contains all the sequences from all the samples used in the annotation workflow. Each
sequence is associated with a specific id. This id must contain at the end an underscore
followed by the name of the sample associated with that sequence (e.g.:
sequenceID_samplename)."
      },
      {
        "name": "metadata_file",
        "label": "Metadata CSV File",
        "description": "CSV file containing information about the sample (required information:
Longitude/Latitude)",
        "path": "/mnt/inputs/metadata.csv",
        "type": "TabularDataSet"
      }
```

```
  ],
  "outputs": [
    {
      "type": "Zip",
      "label": "Extraction Results",
      "name": "extraction_result",
      "path": "/mnt/outputs/extraction_result.zip",
      "description": "Zip file containing one FASTA file per species identified in the samples"
    },
    {
      "type": "Zip",
      "label": "Extraction Temporary Files",
      "name": "extraction_temp",
      "path": "/mnt/outputs/extraction_temp.zip",
      "description": "Temporary files are created during the workflow; they are not used after.
They  are kept in case check needs to be done if an anomaly is detected. They consist in four
different files per species: the list of sequences to be extracted from the result FASTA, a list of
sequence to be extract from the database, a FASTA file containing the sequences extracted
from the result FASTA and a FASTA file containing the sequences extracted from the database"
    },
    {
      "type": "Zip",
      "label": "Alignment Results",
      "name": "alignment_result",
      "path": "/mnt/outputs/alignment_result.zip",
      "description": "Zip file with one alignment file per species. They contain the result of the
alignment calculated by the command tool mafft. A specific output format is selected: clustal
format."
    },
    {
      "type": "Zip",
      "label": "Haplotype Check Alignment Output",
      "name": "haplotype_check_alignment_output",
      "path": "/mnt/outputs/haplotype_check_alignment_output.zip",
      "description": "Zip file containing txt files with info about checking alignment process"
    },
    {
      "type": "Zip",
      "label": "Haplotype Check Alignment Plots",
      "name": "haplotype_check_alignment_plot",
      "path": "/mnt/outputs/haplotype_check_alignment_plot.zip",
      "description": "Zip file containing png files with info about checking alignment plots"
    },
    {
      "type": "Zip",
      "label": "Haplotype Network Plots",
      "name": "haplotype_network_plot",
      "path": "/mnt/outputs/haplotype_network_plot.zip",
      "description": "Zip file containing png files of haplotype networks (graphical representation
that illustrates the relationships between different haplotypes—a group of closely linked genetic
markers inherited together—within a population or species)"
    },
    {
      "type": "Zip",
      "label": "Haplotype Map Plots",
      "name": "haplotype_map_plot",
      "path": "/mnt/outputs/haplotype_map_plot.zip",
      "description": "Zip file containing png files of haplotype map (haplotype map, or HapMap is
a tool that allows researchers to find genes and genetic variations that affect health and
disease)"
    }
  ],
  "resources": {
    "cores": 2,
    "memory": 512,
    "gpuNeeded": false,
    "gpuMemory": 1024,
    "estimatedTime": 4
  },
  "tags": [ ],
  "license": "GPL v3",
  "version": "0.0.1",
  "dependencies": [],
  "publicationDate": "Mon, 18 Nov 2024 13:00:00 GMT",
  "author": "LifeWatch ERIC",
  "bugs": {
    "email": "julio.paneque@lifewatch.eu",
    "url": "https://helpdesk.lifewatch.eu/"
  },
  "citation": null,
  "testPath": "emptyTest.sh",
  "metaDataCatalogueUrl":
```

```
"https://metadatacatalogue.lifewatch.eu/srv/eng/catalog.search#/metadata/<>"
  }
```

2. Creation of the Main Execution Script

The main execution script was developed to validate inputs, manage execution based on parameters, and verify outputs. This script orchestrates the workflow's operations within a Docker container.

Execution script content:

```
#! /bin/bash

# Helper function
throwError() { echo "ERROR: $@" 1>&2; exit 1; }

# Check input files
if [ ! -f /mnt/inputs/database_sequences.fasta ]; then
   throwError "No database sequences file found."
fi

if [ ! -f /mnt/inputs/marker.csv ]; then
   throwError "No marker file found."
fi

if [ ! -f /mnt/inputs/marker_sequences.fasta ]; then
   throwError "No marker sequences file found."
fi

if [ ! -f /mnt/inputs/metadata.csv ]; then
   throwError "No metadata file found."
fi

#  Read parameters
USE_DATABASE=$(echo "$@" | grep -oP '(?<=--use_database=)\S+')
USE_DATABASE=${USE_DATABASE:-FALSE}
echo "Use database: $USE_DATABASE"

if [ "$USE_DATABASE" = "TRUE" ]; then
   OPTIONAL_DATABASE_SEQUENCES="-d /mnt/inputs/database_sequences.fasta"
   NUMBER_SEQUENCES=5
else
   OPTIONAL_DATABASE_SEQUENCES=""
```

```
    NUMBER_SEQUENCES=2
fi

# TEMP: Prepare output folders
rm -rf /mnt/outputs/*
mkdir -p /mnt/outputs/extraction/
mkdir -p /mnt/outputs/haplotype_results/


# Get FASTA-file for each species
Rscript --vanilla ./code/get_extract.R -r /mnt/inputs/marker.csv -f
/mnt/inputs/marker_sequences.fasta $OPTIONAL_DATABASE_SEQUENCES -o
/mnt/outputs/extraction/ -s $NUMBER_SEQUENCES

# Get alignment
for infile in /mnt/outputs/extraction/alignment/*_toalign.fasta
do
    base=$(basename ${infile} _toalign.fasta)
    ./.software/mafft-linux64/mafft.bat --localpair --maxiterate 1000 --clustalout ${infile} >
/mnt/outputs/extraction/alignment/${base}_alignment.aln
done

# Get haplotype network
for infile in /mnt/outputs/extraction/alignment/*_alignment.aln
do
    base=$(basename ${infile} _alignment.aln)
    Rscript --vanilla ./code/haplotype.R -s ${infile} -m /mnt/inputs/metadata.csv -o
/mnt/outputs/haplotype_results/ -l 300 -d $USE_DATABASE
done

# Zip results
zip -q -j /mnt/outputs/extraction_result.zip /mnt/outputs/extraction/alignment/*.fasta
zip -q -j /mnt/outputs/extraction_temp.zip /mnt/outputs/extraction/temp/*
zip -q -j /mnt/outputs/alignment_result.zip /mnt/outputs/extraction/alignment/*.aln
zip -q -j /mnt/outputs/haplotype_check_alignment_output.zip
/mnt/outputs/haplotype_results/check_alignment/*.txt
zip -q -j /mnt/outputs/haplotype_check_alignment_plot.zip
/mnt/outputs/haplotype_results/check_alignment/*.png
zip -q -j /mnt/outputs/haplotype_network_plot.zip /mnt/outputs/haplotype_results/network/*.png

if [ "$USE_DATABASE" = "FALSE" ]; then
    zip -q -j /mnt/outputs/haplotype_map_plot.zip /mnt/outputs/haplotype_results/map/*.png
```

```
fi

# Check output files
if [ -z "$(ls -A /mnt/outputs/extraction_result.zip)" ]; then
    throwError "The extraction_result.zip file was not created correctly."
fi

if [ -z "$(ls -A /mnt/outputs/extraction_temp.zip)" ]; then
    throwError "The extraction_temp.zip file was not created correctly."
fi

if [ -z "$(ls -A /mnt/outputs/alignment_result.zip)" ]; then
    throwError "The alignment_result.zip file was not created correctly."
fi

if [ -z "$(ls -A /mnt/outputs/haplotype_check_alignment_output.zip)" ]; then
    throwError "The haplotype_check_alignment_output.zip file was not created correctly."
fi

if [ -z "$(ls -A /mnt/outputs/haplotype_check_alignment_plot.zip)" ]; then
    throwError "The haplotype_check_alignment_plot.zip file was not created correctly."
fi

if [ -z "$(ls -A /mnt/outputs/haplotype_network_plot.zip)" ]; then
    throwError "The haplotype_network_plot.zip file was not created correctly."
fi

if [ "$USE_DATABASE" = "FALSE" ]; then
    if [ -z "$(ls -A /mnt/outputs/haplotype_map_plot.zip)" ]; then
        throwError "The haplotype_map_plot.zip file was not created correctly."
    fi
fi
```

1. Additional Files

3. Dockerfile

Dockerfile content:

```
FROM rocker/r2u:24.04

RUN apt-get update \
    && apt-get install -y \
    gcc \
    seqtk \
    curl \
    openssl \
    libcurl4-openssl-dev \
    libssl-dev \
    libgdal-dev \
    libgeos-dev \
    libproj-dev \
    libudunits2-dev \
    proj-bin \
    r-cran-magrittr \
    r-cran-doparallel \
    r-cran-data.table \
    r-cran-stringr \
    r-cran-tidyr \
    r-cran-optparse \
    r-cran-dplyr \
    r-cran-ggplot2 \
    r-cran-sf \
    r-cran-cowplot \
    r-cran-cowplot \
    r-cran-rjson \
    r-cran-zip \
    r-cran-ape \
    r-cran-pegas \
    r-cran-rnaturalearth \
    r-cran-rnaturalearthdata

WORKDIR /aneris

COPY . .
RUN chmod -R +x /aneris/.software

ENTRYPOINT ["/bin/bash", "/aneris/docker-entrypoint.sh"]
```

4. LICENSE

LICENSE file content:

5. .gitignore

.gitignore content:

```
*.Rproj
.Rhistory
.RData
.Rproj.user
data/
```

6. .dockerignore

.dockerignore content:

```
*
!docker-entrypoint.sh
!.software
```

!code

7. README.md

README.md content:

```
# OMB products

## Summary products to be developed

- Biodiversity
- NIS
- Intraspecific variation

## Data folder

Contains example data that can be used to test the code before using your own data. It also provides a description of the fields that are mandatory for the analysis.

## Code folder

Contains the code for each OMB product as well as the workflow that was used to identify the species present in the raw FASTA files.

## Schema folder

Contains the .drawio files, each representing a schematic workflow for each OMB product. The detail of the workflow used to process the raw FASTA files into COI.fasta and COI.csv files is also present.
```