Optimal Representative Strain Selector - User Manual

# Table of Contents

1. Introduction

2. Installation Guide

3. Usage Instructions

- General Command Format

- Options

4. Pipeline Overview

- Genomic Score

- Ecological Score

- Functional Score

5. Example Run

6. Troubleshooting

7. LAORS

# Introduction

The Optimal Representative Strain Selector (ORS) is a pipeline designed for the comprehensive comparative analysis of genomes, focusing on a custom set of input genomes belonging to the same bacterial species. The pipeline performs preliminary quality filtering and redundancy reduction, selecting high-quality and representative genomes for further comparison.  
  
The ORS generates three distinct scores:

- Genomic Score: Based on double normalized ANI scores between selected genomes.  
- Ecological Score: Determined by double tracking selected genomes in a defined environment.  
- Functional Score: Derived from comparisons of functional gene families among selected genomes.

These three scores are then merged into the final output file Representative\_Score.txt. The merged score is reported in the last column named “Representative\_Score”.

# Installation Guide

### Prerequisites  
  
- Ubuntu 24.04

- Miniconda  
  
  
### Installation Steps  
  
1. Download the ORS\_Installed.sh file to a desired directory.  
2. Navigate to the extracted directory in the terminal.  
3. Run the installation script:  
  
```bash  
./ORS\_Installer.sh  
```  
  
This installer will copy all the requested software to `$HOME/Documents/Optimal\_Representative\_Strain/` and will install all dependencies in a dedicated miniconda environment named `RSS\_selector`.  
  
4. Follow the on-screen instructions to complete the installation.

# Usage Instructions

### General Command Format  
  
```bash  
ORS\_Local.sh -P [projectname] -T [threadsnumber] -t [thresholdnumber] -F [Yes/No] -R [Yes/No] -X [Yes/No] -D [Path/To/Folder/Datasets] -G [Path/To/Folder/Genomes]  
```  
  
### Options  
  
- `-P` or `--PROJECT`: Name assigned to this analysis.  
- `-T` or `--THREADS`: Number of threads to use for this analysis (typically 16/32/64).  
- `-t` or `--THRESHOLD`: Minimum number of mapped reads required to consider a strain tracked (suggested 20,000).  
- `-G` or `--GENOMES\_PATH`: Path to the directory containing input genomes. If not declared, input genomes must be placed inside the automatically created `input\_genomes` directory within the main directory.  
- `-D` or `--DATASETS\_PATH`: Path to the directory containing input datasets. If not declared, input datasets must be placed inside the automatically created `input\_datasets` directory within the main directory.  
- `-F` or `--FILTERING`: Removal of tRNA and rRNA from datasets to optimize mapping analysis (suggested Yes).  
- `-E` or `--ENVIRONMENT`: Type of environment analyzed (e.g., Adult\_Gut, Dairy, Infant\_Gut).  
- `-R` or `--REDUNDANT`: Reduction of genome nucleotide redundancy by collapsing too similar genomes, retaining only the best one (suggested Yes). If requested, perform dereplication of input genomes in order to reduce redundancy and speed up analysis. A table with all the details regarding removed genomes and their association with retained Representatives is available in the output file Dereplicated\_Genomes\_Table.tsv.  
- `-X` or `--SKIPDATABASE`: Advanced option for redoing the analysis (suggested No).  
- `-h`: Display the help message.  
  
### Example Command  
  
```bash  
ORS\_Local.sh -P TestName -T 32 -t 20000 -F Yes -R Yes -X No -D Path/To/Folder/Datasets -G Path/To/Folder/Genomes  
```

NOTE: for species typically found at low abundance, a lower -t value is suggested (e.g. 1000).

# Pipeline Overview

The ORS pipeline consists of multiple phases:  
  
1. \*\*Setup and Optional Dereplication Phase\*\*: Initializes the project and validates the provided options. If requested, perform dereplication of input genomes in order to reduce redundancy and speed up analysis. A table with all the details regarding removed genomes and their association with retained Representatives is available in the output file Dereplicated\_Genomes\_Table.tsv.  
2. \*\*Mapping Analysis\*\*: Maps reads to the input genomes, performing filtering and redundancy reduction.  
3. \*\*Ecological Analysis\*\*: Evaluates the ecological relevance of the genomes.  
4. \*\*Functional Analysis\*\*: Compares functional gene families among the genomes.  
5. \*\*Representative Strain Selection\*\*: Identifies the optimal Representative strain based on the calculated scores.  
  
### Genomic Score  
  
The Genomic Score is obtained through double normalized ANI (Average Nucleotide Identity) scores between the selected genomes, ensuring that only high-quality and representative genomes are compared.  
  
### Ecological Score  
  
The Ecological Score is determined by tracking the presence of selected genomes within a defined environment using the provided metagenomic datasets.  
  
### Functional Score  
  
The Functional Score is derived from comparisons of functional gene families among the selected genomes, highlighting the functional diversity and similarities.

# Example Run

To execute the ORS pipeline, use the following command:  
  
```bash  
ORS\_Local.sh -P MyProject -T 32 -t 20000 -F Yes -R Yes -X No -D /path/to/datasets -G /path/to/genomes  
```

# Troubleshooting

- Ensure all input paths are correct and accessible.  
- Verify that the `RSS\_selector` conda environment is activated before running the script.  
- Check the `Log.txt` file for detailed error messages and status updates.

# Local Optimal Representative Selector Software

The Local Optimal Representative Selector (LORS) is a tool with graphical UI designed to facilitate the selection and verification of optimal Representative strains amongst a set of locally available strains for which a genome is available. The user must select an environmental database created through the ORS software and locally available genomes will be checked against the Representative previously determined by the ORS software. This software provides a user-friendly interface to manage genome comparisons and ensure high-quality and representative genomes are selected as close-to-Representative strains for specific environments.

### Functionality:

1. Selecting Genome Files:

- Users can select a folder containing their genome files in FASTA format.

- The selected files are displayed in a list for verification.

2. Choosing an Environmental Database:

- Users can choose from a list of pre-existing environmental databases.

- Alternatively, users can specify a custom database path.

3. Defining a Threshold:

- A checkbox option to define thresholds for newly obtained Representative genomes.

- Recommended to use this option if the environmental database is new or additional species have been added.

4. Running the Analysis:

- Upon clicking the "Submit" button, the software processes the selected genome files against the chosen environmental database.

- If the threshold option is selected, it runs additional scripts to enforce the defined thresholds.

How to Use

1. Launch the Application

- Run the Local Optimal Representative Selector GUI script.

2. Select Genome Folder

- Click on the "Browse" button to select the folder containing your genome files.

3. Choose Environmental Database:

- Select a database from the dropdown menu or click "Custom" to provide a custom path.

4. Request Threshold calculation

- Check the "Threshold" option if necessary.

5. Submit for Analysis

- Click the "Submit" button to start the analysis process.

- Monitor the progress and check for any popup messages indicating the status or errors.

6. Interrupt if Needed

- Click the "Interrupt" button to stop the process at any time.

This tool streamlines the workflow for selecting optimal Representative strains, making it accessible even to users with limited bioinformatics expertise. By automating critical steps and providing clear user prompts, the Local Optimal Representative Selector ensures accurate and efficient genome comparisons tailored to specific environmental contexts.

The output is available as a