SingleM and Lyrebird Analysis Protocol

> **Note to Users**: This protocol serves as a reference guide to help you understand the data and potential analysis workflows. You are strongly encouraged to explore the official documentation for SingleM and Lyrebird to gain a deeper understanding of the tools. Additionally, while the R code provided illustrates the general analytical approach, it may require adaptation for your specific dataset. Always refer to the official tidyverse and miaverse documentation for the most accurate and up-to-date syntax.

1. Introduction

This protocol is designed to guide researchers in understanding and analyzing metagenomic taxonomic profiles generated by **SingleM** (targeting Bacteria and Archaea) and **Lyrebird** (targeting dsDNA phages/Caudoviricetes).

1.1 Principles of SingleM

SingleM is a tool for the rapid identification of microbial species and estimation of their relative abundances in metagenomic data. * **Core Principle**: Unlike whole-genome alignment methods, SingleM utilizes a set of highly conserved **Single Copy Marker Genes**. * **Advantage**: It is particularly powerful at detecting "novel" species not represented in reference databases. It aligns reads to Hidden Markov Models (HMMs) of marker genes, extracts sequences covering specific windows, and clusters them into Operational Taxonomic Units (OTUs). * **Reference**: Woodcroft et al., *Nature Biotechnology* (2025). "Comprehensive taxonomic identification of microbial species in metagenomic data using SingleM and Sandpiper".

1.2 Principles of Lyrebird

Lyrebird is a specialized module within the SingleM framework designed for profiling **dsDNA phage (Caudoviricetes)** communities. * **Core Principle**: Since phages lack the universal single-copy marker genes found in bacteria, Lyrebird employs an expanded set of **500+ marker genes**. * **Taxonomy**: It uses a custom taxonomy based on vConTACT3 clustering, enabling the identification of both known and unknown phages. * **Advantage**: Optimized for viral sequences in metagenomes, it captures significantly more viral diversity than traditional contig-assembly approaches. * **Reference**: "Lyrebird: a tool for profiling dsDNA phage communities in metagenomic data. (in preparation)".

---

2. Interpretation of Output Files

The pipeline generates a comprehensive set of files for each sample (e.g., `p0086`). Below is a detailed interpretation of each file type, applicable to both `singleM` and `lyrebird` outputs.

2.1 Primary Profile & Visualization

File Name	Description & Usage
`*_taxonomic_profile.tsv`	**Taxonomic Profile**. The primary output listing detected taxa and their estimated relative abundances. This is the most direct summary of "who is there and in what proportion." Generated by `singlem summarise --output-taxonomic-profile`.
`*_taxonomic_profile_long.tsv`	**Detailed Taxonomic Profile**. A long-format version of the profile containing additional quality control metrics, such as sequence coverage, standard deviation of coverage, and number of reads mapped. Generated by `singlem summarise --output-taxonomic-profile-with-extras`.
`*_taxonomic_krona.html`	**Krona Chart**. An interactive HTML visualization displaying the taxonomic hierarchy in a sunburst chart. Useful for intuitive exploration of community composition. Generated by `singlem summarise --output-taxonomic-profile-krona`.

2.2 OTU Tables (Sequence Level)

File Name	Description & Usage
`*_combined_OTU.tsv`	**Raw OTU Table**. Contains counts of unique marker gene sequences (OTUs) detected in the sample. This represents the finest resolution of data. Generated by `singlem summarise --output-otu-table`.
`*_clustered_OTU.tsv`	**Clustered OTU Table**. OTUs are clustered based on sequence similarity (approximating species-level units). This reduces data sparsity and is often used for community analysis. Generated by `singlem summarise --cluster --output-otu-table`.
`*_clustered_OTU_details.tsv`	**Cluster Details**. A mapping file showing which original raw OTUs were grouped into which cluster. Useful for tracing specific sequences. Generated by `singlem summarise --cluster --clustered-output-otu-table`.

2.3 Analysis-Ready Files (For R/Downstream)

These files are typically generated by post-processing scripts (e.g., `singleM_profile2otu.py`) to facilitate import into analysis tools like R.

File Name	Description & Usage
`*_OTU_abundance.tsv`	**Abundance Matrix**. A formatted table where rows are OTUs and columns are samples (or abundance counts). This is the primary input for the `assay` slot in R objects.
`*_OTU_taxonomy.tsv`	**Taxonomy Table**. Maps each OTU ID to its full taxonomic lineage (Domain, Phylum, Class, Order, Family, Genus, Species). This is the input for the `rowData` slot in R objects.
`*_OTU_tree.nwk`	**Phylogenetic Tree**. A Newick-formatted tree representing the evolutionary relationships between OTUs. Essential for calculating phylogenetic diversity metrics (e.g., UniFrac).

2.4 Aggregated Taxonomic Tables

File Name	Description & Usage

`*-domain.tsv`
`*-phylum.tsv`
`*-class.tsv`
`*-order.tsv`
`*-family.tsv`
`*-genus.tsv`
`*-species.tsv`
	Rank-Specific Abundance Tables. These files contain relative abundance data aggregated at specific taxonomic ranks. They are ready-to-use for generating stacked bar plots or performing differential abundance analysis at a specific level (e.g., "How do Phyla differ between groups?").

3. Downstream Analyses Protocol (in R)

This section guides you through analyzing the data using modern R packages: `tidyverse` for data manipulation and `mia` (Microbiome Analysis) for storing and analyzing microbiome data using the `TreeSummarizedExperiment` (TSE) container.

3.1 Prerequisites

```r
# Install packages if not already installed
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install(c("mia", "miaViz", "scater", "TreeSummarizedExperiment"))
install.packages("tidyverse")

# Load libraries
library(tidyverse)
```

```
library(mia)
library(miaViz)
library(scater)
```

3.2 Data Import & TSE Construction

We will construct a `TreeSummarizedExperiment` (TSE) object using the analysis-ready files.

```
# Define file paths (adjust paths as necessary)

otu_file <- "p0086-singleM_combined_OTU.tsv" # or _OTU_abundance.tsv
tax_file <- "p0086-singleM_OTU_taxonomy.tsv"
tree_file <- "p0086-singleM_OTU_tree.nwk"

# 1. Load OTU/Abundance Data
# Ensure the data is numeric and row names are OTU IDs
otu_data <- read_tsv(otu_file) %>%
  column_to_rownames("OTU_ID") %>%
  as.matrix()

# 2. Load Taxonomy Data
# Ensure row names match OTU IDs in the abundance data
tax_data <- read_tsv(tax_file) %>%
  column_to_rownames("OTU_ID") %>%
  DataFrame() # Convert to DataFrame for TSE compatibility

# 3. Load Phylogenetic Tree
tree_data <- ape::read.tree(tree_file)

# 4. Create TreeSummarizedExperiment Object
tse <- TreeSummarizedExperiment(
  assays = list(counts = otu_data),
  rowData = tax_data,
  rowTree = tree_data
)

# Inspect the object
print(tse)
```

3.3 Basic Analyses & Visualization

A. Alpha Diversity (Within-Sample Diversity)

Measures the richness and evenness of species within a single sample.

```
# Calculate Shannon Index and Observed Features
tse <- mia::estimateDiversity(tse, assay.type = "counts", index = "shannon")
tse <- mia::estimateRichness(tse, assay.type = "counts", index = "observed")

# Visualize (assuming you have sample metadata in colData)
# If you have metadata, merge it: colData(tse) <- cbind(colData(tse), metadata)
colData(tse) %>%
  as.data.frame() %>%
  ggplot(aes(x = "Sample", y = shannon)) +
  geom_boxplot() +
  geom_jitter(width = 0.2) +
  theme_bw() +
  labs(title = "Alpha Diversity (Shannon Index)", y = "Shannon Index")
```

B. Beta Diversity (Between-Sample Diversity)

Measures how different the community structure is between samples.

```
# Transform counts to relative abundance
tse <- transformAssay(tse, assay.type = "counts", method = "relabundance")

# Calculate Bray-Curtis distance and perform PCoA
tse <- runMDS(tse, FUN = vegan::vegdist, method = "bray", assay.type = "relabundance", name = "PCoA_BC"

# Plot PCoA
plotReducedDim(tse, dimred = "PCoA_BC") +
  theme_bw() +
  labs(title = "PCoA (Bray-Curtis Dissimilarity)")
```

C. Taxonomic Composition

Visualizing the abundance of major taxa.

```
# Agglomerate data to Phylum level
tse_phylum <- agglomerateByRank(tse, rank = "Phylum")

# Transform to relative abundance
tse_phylum <- transformAssay(tse_phylum, assay.type = "counts", method = "relabundance")

# Plot top 10 most abundant Phyla
plotAbundance(tse_phylum, rank = "Phylum", order_rank_by = "abundance", n = 10) +
  labs(title = "Top 10 Phyla Composition")
```

---

4. Biological Questions Addressed

By applying SingleM and Lyrebird pipelines, you can address fundamental ecological questions:

1. **Community Composition**: What is the taxonomic makeup of the microbial (SingleM) and viral (Lyrebird) communities? Are they dominated by known or novel taxa?
2. **Diversity Gradients**: How does microbial/viral diversity change across different environments or treatments?
3. **Novelty Discovery**: SingleM and Lyrebird excel at detecting uncultured organisms. You can quantify the proportion of the community that represents novel lineages (e.g., “What percentage of the viral community is unknown?”).
4. **Host-Virus Dynamics**: By integrating SingleM (host) and Lyrebird (virus) profiles, you can explore potential interactions. For example, does the abundance of specific phage groups correlate with specific bacterial hosts?
5. **Biomarker Identification**: Which specific taxa are significantly associated with a particular environmental condition or disease state?