CloVR-ITS: Automated ITS amplicon sequence analysis pipeline for the characterization of fungal communities – standard operating procedure, version 1.0

James Robert White, the CloVR team, Owen White, Samuel V. Angiuoli & W. Florian Fricke

The Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, MD 21201, USA

Abstract

The CloVR-ITS pipeline employs several well-known phylogenetic tools and protocols for the analysis of ribosomal ITS sequence datasets:

A) QIIME [1] – a Python-based workflow package, allowing for sequence processing and phylogenetic analysis using different methods including UCLUST [2];

B) UCHIME [3] – a tool for rapid identification of chimeric sequence fragments;

C) Mothur [4] – a C++-based software package for ecology-based analysis of phylogenetic sequences;

D) BLASTN [5] for taxonomic assignment of sequences using a custom database;

E) Metastats [6] and custom R scripts used to generate additional statistical and graphical evaluations.

CloVR-ITS accepts as input raw multiplex 454-pyrosequencer output (i.e. pooled pyrotagged sequences from multiple samples), or alternatively, pre-processed sequences from multiple samples in separate files. This protocol became first available in CloVR beta version 1.0-RC4.

Software

Step	Program	Version	URL	Ref.
Preprocessing, OTU clustering	QIIME	1.1.0	http://qiime.sourceforge.net/	[1]
Chimera detection	UCHIME	4.0.87	http://www.drive5.com/uchime/	[3]
Alpha-diversity analysis	Mothur	1.12.0	http://www.mothur.org/	[4]
Taxonomic classification of sequences	BLASTN	2.2.21	http://blast.ncbi.nlm.nih.gov	[5]
Differential taxonomic prevalence calculation	Metastats	1.0	http://metastats.cbcb.umd.edu/	[6]
Statistical and graphical evaluation	R	2.10.1-2	http://www.r-project.org/	

Reference data

Database	Data	URL	Ref.
ITS2-derived custom database	Curated full ribosomal ITS sequences (not only ITS2) with taxonomic information down to the species level	http://its2.bioapps.biozentrum.uni-wuerzburg.de/	[7]

Pipeline input

Data	Suffix	Description
Multiple sequence pool	.fasta	Pool of un-trimmed, un-checked, un-binned pyrotagged 454 sequences
Multiple fasta files	.fasta	Trimmed and binned sequences, 1 file per sample
Mapping	.txt	Sample-associated feature/metadata table (tab-delimited)
Quality files (optional)	.qual	Quality scores corresponding to single multiplex input fasta file or multiple input fastas

Pipeline output

Data	Suffix	Description
Filtered sequences	.fna	Sequences passing QIIME-based poor-quality filter (filename: **`seqs.fna`**)
Detected chimeras	.txt	Sequence names from **`seqs.fna`** (and OTU representatives) identified as putative chimeras (filename: **`allchimeraids.txt`**)

OTU assignments	.txt	List of genus-level OTUs (QIIME)
Alpha diversity	.rarefaction	Rarefaction numerical curves separated by sample (Mothur)
	.pdf	Rarefaction plots separated by metadata type (Leech/CloVR)
	.summary	Richness and diversity estimators (Mothur)
BLAST hits	.raw	BLASTN to reference datasets results table ("**-m 8**" format)
Taxonomic assignments	.tsv	Table (tab-delimited) displaying taxonomic assignment counts for each sample
Metastats output	.csv	Differentially abundant taxonomic groups (as pre-defined in Metadata input)
Skiff clustering	.pdf	Heatmap and two-way clustering based on taxonomic assignment abundances
Pie charts	.pdf	Pie charts describing assignment abundances for up to 12 samples (not performed if >12 samples are given)
Stacked histograms	.pdf	Stacked histograms displaying relative abundances for up to 50 samples and 25 features (not performed if beyond these thresholds)

A. Requirements for pipeline Input

To run the full CloVR-ITS analysis track, at least two different input files have to be provided by the user: a sequence file in fasta format and a tab-delimited mapping file. Sequence data may consist of a single fasta file that contains sequences from multiple samples[1], or multiple fasta files with trimmed and binned sequences such that each file represents a single sample. No two fasta headers within any submitted file may be identical. The mapping file provides sample-associated metadata information with the following formatting requirements, depending on the type of input sequence data:

A.1 Mapping file requirements for multiplex runs on a single sequence pool

A single fasta files can be provided describing a multiplex sequencing run. In this case the mapping file needs to be in QIIME format, e.g.

```
#SampleID       BarcodeSequence   LinkerPrimerSequence      Treatment     Description
Sample_1        AGCACGAGCCTA      CATGCTGCCTCCCGTAGGAGT     Control       mouse_ID_300
Sample_2        AGCACGAGCCTA      CATGCTGCCTCCCGTAGGAGT     Diabetic      mouse_ID_354
Sample_3        AACTCGTCGATG      CATGCTGCCTCCCGTAGGAGT     Control       mouse_ID_355
Sample_4        ACAGACCACTCA      CATGCTGCCTCCCGTAGGAGT     Diabetic      mouse_ID_356
```

The following rules apply:

1. All entries are tab-delimited.

[1] i.e. individually pyrotagged by sample-specific barcodes as commonly used in the 454 Amplicon Sequencing protocol (http://www.454.com/products-solutions/experimental-design-options/amplicon-sequencing.asp)

2. All entries in every column are defined (no empty fields).
3. The header line begins with the following fields:
   #SampleID<tab>BarcodeSequence<tab>LinkerPrimerSequence
4. The header line must end with the field Description, i.e. the total number of columns is four or more.
5. The **BarcodeSequence** and **LinkerPrimerSequence** fields have valid IUPAC DNA characters.
6. There are no duplicate header fields and no duplicate entries in the **#SampleID** column.
7. No header fields or corresponding entries contain invalid characters (only alphanumeric and underscore characters allowed).
8. There are no duplicates when the primer and barcodes are appended.

A.2 Mapping file requirements for multiple input fasta files

Multiple fasta files can be provided so that each file comprises sequences from a different sample. In this case, the mapping file must meet the following style requirements:

```
#File       SampleName   ph     Treatment   Temperature   Description
A.fasta       sampleA     high    control       mild        patientA
B.fasta       sampleB     high     sick        medium       patientB
C.fasta       sampleC     low     treated       high        patientC
```

where:

1. All entries are tab-delimited.
2. All entries in every column are defined (no empty fields).
3. The header line begins with: **#File<tab>SampleName**.
4. The **#File** column contains the names of all input fasta files and does not contain duplicate entries.
5. There are no duplicate header fields.
6. No header fields or corresponding entries contain invalid characters (only alphanumeric and underscores characters allowed).

A.3 Pairwise comparisons with Metastats

To utilize the Metastats statistical methodology, which detects differential abundances of taxa between two sample groups, the associated header field must end with "**_p**", (e.g. "**Treatment_p**", or "**ph_p**"). If a header with the "**_p**" ending exists, pairwise Metastats calculations will be carried out between all groups specified in the corresponding column (provided that a group contains at least three samples).

A.4 Providing quality scores with sequence data

To include quality scores as input, for each input fasta file <prefix>.fasta there must exist a separate quality score file <prefix>.qual. For example, if the input fasta files are **A.fasta**, **B.fasta** and **C.fasta**, then there must also exist **A.qual**, **B.qual**, and **C.qual** for quality filtering to be performed[2]. The quality score files are tagged similarly to the input fasta files before starting a pipeline.

B. Sequence preprocessing

Input data are initially assessed for quality and chimeric sequences. Problematic sequences are removed before subsequent processing.

[2] Note: fasta and quality files can be retrieved from an sff file using the Roche/454 proprietary program `sffinfo`.

B.1 File consistency check

All input fasta files are first checked for consistency with the input mapping file. If a fasta file listed within the mapping file does not exist, or if an input fasta file is not listed in the mapping file, the pipeline will halt with an error. Likewise consistency is checked for any input quality score files.

B.2 Splitting by samples and quality filtering

To check each read from the sequence pool for quality and to sort sequences based on the sample-specific barcodes, the QIIME script **`split_libraries.py`** is used with the following parameters:

`--min-seq-length 100` (sets the minimum sequence length to 100 bp)

`--max-seq-length 2000` (sets the minimum sequence length to 2000 bp)

`--barcode-type variable_length` (disables barcode corrections and allows for unique barcodes with varying lengths)

`--max-homopolymer 8` (sets the maximum homopolymer length to 8 bp)

`--min-qual-score 25` (sets the minimum average quality score to 25, applies only when quality scores are provided to the pipeline)

`--max-ambig 0` (sets the maximum number of ambiguous bases allowed to 0)

The output of this component (**`seqs.fna`**) is a single set of filtered reads identified by sample and meeting the above quality criteria.

B.3 Selection of high identity clusters

To assist in *de novo* chimera detection and downstream taxonomic analysis, sequences are clustered into high identity OTUs using a 99% identity threshold and the QIIME command **`pick_otus.py`**. We allow for reverse complement searching by UCLUST here. The longest sequences in each stringent cluster are selected as OTU representatives using **`pick_rep_set.py`**. The relative abundance of each OTU is denoted with each representative sequence for UCHIME.

B.4 Chimera identification and removal

To detect putative chimeric sequences in the filtered data, representative sequences are input to UCHIME (using *de novo* mode with default parameters). Representatives assigned as chimeras propagate the assignment across their clusters, and a single list of all putative chimeras is output. All chimeric sequences are then removed from consideration before the next step in the pipeline.

C. Sequence processing

C.1 Sequence clustering

The QIIME script **`pick_otus.py`** is used to cluster all non-chimeric reads from all samples into genus-level operational taxonomic units (OTUs) based on a nucleotide sequence identity threshold. The clustering program for this step is UCLUST [2] and the nucleotide sequence identity threshold for all reads within an OTU is 85%. UCLUST is set to examine both the forward and reverse complement sequences during clustering.

C.2 Alpha-diversity analysis.

Genus-level OTUs created by the QIIME commands above are reorganized and input to Mothur which uses the scripts **`read.otu`**, **`rarefaction.single`**, and **`summary.single`** to generate rarefaction curves and estimators of species diversity for each sample. Finally a custom program called Leech is used to plot all rarefaction curves together defined by varying color schemes related to the input mapping file.

C3. Taxonomic annotation of high identity clusters

All non-chimeric representative sequences from the 99% clusters generated in step **B.3** are searched against a custom database of ITS reference sequences from known species using BLASTN with the following options: "**`-e 1.0e-5`**" (e-value threshold), "**`-b 10`**" (number of hits to show) and "**`-m 8`**" (tabular output). Each sequence is assigned to the taxonomic lineage of the best

BLAST alignment covering at least 90% of the query sequence length and matching with a minimum identity of 90%, 85%, 75%, 70%, and 60% identity for species, genus, family, order, and class-level assignments, respectively. Representatives without alignments of sufficient coverage or identity at a specific taxonomic-level are denoted as "Unclassified." Hits are propagated across the corresponding clusters.

D. Additional analysis using Metastats and the R statistical package

The output from the taxonomic classification of each sequence from all samples by the BLAST-based classification step is further analyzed and graphically represented using the Metastats program [6] and customized scripts in the R programming language.

D.1 Detection of differentially abundant features

Metastats uses count data from annotated sequences to compare two populations in order to detect differentially abundant features [6]. BLASTN results are processed to detect different taxonomic groups at multiple levels (class, order, family, genus, species). Metastats produces a tab-delimited table displaying the mean relative abundance of a feature, variance and standard error together with a p value and q value to describe significance of the detected variations (see project website: http://metastats.cbcb.umd.edu/). Note Metastats can run analyses of 1 sample vs. 1 sample, or *N* samples vs. *M* samples, where *N* and *M* are greater than 1. It cannot perform a comparison of 1 sample vs. 2 samples.

D.2 Stacked histogram generation

Custom R scripts are used to normalize taxonomic group counts to relative abundances. Stacked histograms of the relative abundances are generated in the .pdf format, if there are at most 50 samples and at most 25 taxon groups. Beyond these limits a visualized histogram is not generated.

D.3 Unsupervised sample clustering

A custom R script called skiff is used to normalize taxon counts and to calculate distance matrices for samples and taxonomic groups, using a Euclidean distance metric. Complete-linkage (furthest neighbor) clustering is employed to create dendrograms of samples and taxa in the .pdf format. The R packages RColorBrewer and gplots are included in this task.

D.4 Pie chart visualization

Custom R scripts are used to form pie charts displaying proportions of sequences assigned to specific functional and taxonomic levels for up to 12 samples. Outputs are in .pdf format. For more than 12 samples this function is not performed, as the visual comparison for the user would be cumbersome.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0949201 and by the National Human Genome Research Institute under Grant No. 5RC2HG005597-02.

References

1. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI, et al: QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 2010, 7:335-336.
2. Edgar RC: Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 2010.

3. Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R: **UCHIME improves sensitivity and speed of chimera detection.** *Bioinformatics* 2011, **27:**2194-2200.
4. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, et al: **Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities.** *Appl Environ Microbiol* 2009, **75:**7537-7541.
5. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res* 1997, **25:**3389-3402.
6. White JR, Nagarajan N, Pop M: **Statistical methods for detecting differentially abundant features in clinical metagenomic samples.** *PLoS Comput Biol* 2009, **5:**e1000352.
7. Koetschan C, Hackl T, Muller T, Wolf M, Forster F, Schultz J: **ITS2 database IV: Interactive taxon sampling for internal transcribed spacer 2 based phylogenies.** *Molecular phylogenetics and evolution* 2012.