Title

CloVR-16S: Phylogenetic microbial community composition analysis based on 16S ribosomal RNA amplicon sequencing – standard operating procedure v.1.0

James Robert White, Cesar Arze, Kevin Galens, Malcolm Matalka, Stephen Mekosh, David R. Riley, Mahesh Vangala, Owen White, Samuel V. Angiuoli & W. Florian Fricke

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

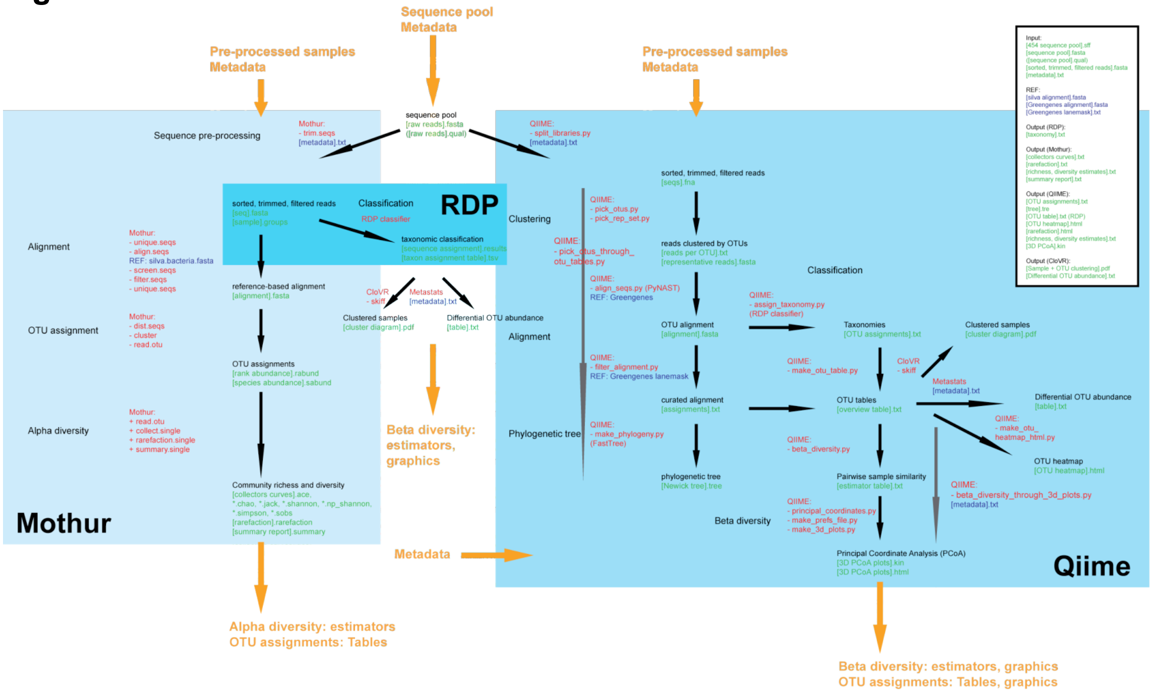
Abstract

The CloVR-16S pipeline employs several well-known phylogenetic tools and protocols for the analysis of 16S rRNA sequence datasets:

A) Mothur [1] – a C++-based software package used for clustering 16S sequences into operational taxonomic units (OTUs). Mothur creates OTUs using a matrix that describes phylogenetic distances between representative sequences and subsequently estimates within-sample diversity (α-diversity);
B) the Ribosomal Database (RDP) naive Bayesian classifier [2] assigns each 16S sequence to a taxonomy with associated empirical probabilities based on oligonucleotide frequencies;
C) Qiime [3] – a python-based workflow package, allows for sequence processing and phylogenetic analysis using different methods including phylogenetic distance (UniFrac [4]) for within- (α-diversity) and between- (β-diversity) sample analysis;
D) Additional statistical and graphical evaluation within the pipeline includes Metastats [5] and custom R scripts implemented in CloVR.

Though some of the different protocols used in CloVR-16S overlap in purpose (e.g. OTU clustering), the end-user benefits from their overall complementary nature as they focus on different aspects of the phylogenetic analysis. CloVR-16S accepts as input raw multiplex 454-pyrosequencer output, i.e. pooled sequences from multiple samples, or alternatively, pre-processed sequences from multiple samples in separate files.

Figure



Software

Step	Program	Version	Weblink	Reference
Distance-based OTU classification and phylogeny	Mothur	1.12.0	http://www.mothur.org/	[1]
Bayesian taxonomic classification	RDP classifier	2.0	http://rdp.cme.msu.edu/classifier/classifier.jsp	[2]
Phylogenetic distance-based sample comparison and phylogeny	Qiime	1.1.0	http://qiime.sourceforge.net/	[3]
Differential taxonomic prevalence calculation	Metastats	1.0	http://metastats.cbcb.umd.edu/	[5]
Statistical evaluation	R	2.10.1-2	http://www.r-project.org/	

Reference data

Database	Data	Version	Weblink	Reference
Silva	Curated 16S and 18S rRNA sequence alignment	102	http://www.arb-silva.de/	[6]
Greengenes	Curated 16S rRNA sequence alignment (core set imputed)		http://greengenes.lbl.gov/	[7]
	Lane mask		http://greengenes.lbl.gov/	[8]

Pipeline input

Data	Suffix	Description
Multiple sequence pool	.fasta	Pool of un-trimmed, un-checked, un-binned sequences
Multiple fasta files	.fasta	Trimmed and binned sequences, 1 file per sample
Metadata	.txt	Sample-associated features (tab-delimited)

Pipeline output

Data	Suffix	Description
Taxonomic assignments	.tsv	Assignments of every read (RDP classifier/Qiime)
OTU assignments	.txt	Table showing OTU sample compositions (RDP classifier/Qiime)
	.html	OTU heatmap (Qiime)
Alpha diversity	.txt	Collectors curves (Mothur)
	.txt	Rarefaction curves (Mothur)
	.txt	Richness and diversity estimates (Mothur)
	.txt	Richness and diversity estimates (Qiime)
	.txt	Summary report (Mothur)
Beta diversity	.kin	3D PCoA plots (Qiime)
	.pdf	Taxonomic prevalence-based sample clustering (CloVR)
	.pdf	Taxonomic prevalence-based stacked histograms (R)
	.csv	Differentially abundant taxonomic groups (Metastats)

A. Requirements for Pipeline Input

To run the full CloVR-16S analysis track, at least two different input files have to be provided by the user: a sequence file in the .fasta format and a tab-delimited metadata file in the .txt format. Sequence data may consist of a single .fasta file that contains sequences from multiple samples, individually tagged 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). No two FASTA headers within any submitted file may be identical. The metadata file provides sample-associated information with the following Qiime-based formatting requirements:

```
#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
```

where:

1. All entries are tab-delimited.
2. All entries in every column are defined.
3. The header line begins with the following fields:
"#SampleID<tab>BarcodeSequence<tab> LinkerPrimerSequence".
4. The header line must end with the field "Description".
5. The BarcodeSequence and LinkerPrimerSequences fields have valid IUPAC DNA characters.
6. There are no duplicate header fields.
7. No header fields or corresponding entries contain invalid characters (alphanumeric and underscore only allowed).
8. There are no duplicates when the primer and barcodes are appended.

Alternatively, multiple fasta files can be provided so that each file comprises sequences from different samples. In this case, the metadata file must meet the following requirements:

```
#File        SampleName    ph     Description
A.fasta       sampleA      high     control
B.fasta       sampleB      high      sick
C.fasta       sampleC       low     treated
```

where:

1. All entries are tab-delimited.
2. All entries in every column are defined.
3. The header line begins with: "#File<tab>".
4. There are no duplicate header fields or file names.
5. No header fields or corresponding entries contain invalid characters (alphanumeric and underscores only allowed).
6. The header line must end with the field "Description".

Pairwise comparisons: To utilize the Metastats statistical methodology for differential abundance detection, the associated header field must end with "`_p`", (e.g. "`Treatment_p`", or "`ph_p`"). Otherwise Metastats will skip pairwise analysis of the header field.

B. Sequence Processing and Analysis with Mothur

The Mothur component of CloVR-16S follows in large parts the pyrosequencing 16S rRNA sequence analysis example on the Mothur wiki page (http://www.mothur.org/wiki/Costello_stool_analysis). Sequence pools are pre-processed (trimmed, sorted, filtered), aligned against a reference (the curated Silva 16S and 18S rRNA alignment [6]), further processed to remove redundancy and to filter the alignment, used to generate a distance matrix, clustered and assigned to OTUs. Based on the sample OTU classification, the within-sample community composition is analyzed using common richness and diversity estimators as well as collectors and rarefaction curves (α-diversity).

B.1. Sequence pre-processing

To check each read from the sequence pool for quality and to sort sequences based on the sample-specific barcode, the "trim.seqs" program is used with the following parameters:

- "minlength=100" (minimum sequence length)
- "maxhomop=8" (maximum homopolymer length)
- "maxambig=0" (maximum number of ambiguous base calls)
- "flip=F" (do not use the reverse complement of the sequences -- *reverse complements are considered in the alignment step*).

All length parameters refer to base pairs (bp). This command generate a new trimmed .fasta and .groups file, which are used in the downstream analysis.

B.2 Alignment

To speed up the downstream analysis and to facilitate the analysis of large data sets, identical sequences, which can constitute a significant component of the sequence data are removed, using the "unique.seqs" command. The non-redundant sequence dataset is then aligned against the Greengenes reference core imputed alignment, which is available from the Greengenes website (http://greengenes.lbl.gov/Download/Sequence_Data/Fasta_data_files), using the "align.seqs" command with the default parameters and "flip=t" (if the alignment of a sequence read falls below the default threshold [0.50], the reverse complement is tried). In order to keep only those sequence reads that produce alignments of a minimum length of 50 bp, the "screen.seqs" command is run with the "minlength=50" option. With the "filter.seqs" command in combination with the "vertical=T" option, any column, which only contains gaps, is removed from the alignment. Since the trimming of the alignment has created new duplicate sequences, identical sequences are removed again using the "unique.seqs" command.

B.3. Threshold for the maximum number of sequences

Since the following steps of the Mothur pipeline can be computationally very demanding, the threshold for the number of unique sequence reads from the alignment of the previous step is set to 50,000. If the number of sequences exceeds the threshold, the dataset is not further processed with the Mothur component but instead analyzed through the remaining CloVR-16S components.

B.4. Clustering and OTU assignment

In Mothur, sequence reads are assigned to OTUs based on uncorrected pairwise distances between all aligned sequences. With the "dist.seqs" command a column-formatted distance matrix is generated, using the "cutoff=0.10" option, which limits the distance matrix to keep only sequence reads with a distance smaller than 0.10 (at least 90% similar). Using the default "furthest neighbor" option, the "cluster" command assigns sequence reads to OTUs based on the distance matrix generated in the previous step.

B.5. α-diversity analysis

To perform the α-diversity analysis, the OTU clustering results are read into Mothur with the "read.otu" command and the "label=unique-0.03-0.05-0.10" option to output all OTU levels using 97%, 95% and 90% similarity thresholds, respectively. The command "collect.single" generates collector's curves that describe how comprehensively a microbial community has been assessed in the sample. This is done by calculating how community richness and diversity change as more individuals from the community are sampled. "collect.single" is performed with the "freq=5" option, which sets the frequency with which the richness and diversity are calculated to every 5 sequences. To generate intra-sample rarefaction curves, applying a re-sampling without replacement approach, the "rarefaction.single" command is used with the same "freq=5" option. Rarefaction curves provide a way of comparing the richness observed in different samples. The "summary.single" command produces a summary file of various richness and diversity estimators for each sample.

C. RDP classification of all sequence reads

The output of the pre-processing step (B.1. Mothur:trim.seqs), i.e. all sorted, trimmed and filtered sequence reads, are classified using the RDP classifier tool [2], as described on the Ribosomal Database Project website (http://rdp.cme.msu.edu/classifier/classifier.jsp). As output, a results file is created, which contains the taxonomic classification of each sequence read from all samples, including a confidence score (up to 1.0) assigned by the RDP classifier. In addition, tab-delimited table files (.tsv) are generated, which show the composition of each sample at different taxonomic levels, including phylum, class, order, family and genus. Assignments with confidence values below 0.8 (80%) are assigned as “unknown” for the generation of the .tsv files.

D. Sequence processing and analysis with Qiime

The Qiime component of CloVR-16S follows the Overview Tutorial on the Qiime website (http://qiime.sourceforge.net/tutorials/tutorial.html). It uses the same unprocessed sequence pools as the Mothur component as input and takes advantage of three Qiime workflow scripts to combine related steps of the analysis pipeline: "pick_otus_through_otu_table.py" is used for sequence clustering, alignment, classification and phylogenetic tree prediction; and "beta_diversity_through_3d_plots.py" is used to calculate β-diversity estimators and to generate 3D Principal Coordinate Analysis (PCoA) plots for the graphical representation of differences in microbial community compositions between samples (β-diversity).

D.1. Sequence pre-processing

To assign multiplex reads from sequence pools to specific samples using sequence barcodes, as well as to remove low-quality reads and to filter reads by length, the "split_libraries.py" script is used. This step removes all reads from the analysis that do not have the user-specified barcode sequence. The following options are used: "--min-seq-length 100" (sets the minimum sequence length to 100 bp), "--barcode-type variable_length" (disables barcode corrections), and "--max-homopolymer 8" (sets the maximum homopolymer length to 8 bp).

D.2. Sequence clustering, alignment, classification and phylogenetic tree prediction

The "pick_otus_through_otu_table.py" workflow script calls the following Python scripts: 1) "pick_otus.py" is used to cluster reads from all samples into OTUs based on nucleotide sequence identity. The clustering program for this step is "Uclust" [9] and the nucleotide sequence identity threshold for all reads within an OTU is 97%. 2) "assign_taxonomy.py" uses the RDP classifier [2] with a confidence threshold of 0.8 to assign each OTU-representing read to a known taxon. A .txt file is created by this script, which shows the most specific classification of each read above the confidence threshold, i.e. the resolution of the classification varies between reads, showing taxonomic lineages of different lengths. 3) "make_otu_table.py" generates an OTU table from the classification results, together with the information about the number of reads that each OTU represents, which specifies the OTU counts that each sample contains for each taxonomic assignment. 4) "align_seqs.py" uses the PyNAST tool [10] to align OTU-representing reads against the Greengenes reference alignment [7]. 5) "filter_alignment.py" uses the Greengenes Lane mask [8] to defines those positions from the alignment that will be ignored when building the phylogenetic tree. 6) "make_phylogeny.py" uses the "FastTree" program [11] to generate a phylogenetic tree in the Newick format.

D.3. Beta diversity sample analysis

The "beta_diversity_through_3d_plots.py" workflow script calls the following Python scripts: 1) "beta_diversity.py" takes the OTU table and phylogenetic tree as input to calculate beta diversity estimators, including phylogenetic distance as measured through weighted and unweighted UniFrac analysis [4], and to generate a distance matrix. 2) "principal_coordinates.py" maps the multidimensional variation between samples from the distance matrix on three principal coordinates. 3) "make_prefs_file.py" sets the parameters for the PCoA display based on the user-provided metadata information. 4) "make_3d_plots.py" generates 3D PCoA plots in the .html and .kin format, which can be opened with a web browser or the free KiNG Display Software, which is available from http://kinemage.biochem.duke.edu/software/king.php.

E. Additional beta diversity analysis using Metastats and the R statistical package

The output from the taxonomic classification of each sequence read from all samples by the RDP classification step (see section C) and of the RDP-based classification of the OTU-representing sequence reads from all samples by "Qiime:assign.taxonomy" is further analyzed and graphically represented using the "Metastats" program [5] and customized scripts in the R programming language.

E.1. Detection of differentially abundant features

The "Metastats" program uses count data from the taxonomic assignment of sequences with the RDP classifier to compare two samples in order to detect differentially abundant features [5]. The results are calculated on different taxonomic levels (phylum, class, order, family, genus) and presented as a table in the .txt format, which shows 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/).

E.2. Stacked histogram generation

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

E.3. Unsupervised sample clustering

Custom R scripts are 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.

References

1. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, et al. (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol 75: 7537-7541.

2. Wang Q, Garrity GM, Tiedje JM, Cole JR (2007) Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol 73: 5261-5267.

3. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, et al. (2010) QIIME allows analysis of high-throughput community sequencing data. Nat Methods 7: 335-336.

4. Lozupone C, Knight R (2005) UniFrac: a new phylogenetic method for comparing microbial communities. Appl Environ Microbiol 71: 8228-8235.

5. White JR, Nagarajan N, Pop M (2009) Statistical methods for detecting differentially abundant features in clinical metagenomic samples. PLoS Comput Biol 5: e1000352.

6. Pruesse E, Quast C, Knittel K, Fuchs BM, Ludwig W, et al. (2007) SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. Nucleic Acids Res 35: 7188-7196.

7. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, et al. (2006) Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. Appl Environ Microbiol 72: 5069-5072.

8. Lane DJ (1991) 16S/23S rRNA sequencing. In: Stackebrandt E, Goodfellow M, editors. Nucleic Acid Techniques in Bacterial Systematics. New York: Wiley. pp. 115-175.

9. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. Bioinformatics.

10. Caporaso JG, Bittinger K, Bushman FD, DeSantis TZ, Andersen GL, et al. (2010) PyNAST: a flexible tool for aligning sequences to a template alignment. Bioinformatics 26: 266-267.

11. Price MN, Dehal PS, Arkin AP (2010) FastTree 2--approximately maximum-likelihood trees for large alignments. PLoS One 5: e9490.