All Inputs

Extra information

The info files describe the pairs of barcodes, sample names, and primer sequences. As well as the meta-data such as pH , depth, or temperature etc.

info1.json
info2.json
info3.json
info4.json

Raw reads

The raw files are provided in FASTQ format. Two per pool (MID sequence already removed by Illumina software).

1_fwd.fastq
1_rev.fastq
2_fwd.fastq
2_rev.fastq
3_fwd.fastq
3_rev.fastq
4_fwd.fastq
4_rev.fastq

16S database of choice (e.g. Silva)

Other possibilities include Green genes and RDP. Several version of Silva exists, for instance "Silva Bacerial", "Silvamod", etc.

16Sdb.fasta

id_to_taxonomy.txt

The Pipeline

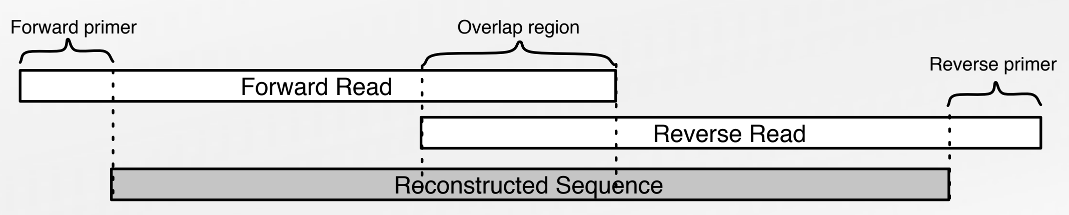
Demultiplexing

Using the barcodes, every read pair is sorted into a separate file. The threshold for barcode matching is usually set to zero base pairs mismatch. This creates several categories.

1) The reads which have a known barcode at the begining of both of the sequences in the pair. Moreover both these barcodes match two columns on the same line in the *barcodes.txt* file. These reads are labeled "Success".
2) The reads which have no barcode on either of the sequences in the pair.
3) The reads which have a known barcode on only one of the two sequences of the pair.
4) The reads which have two recognizable barcodes but they both match in the same column of the *barcodes.txt* file
5) The reads with two barcodes that match different lines in the *barcodes.txt* file.

Assembly

Using only sequences pairs from category 1 above (with matching barcodes), we attempt to assemble them. This consists in finding the matching overlap between the end of the forward read and the end of the reverse read. We use the PANDAseq algorithm for this:
http://www.biomedcentral.com/1471-2105/13/31



This provides some type of quality filtering, as sequences with badly sequenced base pairs will likely not assemble and be rejected.

Primer presence check

We search for the forward primer and reverse primer at the start and end of each read respectively. Prior to this search the reads are flipped to all run from the 3' to the 5' of the 16S RNA gene using the barcodes for orientation. Most reads have both primers in the expected positions, some reads only have one of the two primers and some are missing the primers entirely.

```
  341F    5'- CCTACGGGNGGCWGCAG   -3'
805R    5'- GACTACHVGGGTATCTAATCC   -3'
```

At this stage, only reads with good primers are kept for further processing.

Remove reads with undetermined bases

All reads with 'N' type bases are discarded.

Quality filtering

If a window of 10 base pairs falls below a score of 5, we discard the sequence.

Length cutoff

We reject those sequences that have an overlapping region longer than 100 base pairs.

Trim barcodes

We remove the barcode on both ends and proceed to further processing

Chimeras checks

Using uchime in devnovo mode and uchime in reference mode http://drive5.com/usearch/manual/uchime_ref.html

Filter unused

Remove reads that are have barcodes matching to an unused set of barcodes

Trim primers

We remove the barcode and the primer on both ends proceed to further processing

Pipeline early exit

For those who want the data at this point they can use the outputs generated that plug into Mothur or Qiime.

Qiime output

4 x fasta file (trimmed) with sequence names as "pool3_barcode47_12345"

Mothur output

4 x sequence file (trimmed)
4 x quality file
4 x groups file with names as: "pool3_barcode47"

OTU clustering

Depends on the type of question one wants to answer or the type of diversity one wants to measure. Here is the step in the pipeline where one would cluster our sequences with a given threshold as to create operational taxonomic units. See article "UPARSE" from Egdar. Or CD-HIT-OTU page on the Weizhong lab website.

Cluster size graphs

Using the clustered formed by UPARSE we plot some distributions of the cluster sizes.

Similarity search and assignment

Using the CREST classifier we search for similarities in the 16S database SILVAMOD. Assign sequences to the species level (or higher) using the best hit.

Filter table

We remove all sequences identified as 'Plastid', 'Mitochondrion', 'Thaumarchaeota', 'Crenarchaeota' or 'Euryarchaeota' and make a new OTU table with them removed.

Species bar chart

Using the OTU table and the assignments we draw a species bar chart detailing the composition at different levels. Either only at the phylum level, or straight at the tips of the taxonomy on the species level.

NMDS plot

Using the VEGAN package in R, we can make some ordination plots.

Align

The representative of each cluster against 97% clustered version of the Silva SSURef non-redundant release 111 with mother align.

BETA dispersion plot

Using the VEGAN package in R, we can check what is the variability within a pool.

Build tree

the default settings of FastTree.

Compute Unifrac distance

The tree and the OTU table are fed into the weighted unifrac implementation in PyCogent.

PERMANOVA

Using the VEGAN package in R, distances are measured using the "Horn" metric. By rarefying (downsampling) we can also use the "Bray" metric.

Other statistics

Every study will require the use of custom-build tests and employ single-use statistical scripts. These can be added to the code when they are needed.