WLS_mothur

Kim Dill-McFarland

April 20, 2017

Introduction

The Wisconsin Longitudinal Study (WLS) is a long-term study of 1957 high school graduates in Wisconsin (N=10317). The survey-based data includes physical and mental health as well as human-human and human-environment interactions from late adolescence (1957) through 2011. These include variables such as social background, aspirations, schooling, military service, labor market experiences, family characteristics and events, social participation, psychological characteristics and retirement.

> Herd, Pamela, Deborah Carr, and Carol Roan. 2014. "Cohort Profile: Wisconsin Longitudinal Study (WLS)." International Journal of Epidemiology 43:34-41 PMCID: PMC3937969

In 2015-6, a microbiota component was added to the WLS with fecal sample collection. The bacterial microbiota was determined by amplicon sequencing of the 16S rRNA gene, variable region 4 (V4). Graduates (g, N=179), their spouses (p, N=63), their siblings (s, N=134), and their sibling's spouses (e, N=32) were successfully sequenced.

This document contains mothur code for processing fastq files to yield

- a normalized OTU table
- taxonomic classification of OTUs
- alpha-diversity measures
- coverage

More information and access to the data available at http://www.ssc.wisc.edu/wlsresearch/

Sequence clean-up

Process fastqs to

- remove poor quality sequences
- remove chimeras
- remove non-bacterial sequences
- cluster very similar sequences

Create file of sample names and corresponding fastq file names

```
make.file(inputdir=/home/GLBRCORG/dillmcfarlan/WLS)
```

Combine paired end reads into contigs

```
make.contigs(file=WLS.txt, processors=10)

summary.seqs(fasta=WLS.trim.contigs.fasta, processors=10)
```

Remove poor quality sequences (ambiguous base pairs, long homopolymers, short contigs)

```
screen.seqs(fasta=WLS.trim.contigs.fasta, group=WLS.contigs.groups, maxambig=0, maxhomop=8, minlength=20
```

Define unique sequences and combine `.names` and `.groups` into a `.count_table` to reduce processing needs

```
unique.seqs(fasta=WLS.trim.contigs.good.fasta)
```

```
count.seqs(name=WLS.trim.contigs.good.names, group=WLS.contigs.good.groups)
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.fasta, count=WLS.trim.contigs.good.count_table)
```

Align data to the SILVA reference data base. Remove sequences that do not align to the correct region.

```
align.seqs(fasta=WLS.trim.contigs.good.unique.fasta, reference=/home/GLBRCORG/dillmcfarlan/mothur files/
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.align, count=WLS.trim.contigs.good.count_table)
```

```
screen.seqs(fasta=WLS.trim.contigs.good.unique.align, count=WLS.trim.contigs.good.count_table, summary=W
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.good.align, count=WLS.trim.contigs.good.good.count_tabl
```

Remove extraneous columns in alignment and re-unique

```
filter.seqs(fasta=WLS.trim.contigs.good.unique.good.align, vertical=T, trump=.)
```

```
unique.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.fasta, count=WLS.trim.contigs.good.good.coun
```

Pre-cluster sequences with 2 or fewer base pair differences to return sequencing error

```
pre.cluster(fasta=WLS.trim.contigs.good.unique.good.filter.unique.fasta, count=WLS.trim.contigs.good.un
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=WLS.trim.con
```

Define and remove chimeras

```
chimera.uchime(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=WLS.trim.c
```

```
remove.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=WLS.trim.conti
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=WLS.trim
```

Classify sequences to SILVA and remove unknown, Archaea, and Eukaryota

```
classify.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=WLS.tr
```

```
remove.lineage(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=WLS.t
```

```
summary.seqs(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta, count=WLS
```

Remove singletons

```
split.abund(fasta=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta, count=WLS
```

```
count.groups(count=WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.abund.count_tabl
```

Rename **final** files

```
system(cp WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.abund.fasta WLS.final.fas
```

```
system(cp WLS.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.abund.count_table WLS.fin
```

Define OTUs

Calculate a distance matrix and cluster using average neighbor calculation.

```
dist.seqs(fasta=WLS.final.fasta, processors=20)
```

```
cluster.split(column=WLS.final.dist, count=WLS.final.count_table, method=average, processors=15)
```

Define OTUs at 97% similarity.

```
make.shared(list=WLS.final.an.unique_list.list, count=WLS.final.count_table, label=0.03)
```

Coverage

Calculate Good's coverage.

```
summary.single(shared=WLS.final.an.unique_list.shared, label=0.03, calc=nseqs-sobs-coverage)
```

Normalize

Normalize OTU table to 10,000 sequences per sample. This will remove 18 low coverage samples.

```
normalize.shared(shared=WLS.final.an.unique_list.shared, method=totalgroup, norm=10000)
```

Alpha-diversity

```
summary.single(shared=WLS.final.an.unique_list.0.03.norm.shared, label=0.03, calc=nseqs-sobs-coverage-b
```

Classification

Classify OTUs to the GreenGenes reference database

```
classify.seqs(fasta=WLS.final.fasta, count=WLS.final.count_table, template=/home/GLBRCORG/dillmcfarlan/
```

```
classify.otu(list=WLS.final.an.unique_list.list, taxonomy=WLS.final.gg.wang.taxonomy, count=WLS.final.c
```

Representative sequences

Pick representative sequences for OTUs

```
get.oturep(column=WLS.final.dist, list=WLS.final.an.unique_list.list, fasta=WLS.final.fasta, count=WLS.
```

Rename sequences by OTU number using python script `clean_repFasta_FAST.py`. This outputs `clean_repFasta.fasta`

Calculate distance matrix of renamed representative sequences. This distance matrix is used for tree generation for a phyloseq object in `WLS_data_manipulation`.

```
dist.seqs(fasta=clean_repFasta.fasta, output=lt, processors=15)
```