AbxD01 Analysis

Alyx Schubert

September 29, 2014

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

# Introduction

This is a digital notebook to accompany the paper titled, “*” that will be published in* . It was written in R markdown and converted to html using the R knitr package. This enables us to embed the results of our analyses directly into the text to allow for a completely reproducible data analysis pipeline. A github repository is available where you can pull down your own version of the notebook to modify our analysis or adapt it to your analysis.

This document was generated using mothur v.1.33 and R.

# Prcessing 16S gene sequence data in mothur

## First, get the required files.

Sequence files can be found here \_.

The following files can be found through the mothur wiki: silva.bacteria.fasta trainset9\_032012.pds.tax trainset9\_032012.pds.fasta

The following files can be found on the github repository for this notebook: abxD01.files

## Now for the curation pipeline.

These sequences were generated using the Illumina MiSeq Platform. For data processing, we followed the MiSeq SOP outlined in the [Kozich et al. manuscript](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3753973/). These commands come from the outline described in the [MiSeq SOP](http://www.mothur.org/wiki/MiSeq_SOP).

make.contigs(file=abxD01.files, processors=14)  
summary.seqs(fasta=abxD01.trim.contigs.fasta, processors=14)  
screen.seqs(fasta=abxD01.trim.contigs.fasta, group=abxD01.contigs.groups, summary=abxD01.trim.contigs.summary, maxambig=0, maxlength=275, processors=14)  
unique.seqs(fasta=abxD01.trim.contigs.good.fasta)  
count.seqs(name=abxD01.trim.contigs.good.names, group=abxD01.contigs.good.groups)  
summary.seqs(count=abxD01.trim.contigs.good.count\_table)  
#Make sure silva.bacteria.fasta file in folder  
pcr.seqs(fasta=silva.bacteria.fasta, start=11894, end=25319, keepdots=F, processors=14)  
system(mv silva.bacteria.pcr.fasta silva.v4.fasta)  
summary.seqs(fasta=silva.v4.fasta)  
align.seqs(fasta=abxD01.trim.contigs.good.unique.fasta, reference=silva.v4.fasta, processors=14)  
summary.seqs(fasta=abxD01.trim.contigs.good.unique.align, count=abxD01.trim.contigs.good.count\_table)  
screen.seqs(fasta=abxD01.trim.contigs.good.unique.align, count=abxD01.trim.contigs.good.count\_table, summary=abxD01.trim.contigs.good.unique.summary, start=1968, end=11550, maxhomop=8)  
summary.seqs(fasta=current, count=current)  
filter.seqs(fasta=abxD01.trim.contigs.good.unique.good.align, vertical=T, trump=.)  
unique.seqs(fasta=abxD01.trim.contigs.good.unique.good.filter.fasta, count=abxD01.trim.contigs.good.good.count\_table)  
pre.cluster(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.fasta, count=abxD01.trim.contigs.good.unique.good.filter.count\_table, diffs=2)  
chimera.uchime(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.count\_table, dereplicate=t, processors=14)  
remove.seqs(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.fasta, accnos=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.accnos)  
summary.seqs(fasta=current, count=current)  
#Make sure the reference and training set files are in the analysis folder  
classify.seqs(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count\_table, reference=trainset9\_032012.pds.fasta, taxonomy=trainset9\_032012.pds.tax, cutoff=80)  
remove.lineage(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count\_table, taxonomy=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.taxonomy, taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)  
remove.groups(count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.count\_table, fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta, taxonomy=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.taxonomy, groups=mock1-mock2-mock3-mock4-mock5-mock6-mock7-mock8)  
  
#renamed fasta, count\_table, taxonomy to have shorter names  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy abxD01.final.taxonomy)  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count\_table abxD01.final.count\_table)  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta abxD01.final.fasta)  
  
set.current(fasta=abxD01.final.fasta, count=abxD01.final.count\_table, taxonomy=abxD01.final.taxonomy)

## Creating phylotypes

Continuing the pipeline outlined on the mothur wiki's MiSeq SOP.

#PHYLOTYPE analysis:  
#Note: label=1 is the genus level, which is the finest you can do, then label=2 is family and so forth  
phylotype(taxonomy=abxD01.final.taxonomy)  
make.shared(list=abxD01.final.tx.list, count=abxD01.final.count\_table)  
count.groups()  
#The following size 1625 was chosen based on the results of count.groups(). I tried to minimize the number of samples at Day 0 that were dropped at that minimum cutoff.  
sub.sample(shared=abxD01.final.tx.shared, size=1625)  
classify.otu(list=abxD01.final.tx.list, count=abxD01.final.count\_table, taxonomy=abxD01.final.taxonomy)  
#tx.2=family, tx.3=order, etc  
get.relabund(shared=abxD01.final.tx.2.subsample.shared)   
get.relabund(shared=abxD01.final.tx.3.subsample.shared)  
get.relabund(shared=abxD01.final.tx.4.subsample.shared)   
get.relabund(shared=abxD01.final.tx.5.subsample.shared)   
get.oturep(column=abxD01.final.dist, list=abxD01.final.tx.list, label=2-3-4-5, fasta=abxD01.final.fasta, count=abxD01.final.count\_table)

## Creating OTUs

It is worth noting that while phylotype analysis followed the listed protocol exactly, we deviated from the listed SOP for the OTU analysis. This was due to the extensive size of this dataset and limitations in our computing abilities. Thus, in a **separate** folder we copied over the following files from earlier in the protocol: abxD01.trim.contigs.good.unique.good.filter.unique.fasta abxD01.trim.contigs.good.unique.good.filter.count\_table trainset9\_032012.pds.fasta trainset9\_032012.pds.tax

**If this isn't done in a new folder, you will write over all your previous files because they have the same name.**

#Changed diffs from 2 to 3  
pre.cluster(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.fasta, count=abxD01.trim.contigs.good.unique.good.filter.count\_table, diffs=3)  
chimera.uchime(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.count\_table, dereplicate=t, processors=14)  
remove.seqs(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.fasta, accnos=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.accnos)  
summary.seqs(fasta=current, count=current)  
#Make sure the reference and training set files are in the analysis folder  
classify.seqs(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count\_table, reference=trainset9\_032012.pds.fasta, taxonomy=trainset9\_032012.pds.tax, cutoff=80)  
remove.lineage(fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count\_table, taxonomy=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.taxonomy, taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)  
remove.groups(count=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.count\_table, fasta=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta, taxonomy=abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.taxonomy, groups=mock1-mock2-mock3-mock4-mock5-mock6-mock7-mock8)  
  
#renamed fasta, count\_table, taxonomy to have shorter names  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy abxD01.final.taxonomy)  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count\_table abxD01.final.count\_table)  
system(cp abxD01.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta abxD01.final.fasta)  
  
set.current(fasta=abxD01.final.fasta, count=abxD01.final.count\_table, taxonomy=abxD01.final.taxonomy)  
  
#OTU ANALYSIS:  
#changed SOP taxlevel from 4 to 5  
cluster.split(fasta=abxD01.final.fasta, count=abxD01.final.count\_table, taxonomy=abxD01.final.taxonomy, splitmethod=classify, taxlevel=5, cutoff=0.20)  
make.shared(list=abxD01.final.an.unique\_list.list, count=abxD01.final.count\_table, label=0.03)  
count.groups()  
  
#Note: the size varies based on what you see on count.groups, need to check the distribution of day 0 groups and clindamycin group  
sub.sample(shared=abxD01.final.an.unique\_list.shared, size=1625)  
classify.otu(list=abxD01.final.an.unique\_list.list, count=abxD01.final.count\_table, taxonomy=abxD01.final.taxonomy, label=0.03, cutoff=80)  
get.relabund(shared=abxD01.final.an.unique\_list.0.03.subsample.shared)  
get.oturep(column=abxD01.final.dist, list=abxD01.final.an.unique\_list.list, label=0.03, fasta=abxD01.final.fasta, count=abxD01.final.count\_table)  
collect.single(shared=abxD01.final.an.unique\_list.0.03.subsample.shared, calc=chao-invsimpson-sobs-simpsoneven-shannon-jclass, freq=100)  
rarefaction.single(freq=100)  
#Note summary.single--use the original shared file, and put subsample as same as before, will subsample 1000 times and get avgs  
summary.single(calc=nseqs-coverage-chao-sobs-invsimpson-simpsoneven-shannoneven-shannon, shared=abxD01.final.an.unique\_list.shared, subsample=1625)

# Data Analysis