Docherty Lab Illumina MiSeq Guide-Short Verision
Modified from MiSeq SOP: http://www.mothur.org/wiki/MiSeq_SOP

MAKING CONTIGS OUT OF SEQUENCE FILES (most time consuming step)

```
make.contigs(file=stability.files, processors=8)
```

```
summary.seqs(fasta=stability.trim.contigs.fasta)
```

GETS RID OF SEQUENCES THAT ARE MYSTERIOUSLY LONG

```
screen.seqs(fasta=stability.trim.contigs.fasta, group=stability.contigs.groups, maxambig=0,
maxlength=275)
```

```
summary.seqs()
```

MAKES US ONLY HAVE TO USE UNIQUE SEQUENCES INSTEAD OF ALL SEQUENCES, GENERATES A TABLE OF HOW MANY SEQUENCES FALL INTO EACH UNIQUE SEQUENCE CATEGORY

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

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

```
summary.seqs(count=stability.trim.contigs.good.count_table)
```

ALIGNS SEQUENCES TO A REFERENCE ALIGNMENT (SILVA) AT SPECIFIC POSITIONS

```
pcr.seqs(fasta=silva.bacteria.fasta, start=11894, end=25319, keepdots=F, processors=8)
```

```
system(rename silva.bacteria.pcr.fasta silva.v4.fasta)
```

```
summary.seqs(fasta=silva.v4.fasta)
```

```
align.seqs(fasta=stability.trim.contigs.good.unique.fasta, reference=silva.v4.fasta)
```

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

GETS RID OF SEQUEENCES THAT DO NOT ALIGN OR HAVE HOMOPOLYMERS THAT ARE 8 BASE PAIRS OR LONGER; CREATES A NEW TABLE OF UNIQUE SEQUENCES

```
screen.seqs(fasta=stability.trim.contigs.good.unique.align,
count=stability.trim.contigs.good.count_table, summary=stability.trim.contigs.good.unique.summary,
start=1968, end=11550, maxhomop=8)
```

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

```
unique.seqs(fasta=stability.trim.contigs.good.unique.good.filter.fasta,
count=stability.trim.contigs.good.good.count_table)
```

MERGES SEQUENCES THAT HAVE ONLY 2 BASE PAIR DIFFERENCES FOR EVERY 100 BASE PAIRS

```
pre.cluster(fasta=stability.trim.contigs.good.unique.good.filter.unique.fasta,
count=stability.trim.contigs.good.unique.good.filter.count_table, diffs=2)
```

**set # of differences to error rate of taq to correct for taq mistakes.

IDENTIFIES CHIMERIC SEQUENCES AND REMOVES THEM

```
chimera.uchime(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.count_table, dereplicate=t)
```

```
remove.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
accnos=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.accnos)
```

```
summary.seqs(fasta=current, count=current)
```

CLASSIFIES SEQUENCES INTO BACTERIA, ARCHAEA, EUKARYA, UNKNOWN, CHLOROPLAST, MITOCHONDRIA AND REMOVES EVERYTHING EXCEPT BACTERIA

**because our primers look at the v4 region of the 16s rRNA subunit, we only want sequences that look at bacteria and this will get rid of all the other junk

```
classify.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta,
count=stability.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=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.taxonomy,
taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)
```

ASSESSING ERROR RATES (THIS DOESN’T REALLY WORK BECAUSE WE DIDN’T RUN A MOCK COMMUNITY WITH OUR CURRENT DATA – THIS WILL HAPPEN WITH FUTURE SAMPLES THAT ARE SUBMITTED)

```
get.groups(count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.count_table, fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta,
groups=Mock)
```

```
seq.error(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta,
reference=HMP_MOCK.v35.fasta, aligned=F)
```

```
dist.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta,
cutoff=0.20)
```

```
cluster(column=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.dist,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table)
```

```
make.shared(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table, label=0.03)
```

```
rarefaction.single(shared=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.shared)
```

REMOVING THE MOCK COMMUNITY FROM ANALYSIS

```
remove.groups(count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.count_table, fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.taxonomy, groups=Mock)
```

PREPPING SEQUENCES FOR ANALYSIS

```
dist.seqs(fasta= stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta,
cutoff=0.20)
```

```
cluster(column=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.dist,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table)
```

OR

```
cluster.split(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy, splitmethod=classify, taxlevel=4, cutoff=0.15)
```

DETERMINE HOW MANY SEQUENCES THERE ARE IN EACH OTU

```
make.shared(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table, label=0.03)
```

CREATES A FILE THAT TELLS YOU THE TAXONOMY OF EACH OTU

```
classify.otu(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy, label=0.03)
```

OPEN THIS FILE:
stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.0.03.cons.taxonomy

PHYLOTYPE (YOU COULD ALSO DO PHYLOGENETIC) – THIS IS GOING TO SEPARATE OUT YOUR SEQUENCES BY EACH SAMPLE

```
phylotype(taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy)
```

```
make.shared(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.tx.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table, label=1)
```

```
classify.otu(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.tx.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.pick.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.pick.taxonomy, label=1)
```

OPEN THIS FILE:
stability.an.shared

RENAMING FILES

```
system(rename
stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.shared
stability.an.shared)
```

```
system(rename
stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.an.unique_list.0.03.cons.taxonomy stability.an.cons.taxonomy)
```

RAREFYING DATA SO THAT SAMPLES ARE COMPARABLE

```
count.groups(shared=stability.an.shared)
```

```
sub.sample(shared=stability.an.shared, size=2441)
```

**you change the size to the lowest size number so that all sequences are equal and can be accurately be compared

CALCULATING ALPHA DIVERSITY

```
collect.single(shared=stability.an.shared, calc=chao-invsimpson, freq=100)
```

```
rarefaction.single(shared=stability.an.shared, calc=sobs, freq=100)
```

```
summary.single(shared=stability.an.shared, calc=nseqs-coverage-sobs-invsimpson, subsample=2441)
```

OPEN THIS FILE:
stability.an.groups.ave-std.summary