mothur > make.contigs(file=REULAKES.files, processors=8)

mothur > summary.seqs(fasta=REULAKES.trim.contigs.fasta)

mothur > screen.seqs(fasta=REULAKES.trim.contigs.fasta, group=REULAKES.contigs.groups, summary=REULAKES.trim.contigs.summary, maxambig=0, maxlength=254, minlength=251, maxhomop=8)

mothur > unique.seqs(fasta=REULAKES.trim.contigs.good.fasta)

mothur > summary.seqs(fasta=REULAKES.trim.contigs.good.unique.fasta, name=REULAKES.trim.contigs.good.names)

mothur > count.seqs(name=REULAKES.trim.contigs.good.names)

mothur > count.groups(group=REULAKES.contigs.good.groups)

mothur > align.seqs(fasta=REULAKES.trim.contigs.good.unique.fasta, reference=silva.v4.fasta)

mothur > remove.seqs(fasta=REULAKES.trim.contigs.good.unique.align, name=REULAKES.trim.contigs.good.names, group=REULAKES.contigs.good.groups, accnos=REULAKES.trim.contigs.good.unique.flip.accnos)

mothur > count.seqs(name=REULAKES.trim.contigs.good.pick.names, group=REULAKES.contigs.good.pick.groups)

mothur > summary.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.align, count=REULAKES.trim.contigs.good.pick.count\_table)

mothur > screen.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.align, count=REULAKES.trim.contigs.good.pick.count\_table, summary=REULAKES.trim.contigs.good.unique.pick.summary, start=1968, end=11550)

mothur > summary.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.align, count=REULAKES.trim.contigs.good.pick.good.count\_table)

mothur > filter.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.align, vertical=T)

mothur > unique.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.fasta, count=REULAKES.trim.contigs.good.pick.good.count\_table)

mothur > pre.cluster(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.count\_table, diffs=2)

chimera.uchime(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.count\_table, dereplicate=t, processors=8)

remove.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.fasta, accnos=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.accnos)

summary.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.pick.count\_table)

classify.seqs(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.pick.count\_table, reference=trainset9\_032012.pds.fasta, taxonomy=trainset9\_032012.pds.tax, cutoff=80)

mothur > remove.lineage(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.pick.count\_table, taxonomy=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pds.wang.taxonomy, taxon=Chloroplast-Mitochondria-unknown-Eukaryota)

mothur > set.current(fasta=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pick.fasta, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.pick.pick.count\_table, taxonomy=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pds.wang.pick.taxonomy, processors=16)

cluster.split(fasta=current, count=current, taxonomy=current, splitmethod=classify, taxlevel=4, cutoff=0.15, processors=16)

ALPHA DIVERSITY:

mothur > make.shared(list=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pick.an.unique\_list.list, count=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.denovo.uchime.pick.pick.count\_table, label=0.03)

mothur > count.groups(shared=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pick.an.unique\_list.shared)

mothur > summary.single(shared=REULAKES.trim.contigs.good.unique.pick.good.filter.unique.precluster.pick.pick.an.unique\_list.0.03.subsample.shared, calc=chao-npshannon, subsample=T)