**Mothur pipeline log**

Before mothur:

1. Created excel sheet “analysis”
   * Makes excel sheet of fastq files to perform make.file

Mothur:

1. mothur > make.file(inputdir=., type=fastq, prefix=analysis)
   * Imported my excel file “analysis” which describes my fastq files and their respective titles (treatment types)
2. mothur > make.contigs(file=analysis.files, processors=8)
   * made contigs out of my excel file “analysis” describing my fastq files
3. mothur > summary.seqs(fasta=analysis.trim.contigs.fasta)
   * Looked at the summary of my trimmed contig files
   * Average sequence length was ~304 bp?
     1. 355 bp (average read on MiSeq run) – 50 bp (trimmed on mothur)
4. mothur > screen.seqs(fasta=analysis.trim.contigs.fasta, group=analysis.contigs.groups, maxambig=0, maxlength=325)
   * Eliminated any ambiguities as well as any sequences > 325 bp (Less than 2.5% of all sequences)
5. mothur > unique.seqs(fasta=analysis.trim.contigs.good.fasta)
   * Got only unique sequences, merge duplicates
6. mothur > count.seqs(name=analysis.trim.contigs.good.names, group=analysis.contigs.good.groups)
   * This will generate a file called analysis.trim.contigs.good.count\_table. In subsequent commands we'll use it by using the count option
7. mothur > summary.seqs(count=analysis.trim.contigs.good.count\_table)
   * Summary count of the unique sequence table we just made
8. Before pcr.seqs:
   * Used <https://mothur.org/blog/2016/Customization-for-your-region/> to find E. coli reference for my primer set
   * My primers:
     1. 338F: ACTCCTACGGGAGGCAGCAG
     2. 518R: ATTACCGCGGCTGCTGG
     3. Reverse complement of 518R: TAATGGCGCCGACGACC
     4. Reverse of iii: CCAGCAGCCGCGGTAAT
     5. FINAL *E. COLI* REFERENCE SEQUENCE: ACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGCAAGCCTGATGCAGCCATGCCGCGTGTATGAAGAAGGCCTTCGGGTTGTAAAGTACTTTCAGCGGGGAGGAAGGGAGTAAAGTTAATACCTTTGCTCATTGACGTTACCCGCAGAAGAAGCACCGGCTAACTCCGTGCCAGCAGCCGCGGTAAT
   * Created e.coli.v4.txt, a .txt file of my reference sequence
   * Downloaded silva.nr\_v138.tgx; will use as the SILVA seed reference file
   * Unzipped silva.nr\_v138.tgx using tar -xzf
     1. Created silva.nr\_v138.align and silva.nr\_v138.tax
   * mothur > align.seqs(fasta=e.coli.v4.txt, reference=silva.nr\_v138.align)
   * mothur > summary.seqs(fasta=e.coli.v4.align)
9. mothur > pcr.seqs(fasta=silva.nr\_v138.align, start=6334, end=13862, keepdots=FALSE)
   * Coordinates used to trim silva.nr: Start=6334, end=13862
10. mothur > rename.file(input=silva.nr\_v138.pcr.align, new=silva.v4.fasta)
    * Renamed the output file of pcr.seqs to silva.v4.fasta
11. mothur > align.seqs(fasta=analysis.trim.contigs.good.unique.fasta, reference=silva.v4.fasta)
    * Aligned my trimmed sequences to the trimmed reference database
12. mothur > summary.seqs(fasta=analysis.trim.contigs.good.unique.align, count=analysis.trim.contigs.good.count\_table)
    * View summary of align.seqs
    * A screen shot of a computer

      Description automatically generated
13. mothur > screen.seqs(fasta=analysis.trim.contigs.good.unique.align, count=analysis.trim.contigs.good.count\_table, summary=analysis.trim.contigs.good.unique.summary, start=52, end=7528, maxhomop=8)
    * Trimmed the sequences from start position 52 to end position 7528
14. mothur > filter.seqs(fasta=analysis.trim.contigs.good.unique.good.align, vertical=T, trump=.)
    * Pull out columns in alignment with overhangs and gap characters (“-“)
    * A picture containing black, phone, screen

      Description automatically generated
15. mothur > unique.seqs(fasta=analysis.trim.contigs.good.unique.good.filter.fasta, count=analysis.trim.contigs.good.good.count\_table)
    * Get unique sequences after the filter we just performed
16. mothur > pre.cluster(fasta=analysis.trim.contigs.good.unique.good.filter.unique.fasta, count=analysis.trim.contigs.good.unique.good.filter.count\_table, diffs=2)
    * This command will split the sequences by group and then sort them by abundance and go from most abundant to least and identify sequences that are within 2 nt of each other. If they are then they get merged
17. mothur > chimera.vsearch(fasta=analysis.trim.contigs.good.unique.good.filter.unique.precluster.fasta, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.count\_table, dereplicate=t)
    * Search for chimeras to later remove
18. mothur > remove.seqs(fasta=analysis.trim.contigs.good.unique.good.filter.unique.precluster.fasta, accnos=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.accnos)
    * Removes chimeras (7041 sequences total)
19. mothur > classify.seqs(fasta=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.count\_table, reference=silva.nr\_v138.align, taxonomy= silva.nr\_v138.tax, cutoff=80)
    * Classify sequences by kingdom so we can later remove undesirable lineages
20. mothur > remove.lineage(fasta=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.count\_table, taxonomy=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.nr\_v138.wang.taxonomy, taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)
    * Removes non-bacterial sequences
21. Skipped assessing error rates because no mock dataset was used
22. mothur > dist.seqs(fasta=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta, cutoff=0.03)
23. mothur > cluster(column=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.dist, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.pick.count\_table)
    * Cluster sequences into OTUs
24. mothur > make.shared(list=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.opti\_mcc.list, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.pick.count\_table, label=0.03)
    * Know how many sequences are in each OTU group
25. mothur > classify.otu(list=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.opti\_mcc.list, count=analysis.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.pick.count\_table, taxonomy= analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.nr\_v138.wang.taxonomy, label=0.03)
    * Know the taxonomy for each OTU
    * OTU table complete!

**Analysis**

1. mothur > rename.file(taxonomy=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.opti\_mcc.0.03.cons.taxonomy, shared=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.opti\_mcc.shared)
   * Renamed my shared and taxonomy file to make my downstream analysis easier to type
2. mothur > count.groups(shared=analysis.opti\_mcc.shared)
   * see how many sequences we have in each sample
   * A screenshot of a computer

     Description automatically generated
3. mothur > sub.sample(shared=analysis.opti\_mcc.shared, size=24390)
   * Jwax Kan 3 is our smallest dataset at 24390 sequences. We will make a subsample of all our data using this parameter

**Lefse:**

1. Created straindesign.txt in notepad
   * Created straindesign using interactiondesign and editing the last column
   * Removed dashes in straindesign in notepad
2. mothur > lefse(shared=analysis.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.opti\_mcc.shared, design=straindesign.txt)
   * Ran lefse with analysis.opti\_mcc.0.03.subsample.shared