Mothur pipeline

Mothur software is an independent, open-source tool for describing and comparing microbial communities

Mac version
Using ReadLine,Boost,HDF5,GSL
mothur v.1.44.3
Last updated: 10/19/2020
by
Patrick D. Schloss
Department of Microbiology & Immunology
University of Michigan
http://www.mothur.org
When using, please cite:
Schloss, P.D., et al., Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol, 2009. 75(23):7537-41.
Distributed under the GNU General Public License
Type 'help()' for information on the commands that are available
For questions and analysis support, please visit our forum at https://forum.mothur.org
Type 'quit()' to exit program
[NOTE]: Setting random seed to 19760620.
Interactive Mode

Set the input directory for Mothur.

mothur >
mothur > set.dir(input=/Users/Desktop/mothur)
Mothur's directories:
inputDir=/Users /Desktop/mothur/

The below command is used in Mothur to create a file with a specific type and prefix.

mothur > make.file(inputdir=., type=gz, prefix=stability)
Setting input directory to: /Users/Desktop/mothur/

Contigs are contiguous sequences of DNA that are generated during the process of genome assembly, which is the process of combining DNA sequences from various sources to reconstruct a genome. During assembly, the reads are aligned to each other and overlapping regions are used to construct contigs. The below command is used to create contigs file from a list of files.

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

 # Below command generates a summary of the sequences in a fasta file, including information such as the number of sequences, the total number of bases, the average length of the sequences, and the standard deviation of the sequence lengths. It also provides the number of sequences per length and the total number of sequences with ambiguities.

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

 # The below command is used in Mothur to filter sequences based on certain criteria. [In this case, we are discarding any sequences that contain more than 0 ambiguous bases (indicated by the maxambig=0 argument) and we are discarding any sequences that are longer than 500 bases (indicated by the maxlength=500 argument)

mothur > screen.seqs(fasta=stability.trim.contigs.fasta, group=stability.contis.groups, maxambig=0, maxlength=500)

 # It displays the current settings for various parameters that are used by Mothur commands

mothur > get.current()

 # The below command is used in Mothur to remove redundant sequences from a fasta file.

mothur > unique.seqs(fasta=stability.trim.contigs.fasta)

 # The below command is used in Mothur to count the number of sequences in a fasta file and group them by a specific criterion and get an idea of the distribution of sequences among different groups.

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

 # Below command generates a summary of the sequences in a fasta file, including information such as the number of sequences, the total number of bases, the average length of the sequences, and the standard deviation of the sequence lengths. It also provides the number of sequences per length and the total number of sequences with ambiguities.

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

 # The below command is used in Mothur to perform PCR on a fasta file using a set of primer sequences. Additionally, the pcr.seqs command can be used to isolate specific sequences from a larger dataset, and to improve the accuracy of downstream analysis by removing sequences that are unlikely to align well or that are unlikely to be of interest.

mothur > pcr.seqs(fasta=ecoli.16srrna.fasta, oligos= oligos_primers.txt)

 # Aligning the sequence in the fasta file to the reference sequence

mothur > align.seqs(fasta=stability.trim.contigs.good.unique.fasta, reference=silva.nr_v138.fasta)

 # Screen the sequence and remove sequence that doesn't meet the given criteria. The criteria are specified by the start, end and maxhomop (maximum number of consecutive homopolymers) parameters in the function

mothur > 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)

 # The below code filters the sequence in the fasta file. The criteria are specified by the vertical and trump parameters in the functions. The vertical parameter filters the sequences based on the vertical representation of the sequence alignment. It allows to filter out columns that have too much gap or too much homopolymers and keep only columns that have a minimal number of gaps and a minimal number of homopolymers.
The trump parameter removes any character other than the nucleotides (A, C,G,T).

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

 # The below command is used in Mothur to remove redundant sequences from a fasta file.

mothur > unique.seqs(fasta=stability.trim.contigs.good.unique.good.filter.fast, count=stability.trim.contigs.good.good.count_table)

 # The pre.cluster function uses a two-stage clustering approach. It first clusters sequences that are identical, then sequences that differ by one base, then sequences that differ by two bases and so on. The "diffs" parameter specifies the maximum number of differences that sequences can have in order to be considered in the same cluster.

This step is done to reduce the number of sequences and increase the computational efficiency of the later steps. The remaining sequences are considered to be more distinct, and are used for further analysis such as creating distance matrix, OTU clustering, etc.

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

 # The below function identifies and removes chimeric sequences from the fasta file. Chimeras are sequences that are artificially created by the polymerase during PCR amplification and sequencing process. They are formed by the fusion of sequences from different sources. These sequences can lead to an overestimation of diversity and can bias the results of downstream analysis.

mothur > chimera.vsearch(fasta=stability.trim.contigs.good.unique.good.filter.nique.precluster.fasta, count=stability.trim.contigs.good.unique.good.filter.unique.precluster.count_table, dereplicate=t)

 # This function removes sequences specified in the file

mothur > remove.seqs(fasta=stability.trim.contigs.good.unique.good.filter.uniqe.precluster.fasta, accnos=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.accnos)

 # The classify.seqs function assigns a taxonomic classification to each sequence in the fasta file based on the reference alignment and taxonomy files.

mothur > classify.seqs(fasta=/Users /Desktop/mothur/stability.trim.contgs.good.unique.good.filter.unique.precluster.pick.fasta , count=/Users /Desktop/mothur/stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.count_table , reference=/Users /Desktop/mothur/silva.nr_v138/silva.nr_v138.align , taxonomy=/Users /Desktop/mothur/silva.nr_v138/silva.nr_v138.tax , cutoff=80)

 # Remove the specified lineages "Chloroplast-Mitochondria-unknown-Eukaryota" from the fasta file

mothur > remove.lineage(fasta=/Users/Desktop/mothur/stability.trim.conigs.good.unique.good.filter.unique.precluster.pick.fasta , count=/Users/genecis/Desktop/mothur/stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.count_table , taxonomy=/Users /Desktop/mothur/stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.nr_v138.wang.taxonomy , taxon=Chloroplast-Mitochondria-unknown-Eukaryota)

 # This function clusters the sequences in the fasta file into subclusters based on their taxonomy. The splitmethod parameter is set to "classify", which means that the function will use the taxonomic assignment of each sequence as the basis for clustering.
The taxlevel parameter is set to 4, which means that the function will cluster the sequences based on their taxonomy at the fourth level.
The cutoff parameter is set to 0.03, which means that the function will consider a subcluster to be valid if it contains at least 0.03 (3%) of the total sequences.

mothur > cluster.split(fasta=/Users/Desktop/mothur/stability.trim.contgs.good.unique.good.filter.unique.precluster.pick.pick.fasta , count=/Users /Desktop/mothur/stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.pick.count_table , taxonomy=/Users /Desktop/mothur/stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.nr_v138.wang.pick.taxonomy , splitmethod=classify, taxlevel=4, cutoff=0.03)