Research and Innovation
Nevada Bioinformatics Center

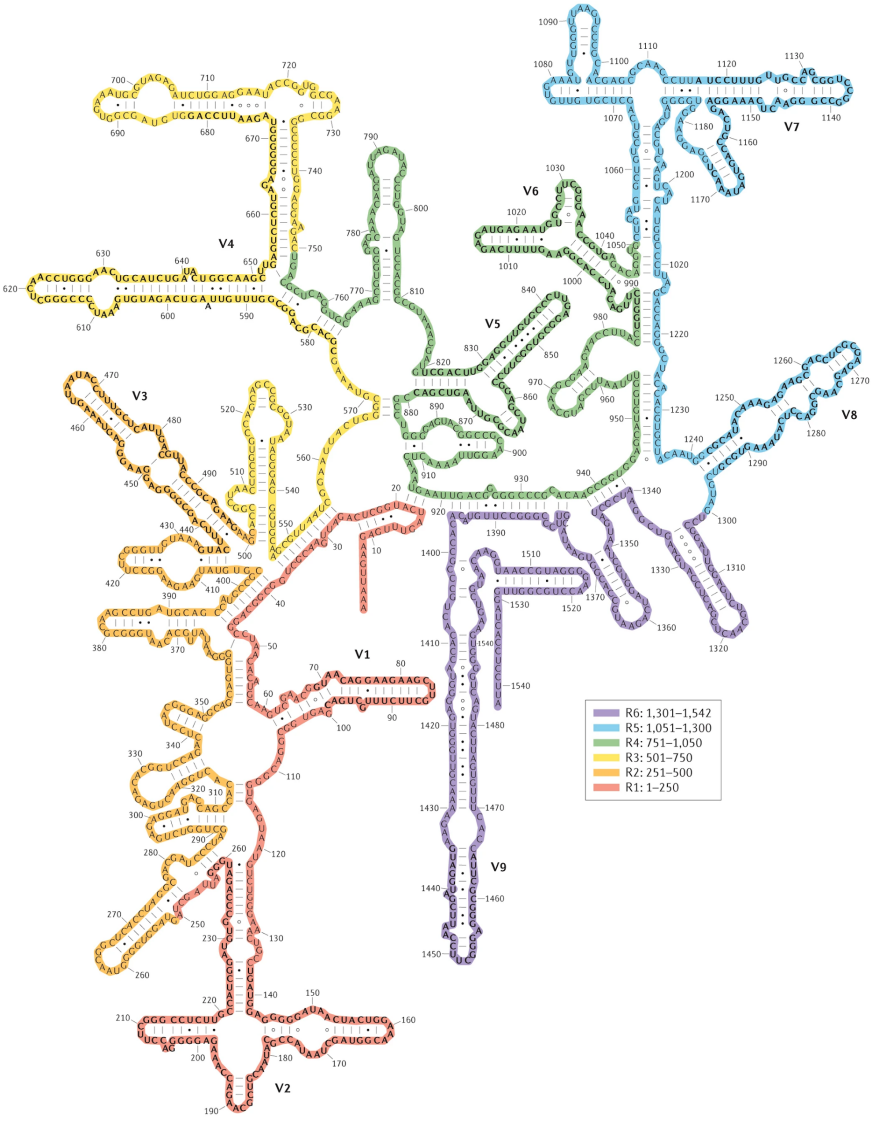
Tutorial: How antibiotics kill healthy gut bacteria? Using mothur on Galaxy

In this tutorial we will perform an analysis based on the Standard Operating Procedure (SOP) for MiSeq data, developed by the Schloss lab, the creators of the mothur software package. Released in 2009, Mothur is a bioinformatics tool designed for assigning taxonomic classifications to 16S rRNA sequences.

Background

The 16S rRNA gene has several properties that make it ideally suited for our purposes

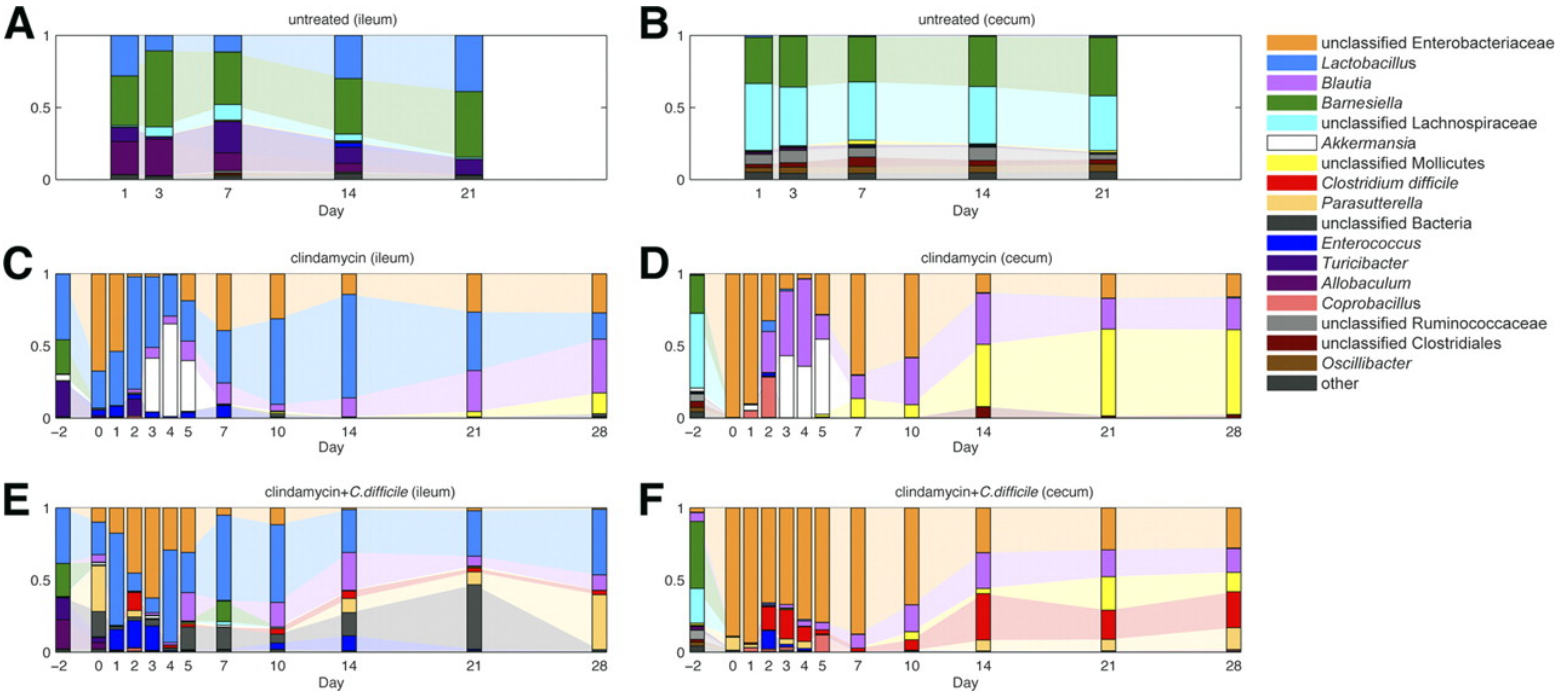
- Present in all prokaryotes
- Highly conserved + highly variable regions
- Huge reference databases



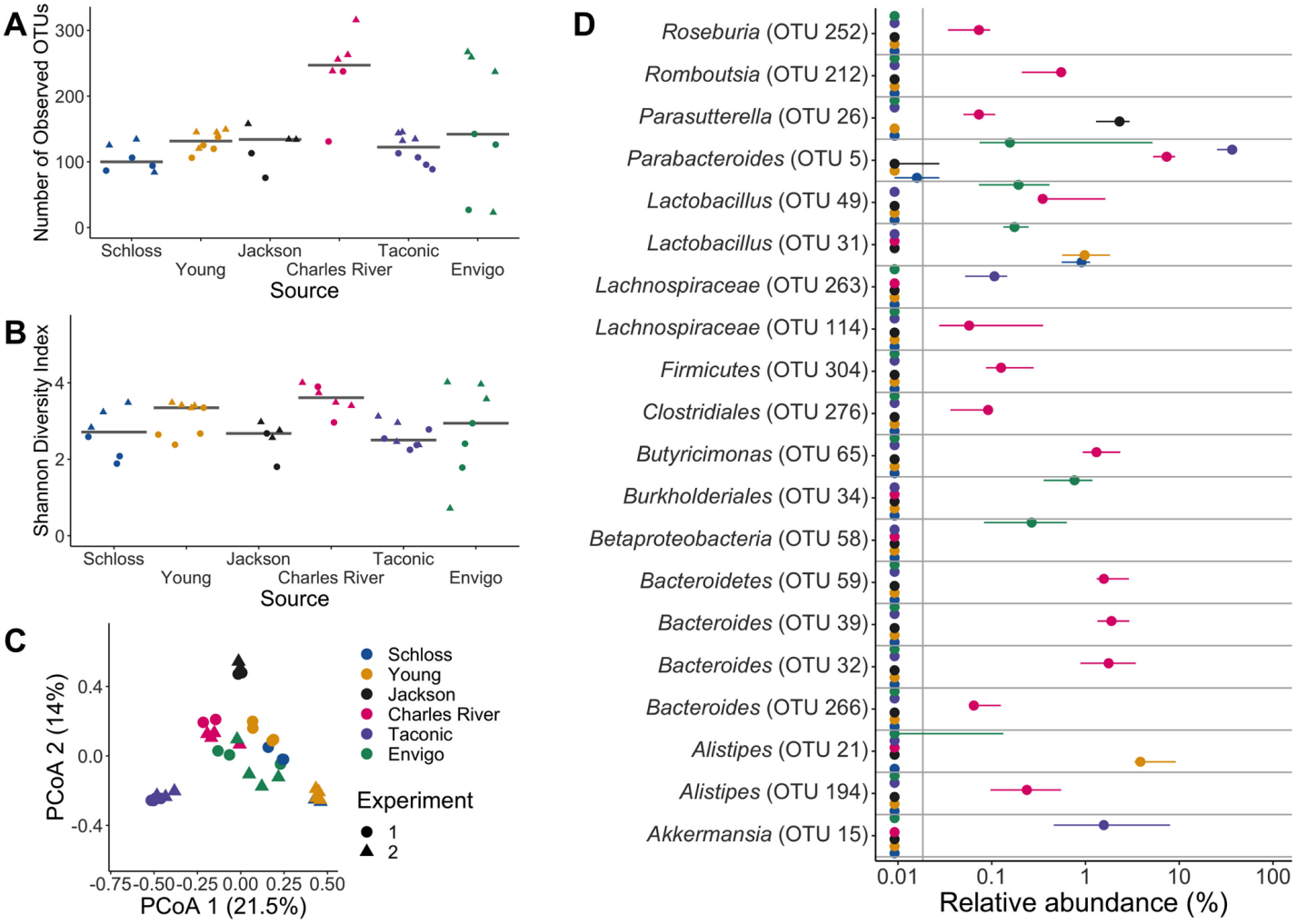
Understanding our Data

In this tutorial we will be using data from the paper: ' The Initial Gut Microbiota and Response to Antibiotic Perturbation Influence Clostridioides difficile Clearance in Mice'. Tomkovich *et al*. 2020. *mSphere*.

- Mice from 6 different sources (two labs at UM, 4 commercial vendors) were treated with clindamycin, followed by a *C. difficile* challenge, and then *C. difficile* colonization levels were measured throughout the infection.



- Clindamycin effect on microbiome: see above figure from Buffie *et al.* 2011. *Infection and Immunity.*
- All mice were C57BL/6 models: *should* be biologically identical, and should have the same response to the antibiotic treatment and pathogen infection.
- Findings: while all mice were colonized with *C. difficile* 1 day post-infection, variation emerged from days 3 to 7 post-infection with animals from some sources colonized with *C. difficile* for longer and at higher levels.
- What could be causing this? Likely differences in the baseline gut microbiome of these mice!
- In the figure below, we see that at the baseline, the microbiomes of these mice are altered, in fact there were over 260 significantly different OTUs. The top 20 of which are plotted in D.



- Findings of the paper are significant because different studies using the C57BL/6 mouse are assuming that the mouse colony they have behaves the same biologically as all other mice of that breed. But we see studies using mice from different sources are not so easily compared.
- **For today:** we will be using only 14 'Schloss' samples taken from days -1, 0, 2, and 5 to see how the gut microbiome (in only these mice) responds to antibiotics treatment and pathogen infection over the course of about a 1 week.

Importing the data into Galaxy

Now that we know what our input data is, let's import it into our Galaxy history: All data required for this tutorial is available from Zenodo: https://zenodo.org/record/6015413

1. Navigate to the **UPLOAD DATA** button on the upper left portion of the galaxy window, below Tools.
2. There are several possibilities to upload the data depending on how many datasets you have and what their origin is. Data can be imported
    a. from your local file system
    b. from a given URL - which we will be using
    c. from a shared data library on the Galaxy server you are working on
3. We want to use URLs to access all of the workshop data from Zenodo. So click on P**ASTE/FETCH DATA** at the bottom of the window and copy and paste this list into the blank space:
4. Once imported, press **START**. This might take several minutes.

https://zenodo.org/record/6015413/files/S11_D0_E2_S138_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_D0_E2_S138_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_D2_E2_S33_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_D2_E2_S33_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_D5_E2_S177_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_D5_E2_S177_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_Dn1_E2_S109_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S11_Dn1_E2_S109_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_D0_E2_S143_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_D0_E2_S143_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_D5_E2_S179_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_D5_E2_S179_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_Dn1_E2_S98_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S12_Dn1_E2_S98_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D0_E2_S137_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D0_E2_S137_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D2_E2_S28_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D2_E2_S28_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D5_E2_S194_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_D5_E2_S194_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_Dn1_E2_S106_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S21_Dn1_E2_S106_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_D0_E2_S136_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_D0_E2_S136_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_D5_E2_S176_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_D5_E2_S176_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_Dn1_E2_S97_L001_R1_001.fastq.gz
https://zenodo.org/record/6015413/files/S22_Dn1_E2_S97_L001_R2_001.fastq.gz
https://zenodo.org/record/6015413/files/silva.bacteria.fasta
https://zenodo.org/record/6015413/files/trainset16_022016.pds.tax
https://zenodo.org/record/6015413/files/trainset16_022016.pds.fasta

Creating Paired List for our Dataset

GREAT! Now our data should be imported into our Galaxy "History" and we can proceed with our analyses:

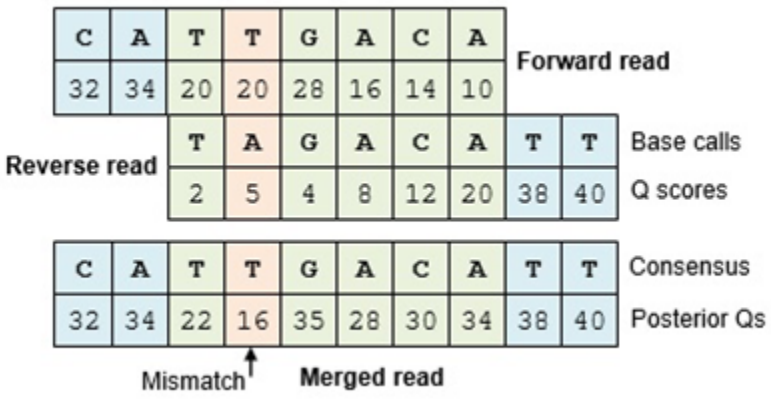
1. Click on the checkmark icon at the top of your history.
2. Select all the FASTQ files (28 in total)
    a. **Tip**: type fastq in the search bar at the top of your history to filter only the FASTQ files; you can now use the All button at the top instead of having to individually select all 40 input files
    b. Click on for all selected
    c. Select Build List of Dataset Pairs from the dropdown menu
3. In the next dialog window you can create the list of pairs. By default Galaxy will look for pairs of files that differ only by a `_1` and `_2` part in their names. In our case however, **these need to be changed**.
4. Change these values accordingly
    a. Change `_1` to `_L001_R1_001.fastq` in the text field on the top left
    b. Change `_2` to `_L001_R2_001.fastq` in the text field on the top right
5. You should now see a list of pairs suggested by Galaxy.
6. These files should be ready to be paired. I like to go down the list and pair each one manually, checking the name as I go. When all 14 samples are paired, we can give our list a name.

Quality Control

For more information on the topic of quality control, please see the Galaxy training materials here.

Before starting any analysis, it is always a good idea to assess the quality of your input data and improve it where possible by trimming and filtering reads. The mothur toolsuite contains several tools to assist with this task. We will begin by merging our reads into contigs, followed by filtering and trimming of reads based on quality score and several other metrics.

Create contigs from paired-end reads: In this experiment, paired-end sequencing of the ~253 bp V4 region of the 16S rRNA gene was performed. The sequencing was done from either end of each fragment (paired-end). Because the reads coming off the Illumina MiSeq are about 250 bp in length, this results in a significant overlap between the forward and reverse reads in each pair. We will combine these pairs of reads into contigs.



The **Make.contigs** tool creates the contigs, and uses the paired collection as input. Make.contigs will look at each pair, take the reverse complement reverse read, and then determine the overlap between the two sequences. Where an overlapping base call

differs between the two reads, the quality score is used to determine the consensus base call. A new quality score is derived by combining the two original quality scores in both of the reads for all the overlapping positions.

- In the tools section, search for “Make.contigs”
- Click on the Tool, with the following parameters:
- Under **Select a way to provide forward and reverse fastq files ?,** select ‘Multiple pairs - Combo mode’
- Give it the file list that we just created
- All other parameters for overlapping can be left to default.
- Multiple pairs - Combo mode
- At bottom of the tool, click “Execute”. This will take several minutes to run.

Data Cleaning

Next, we want to improve the quality of our data. To this end we will run a workflow that performs the following steps:

1. **Filter by length**: We know that the V4 region of the 16S gene is around 250 bp long. Anything significantly longer was likely a poorly assembled contig. We will remove any contigs longer than 275 base pairs using the *Screen.seqs* tool
2. **Remove low quality contigs**: We will also remove any contigs containing too many ambiguous base calls.
3. **Deduplicate sequences**: Since we are sequencing many of the same organisms, there will likely be many identical contigs. To speed up downstream analysis we will determine the set of unique contigs using *Unique.seqs* tool

First we need to copy the url of the workflow so that we can import it into our **Workflows** tab in Galaxy: https://usegalaxy.eu/u/lucas-bishop/w/workflow-1-quality-control-galaxy-training-16s-microbial-analysis-with-mothur-imported-from-uploaded-file

1. Navigate to “Import”
2. Enter the URL.
3. Once the workflow is imported into Galaxy, press the Play button on the right hand side of the page.

This workflow relies on two files that were produced when we used the *Make.contigs* tool: A “Group” file and a “Contigs” file. These files tell the workflow the overlapped sequences and the sample name to go with them.

1. Double check that the Contigs option is set to “trim.contigs.fasta” and the groups file is the most up to date one. The other fields will auto populate.
2. Run the workflow.
3. Take a look at the output of the *Count.seqs* command. it summarizes the number of times each unique sequence was observed across each of the samples.

Sequence Alignment

Now we are ready to align our quality filtered data to a reference. Aligning to a reference database is important because it keeps the direction of the sequence intact and positional homology. Alignments with secondary structure information improve OTU assignment.

We are going to use two mothur commands, *Align.seqs* and *Summary.seqs* to align our sequences to a database (silva.bacteria.fasta) and then produce a summary of that alignment, including positions where our data starts and stops.

1. Search for *Align.seqs* in the tools menu.
    a. Check the fasta file with our sequences is the output of *Unique.seqs*
    b. Select reference template from **Your History**
    c. The reference template is "silva.bacteria.fasta"
    d. Leave everything else Default and click "Execute" this will take a couple minutes to run.
2. Search for *Summary.seqs* in the tools menu
    a. Make sure you use the .align file that was just created in your history
    b. Click "Execute" and take a look at the .summary file that was just created.
    c. Take a look at the file. It should have information about our alignment. Most importantly, we see it starts at position 13862 and ends at position 23444.

Great! Now we have an alignment, and know some of the quality information about it. To ensure that all our reads overlap our region of interest, we will:

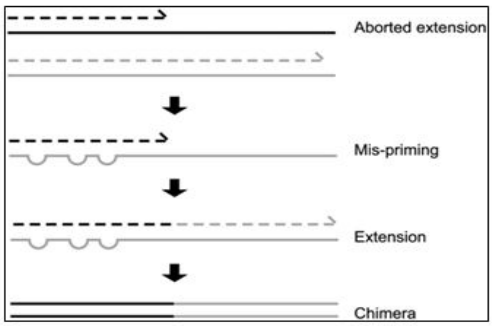
1. Remove any reads not overlapping the region V4 region using *Screen.seqs* tool
2. Remove any overhang on either end of the V4 region to ensure our sequences overlap *only* the V4 region, using *Filter.seqs* tool
3. Clean our alignment file by removing any columns that have a gap character (`-`, or `.` for terminal gaps) at that position in every sequence (also using *Filter.seqs* tool)
4. Group near-identical sequences together with *Pre.cluster* tool
5. Sequences that only differ by one or two bases at this point are likely to represent sequencing errors rather than true biological variation, so we will cluster such sequences together.
6. Remove Sequencing artifacts known as *chimeras* (discussed in next section) with the *Chimera.vsearch* tool. This uses the VSEARCH algorithm to detect chimeric sequences.

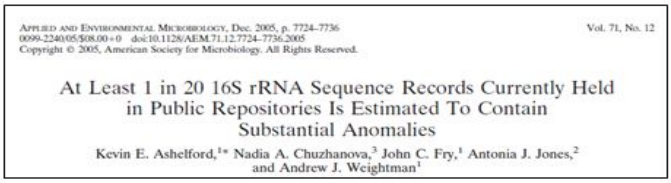
To do so, import the next workflow from:
https://usegalaxy.eu/u/lucas-bishop/w/workflow-2-data-cleaning-and-chimera-removal-galaxy-training-16s-microbial-analysis-with-mothur-imported-from-uploaded-file

After the workflow is imported, check that the Aligned sequences and the Count table are the most recent ones, and click "Execute". Think about it:

1. How many chimeric sequences were detected? **HINT: 'vsearch.accnos' in History**
2. How many sequences remain after this cleaning workflow? **HINT: search for 'summary.seqs' in History**

What is a Chimeric Sequence?




APPLIED AND ENVIRONMENTAL MICROBIOLOGY, Dec. 2005, p. 7724–7736 Vol. 71, No. 12
0099-2240/05/$08.00+0 doi:10.1128/AEM.71.12.7724–7736.2005

At Least 1 in 20 16S rRNA Sequence Records Currently Held in Public Repositories Is Estimated To Contain Substantial Anomalies

Kevin E. Ashelford,[1]* Nadia A. Chuzhanova,[3] John C. Fry,[1] Antonia J. Jones,[2] and Andrew J. Weightman[1]

- In Greek mythology:
  - A creature that was an amalgam of multiple animals
  - Body of a lion, head of a goat, tail resembling a snake
- In your sequence data:
  - The combination of multiple sequences during PCR to create a hybrid
- In sequence databases:
  - A not-so-small nightmare of junk data
  - Mis-annotation
  - Enhanced "discovery" of novel organisms

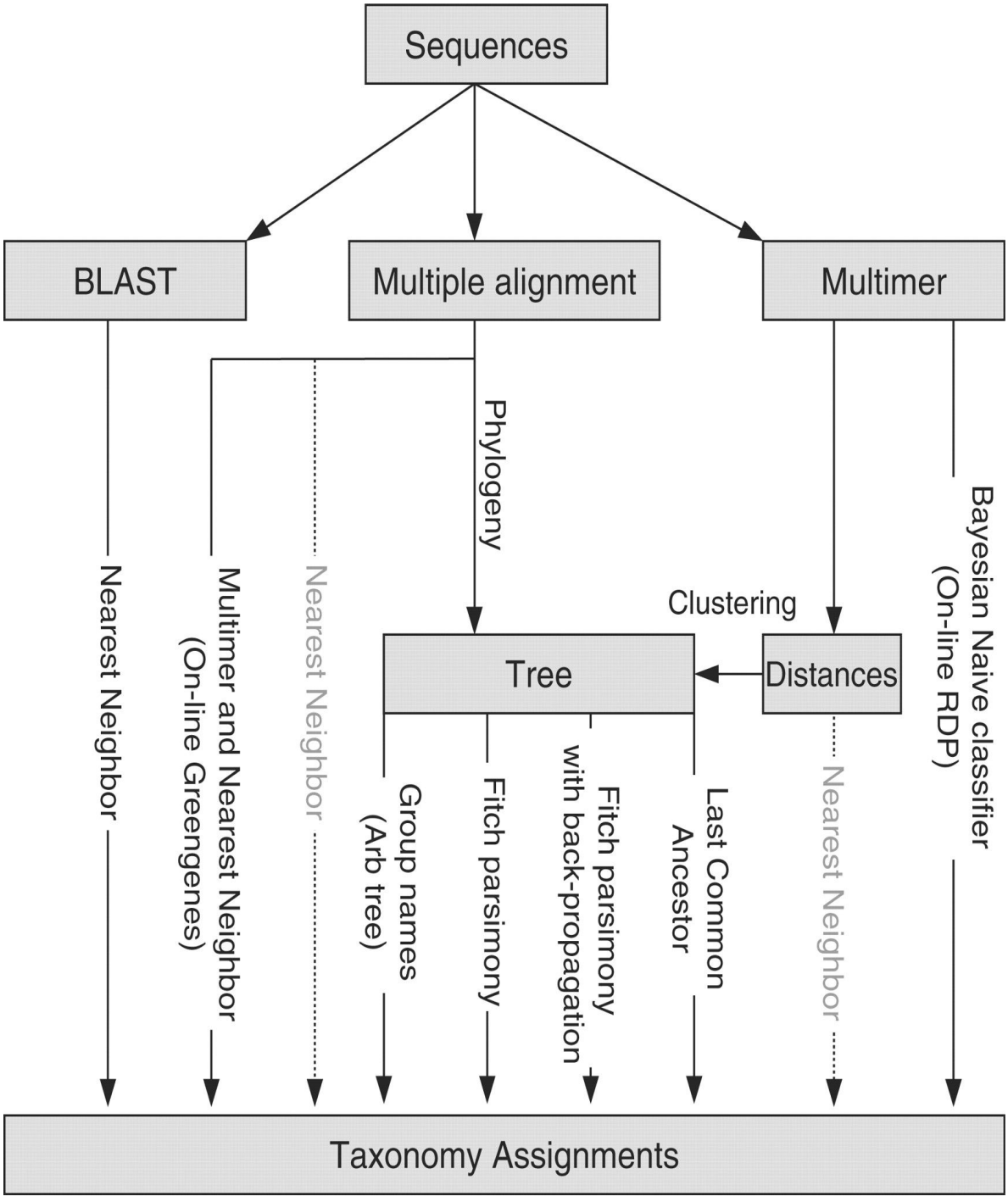
Chimera generation figure from: Haas et al. 2011, Genome Research 21:494-504

Taxonomic Classification

Now that we have thoroughly cleaned our data, we are finally ready to assign a taxonomy to our sequences. We will do this using a Bayesian classifier (via the *Classify.seqs* tool) and a mothur-formatted training set provided by the Schloss lab based on the RDP (Ribosomal Database Project, Cole *et al.* 2013) reference taxonomy.

1. Import workflow for classification from the following link and press the start button https://usegalaxy.eu/u/lucas-bishop/w/workflow-3-classification-galaxy-training-16s-microbial-analysis-with-mothur-imported-from-uploaded-file
2. Make sure that the first two inputs are the cleaned files (*.pick.*)
3. Check that the training datasets, fasta and tax files, are the ones we imported from Zenodo (they both begin with 'trainset')
4. Run workflow. This will take a little while to classify all sequences.
5. Take a look at the new outputs. How are the count tables different?

After classifying all of the sequences, this workflow then removes using the *Remove.seqs* tool all sequences classified as Eukarya, Mitochondrial, Archaea, Chloroplast, or unknown.


Sequences
BLAST
Multiple alignment
Multimer
Phylogeny
Clustering
Tree
Distances
Nearest Neighbor
Multimer and Nearest Neighbor
(On-line Greengenes)
Nearest Neighbor
Group names
(Arb tree)
Fitch parsimony
Fitch parsimony
with back-propagation
Last Common
Ancestor
Nearest Neighbor
Bayesian Naive classifier
(On-line RDP)
Taxonomy Assignments

OTU Clustering

In 16S sequencing approaches, OTUs are clusters of similar sequence variants of the 16S rRNA marker gene sequence. Each of these clusters is intended to represent a taxonomic unit of a bacterial species or genus depending on the sequence similarity threshold. Typically, OTU clusters are defined by a 97% identity threshold of the 16S gene sequence variants at species level. 98% or 99% identity is suggested for strain separation.

To do OTU clustering, we will be importing a workflow that is going to run the following mothur commands to calculate pairwise distances, cluster them by distance, generate OTUs based on these distances/cutoff thresholds, and then make a 'shared' file which has counts and taxonomy: https://usegalaxy.eu/u/lucas-bishop/w/workflow-4-mock-otu-clustering-galaxy-training-16s-microbial-analysis-with-mothur-imported-from-uploaded-file

After running this workflow, examine the 'taxonomy' output of the *Classify.otu* step. It should look something like this:

OTU Size Taxonomy

Otu001 63576
Bacteria(100);Proteobacteria(100);Gammaproteobacteria(100);Enterobacteriales(100);Enterobacteriaceae(100);Enterobacteriaceae_unclassified(100);

Otu002 40325
Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);

Otu003 32406
Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Porphyromonadaceae(100);Porphyromonadaceae_unclassified(100);

Otu004 20670
Bacteria(100);Verrucomicrobia(100);Verrucomicrobiae(100);Verrucomicrobiales(100);Verrucomicrobiaceae(100);Akkermansia(100);

Diversity Analysis

Species diversity consists of three components: species richness, taxonomic or phylogenetic diversity and species evenness.

- Species richness = the number of different species in a community.
- Species evenness = how even in numbers each species in a community is.
- Phylogenetic diversity = how closely related the species in a community are.

Now we have all of the output files to begin our diversity analysis. One thing to do which is common in microbiome studies (but not required) is to subsample our samples to an even depth so that we can compare between different treatment groups.

We can view the number of reads in each group by viewing the summary of the *Count.groups* command. Inspecting that file we see the following:

S11_D0_E2_S138
S11_D2_E2_S33
S11_D5_E2_S177
S11_Dn1_E2_S109
S12_D0_E2_S143
S12_D5_E2_S179
S12_Dn1_E2_S98
S21_D0_E2_S137
S21_D2_E2_S28
S21_D5_E2_S194
S21_Dn1_E2_S106
S22_D0_E2_S136
S22_D5_E2_S176
S22_Dn1_E2_S97

This shows us the number of reads in each group. Typically we subsample to the group with the lowest count, but one of the samples was problematic: S21_Dn1_E2_S106. So for subsampling we would want to use the next lowest group count: 5332. We will visualize this in the next and final step.

To make sure that all samples have been sampled to a sufficient depth, we want to use a rarefaction command, *Rarefaction.single* to rarefy the sequences. Do rarefaction with the final workflow: https://usegalaxy.eu/u/lucas-bishop/w/workflow-6-alpha-diversity-galaxy-training-16s-microbial-analysis-with-mothur-imported-from-uploaded-file