Analysis of 16S rRNA data from VAMPS (https://vamps.mbl.edu/)
Brazelton Lab, May 2016
https://baas-becking.biology.utah.edu/

This workflow was used for the following publications:
Brazelton et al. (submitted) Metagenomic identification of active methanogens and methanotrophs in serpentinite springs of the Voltri Massif, Italy.

Data processing primarily based on Schloss et al. (2013) (http://www.mothur.org/wiki/MiSeq_SOP)
Other references:
Phyloseq (https://joey711.github.io/phyloseq/)
USEARCH (http://www.drive5.com/usearch/)

All Brazelton lab scripts are provide at our github site: https://github.com/Brazelton-Lab

Note that these commands are written exactly as implemented on the Brazelton lab server. Other users would have to adapt for their own system. For example, we use the cluster workload manager SLURM (http://slurm.schedmd.com/), which is why many commands begin with 'srun'.

To begin, make a new directory for this project and link the appropriate files to it:

```
mkdir <project>
ln -svi /path/to/VAMPS_files/*  /path/to/new_project_directory/
cd <project>
```

VAMPS files typically contain only one representative of each sequence, and the abundance of that sequence in that sample is reported at the end of the FASTA header of each sequence. We have a Python script that will create a new FASTA file that copies each FASTA entry to represent the abundance in the sample. In other words, a sequence with a "7" at the end of the FASTA header will be copied 6 times so that there are 7 examples in the new FASTA file. Run this on each of your FASTA files and check the output:

```
srun fasta-expander-vamps-2013.py <each_file_name>.fa
```

Make a group file (tab-delimited file containing all of the sequence ids in a project with their associated sample). Use the new FASTA files you made above:

```
srun group_from_filenames.py --separator . --position 1 *.expanded.fa >
<project>.group
```

Combine all samples from the same project:

```
cat *.expanded.fa > <project>.expanded.fa
```

Now in mothur:

```
srun --ntasks 8 --pty mothur
```

```
count.groups(group=<project>.group)
unique.seqs(fasta=<project>.expanded.fa)
count.seqs(name=current, group=current)
summary.seqs(count=current, processors=<CPUs>)
```

[copy and paste results from summary into your notebook. check that the numbers reflect the expanded abundances, not the smaller one-representative numbers in the original VAMPS file]

Check what the most recent version of the reference database is

```
align.seqs(fasta=current,
reference=/srv/databases/silva/silva.SSURef_123.1/SILVA_123.1_SSURef_Nr99_tax
_silva_full_align_trunc.bact.fasta)
summary.seqs(fasta=current, count=current)
```

[copy and paste results from summary into your notebook]

```
filter.seqs(fasta=current, vertical=T)
summary.seqs(fasta=current, count=current)
```

[copy and paste results from summary into your notebook]

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

[copy and paste results from summary into your notebook]

```
classify.seqs(fasta=current, count=current,
reference=/srv/databases/silva/silva.SSURef_123.1/SILVA_123.1_SSURef_Nr99_tax
_silva_full_align_trunc.bact.fasta,
taxonomy=/srv/databases/silva/silva.SSURef_123.1/SILVA_123.1_SSURef_Nr99_tax_
silva_full_align_trunc.bact.taxonomy, cutoff=80)
```

Now you can leave mothur:

```
quit()
```

compress or remove the alignment file:

```
gzip --best *.align
```

or (preferred)

```
rm *.align
```

Remove other unnecessary files:

```
rm *pick.count_table
rm *good.count_table
rm *filter.count_table
rm *filter.fasta
rm *unique.fasta
rm *.map
```

Output 1: The final count_table - the one with the longest name and ending with "precluster.count_table" - is one of the final products. Copy this to a different directory where you will collect all of the final output files.
For example:

```
mkdir final_output
cp *.unique.count_table final_output/
```

Convert the count table into a shared file:

```
srun convert_count_to_shared <count_table_file>
```

Start mothur again to calculate some diversity statistics:

```
srun --ntasks <CPUs> --pty mothur

rarefaction.single(shared=[shared file])
summary.single(shared=current)
summary.shared(shared=current,
calc=sharedsobs-sorclass-sorabund-morisitahorn)
tree.shared(shared=current, calc=morisitahorn)
tree.shared(shared=current, calc=sorclass)
heatmap.sim(shared=current, calc=morisitahorn-sorclass)

quit()
```

Delete the .rabund files:

```
rm *.rabund
```

Output 2: Copy the SIX resulting files into the final output directory. These files end with '.groups.rarefaction', '.groups.summary', '.count_table.summary', '.tre', and '.sim.svg

Add full phylogeny to taxon name and optionally split summary by taxonomic levels:

```
srun modify_tax_summary --tax_level 3,4,5,6 <filename>.tax.summary>
```

Output 3: Copy these modified taxonomy summary files into the final output directory.

Generate a krona graph from the taxonomy summary file:

```
srun plot_tax_summary.py --split <filename>.tax.summary <output_prefix>
mkdir krona
mv *.krona krona/
```

```
srun ktImportText -o <prefix>.krona.html krona/*.krona
```

Output 4: Copy this krona.html file into the final output directory.

The End!

Description of Output Files

1. The file ending in **'.count_table'** is a tab-delimited text file that you could open with a text editor or with a spreadsheet program like Excel. It lists the number of times each sequence occurs in each sample.
2. The **'.groups.rarefaction'** file is a tab-delimited text file that you could open with a text editor or with a spreadsheet program like Excel. It contains the data you would need to make a rarefaction plot, which shows the number of different kinds of sequences in each sample as a function of the total number of sequences in that sample. The **'.groups.summary'** file is a tab-delimited text file that you could open with a text editor or with a spreadsheet program like Excel. It contains various alpha diversity statistics for each sample. Consult http://mothur.org/wiki/Calculators for more information. The **'.count_table.summary'** file is a tab-delimited text file that you could open with a text editor or with a spreadsheet program like Excel. It contains the number of shared sequences and the percent **dissimilarity** between each pair of samples according to three different beta diversity calculators (see above). The **'.tre'** files are dendrograms that can be opened in any phylogenetic tree software such as Dendroscope. They depict the whole-community similarities among the samples in the study according to binary presence/absence of shared sequences (sorclass) or according to community structure including differential abundance of shared sequences (morisitahorn). The **'sim.svg'** file is an image file that can be opened with Firefox or (preferably) with an SVG viewer like GIMP. It shows the same information as the **'.tre'** files, except that similarities are shown as gradations of color rather than as a hierarchical dendrogram. The sample labels are impossible to read. Sorry, I don't know how to solve this yet. You could make your own heat map with the information in the **'.count_table.summary'** file.
3. The **'.tax.summary.*'** files are tab-delimited text files that you could open with a text editor or with a spreadsheet program like Excel. They report the taxonomic classifications of each sequence and the abundance of that taxonomic classification in each sample. The '.tax.summary.mod' file includes all taxonomic ranks together, and the other files contain only one taxonomic rank.
4. The **'.krona.html'** file can be opened with any web browser. It provides a graphical visualization of the taxonomical classifications in the '.tax.summary.*' files described above. The charts are interactive and can be explored by double-clicking on a section of the chart. Only one sample is shown at a time, and other samples can be chosen by clicking on the sample name in the menu on the left-hand side.