Tutorial for making a custom kraken2 database with phylogenetic cluster sequence variants (PSVs)

Author: Enrique Doster (enriquedoster@tamu.edu)

This tutorial outlines how to take reference genomes for *Mannheimia haemolytica*, make a phylogenetic tree based on sequence similarity using "mashTree", run "TreeCluster.py" to cluster genomes based on various methods, relabel genomes to match the new PSV labels in the kraken2 format, and finally build the database.

In practice, this database will be used after running non-host reads through a "screening" database such as the "plusPF" which contains references from a wide variety of organisms, and filtering out reads classified in the lineage of Pasteurellaceae (only children). In theory, running your samples against a "screening" database will reduce false-positive classifications, which can then be run through this "confirmation" database which narrows kraken2's options for classifications.

Table of contents

Step 1: Acquiring reference genomes

Determine which genomes to include in your database. We'll download the Pasteurellaceae genomes (not including Mh) and the Mh separetely to handle them differently.

I use the "ncbi-genome-download" tool.

For Pasteurellaceae (4495 genomes), I used the ncbi website to identify all Pasteurellaceae genomes that aren't MH (n=4495). I downloaded that into a txt file and ran this command.

```
ncbi-genome-download bacteria -s genbank -A past_not_MH_genome_accessions.txt -H -F fasta -o genbank_Pasteurellaceae_genomes_noMH --metadata-table metadata_genbank_Pasteur_noMH.txt

WARNING: Skipping entry, as it has no ftp directory listed: 'GCA_031583005.1'
```

For Mannheimia haemolytica (taxaid 75985), genbank contained 1801 genomes.

```
ncbi-genome-download bacteria -s genbank -T 75985 -H -F fasta -o
Mh_genbank_genomes --metadata-table metadata_genbank_MH.txt
```

Also added 305 genomes from Will which will be publicly available soon in these accessions (XXXX-XXXX).

Step 2: Create a phylogenetic tree

A total of 2106 genomes were analyzed using mashTree.

```
conda activate metaSNV
module load GCC/7.3.0-2.30  OpenMPI/3.1.1 Mash/2.1

mashtree genome_folders/All_Mh_genomes/*.gz --numcpus 48 --mindepth 0 >
Final_tree_min0.dnd
```

Step 3: Cluster genomes into PSVs

Next, the goal is to analyze the phylogenetic tree with clustering data and determine the best method and threshold combination that breaks up your genome into clusters that "make sense" to you and the purposes of your analysis. Our goal was to maximize the number of clusters, while reducing the number of singletons (genomes not in a cluster), and reducing the size of the largest cluster. We use TreeCluster and the following script to create a series of test results and look for the best tree clustering.

```
#!/bin/bash
# Loop to run TreeCluster.py on a single tree with a combination of methods and
thresholds

# Define the list of threshold values
thresholds=("0.0001" "0.0002" "0.0003" "0.0004" "0.0005" "0.0006" "0.0007"
"0.0008" "0.0009" "0.001" "0.002" "0.003" "0.004" "0.005" "0.006" "0.007" "0.008"
"0.009" "0.01")

# List of methods to test
methods=("avg_clade" "leaf_dist_avg" "leaf_dist_max" "leaf_dist_min" "length"
"length_clade" "max" "max_clade" "med_clade" "root_dist" "single_linkage"
"single_linkage_cut" "single_linkage_union" "sum_branch" "sum_branch_clade")

# Loop through the methods, make sure to change the tree name to match yours
for method in "${methods[@]}"; do
    # Loop through the threshold values
    for threshold in "${thresholds[@]}"; do
        output_file="output_TreeCluster_labels_${method}_t_${threshold}.txt"
        command="TreeCluster.py -i Final_tree_min0.dnd -o $output_file -t
$threshold -m $method"
        echo "Running: $command"
        eval $command
    done
done
```

Now, you'll have many ".txt" files, each with two columns containing the genome file names and the proposed cluster number.

To evaluate all these results, we parse the text files and summarize them using the following command:

```
python parse_ClusterTree_results.py '*.txt' output_results.csv
```

Manual evaluation of cluster membership

We selected the results from a few clustering methods and thresholds that balanced the number of clusters, number of singletons, and the average cluster membership. Then, we visualized the tree with the metadata file for each method using TreeViewer. Based on visual inspection, we selected the clustering method and threshold that performed best in clustering genomes.

Then, to help inspection, tree nodes with genomes in the same cluster were collapsed into a "cartoon node", leaving only singletons which were not clustered. Finally, we followed a protocol of clustering singletons into a nearby PSV only if they were part of a monophyletic node with a PSV group. Any other singleton not meeting this criteria was labeled as a singleton with a unique number. We manually updated the metadata file with a new column with the "ProposedLabel" for each genome, including the clusters from TreeCluster, the manual clustering for singletons, and the new unique singleton labels.

Step 4: Relabel genomes with new IDs

This script makes new taxIDs for each unique PSV and labels genomes appropriately while also creating a names.dmp and nodes.dmp file for use in making the custom kraken database.

The first argument is the metadata file which must contain

```
#
python parse_genome_labels_proposed_PSV.py --metadata_file
Final_output_TreeCluster_labels_avg_clade_t_0.0003.txt --genome_folder
genome_folders/All_Mh_genomes/ --output_folder
genome_folders/ReLabeled_PSV_Mh_genomes --label_prefix "Mh_" --parent_taxid
"75985"
```

This other script just uses the ncbi metadata file to relabel genomes using the taxid and name used by NCBI. Depending on your database structure, you might not need this. We elected that the analysis workflow would start by running reads against a "screening database" with a broad base of genomes. We used "plusPF" from the Ben Langmead kraken database repository. Then we will extract reads classified as Pasteurellaceae and any of the "children" taxa in it's lineage. Lastly, we will run those reads against our "confirmation database" which will include the PSV labeled Mannheimia genomes along with all other Pasteurellaceae genomes.

In my experience, trying to make a custom database with these genomes without changing the file headers led to some genomes not being processed correctly. Re-labeling these genomes with the taxaId given by NCBI and in the format specified by kraken2 seems to be a work around.

```
python parse_NCBI_genome_labels.py metadata_genbank_Pasteur_noMH.txt
genbank_Pasteurellaceae_genomes_noMH/ Labeled_Pasteur_genbank_genomes
```

Step 5: Build kraken database

Start by downloading the NCBI taxonomy and creating the kraken2 database folder as shown below. Note, once the db folder is made, you can use the command `kraken2-build --download-library bacteria --db $DBNAME` to download various types of genomes to include in the database. Full list here.

```
kraken2-build --download-taxonomy --db
/scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db
```

Then, we'll modify some of those files by adding output from the genome labeling step for the PSV genomes. Change the directory to match the path to your labeled genomes and to fit with your database name.

```
cat
/scratch/user/enriquedoster/clean_Mh_database/genome_folders/ReLabeled_PSV_Mh_geno
mes/names.dmp >>
/scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db/taxonomy/
names.dmp

cat
/scratch/user/enriquedoster/clean_Mh_database/genome_folders/ReLabeled_PSV_Mh_geno
mes/nodes.dmp >>
/scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db/taxonomy/
nodes.dmp

cat
/scratch/user/enriquedoster/clean_Mh_database/genome_folders/ReLabeled_PSV_Mh_geno
mes/accession2taxid >>
/scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db/taxonomy/
nucl_gb.accession2taxid
```

Before adding the labeled genomes to the database, we have to unzip the fasta files.

```
gunzip
/scratch/user/enriquedoster/clean_Mh_database/genome_folders/ReLabeled_PSV_Mh_geno
```

```
mes/*
```

Now we can update genomes that will be included in the kraken2 database. In the following steps, I'll use a "for" loop to go through all PSV labeled genomes and add them one at a time. I do the same thing for the Pasteurellaceae genomes and then finally run the command to build the kraken database.

I recommend running the following steps with an sbatch script as it can take a few hours depending on the number of genomes in the database.

```
#add PSV Mh genomes to database
for file in /scratch/user/enriquedoster/clean_Mh_database/genome_folders/ReLabeled_PSV_Mh_genomes/*a
do
    echo $file
    kraken2-build --add-to-library $file --db /scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db
done

# Add Pasteurellaceae genomes
for file in /scratch/user/enriquedoster/clean_Mh_database/genome_folders/Labeled_Pasteur_genbank_genomes/*a
do
    echo $file
    kraken2-build --add-to-library $file --db /scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db
done


# Build database
kraken2-build --build --db /scratch/group/vero_research/databases/kraken2/Pasteurellaceae_Mh_PSV_db
```