Building a Reference Tree and Phylogenetic Placement of Amplicon Sequence Variants (ASVs)

David Kaplan

10/5/20

The whole process can is comprised of a few distinct steps:

1. Downloading the reference sequences that will be used to build the reference tree
2. Aligning the reference sequences using multiple alignment
3. Filter the reference sequences and realign
4. Building the reference tree
5. Doing phylogenetic placement of your ASVs on the reference tree
6. Exporting the files into a usable format for Python

**Downloading reference sequences**

To build the reference tree, we used a set of sequences that are high quality. The database we used to download the sequences was RDP. We only used Typed, isolated strains of good quality that were longer than 1200 base pairs (bp) long. We downloaded the Bacteria and Archaea datasets without any gaps in fasta format. You also need to download these in genbank format for later.

**Aligning the reference sequences**

Once you have the sequences described above*,* we use the RDP aligner with for the Bacteria 16S gene to align the fasta sequences. This aligner can take about half a day for these many sequences.

**Filter the reference sequences and realign**

When you look at the output of this alignment (I recommend you use the program *Jalview* for visualization), you will see that the total length of the alignment is ~8000 bp long. This does not make sense because the average length of the Bacterial 16S gene is 1522 bp long. These extra bp come from insertions of the sequences that could come from an array of different things – missassembly of the 16S gene when they are sequencing the reference sequences, mislabeling of genes, sequence is not actually a 16S, etc. We want to get the total length of the multiple alignment closer to 1522. These were the steps that we took to filter the reference sequences. Note that all of the below scripts are run in Python 3:

1. Delete sequences with unaligned lengths >= 1600. We use the prefiltering script *phylogenetic\_placement/prefilter\_reference\_sequences.py*
2. Delete sequences with insertions that are *rare*. We defined a rare insertion as a base pair position in the aligned sequence space where there were 5 or less sequences that had ungapped base pairs in that position. We used the script *phylogenetic\_placement/filter\_reference\_sequences.py* to perform this filtering.
3. Perform the same RDP alignment on the subset of the sequences produced at the end of part 2.

**Build the reference tree with pplacer**

Now that we have reference sequences that are multiply aligned, we are ready to build the reference tree/reference package. The script is based on the common repository in ErisOne: `*/data/cctm/16S\_rRNA/16S\_rRNA\_Analysis\_pipelines/PhylogeneticPlacement\_buildRefPackage*`. The only things that re different that we need to subsample the RDP references for the sequences that we filtered, and we are using the alignment from RDP instead of using MUSCLE align.

First we need to make the genbank files with our filtered subset of reference sequences. Place those files in *phylogenetic\_placement/build\_ref\_package/example/Download\_DBs*. Next, copy the file *tmp/rdp\_combined\_prefiltered\_gapfiltered\_align.fa* in *phylogenetic\_placement/build\_ref\_package/example/alignments.fa*. Make sure to rename it to *alignments.fa*.

You need to run the script to build the reference tree with python 2. I ran this with Python 2.7.18. On ErisOne run:

`conda create -n phyloplacement\_py27 python=2.7.18`

`pip install bipython=1.76`

Then you can run *phylogenetic\_placement/build\_ref\_package/src/Build\_RefPackage\_pplacer.py*. I ran the script with the arguments:

`python Build\_RefPackage\_pplacer.py -a /data/cctm/darpa\_perturbation\_mouse\_study/phylogenetic\_placement/build\_ref\_package/example -v RDP-11-5`

**Phylogenetic Placement**

We finally get to the placement! Before we start, make a python3 environment and install the packages:

`pip install biopython`

`pip install ete3`

`pip install six`

`pip install guppy3`

`conda install -c bioconda hmmer`

`conda install -c bioconda pplacer`

Which we run with the script *phylogenetic\_placement/scripts/place\_seqs.py*. To increase accuracy of the placement, we only do placement in the v4 region of the reference sequences. To do that we first need to find the start and end positions of the v4 region. We do this by locating the universal primers which are:

5’-[Illumina adaptor]-[unique barcode]-[sequencing primer pad]-[linker]-[primer] Read 1 (fwd primer): AATGATACGGCGACCACCGAGATCTACAC-NNNNNNNN-TATGGTAATT-GTGTGCCAGCMGCCGCGGTAA

Read 2 (rev primer):

CAAGCAGAAGACGGCATACGAGAT-NNNNNNNN-AGTCAGTCAG-CCGGACTACHVGGGTWTCTAAT

Run place\_seqs.py with:

` python scripts/place\_seqs.py --v4-region-start 868 --v4-region-end 1161 --refpkg build\_ref\_package/example/RDP\_RefPackage/refpkg/RDP-11-5\_TS\_Processed.refpkg/ --query-reads data/query\_reads.fa --verbose 1`

MITRE input data:

* Abundance table
  + Counts
* Sample metadata table
  + Sample ID, Subject ID, timepoint
* Subject data table
  + i.e. targets
* pplacer results
  + .jplace file
  + Placement.jplace
* Sequence key
  + OTU to identifier
  + Asvs\_to\_place.fa
* Pplacer sequence information:
  + RDP-11-5\_TS\_Processed\_taxaTable.csv
* Placement table
  + Dada2-taxonomy-rdp.csv