Analysis for reference based OTU picking from DictDb, then Greengenes, then denovo OTU picking from both.

1. Using Uclust, reference based OTU picking with DictDb as a reference database. The seqs that failed to cluster to a DictDb reference sequence will be used in next step.

parallel\_pick\_otus\_uclust\_ref.py -i AllDiet\_seqs.fna -o ref\_otus\_from\_dictdb/ -r DictDb/DictDb\_v3\_V4.fasta -O 10

1. Sequence identifiers (in .txt file) of sequences that failed to cluster to the DictDb database will be used to filter the seqs out of the original .fna file and the resulting file will be stored in ‘ref\_otus\_from\_gg’ folder.

filter\_fasta.py -f AllDiet\_seqs.fna -o ref\_otus\_from\_gg/ref\_fail\_from\_dictdb.fasta -s ref\_otus\_from\_dictdb/AllDiet\_seqs\_failures.txt

1. The filtered .fna file will now be used to cluster OTUs using the Greengenes database. The seqs that fail to cluster to a Greengenes reference will be used in the next step.

parallel\_pick\_otus\_uclust\_ref.py -i ref\_otus\_from\_gg/ref\_fail\_from\_dictdb.fasta -o ref\_otus\_from\_gg/ -r greengenes/97\_otus\_v4.fasta -O 10

1. Sequence identifiers of seqs that failed to cluster to a greengenes reference will be used to filter the seqs out of the original .fna file and the resulting file will be stored in ‘denovo\_otus’ folder.

filter\_fasta.py -f AllDiet\_seqs.fna -o denovo\_otus/ref\_fail\_from\_gg.fasta -s ref\_otus\_from\_gg/ref\_fail\_from\_dictdb\_failures.txt

1. The filtered .fna file will be clustered denovo using a combined dictdb/greengenes database as the training set.

pick\_otus.py -i denovo\_otus/ref\_fail\_from\_gg.fasta -o denovo\_otus/ -m uclust\_ref -r gg\_dictdb/gg\_dictdb\_filtered.fasta

1. Representative sets were made for dictdb set and gg set

pick\_rep\_set.py -i ref\_otus\_from\_dictdb/AllDiet\_seqs\_otus.txt -r DictDb/DictDb\_v3\_V4.fasta -o dictdb\_Rep\_Set.fasta

pick\_rep\_set.py -i ref\_otus\_from\_gg/ref\_fail\_from\_dictdb\_otus.txt -r greengenes/97\_otus\_v4.fasta -o gg\_Rep\_Set.fasta

1. A rep set was picked for the denovo set

pick\_rep\_set.py -i denovo\_otus/ref\_fail\_from\_gg\_otus.txt -f denovo\_otus/ref\_fail\_from\_gg.fasta -o denovo\_Rep\_Set.fasta

1. Align the denovo rep set for chimera checking

parallel\_align\_seqs\_pynast.py –I denovo\_Rep\_Set.fasta -o ref\_failures/ -e 75 -O 12

1. Chimera checking on the denovo aligned rep set

parallel\_identify\_chimeric\_seqs.py -i ref\_failures/denovo\_Rep\_Set\_aligned.fasta -a gg\_dictdb/gg\_dictdb\_aligned.fasta -O 12

1. Adjust files for processing

$ find ./ -name '\*\_otus.txt' -exec cat {} ';' >otus\_complete.txt

$ grep '>' ref\_failures/denovo\_Rep\_Set\_failures.fasta | tr -d '>' | cut -d\ -f 1,1 >filter\_unaligned.txt

$ cat filter\_unaligned.txt denovo\_Rep\_Set\_aligned\_chimeric.txt > bad\_otus.txt

1. concatenate the 3 rep sets

$ cat dictdb\_Rep\_Set.fasta gg\_Rep\_Set.fasta denovo\_Rep\_Set\_aligned.fasta >Rep\_Set\_aligned.fasta

1. unalign the rep set

$ tr -d '-' <Rep\_Set\_aligned.fasta >Rep\_Set.fasta

1. assign taxonomy

assign\_taxonomy.py -i Rep\_Set.fasta -t gg\_dictdb/gg\_dictdb\_tax.txt -r gg\_dictdb/gg\_dictdb\_filtered.fasta -o RDP\_classifier/

1. make otu table

make\_otu\_table.py -i otus\_complete.txt -o raw\_otu\_table.biom -t RDP\_classifier/Rep\_Set\_tax\_assignments.txt -e bad\_otus.txt

1. filter otu table

filter\_otus\_from\_otu\_table.py -i raw\_otu\_table.biom -o tmp1.biom -n 3

filter\_otus\_from\_otu\_table.py -i tmp1.biom -o otu\_table.biom --min\_count\_fraction 0.00005

1. Rarify all samples to 18000 reads

single\_rarefaction.py -i otu\_table.biom -o even\_table.biom -d 18000

1. These next two commands will create a taxonomy table that you can use to create bar charts or tables of the taxa in your samples. These will output a table for each taxonomic level. The files in the abs/ folder show absolute numbers instead of relative abundance (percentages).

summarize\_taxa.py -i even\_table.biom -o taxa\_summary/

summarize\_taxa.py -i even\_table.biom -o taxa\_summary/abs/ -a

1. This command will create a table with the alpha diversity. There are more metrics available on the Qiime scripts website that you can choose to use. The basic metrics are listed in the command below.

alpha\_diversity.py -i even\_table.biom -o alpha\_even.txt -m osd,simpson,shannon,PD\_whole\_tree -t Rep\_Set\_tree.tree