20230815\_BLAST\_docs

2023-08-15

* This is a short guide on using the NCBI BLAST command line software to identify bacterial taxonomies using DNA sequences
  + The sequences used here are generated from a QIIME run of 16S V4-5 sequences. The goal was to take unclassified taxa of interest (in this case, taxa unclassified to the order or class level for Clostridia/Clostridiales) and get a better ID on who is in the community
* For these runs, I got lucky and the Norman cluster servers had BLAST+ downloaded for use, so I only had to figure out how to download and unpack the reference database (the option “-remote” queues a run on the NCBI servers, so it’s not much different than BLAST-ing on their browser (i.e., fails if there are too many queries)).
* For info on BLAST and installing BLAST+ see the following links (they’re not great instructions, but will get you pointed in the right direction): <https://www.ncbi.nlm.nih.gov/books/NBK52637/>
  + Install BLAST+: <https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>
* To download the database, follow the following:
* Downloading 16S database on the server; all these steps occur on the shell, and should work if copy/pasted
  + Make sure you know where all the files are being written! You’ll need to know their file location for later

wget "ftp://ftp.ncbi.nlm.nih.gov/blast/db/nt.??.tar.gz"  
  
#extract contents of tarballs  
for file in \*.gz  
do  
tar -zxvpf "$file"  
rm "$file"  
done  
  
wget "ftp://ftp.ncbi.nlm.nih.gov/blast/db/taxdb.tar.gz"  
tar -xzf taxdb.tar.gz

* The untarred files are the database files you need. However, there are many “uncultured” and “environmental” sample names in the database, which are not particularly informative, so let’s remove them from consideration. Follow the steps below to reproduce the file.
* I’d recommend remaking this file every so often just to make sure it’s the most recent version
* Download and compile db for excluding any environmental, uncultured, unidentified, taxa
* Use the subtree code from this github <https://github.com/pmenzel/taxonomy-tools#subtree>
  + You’ll need to compile the code by downloading, upacking, and running “make” on the makefile
  + You should be able to do this by navigating to the folder with the makefile:

cd /directory/with/makefile  
make makefile

* Download the taxa file to filter out undesired taxa assignments. Gotta FTP to the location (I used filezilla) <ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxdump.tar.gz>
  + The file is in a weird place: /pub/taxonomy/
  + I had to manually navigate and download the file, which is weird, but so it goes
* Run the following to set up the exclusion list:

#unpack tarball  
tar xf taxdump.tar.gz names.dmp nodes.dmp  
  
#find names to exclude  
grep "scientific name" names.dmp| grep -w -P "uncultured|environmental samples|metagenome|unclassified|unidentified" | cut -f1 >in.txt  
  
#use subtree to make an exclusion list  
/home/suginoka/Metagenomics\_Programs/taxonomy-tools-master/src/subtree -t nodes.dmp -i in.txt > exclusion\_list.txt

* Before running the program below on the HPC, you need to grab files from the QIIME output that end in the following:
  + \*taxonomy.qza
  + \*taxa-plot.qzv
  + \*rep-seqs.qza
  + \*table.qza
  + \*table.qzv
* These are basically zip files, so you can extract the contents into a folder using the native extract function in windows, with 7zip (or other compression/extraction program), or whatever the default option in macs is.
* Within these unzipped folders, you’re looking for the following files (usually stored in the subfolder “data”):
  + taxonomy.tsv
  + level-1.csv through level-7.csv
    - Don’t need these for this tutorial, but these are the kingdom-species OTU tables per sample
  + dna-sequences.fasta
  + feature-table.biom
  + feature-frequency-detail.csv
* The feature-table.biom file has to be processed into a tsv format, rather than a biom format.
  + This table contains the OTU table corresponding to the seq\_id (not taxa id, but the random string that tracks each unique sequence) per sample.
* This conversion can be done with the following code and is relatively quick to run (note that the commands to convert to/from biom format are built into QIIME):

biom convert -i feature-table.biom -o feature\_otu\_table\_from\_biom.txt --to-tsv

* To find the sequences of interest, you’re going to need:
  + taxonomy.tsv
  + dna-sequences.fasta
* Note that here I am specifically pulling Clostridia that are unclassified past the order or class levels
  + The code will need to be rewritten to pull the specific taxa IDs of interest, so make sure you check that your output is correct

#merging taxonomy file with sequence file for BLAST analysis  
tax<-read.table("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Dec2021/indole\_blast\_files/taxonomy.tsv",sep="\t",header=T)  
seq<-read.table("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Dec2021/indole\_blast\_files/dna-sequences.fasta",header=F)  
  
#reformat from fasta to table format  
seq.df<-data.frame(Column1 = seq$V1[c(TRUE, FALSE)],   
 Column2 = seq$V1[c(FALSE, TRUE)])  
seq.df$Column1<-gsub(">","",seq.df$Column1)  
  
#merge sequence table to taxa ID table  
df<-merge(tax,seq.df,by.x="Feature.ID",by.y="Column1")  
  
##################  
#pull specific sequences of interest  
df.1<-df[grep(c("Clostridia"),df$Taxon),]  
#keep only unclassified order and class  
#for many reasons, this isn't a great way of filtering taxa names, but I've basically capped the number of characters that can be in the string name since, if it's been classified past class, there will be more characters than 70 or so (hence choosing 75 as a cutoff)  
df.o<-df.1[!nchar(df.1$Taxon)>75,]  
  
#write.csv(df.o,"C:/Users/ksugino/Desktop/taxonomy\_results/20230328\_all\_seqs\_formatted.csv",row.names = F,quote=F)  
  
  
#reformat into fasta format for BLAST  
format<-data.frame()  
a<-df.o  
for(i in 1:nrow(a)){  
 format<-rbind(format,paste0(">",a$Feature.ID[i]),a$Column2[i])  
}  
  
#write.table(format,"Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Dec2021/indole\_blast\_files/202300801\_clostridia\_seqs.txt",row.names = F,quote=F)

* Now, you can put the files on the HPC and run something like the code below; note that you need your fasta-formatted sequences, the BLAST database, and the exclusion list to remove all uncultured/environmental IDs from the search.
  + You’ll need to change the file locations for a few commands:
    - Change /scratch/suginoka/BLAST\_db/ to the location of the BLAST database
    - Change 20230411\_lacto\_seqs.txt to the text file containing the fasta-formatted sequences
    - Change /scratch/suginoka/BLAST\_db/db/nt to the subfolder with the BLAST database (idk if this is needed, but it’s how I ran it and it works?)
    - Change /scratch/suginoka/BLAST\_db/exclusion\_list.txt to the location/name of the file containing the list of taxa names to exclude from the search

export BLASTDB=$BLASTDB:/scratch/suginoka/BLAST\_db/  
  
 module load BLAST+/2.13.0-gompi-2022a  
 blastn -query 202300801\_clostridia\_seqs.txt -db /scratch/suginoka/BLAST\_db/db/nt -out output.txt -outfmt "6 qseqid sseqid evalue pident stitle staxids sscinames scomnames sblastnames sskingdoms salltitles stitle" -evalue 1e-30 -task megablast -negative\_taxidlist /scratch/suginoka/BLAST\_db/exclusion\_list.txt

* The next steps use the following files:
  + output.txt (generated from the step above)
  + feature-frequency-detail.csv
  + The edited feature\_otu\_table\_from\_biom.txt output from above

df<-read.table("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/output.txt",fill=T,sep="\t",check.names = F,strip.white = F)  
  
#this code is extremely stupid, i'm so sorry  
#note that this code is different than that used in March 2023 for the Chabaan lab's probiotic study. It needs to be different because the separator used between the web-browsers method and the command line method of BLAST are different (I can use \t here but had to use a \*space\* in the web browser method, so it was much clunkier to work with--this code is still kind dumb though)  
#if you see this an know of a better implementation (e.g., awk or something?) please let me know at kameron.sugino@gmail.com  
#if not, well, best of luck!  
  
#collect bacterial ID, and assembly score  
query.id<-df[,1]  
bac.name<-gsub("^([^ ]+ [^ ]+).\*","\\1",df[,5])  
bac.score<-df[,4]  
  
#assemble and take the bacterial IDs with the max alignment score  
#I don't know if there's a better way to do this, but here we are  
reform<-data.frame(query.id,bac.name,bac.score)  
reform.max<-aggregate(reform$bac.score, by = list(reform$query.id), max)  
reform.max.merge<-merge(reform.max,reform,by.x=c("Group.1","x"),by.y=c("query.id","bac.score"))  
reform.max.merge$bac.name<-gsub("\\[|\\]","",reform.max.merge$bac.name)  
  
reform.table<-data.frame(table(reform.max.merge$Group.1,reform.max.merge$bac.name))  
reform.table.e<-reform.table[reform.table$Freq>0,]  
  
#read in and merge total counts (for all samples) for each seq ID  
cnt<-read.csv("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/feature-frequency-detail.csv",header=F)  
  
df.m<-merge(reform.table.e,cnt,by.x="Var1",by.y="V1")  
colnames(df.m)<-c("Feature\_ID","BLAST\_Taxa\_ID","BLAST\_ID\_Count","Sequence\_Count")  
  
final\_ids<-aggregate(df.m$BLAST\_ID\_Count, by = list(df.m$Feature\_ID), max)  
final\_ids.f<-merge(final\_ids,df.m,by.x=c("Group.1","x"),by.y=c("Feature\_ID","BLAST\_ID\_Count"))  
  
#still have duplicate seq IDs, will combine names since it's unclear if there's a best bacterial ID  
dupes<-final\_ids.f[final\_ids.f$Group.1 %in% final\_ids.f$Group.1[duplicated(final\_ids.f$Group.1)],]  
dupes.m<-aggregate(dupes$BLAST\_Taxa\_ID,list(dupes$Group.1),paste, collapse = ",")  
  
#create matrix of no dupes, of dupes, and then merge them  
nodupes<-final\_ids.f[!final\_ids.f$Group.1 %in% final\_ids.f$Group.1[duplicated(final\_ids.f$Group.1)],]  
colnames(nodupes)<-c("seq\_id","BLAST\_ID\_Count","BLAST\_Taxa\_ID","Sequence\_Count")  
  
dupes.m.merge<-merge(dupes.m,dupes,by.x=c("Group.1"),by.y=c("Group.1"))  
dupes.m.merge.e<-dupes.m.merge[!duplicated(dupes.m.merge$Group.1),]  
colnames(dupes.m.merge.e)<-c("seq\_id","BLAST\_Taxa\_ID","BLAST\_ID\_Count","Old\_IDs","Sequence\_Count")  
dupes.f<-dupes.m.merge.e[,-4]  
  
blast\_final<-rbind(nodupes,dupes.f)  
  
#write.csv(blast\_final,"Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/20230808\_feature\_summary\_BLAST+\_aligned.csv",row.names = F)

* Assuming the code above worked, great! (If not, sorry!)
* We still have some processing to do before the data are ready to use.
* We’ll combine the feature summary generated above with the biom table and the metadata file (so we know which sample belongs to which treatment)
  + Again, this code will likely need to be edited to fit your data

#take the new IDs and merge them with the feature otu table from the \*.biom files  
df<-read.csv("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/20230808\_feature\_summary\_BLAST+\_aligned.csv")  
feat<-read.table("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/table.from\_biom.txt",header=T,fill=T)  
meta<-read.csv("Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/FMT\_mouse\_results/FMT-metadata.tsv",sep="\t")  
  
#merge complete otu table with ids to calculate relative abundance later  
otu.comp<-data.frame(gsub("X","",colnames(feat[,-1])),t(feat[,-1]))  
otu.comp.f<-merge(meta,otu.comp,by.x="SeqID",by.y="gsub..X.......colnames.feat....1...")  
  
final.otu<-merge(df,feat,by.x="seq\_id",by.y="OTU")  
#checking that the total number of reads pre seq\_id is equivalent to the total number of reads imported from feature-frequency-detail.csv  
rowSums(final.otu[,-c(1:4)])==final.otu$Sequence\_Count  
  
#need to merge taxa IDs again at this step, otherwise there will be multiple columns of taxa with the same bac. ID  
otu.temp<-final.otu[,-c(1:2,4)]  
otu.test<-aggregate(data = otu.temp, . ~ BLAST\_Taxa\_ID , sum)  
  
id<-gsub("X","",colnames(final.otu[,-c(1:4)]))  
otu<-data.frame(id,t(otu.test[,-c(1)]))  
colnames(otu)<-c("id",otu.test$BLAST\_Taxa\_ID)  
  
df.final<-merge(meta,otu,by.x="SeqID",by.y="id")  
  
#write.table(df.final,"Z:/Research/Friedman-Jonscher Lab/Microbiome data/Indole and PQQ MB/16S\_seq\_Raw\_files\_&\_QIIME\_output/Nov2022/Kams\_QIIME\_files/20230808\_BLAST\_OTU\_table.txt",row.names = F, quote = F,sep='\t')

* The output df.final is the otu file that contains the newly classified reads!
* Note that this file only contains a *subset* of the total number of reads from microbiome, comprised of whatever sequences you pulled and BLASTED at the beginning