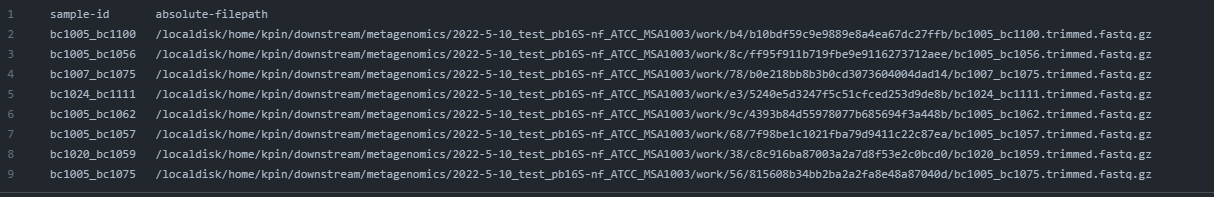
参考：[Analyzing PacBio HiFi Mock Community 16S Data with QIIME 2 · PacificBiosciences/pb-16S-nf Wiki (github.com)](https://github.com/PacificBiosciences/pb-16S-nf/wiki/Analyzing-PacBio-HiFi-Mock-Community-16S-Data-with-QIIME-2)

下机得到经过SMRT link质控的CCS reads

#首先制作manifest文件



echo -e 'sample-id\tabsolute-filepath' > sample.manifest

#创建一个名为 sample.manifest 的文件，并写入表头，包括 sample-id 和 absolute-filepath 两列

for i in data/\*.fastq.gz; do fname=$(basename ${i}); sample=$(echo ${fname} |\

awk '{print gensub(/.\*\_(bc.\*)--.\*(bc.\*).hifi\_reads.fastq.gz/, "\\1\_\\2", "g", $0)}');\

echo -e "${sample}\t$(readlink -f ${i})" >> sample.manifest;\

done

#获取文件路径 i 中的文件名，并赋值给变量 fname。

#使用 awk 命令提取样本ID。通过正则表达式提取文件名中的两个关键部分（bc.\*），然后将它们拼接在一起，作为样本ID，并赋值给变量 sample

#将样本ID和文件路径写入 sample.manifest 文件，用制表符 \t 分隔，并通过 >> 追加写入文件。

#导入qiime2

qiime tools import --type 'SampleData[SequencesWithQuality]' \

--input-path sample.manifest \

--output-path samples.qza \

--input-format SingleEndFastqManifestPhred33V2

#可视化reads

qiime demux summarize --i-data samples.qza \

--o-visualization samples.demux.summary.qzv

#制作分组信息

awk 'FNR==1{print "sample\_name","condition"}; FNR>1 && FNR <6 {print $1,"RepA"}; FNR>5 {print $1, "RepB"}' OFS=$'\t' sample.manifest > metadata.tsv

#行号大于1且小于6的行，输出第一列内容（样本名）和 "RepA"（条件信息）

awk 'FNR==1{print "sample\_name","condition"}; FNR>1 && FNR <33 {print $1,"H1"}; FNR>32 && FNR <45{print $1, "H2"}; FNR>44{print $1,"H3"}' OFS=$'\t' sample.manifest > metadata.tsv

#去噪高保真读取到 ASV 中

qiime dada2 denoise-ccs --i-demultiplexed-seqs ./samples.qza \

--o-table dada2-ccs\_table.qza \

--o-representative-sequences dada2-ccs\_rep.qza \

--o-denoising-stats dada2-ccs\_stats.qza \

--p-min-len 1000 --p-max-len 1600 \

--p-front AGRGTTYGATYMTGGCTCAG --p-adapter RGYTACCTTGTTACGACTT \

--p-n-threads 20

#--p-front 为27F引物，--p-adapter为1492R引物

#--p-min-len 1000并指定最小和最大序列长度。对于 16S 序列，建议分别使用 1000 和 1600

#汇总去噪统计

qiime metadata tabulate --m-input-file ./dada2-ccs\_stats.qza \

--o-visualization ./dada2-ccs\_stats.qzv

#生成另一个汇总每个样本观察到的特征（此处为 ASV）的表格：

qiime feature-table summarize --i-table ./dada2-ccs\_table.qza \

--m-sample-metadata-file metadata.tsv \

--o-visualization ./dada2-ccs\_table.qzv

#抽平（不要求）

qiime diversity alpha-rarefaction --i-table dada2-ccs\_table.qza \

--m-metadata-file metadata.tsv \

--o-visualization ./alpha-rarefaction-curves.qzv \

--p-min-depth 10 --p-max-depth 10000

#整理特征表

qiime tools export --input-path dada2-ccs\_table.qza --output-path arrange

biom convert -i arrange/feature-table.biom -o arrange/otu\_table.tsv --to-tsv

cd arrange

sed -i '1d' otu\_table.tsv

sed -i 's/#OTU ID/ASV/' otu\_table.tsv

#R更改特征表名称（在qiime2中输入R弹出R的程序）

install.packages("pacman")

R：

library(pacman)

pacman::p\_load(tidyverse,magrittr,stringr)

otu <- "otu\_table.tsv" %>%

read.delim(check.names = FALSE,header = T,sep="\t")

rown <- paste0("ASV",seq\_len(nrow(otu)))

otu[,1] <- rown

colnames(otu)[1] <- paste0("ASV",colnames(data)[1])

write.table (otu,file ="otu\_table.tsv",sep ="\t",row.names = T)

（生成tsv文件后，用excel删除第一列，第一行后移一位）

#将TSV格式转换为BIOM

biom convert -i otu\_table.tsv -o feature-table.biom --to-hdf5 --table-type="OTU table"

qiime tools import --input-path feature-table.biom --type 'FeatureTable[Frequency]' --input-format BIOMV210Format --output-path otu\_table-rename.qza

#导出代表性序列fasta

cd ..

qiime tools export --input-path dada2-ccs\_rep.qza --output-path arrange

cd arrange

less dna-sequences.fasta |paste - -|sed '1i ASVID,seq' > rep.fa

R：

library(pacman)

pacman::p\_load(tidyverse,magrittr,stringr)

rep <- "rep.fa" %>%

read.delim(check.names = FALSE, row.names = 1) %>%

set\_rownames(paste0(">ASV", seq\_len(nrow(.))))

write.table (rep,file ="rep.xls", sep ="\t", row.names = T)

Qiime2：

less rep.xls|sed '1d'|sed 's/"//g'|sed 's/\r//g'|tr "\t" "\n" > rep-seqs.fasta

#将代表序列转换成qza格式

qiime tools import --type 'FeatureData[Sequence]' --input-path rep-seqs.fasta --output-path rep-seqs-rename.qza

#特征表统计

qiime feature-table summarize --i-table otu\_table-rename.qza --o-visualization otu\_table-rename.qzv

#代表序列统计

qiime feature-table tabulate-seqs --i-data rep-seqs-rename.qza --o-visualization raw.fq.list

#比对代表性序列，并构建系统发育树

#sepp插入法

time qiime fragment-insertion sepp --i-representative-sequences rep-seqs-rename.qza --i-reference-database sepp-refs-silva-128.qza --o-tree tree.qza --o-placements tree\_placements.qza --p-threads 18 --p-debug --verbose

#常规mafft

qiime phylogeny align-to-tree-mafft-fasttree --i-sequences rep-seqs-rename.qza --o-alignment aligned-rep-seqs.qza --o-masked-alignment masked-aligned-rep-seqs.qza --o-tree unrooted-tree.qza --o-rooted-tree rooted-tree.qza

#

#ASV分类（这里用的是vsearch而不是Naive Bayesian，因为16S序列分类器的一个流行选择是基于朴素贝叶斯的分类器。然而，根据我们的经验，这对于我们的全长模拟社区ASV来说不是很敏感。附带了一个名为的工具，该工具使用了类似的朴素贝叶斯分类器方法，该方法运行良好（在模拟社区中），但需要在R中使用。当然也可以用classify-sklearn）

qiime feature-classifier classify-consensus-vsearch \

--i-query arrange/rep-seqs-rename.qza \

--i-reference-reads references/silva-138-99-seqs.qza \

--i-reference-taxonomy references/silva-138-99-tax.qza \

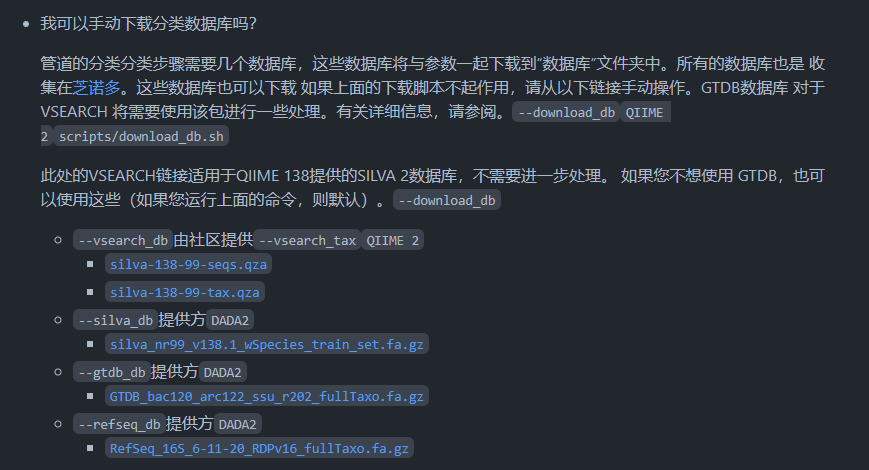
--o-classification ./taxonomy.vsearch.qza \

--p-threads 20 \

--p-maxrejects 100 --p-maxaccepts 10 --p-perc-identity 0.97

#指定对分类命中项的搜索将在 100 个不匹配序列后停止，如果有 10 个分类命中项，则将停止搜索。结合其默认值为 0.51，超过一半的“命中”（此处设置为 10）需要在任何特定分类级别具有相同的分配，以便对序列进行自信分类。最后，指定分类命中率必须至少为 97%。这些选项可确保运行速度非常快（当然比传统方法快得多），并且仅保持高度相同的命中。

#https://data.qiime2.org/2022.2/common/silva-138-99-seqs.qza

#https://data.qiime2.org/2022.2/common/silva-138-99-tax.qza 

#导出文件进行下游分析

qiime tools export --input-path otu\_table-rename.qza --output-path R

biom convert -i ./R/feature-table.biom -o ./R/otu\_table.tsv --to-tsv

qiime tools export --input-path taxonomy.vsearch.qza --output-path R

qiime tools export --input-path rooted-tree.qza --output-path R

cd R

mv tree.nwk rooted\_tree.tre