16S Tutorial

xyz

2021/9/3

Contents

Illumina single end

Splited fastq file

```
QCstat<-read.table("01.CleanData/QCstat.xls",skip = 1)
colnames(QCstat) <-
  c(
    'Sample Name',
    'Raw PE(#)',
    'Combined(#)',
    'Qualified(#)',
    'Nochime(#)',
    'Base(nt)',
    'AvgLen(nt)',
    'Q20',
    'Q30',
    'GC%',
    'Effective%'
  )
sampleName<-QCstat$`Sample Name`
df <- data.frame(
  `sample-id` = sampleName,
  `absolute-filepath` = paste0(
    "/mnt/e/xiongyi/Chile/01.CleanData/",
    sampleName,
    "/",
    sampleName,
    ".fastq"
  ),
  stringsAsFactors = F
)
colnames(df) <- c("sample-id", "absolute-filepath")
write.table(
  df,
  "metadata.tsv",
  quote = F,
  row.names = F,
  sep = "\t"
)
```

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
mkdir qualityTest
cd qualityTest
fastp -i ../01.CleanData/D1/D1.fastq -o D1.filterd.fq
```

View qzv file

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime tools import \
  --type 'SampleData[SequencesWithQuality]' \
```

```
  --input-path metadata.tsv \
  --output-path chile.qza \
  --input-format SingleEndFastqManifestPhred33V2

# check quality
qiime demux summarize \
    --i-data chile.qza \
    --o-visualization chileSummary.qzv
```

Illumina paried end

Import to Qiime

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime tools import \
  --type MultiplexedPairedEndBarcodeInSequence \
  --input-path ../data/fastq \
  --output-path ../temp/multiplexed-seqs.qza
```

```
df<-read.table("../data/mapping.txt",header = F)
df2<-data.frame(
  `sample-id` = paste0("J",df$V1),
  barcode=df$V2
)
colnames(df2) <- c("sample-id", "Barcode")
write.table(
  df2,
  "../data/metadata.tsv",
  quote = F,
  row.names = F,
  sep = "\t"
)
```

Demux

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
# change tmp path
# export TMPDIR=/mnt/f/qiimeTMP
qiime cutadapt demux-paired \
  --i-seqs ../temp/multiplexed-seqs.qza \
  --m-forward-barcodes-file ../data/metadata.tsv \
  --m-forward-barcodes-column Barcode \
  --p-error-rate 0 \
  --o-per-sample-sequences ../temp/demultiplexed-seqs.qza \
  --o-untrimmed-sequences ../temp/untrimmed.qza \
  --verbose &> ../temp/demux.txt
```

Trim adapters

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime cutadapt trim-paired \
  --i-demultiplexed-sequences ../temp/demultiplexed-seqs.qza \
  --p-front-f GTGCCAGCMGCCGCGG \
  --p-error-rate 0 \
  --o-trimmed-sequences ../temp/trimmed-seqs.qza \
  --verbose &> ../temp/trim.txt
```

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime demux summarize \
  --i-data ../temp/trimmed-seqs.qza \
  --o-visualization ../temp/trimmed-seqs.qzv
```

Remove primer

515F GTGCCAGCMGCCGCGG 907R CCGTCAATTCMTTTRAGTTT

remove sequence shorter than 300

```
library(Biostrings)
# CCGTCAATTCMTTTRAGTTT...CCGCGGCKGCTGGCAC
reverseComplement(DNAString("GTGCCAGCMGCCGCGG...AAACTYAAAKGAATTGACGG"))
```

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime cutadapt trim-paired \
  --p-cores 6 \
  --i-demultiplexed-sequences ../temp/trimmed-seqs.qza \
  --p-adapter-f GTGCCAGCMGCCGCGG...AAACTYAAAKGAATTGACGG \
  --p-adapter-r CCGTCAATTCMTTTRAGTTT...CCGCGGCKGCTGGCAC \
  --p-error-rate 0.2 \
  --o-trimmed-sequences ../temp/primer_trimed.qza \
  --verbose &> ../temp/primer_trimed.log

qiime demux summarize \
    --i-data ../temp/primer_trimed.qza \
    --o-visualization ../temp/primer_trimed.qcSummary.qzv
```

Roche 454

Install sff2fastq

```
# github link format for private project
# git@github.com:xyz1396/454pipeline.git
```

```
cd ..
git clone git://github.com/indraniel/sff2fastq.git
cd sff2fastq
make
```

Convert

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
/mnt/e/xiongyi/454Data_JiaLab-2020.12.31/from454sff/sff2fastq/sff2fastq \
  -o qu1.fq \
  JDH5BEV01qu1.sff
/mnt/e/xiongyi/454Data_JiaLab-2020.12.31/from454sff/sff2fastq/sff2fastq \
  -o qu2.fq \
  JDH5BEV02qu2.sff
cat qu*.fq > all.fq
# some base is Q, dada2 cannot parse it
seqkit grep -s -p 'Q' all.fq > all.error.fq
seqkit grep -s -v -p 'Q' all.fq > all.fix.fq
```

Import

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
gzip -c all.fix.fq > all.fastq.gz
qiime tools import \
  --type MultiplexedSingleEndBarcodeInSequence \
  --input-path all.fastq.gz \
  --output-path multiplexed-seqs.qza
```

```
library(readxl)
df<-read_xlsx("sampleList.xlsx")
df2<-data.frame(
  `sample-id` = paste0("J",df$NO.),
  barcode=df$Tag
)
colnames(df2) <- c("sample-id", "Barcode")
write.table(
  df2,
  "metadata.tsv",
  quote = F,
  row.names = F,
  sep = "\t"
)
```

Demux

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime cutadapt demux-single \
  --i-seqs multiplexed-seqs.qza \
  --m-barcodes-file metadata.tsv \
  --m-barcodes-column Barcode \
  --p-error-rate 0 \
  --o-per-sample-sequences demultiplexed-seqs.qza \
  --o-untrimmed-sequences untrimmed.qza \
  --verbose
```

Trim adapters

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime cutadapt trim-single \
  --i-demultiplexed-sequences demultiplexed-seqs.qza \
  --p-front GTGCCAGCMGCCGCGG \
  --p-error-rate 0 \
  --o-trimmed-sequences trimmed-seqs.qza \
  --verbose
```

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime demux summarize \
  --i-data trimmed-seqs.qza \
  --o-visualization trimmed-seqs.qzv
```

For splited sff file

convert

```
ls ../data/sff/ | xargs -n 1 -P 4 \
  bash -c '../sff2fastq/sff2fastq -o ../data/fastq/$0.fq ../data/sff/$0'
```

Make meta data

```
library(stringr)
fileName<-dir("../data/fastq/")
# 454Reads.MID_43.sff.fq is empty
fileName<-fileName[fileName!="454Reads.MID_43.sff.fq"]
sampleID<-str_split(fileName,"\\.",simplify = T)[,2]
filePath<-paste0("/mnt/e/xiongyi/454Data_JiaLab-2020.12.31/from454sff/data/fastq/",fileName)
df <- data.frame(
```

```
  `sample-id` = sampleID,
  `absolute-filepath` = filePath,
  stringsAsFactors = F
)
colnames(df) <- c("sample-id", "absolute-filepath")
write.table(
  df,
  "../data/metadata.tsv",
  quote = F,
  row.names = F,
  sep = "\t"
)
```

Import to qiime

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime tools import \
  --type 'SampleData[SequencesWithQuality]' \
  --input-path ../data/metadata.tsv \
  --output-path ../data/fastq.qza \
  --input-format SingleEndFastqManifestPhred33V2

# check quality
qiime demux summarize \
    --i-data ../data/fastq.qza \
    --o-visualization ../temp/qcSummary.qzv
```

Remove primer

515F GTGCCAGCMGCCGCGG 907R CCGTCAATTCMTTTRAGTTT

remove sequence shorter than 300

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime cutadapt trim-single \
  --p-cores 6 \
  --i-demultiplexed-sequences ../data/fastq.qza \
  --p-adapter GTGCCAGCMGCCGCGG...AAACTYAAAKGAATTGACGG \
  --p-error-rate 0.2 \
  --p-minimum-length 300 \
  --o-trimmed-sequences ../temp/fastq.primer_trimed.qza \
  --verbose &> ../temp/fastq.primer_trimed.log

qiime demux summarize \
    --i-data ../temp/fastq.primer_trimed.qza \
    --o-visualization ../temp/fastq.primer_trimed.qcSummary.qzv
```

Get ASV by dada2

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime dada2 denoise-pyro \
  --p-trunc-len 370 \
  --i-demultiplexed-seqs primer_trimed.qza \
  --p-n-threads 6 \
  --o-table fastq370.table.qza \
  --o-representative-sequences fastq370.rep-seqs.qza \
  --o-denoising-stats fastq370.denoising-stats.qza

qiime metadata tabulate \
    --m-input-file fastq370.denoising-stats.qza \
    --o-visualization fastq370.denoising-stats.qzv
```

Cluster Feature

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime vsearch cluster-features-de-novo \
  --i-sequences fastq370.rep-seqs.qza \
  --i-table fastq370.table.qza \
  --p-perc-identity 0.97 \
  --o-clustered-table fastq370.97.table.qza \
  --o-clustered-sequences fastq370.97.seq.qza \
  --p-threads 6
```

Make annotation database

QIIME2 Data resources Silva 138 SSURef NR99 full-length sequences Silva 138 SSURef NR99 full-length taxonomy

```
cd /mnt/e/xiongyi/454Data_JiaLab-2020.12.31/2021xiongyiAnalysis/db/SILVA_138
qiime feature-classifier extract-reads \
  --i-sequences silva-138-99-seqs.qza \
  --p-f-primer GTGCCAGCMGCCGCGG \
  --p-r-primer CCGTCAATTCMTTTRAGTTT \
  --p-min-length 200 \
  --p-max-length 500 \
  --o-reads silva-138-99-seqs_515F_907R.qza \
  --p-n-jobs 5

qiime feature-classifier fit-classifier-naive-bayes \
   --i-reference-reads silva-138-99-seqs_515F_907R.qza \
   --i-reference-taxonomy silva-138-99-tax.qza \
   --o-classifier silva138_515F_907R_classifier.qza
```

```
qiime feature-classifier classify-sklearn \
  --i-reads ../../denovo.clustered.seq.qza \
  --i-classifier silva138_515F_907R_classifier.qza \
  --o-classification ../../denovo.clustered.taxonomy.naive_bayes.qza
  # need ram larger than 32 gb
  # --p-n-jobs 3
```

Species annotation

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime feature-classifier classify-sklearn \
  --i-reads fastq370.97.seq.qza \
  --i-classifier /mnt/e/xiongyi/454Data_JiaLab-2020.12.31/2021xiongyiAnalysis/db/SILVA_138/silva138_515
  --o-classification denovo.clustered.naive_bayes_taxonomy.qza \
  --p-n-jobs 1
```

Export OTU table and sequence

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate qiime2
qiime tools export \
  --input-path fastq370.97.seq.qza \
  --output-path .

qiime tools export \
  --input-path fastq370.97.table.qza \
  --output-path .

biom convert -i feature-table.biom \
  -o feature-table.tsv --to-tsv

qiime tools export \
  --input-path denovo.clustered.naive_bayes_taxonomy.qza \
  --output-path .
```

Select OTU with more than 1 sequence

```
library(tidyverse)
otu <- read.table("feature-table.tsv",header = T,
                  sep = "\t",comment.char="",skip=1)
colnames(otu)[1]<-"OTU.ID"
tax <- read.table("taxonomy.tsv",
                  header = T,
                  sep = "\t")
```

```
# 3 OTU without annotation
sum(tax$Taxon == "Unassigned")
# tax<-tax[tax$Taxon != "Unassigned",]
sum(is.na(tax$Taxon))
rankTaxon <- data.frame(Taxon = tax$Taxon)
rankTaxon <-
  separate(rankTaxon, Taxon, sep = "; ", into = as.character(1:7))
colnames(rankTaxon) <-
  c("Kingdom",
    "Phylum",
    "Class",
    "Order",
    "Family",
    "Genus",
    "Species")
rankTaxon <- cbind(id = tax$Feature.ID,
                   rankTaxon,
                   Confidence = tax$Confidence)
otuWithTax <- right_join(otu, rankTaxon, by = c("OTU.ID" = "id"))
totalCount <- rowSums(otuWithTax[, 2:ncol(otu)])
totalPercent <- totalCount / sum(totalCount) * 100
otuWithTax$totalCount <- totalCount
otuWithTax$totalPercent <- totalPercent
otuWithTax <- arrange(otuWithTax, desc(totalCount))
otuWithTax$OTU.IX<-paste0("OTU",1:nrow(otuWithTax))
otuWithTax<-otuWithTax[otuWithTax$totalCount > 1,]
write.csv(otuWithTax,
          "OTUByNaive_bayesSortByOneMore.csv",
          row.names = F)
# 10787 OTU with more than 1 sequences
sum(otuWithTax$totalCount > 1)
```