Bash scripts PRJNA307231

Paper: **https://pubmed.ncbi.nlm.nih.gov/30171206/**

Select samples

https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA307231

- Enter SRA experiments
- SRA Run Selector:
  - Assay type: WGS (156 samples): download metadata file: `WGS_metadata.txt`

Select 5 samples VIH pos, 5 samples VIH neg and store in csv file: `samples_metadata.csv`

```
conda install -c bioconda sra-tools parallel fastqc multiqc
```

Download fastq files

split files: stores forward and reverse reads in separate files:

```
mkdir fastq
cut -f 1 samples_metadata.csv | parallel --gnu "fastq-dump {} --split-
files --outdir fastq/"
cd fastq
ls *fastq
SRR6714072_1.fastq  SRR6714074_1.fastq  SRR6714076_1.fastq
SRR6714078_1.fastq  SRR6714088_1.fastq
SRR6714072_2.fastq  SRR6714074_2.fastq  SRR6714076_2.fastq
SRR6714078_2.fastq  SRR6714088_2.fastq
SRR6714073_1.fastq  SRR6714075_1.fastq  SRR6714077_1.fastq
SRR6714079_1.fastq  SRR6714098_1.fastq
SRR6714073_2.fastq  SRR6714075_2.fastq  SRR6714077_2.fastq
SRR6714079_2.fastq  SRR6714098_2.fastq
```

Quality and filter

Quality analysis (fastqc)

```
mkdir fastqc
ls fastq/*.fastq | parallel --gnu "fastqc {} -o fastqc/"
cd fastqc
multiqc .
```

Quality filter (fastp):

- q-: remove sequences with quality <=q20
- l-: minimum sequence length= 50 pb
- *-f: trim first 10 bp from each sequence*
- -c: enable base correction in overlapped regions (only for PE data), default is disabled

```
mkdir fastp
ls fastq/*.fastq  | sort | parallel --gnu --max-args=2 "fastp -i {1} -I
{2} -o fastp/filt_{1} -O fastp/filt_{2} -q 20 -l 50 -c -f 10 -j
fastp/{1/.}_fastp.json -h fastp/{1/.}_fastp.html"
```

https://opensource.com/article/18/5/gnu-parallel

Rename html and json files:

```
mmv \*_1_fastp.json \#1_fastp.json
mmv \*_1_fastp.html \#1_fastp.html
```

Quality analysis after filtering (fastqc)

```
mkdir filt_fastqc
ls fastp/*.fastq | parallel --gnu "fastqc {} -o filt_fastqc/"
cd filt_fastqc
multiqc .
```

GROOT analysis

GROOT scpit to analyse multiple samples simoultaneously: `groot_multsamples.sh`

```
#!/usr/bin/env bash

MYDIR=/home/erubio/Documentos/UOCMaster/bashScripts
seqlen=$1

MYDIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" &> /dev/null && pwd )"

mkdir groot_analysis
cd groot_analysis

groot get -d card
##generates a folder called card.90 in working directory with clustered
card database

groot index -m card.90 -i grootIndex$seqlen -w $seqlen -p 8
##Convert a set of clustered reference sequences to variation graphs and
then index them


ls ../fastp/*.fastq  | sort | parallel --gnu --max-args=2 "groot align -
i grootIndex$seqlen -f {1},{2} -p 8 -g {1/.}-groot-graphs > {1/.}.bam"
##generates bam file from samples and groot graph files

mmv \*_1\* \#1\#2 ##Rename bam files (Remove _1 ending from forward
sequence names)

ls *bam | parallel --gnu "samtools view -F 256 -h {} > {/.}.sam"
##Transform BAM file to SAM file (to execute python function).
##We have removed sequences with flag "not primary alignment"


ls *sam| parallel --gnu "python $MYDIR/../pythonScripts/groot_uniqseq.py
{}"
##Run python script on sam files
ls *-uniqseq.txt| parallel --gnu "grep "\S" {} > {/.}2.txt"
rm -f *-uniqseq.txt
##Remove empty lines
```

```
mmv \*-uniqseq2.txt \#1-uniqseq.sam ##Rename text files to sam files

ls *-uniqseq.sam | parallel --gnu "samtools view -S -b {} > {/.}.bam"
##Transform sam to bam files

##Generate reports (3 reports per sample):

ls *-uniqseq.bam | parallel --gnu "groot report -c 0 --bamFile {} >
{/.}-0report "
ls *-uniqseq.bam | parallel --gnu "groot report --bamFile {} >
{/.}-0.97report "
ls *-uniqseq.bam | parallel --gnu "groot report --bamFile {} --lowCov>
{/.}-lowCov-report "

echo "Report: This will report gene, read count, gene length, coverage
cigar"
```

Run from folder where the fastp folder is:

```
conda activate Groot
bash $MYDIR/groot_multsamples.sh 115
```

RGI analysis

RGI scpit to analyse multiple samples simoultaneously: `RGI_multsamples.sh`

```
#!/usr/bin/env bash
mkdir RGI_analysis
cd RGI_analysis

wget https://card.mcmaster.ca/latest/data ##generates data document
tar -xvf data ./card.json ##generates card.json in the current folder
rgi load --card_json card.json --local ##creates a folder called localdb

version=$(rgi database --version --local) ##obtain the card version we
just downloaded

##This commands will generate: card_annotation.log and
card_database_xx.fasta objects
rgi card_annotation -i card.json > card_annotation.log 2>&1
rgi load -i card.json --card_annotation card_database_v$version.fasta --
local

echo "Downloaded card database version v$version"
echo "Aligning forward and reverse FASTQ reads using Bowtie2 against
v$version CARD database"
```

```
ls ../fastp/*.fastq  | sort | parallel --gnu --max-args=2 -j 1 "rgi bwt
--read_one  {1} --read_two {2}  --aligner bowtie2 --output_file {1/.} --
threads 8 --local"
mmv \*_1\* \#1\#2 ##Rename files (Remove _1 ending from forward sequence
names)
```

> rgi bwt must be run 1 by 1 in parallel function (-j 1)

```
conda activate rgi
bash $MYDIR/RGI_multsamples.sh
```

ARIBA analysis

ARIBA scpit to analyse multiple samples simoultaneously: `ariba_multsamples.sh`

```
#!/usr/bin/env bash
mkdir ariba_analysis
cd ariba_analysis

ariba getref card out.card ##generates files out.card.fa out.card.log
out.card.tsv in current directory
ariba prepareref -f out.card.fa -m out.card.tsv ariba_db ##generates
folder ariba_db

ls ../fastp/*.fastq  | sort | parallel --gnu --max-args=2 "ariba run
ariba_db {1} {2} {1/.}_ariba_results"

prename 's/_1_/_/' *_1_*/ ##Change folder names (Substitute _1_ for _)

##Change report names to include sample name and move them from the
directories
for subdir in filt*; do
    subdir1=${subdir%_*}
    subdir2=${subdir1%_*}
    mv $subdir/report.tsv ${subdir2}_report.tsv; done
```

```
conda activate ariba2
bash $MYDIR/RGI_multsamples.sh
```