

# wind: wORKFLOW FOR PiRNAs AnD BEYONd

Computational workflow for the preprocessing of selected samples from of TCGA regarding  
Breast Cancer

Constantinos Yeles(Konstantinos Geles)

Fri Oct 16 2020

## Contents

The Data set . . . . .	1
i. Downloading the samples . . . . .	1
ii. BAM to FASTQ . . . . .	1
iii. Preprocessing of the samples . . . . .	2
Alignment and Quantification . . . . .	2
i. Transcript abundances with <b>Salmon</b> . . . . .	2
ii. Alignment and quantification of sequenced reads with STAR and Featurecounts . . . . .	3
iii. Alignment with STAR . . . . .	3
iv. R docker . . . . .	3
v. <b>FeatureCounts</b> . . . . .	4
R Session Info . . . . .	4
We work on : . . . . .	5

## The Data set

We will work with a dataset from TCGA which is a subset of 18 samples from TCGA-BRCA. In detail, we used 9 Primary Solid Tumors versus 9 Solid Tissue Normal matched samples. ## Data acquisition and preprocessing

### i. Downloading the samples

To obtain the data from TCGA we used a manifest with the selected samples and followed the instructions on GDC website

### ii. BAM to FASTQ

The acquired files were in BAM format but in order to do the whole workflow we transformed them to fastq format.

```

docker run --rm -ti -v $(pwd):/home/my_data congelos/sncrna_workflow

mkdir my_data/fastq_files
for file in my_data/bam_files/*.bam;
do regex="${file%*.bam}"; samp=`basename ${regex}`;
echo "Processing sample ${samp}";
samtools bam2fq -@ 6 $file > my_data/fastq_files/${samp}.fastq
echo "pigz sample ${samp}";
pigz --best "my_data/fastq_files/${samp}.fastq"
done

```

### iii. Preprocessing of the samples

We perform quality control(QC) on the fastq files to get basic information about the samples. We work with the **Fastqc** tool to perform QC.

```

mkdir -p my_data/Breast_Cancer_TCGA/qc_first

'fastqc' --threads 6 --outdir=my_data/Breast_Cancer_TCGA/qc_first/ my_data/myscratch/fastq_files/*.fastq.gz

for file in my_data/myscratch/fastq_files/*.fastq.gz;
do ./spar_prepare/srna_adapter_cut.sh $file 6;
done

mkdir my_data/Breast_Cancer_TCGA/qc_after

'fastqc' --threads 6 --outdir=my_data/Breast_Cancer_TCGA/qc_after/ my_data/myscratch/fastq_files/*.trim.fastq.gz

exit

```

## Alignment and Quantification

### i. Transcript abundances with Salmon

We will use a public docker image to run salmon

```

# run the docker
docker run --rm -it -v $(pwd):/home/my_data combinelab/salmon

# create the index
salmon index -t ncRNA_transcripts_100bp_RNA_Central_piRNAbank_hg38.fa -i genome_transc_human/ncRNA_Central

mkdir my_data/smallRNA-breast-cancer/GSE129076/quants/

# run the samples

#!/bin/bash

for fn in my_data/smallRNA-breast-cancer/Breast_Cancer_TCGA/fastq_files/*trimmed.fastq.gz;
do samp=`basename ${fn}`;
echo "Processing sample ${samp}";
salmon quant -i my_data/genome_

```

```
transc_human/ncRNA_Central_piRNAB_hg38_index -l A -r ${fn} --seqBias --gcBias --numBootstraps 100 -p 6
done

#save as bam files
for file in my_data/smallRNA-breast-cancer/Breast_Cancer_TCGA/quants/*.sam;
do
regex="${file%%.sam}";
echo "Processing sample ${regex}";
echo samtools view -O bam -o ${regex}.bam -@ 6 ${file};
done
exit
```

## ii. Alignment and quantification of sequenced reads with STAR and Featurecounts

We use the **STAR** aligner and then perform quantification with featureCounts from **Rsubread** package. With the docker image that contains STAR and **Samtools** we get sorted BAM files and use them for quantification / annotation for smallRNAs.

## iii. Alignment with STAR

```
# run docker
docker run --rm -ti -v "$PWD":/home/my_data congelos/sncrna_workflow

# index generation
STAR --runMode genomeGenerate --genomeDir my_data/human_data/GRCh38 --genomeFastaFiles my_data/human_data/GRCh38.fasta

mkdir my_data/smallRNA-breast-cancer/Breast_Cancer_TCGA/star_results

# alignment
for file in my_data/smallRNA-breast-cancer/Breast_Cancer_TCGA/fastq_files/*.trimmed.fastq.gz;
do samp=`basename ${file}`;
regex="${samp%%.trimmed.fastq.gz}";
echo "Processing sample ${samp} start: $(date)";
STAR --genomeDir my_data/genome_transc_human/human_data/GRCh38_2_7_4a --genomeLoad LoadAndKeep --readFilesIn $file;
done

exit
```

Next, we run a docker image which includes various R packages that will be used furthermore in the downstream analysis following featurecounts for the exploratory data analysis of piRNA data

## iv. R docker

```
docker run --rm -v $(pwd):/home/0 -p 8787:8787 -e PASSWORD=12345 -e USER=$UID congelos/rocker_tidyverse
```

From here on we work in R using a browser. we input `http://localhost:8787/` on browser, 0 for username and 12345 for password.

## v. FeatureCounts

```
library(Rsubread)
library(tidyverse)
list.BAM <- list.files(path = "Breast_Cancer_TCGA/star_results",
                      pattern = ".bam$",
                      recursive = TRUE,
                      full.names = T)

path_gtf <- "../genome_transc_human/ncRNA_transcripts_100bp_RNA_Central_piRNAbank_hg38.gtf"
todate <- format(Sys.time(), "%d_%b_%Y")

fc <- featureCounts(files = list.BAM,
                   annot.ext = path_gtf,
                   isGTFAnnotationFile = TRUE,
                   GTF.featureType = "exon",
                   GTF.attrType.extra = c("gene_type", "sRNA_id", "seq_RNA"),
                   nthreads = 8,
                   useMetaFeatures = TRUE,
                   allowMultiOverlap = TRUE,
                   minOverlap = 10,
                   largestOverlap = TRUE,
                   fraction = TRUE,
                   strandSpecific = 0,
                   verbose = TRUE,
                   reportReads = "BAM",
                   reportReadsPath = "Breast_Cancer_TCGA/star_results/")
fc %>% write_rds(str_glue("Breast_Cancer_TCGA/feature_counts_BRCA_TCGA_{todate}.rds"))
```

Next we will follow the workflow of `data_exploration_salmon_fc.RMD`

## R Session Info

```
sessionInfo()
R Under development (unstable) (2019-12-06 r77536)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Debian GNU/Linux 10 (buster)

Matrix products: default
BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/libopenblas-r0.3.5.so

locale:
 [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=C
 [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
 [9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

other attached packages:

```
[1] forcats_0.5.0      stringr_1.4.0      dplyr_0.8.4        purrr_0.3.3
[5] readr_1.3.1        tidyr_1.0.2        tibble_2.1.3        ggplot2_3.2.1
[9] tidyverse_1.3.0    Rsubread_2.1.2     BiocManager_1.30.10
```

loaded via a `namespace` (and not attached):

```
[1] Rcpp_1.0.3          cellranger_1.1.0 pillar_1.4.3        compiler_4.0.0
[5] dbplyr_1.4.2        tools_4.0.0        lubridate_1.7.4     jsonlite_1.6.1
[9] lifecycle_0.1.0     nlme_3.1-144       gtable_0.3.0        lattice_0.20-40
[13] pkgconfig_2.0.3     rlang_0.4.5        reprex_0.3.0        Matrix_1.2-18
[17] cli_2.0.2           DBI_1.1.0          rstudioapi_0.11     xfun_0.12
[21] haven_2.2.0         knitr_1.28         withr_2.1.2         xml2_1.2.2
[25] httr_1.4.1          fs_1.3.1           generics_0.0.2      vctrs_0.2.3
[29] hms_0.5.3           grid_4.0.0         tidymodels_1.0.0    glue_1.3.1
[33] R6_2.4.1            fansi_0.4.1        readxl_1.3.1        modelr_0.1.6
[37] magrittr_1.5        backports_1.1.5    scales_1.1.0        rvest_0.3.5
[41] assertthat_0.2.1    colorspace_1.4-1   stringi_1.4.6       lazyeval_0.2.2
[45] munsell_0.5.0       broom_0.5.5        crayon_1.3.4
```

We work on :

```
[root@localhost GSE124507_brain_project]# cat /etc/*-release
```

```
CentOS Linux release 7.8.2003 (Core)
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
```

```
[root@localhost GSE124507_brain_project]# docker version
```

```
Client: Docker Engine - Community
 Version:           19.03.8
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        afacb8b
 Built:             Wed Mar 11 01:27:04 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.8
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       afacb8b
  Built:            Wed Mar 11 01:25:42 2020
```

```
OS/Arch:      linux/amd64
Experimental:  false
containerd:
  Version:    1.2.13
  GitCommit:  7ad184331fa3e55e52b890ea95e65ba581ae3429
runc:
  Version:    1.0.0-rc10
  GitCommit:  dc9208a3303feef5b3839f4323d9beb36df0a9dd
docker-init:
  Version:    0.18.0
  GitCommit:  fec3683

[root@localhost GSE124507_brain_project]# git version
git version 1.8.3.1

[root@localhost GSE124507_brain_project]# pigz --version
pigz 2.3.4
```