wind: wORKFLOW FOR PiRNAs AnD BEYONd

Computational workflow for the preprocessing of selected samples from the E-MTAB-8115 dataset and some replicates for Testis samples

Constantinos Yeles (Konstantinos Geles)

Thu_Oct_29_2020

Contents

The Data set

We will work on a public dataset uploaded on ArrayExpress with id: E-MTAB-8115, that it has been used in the publication: **Molecular and Functional Characterization of the Somatic PIWIL1/piRNA Pathway in Colorectal Cancer Cells** and some other samples from testis tissue.

Data aqcuisition and preprocessing

i. Downloading the samples

We use a script to download the fastq samples with samtools-kit that it is included in the docker with the name download_SRA.sh

Using the **SRA selector** we download a file with the Accession List and rename the file to **GSE68246_samples.txt**

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

# run the script to download the SRA
./download_SRA.sh GSE124507_samples.txt 8
```

ii. 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 my_data/qc_first

'fastqc' --threads 6 --outdir=my_data/qc_first/ my_data/samples/*.fastq.gz

for file in my_data/samples/*.fastq.gz;
do
./spar_prepare/smrna_adapter_cut.sh $file 6;
done

mkdir my_data/samples/qc_after

'fastqc' --threads 6 --outdir=my_data/qc_after/ my_data/samples/*.trimmed.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_Centr

mkdir my_data/quants/
# run the samples

#!/bin/bash

for fn in my_data/testis_colo_pub_workflow/samples/*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 --g
done

#save as bam files
for file in my_data/smallRNA-breast-cancer/GSE68246/quants/*.sam;
do
```

```
regex="${file%%.sam}";
echo samtools view -O bam -o ${regex}.bam -@ 6 ${file};
done
exit
```

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 a docker images that contains STAR and **Samtools** we get sorted BAM files and use them for quantification / annotation for smallRNAs.

ii. Alignment with STAR

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

STAR --runMode genomeGenerate --genomeDir my_data/mouse_data/GRCh38 --genomeFastaFiles my_data/mouse_da

mkdir my_data/testis_colo_pub_workflow/star

for file in my_data/testis_colo_pub_workflow/samples/*.trimmed.fastq.gz;
do
samp=`basename ${file}`; regex="${samp%%.trimmed.fastq.gz}";
echo "Processing sample ${samp} start: $(date)";
STAR --genomeDir my_data/genome_transc_human/human_data/GRCh38_2_7_4a --genomeLoad LoadAndKeep --readFi
echo "end:$(date)";
done

exit
```

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

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 and 0 for username and 12345 for password.

iv. featureCounts

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

```r
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 = 10,
                    useMetaFeatures = TRUE,
                    allowMultiOverlap = TRUE,
                    minOverlap = 10,
                    largestOverlap = TRUE,
                    fraction = TRUE,
                    strandSpecific = 0,
                    verbose = TRUE,
                    reportReads = "BAM",
                    reportReadsPath = "../testis_colo_pub_workflow/star")
fc %>% write_rds(str_glue("../testis_colo_pub_workflow/feature_counts_testis_colo205_{todate}.rds"))
```

Next we will follow the workflow of data_exploration_salmon_fc ## R Session Info

```r
sessionInfo()
```

R version 4.0.0 (2020-04-24) Platform: x86_64-pc-linux-gnu (64-bit) Running under: Ubuntu 18.04.4 LTS

Matrix products: default BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.2.20.so

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

attached base packages: [1] grid splines stats4 parallel stats graphics grDevices utils datasets methods base

other attached packages: [1] VennDiagram_1.6.20 futile.logger_1.4.3 ggseqlogo_0.1
[4] DESeq2_1.29.7 SummarizedExperiment_1.19.6 DelayedArray_0.15.7
[7] matrixStats_0.56.0 RColorBrewer_1.1-2 pheatmap_1.0.12
[10] rafalib_1.0.0 NOISeq_2.33.3 Matrix_1.2-18
[13] Biobase_2.49.0 edgeR_3.31.4 limma_3.45.9
[16] tximport_1.17.3 plyranges_1.9.3 GenomicRanges_1.41.5
[19] GenomeInfoDb_1.25.8 IRanges_2.23.10 S4Vectors_0.27.12
[22] BiocGenerics_0.35.4 data.table_1.12.8 forcats_0.5.0
[25] stringr_1.4.0 dplyr_1.0.0 purrr_0.3.4
[28] readr_1.3.1 tidyr_1.1.0 tibble_3.0.1
[31] ggplot2_3.3.1 tidyverse_1.3.0 BiocManager_1.30.10

loaded via a namespace (and not attached): [1] colorspace_1.4-1 hwriter_1.3.2 ellipsis_0.3.1 XVector_0.29.3
[5] fs_1.4.1 rstudioapi_0.11 bit64_0.9-7 AnnotationDbi_1.51.3
[9] fansi_0.4.1 lubridate_1.7.8 xml2_1.3.2 R.methodsS3_1.8.0
[13] geneplotter_1.67.0 jsonlite_1.6.1 Rsamtools_2.5.3 broom_0.5.6
[17] annotate_1.67.0 dbplyr_1.4.4 png_0.1-7 R.oo_1.23.0
[21] compiler_4.0.0 httr_1.4.1 backports_1.1.7 assertthat_0.2.1
[25] cli_2.0.2 formatR_1.7 prettyunits_1.1.1 tools_4.0.0
[29] gtable_0.3.0 glue_1.4.1 GenomeInfoDbData_1.2.3 rappdirs_0.3.1

[33] ShortRead_1.47.2 Rcpp_1.0.4.6 cellranger_1.1.0 vctrs_0.3.1
[37] Biostrings_2.57.2 nlme_3.1-148 rtracklayer_1.49.4 rvest_0.3.5
[41] lifecycle_0.2.0 XML_3.99-0.3 zlibbioc_1.35.0 scales_1.1.1
[45] aroma.light_3.19.0 vroom_1.2.1 hms_0.5.3 lambda.r_1.2.4
[49] curl_4.3 memoise_1.1.0 biomaRt_2.45.2 latticeExtra_0.6-29
[53] stringi_1.4.6 RSQLite_2.2.0 genefilter_1.71.0 GenomicFeatures_1.41.2
[57] BiocParallel_1.23.2 rlang_0.4.6 pkgconfig_2.0.3 bitops_1.0-6
[61] lattice_0.20-41 GenomicAlignments_1.25.3 bit_1.1-15.2 tidyselect_1.1.0
[65] magrittr_1.5 R6_2.4.1 generics_0.0.2 DBI_1.1.0
[69] pillar_1.4.4 haven_2.3.1 withr_2.2.0 survival_3.1-12
[73] RCurl_1.98-1.2 EDASeq_2.23.2 modelr_0.1.8 crayon_1.3.4
[77] futile.options_1.0.1 utf8_1.1.4 BiocFileCache_1.13.0 jpeg_0.1-8.1
[81] progress_1.2.2 locfit_1.5-9.4 readxl_1.3.1 blob_1.2.1
[85] reprex_0.3.0 digest_0.6.25 xtable_1.8-4 R.utils_2.9.2
[89] openssl_1.4.1 munsell_0.5.0 askpass_1.1 ## 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"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"

CentOS Linux release 7.8.2003 (Core)
CentOS Linux release 7.8.2003 (Core)


[root@localhost GSE124507_brain_project]# docker version

Client: Docker Engine - Community
 Version:           19.03.13
 API version:       1.40
 Go version:        go1.13.15
 Git commit:        4484c46d9d
 Built:             Wed Sep 16 17:03:45 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.13
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.13.15
  Git commit:       4484c46d9d
  Built:            Wed Sep 16 17:02:21 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.3.7
  GitCommit:        8fba4e9a7d01810a393d5d25a3621dc101981175
 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
```

Venn diagram

common expressed piRNA between testis samples and COLO205

```
library(VennDiagram)

salmon_FC_groups <- read_tsv("../testis_colo_pub_workflow/ExpDatAnalysis_testis_colo205_hg38_22_Oct_2020

salmon_FC_groups %>%
  filter(gene_type == "piRNA") %>%
  filter(Testis_pool_salmon > 0 | Testis_pool_fc > 0)

salmon_FC_groups %>%
     filter(gene_type == "piRNA") %>%
     filter(COLO205_salmon > 0 | COLO205_fc > 0)

salmon_FC_groups %>%
     filter(gene_type == "piRNA") %>%
     filter(COLO205_salmon > 0 | COLO205_fc > 0,
            Testis_pool_salmon > 0 | Testis_pool_fc > 0)


grid.newpage()                                        # Move to new plotting page
draw.pairwise.venn(area1 = 6982,                          # Add name to each set
                 area2 = 240,
                 cross.area = 234,
                 category = c("testis", "COLO205"),
                 fill = c("red", "blue"),
                 lty = "blank")
```