wind: wORKFLOW FOR PiRNAs AnD BEYONd

Computational workflow for Differential Expression analysis of GSE68246 public, regarding Human Breast MCF-7 Cell Line with Cancer Stem Cell Properties

Constantinos Yeles (Konstantinos Geles)

Thu Oct 29 2020

Contents

Introduction

Following the data_exploration_salmon_fc worklow in most cases we want to perform differential expression (DE) analysis. We follow instructions from various packages utilized for DE with the objects resulted from the previous workflow.

Load libraries

```r
suppressPackageStartupMessages({
  library('tidyverse')
  library('edgeR')
  library('DESeq2')
})
```

Add date of the analysis

We use it as an identifier for the folder and generally the analysis

```r
todate <- format(Sys.time(), "%d_%b_%Y")
```

Make the directory for the results of the DE analysis

```r
my_basename <- "GSE68246"
my_exp <- "Breast_cancer"
my_exp_sal <- "salmon"
my_exp_fc <- "featureCounts"
dat_path <-  str_glue("{my_basename}/DEA_{my_exp}_{todate}/{c(my_exp_sal,my_exp_fc)}") %>%
  as.list %>%
  set_names("salmon","featureCounts")
dat_path %>% map(~dir.create(str_glue("./{.x}"), recursive = TRUE))
```

1. Import the normalized files

```r
list_norm_dgls <- list.files(path = my_basename, pattern = "list_norm_dgls.+rds",
  recursive = TRUE, full.names = TRUE)

# load salmon normalized files
salmon_norm <- list_norm_dgls %>%
  unlist %>%
  str_detect("salmon") %>%
  list_norm_dgls[.] %>%
  read_rds()

# load featurecounts normalized files
fc_norm <- list_norm_dgls %>%
  unlist %>%
  str_detect("featureCounts") %>%
  list_norm_dgls[.] %>%
  read_rds()
```

2. Extract normalized objects

We will work with TMM normalization and TMM voom with quality weights

```
salmon_edgR_TMM <- salmon_norm[["TMM"]]
salmon_vm_QW_TMM <- salmon_norm[["voomQW_TMM"]]
fc_edgR_TMM <- fc_norm[["TMM"]]
fc_vm_QW_TMM <- fc_norm[["voomQW_TMM"]]
```

3. Create the design matrix

If we load the voom object we can extract the design matrix otherwise we can create it again from the dgl object.

```
#1 voom object
design <- salmon_vm_QW_TMM$design
#or dgl object
targets <- salmon_edgR_TMM$samples
design <- model.matrix(~0 + targets$group , data = targets)
colnames(design) <- colnames(design) %>%
  str_remove(".+\\$group")
```

4. EdgeR

Perform the analysis with edgeR TMM normalization for both salmon and featurecounts

```
# design ----
con_mat <- makeContrasts(
  Spheroid_Mono = MCF_7_Spheroid - MCF_7_Monolayer,
  levels = design)

## salmon ----
salmon_edgR_TMM <- estimateDisp(salmon_edgR_TMM, design = design, robust=TRUE)
salmon_edgR_TMM <- glmQLFit(salmon_edgR_TMM, design, robust = TRUE)

DE_salmon_edgR <- con_mat %>%
  colnames() %>%
  set_names() %>%
 map(~glmQLFTest(salmon_edgR_TMM, contrast = con_mat[,.x]) %>%
  topTags(n = nrow(.), adjust.method = "BH", sort.by = "PValue", p.value = 1) %>%
   .$table %>%
  as_tibble(rownames = "smallRNA") %>%
  write_tsv(str_glue("{dat_path[['salmon']]}/DE_salmon_edgR_TMM_{.x}.txt"))
 )

hist(DE_salmon_edgR[[1]]$PValue, breaks = 0:20/20,
     col = "grey50", border = "white")

salmon_edgeR_TMM_p <- DE_salmon_edgR[[1]] %>%
  mutate(salmon_edgeR = if_else(
    FDR >= 0.05, 0, if_else(
      logFC > 0, 1, -1
    )
  )) %>%
  select(smallRNA , salmon_edgeR )
```

```
## featureCounts ----
fc_edgR_TMM <- estimateDisp(fc_edgR_TMM, design = design, robust=TRUE)
fc_edgR_TMM <- glmQLFit(fc_edgR_TMM, design, robust = TRUE)

DE_FC_edgR <- con_mat %>% colnames() %>% set_names() %>%
 map(~glmQLFTest(fc_edgR_TMM, contrast = con_mat[,.x]) %>%
  topTags(n = nrow(.), adjust.method = "BH", sort.by = "PValue", p.value = 1) %>%
   .$table %>%
  as_tibble(rownames = "smallRNA") %>%
  write_tsv(str_glue("{dat_path[['featureCounts']]}/DE_fc_edgR_TMM_{.x}.txt"))
 )

hist(DE_FC_edgR[[1]]$PValue, breaks = 0:20/20,
     col = "grey50", border = "white")

fc_edgeR_TMM_p <- DE_FC_edgR[[1]] %>%
  mutate(fc_edgeR = if_else(
    FDR >= 0.05, 0, if_else(
      logFC > 0, 1, -1
    )
  )) %>%
  select(smallRNA , fc_edgeR )

# venn diagram for salmon/fc edgeR -----
results <-  salmon_edgeR_TMM_p %>%
  inner_join(fc_edgeR_TMM_p) %>%  select(-smallRNA)

pdf(str_glue("{dat_path}/venn_diagram_DE_salmon_fC_edgeR.pdf"))
vennDiagram(results,
    include=c("up", "down"),
    counts.col=c("red", "blue"),
    circle.col = c("red", "blue", "green3"))
dev.off()
```

5. Limma

```
# design ----
### same as before
## salmon ----
salmon_vm_QW_TMM <- lmFit(salmon_vm_QW_TMM, design = design)
salmon_vm_QW_TMM <- contrasts.fit(salmon_vm_QW_TMM, con_mat)
salmon_vm_QW_TMM <- eBayes(salmon_vm_QW_TMM, robust = TRUE)

salmon_DES <- con_mat %>% colnames() %>% set_names() %>%
 map(~salmon_vm_QW_TMM %>% topTable(., coef = .x,
                            confint = TRUE,
                            number = nrow(.),
                            adjust.method = "fdr",
                            sort.by = "p") %>%
  as_tibble(rownames = "smallRNA") %>%
  rename_at(vars(logFC:B), list(~str_c(.,"_",!!quo(.x)))) %>%
```

```
  write_tsv(str_glue("{dat_path[['salmon']]}/DE_salmon_vm_QW_TMM_{.x}.txt"))
 )

hist(salmon_DES[[1]] %>% select(starts_with("P.Value")) %>% deframe(),
     breaks = 0:20/20,
     col = "grey50", border = "white")

salmon_vm_QW_TMM_p <- salmon_DES[[1]] %>%
  mutate(salmon_voomQ = if_else(
    adj.P.Val_Spheroid_Mono >= 0.05, 0, if_else(
      logFC_Spheroid_Mono > 0, 1, -1
    )
  )) %>%
  select(smallRNA , salmon_voomQ )

## featureCounts ----
fc_vm_QW_TMM <- lmFit(fc_vm_QW_TMM, design = design)
fc_vm_QW_TMM <- contrasts.fit(fc_vm_QW_TMM, con_mat)
fc_vm_QW_TMM <- eBayes(fc_vm_QW_TMM, robust = TRUE)

fc_DES <- con_mat %>% colnames() %>% set_names() %>%
 map(~fc_vm_QW_TMM %>% topTable(., coef = .x,
                                confint = TRUE,
                                number = nrow(.),
                                adjust.method = "fdr",
                                sort.by = "p") %>%
  as_tibble(rownames = "smallRNA") %>%
  rename_at(vars(logFC:B), list(~str_c(.,"_",!!quo(.x)))) %>%
  write_tsv(str_glue("{dat_path[['featureCounts']]}/DE_fc_vm_QW_TMM_{.x}.txt"))
 )

hist(fc_DES[[1]] %>% select(starts_with("P.Value")) %>% deframe(),
     breaks = 0:20/20,
     col = "grey50", border = "white")

fc_vm_QW_TMM_p <- fc_DES[[1]] %>%
  mutate(fc_voomQ = if_else(
    adj.P.Val_Spheroid_Mono >= 0.05, 0, if_else(
      logFC_Spheroid_Mono > 0, 1, -1
    )
  )) %>%
  select(smallRNA , fc_voomQ )

# venn diagram for salmon/fc limma -----
nc_RNA_categories <- plyranges::read_gff2("../genome_transc_human/ncRNA_transcripts_100bp_RNA_Central_p
  as_tibble() %>%
  select(gene_id, gene_type) %>%
  distinct(gene_id, .keep_all = TRUE)

results <-  salmon_vm_QW_TMM_p %>%
  inner_join(fc_vm_QW_TMM_p) %>%  select(-smallRNA)

pdf(str_glue("{dat_path}/venn_diagram_DE_salmon_fC_limma_Spheroid_Mono.pdf"))
```

```
vennDiagram(results,
    include=c("up", "down"),
    counts.col=c("red", "blue"),
    circle.col = c("red", "blue", "green3"))
dev.off()
# join both results ----
identical(fc_DES %>% names(), salmon_DES %>% names)

map2(fc_DES, salmon_DES, ~.x %>%
       select_at(vars(starts_with(c("smallRNA","logFC",
                                    "P.Value","adj.P.Val")))) %>%
       rename_at(vars(!matches("smallRNA")), list(~str_c(.,"_FC"))) %>%
       full_join(.y%>%
       select_at(vars(starts_with(c("smallRNA","logFC",
                                    "P.Value","adj.P.Val")))) %>%
       rename_at(vars(!matches("smallRNA")), list(~str_c(.,"_salmon"))))
     ) %>%
  purrr::reduce(full_join) %>%
  inner_join(nc_RNA_categories, by = c("smallRNA" = "gene_id")) %>%
  write_tsv(str_glue("{str_remove(dat_path[1],'/salmon|/featureCounts')}/all_comparisons_voom_QW_TMM_sa
```

6. Compare the DE results with public results

```
## import the public results -----
public_res <- "Public_results_GSE68246.csv" %>%
  read_tsv() %>%
  mutate(logFC = gtools::foldchange2logratio(FC)) %>%
  select(smallRNA, FC, logFC, sequence) %>%
  rename_all(.funs = ~ .x %>% str_c("_public_res"))

## import the sequences from the GTF file -----
smallRNAs_gtf <- list.files(pattern = "_100bp_RNA_Central_piRNAbank_hg38.gtf") %>%
  plyranges::read_gff2() %>%
  as_tibble() %>%
  select(gene_id,  seq_RNA,gene_type) %>%
  distinct(gene_id, .keep_all = TRUE) %>%
  dplyr::rename("smallRNA" = gene_id)

## import the complete DE table with fold changes ----
all_comp <- read_tsv(str_glue("{str_remove(dat_path[1],'/salmon|/featureCounts')}/all_comparisons_voom_

## join all comparisons with smallRNAs_gtf ----
public_and_all_comp <- all_comp %>%
  left_join(smallRNAs_gtf) %>%
  full_join(public_res, by = c("seq_RNA" = "sequence_public_res")) %>%
  write_tsv(str_glue("{str_remove(dat_path[1],'/salmon|/featureCounts')}/public_and_all_comp_voom_QW_TM
```

7. Find predicted targets

```
suppressPackageStartupMessages(library(plyranges))
# load gtf of smallRNAs
smallRNAs_gtf  <- smallRNAs_gtf %>%
  filter(gene_type == "piRNA")

#load targets
targets_all <- read_tsv("piRNA_predicted_Targets.txt")

# targets DE union
targets_DEs_keep <- all_comp %>%
  filter(gene_type == "piRNA") %>%
  inner_join(targets_all) %>%
  write_tsv(str_glue("{str_remove(dat_path[1],'/salmon|/featureCounts')}/all_comp_voom_QW_DE_targets_pr
```

8. Make a heatmap of differential expressed piRNAs

i. Load the libraries

```
library(wesanderson)
library(ComplexHeatmap)
library(circlize)
```

ii. Load data

```
# load the piRNAs log fold changes
piRNAs_DE <- list.files(pattern = "all_comparisons_voom_QW_TMM") %>%
  read_tsv() %>%
  filter(gene_type == "piRNA",
         across(.cols = contains("adj.P.Val"),
                .fns = ~ .x< 0.05)) %>%
  dplyr::select(smallRNA, contains("logFC")) %>%
  write_tsv(str_glue("{str_remove(dat_path[1],'/salmon|/featureCounts')}/all_comp_voom_QW_DE_piRNA_{toda


# load the piRNA expression matrix----
## featurecounts
fc_list <- "list_norm_dgls_featureCounts.rds" %>%
  read_rds() %>%
  .[["TMM"]]

fc_cpm <- fc_list  %>%
  cpm(log =TRUE) %>%
  .[rownames(.) %in% piRNAs_DE$smallRNA,]

# make the matrices for the heatmap -----
FC_mat_1 <- fc_cpm %>%
```

```
  t() %>% scale() %>% t()
FC_mat_1 %>% dim()
FC_mat_1 %>% head()
hist(FC_mat_1)

# logFCS
lfc_piRNAs_DE <- piRNAs_DE %>%
  column_to_rownames("smallRNA") %>%
  as.matrix()

lfc_piRNAs_DE %>% dim()
lfc_piRNAs_DE %>% head()
hist(lfc_piRNAs_DE)

lfc_piRNAs_DE <- lfc_piRNAs_DE[rownames(FC_mat_1) ,]

colnames(FC_mat_1) <- colnames(FC_mat_1) %>% str_remove("_GR_")

# add the Annotation ----
#expression
ha_1 <- HeatmapAnnotation(Group = fc_list$samples$group,
                          annotation_name_side = "left",
                          col = list(Group = fc_list$colours %>%
                                       set_names(fc_list$samples$group)))

# lFCS
ha_1_LFCs <- HeatmapAnnotation(Method = c("FeatureCounts", "salmon"),
                               col = list(Method = wes_palettes$Moonrise1[c(2,3)] %>%
                          set_names("FeatureCounts", "salmon")))

## Colours of heatmap -----
#expression
f_1 <- colorRamp2(c(round(quantile(FC_mat_1, probs = 0.25)),
                    median(FC_mat_1),
                    round(quantile(FC_mat_1, probs = 0.75))),
                  c("blue", "black", "yellow"))

# lFCS
f_1_LFCs <- colorRamp2(c(-2,
                    0,
                    2,
                    4),
                    c("forestgreen", "black", "red", "red4"))

## Heatmaps ------
ht_1 <- Heatmap(matrix = FC_mat_1, #data
        top_annotation = ha_1, #annot
        col = f_1, #colors data
        show_row_dend = TRUE,
        show_row_names = FALSE,
        show_column_names = FALSE,
        name = "z-score equivalent expression",
        clustering_distance_columns = "spearman",
```

```
        clustering_method_columns =  "ward.D2",
        clustering_method_rows = "ward.D2",
        clustering_distance_rows = "spearman",
        row_dend_reorder = TRUE
)
rownames(lfc_piRNAs_DE) <- lfc_piRNAs_DE %>%
  rownames() %>%
  str_remove("_GR_.+")
ht_1_lFCs <- Heatmap(matrix = lfc_piRNAs_DE, #data
        top_annotation = ha_1_LFCs, #annot
        col = f_1_LFCs, #colors data
        show_row_dend = FALSE,
        show_row_names = TRUE,
        show_column_names = FALSE,
        name = "Log Fold Change",
        clustering_distance_columns = "spearman",
        clustering_method_columns =  "ward.D2",
        clustering_method_rows = "ward.D2",
        clustering_distance_rows = "spearman",
        row_dend_reorder = TRUE
)

draw(ht_1+ht_1_lFCs,column_title = str_glue("Heatmap of {nrow(FC_mat_1)} DE piRNAs"),
     merge_legend = TRUE)

tiff("GSE68246_spheroid_vs_mono_heatmap_FC.tiff",
     compression = "none", height = 10, width = 14,  units = 'in', res = 300)
draw(ht_1+ht_1_lFCs,
     column_title = str_glue("Heatmap of {nrow(FC_mat_1)} DE piRNAs"),
     merge_legend = TRUE)
dev.off()
```