Analysis of A673 EWS-FLI1 RNAseq Timecourse

Christopher Hughes

Abstract

This document details analysis of RNAseq data acquired to examine the impact of EWS-FLI1 expression in A673 EwS cells grown in 2D adherent culture conditions. The raw fastq data has already been processed and aligned, so here we will work with the outputs for quantification.

Data processing

Read in the sample annotation details and make the txdb that we can use later on for annotation gene identifiers.

```
##############################################################################################
#read a table containing the sample information for our RNAseq runs
samples = read_tsv(paste(baseRepository, '/sequencing20210421_a673TimecourseRnaSeqOutput/
samples
```

```
# A tibble: 24 x 6
   sampleName cell          treatment barcode experiment batch
   <chr>      <chr>         <chr>     <chr>   <chr>      <chr>
 1 day0       a673shEwsFli1 none      ATCACG  PX1955     setA
 2 day0       a673shEwsFli1 none      ATCACG  PX1956     setB
 3 day0       a673shEwsFli1 none      ATCACG  PX1957     setC
 4 day7       a673shEwsFli1 none      CGATGT  PX1955     setA
 5 day7       a673shEwsFli1 none      CGATGT  PX1956     setB
 6 day7       a673shEwsFli1 none      CGATGT  PX1957     setC
 7 day9       a673shEwsFli1 none      TTAGGC  PX1955     setA
 8 day9       a673shEwsFli1 none      TTAGGC  PX1956     setB
 9 day9       a673shEwsFli1 none      TTAGGC  PX1957     setC
10 day10      a673shEwsFli1 none      TGACCA  PX1955     setA
# ... with 14 more rows
```

```
#use the sample info to build a file list
files = file.path(baseRepository, 'sequencing20210421_a673TimecourseRnaSeqOutput', paste(
all(file.exists(files))
```

```
[1] TRUE
```

For Salmon analysis, I am generally following the documentation found here and here. When we import our data, we want a table that allows us to link gene and transcript identifiers. For this we use the GTF associated with our database files that we used during the alignment process.

```
##########################################################################################################
#build the txdb from the gtf file
myTxdb = makeTxDbFromGFF('D:/databases/projectEwsDlg2/baseGenomeFiles/genome.gtf')
k = keys(myTxdb, keytype = 'TXNAME')
tx2gene = AnnotationDbi::select(myTxdb, k, 'GENEID', 'TXNAME')
head(tx2gene)
```

```
             TXNAME            GENEID
1 ENST00000456328.2 ENSG00000223972.5
2 ENST00000450305.2 ENSG00000223972.5
3 ENST00000473358.1 ENSG00000243485.5
4 ENST00000469289.1 ENSG00000243485.5
5 ENST00000607096.1 ENSG00000284332.1
6 ENST00000606857.1 ENSG00000268020.3
```

Read in the Salmon data.

```
##########################################################################################################
#read the salmon data
txi = tximport(files,
               type = 'salmon',
               tx2gene = tx2gene)
names(txi)
```

```
[1] "abundance"           "counts"              "length"
[4] "countsFromAbundance"
```

```
head(txi$counts)
```

```
                      [,1]     [,2]     [,3]    [,4]    [,5]    [,6]    [,7]
ENSG00000000003.15  212.236  159.664  152.819 210.119 133.415 147.998 135.813
ENSG00000000005.6     0.000    2.000    0.000   2.000   2.000   3.000   0.000
ENSG00000000419.14  166.581  162.269  150.266 144.745 171.800 184.334 183.982
ENSG00000000457.14  312.000  257.000  225.000 457.001 467.001 509.000 361.256
ENSG00000000460.17 1605.776 1327.103 1402.854 768.626 785.761 846.681 426.045
ENSG00000000938.13    2.000    3.000    0.000   0.000   2.000   1.000   3.000
                      [,8]    [,9]   [,10]   [,11]   [,12]   [,13]   [,14]
ENSG00000000003.15 170.763 153.068  90.346  84.724  87.477 143.239 179.608
ENSG00000000005.6    0.000   1.000   2.000   0.000   0.000   0.000   0.000
ENSG00000000419.14 171.550 117.608  58.891  76.054  91.916 190.093 218.760
```

```
ENSG00000000457.14 415.000 335.030 265.350 333.081 336.999 429.328 389.999
ENSG00000000460.17 540.045 479.675 575.728 663.027 641.836 706.702 661.141
ENSG00000000938.13   4.000   1.000   1.000   1.000   0.000   1.000   2.000
                    [,15]    [,16]    [,17]    [,18]   [,19]   [,20]   [,21]
ENSG00000000003.15 152.974  146.293  190.925  144.732 110.248  98.482  72.622
ENSG00000000005.6    0.000    0.000    0.000    0.000   1.000   0.000   0.000
ENSG00000000419.14 141.241  154.386  193.659  120.473 420.225 331.625 298.531
ENSG00000000457.14 365.999  339.065  394.033  326.066 555.999 540.000 489.000
ENSG00000000460.17 669.095 1443.887 1766.727 1346.451 820.397 810.651 828.388
ENSG00000000938.13   1.000    2.000    1.000    0.000   4.000   1.000   2.000
                     [,22]    [,23]    [,24]
ENSG00000000003.15  170.781  269.282  206.517
ENSG00000000005.6     2.000    4.000    1.000
ENSG00000000419.14  145.677  144.655  134.878
ENSG00000000457.14  295.001  394.000  262.999
ENSG00000000460.17 1277.660 1714.595 1300.942
ENSG00000000938.13    1.000    0.000    1.000
```

```
##############################################################################################
#perform the deseq analysis
ddsTxi = DESeqDataSetFromTximport(txi,
                                  colData = samples,
                                  design = ~ sampleName)
dds = DESeq(ddsTxi)
keep = rowSums(counts(dds)) >= 10
dds = dds[keep,]
```

Extract the DESeq data and plot for the different comparisons of interest.

```
##############################################################################################
#parse the deseq data - change the first two values to get different comparisons
daysToCompare = c('day0','day9','day10','day11','day14','day17','day22')
for (i in 1:length(daysToCompare)){
  datasetFirst = daysToCompare[i]
  datasetSecond = 'day7'
  res = results(dds, contrast = c('sampleName', datasetFirst, datasetSecond))
  ens.str = substr(rownames(res), 1, 15)
  res$symbol = mapIds(org.Hs.eg.db,
                      keys=ens.str,
                      column="SYMBOL",
                      keytype="ENSEMBL",
                      multiVals="first")
  resOrdered = res[order(res$pvalue),]

  #save the data
```

```
  saveRDS(as.data.frame(resOrdered),
          paste(baseRepository, '/sequencing20210421_a673TimecourseRnaSeqOutput/dataset_d
  write.csv(as.data.frame(resOrdered),
            file = paste(baseRepository, '/sequencing20210421_a673TimecourseRnaSeqOutput/


  ##############################################################################################
  #assign colors based on fold change and p-values
  rnaExp = as.data.frame(resOrdered)
  rnaExp$logPValue = -log10(rnaExp$padj)
  rnaExp$logPValueScaled = ifelse(rnaExp$logPValue > 300, 300, rnaExp$logPValue)
  rnaExp$pColors = ifelse(rnaExp$padj <= 0.001 & rnaExp$log2FoldChange >= 1, brewer.pal(3
                          ifelse(rnaExp$padj <= 0.001 & rnaExp$log2FoldChange <= -1, brew

  #assign text labels to specific genes of interest
  goi = c('DLG2','LOX','PRKCB')
  rnaExp$pText = ifelse(rnaExp$symbol %in% goi, rnaExp$symbol, '')

  #create the plot and save it
  ggplot(rnaExp, aes(log2FoldChange, logPValueScaled)) +
    geom_point(size = 1, color = rnaExp$pColors, alpha = 0.75) +
    labs(x = paste('log2(',datasetFirst,' - ',datasetSecond,')', sep = ''), y = '-log10(A
    geom_text_repel(label = rnaExp$pText, nudge_x = -3, nudge_y = -150, max.overlaps = 15
    scale_x_continuous(limits = c(-10,10), breaks = seq(-10,10,2)) +
    scale_y_continuous(limits = c(0,300), breaks = seq(0,500,50)) +
    geom_vline(xintercept = c(-1,1), linetype = 'dashed') +
    geom_hline(yintercept = -log10(0.001), linetype = 'dashed') +
    theme_classic()
  ggsave(paste(baseRepository, '/sequencing20210421_a673TimecourseRnaSeqOutput/scatter_de
         height = 2, width = 2, useDingbats = FALSE)
}
```