Post-processing QC for Heaney et al. 2021

Sanjeev V Namjoshi (snamjoshi87@utexas.edu)

February 23rd, 2017 (revised June 15th, 2019)

Here we perform some exploratory data analysis in order to confirm the quality of our data. This includes PCA, hierarchical clustering between the different groups, and correlation of counts among the replicates.

Load packages

```
library(DESeq2)
library(magrittr)
library(ggplot2)
library(gridExtra)
library(pheatmap)
library(RColorBrewer)
```

Load data and process

The count data from the different sequencing runs are loaded and combined together into one unit. Next, a metatable is created for easier downstream processing. Then, technical replicates are collapsed by DESeq2 to create the final dataset used for the QC analysis. We apply the regularized log transformation to the data so that it is more Gaussian, a requirement for the QC methods.

```
### Load count data
countData1 <- read.csv("countData_july.csv", header = TRUE)
rownames(countData1) <- countData1[ ,1]
countData1[ ,1] <- NULL

countData2 <- read.csv("countData_aug.csv", header = TRUE)
countData2[ ,1] <- NULL
names(countData2) <- gsub("_T1", "_T3", names(countData2))
names(countData2) <- gsub("_T2", "_T4", names(countData2))

countData <- cbind(countData1, countData2) %>% as.data.frame()

### Create metatable
metaTable <- data.frame(LibraryName = names(countData),
                        technical = c(rep(1:2, 21), rep(3:4, 21)))

metaTable$LibraryName <- as.factor(metaTable$LibraryName)
metaTable$technical <- as.factor(metaTable$technical)
metaTable$Counts <- paste(metaTable$LibraryName, "count", sep = ".")
```

```
rownames(metaTable) <- metaTable[ ,1]   # Set first column to row names
metaTable[ ,1] <- NULL

### Collapse technical replicates
dd <- DESeqDataSetFromMatrix(countData = countData,
                             colData = metaTable,
                             design = ~ technical)

dd$technical <- gsub("_T1|_T2|_T3|_T4", "", row.names(metaTable))
ddRep <- collapseReplicates(dd, groupby = dd$technical)
ddRep$technical <- as.factor(ddRep$technical)

### rlog transformation of data
rd <- rlog(ddRep)
ddRep <- estimateSizeFactors(ddRep)
rdCounts <- assay(rd) %>% as.data.frame()
```

Count correlations (figure not rendered, see FigureS2.Rmd)

A simple way to visualize the precision of our independent biological replicates is simply by graphing their counts against each other. We expect a high correlation between all different groups. The data shows a high amount of correlation between log-transformed counts among the different groups.

```
### Saline WT scatterplots
p2 <- ggplot(rdCounts, aes(SAL_WT1, SAL_WT2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p3 <- ggplot(rdCounts, aes(SAL_WT2, SAL_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p4 <- ggplot(rdCounts, aes(SAL_WT1, SAL_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Saline KO scatterplots
p5 <- ggplot(rdCounts, aes(SAL_KO1, SAL_KO2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p6 <- ggplot(rdCounts, aes(SAL_KO2, SAL_KO3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p7 <- ggplot(rdCounts, aes(SAL_KO1, SAL_KO3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Ro WT scatteplots
p8 <- ggplot(rdCounts, aes(RO_WT1, RO_WT2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()
```

```
p9 <- ggplot(rdCounts, aes(RO_WT2, RO_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p10 <- ggplot(rdCounts, aes(RO_WT1, RO_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Ro KO histogram
p11 <- ggplot(rdCounts, aes(RO_KO2, RO_KO2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Ro Rap WT scatterplots
p12 <- ggplot(rdCounts, aes(RO_RAP_WT1, RO_RAP_WT2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p13 <- ggplot(rdCounts, aes(RO_RAP_WT2, RO_RAP_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p14 <- ggplot(rdCounts, aes(RO_RAP_WT1, RO_RAP_WT3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Ro Rap KO scatterplots
p15 <- ggplot(rdCounts, aes(RO_RAP_KO1, RO_RAP_KO2)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p16 <- ggplot(rdCounts, aes(RO_RAP_KO2, RO_RAP_KO3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

p17 <- ggplot(rdCounts, aes(RO_RAP_KO1, RO_RAP_KO3)) +
  geom_point(size = 0.5, alpha = 0.3) +
  theme_bw()

### Fully arranged (766 x 673)
grid.arrange(p2, p3, p4, p5,
             p6, p7, p8, p9,
             p10, p11, p12, p13,
             p14, p15, p16, p17,
             ncol = 4,
             nrow = 4)
```

Principal component analysis (PCA) (figure not rendered, see FigureS2.Rmd)

Principal component analysis is another means of visualizing the similarity in count measurements between the different groups. The adjusted axes allow us to see whether or not the different groups cluster together. The data shows that each replicate for WT and KO cluster together due to similarity in variance. The only

notable exception is the saline KO replicate 2 which is slightly off compared to the others. Similarly, the Ro KO replicate 2 is also quite far off from the others. This is not completely unexpected however, since we did note a poor alignment rate and unusual QC from the FastQC files.

```
# Arrange variables for plotting with ggplot2
rdNoTotal <- rd[ ,rd$technical %in% colData(rd)[3:18, 1]]
pcaDatNoTotal <- plotPCA(rdNoTotal, intgroup = "technical", returnData = TRUE)
percentVar <- round(100 * attr(pcaDatNoTotal, "percentVar"))

ggplot(pcaDatNoTotal, aes(PC1, PC2, color = technical, label = name)) +
  geom_point(size = 3) +
  geom_text(aes(label = name), size = 3, hjust = 0.3, vjust = -0.5) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  geom_hline(aes(yintercept = 0)) +
  geom_vline(aes(xintercept = 0)) +
  scale_color_manual(values = c(rep("darkolivegreen3", 1), rep("cadetblue3", 3), rep("dodgerblue3", 3),
  theme_bw() +
  theme(legend.position = "none")
```

Euclidean distance matrix (figure not rendered, see FigureS2.Rmd)

Finally, we used a clustered heatmap to visualize the similarity between the different samples. Overall, many groups clustered together though some replicates, particularly the Saline KO replicate 2, clustered better with other libraries than it did its own replicates. It is contained in a branch on its own which is a parent branch to two other libraries Ro+Rapa KO replicate 2 and Saline KO replicate 3 which are also off on their own. In general, since the KO libraries are constructed from null-antibody tissue we expect their to be some level of background binding that will be inconsistant among the different samples. Since the wild-type material clusters together properly, we have kept all libraries in going forward.

```
sampleDists <- dist(t(assay(rdNoTotal)))

sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- colnames(rdNoTotal)
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```

Session info:

```
## R version 3.3.2 (2016-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.1 LTS
##
## 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=en_US.UTF-8
```

```
##  [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] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] RColorBrewer_1.1-2         pheatmap_1.0.8
##  [3] gridExtra_2.2.1            ggplot2_2.2.0
##  [5] magrittr_1.5               DESeq2_1.14.0
##  [7] SummarizedExperiment_1.4.0 Biobase_2.34.0
##  [9] GenomicRanges_1.26.1       GenomeInfoDb_1.10.1
## [11] IRanges_2.8.1              S4Vectors_0.12.0
## [13] BiocGenerics_0.20.0
##
## loaded via a namespace (and not attached):
##  [1] genefilter_1.56.0    locfit_1.5-9.1       splines_3.3.2
##  [4] lattice_0.20-34      colorspace_1.3-1     htmltools_0.3.5
##  [7] yaml_2.1.14          survival_2.40-1      XML_3.98-1.5
## [10] foreign_0.8-67       DBI_0.5-1            BiocParallel_1.8.1
## [13] plyr_1.8.4           stringr_1.1.0        zlibbioc_1.20.0
## [16] munsell_0.4.3        gtable_0.2.0         evaluate_0.10
## [19] memoise_1.0.0        labeling_0.3         latticeExtra_0.6-28
## [22] knitr_1.15.1         geneplotter_1.52.0   AnnotationDbi_1.36.0
## [25] htmlTable_1.7        Rcpp_0.12.8          acepack_1.4.1
## [28] xtable_1.8-2         scales_0.4.1         backports_1.0.4
## [31] Hmisc_4.0-0          annotate_1.52.0      XVector_0.14.0
## [34] digest_0.6.10        stringi_1.1.2        grid_3.3.2
## [37] rprojroot_1.1        tools_3.3.2          bitops_1.0-6
## [40] lazyeval_0.2.0       RCurl_1.95-4.8       tibble_1.2
## [43] RSQLite_1.1          Formula_1.2-1        cluster_2.0.5
## [46] Matrix_1.2-7.1       data.table_1.9.8     assertthat_0.1
## [49] rmarkdown_1.2        rpart_4.1-10         nnet_7.3-12
```