PAPER TITLE TO BE DEFINED (in common.yaml)

1-bulkRNAseq: refilled lung IMs

2021-11-17 14:12:45 +0100

Abstract

Lung interstitium macrophages (IMs) are non-alveolar resident tissue macrophages which contribute to the lung homeostasis. These cells were reported to be heterogeneous by our group and other teams, which contains two main distinct subpopulations: CD206+ IMs and CD206- IMs. However, the exact origin of IMs and the transcriptional programs that control IM differentiation remains unclear. In recent report, we analyzed the refilled IMs in the course of time after induced IM depletion with single-cell RNA sequencing (10X Genomics Chromium) and bulk RNA sequencing.

Contents

1 Description

Lung interstitium macrophages (IMs) are non-alveolar resident tissue macrophages which contribute to the lung homeostasis. These cells were reported to be heterogeneous by our group and other teams, which contains two main distinct subpopulations: CD206+ IMs and CD206- IMs. However, the exact origin of IMs and the transcriptional programs that regulate IM differentiation remains unclear. In recent report, we analyzed the refilled IMs in the course of time after induced IM depletion with single-cell RNA sequencing (10X Genomics Chromium) and bulk RNA sequencing.

In this study, the de novo refilled CD206+ and CD206- IMs on Day 14 post-depletion were compared to those without depletion. Alvelar macrophages (AMs) samples were also included in this analysis and served as a reference. Results showed high similarity between de novo refilled and original IMs for both CD206+ and CD206- subsets. Only genes related to cell cycling were found upregulated in de novo refilled IMs.

Total RNA was extracted and concentrated from the different samples with the RNA Clean & Concentrator 5 (Zymo Research). Possible DNA contaminant were removed with DNAse I. RNA quality and quantity were evaluated using a 2100 bioanalyzer (Agilent) and the Quant-iT™ RiboGreen™ RNA Assay Kit (ThermoFisher). The RNA Integrity Number (RIN) was greater than 7 for all samples. In order to generate the libraries using the Truseq stranded mRNA kit (Illumina), 100 ng of RNA was used. These libraries were sequenced on an Illumina Novaseq sequencer on a SP flow cell in single read 100 bp. Sequence alignment with the human genome (version GRCh37), sequence counting and quality control were performed using the nf-core/rnaseq pipeline.[1] RNA-seq data were analyzed using R Bioconductor (3.12) and DESeq2 package (version 1.26.0)[2].

2 Counting from fastq data using nf-core/rnaseq pipeline

The following codes were used to do the mapping and counting.

```
nextflow run nf-core/rnaseq --input sample_list.csv --fasta GRCm38/fasta/
   genome.fa --gtf GRCm38/genes/genes.gtf --outdir counts/bulkRNAseq/ -
   profile docker
```

`sample_list.csv` is text file with 5 columns: group, replicate, fastq_1, fastq_2 and strandedness. Prepared following to the software's instructions.

3 Counts data processing

```
library(DESeq2)
library(ggplot2)
library(pheatmap)
library(RColorBrewer)
library(EnhancedVolcano)
library(forcats)

COUNTS <- read.table("./salmon.merged.gene_counts.csv", sep = "\t", header
    = T, row.names = NULL)

dim(COUNTS)
```

```
## [1] 22597    19
```

Make gene names as rownames:

```
Genes <- COUNTS$gene_id
rownames(COUNTS) = make.names(Genes, unique = TRUE)
```

```
COUNTS <- COUNTS[, -1]
COUNTS <- round(COUNTS, digits = 0)
head(COUNTS, 3)
```

```
## # A tibble: 3 x 18
##   AM.WT_R1 AM.WT_R2 AM.WT_R3 CD206neg.IM.CRE_~ CD206neg.IM.CRE~
   CD206neg.IM.CRE~
##      <dbl>    <dbl>    <dbl>             <dbl>            <dbl>
             <dbl>
## 1     3868     4493     4063              1944             1882
              2059
## 2        0        0        0                 0                0
                 0
## 3      123      156      147                30               54
                82
## # ... with 12 more variables: CD206neg.IM.WT_R1 <dbl>, CD206neg.IM.
   WT_R2 <dbl>,
## #   CD206neg.IM.WT_R3 <dbl>, CD206pos.IM.CRE_R1 <dbl>,
## #   CD206pos.IM.CRE_R2 <dbl>, CD206pos.IM.CRE_R3 <dbl>,
## #   CD206pos.IM.WT_R1 <dbl>, CD206pos.IM.WT_R2 <dbl>, CD206pos.IM.WT_R3
    <dbl>,
## #   Ly6Cpos.Mo.WT_R1 <dbl>, Ly6Cpos.Mo.WT_R2 <dbl>, Ly6Cpos.Mo.WT_R3 <
   dbl>
```

```
library(org.Mm.eg.db)
symbols <- mapIds(org.Mm.eg.db, keys = rownames(COUNTS), keytype = "
   ENSEMBL", column = "SYMBOL")
symbols.uniq <- na.omit(unique(symbols))

# remove adundant ensembl ids:
COUNTS <- COUNTS[match(symbols.uniq, symbols), ]

# use symbols as rownames:
rownames(COUNTS) <- symbols.uniq

head(COUNTS)
```

```
## # A tibble: 6 x 18
##   AM.WT_R1 AM.WT_R2 AM.WT_R3 CD206neg.IM.CRE_~ CD206neg.IM.CRE~
   CD206neg.IM.CRE~
##      <dbl>    <dbl>    <dbl>             <dbl>            <dbl>
             <dbl>
## 1     3868     4493     4063              1944             1882
              2059
## 2        0        0        0                 0                0
                 0
## 3      123      156      147                30               54
                82
## 4        0        3        0                43               20
                16
## 5        5       10        0                 0                0
                 0
```

```
## 6      270      373      213                168              266
                294
## # ... with 12 more variables: CD206neg.IM.WT_R1 <dbl>, CD206neg.IM.
   WT_R2 <dbl>,
## #   CD206neg.IM.WT_R3 <dbl>, CD206pos.IM.CRE_R1 <dbl>,
## #   CD206pos.IM.CRE_R2 <dbl>, CD206pos.IM.CRE_R3 <dbl>,
## #   CD206pos.IM.WT_R1 <dbl>, CD206pos.IM.WT_R2 <dbl>, CD206pos.IM.WT_R3
    <dbl>,
## #   Ly6Cpos.Mo.WT_R1 <dbl>, Ly6Cpos.Mo.WT_R2 <dbl>, Ly6Cpos.Mo.WT_R3 <
   dbl>
```

4 Make metadata for bulkRNAseq samples

```
SampleSheet <- data.frame(groupName = rep(c("AM", "CD206- refilled IM", "
   CD206- control IM",
    "CD206+ refilled IM", "CD206+ control IM", "Ly6C+ Mo"), each = 3),
       cellType1 = c(rep("Mac",
    15), rep("Mo", 3)), cellType2 = c(rep("AM", 3), rep("IM", 12), rep("Mo
       ", 3)),
    cellType3 = rep(c("AM", "CD206neg IM", "CD206neg IM", "CD206pos IM", "
       CD206pos IM",
        "Ly6Cpos Mo"), each = 3), treatment = c(rep(rep(c("control", "
           refilled"),
        each = 3), 2), rep("control", 6)))
rownames(SampleSheet) <- colnames(COUNTS)

SampleSheet
```

```
## # A tibble: 18 x 5
##    groupName          cellType1 cellType2 cellType3   treatment
##    <chr>              <chr>     <chr>     <chr>       <chr>
##  1 AM                 Mac       AM        AM          control
##  2 AM                 Mac       AM        AM          control
##  3 AM                 Mac       AM        AM          control
##  4 CD206- refilled IM Mac       IM        CD206neg IM refilled
##  5 CD206- refilled IM Mac       IM        CD206neg IM refilled
##  6 CD206- refilled IM Mac       IM        CD206neg IM refilled
##  7 CD206- control IM  Mac       IM        CD206neg IM control
##  8 CD206- control IM  Mac       IM        CD206neg IM control
##  9 CD206- control IM  Mac       IM        CD206neg IM control
## 10 CD206+ refilled IM Mac       IM        CD206pos IM refilled
## 11 CD206+ refilled IM Mac       IM        CD206pos IM refilled
## 12 CD206+ refilled IM Mac       IM        CD206pos IM refilled
## 13 CD206+ control IM  Mac       IM        CD206pos IM control
## 14 CD206+ control IM  Mac       IM        CD206pos IM control
## 15 CD206+ control IM  Mac       IM        CD206pos IM control
## 16 Ly6C+ Mo           Mo        Mo        Ly6Cpos Mo  control
## 17 Ly6C+ Mo           Mo        Mo        Ly6Cpos Mo  control
## 18 Ly6C+ Mo           Mo        Mo        Ly6Cpos Mo  control
```

```
write.csv(SampleSheet, file = "./SampleSheet_metadata.csv")
```

5 DESeq2 analysis

```
dds <- DESeqDataSetFromMatrix(countData = COUNTS, colData = SampleSheet,
   design = ~cellType3 +
    treatment)

dds
```

```
## class: DESeqDataSet
## dim: 21616 18
## metadata(1): version
## assays(1): counts
## rownames(21616): Gnai3 Pbsn ... Btbd35f19 Cldn34c3
## rowData names(0):
## colnames(18): AM.WT_R1 AM.WT_R2 ... Ly6Cpos.Mo.WT_R2 Ly6Cpos.Mo.WT_R3
## colData names(5): groupName cellType1 cellType2 cellType3 treatment
```

5.1 Perform rlog transformation for distances and PCA

```
# keep only genes with more than a single read
dds <- dds[rowSums(counts(dds)) > 1, ]

# perform rlog transformation for distances (for clustering) and PCA
rld <- rlog(dds)
```

```
dds <- dds[rowSums(counts(dds)) > 1, ]
nrow(dds)
```

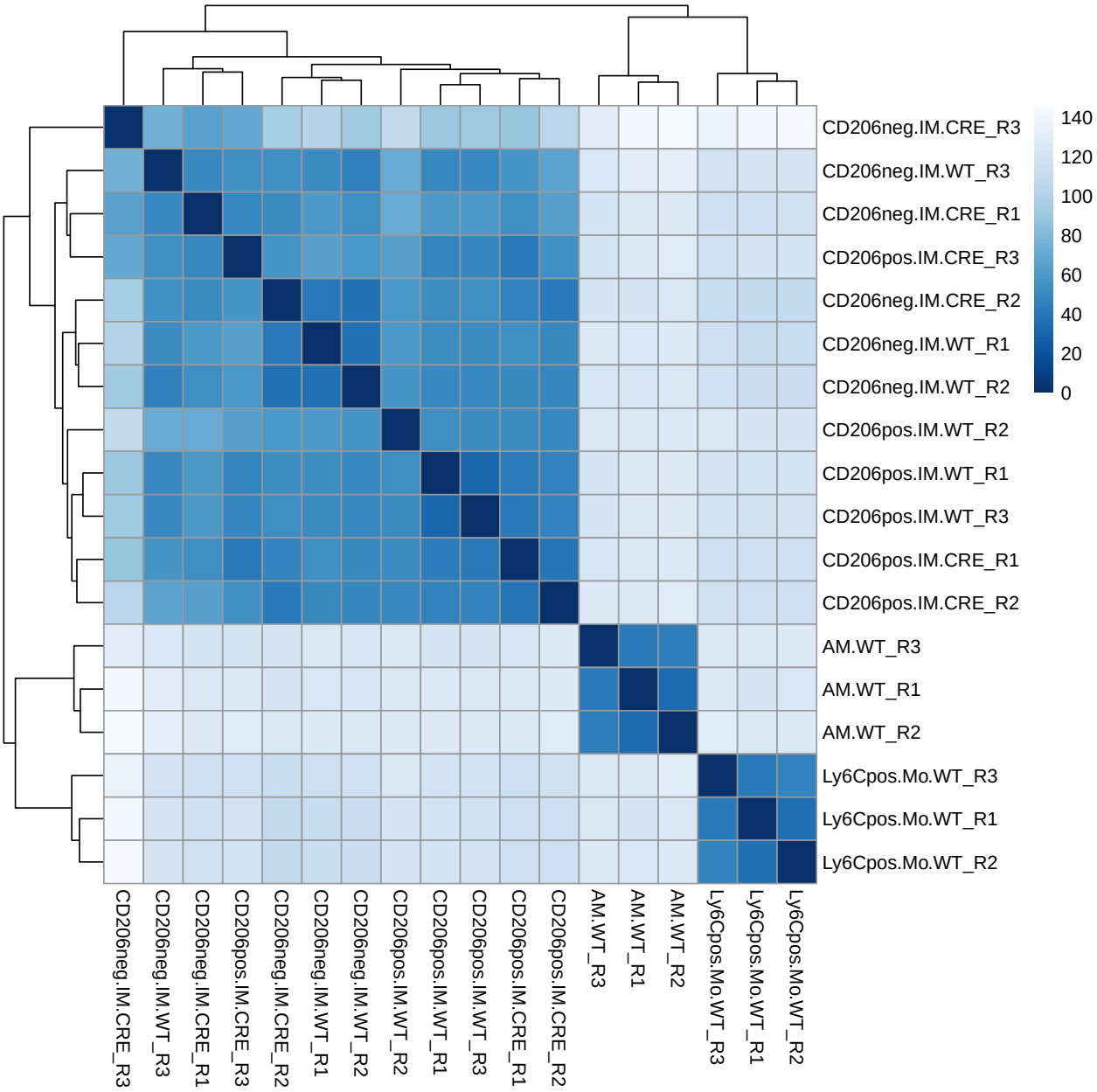
```
## [1] 16677
```

Calculate sample-to-sammple distances

```
sampleDists <- dist(t(assay(rld)))
sampleDistMatrix <- as.matrix(sampleDists)
```

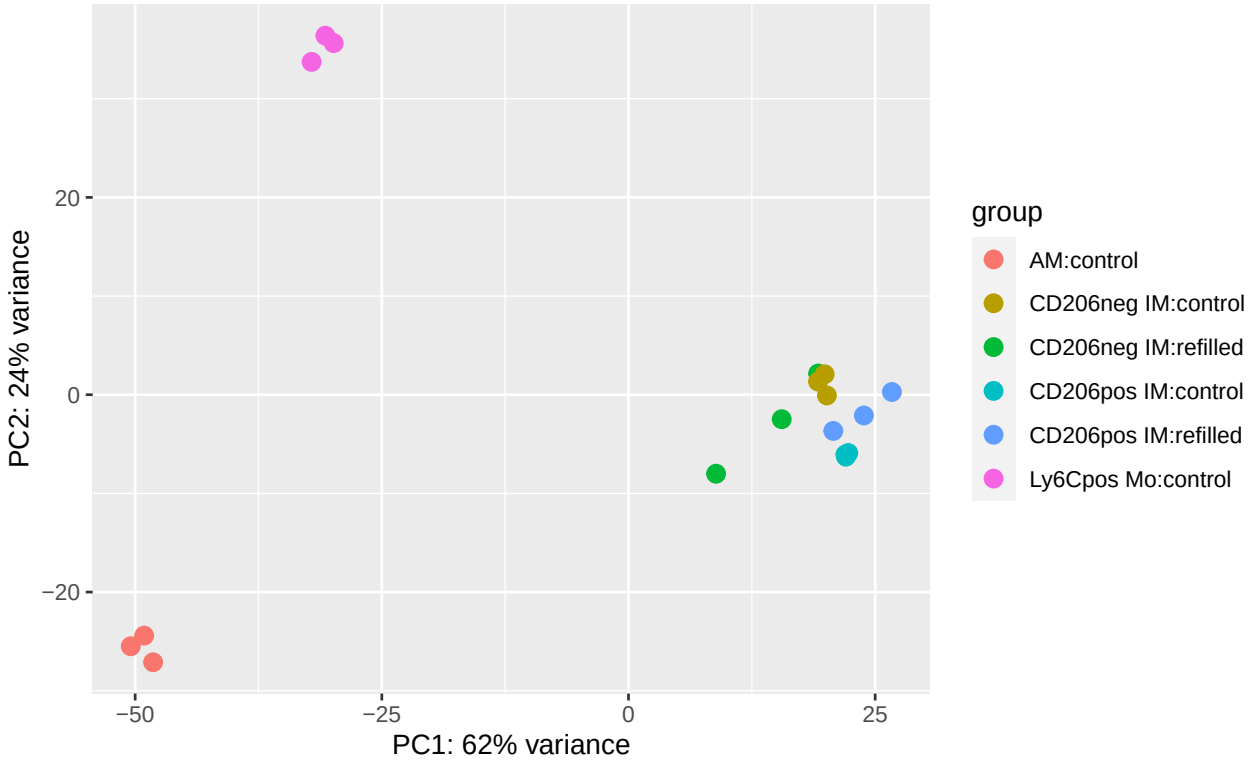
5.2 Heatmap

```
colors <- colorRampPalette(rev(brewer.pal(ncol(COUNTS), "Blues")))(255)
heatmap <- pheatmap(sampleDistMatrix, clustering_distance_rows =
   sampleDists, clustering_distance_cols = sampleDists,
    col = colors)
```



5.3 PCA analysis

```
plotPCA <- plotPCA(rld, intgroup = c("cellType3", "treatment"))
plotPCA
```



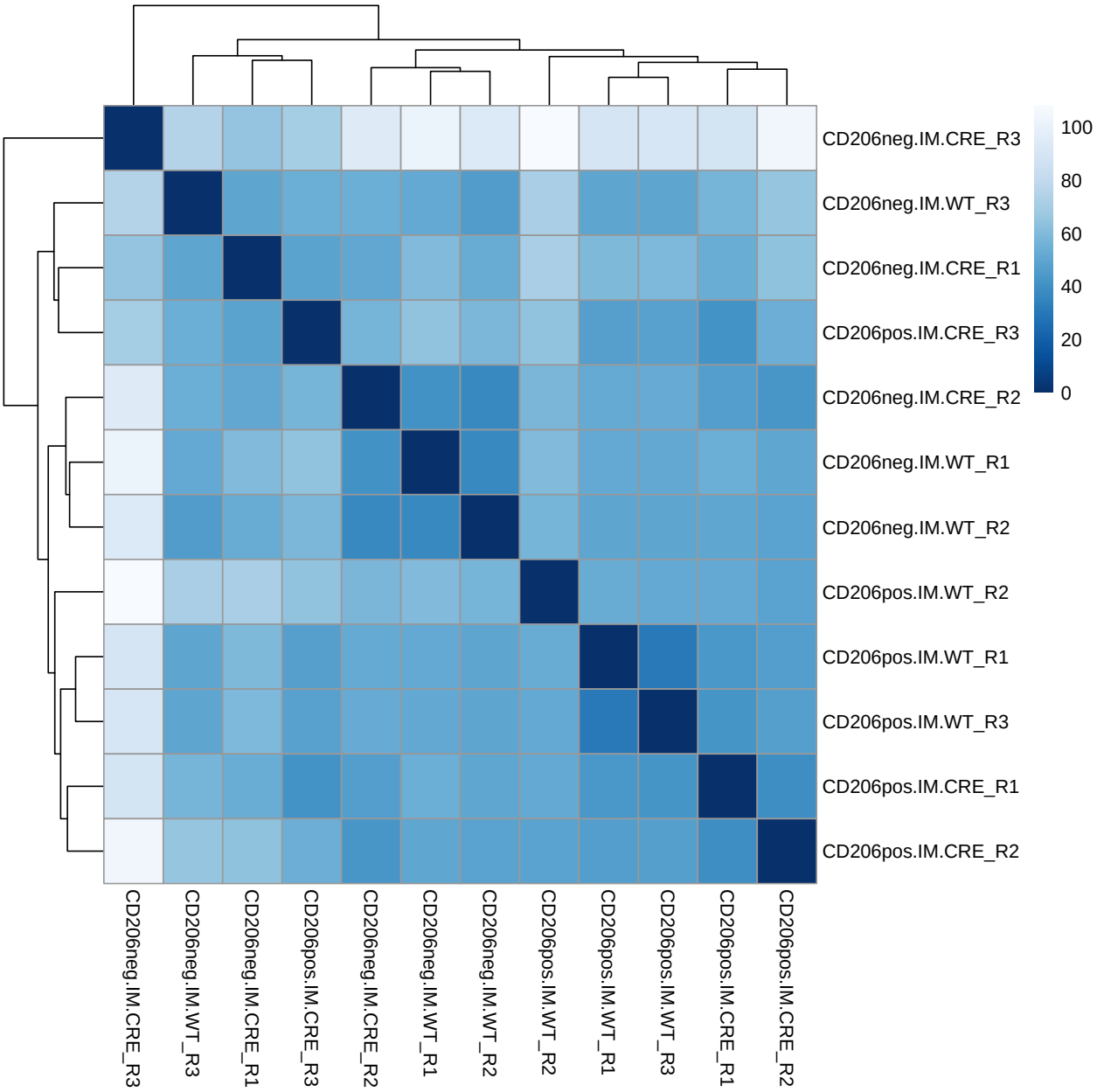
AM, IM and monocytes are grouped together. The refilled/control IMs are highly similar.

Redo PCA with only IMs:

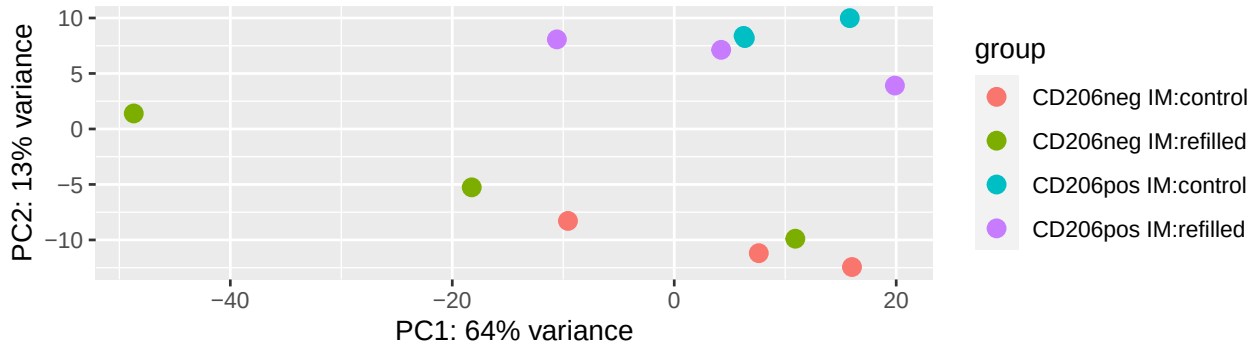
```
rld.im <- rld[, SampleSheet$cellType2 == "IM"]
sampleDists.im <- dist(t(assay(rld.im)))
sampleDistMatrix.im <- as.matrix(sampleDists.im)
```

Heatmap

```
heatmap <- pheatmap(sampleDistMatrix.im, clustering_distance_rows =
   sampleDists.im,
    clustering_distance_cols = sampleDists.im, col = colors)
```




```
plotPCA(rld.im, intgroup = c("cellType3", "treatment"))
```



Nearly no difference could be found between refilled and control IM in any subset.

5.4 Differentially expressed (DE) genes in comparing refilled vs control IMs

```
# redo with only IM samples:
dds.im <- DESeqDataSetFromMatrix(countData = COUNTS[, SampleSheet$
   cellType2 == "IM"],
    colData = SampleSheet[SampleSheet$cellType2 == "IM", ], design = ~
       cellType3 +
        treatment)
dds.im <- dds.im[rowSums(counts(dds.im)) > 1, ]

dds.im <- DESeq(dds.im)
resultsNames(dds.im)
```

```
## [1] "Intercept"
## [2] "cellType3_CD206pos.IM_vs_CD206neg.IM"
## [3] "treatment_refilled_vs_control"
```

```
res_refilled_vs_control <- results(dds.im, contrast = c("treatment", "
   refilled",
    "control"))
summary(res_refilled_vs_control)
```

```
##
## out of 16424 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 831, 5.1%
## LFC < 0 (down)     : 900, 5.5%
## outliers [1]       : 59, 0.36%
## low counts [2]     : 1902, 12%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res_refilled_vs_control_Shrunk <- lfcShrink(dds.im, contrast = c("
   treatment", "refilled",
    "control"), res = res_refilled_vs_control, type = "normal")

refilled_vs_control <- merge(x = as.data.frame(res_refilled_vs_control), y
    = as.data.frame(res_refilled_vs_control_Shrunk),
    by = c(0, 1))

head(refilled_vs_control)
```

```
## # A tibble: 6 x 12
##   Row.names     baseMean log2FoldChange.x lfcSE.x  stat.x     pvalue.x
      padj.x
##   <I<chr>>         <dbl>            <dbl>   <dbl>   <dbl>        <dbl>
       <dbl>
## 1 0610009B22Rik     67.6          0.00807   0.344  0.0235 0.981
           9.93e-1
## 2 0610010F05Rik    329.           0.0504    0.159  0.316  0.752
           8.94e-1
```

```
## 3 0610010K14Rik    408.           0.198     0.101  1.95   0.0509          6
           2.38e-1
## 4 0610012G03Rik    221.           0.264     0.140  1.88   0.0596          7
           2.59e-1
## 5 0610030E20Rik    658.          -0.287     0.111 -2.58   0.00986         8
          8.83e-2
## 6 0610040J01Rik    184.          -0.914     0.161 -5.67   0.0000000140    9
     1.56e-6
## # ... with 5 more variables: log2FoldChange.y <dbl>, lfcSE.y <dbl>,      10
## #   stat.y <dbl>, pvalue.y <dbl>, padj.y <dbl>                            11
```

6 Export DE genes for other analyses

```
Genes2 <- refilled_vs_control$Row.names                                      1
rownames(refilled_vs_control) = make.names(Genes2, unique = TRUE)            2
refilled_vs_control <- refilled_vs_control[, -1]                             3
```

Filter

```
refilled_vs_control <- refilled_vs_control[!is.na(refilled_vs_control$padj   1
   .y), ]
refilled_vs_control_1 <- subset(refilled_vs_control, padj.y < 0.05)          2
dim(refilled_vs_control_1)                                                   3
```

```
## [1] 1179   11                                                            1
```

```
refilled_vs_control_ordered <- refilled_vs_control_1[order(-refilled_vs_     1
   control_1$log2FoldChange.y),
    ]                                                                        2
refilled_vs_control_ordered                                                  3
```

```
## # A tibble: 1,179 x 11                                                   1
##    baseMean log2FoldChange.x lfcSE.x stat.x pvalue.x   padj.x            2
   log2FoldChange.y
##       <dbl>            <dbl>   <dbl>  <dbl>    <dbl>    <dbl>           3
             <dbl>
##  1    526.              2.25  0.140   16.1  2.06e-58 2.98e-54           4
             1.97
##  2    173.              1.66  0.169    9.82 9.03e-23 1.19e-19           5
             1.38
##  3    127.              1.84  0.218    8.44 3.10e-17 1.87e-14           6
             1.37
##  4   1072.              1.32  0.107   12.4  3.31e-35 7.97e-32           7
             1.22
##  5    550.              1.32  0.147    9.03 1.79e-19 1.23e-16           8
             1.15
##  6   2613.              1.22  0.0953  12.8  3.09e-37 1.49e-33           9
             1.14
##  7    444.              1.26  0.139    9.06 1.29e-19 9.31e-17           10
             1.10
##  8     59.4             1.77  0.300    5.91 3.51e- 9 4.66e- 7           11
             1.07
```

```
##  9    407.              1.19  0.148    8.04 9.26e-16 4.96e-13
               1.03
## 10   1483.              1.14  0.141    8.10 5.45e-16 3.03e-13
               0.997
## # ... with 1,169 more rows, and 4 more variables: lfcSE.y <dbl>, stat.y
    <dbl>,
## #   pvalue.y <dbl>, padj.y <dbl>
```

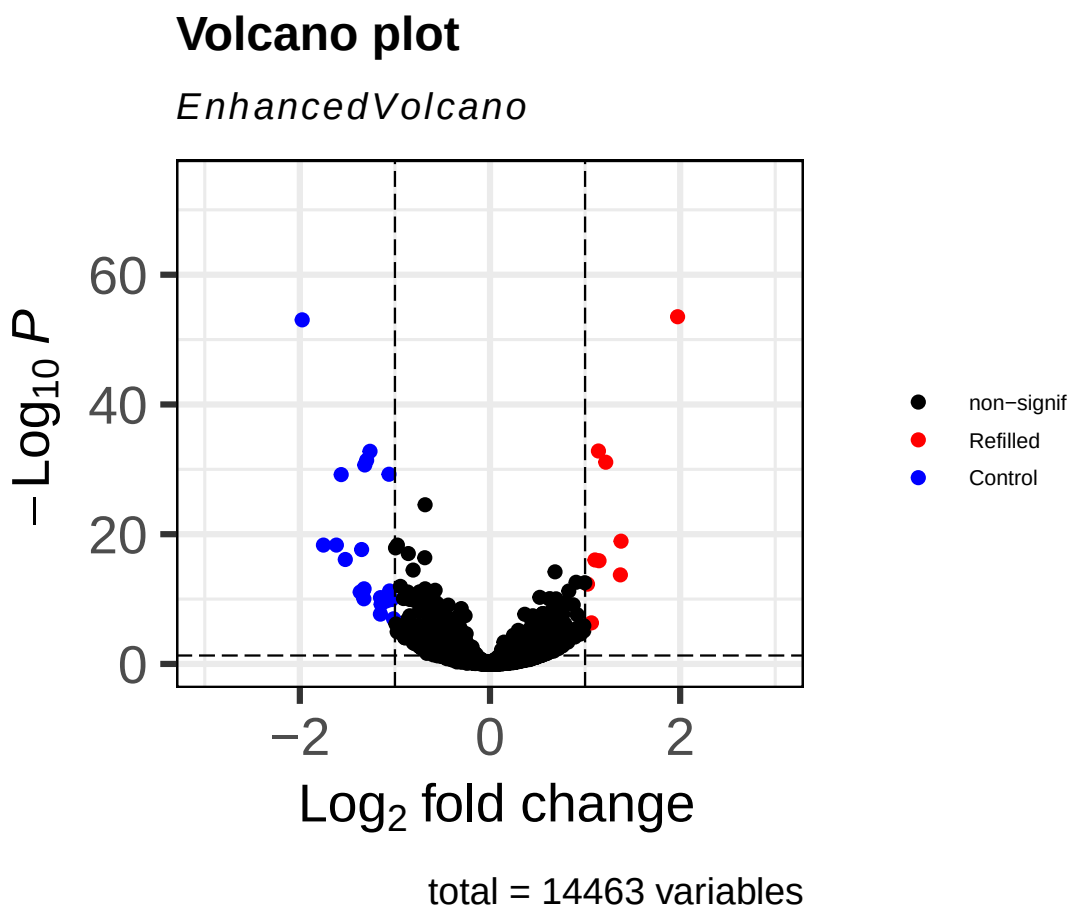
Save data for other analyses

```
write.table(as.data.frame(refilled_vs_control_ordered), "./Results_
   refilled_vs_control_in_IMs.txt",
    sep = "\t", row.names = T, col.names = T)
```

7 Volcano plots

```
keyvals <- rep("black", nrow(refilled_vs_control))
names(keyvals) <- rep("non-signif", nrow(refilled_vs_control))

keyvals[which(refilled_vs_control$log2FoldChange.y > 1)] <- "red"
names(keyvals)[which(refilled_vs_control$log2FoldChange.y > 1)] <- "
   Refilled"

keyvals[which(refilled_vs_control$log2FoldChange.y < -1)] <- "blue"
names(keyvals)[which(refilled_vs_control$log2FoldChange.y < -1)] <- "
   Control"
```

```
EnhancedVolcano(refilled_vs_control, lab = rownames(refilled_vs_control),
   x = "log2FoldChange.y",
    y = "padj.y", xlim = c(-3, 3), ylim = c(0, -log10(1e-74)), labSize =
       0, pCutoff = 0.05,
    FCcutoff = 1, colAlpha = 1, colCustom = keyvals, legendLabSize = 8,
       legendIconSize = 2,
    border = "full", legendPosition = "right", axisLabSize = 20)
```



What are these genes?

```
# the Refilled-related genes:
rownames(refilled_vs_control)[names(keyvals) == "Refilled"]
```

```
## [1] "AA467197" "Cd109"    "Cd226"    "Cspg4"    "Dpysl3"   "Gm21188"  "
    Igf2r"
## [8] "Rarb"     "S100a4"
```

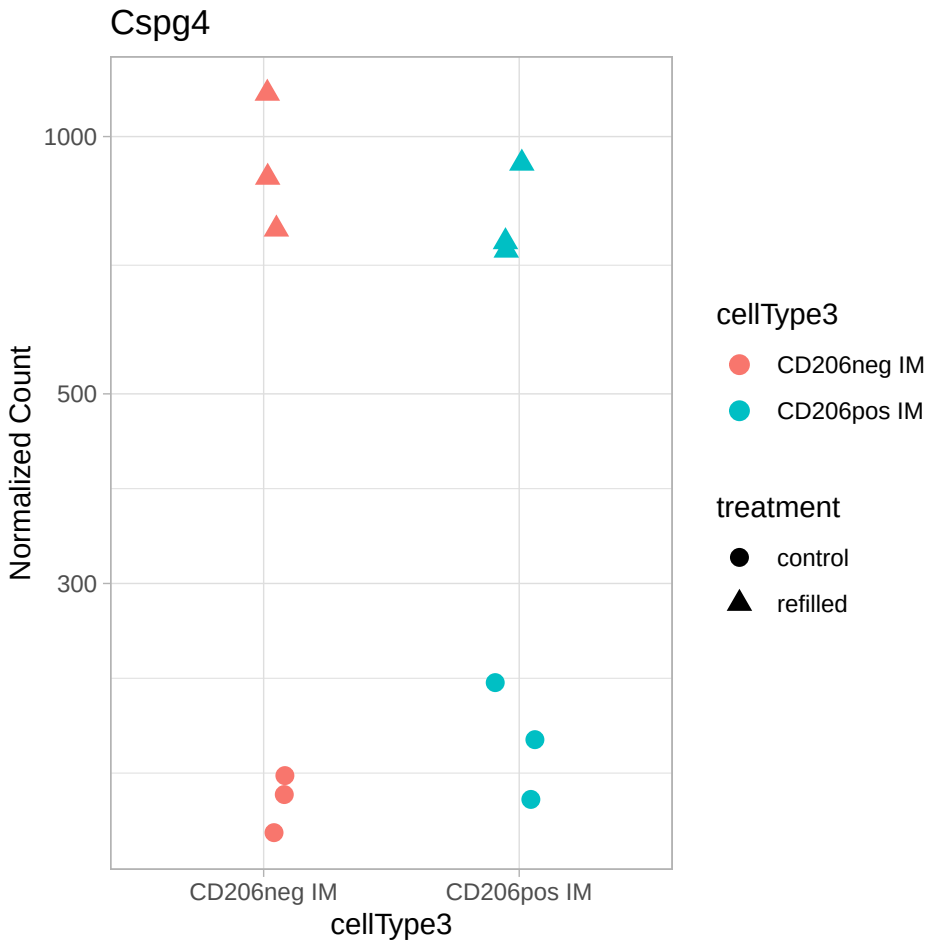
```
# the Refilled-related genes:
rownames(refilled_vs_control)[names(keyvals) == "Control"]
```

```
##  [1] "Adam22"  "Capn3"   "Cd22"    "Cd4"     "Cpne8"   "Edil3"   "H2.M2
    "
##  [8] "Igf2bp3" "Il18r1"  "Ildr2"   "Klhl13"  "Lilra5"  "Lrrc3"   "Mmp13
    "
## [15] "Nes"     "Peg10"   "Plag1"   "Ppp1r9a" "Tcim"    "Tmem119" "
    Zfp827"
```

7.1 Plot the top DE genes

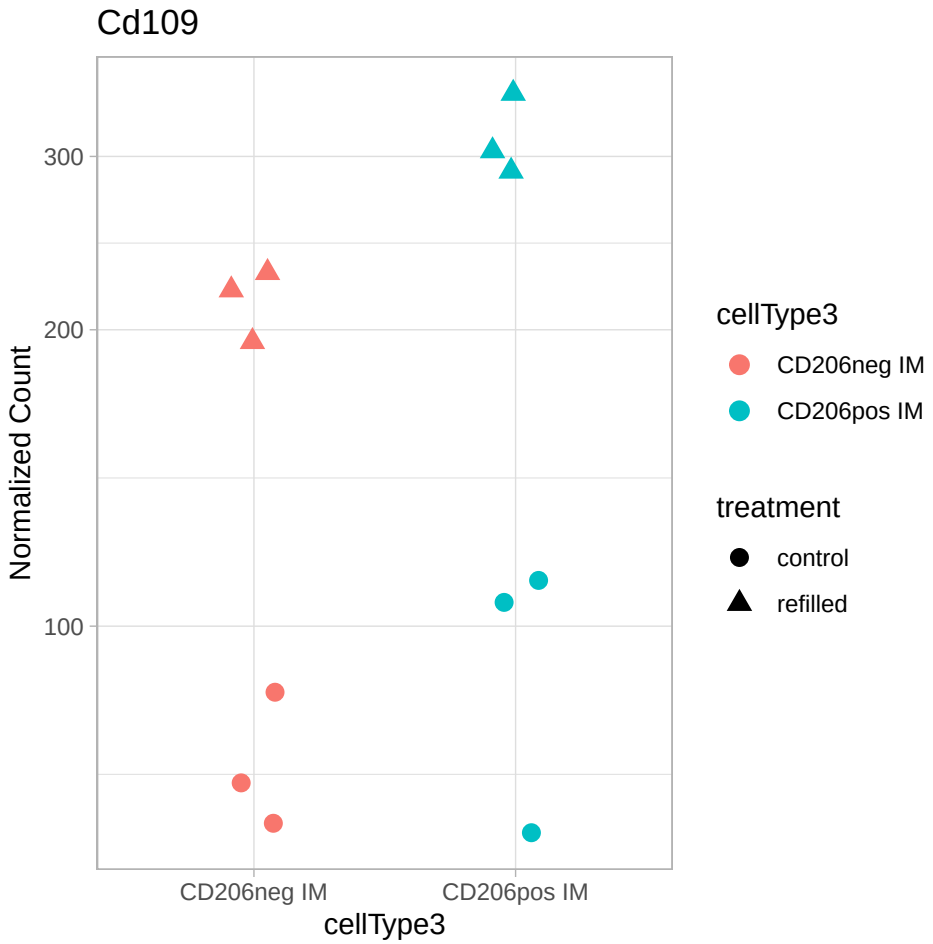
```
# plotCount Cspg4
data <- plotCounts(dds.im, gene = "Cspg4", intgroup = c("cellType3", "
    treatment"),
```

```
    returnData = TRUE)

ggplot(data, aes(x = cellType3, y = count, color = cellType3, shape =
    treatment)) +
    scale_y_log10() + geom_point(position = position_jitter(width = 0.1,
        height = 0),
    size = 3) + ggtitle("Cspg4") + theme(plot.title = element_text(hjust =
        0.5, size = 20)) +
    # scale_color_manual(values = c('#371dad','#ff8e03','#74c72a'))+
ylab("Normalized Count") + theme_linedraw() + theme_light()
```

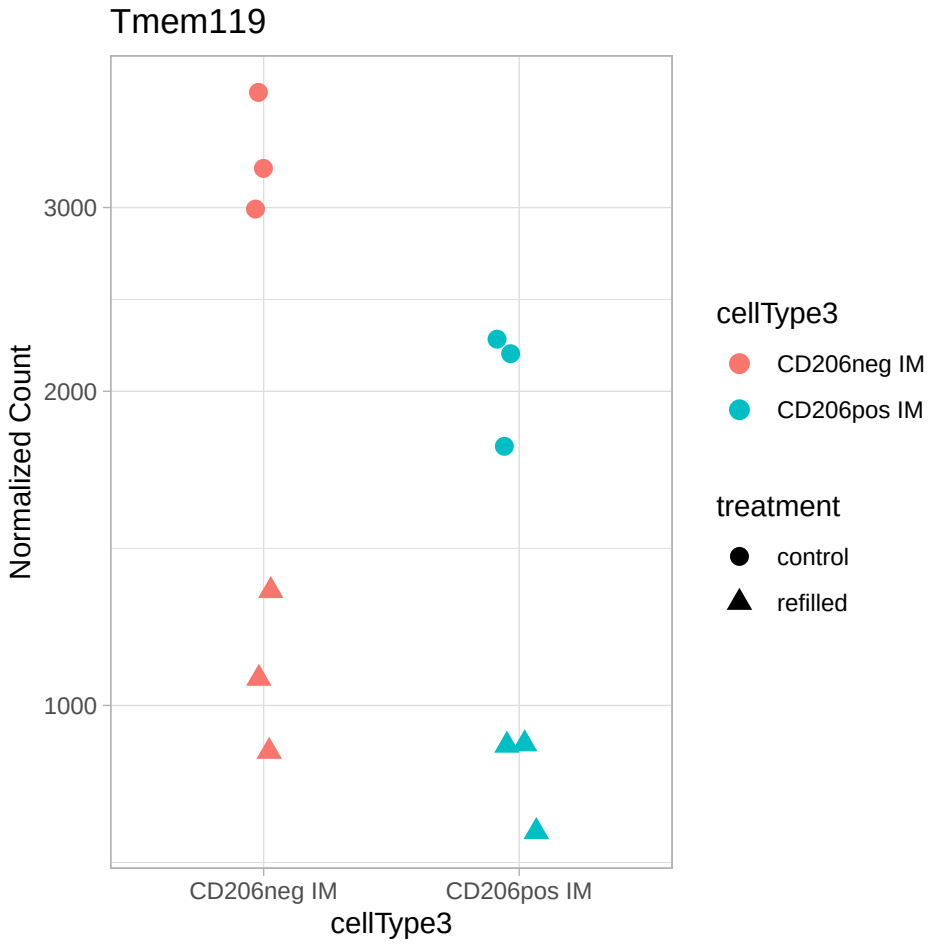



```
# plotCount Cd109
data <- plotCounts(dds.im, gene = "Cd109", intgroup = c("cellType3", "
    treatment"),
    returnData = TRUE)

ggplot(data, aes(x = cellType3, y = count, color = cellType3, shape =
    treatment)) +
    scale_y_log10() + geom_point(position = position_jitter(width = 0.1,
        height = 0),
    size = 3) + ggtitle("Cd109") + theme(plot.title = element_text(hjust =
        0.5, size = 20)) +
    # scale_color_manual(values = c('#371dad','#ff8e03','#74c72a'))+
```

```
ylab("Normalized␣Count") + theme_linedraw() + theme_light()
```




```
# plotCount Tmem119
data <- plotCounts(dds.im, gene = "Tmem119", intgroup = c("cellType3", "
   treatment"),
    returnData = TRUE)

ggplot(data, aes(x = cellType3, y = count, color = cellType3, shape =
   treatment)) +
    scale_y_log10() + geom_point(position = position_jitter(width = 0.1,
       height = 0),
    size = 3) + ggtitle("Tmem119") + theme(plot.title = element_text(hjust
       = 0.5,
    size = 20)) + ylab("Normalized␣Count") + theme_linedraw() + theme_
       light()
```



8 Session information

Nextflow:

```
Nextflow version: version 21.03.0.edge, build 5518 (05-03-2021 10:52 UTC)
Workflow profile: docker
Workflow repository: https://github.com/nf-core/rnaseq, revision master (
    commit hash 3643a94411b65f42bce5357c5015603099556ad9)
```

Software version used by Workflow:

```
bedtools    2.29.2
bioconductor-summarizedexperiment   1.20.0
bioconductor-tximeta    1.8.0
deseq2  1.28.0
dupradar    1.18.0
fastqc  0.11.9
nextflow    21.03.0.edge
nf-core/rnaseq  3.0
picard  2.23.9
preseq  2.0.3
qualimap    2.2.2-dev
rseqc   3.0.1
salmon  1.4.0
```

```
samtools    1.10
star    2.6.1d
stringtie   2.1.4
subread 2.0.1
trimgalore  0.6.6
ucsc    377
```

R sesssion:

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils
   datasets
## [8] methods   base
##
## other attached packages:
##  [1] org.Mm.eg.db_3.12.0         AnnotationDbi_1.52.0
##  [3] forcats_0.5.1               EnhancedVolcano_1.8.0
##  [5] ggrepel_0.9.1               RColorBrewer_1.1-2
##  [7] pheatmap_1.0.12             ggplot2_3.3.5
##  [9] DESeq2_1.30.1               SummarizedExperiment_1.20.0
## [11] Biobase_2.50.0              MatrixGenerics_1.2.1
## [13] matrixStats_0.60.0          GenomicRanges_1.42.0
## [15] GenomeInfoDb_1.26.7         IRanges_2.24.1
## [17] S4Vectors_0.28.1            BiocGenerics_0.36.1
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           bit64_4.0.5            ash_1.0-15
##  [4] httr_1.4.2             tools_4.0.3            utf8_1.2.2
##  [7] R6_2.5.0               KernSmooth_2.23-20     vipor_0.4.5
## [10] DBI_1.1.1              colorspace_2.0-2       withr_2.4.2
## [13] tidyselect_1.1.1       ggrastr_0.2.3          ggalt_0.4.0
## [16] bit_4.0.4              compiler_4.0.3         extrafontdb_1.0
## [19] cli_3.0.1              formatR_1.11           DelayedArray_0.16.3
## [22] labeling_0.4.2         scales_1.1.1           proj4_1.0-10.1
## [25] genefilter_1.72.1      stringr_1.4.0          digest_0.6.27
## [28] rmarkdown_2.9          XVector_0.30.0         pkgconfig_2.0.3
```

```
## [31] htmltools_0.5.1.1      extrafont_0.17          highr_0.9               43
## [34] fastmap_1.1.0          maps_3.3.0              rlang_0.4.11            44
## [37] rstudioapi_0.13        RSQLite_2.2.7           farver_2.1.0            45
## [40] generics_0.1.0         BiocParallel_1.24.1     dplyr_1.0.7             46
## [43] RCurl_1.98-1.3         magrittr_2.0.1          GenomeInfoDbData_1      47
    .2.4
## [46] Matrix_1.3-4           Rcpp_1.0.7              ggbeeswarm_0.6.0        48
## [49] munsell_0.5.0          fansi_0.5.0             lifecycle_1.0.0         49
## [52] stringi_1.7.3          yaml_2.2.1              MASS_7.3-53             50
## [55] zlibbioc_1.36.0        grid_4.0.3              blob_1.2.2              51
## [58] crayon_1.4.1           lattice_0.20-41         splines_4.0.3           52
## [61] annotate_1.68.0        locfit_1.5-9.4          knitr_1.33              53
## [64] pillar_1.6.2           geneplotter_1.68.0      XML_3.99-0.6            54
## [67] glue_1.4.2             evaluate_0.14           vctrs_0.3.8             55
## [70] Rttf2pt1_1.3.9         gtable_0.3.0            purrr_0.3.4             56
## [73] assertthat_0.2.1       cachem_1.0.5            xfun_0.24               57
## [76] xtable_1.8-4           survival_3.2-7          tibble_3.1.3            58
## [79] beeswarm_0.4.0         memoise_2.0.0           ellipsis_0.3.2          59
```

References

1. Ewels, P. A. *et al.* The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology* (2020) doi:10.1038/s41587-020-0439-x.

2. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15**, 550 (2014).