Mafb-restricted local monocyte proliferation precedes lung interstitial macrophage differentiation

1-bulkRNAseq: refilled lung IMs

2022-03-09 00:33:45 +0100

Abstract

Resident tissue macrophages (RTM) are differentiated immune cells populating distinct niches and exhibiting important tissue-supportive functions. RTM maintenance is thought to rely on either monocyte engraftment and differentiation, or RTM self-renewal. Here, we developed an inducible mouse model of lung interstitial macrophage (IM) niche depletion and repopulation to investigate IM development in vivo. Using time-course single-cell RNA-sequencing analyses, bone marrow chimeras and gene targeting, we found that engrafted Ly6C+ classical monocytes could self-renew locally in a CSF1R-dependent manner before their differentiation into RTM. We further showed that the switch from monocyte proliferation towards IM subset specification was controlled by MafB, while c-Maf specifically regulated the identity of the CD206+ IM subset. Our data shed new light on the transcriptional regulation of IM development and provide evidence that, in the mononuclear phagocyte system, self-renewal is not merely restricted to myeloid progenitor cells and mature macrophages, but is also a tightly regulated capability of mature monocytes developing into RTM in vivo.

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. Sequence alignment with the mouse genome (version GRCm38), 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)
```

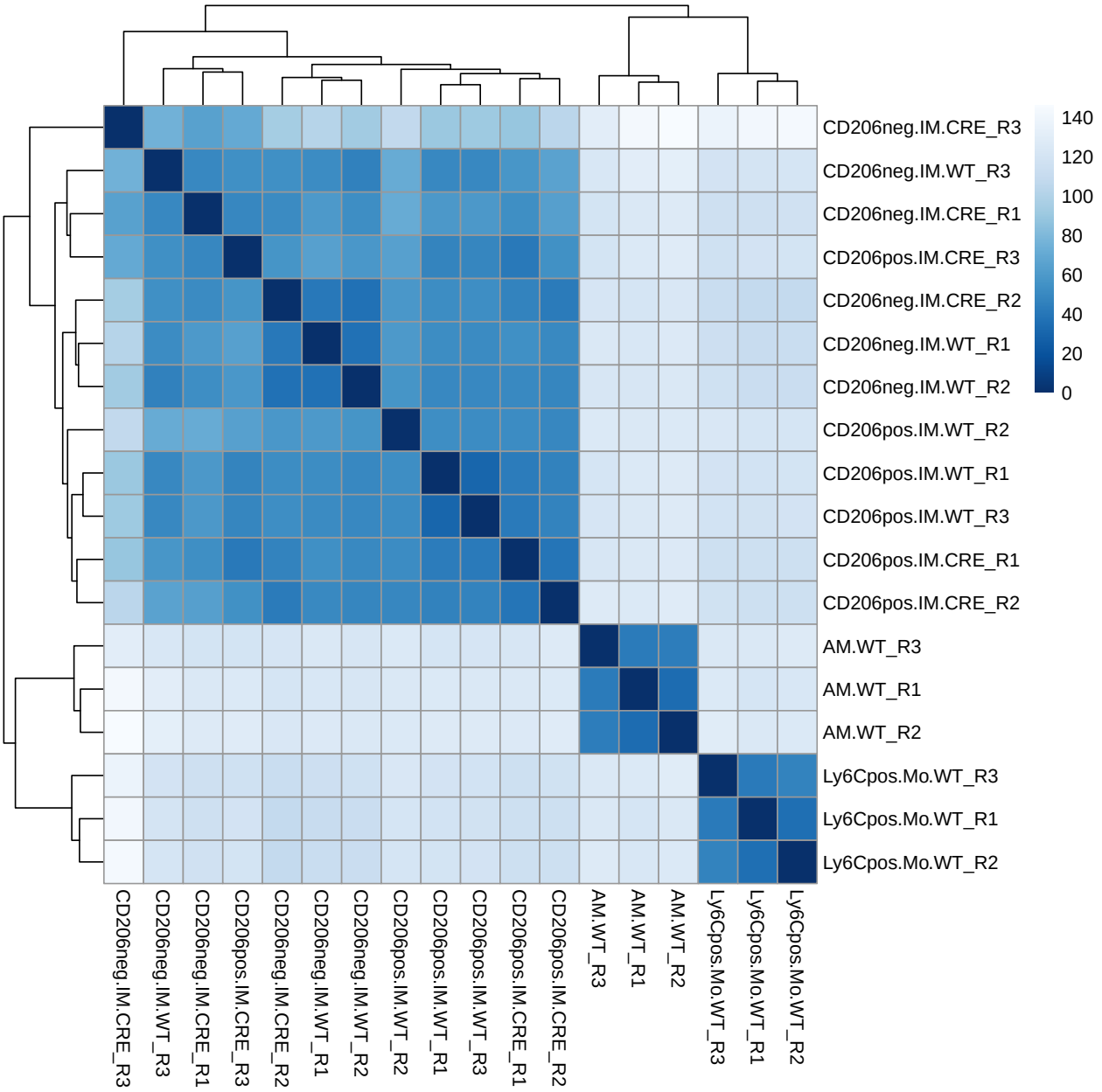
```
## [1] 16677
```

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

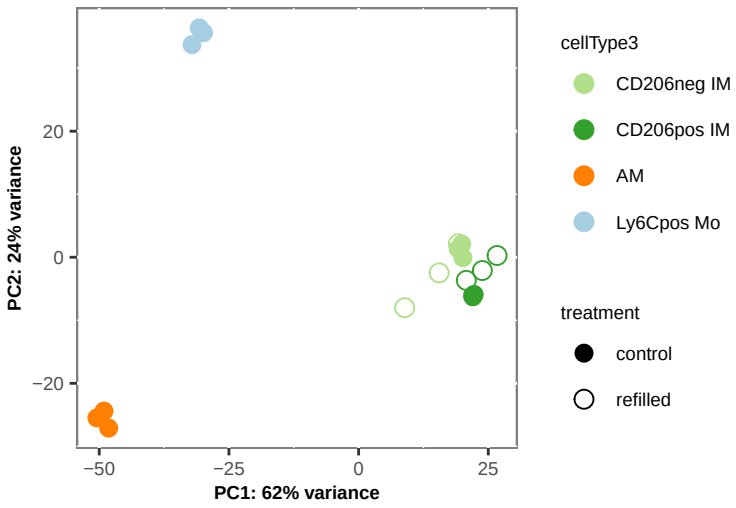
Calculate PCs:

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

Construct plot:

```
percentVar<-round(100 * attr(PlotData, "percentVar"))

ggplot(PlotData,aes(PC1,PC2)) +
    geom_point(size=3, aes(color = cellType3, shape = treatment)) +
    xlab(paste0("PC1: ", percentVar[1],"% variance")) +
    ylab(paste0("PC2: ", percentVar[2],"% variance")) +
```

```
  scale_color_manual(
    breaks = c( "CD206neg␣IM",
                "CD206pos␣IM",
                "AM",
                "Ly6Cpos␣Mo"),
    values = c(
                        "#B2DF8A", # CD206-
                        "#33A02C", # CD206+
                        "#FF7F00", # AM
                        "#A6CEE3" # Cla mo
                      )) +
  scale_shape_manual(
    breaks = c(
      "control",
      "refilled"
    ),
    values=c(
      16, #"CD206neg IM:control",
      1 #"CD206neg IM:refilled",
    ))+
  theme(
    aspect.ratio=1,
    panel.background = element_rect(fill = "white", colour = "grey50"),
    axis.text=element_text(size=7),
        axis.title=element_text(size=7,face="bold"),
    legend.key = element_blank(),
    legend.text = element_text(size=7),
    legend.title = element_text(size = 7),
  )
```



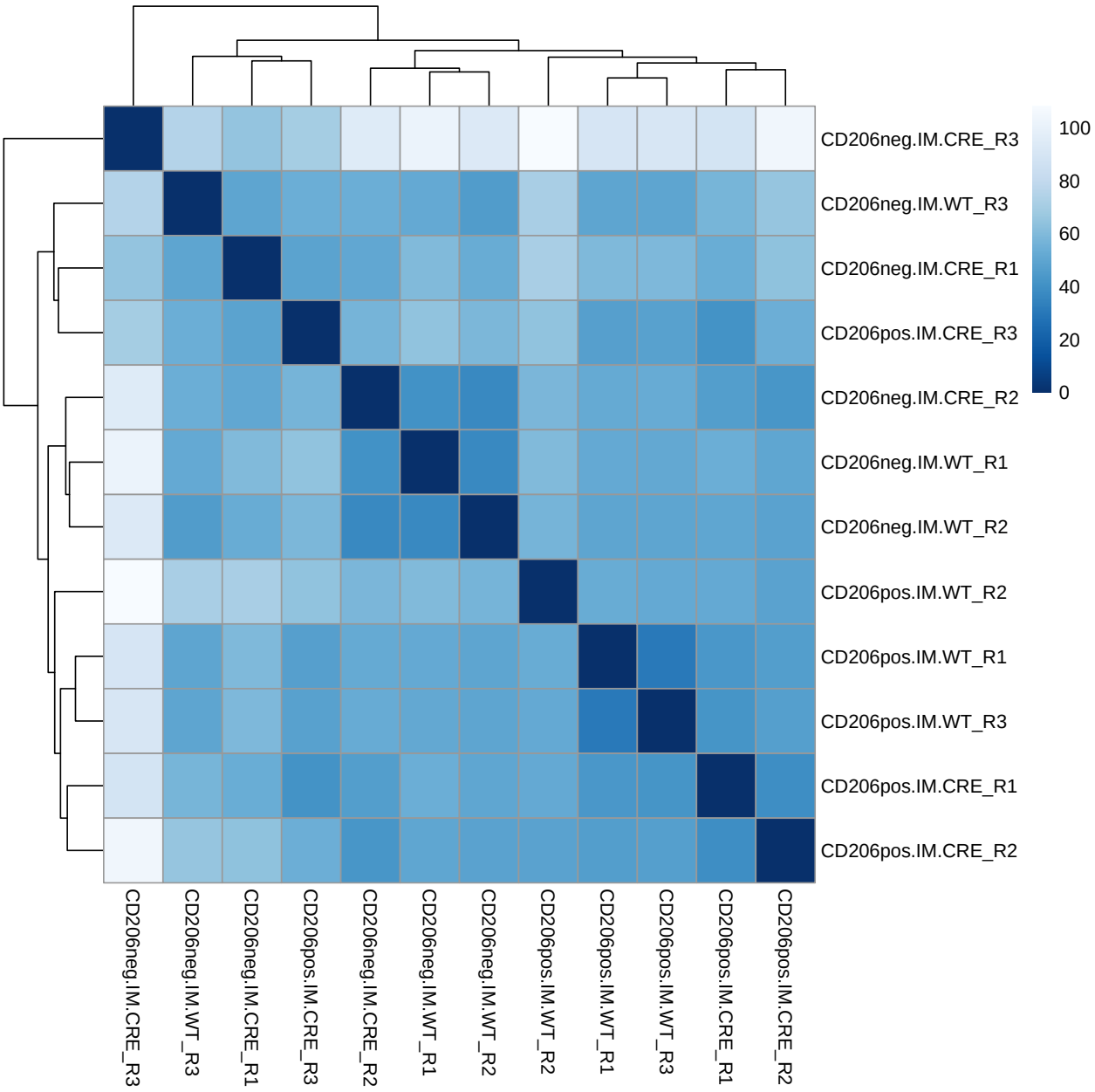
AM, IM and monocytes are grouped together. The refilled/control IMs are highly similar.

Redo PCA with only IMs:

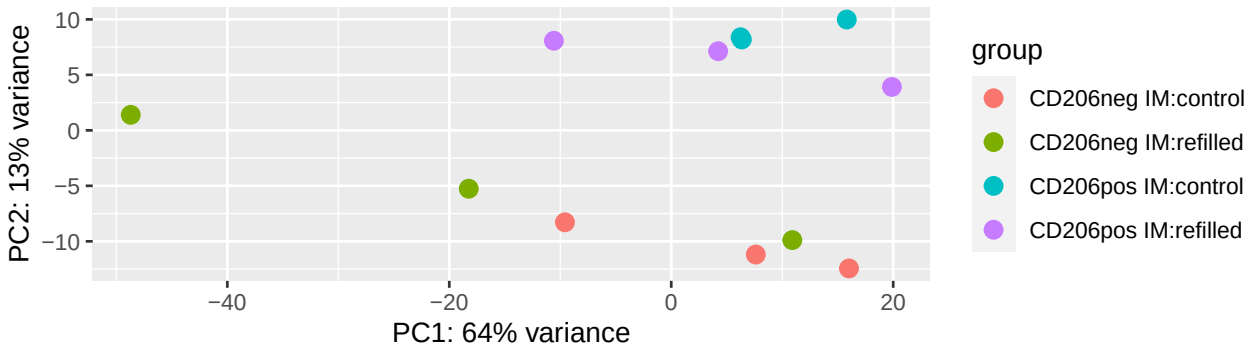
```
rld.im <- rld[, SampleSheet$cellType2 == "IM"]
sampleDists.im <- dist(t(assay(rld.im)))
sampleDistMatrix.im <- as.matrix(sampleDists.im)
```

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))
```

5.5 Differentially expressed (DE) genes in comparing refilled vs control CD206+ IMs

```
# redo with only CD206+ samples:
dds.cd206pos.im <- DESeqDataSetFromMatrix(countData = COUNTS[, SampleSheet$cellType3 ==
    "CD206pos␣IM"], colData = SampleSheet[SampleSheet$cellType3 == "CD206pos␣IM",
    ], design = ~treatment)
dds.cd206pos.im <- dds.cd206pos.im[rowSums(counts(dds.cd206pos.im)) > 1, ]

dds.cd206pos.im <- DESeq(dds.cd206pos.im)
resultsNames(dds.cd206pos.im)
```

```
## [1] "Intercept"                      "treatment_refilled_vs_control"
```

```
res_refilled_vs_control_CD206pos <- results(dds.cd206pos.im, contrast = c("treatment",
    "refilled", "control"))
summary(res_refilled_vs_control_CD206pos)
```

```
##
## out of 15463 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 170, 1.1%
## LFC < 0 (down)     : 306, 2%
## outliers [1]       : 2, 0.013%
## low counts [2]     : 1499, 9.7%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res_refilled_vs_control_CD206pos_Shrunk <- lfcShrink(dds.cd206pos.im, contrast = c("treatment",
    "refilled", "control"), res = res_refilled_vs_control_CD206pos, type = "normal")

refilled_vs_control_CD206pos <- merge(x = as.data.frame(res_refilled_vs_control_CD206pos),
    y = as.data.frame(res_refilled_vs_control_CD206pos_Shrunk), by = c(0, 1))
```

5.6 Differentially expressed (DE) genes in comparing refilled vs control CD206- IMs

```
# redo with only CD206- samples:
dds.CD206neg.im <- DESeqDataSetFromMatrix(countData = COUNTS[, SampleSheet$cellType3 ==
    "CD206neg␣IM"], colData = SampleSheet[SampleSheet$cellType3 == "CD206neg␣IM",
    ], design = ~treatment)
dds.CD206neg.im <- dds.CD206neg.im[rowSums(counts(dds.CD206neg.im)) > 1, ]
```

```

dds.CD206neg.im <- DESeq(dds.CD206neg.im)
resultsNames(dds.CD206neg.im)
```

```
## [1] "Intercept"                      "treatment_refilled_vs_control"
```

```
res_refilled_vs_control_CD206neg <- results(dds.CD206neg.im, contrast = c(
   "treatment",
    "refilled", "control"))
summary(res_refilled_vs_control_CD206neg)
```

```
##
## out of 16093 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 328, 2%
## LFC < 0 (down)     : 294, 1.8%
## outliers [1]       : 1, 0.0062%
## low counts [2]     : 2184, 14%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res_refilled_vs_control_CD206neg_Shrunk <- lfcShrink(dds.CD206neg.im,
   contrast = c("treatment",
    "refilled", "control"), res = res_refilled_vs_control_CD206neg, type =
        "normal")

refilled_vs_control_CD206neg <- merge(x = as.data.frame(res_refilled_vs_
   control_CD206neg),
    y = as.data.frame(res_refilled_vs_control_CD206neg_Shrunk), by = c(0,
       1))
```

6 Export DE genes for other analyses

```
Genes2 <- refilled_vs_control$Row.names
Genes2.CD206pos <- refilled_vs_control_CD206pos$Row.names
Genes2.CD206neg <- refilled_vs_control_CD206neg$Row.names

rownames(refilled_vs_control) = make.names(Genes2, unique = TRUE)
rownames(refilled_vs_control_CD206pos) = make.names(Genes2.CD206pos,
   unique = TRUE)
rownames(refilled_vs_control_CD206neg) = make.names(Genes2.CD206neg,
   unique = TRUE)


refilled_vs_control <- refilled_vs_control[, -1]
refilled_vs_control_CD206pos <- refilled_vs_control_CD206pos[, -1]
refilled_vs_control_CD206neg <- refilled_vs_control_CD206neg[, -1]
```

Filter

```
refilled_vs_control <- refilled_vs_control[!is.na(refilled_vs_control$padj
   .y), ]
refilled_vs_control_CD206pos <- refilled_vs_control_CD206pos[!is.na(
   refilled_vs_control_CD206pos$padj.y),
    ]
refilled_vs_control_CD206neg <- refilled_vs_control_CD206neg[!is.na(
   refilled_vs_control_CD206neg$padj.y),
    ]

refilled_vs_control_1 <- subset(refilled_vs_control, padj.y < 0.05)
refilled_vs_control_1_CD206pos <- subset(refilled_vs_control_CD206pos,
   padj.y < 0.05)
refilled_vs_control_1_CD206neg <- subset(refilled_vs_control_CD206neg,
   padj.y < 0.05)
```

```
refilled_vs_control_ordered <- refilled_vs_control_1[order(-refilled_vs_
   control_1$log2FoldChange.y),
    ]
refilled_vs_control_ordered_CD206pos <- refilled_vs_control_1_CD206pos[
   order(-refilled_vs_control_1_CD206pos$log2FoldChange.y),
    ]
refilled_vs_control_ordered_CD206neg <- refilled_vs_control_1_CD206neg[
   order(-refilled_vs_control_1_CD206neg$log2FoldChange.y),
    ]
```

Save data for other analyses

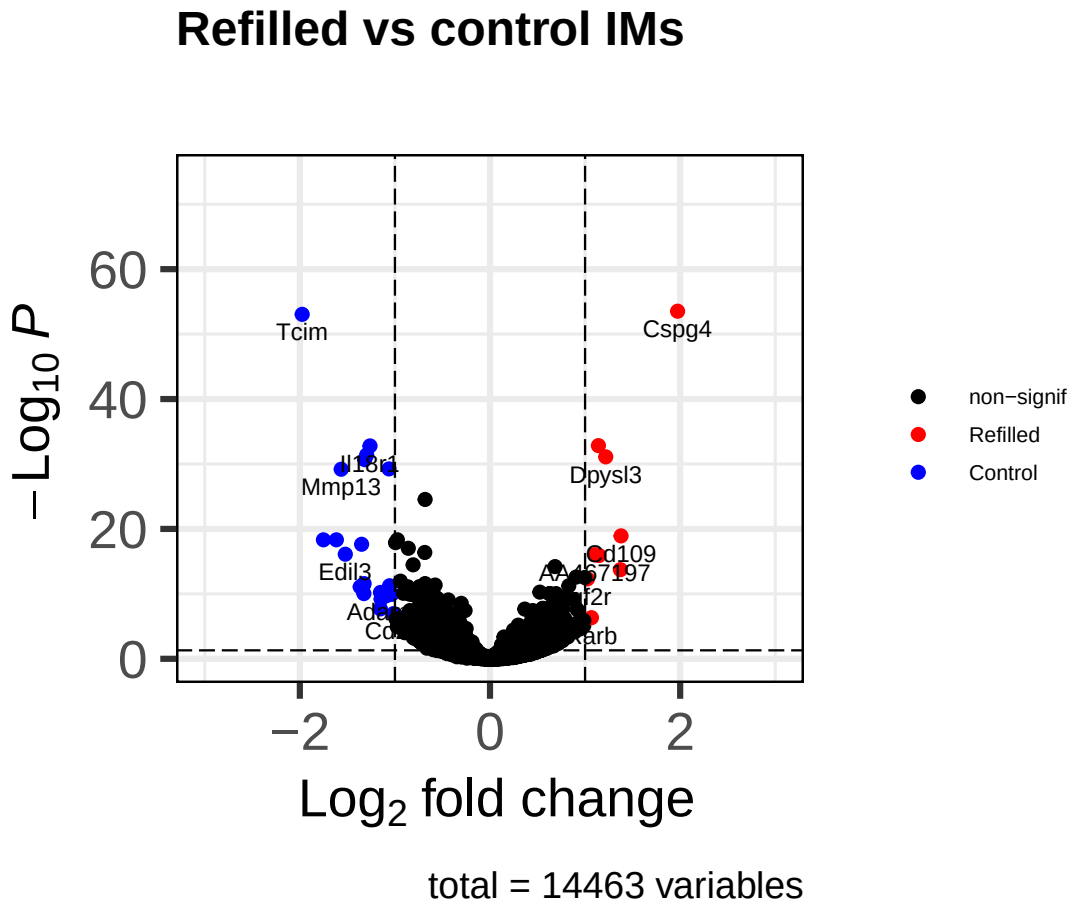
```
write.table(as.data.frame(refilled_vs_control_ordered), "./Results_
   refilled_vs_control_in_IMs.txt",
    sep = "\t", row.names = T, col.names = T)
write.table(as.data.frame(refilled_vs_control_ordered_CD206pos), "./
   Results_refilled_vs_control_in_CD206pos_IMs.txt",
    sep = "\t", row.names = T, col.names = T)
write.table(as.data.frame(refilled_vs_control_ordered_CD206neg), "./
   Results_refilled_vs_control_in_CD206neg_IMs.txt",
    sep = "\t", row.names = T, col.names = T)
```

7 Presentation of DE genes in volcano plots

7.1 Refilled vs control IMs

```
de.res <- refilled_vs_control

de.res$Gene <- rownames(de.res)
keyvals <- rep("black", nrow(de.res))
names(keyvals) <- rep("non-signif", nrow(de.res))

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

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

```
plot.vol <- EnhancedVolcano(de.res, subtitle = "", lab = rownames(de.res),
    x = "log2FoldChange.y",
    y = "padj.y", xlim = c(-3, 3), ylim = c(0, -log10(1e-74)), labSize =
       3, pCutoff = 0.05,
    FCcutoff = 1, colAlpha = 1, colCustom = keyvals, legendLabSize = 8,
       legendIconSize = 2,
    border = "full", legendPosition = "right", axisLabSize = 20, title = "
       Refilled vs control IMs")

plot.vol
```



What are these genes?

```
# the Refilled-related genes total IMs:
rownames(de.res)[names(keyvals) == "Refilled"]
```

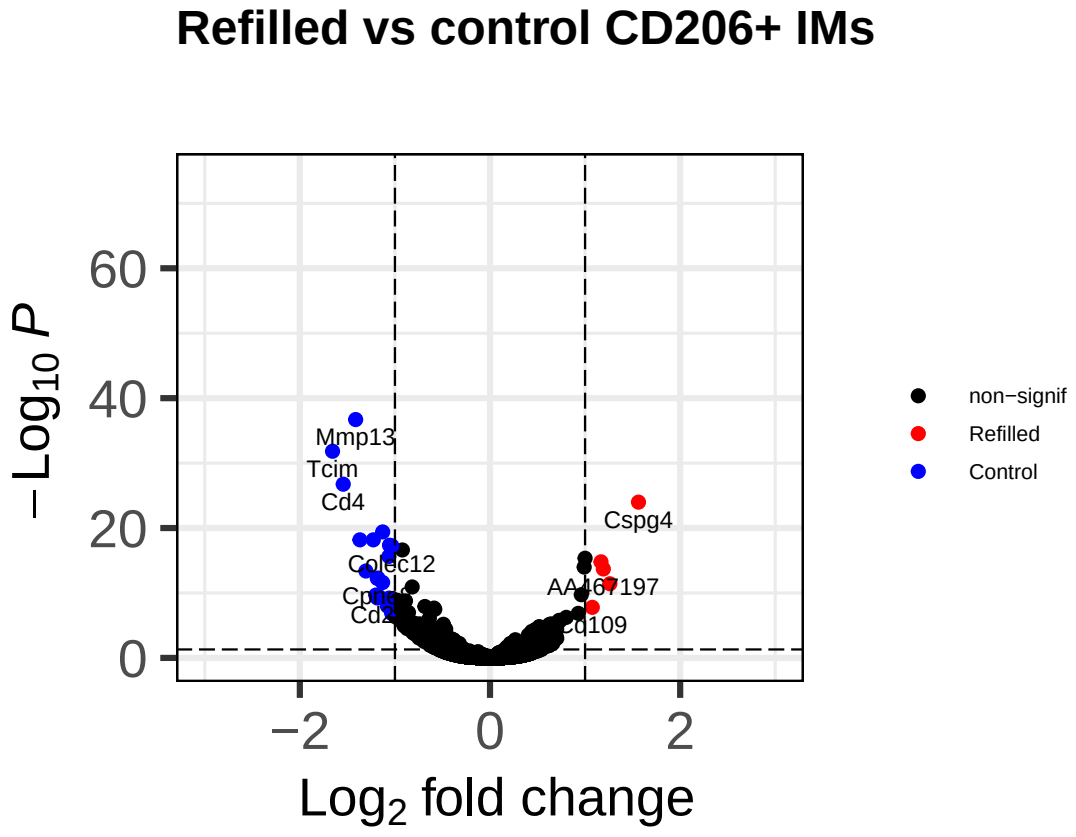
```
## [1] "AA467197" "Cd109"    "Cd226"    "Cspg4"    "Dpysl3"   "Gm21188"  "
    Igf2r"
## [8] "Rarb"     "S100a4"
```

```
# the Refilled-related genes total IMs:
rownames(de.res)[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"
```

7.2 Refilled vs control CD206+ IMs

```
de.res <- refilled_vs_control_CD206pos

de.res$Gene <- rownames(de.res)
keyvals <- rep("black", nrow(de.res))
names(keyvals) <- rep("non-signif", nrow(de.res))

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

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

plot.vol <- EnhancedVolcano(de.res, subtitle = "", lab = rownames(de.res),
    x = "log2FoldChange.y",
    y = "padj.y", xlim = c(-3, 3), ylim = c(0, -log10(1e-74)), labSize =
       3, pCutoff = 0.05,
    FCcutoff = 1, colAlpha = 1, colCustom = keyvals, legendLabSize = 8,
       legendIconSize = 2,
    border = "full", legendPosition = "right", axisLabSize = 20, title = "
       Refilled␣vs␣control␣CD206+␣IMs")

plot.vol
```



What are these genes?

```
# the Refilled-related genes CD206+ IMs:
rownames(de.res)[names(keyvals) == "Refilled"]
```

```
## [1] "AA467197" "Cd109"    "Cd226"    "Cspg4"    "Gm21188"
```

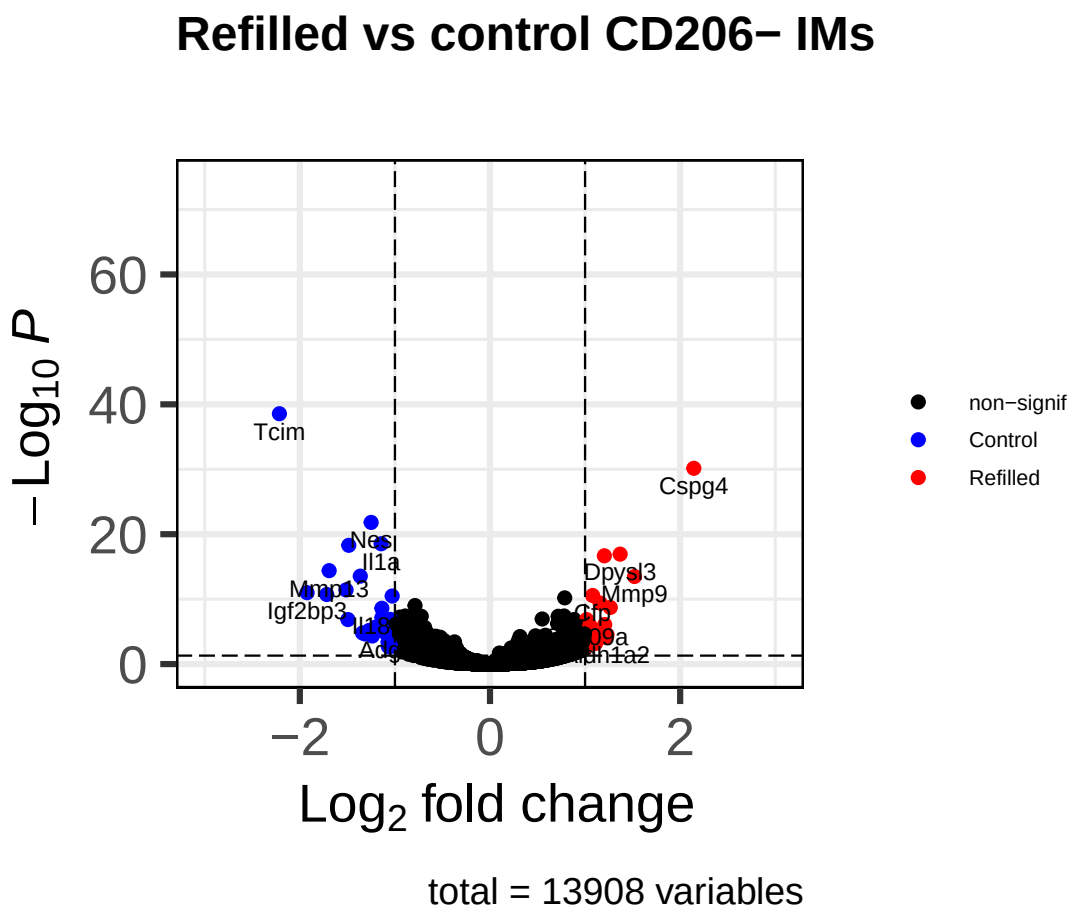
```
# the Refilled-related genes CD206+ IMs:
rownames(de.res)[names(keyvals) == "Control"]
```

```
##  [1] "Cd22"    "Cd4"     "Colec12" "Cpne8"   "Edil3"   "Fscn1"   "H2.M2
    "
##  [8] "Il18r1"  "Lilra5"  "Lzts2"   "Mmp13"   "Nbea"    "Peg10"   "
    Pla2g2d"
## [15] "Ppp1r9a" "Sgce"    "Siglecf" "Sort1"   "Spats2l" "Tcim"    "Tln2"
## [22] "Tmem119" "Vcam1"
```

7.3 Refilled vs control CD206- IMs

```
de.res <- refilled_vs_control_CD206neg

de.res$Gene <- rownames(de.res)
keyvals <- rep("black", nrow(de.res))
names(keyvals) <- rep("non-signif", nrow(de.res))
```

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

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

plot.vol <- EnhancedVolcano(de.res, subtitle = "", lab = rownames(de.res),
    x = "log2FoldChange.y",
    y = "padj.y", xlim = c(-3, 3), ylim = c(0, -log10(1e-74)), labSize =
       3, pCutoff = 0.05,
    FCcutoff = 1, colAlpha = 1, colCustom = keyvals, legendLabSize = 8,
       legendIconSize = 2,
    border = "full", legendPosition = "right", axisLabSize = 20, title = "
       Refilled vs control CD206- IMs")

plot.vol
```




```
# the Refilled-related genes CD206- IMs:
rownames(de.res)[names(keyvals) == "Refilled"]
```

```
##  [1] "Aldh1a2" "Cd109"   "Cd209a"  "Cd59a"   "Cfb"     "Cfp"     "Cspg4
   "
##  [8] "Ctsk"    "Dpysl3"  "Fabp5"   "Fgfr1"   "Hr"      "Igf2r"   "Mmp9"
```

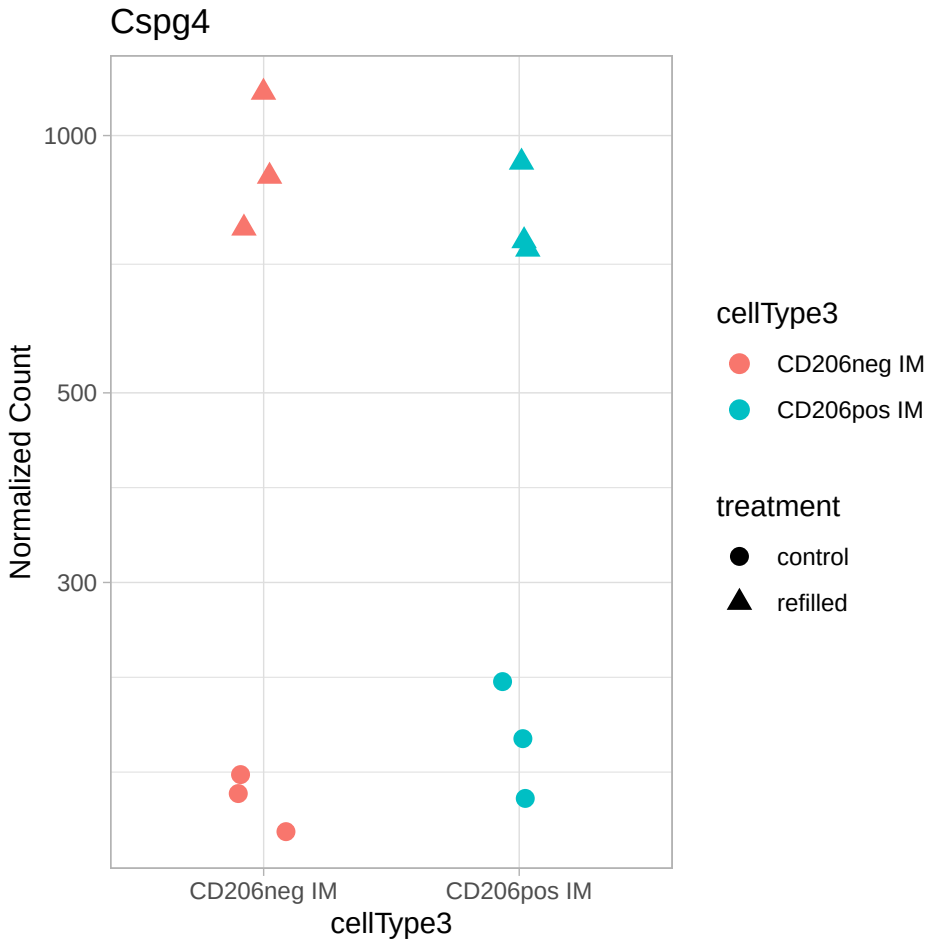
```
## [15] "Phyhd1"  "Retnla"  "S100a4"  "Vldlr"
```

```
# the Refilled-related genes CD206- IMs:
rownames(de.res)[names(keyvals) == "Control"]
```

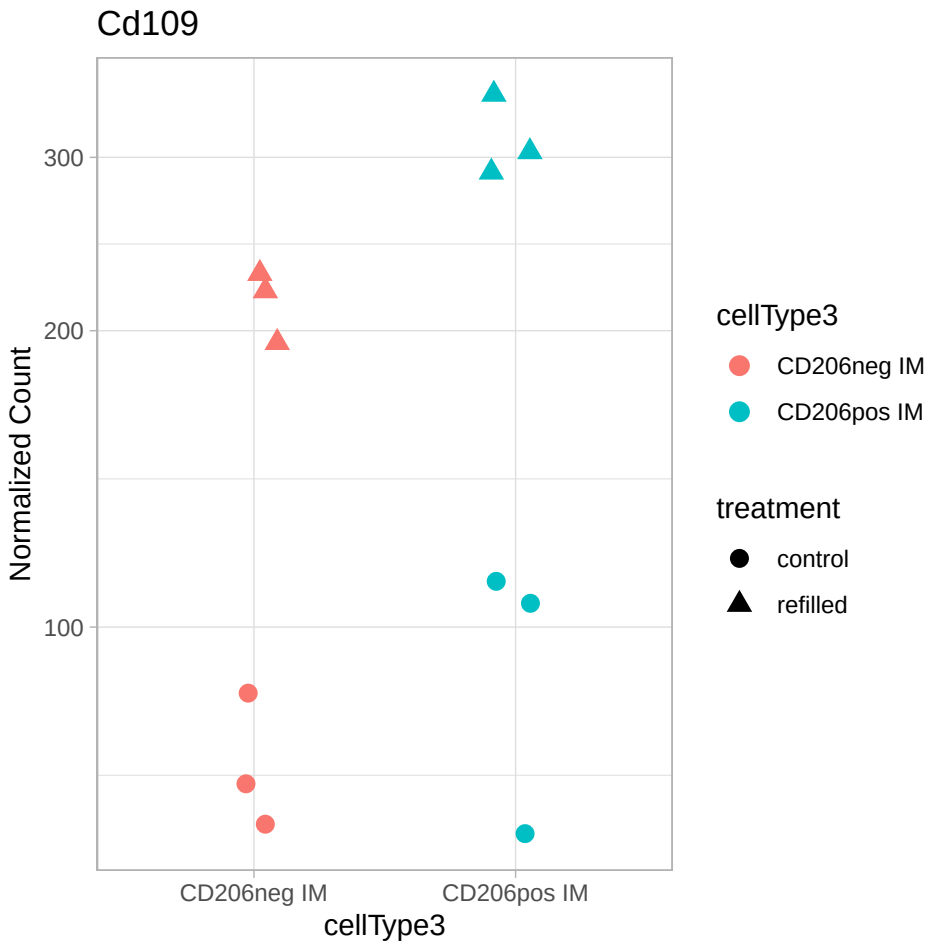
```
##  [1] "Adgrl3"   "Btbd11"   "Capn3"    "Cd4"      "Edil3"    "H2.M2"
##  [7] "Hs3st3a1" "Igf2bp3"  "Il18r1"   "Il1a"     "Ildr2"    "Klhl13"
## [13] "Lilra5"   "Lrrc3"    "Mmp13"    "Nes"      "Oplah"    "P3h2"
## [19] "Peg10"    "Ppp1r9a"  "Rcbtb2"   "Serpinf2" "Tcim"     "Tmem119"
## [25] "Unc13b"   "Zfp827"
```

7.4 Plot the top DE genes

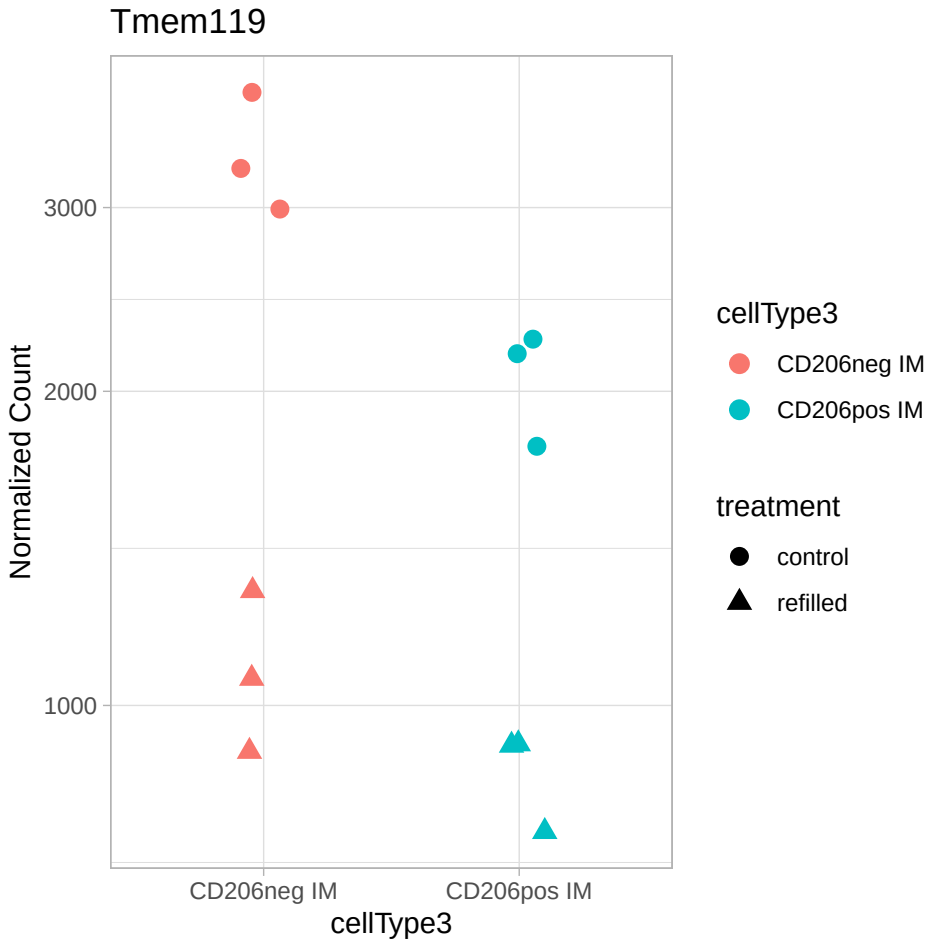
```
# plotCount Cspg4
data <- plotCounts(dds.im, gene = "Cspg4", intgroup = c("cellType3", "
   treatment"),
    returnData = TRUE)

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_0.61.0          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_2.5.1               KernSmooth_2.23-20     vipor_0.4.5
## [10] DBI_1.1.2              colorspace_2.0-3       withr_2.4.3
## [13] tidyselect_1.1.1       ggrastr_1.0.1          ggalt_0.4.0
## [16] bit_4.0.4              compiler_4.0.3         extrafontdb_1.0
## [19] textshaping_0.3.6      cli_3.2.0              formatR_1.11
## [22] DelayedArray_0.16.3    labeling_0.4.2         scales_1.1.1
## [25] proj4_1.0-11           genefilter_1.72.1      systemfonts_1.0.4
## [28] stringr_1.4.0          digest_0.6.29          rmarkdown_2.11
```

```
## [31] XVector_0.30.0          pkgconfig_2.0.3         htmltools_0.5.2         43
## [34] extrafont_0.17          highr_0.9               fastmap_1.1.0           44
## [37] maps_3.4.0              rlang_1.0.1             rstudioapi_0.13         45
## [40] RSQLite_2.2.10          farver_2.1.0            generics_0.1.2          46
## [43] BiocParallel_1.24.1     dplyr_1.0.8             RCurl_1.98-1.6          47
## [46] magrittr_2.0.2          GenomeInfoDbData_1.2.4  Matrix_1.4-0            48
## [49] Rcpp_1.0.8              ggbeeswarm_0.6.0        munsell_0.5.0           49
## [52] fansi_1.0.2             lifecycle_1.0.1         stringi_1.7.6           50
## [55] yaml_2.3.5              MASS_7.3-53             zlibbioc_1.36.0         51
## [58] grid_4.0.3              blob_1.2.2              crayon_1.5.0            52
## [61] lattice_0.20-41         splines_4.0.3           annotate_1.68.0         53
## [64] locfit_1.5-9.4          knitr_1.37              pillar_1.7.0            54
## [67] geneplotter_1.68.0      XML_3.99-0.8            glue_1.6.1              55
## [70] evaluate_0.15           vctrs_0.3.8             Rttf2pt1_1.3.10         56
## [73] gtable_0.3.0            purrr_0.3.4             assertthat_0.2.1        57
## [76] cachem_1.0.6            xfun_0.29               xtable_1.8-4            58
## [79] ragg_1.2.2              survival_3.2-7          tibble_3.1.6            59
## [82] beeswarm_0.4.0          memoise_2.0.1           ellipsis_0.3.2          60
```

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).