Preprocessing

Vlatka Antolovic

Dependencies

```
library(scran) # needs to be run in R version 4 or higher
```

Info

- 'data' (original read counts)
- 'rawData' (outlier cells and outdated gene models removed)
- 'normData' (normalised with size factor from 'scran' package)

Set working directory (full path of the 'scRNAseq_Pipeline' folder).

```
setwd("C:/Users/Vlatka/Documents/scRNAseq_Pipeline")
```

Import functions

```
source('Functions/Fun_umi_mapped_Plots.R')
```

1 Data import

Import samples. Customize cell names. Merge replicates.

```
data1 <- read.csv('./Data/Sample_1.csv', row.names = 1) #13865 x 2925
numCells1 <- dim(data1)[2]
names(data1) = paste0(rep('rep1_', numCells1), 1:numCells1)

data2 <- read.csv('./Data/Sample_2.csv', row.names = 1)
numCells2 <- dim(data2)[2]
names(data2) = paste0(rep('rep2_', numCells2), 1:numCells2) #13865 x 2404

data <- cbind(data1, data2) #13865 x 5329
```

Indexing data by replicates. Not needed in this case, but 'cell_ind' is currently still used as such in the 'umiPlot' function.

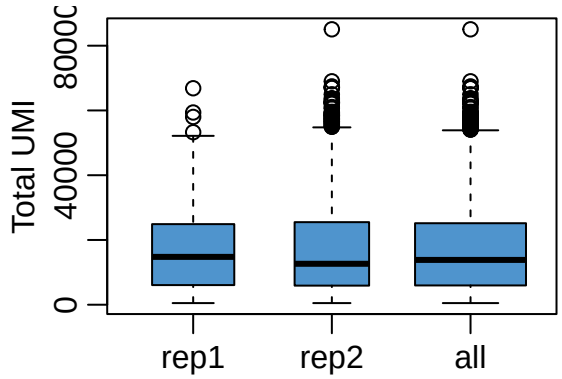
```
cell_ind <- list()
for (i in 1:2) {
  cell_ind[[i]] <- grep(paste0('^rep',i), names(data))
}
```

2 Remove outlier cells

2.1 Total UMI counts per cell

Box plot of UMI counts per cell. Replicate 1 was sampled further into the bacterial lawn, i.e. it contains more feeding cells than replicate 2. That could account for the higher median UMI counts in the 1st replicate (undifferentiated cells are known to have higher total transcript count). On the same note, as replicate 2 contains more differentiated (aggregated; stickier) cells, that could account for more outliers in 2nd replicate (doublet cells).

```
totalumi = colSums(data)
totalumi_1 = colSums(data1)
totalumi_2 = colSums(data2)

# Boxplot
par(mgp = c(1.7, .6, 0), mar = c(2, 3, .3, .3))
boxplot(totalumi_1, totalumi_2, totalumi,
        names = c("rep1","rep2", "all"), ylab = "Total UMI",
        varwidth = T, col = "steelblue3")
```



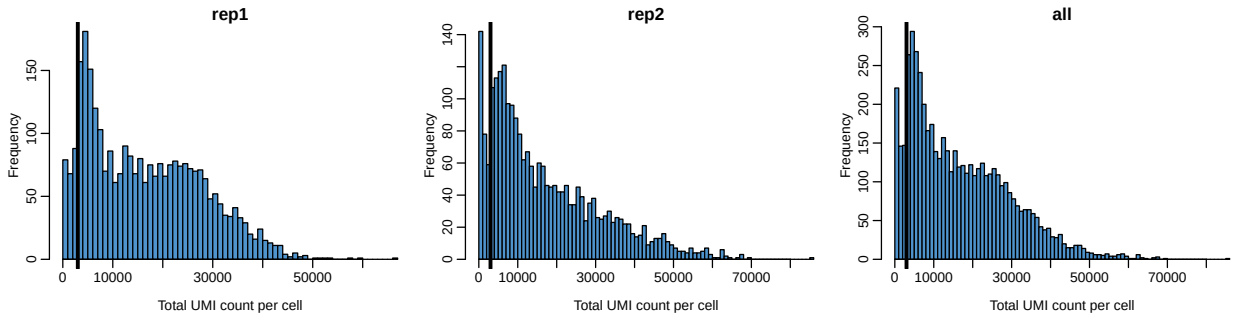
2.1a High total UMI outliers Decide on a cut-off value. Since the two biological replicates were loaded on the same chip (different inputs) and all the reagents were the same, we decided on the common threshold for both replicates.

```
# Cut-off (outliers)
upperCutoff <- quantile(totalumi, 0.75) + 1.5 * IQR(totalumi)

# Number of outliers (common threshold for both reps)
#sum(totalumi_1 > upperCutoff) #3
#sum(totalumi_2 > upperCutoff) #47
```

2.1b Low total UMI outliers Histogram of UMI counts per cell. The lower cut-off value is marked with vertical line.

```
# Cut-off
lowCutoff = 3000 # based on visual inspection

# Number of outliers
#sum(totalumi_1 < lowCutoff) #234
```

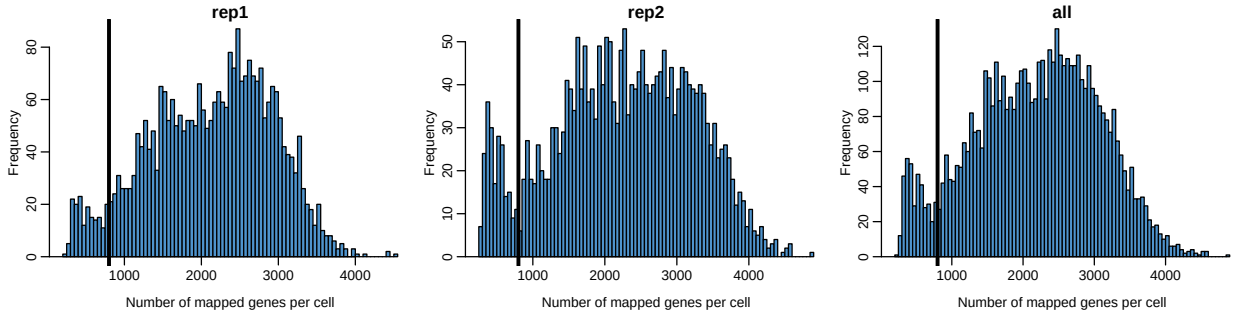
```
#sum(totalumi_2 < lowCutoff) #279

# HISTOGRAM
par(mfrow = c(1,3))
histumi(totalumi_1, 'rep1', lowCutoff)
histumi(totalumi_2, 'rep2', lowCutoff)
histumi(totalumi, 'all', lowCutoff)
```



2.2 Number of mapped genes per cell

Histogram of the number of mapped genes per cell. Lower cut-off value is marked with vertical line.

```
mapped = colSums(data > 0)
mapped1 = colSums(data1 > 0)
mapped2 = colSums(data2 > 0)

#---------LOW OUTLIERS (number of mapped genes)-----------
# Cut-off
cutoff = 800 # based on visual inspection

# Number of outliers
#sum(mapped1 < cutoff) #176
#sum(mapped2 < cutoff) #217

# HISTOGRAM
par(mfrow = c(1,3))
histm(mapped1, 'rep1', cutoff)
histm(mapped2, 'rep2', cutoff)
histm(mapped, 'all', cutoff)
```

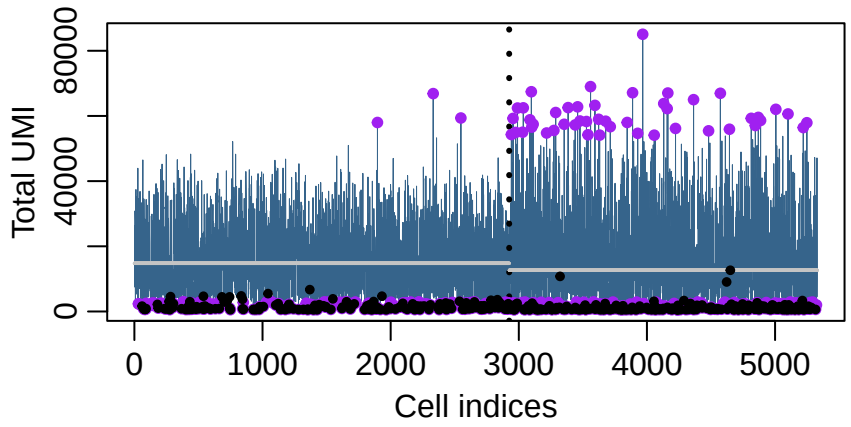


2.3 Removing the outlier cells

```
out1 <- totalumi > upperCutoff; #sum(out1) # high_UMI: 50
out2 <- totalumi < lowCutoff; #sum(out2) # low_UMI: 513
out3 <- mapped < cutoff; #sum(out3) # low_mapped: 393

out <- out1 | out2 | out3
#sum(out) # 586, a lot of 'out2' and 'out3' overlap

# Removing the outlier cells
rawData0 <- data; rawData0[,out] <- NULL # 13865 x 4743
```

Plot outliers Plot total UMI counts per cell. Dotted vertical line marks separates replicates. Gray horizontal line marks the median values for replicate 1 and 2, respectively.

```
umiPlot(totalumi, cell_ind)
points(which(out1==T), totalumi[out1], pch=20, cex=1, col='purple') # high total UMIs
points(which(out2==T), totalumi[out2], pch=20, cex=1, col='purple') # low total UMIs
points(which(out3==T), totalumi[out3], pch=20, cex=.8, col='black') # low num. of mapped genes
```

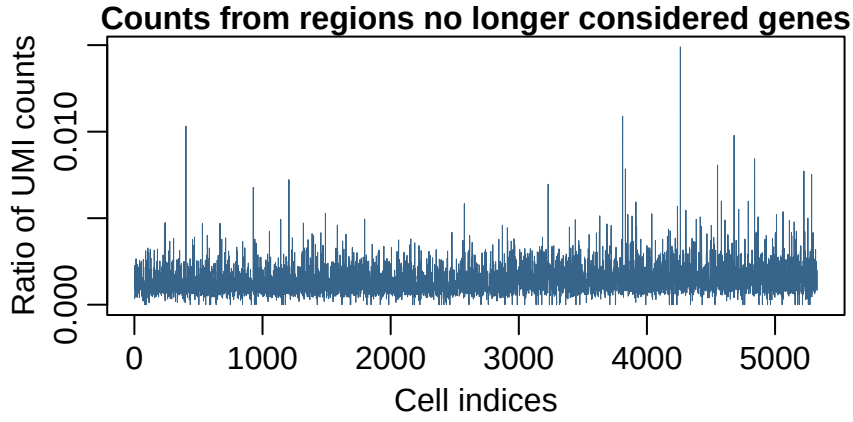


3 Removing outdated gene models

Removing gene models not considered genes anymore (removed from the gene list on dictybase.org)

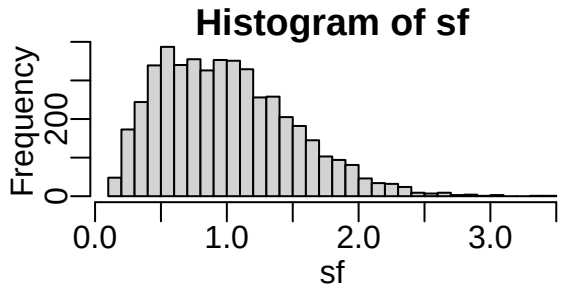
```
# List of gene models currently present in 'dictyBase'
info <- read.csv('./Data/geneInfo.csv', row.names = 1)

# Keeping only those genes
rawData <- rawData0[rownames(info),] # 13231 x 4743

# Check: Plot percentage of reads coming from genes removed from dictyBase
# ~ 0.1 %
genes_out <- setdiff(rownames(data), rownames(info)) # 634 genes
perc_out <- colSums(data[genes_out,]) / colSums(data)
# median(perc_out) # 0.001241188
```

```
par(mgp = c(1.7, .6, 0), mar = c(3, 3, 1, .3))
plot(perc_out, main = "Counts from regions no longer considered genes",
     xlab = "Cell indices", ylab = "Ratio of UMI counts", type = "l",
     col = "steelblue4", lwd = .5, cex.main = 1,)
```



4 Normalization

```
# Size factors from 'scran' package (pooling and deconvolution)
# A: Pre-clustering based on expression profiles
sfCluster <- quickCluster(rawData)
# B: Deconvolution
sf <- sizeFactors(computeSumFactors(SingleCellExperiment(list(counts = rawData)),
                                                clusters = sfCluster))
# Check the distribution of size factors
par(mgp = c(1.3, .4, 0), mar = c(2.3, 2.3, 1, .1))
hist(sf, 30)
```




```
#---------Normalize read counts-----------
normData = t(t(rawData) / sf)

#---------EXPORT-----------
# outfolder <- './Data/'
# write.csv(normData, paste0(outfolder, 'norm_counts.csv'))
```

Session information

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] scran_1.22.1                scuttle_1.4.0
##  [3] SingleCellExperiment_1.16.0 SummarizedExperiment_1.24.0
##  [5] Biobase_2.54.0              GenomicRanges_1.46.1
##  [7] GenomeInfoDb_1.30.1         IRanges_2.28.0
##  [9] S4Vectors_0.32.4            BiocGenerics_0.40.0
## [11] MatrixGenerics_1.6.0        matrixStats_0.61.0
##
## loaded via a namespace (and not attached):
##  [1] statmod_1.4.36            locfit_1.5-9.5
##  [3] xfun_0.35                 beachmat_2.10.0
##  [5] BiocSingular_1.10.0       lattice_0.20-44
##  [7] htmltools_0.5.4           yaml_2.3.5
##  [9] rlang_1.0.6               BiocParallel_1.28.3
## [11] dqrng_0.3.0               GenomeInfoDbData_1.2.7
## [13] stringr_1.4.0             zlibbioc_1.40.0
## [15] rsvd_1.0.5                ScaledMatrix_1.2.0
## [17] codetools_0.2-18          evaluate_0.15
## [19] knitr_1.41                fastmap_1.1.0
## [21] irlba_2.3.5               parallel_4.1.1
## [23] highr_0.9                 BiocNeighbors_1.12.0
## [25] Rcpp_1.0.8.3              edgeR_3.36.0
## [27] limma_3.50.3              DelayedArray_0.20.0
## [29] XVector_0.34.0            digest_0.6.29
## [31] metapod_1.2.0             stringi_1.7.6
## [33] bluster_1.4.0             grid_4.1.1
## [35] cli_3.1.1                 tools_4.1.1
## [37] bitops_1.0-7              magrittr_2.0.2
## [39] RCurl_1.98-1.6            cluster_2.1.2
## [41] pkgconfig_2.0.3           Matrix_1.3-4
## [43] DelayedMatrixStats_1.16.0 sparseMatrixStats_1.6.0
## [45] rmarkdown_2.18            rstudioapi_0.13
```

```
## [47] igraph_1.3.1              compiler_4.1.1
```

```
Sys.Date()
```

```
## [1] "2023-09-07"
```