Supplementary code for: A BASiCS workflow for expression variability analysis using scRNA-Seq data

Alan O'Callaghan Nils Eling John C. Marioni Catalina A. Vallejos

Contents

1 Introduction

This file provides the code required to pre-process the scRNAseq datasets analysed in the F1000Research BASiCS Workflow. In particular, we provide the code used to perform quality control for the presomitic and somitic mesoderm cells extracted from Ibarra-Soria et al. (2018).

First, we load all necessary libraries:

```r
library("SingleCellExperiment")
library("scater")
library("scran")
library("BASiCS")
library("ggplot2")
library("ggpointdensity")
library("biomaRt")
library("patchwork")
library("viridis")
library("readxl")
## set default theme for plots
theme_set(theme_bw())
```

2 Download and load the data

The full dataset, described by Ibarra-Soria et al. (2018), is stored under the accession number E-MTAB-6153 on ArrayExpress and can be obtained from here. This workflows uses the *processed* data which consists of 2 files:

- `rawCounts.tsv`: matrix of raw expression counts (UMI-based) for all genes and cells.
- `cellAnnotation.tsv`: cell type labels identified by (Ibarra-Soria et al. 2018).

These will be downloaded into a *downloads* folder.

First, we download and load the `rawCounts.tsv` file (832.9 MB in size).

```r
options(timeout = 10000)

# Check whether the 'downloads' directory exists. If not, create it.
if (!dir.exists("downloads")) {
  dir.create("downloads", showWarnings = FALSE)
}

# Check whether the data was downloaded already
if (!file.exists("downloads/rawCounts.tsv")) {
  website <- "https://www.ebi.ac.uk/"
  folder <- "biostudies/files/E-MTAB-6153/"
  file <- "rawCounts.tsv"
  download.file(
    paste0(website, folder, file),
    destfile = "downloads/rawCounts.tsv"
  )
}
```

```r
# Load the downloaded file (this will take some time!)
raw_counts <- read.delim("downloads/rawCounts.tsv", header = TRUE)
```

These data consist of 20809 genes and 20819 cells, i.e. this is the raw data prior to the QC steps applied by Ibarra-Soria et al. (2018).

Prior to using BASiCS, it is important to perform quality control to remove poor quality cells and to control for potential heterogeneous substructure within the population of cells under study. In this case, (Ibarra-Soria et al. 2018) has already characterised population structure using clustering techniques. Here, we will use such information in order to define the cell populations to be use as the input for BASiCS.

The cluster labels identified in the original publication are then loaded (`cellAnnotation.tsv` file; 883 KB in size):

```r
# Check whether the data was downloaded already
if (!file.exists("downloads/cellAnnotation.tsv")) {
  website <- "https://www.ebi.ac.uk/"
  folder <- "biostudies/files/E-MTAB-6153/"
  file <- "cellAnnotation.tsv"
  download.file(
    paste0(website, folder, file),
    destfile = "downloads/cellAnnotation.tsv"
  )
}
# Load the downloaded file
cluster_labels <- read.table("downloads/cellAnnotation.tsv",
  sep = "\t", header = TRUE, stringsAsFactors = FALSE)
```

The table above contains annotation information for all 20819 cells.

The data contains information about the three samples from which the cells were extracted. This information can help us to explore potential batch effects.

```
cluster_labels$sample <- as.factor(cluster_labels$sample)
table(cluster_labels$sample)
```

```
##
##  1.1  1.2    2
## 7731 8341 4747
```

The authors integrated the data from these three samples using the `cellranger aggr` program, which uses downsampling to generate samples with similar depth.

Furthermore, the data contains the 20 major cell types identified by the authors. The number of cells per cluster is displayed below:

```
table(cluster_labels$cellType, useNA = "ifany")
```

```
##
##                 amnion                  blood                cardiac
##                    770                   2079                    700
##            endothelial extraembryonicEctoderm extraembryonicEndoderm
##                    872                    367                   1876
## extraembryonicMesoderm              forebrain                foregut
##                   1017                    764                    185
##    mesodermProgenitors            midHindbrain             midHindgut
##                   1696                   1242                    253
##          mixedMesoderm            neuralCrest             neuralTube
##                   2043                    332                   1128
##              notochord    pharyngealMesoderm               placodes
##                     85                    743                   1345
##     presomiticMesoderm        somiticMesoderm                   <NA>
##                   1150                    739                   1433
```

Note that `cellType` is missing for 1433 cells. The latter correspond to cells discarded during QC by (Ibarra-Soria et al. 2018). Such cells are labelled with a `badQuality` tag in `cluster_label$cell`. The QC used by the authors consisted of filtering out cells for which less than 1,000 genes were detected or that had more than 3% of reads mapped to mitochondrial genes. They also removed potential doublets defined as cells for which expression of *Xist* and any of *Kdm5d*, *Eif2s3y*, *Gm29650*, *Uty* or *Ddx3y* (genes located in the Y chromosome) was detected. 19386 were used by the authors in their analysis after applying this exclusion criteria.

Note that the authors also identified some sub-structure within these main clusters. For example, they identified two sub-types of presomitic mesoderm cells:

```
# Extract sub-type cell labels (sample sub-index is removed)
cluster_labels$subCellType <- sub("_.*", "", cluster_labels$cell)
with(cluster_labels[cluster_labels$cellType == "presomiticMesoderm", ],
     table(subCellType, cellType))
```

```
##                        cellType
## subCellType             presomiticMesoderm
##   presomiticMesoderm.a                 875
##   presomiticMesoderm.b                 275
```

3 Select populations of interest

We selected the somitic ($n = $ 739) and pre-somitic mesoderm ($n = $ 1150) cells. This is an arbitrary choice for illustration purposes only. The steps of this workflow could be applied to any other pair of cell types.

```
# Identifies which cells are in the groups of interest
ind_som <- which(
  cluster_labels$cellType == "somiticMesoderm" |
  cluster_labels$cellType == "presomiticMesoderm"
)
# Selects the cells of interest from the raw counts and cluster-label metadata
raw_counts <- raw_counts[, ind_som]
cluster_labels <- cluster_labels[ind_som, ]
```

After selecting these cells, the distribution of the number of cells across samples is displayed below:

```
table(cluster_labels$cellType, cluster_labels$sample)
```

```
##
##                       1.1 1.2   2
##   presomiticMesoderm 452 455 243
##   somiticMesoderm    290 289 160
```

4 Generating a SingleCellExperiment object

For pre-processing and visualisation purposes, we load the data into a `SingleCellExperiment` object. The metadata will be stored in the `colData` slot.

```
droplet_sce <- SingleCellExperiment(
  assays = list(counts = as(as.matrix(raw_counts), "dgCMatrix")),
  colData = cluster_labels
)
# Delete the original table of raw expression counts as no longer needed
rm(raw_counts)
```

5 Gene annotation using BioMart

Input data was annotated using Ensembl gene identifiers. To facilitate interpretation, it is often useful to obtain a mapping from Ensembl gene IDs to gene symbols using the BioMart suite (http://www.biomart.org) via the Bioconductor package *biomaRt* (Durinck et al. 2009). This can also be used to obtain gene-pathways mappings and other metadata (e.g. gene length), useful for performing functional analysis of gene sets identified in downstream analyses.

```r
# Check if data storage folder exists and create it if needed
if (!dir.exists("rds")) {
  dir.create("rds", showWarnings = FALSE)
}
if (!file.exists("rds/genenames.Rds")) {
  # Initialize mart and dataset
  ensembl <- useEnsembl(
    biomart = "genes",
    version = 104,
    dataset = "mmusculus_gene_ensembl"
  )
  # Select gene ID and gene name
  genenames <- getBM(
    attributes = c("ensembl_gene_id", "external_gene_name", "gene_biotype",
                   "chromosome_name", "description"),
    mart = ensembl
  )
  rownames(genenames) <- genenames$ensembl_gene_id
  # Store information for future use
  saveRDS(genenames, "rds/genenames.Rds")
}
# Load pre-downloaded gene annotation information
genenames <- readRDS("rds/genenames.Rds")
```

We add this information as `rowData` within the `SingleCellExperiment` object created above.

```r
## Merge biomaRt annotation
rowdata <- data.frame(ensembl_gene_id = rownames(droplet_sce))
rowdata <- merge(rowdata, genenames, by = "ensembl_gene_id", all.x = TRUE)
rownames(rowdata) <- rowdata$ensembl_gene_id
# Sort to match the order in the original data
rowdata <- rowdata[rownames(droplet_sce),]
## Check if  order is correct after merge;
stopifnot(all(rownames(rowdata) == rownames(droplet_sce)))
## add to the SingleCellExperiment object
rowData(droplet_sce) <- rowdata
```

For the remaining analysis, we will only focus on the 17,500 protein coding genes that are contained in the data. These are selected below.

```r
ind_pc <- which(rowData(droplet_sce)$gene_biotype == "protein_coding")
droplet_sce <- droplet_sce[ind_pc, ]
```

6 QC and exploratory data analysis

In order to perform quality control (QC), we calculate some cell-specific summary statistics: the total number of counts per cell (`sum`) and the total number of detected genes per cell (`detected`). These will be added in the `colData` slot using the `addPerCellQC()` function from the *scater* package. Note that additional QC metrics are often calculated (e.g. the percentage of reads mapped to mitochondrial reads). These are not calculated here as the authors already applied relevant QC.

```
droplet_sce <- addPerCellQC(droplet_sce)
```

The distribution of these metrics per each cluster can be visualised as follows:

```
p_cell_qc1 <- plotColData(droplet_sce, y = "sum", x = "subCellType") +
  geom_hline(yintercept = 25000, lty = 2, col = "red") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
p_cell_qc2 <- plotColData(droplet_sce, y = "detected", x = "subCellType") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
p_cell_qc3 <- plotColData(droplet_sce, y = "sum", x = "sample") +
  geom_hline(yintercept = 25000, lty = 2, col = "red")
p_cell_qc4 <- plotColData(droplet_sce, y = "detected", x = "sample")

(p_cell_qc1 + p_cell_qc2) / (p_cell_qc3 + p_cell_qc4)
```

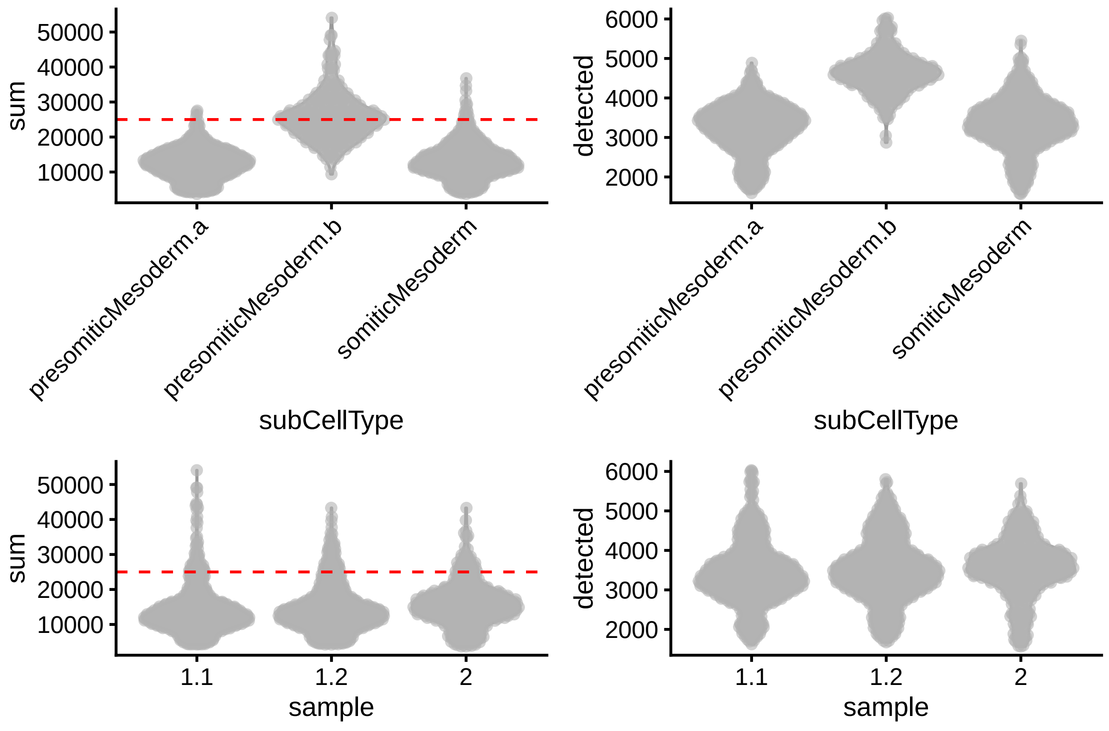

Figure 1: Cell-level QC metrics stratified by cell sub-type and sample (before QC). For each cell cluster: distribution of the total number of UMIs per cell (left) and the total number of detected genes per cell (right).

As seen in Figure 1, the `presomiticMesoderm.b` sub-cluster is characterised by a substantially larger number of UMIs and detected genes. These cells have been highlighted by the authors as potential doublets.

Therefore, as in the original publication, the entire sub-cluster will be removed from subsequent analyses. For this workflow, we also apply an additional exclusion criteria to remove cells with more 25,000 UMIs as potential doublets.

```
ind_retain <- colData(droplet_sce)$subCellType != "presomiticMesoderm.b" &
  colData(droplet_sce)$sum <= 25000
droplet_sce <- droplet_sce[, ind_retain]
```

After applying this filter, the dataset contains 1595 cells: 871 in the presomitic Mesoderm group and 724 in the somitic Mesoderm group.

The distribution of the total number of UMIs and detected genes after QC is shown in Figure 2.

```
p_cell_qc1 <- plotColData(droplet_sce, y = "sum", x = "subCellType")  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
p_cell_qc2 <- plotColData(droplet_sce, y = "detected", x = "subCellType") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
p_cell_qc3 <- plotColData(droplet_sce, y = "sum", x = "sample") +
  geom_hline(yintercept = 25000, lty = 2, col = "red")
p_cell_qc4 <- plotColData(droplet_sce, y = "detected", x = "sample")

(p_cell_qc1 + p_cell_qc2) / (p_cell_qc3 + p_cell_qc4)
```

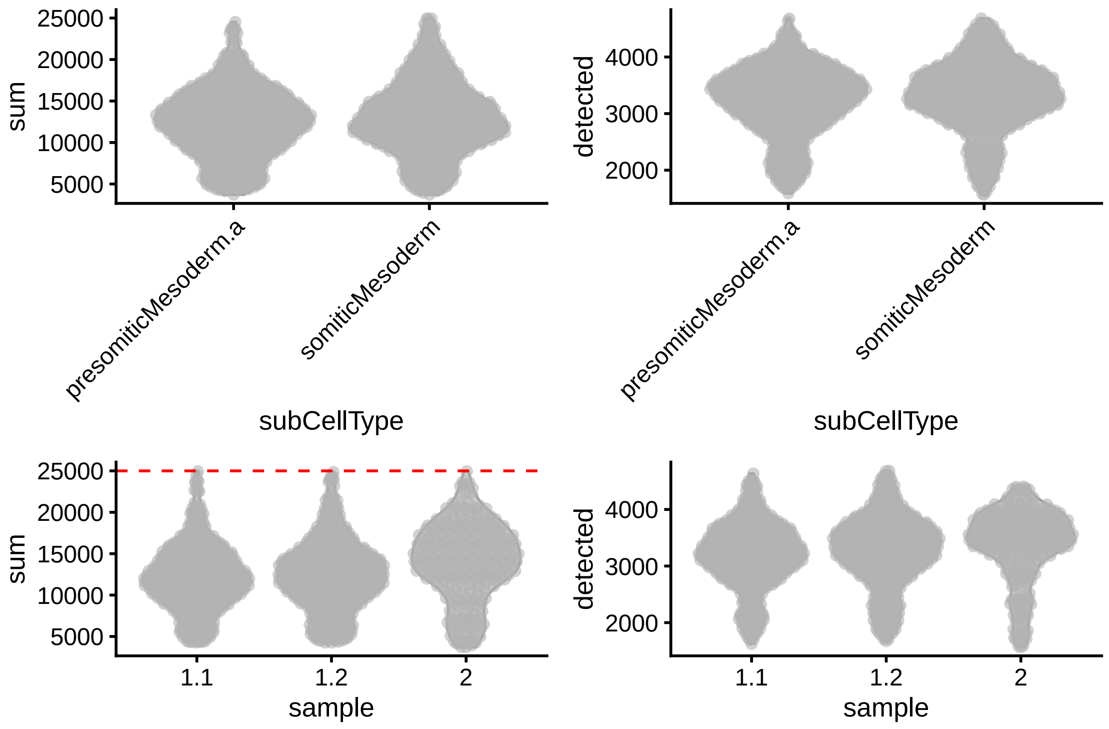

Figure 2: Cell-level QC metrics stratified by cell sub-type and sample (after QC). For each cell cluster: distribution of the total number of UMIs per cell (left) and the total number of detected genes per cell (right).

To further explore the underlying structure of the data, we perform global scaling normalisation using *scran* and principal component analysis (PCA) of log-transformed normalised expression counts using *scater*.

As seen in Figure 3, this analysis suggests the absence of strong batch effects. It should be noted that the estimation of global scaling normalisation factors using *scran* is not strictly necessary in the *BASiCS* workflow. Here, we only use it as part of the exploratory data analysis. Moreover, count-based models for dimensionality reduction (e.g. (Townes et al. 2019; Lopez et al. 2018)) could be used as an alternative to PCA, removing the need for log normalisation.

```r
## Global scaling normalisation
droplet_sce <- computeSumFactors(droplet_sce, colData(droplet_sce)$subCellType)
## Obtain log-normalised expression counts
droplet_sce <- logNormCounts(droplet_sce)
## Run PCA
droplet_sce <- runPCA(droplet_sce)
## Plot first two PCs colour-coded according to cell-level metadata
p_type <- plotPCA(
    droplet_sce,
    ncomponents = c(2, 1),
    colour_by = "subCellType"
  ) +
  theme(legend.position = "bottom")
p_detected <- plotPCA(
    droplet_sce,
    ncomponents = c(2, 1),
    colour_by = "detected"
  ) +
  theme(legend.position = "bottom")
p_batch <- plotPCA(
    droplet_sce,
    ncomponents = c(2, 1),
    colour_by = "sample"
  ) +
  theme(legend.position = "bottom")
p_type + p_detected  + p_batch
```

As seen inn Figure 3, the first principal component (which explains approximately 9% of the total variability) captures the presomitic versus somitic Mesoderm structure of the data. The second principal component (which explains approximately 3% of the total variability) is correlated with the number of detected features per cell. We do not observe substantial batch effects driven by the animal of origin from which cells were collected (although some minor differences are seen, due to differences in the number of detected genes).

In addition to cell-specific QC, we also recommend a gene filtering step prior to using *BASiCS*. The purpose of this filter is to remove lowly expressed genes that were largely undetected through sequencing, making reliable variability estimates difficult to obtain. Here, we remove genes that are not detected in at least 20 cells across all cells. This is to ensure a reliable estimate of variability, roughly in line with the sample size requirements for the negative binomial distribution outlined in Lloyd-Smith (2007).

```r
## Calculate gene-level QC metrics using scater
droplet_sce <- addPerFeatureQC(droplet_sce, exprs_values = "counts")
## Remove genes with zero total counts across all cells
droplet_sce <- droplet_sce[rowData(droplet_sce)$detected != 0, ]

## Transform "detected" into number of cells and define inclusion criteria
rowData(droplet_sce)$detected_cells <-
  rowData(droplet_sce)$detected * ncol(droplet_sce) / 100
```

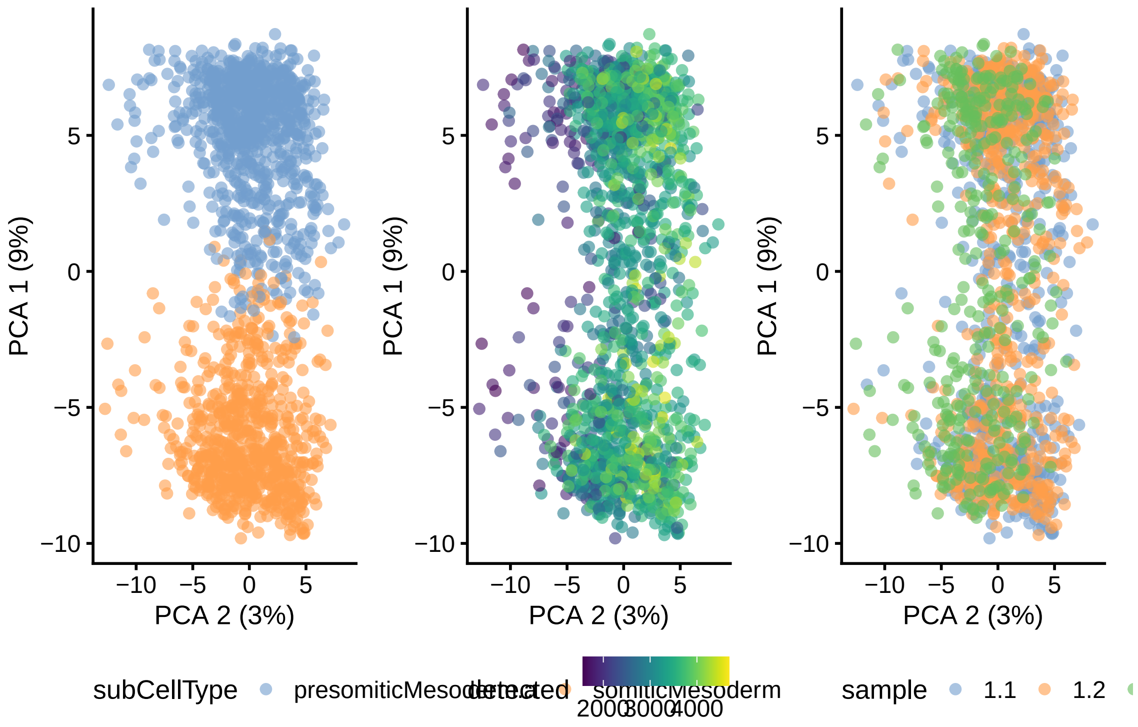

Figure 3: First two principal components of log-transformed expression counts after scran normalisation. Colour indicates the experimental condition (left), the number of detected genes (middle) and animal of origin (right) for each cell.

```r
## Set detection threshold to detecting expression in at least 20 cells
detected_threshold <- 20
rowData(droplet_sce)$include_gene <-
  rowData(droplet_sce)$detected_cells >= detected_threshold

## Plot gene-level metrics, highlighting those included for our analysis
plotRowData(droplet_sce,
    x = "detected_cells",
    y = "mean",
    colour_by = "include_gene"
  ) +
  xlab("Number of cells in which expression was detected") +
  ylab("Average number of read counts across all cells") +
  scale_x_log10() +
  scale_y_log10() +
  theme(legend.position = "bottom") +
  geom_vline(
    xintercept = detected_threshold,
    linetype = "dashed",
    col = "grey60"
  )
```

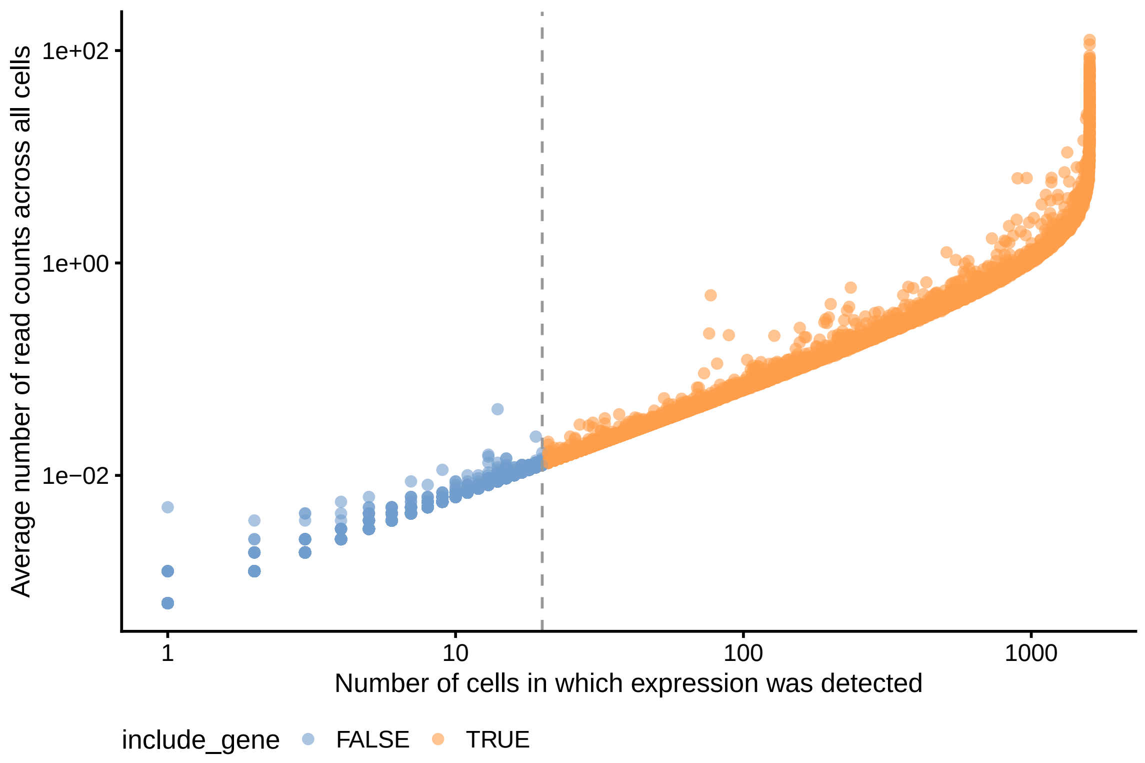

Figure 4: Average UMI-count for each gene is plotted against the number of cells in which that gene was detected. Dashed grey lines are shown at the thresholds below which genes are removed.

```r
## Apply gene filter
droplet_sce <- droplet_sce[rowData(droplet_sce)$include_gene, ]
```

The final dataset used in subsequent analyses contains 1595 cells and 10862 genes. As the analysis will be carried out separately for pre-somitic and somitic mesoderm cells, we create separate `SingleCellExperiment` object for each group of cells. These datasets contain 871 and 724, respectively.

```
sce_psm <- droplet_sce[, colData(droplet_sce)$cellType == "presomiticMesoderm"]
sce_sm <- droplet_sce[, colData(droplet_sce)$cellType == "somiticMesoderm"]
```

These are then stored for their use in the main manuscript.

```
saveRDS(sce_psm, "rds/sce_psm.Rds")
saveRDS(sce_sm, "rds/sce_sm.Rds")
```

7 Session information

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.3 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/libopenblasp-r0.3.20.so;  LAPACK version 3.10.0
##
## 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
##
## time zone: Etc/UTC
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] readxl_1.4.3                viridis_0.6.4
##  [3] viridisLite_0.4.2           patchwork_1.1.3
##  [5] biomaRt_2.58.0              ggpointdensity_0.1.0
##  [7] BASiCS_2.14.0               scran_1.30.0
##  [9] scater_1.30.0               ggplot2_3.4.4
## [11] scuttle_1.12.0              SingleCellExperiment_1.24.0
## [13] SummarizedExperiment_1.32.0 Biobase_2.62.0
## [15] GenomicRanges_1.54.1        GenomeInfoDb_1.38.1
## [17] IRanges_2.36.0              S4Vectors_0.40.1
## [19] BiocGenerics_0.48.1         MatrixGenerics_1.14.0
## [21] matrixStats_1.1.0           knitr_1.45
## [23] BiocStyle_2.30.0
##
## loaded via a namespace (and not attached):
##   [1] tensorA_0.36.2            magrittr_2.0.3
##   [3] ggbeeswarm_0.7.2          farver_2.1.1
##   [5] rmarkdown_2.25            zlibbioc_1.48.0
##   [7] vctrs_0.6.4               memoise_2.0.1
##   [9] DelayedMatrixStats_1.24.0 RCurl_1.98-1.13
##  [11] htmltools_0.5.6.1         S4Arrays_1.2.0
##  [13] progress_1.2.2            curl_5.1.0
##  [15] distributional_0.3.2      BiocNeighbors_1.20.0
##  [17] cellranger_1.1.0          SparseArray_1.2.2
##  [19] cachem_1.0.8              igraph_1.5.1
##  [21] mime_0.12                 lifecycle_1.0.3
##  [23] pkgconfig_2.0.3           rsvd_1.0.5
##  [25] Matrix_1.6-1.1            R6_2.5.1
##  [27] fastmap_1.1.1             GenomeInfoDbData_1.2.11
##  [29] shiny_1.7.5.1             digest_0.6.33
##  [31] colorspace_2.1-0          AnnotationDbi_1.64.1
##  [33] dqrng_0.3.1               irlba_2.3.5.1
##  [35] RSQLite_2.3.3             beachmat_2.18.0
##  [37] labeling_0.4.3            filelock_1.0.2
##  [39] fansi_1.0.5               httr_1.4.7
##  [41] abind_1.4-5               compiler_4.3.2
##  [43] bit64_4.0.5               withr_2.5.1
##  [45] backports_1.4.1           BiocParallel_1.36.0
##  [47] DBI_1.1.3                 highr_0.10
##  [49] hexbin_1.28.3             MASS_7.3-60
##  [51] rappdirs_0.3.3            DelayedArray_0.28.0
##  [53] bluster_1.12.0            tools_4.3.2
##  [55] vipor_0.4.5               beeswarm_0.4.0
##  [57] httpuv_1.6.12             glue_1.6.2
##  [59] promises_1.2.1            grid_4.3.2
##  [61] checkmate_2.3.0           cluster_2.1.4
##  [63] generics_0.1.3            gtable_0.3.4
##  [65] hms_1.1.3                 xml2_1.3.5
##  [67] BiocSingular_1.18.0       ScaledMatrix_1.10.0
##  [69] metapod_1.10.0            utf8_1.2.4
##  [71] XVector_0.42.0            stringr_1.5.0
##  [73] ggrepel_0.9.4             pillar_1.9.0
##  [75] ggExtra_0.10.1            limma_3.58.1
##  [77] posterior_1.5.0           later_1.3.1
##  [79] dplyr_1.1.3               BiocFileCache_2.10.1
##  [81] lattice_0.22-5            bit_4.0.5
##  [83] tidyselect_1.2.0          locfit_1.5-9.8
##  [85] Biostrings_2.70.1         miniUI_0.1.1.1
##  [87] gridExtra_2.3             bookdown_0.36
##  [89] edgeR_4.0.1               xfun_0.40
##  [91] statmod_1.5.0             stringi_1.7.12
##  [93] yaml_2.3.7                evaluate_0.22
##  [95] codetools_0.2-19          tibble_3.2.1
##  [97] BiocManager_1.30.22       cli_3.6.1
##  [99] xtable_1.8-4              munsell_0.5.0
## [101] Rcpp_1.0.11               dbplyr_2.4.0
## [103] coda_0.19-4               png_0.1-8
## [105] XML_3.99-0.15             parallel_4.3.2
## [107] ellipsis_0.3.2            assertthat_0.2.1
## [109] blob_1.2.4                prettyunits_1.2.0
## [111] sparseMatrixStats_1.14.0  bitops_1.0-7
## [113] scales_1.2.1              crayon_1.5.2
## [115] rlang_1.1.1               cowplot_1.1.1
## [117] KEGGREST_1.42.0
```

Durinck, Steffen, Paul T Spellman, Ewan Birney, and Wolfgang Huber. 2009. "Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor Package biomaRt." *Nature Protocols* 4 (8): 1184–91. https://doi.org/10.1038/nprot.2009.97.

Ibarra-Soria, Ximena, Wajid Jawaid, Blanca Pijuan-Sala, Vasileios Ladopoulos, Antonio Scialdone, David J. Jörg, Richard C. V. Tyser, et al. 2018. "Defining murine organogenesis at single-cell resolution reveals a role for the leukotriene pathway in regulating blood progenitor formation." *Nature Cell Biology* 20 (2): 127–34. https://doi.org/10.1038/s41556-017-0013-z.

Lloyd-Smith, James O. 2007. "Maximum Likelihood Estimation of the Negative Binomial Dispersion Parameter for Highly Overdispersed Data, with Applications to Infectious Diseases." Edited by Mark Rees. *PLoS ONE* 2 (2): e180. https://doi.org/10.1371/journal.pone.0000180.

Lopez, Romain, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. 2018. "Deep Generative Modeling for Single-Cell Transcriptomics." *Nature Methods* 15 (12): 1053–58. https://doi.org/10.1038/s41592-018-0229-2.

Townes, F. William, Stephanie C. Hicks, Martin J. Aryee, and Rafael A. Irizarry. 2019. "Feature Selection and Dimension Reduction for Single-Cell RNA-Seq Based on a Multinomial Model." *Genome Biology* 20 (1): 295. https://doi.org/10.1186/s13059-019-1861-6.