BASiCS workflow: a step-by-step analysis of expression variability using single cell RNA sequencing data

Nils Eling[1,2,3], Alan O'Callaghan[*4], John C. Marioni[1,2], and Catalina A. Vallejos[†4,5]

[1]**European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK**
[2]**Cancer Research UK Cambridge Institute, University of Cambridge, Li Ka Shing Centre, Cambridge, CB2 0RE, UK**
[3]**Department of Quantitative Biomedicine, University of Zurich, Winterthurerstrasse 190, CH-8057, Zurich, Switzerland**
[4]**MRC Human Genetics Unit, Institute of Genetics & Molecular Medicine, University of Edinburgh, Western General Hospital, Crewe Road, Edinburgh, EH4 2XU, UK**
[5]**The Alan Turing Institute, British Library, 96 Euston Road, London, NW1 2DB, UK**

Abstract Cell-to-cell gene expression variability is an inherent feature of complex biological systems, such as immunity and development. Single-cell RNA sequencing is a powerful tool to quantify this heterogeneity, but it is prone to strong technical noise. In this article, we describe a step-by-step computational workflow which uses the BASiCS Bioconductor package to robustly quantify expression variability within and between known groups of cells (such as experimental conditions or cell types). BASiCS uses an integrated framework for data normalisation, technical noise quantification and downstream analyses, whilst propagating statistical uncertainty across these steps. Within a single seemingly homogeneous cell population, BASiCS can identify highly variable genes that exhibit strong heterogeneity as well as lowly variable genes with stable expression. BASiCS also uses a probabilistic decision rule to identify changes in expression variability between cell populations, whilst avoiding confounding effects related to differences in technical noise or in overall abundance. Using two publicly available datasets, we guide users through a complete pipeline which includes preliminary steps for quality control as well as data exploration using the scater and scran Bioconductor packages. Data for the first case study was generated using the Fluidigm@ C1 system, in which extrinsic spike-in RNA molecules were added as a control. The second dataset was generated using a droplet-based system, for which spike-in RNA is not available. This analysis provides an example, in which differential variability testing reveals insights regarding a possible early cell fate commitment process. The workflow is accompanied by a Docker image that ensures the reproducibility of our results.

Keywords

Single-cell RNA sequencing, expression variability, transcriptional noise, differential expression testing

[*] a.b.o'callaghan@sms.ed.ac.uk
[†] catalina.vallejos@igmm.ed.ac.uk

Introduction

Single-cell RNA-sequencing (scRNA-seq) enables the study of genome-wide transcriptional heterogeneity in cell populations that cannot be detected in bulk experiments [1, 2, 3]. On the broadest level, this heterogeneity can reflect the presence of distinct cell subtypes or states. Alternatively, it can be due to gradual changes along biological processes, such as development and differentiation. Several clustering and pseudotime inference methods have been developed to characterise these types of heterogeneity [4, 5]. However, there is a limited availability of computational tools tailored to study more subtle variability within seemingly homogeneous cell populations. This variability can reflect deterministic or stochastic events that regulate gene expression and, among others, has been reported to increase prior to cell fate decisions [6] as well as during ageing [7].

This article complements existing scRNA-seq workflows based on the Bioconductor package ecosystem (e.g. [8, 9]). We describe a step-by-step analysis which uses *scater* and *scran* to perform quality control (QC) as well as initial exploratory analyses [10, 8]. To robustly quantify transcriptional variability we use *BASiCS* [11, 12, 13] — a Bayesian hierarchical framework that jointly performs data normalisation (global scaling), technical noise quantification and downstream analyses, whilst propagating statistical uncertainty across these steps. Among others, *BASiCS*, has led to new insights about the heterogeneity of immune cells [7].

Within a population of cells, *BASiCS* decomposes the total observed variability in expression measurements into technical and biological components [11]. This enables the identification of *highly variable genes* (HVGs) that capture the major sources of heterogeneity within the analysed cells [14]. HVG detection is often used as feature selection, to identify the input set of genes for subsequent analyses. *BASiCS* can also highlight *lowly variable genes* (LVGs) that exhibit stable expression across the population of cells. These may relate to essential cellular functions and can assist the development of new data normalisation or integration strategies [15].

BASiCS also provides a probabilistic decision rule to perform differential expression analyses between two (or more) pre-specified groups of cells [12, 16]. Whilst several differential expression tools have been proposed for scRNA-seq data (e.g. [17, 18]), some evidence suggests that these do not generally outperform popular bulk RNA-seq tools [19]. Moreover, most of these methods are only designed to uncover changes in overall expression, ignoring the more complex patterns that can arise at the single cell level [20]. Instead, *BASiCS* embraces the high granularity of scRNA-seq data, uncovering changes in cell-to-cell expression variability that are not confounded by differences in technical noise or in overall expression.

Here, we briefly discuss the sources of variability that arise in scRNA-seq data and some of the strategies that have been designed to control or attenuate technical noise in these assays. We also summarise the main features of the Bioconductor packages used throughout this workflow, and provide a description for the underlying statistical model implemented in *BASiCS*. This includes practical guidance to assess the convergence of the Markov Chain Monte Carlo (MCMC) algorithm that is used to infer model parameters as well as recommendations to interpret and post-process the model outputs. Finally, we provide a step-by-step case study.

All source code used to generate the results presented in this article is available on Github. To ensure the reproducibility of this workflow, the analysis environment and all software dependencies are provided as a Docker image [21]. The image can be obtained from Docker Hub.

Sources of variability in scRNA-seq data

The focus of this article is to quantify the magnitude of cell-to-cell expression heterogeneity within seemingly homogeneous cell populations. Here, we briefly describe the underlying sources of heterogeneity that can be captured by cell-to-cell variability estimates derived from scRNA-seq data.

Stochastic variability within a cell population — often referred to as transcriptional *noise* — can arise from intrinsic and extrinsic sources [22, 23]. Classically, extrinsic noise is defined as stochastic fluctuations induced by different dynamic cellular states (e.g. cell cycle, metabolism, intra- and inter-cellular signalling) [24, 25, 26]. In contrast, intrinsic noise arises from stochastic effects on biochemical processes such as transcription and translation [22]. Intrinsic noise can be modulated by genetic and epigenetic modifications (such as mutations, histone modifications, CpG island length and nucleosome positioning) [27, 28, 29] and is usually measured at the gene level [22]. Cell-to-cell gene expression variability estimates derived from scRNA-seq data capture a combination of these effects, as well as deterministic regulatory mechanisms [23]. Moreover, these variability estimates can also be inflated by the technical noise that is typically observed in scRNA-seq data [14].

Different strategies have been incorporated into scRNA-seq protocols to control or attenuate technical noise. For example, external RNA spike-in molecules (such as the set introduced by the External RNA Controls Consortium, ERCC [30]) can be added to each cell's lysate in a (theoretically) known fixed quantity. Spike-ins can assist quality control steps [10], data normalisation [31] and can be used to infer technical noise [14]. Another strategy is to tag individual cDNA molecules using unique molecular identifiers (UMIs) before PCR amplification [32]. Reads that contain the same UMI can be collapsed into a single molecule count, attenuating technical variability associated to cell-to-cell differences in amplification and sequencing depth (these technical biases are not fully removed unless sequencing to saturation [31]). However, despite the benefits associated to the use of spike-ins and UMIs, these are not available for all scRNA-seq protocols [33].

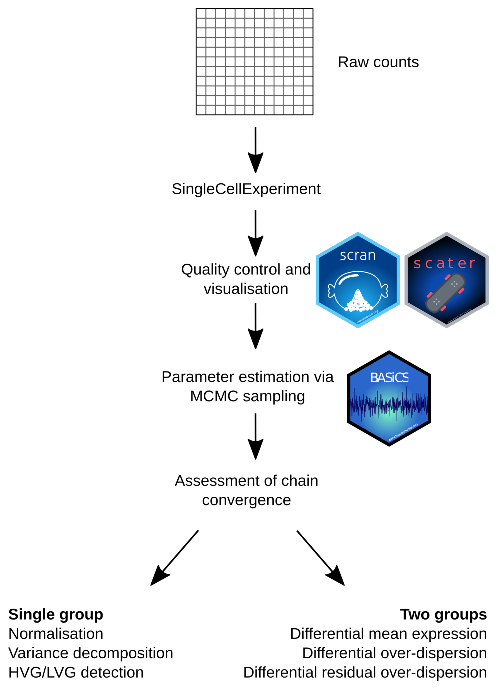
Raw counts
SingleCellExperiment
Quality control and visualisation
scran
scater
Parameter estimation via MCMC sampling
BASiCS
Assessment of chain convergence
Single group
Normalisation
Variance decomposition
HVG/LVG detection
Two groups
Differential mean expression
Differential over-dispersion
Differential residual over-dispersion

Figure 1. Graphical overview for the scRNA-seq analysis workflow described in this manuscript. Starting from a matrix of expression counts, we use the scater and scran Bioconductor packages to perform QC and initial exploratory analyses. To robustly quantify transcriptional heterogeneity within seemingly homogeneous cell populations, we apply the BASiCS Bioconductor package and illustrate how BASiCS can be used to analyse a single or multiple pre-specified groups of cells.

Methods

This step-by-step scRNA-seq workflow is primarily based on the Bioconductor package ecosystem [34]. A graphical overview is provided in Figure 1 and its main components are described below.

Input data

```
library("SingleCellExperiment")
```

We use *SingleCellExperiment* to convert an input matrix of raw read-counts (molecule counts for UMI-based protocols) into a `SingleCellExperiment` object which can also store its associated metadata, such as gene- and cell-specific information. Moreover, when available, the same object can also store read-counts for spike-in molecules (see `altExp()`). A major advantage of using a `SingleCellExperiment` object as the input for scRNA-seq analyses is the interoperability across a large number of Bioconductor packages [34].

Quality control and exploratory analysis

```
library("scater")
library("scran")
library("ggplot2")
```

An critical step in scRNA-seq analyses is to apply QC diagnostics, removing low quality samples that may distort downstream analyses. Among others, QC can help to identify samples that contain broken cells, that are empty or that contain multiple cells [35]. Moreover, lowly expressed genes for which less reliable information is available are typically also removed. The *OSCA* online book provides an extensive overview on important aspects of how to perform QC of scRNA-seq data, including exploratory analyses [34].

To perform QC, we use the *scater* package [10]. The `addPerCellQC` and `addPerFeatureQC` functions are applied to calculate QC metrics for each cell (e.g. total read-count) and gene (e.g. percentage of zeroes across all cells), respectively. The package also provides a suite of visualisation tools that can be used to explore the data under study and its associated QC diagnostic metrics.

The *scran* package offers additional tools for QC diagnostics and a variety of functions scRNA-seq data analysis [8]. It can perform *global scaling* normalisation, calculating cell-specific scaling factors that capture global differences in read-counts across cells (e.g. due to sequencing depth and PCR amplification) [36]. To explore the strenght of transcriptional variability, we use the `modelGeneCV2` function to infer an overall trend between mean expression and the squared coefficent of variation (CV^2) for each gene. To derive gene-specific variability estimates that are not confounded by this overall trend, the `DM` function calculates the distance between CV^2 and a rolling median along the range of mean expression values [37]. DM estimates enable exploratory analyses of cell-to-cell heterogeneity, but a measure of uncertainty is not readily available. As such, gene-specific downstream inference (e.g. differential variability testing) is precluded.

Finally, we also load *ggplot2* to visualise the results of these analyses.

Analysis of cell-to-cell transcriptional variability

```r
library("BASiCS")
```

The *BASiCS* package uses a Bayesian hierarchical framework that borrows information across all genes and cells to robustly quantify transcriptional variability [38]. Similar to the approach adopted in *scran*, *BASiCS* infers cell-specific global scaling normalisation parameters. However, instead of inferring these as a pre-processing step, *BASiCS* uses an integrated approach in which data normalisation and downstream analyses are performed simultaneously, thereby propagating statistical uncertainty. To quantify technical noise, the original implementation of *BASiCS* uses information from extrinsic spike-in molecules as control features, but the model has been extended to address situations in which spike-ins are not available [16].

BASiCS summarises expression patterns through gene-specific *mean* (μ_i) and *over-dispersion* (δ_i) parameters. Mean parameters μ_i quantify the overall expression for each gene i across the population of cells under study. In contrast, δ_i captures the excess of variability that is observed with respect to what would be expected in a homogeneous cell population, after taking into account technical noise. This is used as a proxy to quantify transcriptional variability. Moreover, to account for the strong association that is typically observed between mean expression and over-dispersion estimates, we recently introduced gene-specific *residual over-dispersion* parameters ϵ_i [16]. Similar to DM values implemented in *scran*, these are defined as deviations with respect to an overall regression trend that captures the relationship between mean and over-dispersion values.

Parameter inference is implemented in the `BASiCS_MCMC` function using an adaptive Metropolis within Gibbs algorithm [39], whose convergence can be assessed using the `BASiCS_DiagHist` and `BASiCS_DiagPlot` functions, among others. The output from `BASiCS_MCMC` is a `BASiCS_Chain` object, which can be used for further downstream analyses. In particular, `BASiCS_DetectHVG` and `BASiCS_DetectLVG` can be respectively used to identify highly and lowly variable genes within a cell population. Moreover, `BASiCS_TestDE` is used to perform differential mean and variability analyses between groups of cells.

Case study: analysis of naive CD4$^+$ T cells

As a case study, we use scRNA-seq data generated for CD4$^+$ T cells using the C1 Single-Cell Auto Prep System (Fluidigm®). Martinez-Jimenez *et al.* profiled naive (hereafter also referred to as *unstimulated*) and activated (3 hours using *in vitro* antibody stimulation) CD4$^+$ T cells from young and old animals across two mouse strains to study changes in expression variability during ageing and upon immune activation [7]. They extracted naive or effector memory CD4$^+$ T cells from spleens of young or old animals, obtaining purified populations using either magnetic-activated cell sorting (MACS) or fluorescence activated cell sorting (FACS). External ERCC spike-in RNA [30] was added to aid the quantification of technical variability across all cells and all experiments were performed in replicates (also referred to as batches) to control for batch effects.

Obtaining the data

The matrix with raw read counts can be obtained from ArrayExpress under the accession number E-MTAB-4888. In the matrix, column names contain library identifiers and row names display gene Ensembl identifiers.

```r
if (!file.exists("downloads/raw_data.txt")) {
  # Download raw counts file
  website <- "https://www.ebi.ac.uk/"
  folder <- "arrayexpress/files/E-MTAB-4888/"
  file <- "E-MTAB-4888.processed.1.zip"
  destfile <- "downloads/raw_data.txt.zip"
  download.file(
    paste0(website, folder, file),
    destfile = destfile
  )
  unzip("downloads/raw_data.txt.zip", exdir = "downloads")
  file.remove("downloads/raw_data.txt.zip")
}

# Read in raw data
CD4_raw <- read.table("downloads/raw_data.txt", header = TRUE, sep = "\t")
CD4_raw <- as.matrix(CD4_raw)
```

The input matrix contains data for 1,513 cells and 31,181 genes (including 92 ERCC spike-ins). Information about experimental conditions and batches is available in a metadata file under the same accession number.

```
if (!file.exists("downloads/metadata_file.txt")) {
  # Download raw counts file
  website <- "https://www.ebi.ac.uk/"
  folder <- "arrayexpress/files/E-MTAB-4888/"
  file <- "E-MTAB-4888.additional.1.zip"
  destfile <- "downloads/metadata.txt.zip"
  download.file(
    paste0(website, folder, file),
    destfile = destfile
  )
  unzip("downloads/metadata.txt.zip", exdir = "downloads")
  file.remove("downloads/metadata.txt.zip")
}
# Read in metadata file
CD4_metadata <- read.table(
  "downloads/metadata_file.txt",
  header = TRUE,
  sep = "\t"
)

# Save library identifier as rownames
rownames(CD4_metadata) <- CD4_metadata$X

# Show metadata entries
names(CD4_metadata)
```

```
## [1] "X"            "Strain"      "Age"          "Stimulus"    "Individuals"
## [6] "Celltype"
```

The metadata contains library identifiers (`X`), strain information (`Strain`; *Mus musculus castaneus* or *Mus musculus domesticus*), the age of the animals (`Age`; young or old), stimulation state of the cells (`Stimulus`; naive or activated), batch information (`Individuals`; associated to different mice), and cell type information (`Celltype`; via FACS or MACS purification).

The data and metadata described above are then converted into a *SingleCellExperiment* object.

```
# Separate intrinsic from ERCC counts
bio_counts <- CD4_raw[!grepl("ERCC", rownames(CD4_raw)), ]
spike_counts <- CD4_raw[grepl("ERCC", rownames(CD4_raw)), ]
# Generate the SingleCellExperiment object
sce_CD4_all <- SingleCellExperiment(
  assays = list(counts = as.matrix(bio_counts)),
  colData = CD4_metadata[colnames(CD4_raw), ]
)
# Add read-counts for spike-ins
altExp(sce_CD4_all, "spike-ins") <- SummarizedExperiment(
  assays = list(counts = spike_counts)
)
sce_CD4_all
```

```
## class: SingleCellExperiment
## dim: 31089 1513
## metadata(0):
## assays(1): counts
## rownames(31089): ENSMUSG00000000001 ENSMUSG00000000003 ...
##   ENSMUSG00000106668 ENSMUSG00000106670
## rowData names(0):
## colnames(1513): do4737 do4739 ... do12254 do12257
## colData names(6): X Strain ... Individuals Celltype
## reducedDimNames(0):
## altExpNames(1): spike-ins
```

Throughout our analysis, we focus on naive and activated CD4$^+$ T cells obtained from young *Mus musculus domesticus* animals, purified using MACS-based cell sorting. The following code is used to extract these 146 samples from the full dataset.

```
ind_select <- sce_CD4_all$Strain == "Mus musculus domesticus" &
  sce_CD4_all$Age == "Young" &
  sce_CD4_all$Celltype == "MACS-purified Naive"
sce_naive_active <- sce_CD4_all[, ind_select]
sce_naive_active
```

```
## class: SingleCellExperiment
## dim: 31089 146
## metadata(0):
## assays(1): counts
## rownames(31089): ENSMUSG00000000001 ENSMUSG00000000003 ...
##   ENSMUSG00000106668 ENSMUSG00000106670
## rowData names(0):
## colnames(146): do6113 do6118 ... do6493 do6495
## colData names(6): X Strain ... Individuals Celltype
## reducedDimNames(0):
## altExpNames(1): spike-ins
```

QC and exploratory analysis

The data available at E-MTAB-4888 have been already filtered to remove poor quality samples. The QC applied in [7] removed cells with: (i) fewer than 1,000,000 total reads, (ii) less than 20% of reads mapped to endogenous genes, (iii) less than 1,250 or more than 3,000 detected genes and (iv) more than 10% or fewer than 0.5% of reads mapped to mitochondrial genes. As an illustration, we visualise some of these metrics. We also include another widely used QC diagnostic plot which compares the total number (or fraction) of spike-in counts versus the total number (or fraction) of endogeneous counts. In such a plot, low quality samples are characterised by a high fraction of spike-in counts and a low fraction of endogeneous counts (see Figure 2).

```
# Calculate per cell QC metrics
sce_naive_active <- addPerCellQC(sce_naive_active, use_altexps = TRUE)
p_cellQC1 <- plotColData(sce_naive_active, x = "sum", y = "detected") +
  xlab("Total engogenous reads per cell") +
  ylab("Number of detected genes per cell")
p_cellQC2 <- plotColData(sce_naive_active, x = "sum", y = "altexps_spike-ins_sum") +
  xlab("Total engogenous reads per cell") +
  ylab("Total spike-in reads per cell")
multiplot(p_cellQC1, p_cellQC2, cols = 2)
```

These metrics can also be visualised with respect to cell-level metadata, such as the experimental conditions (active vs unstimulated) and the different mice from which cells were collected (see Figure 3).

```
p_stimulus <- plotColData(
    sce_naive_active,
    x = "sum",
    y = "detected",
    colour_by = "Stimulus") +
  xlab("Total engogenous reads per cell") +
  ylab("Number of detected genes per cell") +
```

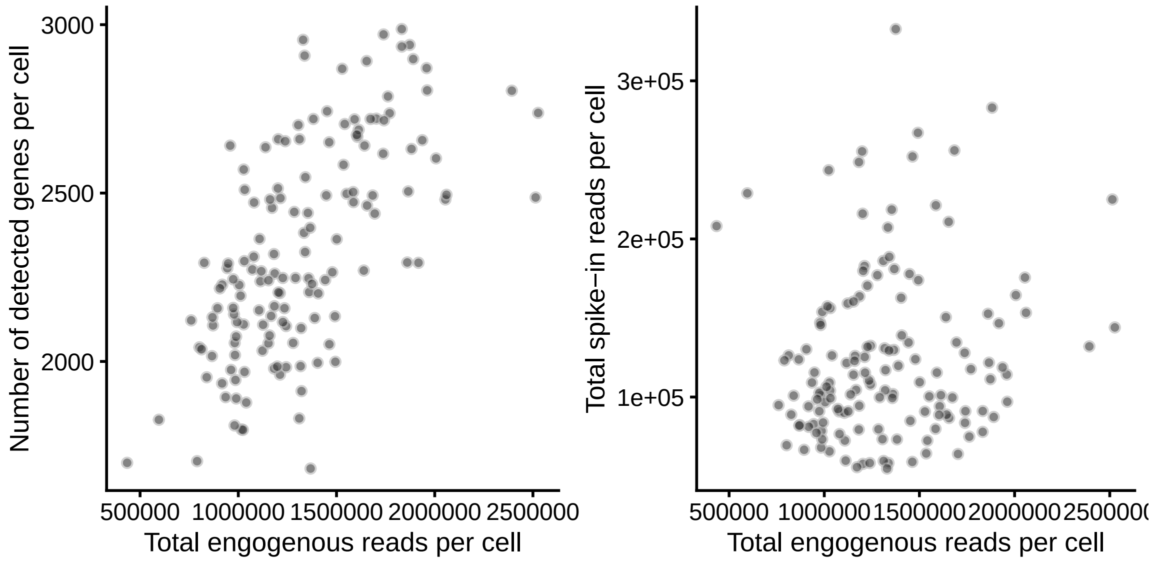

Figure 2. Cell-level QC metrics. The total number of endogenous read-counts (excludes non-mapped and intronic reads) is plotted against the total number of detected genes (left) and the total number of spike-in read-counts (right).

```
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1))

p_batch <- plotColData(
    sce_naive_active,
    x = "sum",
    y = "detected",
    colour_by = "Individuals") +
  xlab("Total engogenous reads per cell") +
  ylab("Number of detected genes per cell") +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1))

multiplot(p_stimulus, p_batch, cols = 2)
```

To further explore the underlying structure of the data, we compute global scaling normalisation factors using *scran* and perform a principal component analysis (PCA) of log-transformed normalised expression counts using *scater*. The results of this analysis are visualised with respect to cell-level metadata and QC metrics. As seen in Figure 4, this analysis suggests the absence of strong batch effects. Moreover, we observe that activated cells tend to exhibit a higher number of expressed genes and total read count (see Figure 5).

```
# Global scaling normalisation
sce_naive_active <- computeSumFactors(sce_naive_active)
sce_naive_active <- logNormCounts(sce_naive_active)
# Calculate PCA
sce_naive_active <- runPCA(sce_naive_active)
# Visualise the conditions and batch structure
p_stimulus <- plotPCA(sce_naive_active, colour_by = "Stimulus") +
  theme(legend.position = "bottom")
p_batch <- plotPCA(sce_naive_active, colour_by = "Individuals") +
  theme(legend.position = "bottom")
multiplot(p_stimulus, p_batch, cols = 2)
```

```
# Visualise number of endogeneous genes detected
p_total_features <- plotPCA(sce_naive_active, colour_by = "detected") +
  theme(
```

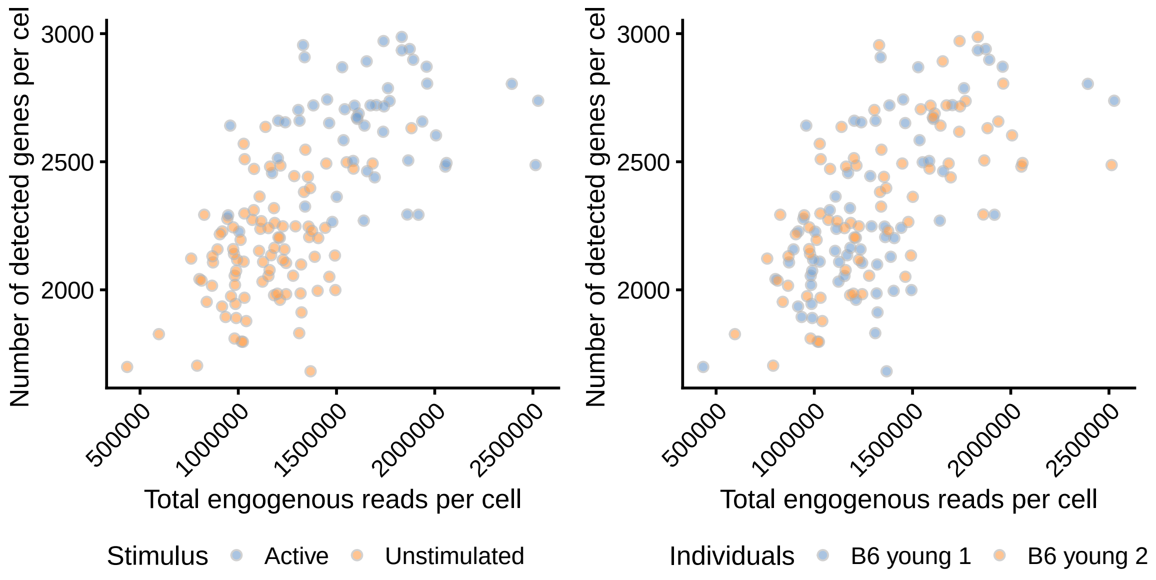

Figure 3. Cell-level QC metrics according to cell-level metadata. The total number of endogenous reads (excludes non-mapped and intronic reads) is plotted against the total number of detected genes. Colour indicates the experimental condition (left) and animal of origin (right) for each cell.

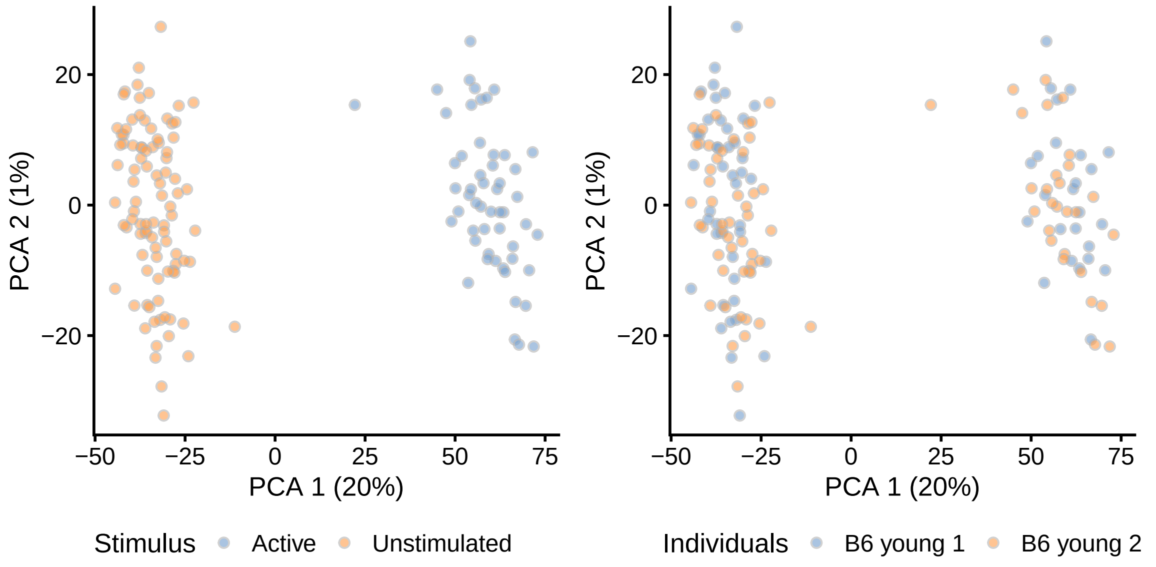

Figure 4. First two principal components of log-transformed expression counts after scran normalisation. Colour indicates the experimental condition (left) and animal of origin (right) for each cell.

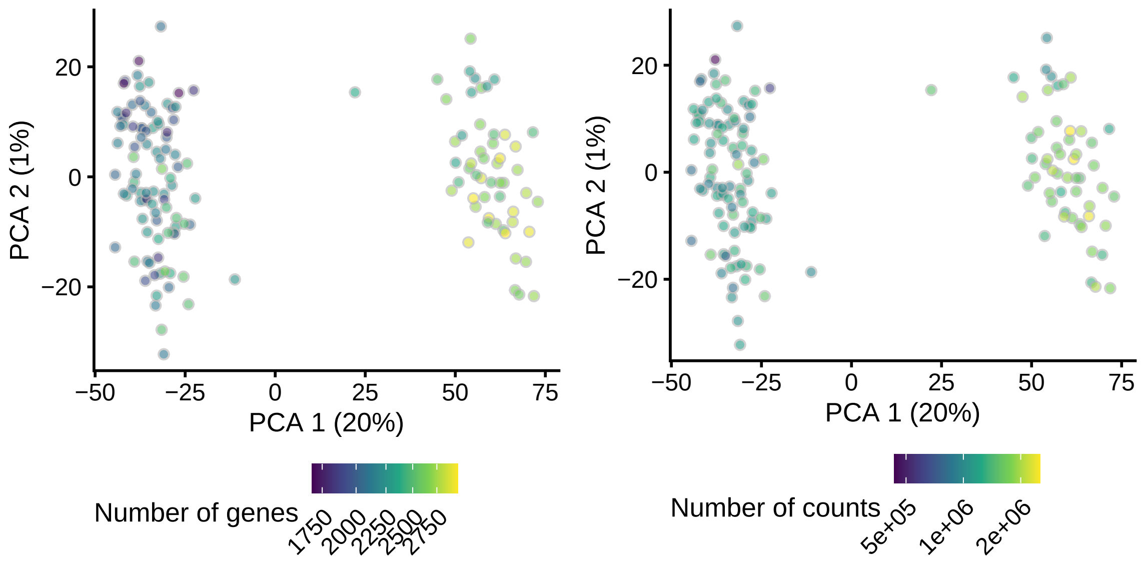

Figure 5. First two principal components of log-transformed expression counts after scran normalisation. Colour indicates the number of genes detected per cell (left) and the total number of endogenous reads per cell (right).

```
    legend.position = "bottom",
    legend.text = element_text(hjust = 1, angle = 45)) +
  scale_fill_viridis_c(name = "Number of genes", trans = "log10")

# Visualise log10-transformed total number of counts
p_total_counts <- plotPCA(sce_naive_active, colour_by = "sum") +
  theme(legend.position = "bottom",
    legend.text = element_text(hjust = 1, angle = 45)) +
  scale_fill_viridis_c(name = "Number of counts", trans = "log10")
multiplot(p_total_features, p_total_counts, cols = 2)
```

Following cell-specific QC, we will remove all genes that are not detected in at least 5 cells across both conditions or for which their average read count (across all cells) is below 1. These thresholds need to be set specifically for each dataset, and careful gene-specific quality metrics need to be closely examined as suggested by the *OSCA* Bioconductor workflow [34].

```
# Remove genes
ind_expressed <- rowSums(counts(sce_naive_active) > 0) >= 5 &
  rowMeans(counts(sce_naive_active)) >= 1
sce_naive_active <- sce_naive_active[ind_expressed, ]
```

The final dataset used in subsequent analyses contains 146 cells and 8953 genes.

Calculating the molecule count for spike-in genes

BASiCS requires the absolute molecule count of the spike-in transcripts that were added to each well. To calculate the molecule count, we require the dilution and the concentration of the spike-in mix. For this, a table of the spike-in concentrations can be downloaded from https://www.thermofisher.com. The file contains the concentrations of 2 ERCC spike-in mixes.

For the CD4$^+$ T cell data, the authors added a 1:50,000 dilution of the ERCC spike-in mix 1 [7]. We will first download the concentration information.

```
# Read in the spike-in concentrations
download_file <- function(file, website, folder, destfile = file) {
  if (!file.exists(destfile)) {
    download.file(paste0(website, folder, file), destfile)
  }
}
website <- "https://assets.thermofisher.com/"
folder <- "TFS-Assets/LSG/manuals/"
file <- "cms_095046.txt"
download_file(file, website, folder, "downloads/spike_info.txt")

ERCC_conc <- read.table(
  "downloads/spike_info.txt",
  sep = "\t", header = TRUE
)
```

The concentration is given in units of $aM\mu l^{-1}$. We will first calculate the concentration in $M\mu l^{-1}$.

```
# Moles per micro litre
ERCC_mmul <- ERCC_conc$concentration.in.Mix.1..attomoles.ul. * (10^(-18))
```

From this, we can calculate the molecule count per μl using the fact that 1 mole comprises $6.02214076 \times 10^{23}$ molecules.

```
# Molecule count per micro litre
ERCC_countmul <- ERCC_mmul * (6.02214076 * (10^23))
```

During the preparation of the reaction mix, the authors diluted this mix by a factor of 1:50,000 [7]. The actual molecule number of spike-ins can therefore be calculated per μl:

```
ERCC_count <- ERCC_countmul / 50000
```

The volume per well in the IFC chip is $9nl$ `https://www.fluidigm.com/faq/ifc-9`. We therfore need to calculate the number of molecules in $9nl$ reaction mix.

```
ERCC_count_final <- ERCC_count * 0.009
```

Depending on the technology used to capture and process single cells, as well as the dilution of the spike-in mix, the absolute number of spike-ins per reaction varies across experiments. The absolute amount of spike-ins per reaction is only used to calibrate the scale of the capture efficiency normalisation parameter ν_j. Since the true concentration of each spike-in molecule is known, BASiCS uses the observed counts of each molecule for each cell to estimate the capture efficiency for each cell, ν_j. Differences in the global scale across all wells of the level of spike-ins molecules thus do not affect the cell-to-cell ratios of ν_j. BASiCS assumes that the quantity of spike-ins is consistent in each well in each experiment. While the quantity used must remain constant between wells and experiments, the scale does not affect the results, provided the spike-ins are at a reasonable level. In particular, they should not be in such a low concentration that they are rarely detected, and they should not be at such a high concentration that the majority of the sequencing reads map to the spike-in molecules.

We can now use the molecule count to prepare the BASiCS data object. To incorporate the spike-in molecule counts within the *SingleCellExperiment* object that BASiCS requires, the first column must contain the names associated to the spike-in molecules. The second column must contain the input number of molecules for the spike-in genes (amount per reaction).

```
# Prepare the data.frame
ERCC_count <- data.frame(
  row.names = ERCC_conc$ERCC.ID,
  Names = ERCC_conc$ERCC.ID,
  count = ERCC_count_final
)
```

Single condition example: Naive CD4$^+$ T cells

To highlight the use of BASiCS to analyse cells in the single-condition case, we select naive CD4$^+$ T cells of young B6 animals. Here, BASiCS can be used to identify highly- and lowly-variable genes and calculate interpretable gene- and cell-specific parameters. Among others, these include gene-specific parameters for mean expression, over-dispersion (biological variability) and residual over-dispersion (biological variability after correcting for the confounding of mean and over-dispersion).

Preparing the BASiCS data object

Many BASiCS functions use objects of the class *SingleCellExperiment* as input. The `newBASiCS_Data` function can be used to create the required `SingleCellExperiment` object based on the following information:

- Counts: a matrix of raw expression counts with dimensions q times n, where q is the number of genes (technical + biological) and n is the number of cells. Gene names must be stored as row names of the counts matrix.
- Tech: a vector of `TRUE`/`FALSE` elements with length q. If `Tech[i]` contains a value of `FALSE`, gene i is biological; otherwise, the gene is spike-in. This vector must be specified in the same order of genes as in the counts matrix.
- SpikeInfo: a `data.frame` with $q - q_0$ rows where q_0 is the number of biological genes. The first column must contain the names associated to the spike-in molecules. The second column must contain the input number of molecules for the spike-in molecules (amount per cell).
- BatchInfo (optional argument): vector of length n to indicate batch structure in situations where cells have been processed using multiple batches.

We will first select the naive cells from the `SingleCellExperiment` object that was generated above.

```r
ind_stimulated <- sce_naive_active$Stimulus == "Unstimulated"
sce_naive <- sce_naive_active[, ind_stimulated]
ind_expressed <- rowSums(counts(sce_naive) > 0) > 1 &
  rowMeans(counts(sce_naive)) >= 1
sce_naive <- sce_naive[ind_expressed, ]
```

Next, we will use the `newBASiCS_Data` function to re-generate the `SingleCellExperiment` object for use with *BASiCS*.

```r
# Here is the first time that we use BASiCS

# Select the ERCC spike-ins of the dataset
counts <- counts(sce_naive)
spikes <- assay(altExp(sce_naive, "spike-ins"))
spikes_present <- rowSums(spikes) != 0
## Remove spike-ins that are not present from matrix and SCE object
spikes <- spikes[spikes_present, ]
altexp_present <- altExp(sce_naive, "spike-ins")[spikes_present, ]
altExp(sce_naive, "spike-ins") <- altexp_present
ind_spike <- c(rep(FALSE, nrow(counts)), rep(TRUE, nrow(spikes)))
spike_input <- ERCC_count[rownames(spikes), ]

# Generate the SingleCellExperiment object
data_naive <- newBASiCS_Data(
  Counts = rbind(counts, spikes),
  Tech = ind_spike,
  SpikeInfo = spike_input,
  BatchInfo = sce_naive$Individuals
)
data_naive
```

```
## class: SingleCellExperiment
## dim: 8427 93
## metadata(1): SpikeInput
## assays(1): counts
## rownames(8427): ENSMUSG00000000001 ENSMUSG00000000056 ...
##   ENSMUSG00000106607 ENSMUSG00000106620
## rowData names(0):
## colnames(93): do6113 do6118 ... do6398 do6399
## colData names(1): BatchInfo
## reducedDimNames(0):
## altExpNames(1): spike-ins
```

Alternatively, when using datasets that contain spike-in molecules, the original SingleCellExperiment object can be extended by simply specifying a BatchInfo slot in the colData object and by adding the SpikeInfo object to the metadata slot.

```
data_naive <- sce_naive
colData(data_naive)$BatchInfo <- colData(sce_naive)$Individuals
metadata(data_naive)$SpikeInput <- spike_input
```

After creating the SingleCellExperiment object that contains all information that *BASiCS* requires, the MCMC sampler can be run to generate posterior estimates of model parameters.

References

[1] Oliver Stegle, Sarah a. Teichmann, and John C. Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, jan 2015. ISSN 1471-0056. doi: 10.1038/nrg3833. URL http://www.nature.com/doifinder/10.1038/nrg3833.

[2] Sanjay M. Prakadan, Alex K. Shalek, and David A. Weitz. Scaling by shrinking: empowering single-cell 'omics' with microfluidic devices. *Nature Reviews Genetics*, 18(6):345–361, 2017. ISSN 1471-0056. doi: 10.1038/nrg.2017.15. URL http://www.nature.com/doifinder/10.1038/nrg.2017.15.

[3] Simona Patange, Michelle Girvan, and Daniel R. Larson. Single-cell systems biology: Probing the basic unit of information flow. *Current Opinion in Systems Biology*, 8:7–15, 2018. ISSN 24523100. doi: 10.1016/j.coisb.2017.11.011. URL https://doi.org/10.1016/j.coisb.2017.11.011.

[4] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics 2018*, 2019. ISSN 1471-0064. doi: 10.1038/s41576-018-0088-9. URL https://www.nature.com/articles/s41576-018-0088-9?utm{_}source=feedburner{&}utm{_}medium=feed{&}utm{_}campaign=Feed{%}3A+nrg{%}2Frss{%}2Fcurrent+{%}28Nature+Reviews+Genetics+-+Issue{%}29.

[5] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, May 2019. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-019-0071-9.

[6] Mitra Mojtahedi, Alexander Skupin, Joseph Zhou, Ivan G. Castaño, Rebecca Y. Y. Leong-Quong, Hannah Chang, Kalliopi Trachana, Alessandro Giuliani, and Sui Huang. Cell Fate Decision as High-Dimensional Critical State Transition. *PLOS Biology*, 14(12):e2000640, December 2016. ISSN 1545-7885. doi: 10.1371/journal.pbio.2000640.

[7] Celia P. Martinez-Jimenez, Nils Eling, Hung-Chang Chen, Catalina A Vallejos, Aleksandra A Kolodziejczyk, Frances Connor, Lovorka Stojic, Timothy F Rayner, Michael J T Stubbington, Sarah A Teichmann, Maike de la Roche, John C Marioni, and Duncan T Odom. Aging increases cell-to-cell transcriptional variability upon immune stimulation. *Science*, 1436:1433–1436, 2017. doi: 10.1126/science.aah4115.

[8] Aaron T. L. Lun, Davis J. McCarthy, and John C. Marioni. A step-by-step workflow for basic analyses of single-cell RNA-seq data. *F1000Research*, 5(2122), 2016. ISSN 2046-1402. doi: 10.12688/f1000research.9501.1.

[9] Beomseok Kim, Eunmin Lee, and Jong Kyoung Kim. Analysis of Technical and Biological Variabilityin Single-Cell RNA Sequencing. In *Computational Methods for Single-Cell Data Analysis*, volume 1935, pages 25–43. 2019. ISBN 978-1-4939-9056-6. doi: 10.1007/978-1-4939-9057-3. URL http://www.ncbi.nlm.nih.gov/pubmed/30758827{%}0Ahttp://link.springer.com/10.1007/978-1-4939-9057-3{_}12{%}0Ahttp://link.springer.com/10.1007/978-1-4939-9057-3.

[10] Davis J. McCarthy, Kieran R. Campbell, Aaron T.L. Lun, and Quin F. Wills. Scater: Pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*, 33(8):1179–1186, 2017. ISSN 14602059. doi: 10.1093/bioinformatics/btw777.

[11] Catalina A. Vallejos, John C. Marioni, and Sylvia Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLOS Computational Biology*, 11:e1004333, 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004333. URL http://dx.plos.org/10.1371/journal.pcbi.1004333.

[12] Catalina A Vallejos, Sylvia Richardson, and John C Marioni. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biology*, 17(70), 2016. doi: 10.1101/035949. URL http://biorxiv.org/content/early/2016/01/05/035949.abstract.

[13] Nils Eling, Arianne C. Richard, Sylvia Richardson, John C. Marioni, and Catalina A. Vallejos. Robust expression variability testing reveals heterogeneous T cell responses. *bioRxiv*, page 237214, 2017. doi: 10.1101/237214. URL https://www.biorxiv.org/content/early/2017/12/21/237214.full.pdf+html.

[14] Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kołodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah a Teichmann, John C Marioni, and Marcus G Heisler. Accounting for technical noise in single-cell RNA-seq experiments., 2013. ISSN 1548-7105. URL http://www.ncbi.nlm.nih.gov/pubmed/24056876.

[15] Yingxin Lin, Shila Ghazanfar, Dario Strbenac, Andy Wang, Ellis Patrick, David M Lin, Terence Speed, Jean Y H Yang, and Pengyi Yang. Evaluating stably expressed genes in single cells. *GigaScience*, 8(9):giz106, September 2019. ISSN 2047-217X. doi: 10.1093/gigascience/giz106.

[16] Nils Eling, Arianne C. Richard, Sylvia Richardson, John C. Marioni, and Catalina A. Vallejos. Correcting the Mean-Variance Dependency for Differential Variability Testing Using Single-Cell RNA Sequencing Data. *Cell Systems*, 7(3): 1–11, 2018. ISSN 24054712. doi: 10.1016/j.cels.2018.06.011. URL `https://linkinghub.elsevier.com/retrieve/pii/S2405471218302783`.

[17] Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–2, jul 2014. ISSN 1548-7105. doi: 10.1038/nmeth.2967. URL `http://www.ncbi.nlm.nih.gov/pubmed/24836921`.

[18] Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K. Shalek, Chloe K. Slichter, Hannah W. Miller, M. Juliana McElrath, Martin Prlic, Peter S. Linsley, and Raphael Gottardo. MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(1):278, December 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0844-5.

[19] Charlotte Soneson and Mark D Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods*, 15(4):255–261, April 2018. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.4612.

[20] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J. McCarthy, Stephanie C. Hicks, Mark D. Robinson, Catalina A. Vallejos, Kieran R. Campbell, Niko Beerenwinkel, Ahmed Mahfouz, Luca Pinello, Pavel Skums, Alexandros Stamatakis, Camille Stephan-Otto Attolini, Samuel Aparicio, Jasmijn Baaijens, Marleen Balvert, Buys de Barbanson, Antonio Cappuccio, Giacomo Corleone, Bas E. Dutilh, Maria Florescu, Victor Guryev, Rens Holmer, Katharina Jahn, Thamar Jessurun Lobo, Emma M. Keizer, Indu Khatri, Szymon M. Kielbasa, Jan O. Korbel, Alexey M. Kozlov, Tzu-Hao Kuo, Boudewijn P.F. Lelieveldt, Ion I. Mandoiu, John C. Marioni, Tobias Marschall, Felix Mölder, Amir Niknejad, Lukasz Raczkowski, Marcel Reinders, Jeroen de Ridder, Antoine-Emmanuel Saliba, Antonios Somarakis, Oliver Stegle, Fabian J. Theis, Huan Yang, Alex Zelikovsky, Alice C. McHardy, Benjamin J. Raphael, Sohrab P. Shah, and Alexander Schönhuth. Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):31, December 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1926-6.

[21] Carl Boettiger. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1): 71–79, January 2015. ISSN 0163-5980. doi: 10.1145/2723872.2723882.

[22] Michael B Elowitz, Arnold J Levine, Eric D Siggia, and Peter S Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002. ISSN 1095-9203. doi: 10.1126/science.1070919. URL `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve{&}db=PubMed{&}dopt=Citation{&}list{_}uids=12183631`.

[23] Nils Eling, Michael D. Morgan, and John C. Marioni. Challenges in measuring and understanding biological noise. *Nature Reviews Genetics*, 20(9):536–548, September 2019. ISSN 1471-0056, 1471-0064. doi: 10.1038/s41576-019-0130-6.

[24] Cristopher J. Zopf, Katie Quinn, Joshua Zeidman, and Narendra Maheshri. Cell-Cycle Dependence of Transcription Dominates Noise in Gene Expression. *PLoS Computational Biology*, 9(7):1–12, 2013. ISSN 1553734X. doi: 10.1371/journal.pcbi.1003161.

[25] Kazunari Iwamoto, Yuki Shindo, and Koichi Takahashi. Modeling Cellular Noise Underlying Heterogeneous Cell Responses in the Epidermal Growth Factor Signaling Pathway. *PLoS Computational Biology*, 12(11):1–18, 2016. ISSN 15537358. doi: 10.1371/journal.pcbi.1005222.

[26] Daniel J. Kiviet, Philippe Nghe, Noreen Walker, Sarah Boulineau, Vanda Sunderlikova, and Sander J. Tans. Stochasticity of metabolism and growth at the single-cell level. *Nature*, 514(7522):376–379, 2014. ISSN 0028-0836. doi: 10.1038/nature13582. URL `http://www.nature.com/doifinder/10.1038/nature13582`.

[27] James Eberwine and Junhyong Kim. Cellular Deconstruction: Finding Meaning in Individual Cell Variation. *Trends in Cell Biology*, 25(10):569–578, 2015. ISSN 18793088. doi: 10.1016/j.tcb.2015.07.004. URL `http://dx.doi.org/10.1016/j.tcb.2015.07.004`.

[28] Andre J. Faure, Jörn M. Schmiedel, and Ben Lehner. Systematic Analysis of the Determinants of Gene Expression Noise in Embryonic Stem Cells. *Cell Systems*, 5(5):471–484, 2017. ISSN 24054720. doi: 10.1016/j.cels.2017.10.003.

[29] Michael D. Morgan and John C. Marioni. CpG island composition differences are a source of gene expression noise indicative of promoter responsiveness. *Genome Biology*, 19(1):1–13, 2018. ISSN 1474760X. doi: 10.1186/s13059-018-1461-x.

[30] External RNA Controls Consortium. Proposed methods for testing and selecting the ERCC external RNA controls. *BMC Genomics*, 6(June 2004):150, 2005. ISSN 1471-2164. doi: 10.1186/1471-2164-6-150.

[31] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14(6):565–571, 2017. ISSN 1548-7091. doi: 10.1038/nmeth.4292. URL `http://www.nature.com/doifinder/10.1038/nmeth.4292`.

[32] Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166, February 2014. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2772.

[33] Ashraful Haque, Jessica Engel, Sarah A. Teichmann, and Tapio Lönnberg. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9(1):75, December 2017. ISSN 1756-994X. doi: 10.1186/s13073-017-0467-4.

[34] Robert A. Amezquita, Vince J. Carey, Lindsay N. Carpp, Ludwig Geistlinger, Aaron T. L. Lun, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, Levi Waldron, Hervé Pagès, Mike Smith, Wolfgang Huber, Martin Morgan, Raphael Gottardo, and Stephanie C. Hicks. Orchestrating Single-Cell Analysis with Bioconductor. Preprint, Genomics, March 2019.

[35] Tomislav Ilicic, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Frederik Otzen Bagger, Davis James Mccarthy, John C Marioni, and Sarah A Teichmann. Classification of low quality cells from single-cell RNA-seq data. *Genome Biology*, 17(29):1–15, 2016. ISSN 1474-760X. doi: 10.1186/s13059-016-0888-1. URL http://dx.doi.org/10.1186/s13059-016-0888-1.

[36] Aaron T. L. Lun, Karsten Bach, and John C. Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17(1):75, 2016. ISSN 1474-760X. doi: 10.1186/s13059-016-0947-7. URL http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0947-7.

[37] Aleksandra A. Kolodziejczyk, Jong Kyoung Kim, Jason C.H. Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N. Natarajan, Alex C. Tuck, Xuefei Gao, Marc Bühler, Pentao Liu, John C. Marioni, and Sarah A. Teichmann. Single cell RNA-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell Stem Cell*, 17:471–485, 2015. ISSN 19345909. doi: 10.1016/j.stem.2015.09.011. URL http://linkinghub.elsevier.com/retrieve/pii/S193459091500418X.

[38] Catalina A. Vallejos, John C. Marioni, and Sylvia Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLOS Computational Biology*, 11(6):e1004333, 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004333. URL http://dx.plos.org/10.1371/journal.pcbi.1004333.

[39] Gareth O. Roberts and Jeffrey S. Rosenthal. Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, January 2009. ISSN 1061-8600, 1537-2715. doi: 10.1198/jcgs.2009.06134.