Computational analysis of single-cell RNA-seq data: challenges, solutions and opportunities

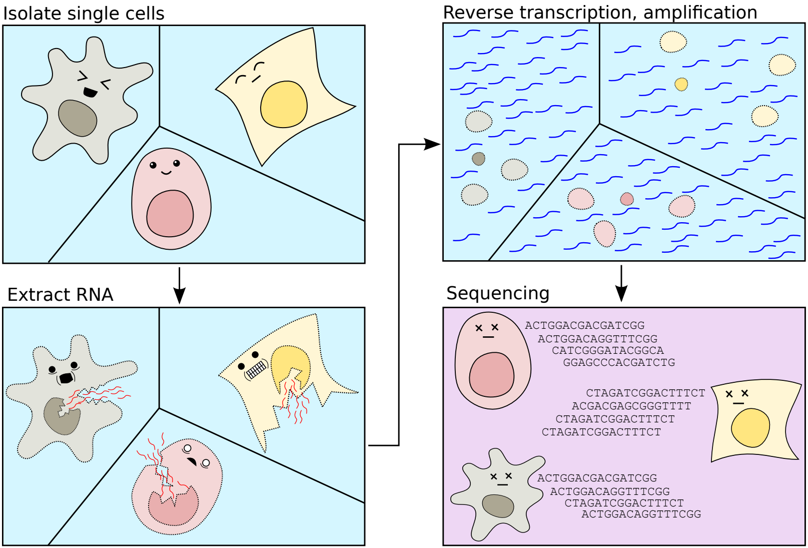
Aaron Lun

CRUK Cambridge Institute

Single-cell Analysis Workshop, TAU

22 May 2018

What is single-cell RNA sequencing (scRNA-seq)?



... using microfluidics, plate-based or droplet-based protocols

Why should we use scRNA-seq?

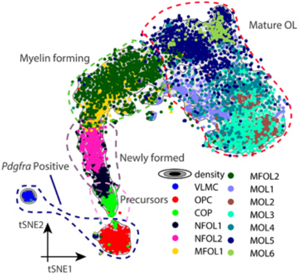
Characterize heterogeneity across a cell population using transcriptome-wide expression profiles (vs. bulk, FACS)

- identify cell “trajectories”, e.g., in differentiation
- define subpopulations at single-cell resolution
- study noise in transcriptional regulation

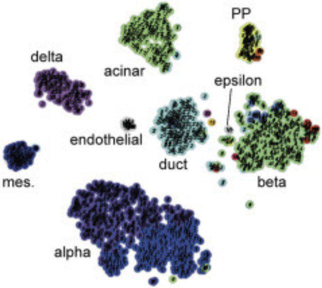
Why should we use scRNA-seq?

Characterize heterogeneity across a cell population using transcriptome-wide expression profiles (vs. bulk, FACS)

- identify cell “trajectories”, e.g., in differentiation
- define subpopulations at single-cell resolution
- study noise in transcriptional regulation



Science (2016), 352:1326-1329



Cell Systems (2016), 3:385-394.e3

What are the challenges in scRNA-seq data analysis?

What is the missing step?

scRNA-seq data → → interesting biology

What are the challenges in scRNA-seq data analysis?

What is the missing step?

scRNA-seq data → **computational analysis** → interesting biology

Generating a cDNA library from a single cell is *hard*:

- high dropout rates, i.e., molecule is present but not captured
- variable capture rates across cells
- low quality cells where mRNA is not captured or lost
- duplicated reads from PCR amplification

What is genuine biology? What is technical noise?

What does scRNA-seq data look like?

In its rawest form[1], FASTQ files after Illumina sequencing.

1. Align to reference genome (e.g., STAR)
2. Count number of reads per gene (e.g., HTSeq)

Output is a count matrix with genes as rows and cells as columns.

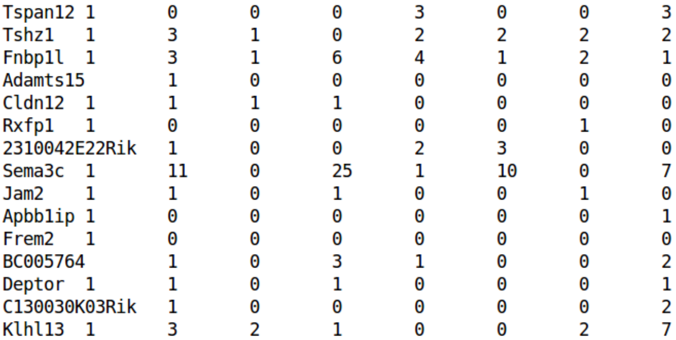
Exceptions and alternatives

- pseudo-aligners, e.g., Salmon, Kallisto
- UMI handling, e.g., with `UMI-tools`
- droplet data, e.g., `CellRanger`

[1] Excluding BCL files.

What does scRNA-seq data look like?

A typical scRNA-seq count matrix:



```
Tspan12 1       0       0       0       3       0       0       3
Tshz1   1       3       1       0       2       2       2       2
Fnbp1l  1       3       1       6       4       1       2       1
Adamts15        1       0       0       0       0       0       0
Cldn12  1       1       1       1       0       0       0       0
Rxfp1   1       0       0       0       0       0       1       0
2310042E22Rik   1       0       0       2       3       0       0
Sema3c  1       11      0       25      1       10      0       7
Jam2    1       1       0       1       0       0       1       0
Apbb1ip 1       0       0       0       0       0       0       1
Frem2   1       0       0       0       0       0       0       0
BC005764        1       0       3       1       0       0       2
Deptor  1       1       0       1       0       0       0       1
C130030K03Rik   1       0       0       0       0       0       2
Klhl13  1       3       2       1       0       0       2       7
```

~100 - 10^6 cells

~10000-40000 genes

Data from *Science* (2015), 347:1138-42

- lots of zeros due to dropout events (**or no expression!**)
- variable total counts across cells - cell-specific biases
- variable counts per gene - part biological, part technical

Performing a basic analysis of scRNA-seq data

Starting from a count matrix:

1. Quality control on the cells
2. Normalization of cell-specific biases
3. Modelling technical noise
4. Dimensionality reduction and clustering

... followed by higher-level analyses and interpretation.

Quality control on the cells

Low-quality cells generated by:

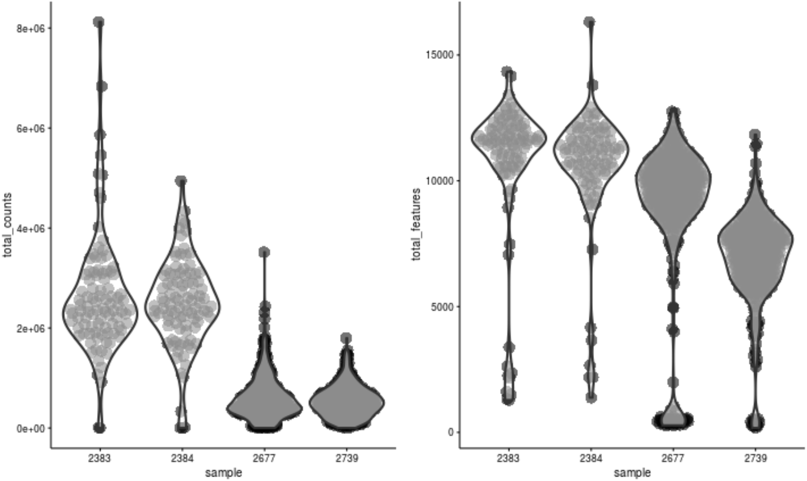
- insufficient sequencing
- failed reverse transcription
- damaged cells during dissociation

We use the following metrics to identify and remove them:

- total number of reads for each cell *(low)*
- total number of expressed features for each cell *(low)*
- percentage of reads mapped to spike-in transcripts *(high)*
- percentage of reads mapped to mitochondrial genes *(high)*

Distributions of total counts, total features

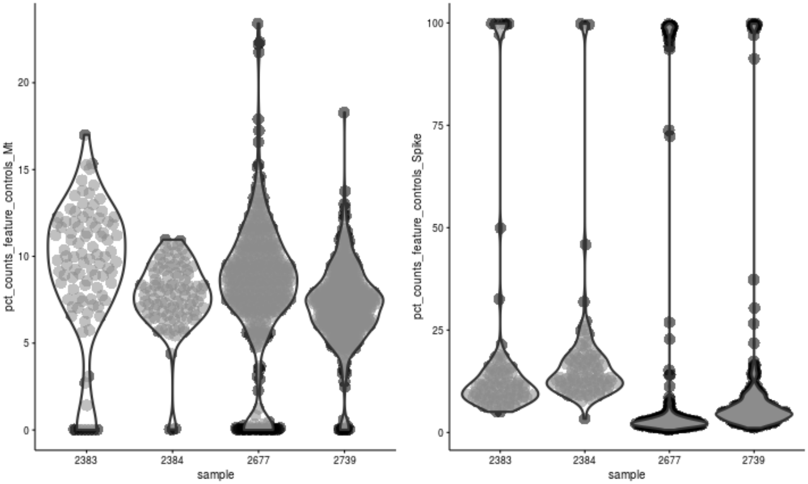
Various batches of human ESCs:



Data from Ferdinand von Meyenn and Wolf Reik at the Babraham Institute

Distributions of spike-in, mitochondrial proportions

Various batches of human ESCs:



Data from Ferdinand von Meyenn and Wolf Reik at the Babraham Institute

What is “low-quality”?

Approach 1

Define fixed thresholds, e.g., at least 100,000 counts per cell

- simple, easy to interpret
- hard to generalize across data sets

What is “low-quality”?

Approach 1

Define fixed thresholds, e.g., at least 100,000 counts per cell

- simple, easy to interpret
- hard to generalize across data sets

Approach 2

Detect outliers in the QC metric distribution: ↪ remove small outliers for total counts, features

↪ remove large outliers for % of spike-in/mitochondrial reads

- adapts to mean/variance of QC metrics across population
- assumes most cells are high-quality, homogeneous metrics

What is “low-quality”?

Approach 1

Define fixed thresholds, e.g., at least 100,000 counts per cell

- simple, easy to interpret
- hard to generalize across data sets

Approach 2

Detect outliers in the QC metric distribution: ↪ remove small outliers for total counts, features

↪ remove large outliers for % of spike-in/mitochondrial reads

- adapts to mean/variance of QC metrics across population
- assumes most cells are high-quality, homogeneous metrics

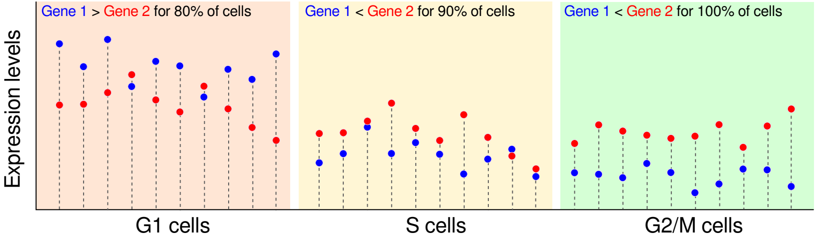
Approach 3

Define low-quality cells as outliers on gene expression

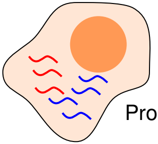
- Risky, may remove cells in rare subpopulations

Assigning the cell cycle phase (cyclone)

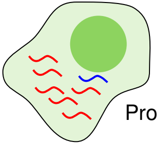
Training to identify informative gene pairs:



Cell cycle phase assignment:



Probably G1



Probably not G1

In practice:

- Use many (1000s) of gene pairs

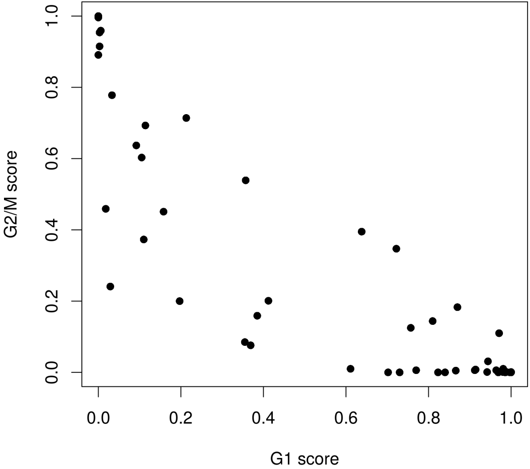
- Use different pairs for each phase

Robust to noise: only uses sign of relative expression

See *Methods* (2015), 85:54-61

Example of a phase score plot

Each point is a T-helper 2 cell: (Data from *Nat. Biotechnol.* (2015), 33:155-160)



Scores are computed from number of pairs supporting that phase.

Normalizing out cell-specific biases

Differences in library size, capture efficiency between cells

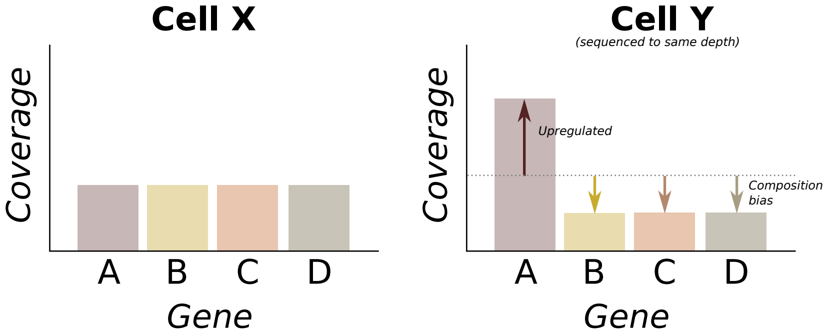
- scaling normalization to remove biases *between* cells
- compute a "size factor" to divide the counts for each cell

To demonstrate: consider counts for a few genes in a few cells

- assume X, Y, Z... are *not* DE between cells
- systematic fold-differences are technical in origin

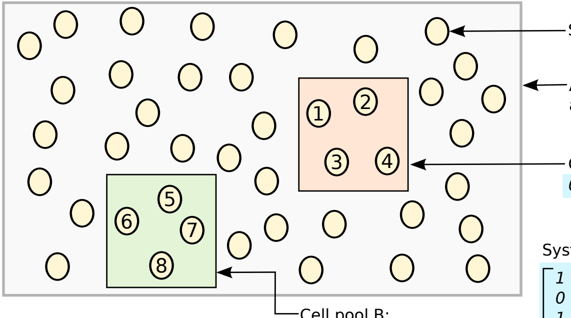
	Cell			
	A	**B**	**C**	**D**
Gene X	10	20	30	40
Gene Y	15	30	45	60
Gene Z	20	40	60	80
	...	...	...	...
Size factor	1	2	3	4

Composition biases due to differential expression



- Normalization by library size provides no protection
- Requires methods robust to DE, e.g., TMM, DESeq
- ... but such methods are not robust to zeroes!

Deconvolution: sharing information across cells



Single cell

All cells (averaged to make a reference pseudo-cell)

Cell pool A:

$\theta_1 + \theta_2 + \theta_3 + \theta_4 = \theta_A$

Cell pool B:

$\theta_5 + \theta_6 + \theta_7 + \theta_8 = \theta_B$

System of linear equations:

$$\begin{bmatrix} 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0 \ldots \\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \ldots \\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \ldots \\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0 \ldots \\ \ldots \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \ldots \end{bmatrix} = \begin{bmatrix} \theta_A \\ \theta_B \\ \theta_C \\ \theta_D \\ \ldots \end{bmatrix}$$

Genome Biol. (2016), 17:75

- Pooling cells to increase counts, avoid problems with zeros.
- Size factor per pool estimated robustly, to protect against DE.
- Solve linear system to obtain a size factor **per cell**.

Normalizing spike-ins separately

Normalization on gene counts corrects for RNA content

- counts for spike-in transcripts not affected by RNA content
- using gene-based size factors will “over-normalize”

Before	Cell A	Cell B
Gene X	10	20
Gene Y	20	40
Gene Z	30	60
Spike 1	5	5

After	Cell A	Cell B
Gene X	20	20
Gene Y	40	40
Gene Z	60	60
Spike 1	10	5

Define sum of spike-in counts as “spike-in size factor”:

- normalize spike-ins by dividing with spike-in size factor
- normalize genes by dividing with gene-based size factor

Normalizing spike-ins separately

Normalization on gene counts corrects for RNA content

- counts for spike-in transcripts not affected by RNA content
- using gene-based size factors will “over-normalize”

Before	Cell A	Cell B
Gene X	10	20
Gene Y	20	40
Gene Z	30	60
Spike 1	5	5

After	Cell A	Cell B
Gene X	20	20
Gene Y	40	40
Gene Z	60	60
Spike 1	10	5

Define sum of spike-in counts as “spike-in size factor”:

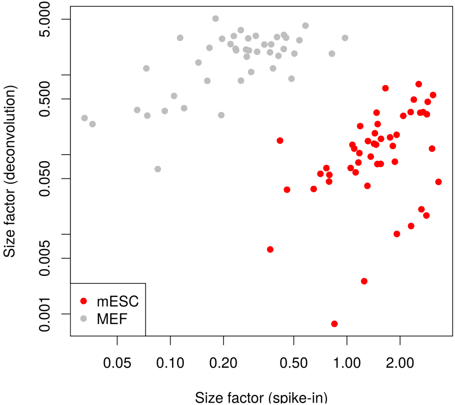
- normalize spike-ins by dividing with spike-in size factor
- normalize genes by dividing with gene-based size factor

Do not use the gene-based size factors on the spike-in counts

Spike-in versus gene-based normalization

Alternatively: normalize genes with spike-in size factors

- when you can't assume most genes are not DE
- when changes in total RNA content are interesting



Data from *Genome Res.* (2011), 21:1160-1167

Computing normalized log-expression values

For gene g in cell i, divide count y_{ig} by size factor s_i to get:

$$\log_2\left(\frac{y_{ig}}{s_i} + 1\right)$$

- differences between log-values represent log-fold changes
- more relevant than absolute differences in counts

	Cell A	Cell B
Gene X	1000	1100
Gene Y	0	20

Related to the concept of "variance stabilization"

Modelling technical and biological variance

How much of cell-to-cell variability is technical vs biological?

$\hookrightarrow$ how do we quantify variance in the first place?

Squared coefficient of variation (CV^2)

Divide variance of (normalized) counts by the squared mean:

$$\frac{\text{var}(z_{ig})}{\bar{z}_g^2} \quad \text{where} \quad z_{ig} = \frac{y_{ig}}{s_i}$$

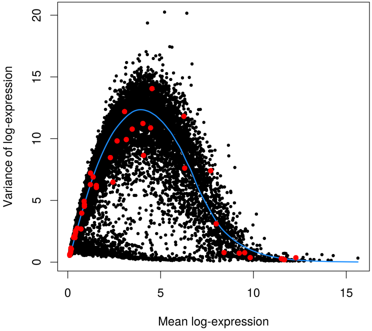
Variance of log-expression

Compute variance of normalized log-expression values:

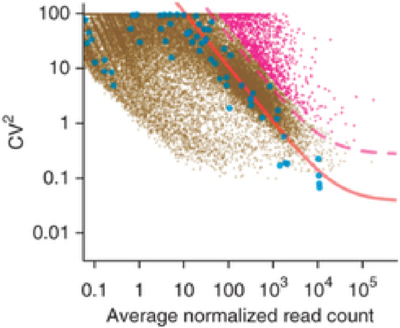
$$\text{var}[\log_2(y_{ig}/s_i + 1)]$$

CV^2, logging try to eliminate the mean-variance trend...

Fitting a trend to the technical variance



Data from *Cell Stem Cell* (2015), 16:712-724



Nat. Methods (2013), 10:1093-1095

Fit a trend to the variance of the spike-ins:

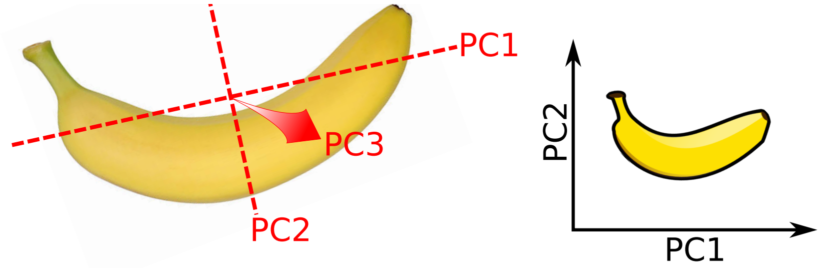
- quantify variance due to technical noise only.
- biological variance = residual from the trend for each gene

Identify interesting genes for downstream steps = feature selection.

Dimensionality reduction with PCA

PCA = principal components analysis

- identifies axes of maximal variance in high-dimensional data
- each principal component (PC) explains less variance

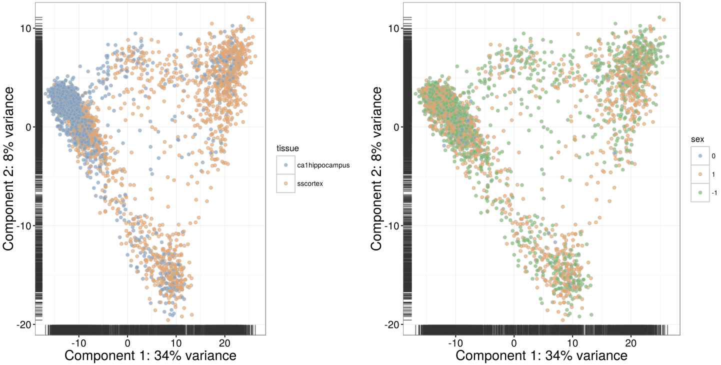


Use the first few (5-100) PCs as a “summary” of the data

- Speed up downstream analyses by reducing dimensionality
- Focus on biology, remove random noise in later PCs

Visualization with PCA

The first 2-3 PCs can be directly used for visualization:



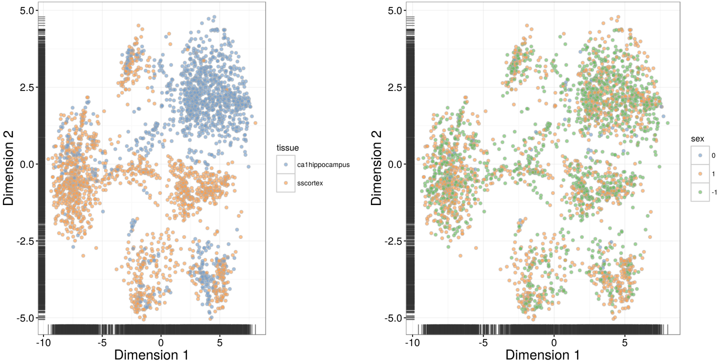
Data from *Science* (2015), 347:1138-42

Simple and efficient, but limited resolution of complex structure.

Visualization with *t*-SNE

Finds a low-dimensional representation of high-dimensional data

- preserve distances to neighbouring cells
- non-linear: not limited to straight axes



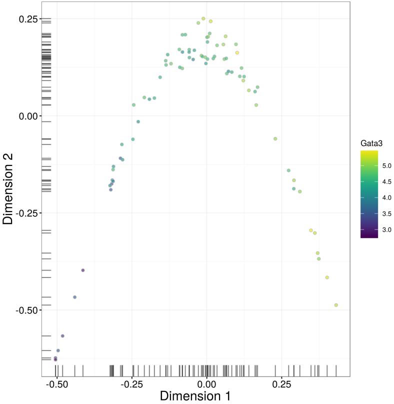
Data from *Science* (2015), 347:1138-42

Powerful, but need to fiddle with random seed and perplexity

Dimensionality reduction: diffusion maps

Uses a diffusion process to model a continuum of expression

- useful for trajectories (e.g., differentiation)



(Data from *Nat. Biotechnol.* (2015), 33:155-160)

... and a lot more, e.g., SOMs, force directed graphs.

A few words on clustering

Aim: To group cells with similar expression profiles
↪ identify and characterize new subpopulations

Lots of algorithms:

- hierarchical flavours
- *k*-means
- community detection (graph-based)

Lots of distance metrics:

- Euclidean
- cosine (i.e., Pearson's correlation, Spearman's rho)

Most methods work well, provided you:

- filter to only use features of interest
- assess cluster separatedness (silhouette width, gap statistic)
- experimentally validate putative clusters.

Wrapping up

Starting from a count matrix:

1. Quality control on cells
2. Normalization of cell-specific biases
3. Modelling technical noise
4. Dimensionality reduction and clustering

... followed by higher-level analyses and interpretation.

Try it yourself!

What's on the horizon?

- Dealing with batch effects (see `?mnnCorrect`)
- Handling huge data sets, e.g., 10X 1.3M neurons
- Integrative analysis of multi-condition data sets

Acknowledgements

John Marioni (boss)
Davis McCarthy (*scater*)
Karsten Bach (deconvolution)
Antonio Scialdone (`cyclone`)
Ferdinand von Meyenn and Wolf Reik (data)

CANCER
RESEARCH
UK

UNIVERSITY OF
CAMBRIDGE