

# Computational Genomics Workshop

September 10 & 11, 2018

## Introduction to single-cell RNA-seq analysis

---

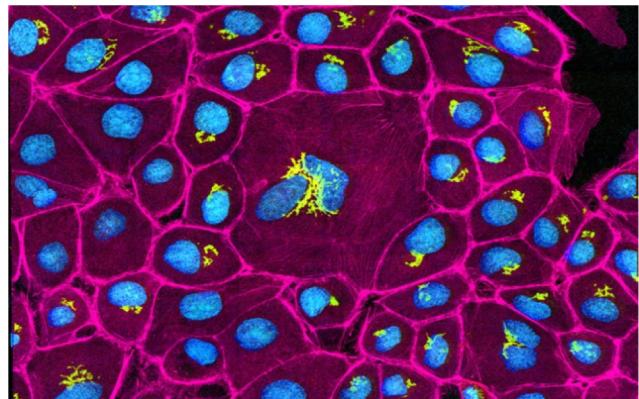
Orr Ashenberg  
Caroline Porter  
Ayshwarya Subramanian

# Goals for today

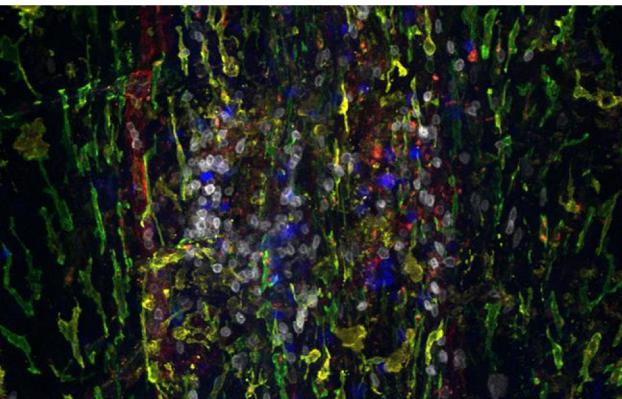
- Give an overview of single-cell RNA-seq data analysis
- Provide hands on experience stepping through an analysis pipeline, including performing quality control and identifying cell type subsets
- Introduce you to the Seurat analysis package

# Incredible diversity in cell types, states, & interactions across human tissues

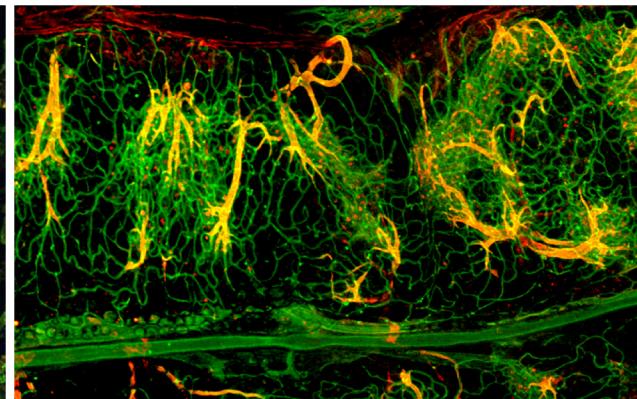
Skin epithelium



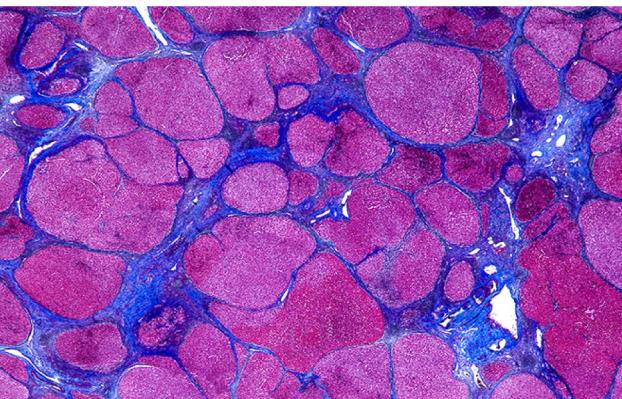
Brain meninges



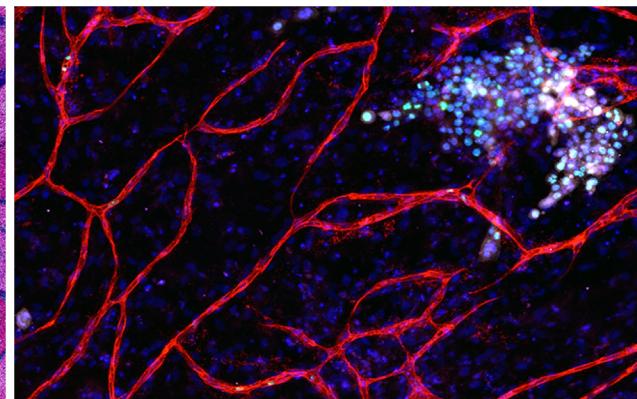
Blood vessels



Small intestine

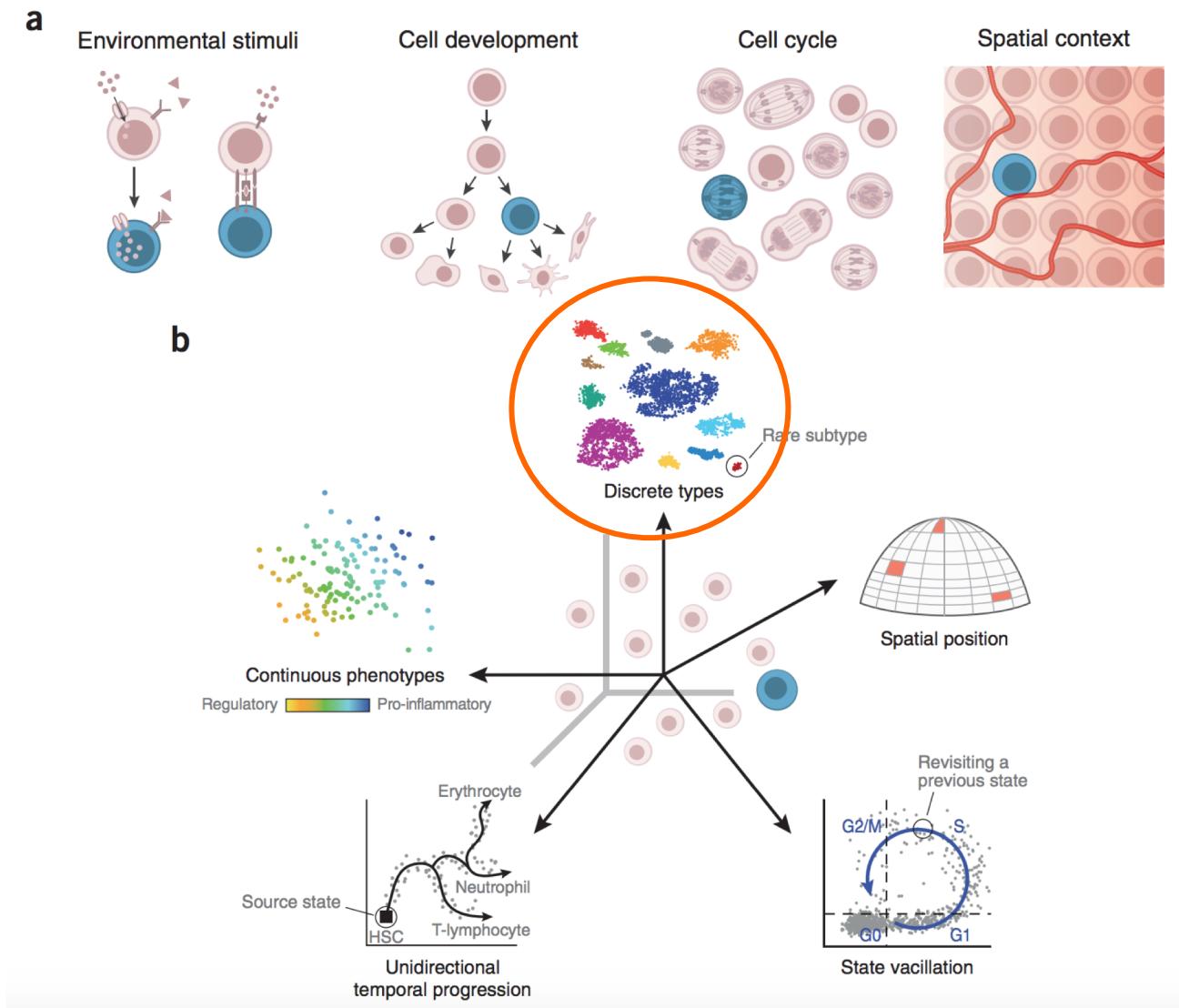


Liver cirrhosis

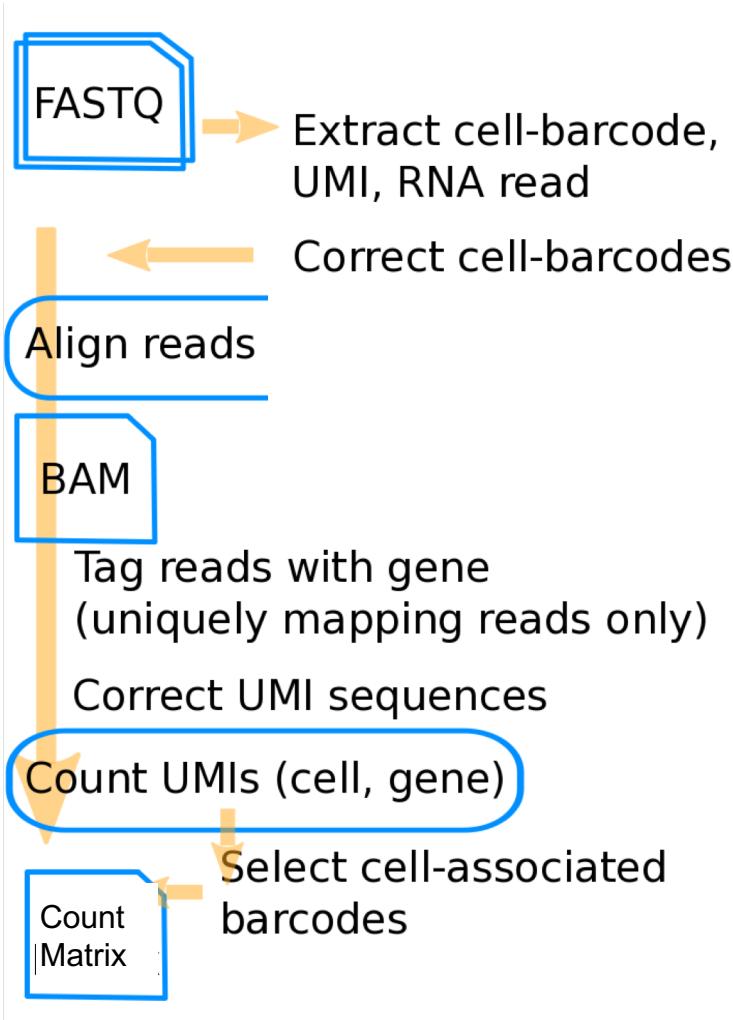


Breast cancer

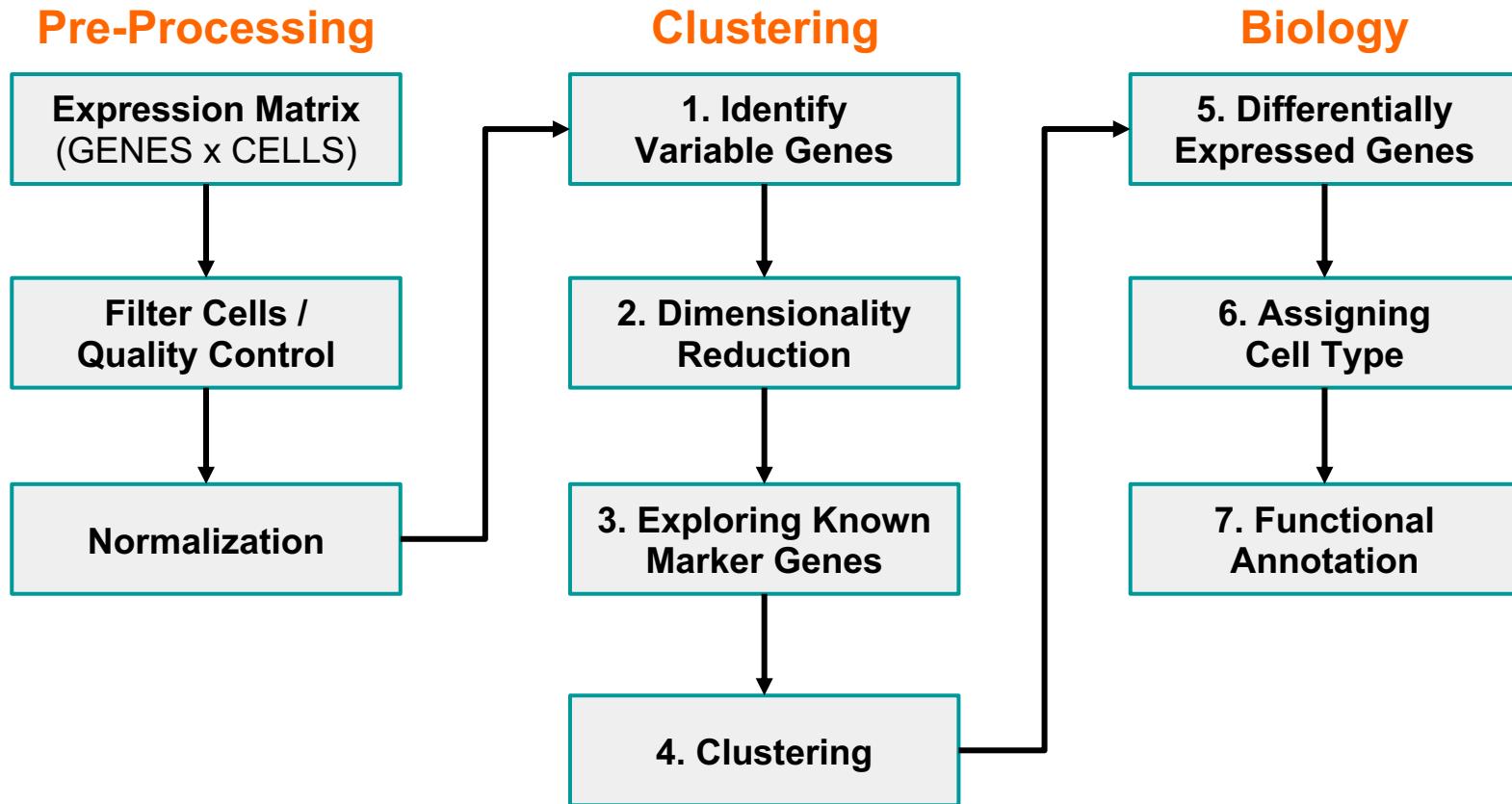
# The identity and fate of a cell are shaped by many features



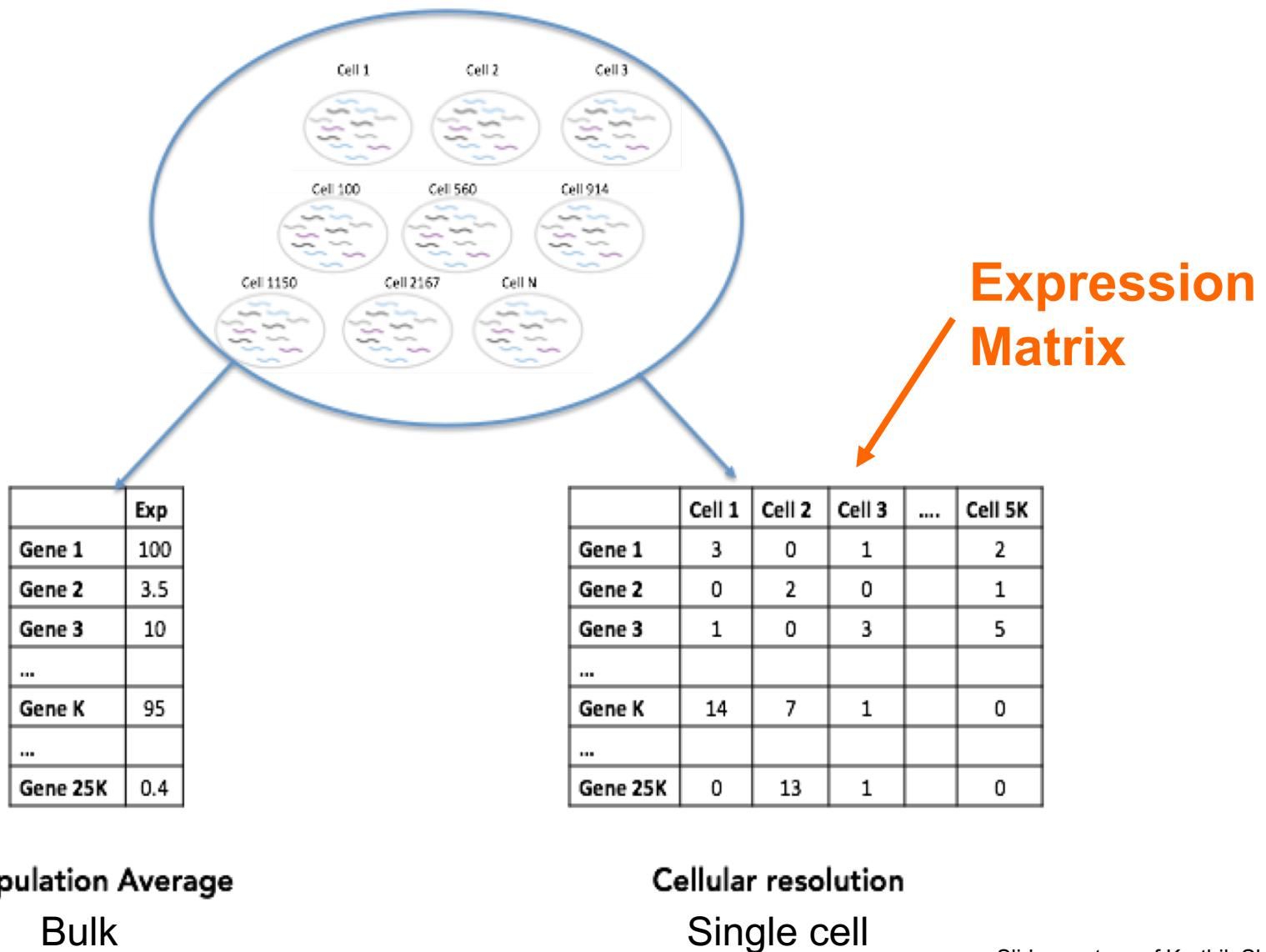
# Single-cell RNA-seq analysis pipeline: Generating the count matrix



# Single-cell RNA-seq analysis pipeline: Analyzing the expression data



# Single-cell and bulk gene expression distributions are very different

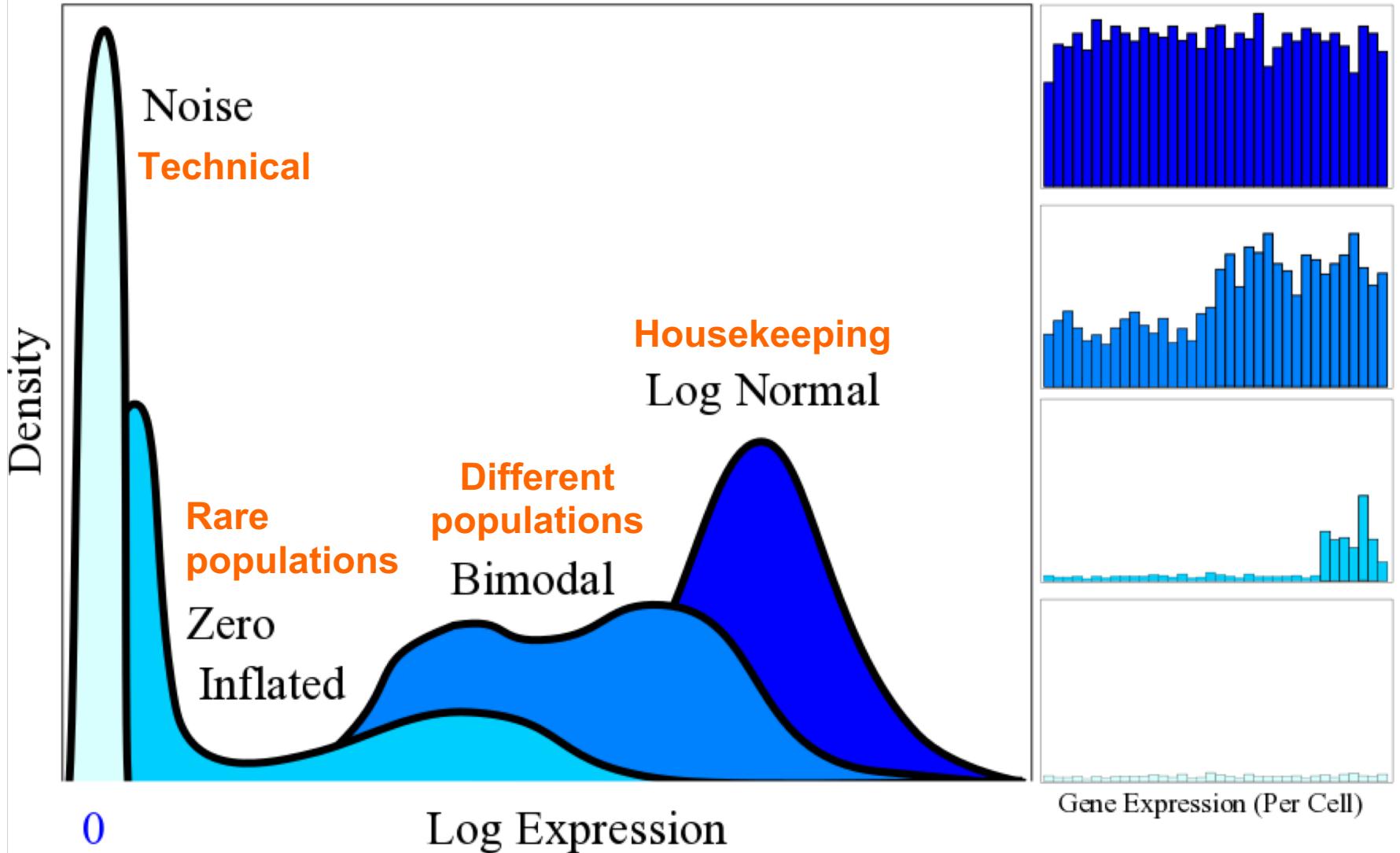


Population Average  
Bulk

Cellular resolution  
Single cell

# Genes have different distributions

Distribution of Expression of a Gene throughout a Study

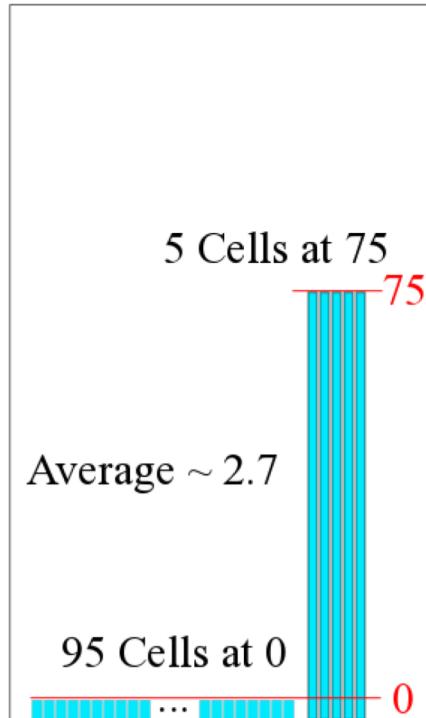


# Some single-cell RNA-seq data challenges to remember

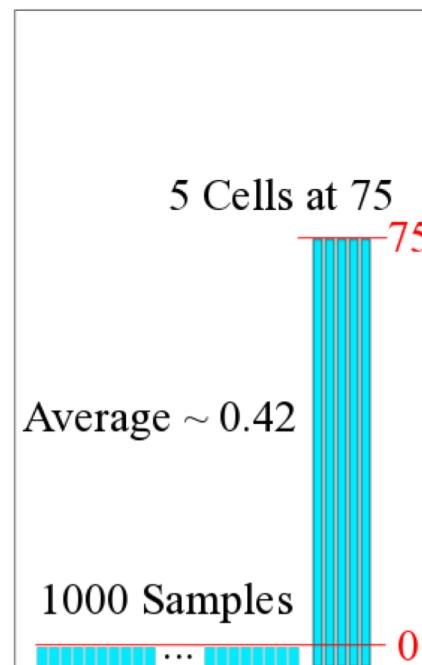
- **Drop out:** data has an excessive amount of zeros due to limiting mRNA

Zero expression doesn't mean the gene isn't on

0s pull down average



Amount of 0s is arbitrary  
(study size, diversity)

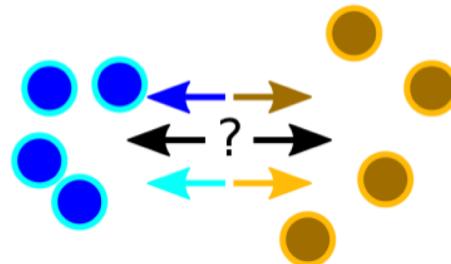


# Some single-cell RNA-seq data challenges to remember

- **Confounding:** quality control metrics have the potential to be confounded with biology

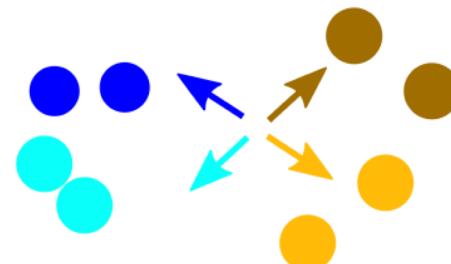
Cell | Site | Treatment

1	Main	A
2	Main	A
3	Main	A
4	Main	A
5	Remote	B
6	Remote	B
7	Remote	B
8	Remote	B



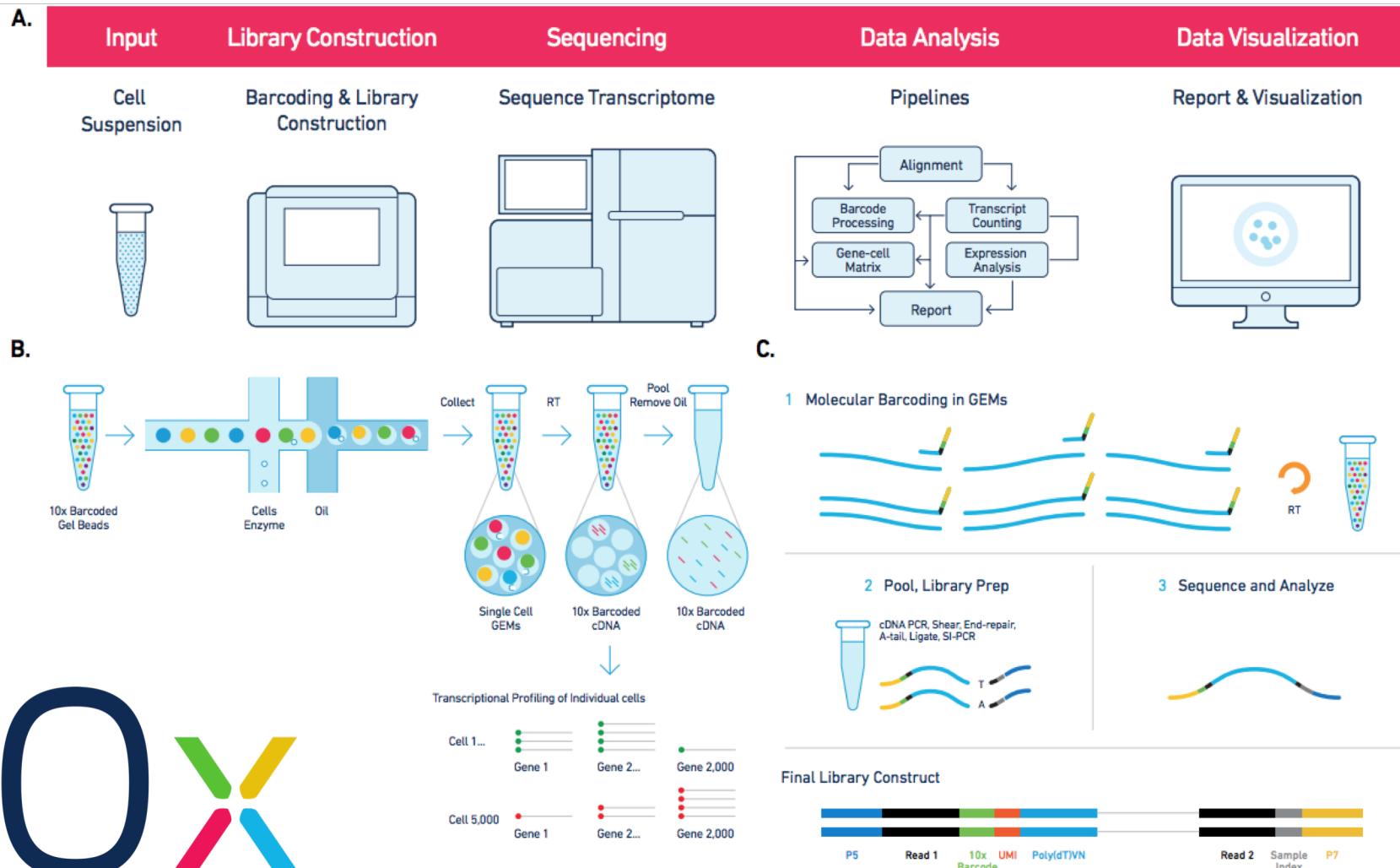
Cell | Site | Treatment

1	Main	A
2	Main	A
3	Main	B
4	Main	B
5	Remote	A
6	Remote	A
7	Remote	B
8	Remote	B



Batch effects can be removed from the data if the batch effect isn't completely confounded with biology

# 10x Genomics: The Chromium Single Cell Gene Expression Solution



# RStudio: integrated development environment for R

The screenshot shows the RStudio interface divided into four quadrants:

- Scripts:** Top-left pane showing an R script named "Untitled1" with code demonstrating various features.
- Environment:** Top-right pane showing the Global Environment with a variable "x" assigned the value 4.
- R Session:** Bottom-left pane showing the R console output, including the R startup message, license information, and a few commands run in the session.
- Help:** Bottom-right pane showing the documentation for the "log" function, specifically the "Logarithms and Exponentials" section.

Large red text labels are overlaid on each quadrant:

- Scripts
- Environment
- R Session
- Help

The RStudio logo is also visible in the top right corner of the interface.

# Seurat

<https://seurat.readthedocs.io>

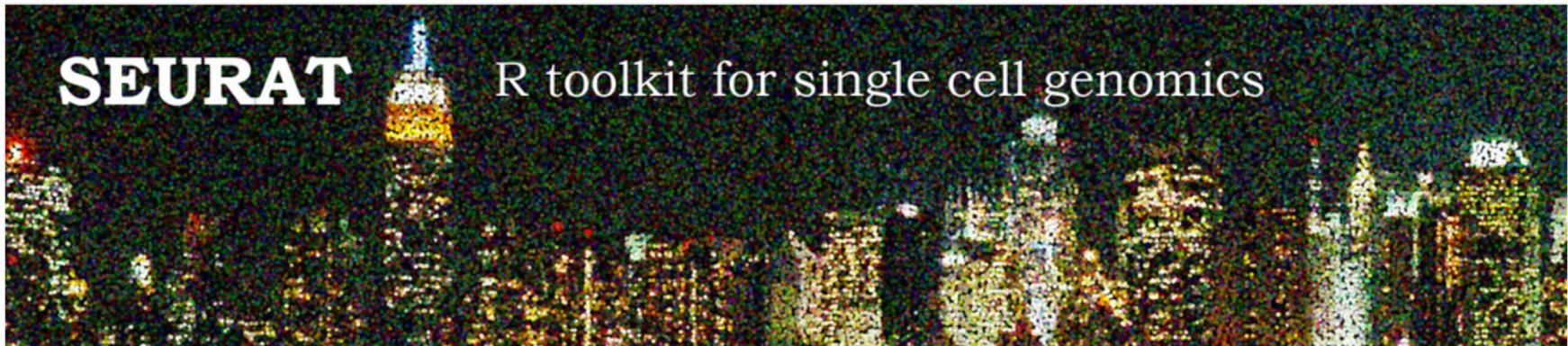
  

# SATIJA LAB

[HOME](#)   [NEWS](#)   [PEOPLE](#)   [RESEARCH](#)   [PUBLICATIONS](#)   [SEURAT](#)   [JOIN/CONTACT](#)

# SEURAT

## R toolkit for single cell genomics



## About

## Install

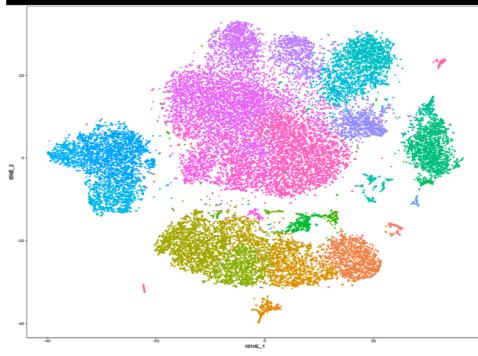
## Get Started

## Frequently Asked Questions

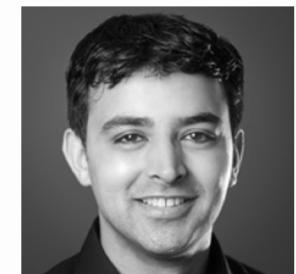
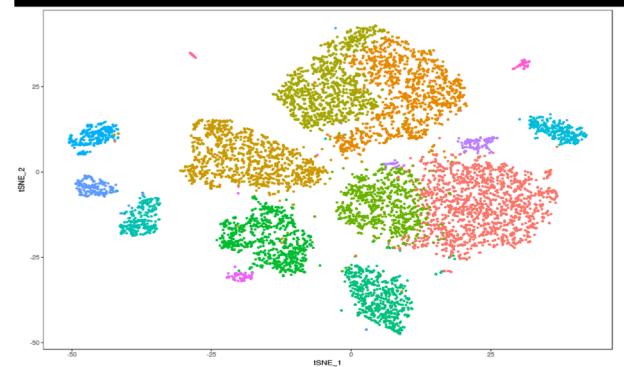
## Frequently Requested Vignettes

Contact

## Command List --- 33,000 PBMCs



## Command List --- 8,500 Pancreas Cells



Rahul Satija

D. Phil, Statistics  
Oxford University  
[rsatija@nygenome.org](mailto:rsatija@nygenome.org)

# PAGODA/SCDE

<http://hms-dbmi.github.io/scde/pagoda.html>



## SCDE

The SCDE package implements a set of statistical methods for analyzing single-cell RNA-seq data, including differential expression analysis (*Kharchenko et al.*) and pathway and geneset overdispersion analysis (*Fan et al.*)

[Home](#)

[Package](#)

[Tutorials](#)

[Help](#)

[View on GitHub](#)

[Kharchenko Lab](#)



### Peter Kharchenko (PI)

Assistant Professor of Biomedical Informatics | DBMI | HSCI

phone: 617-432-7377

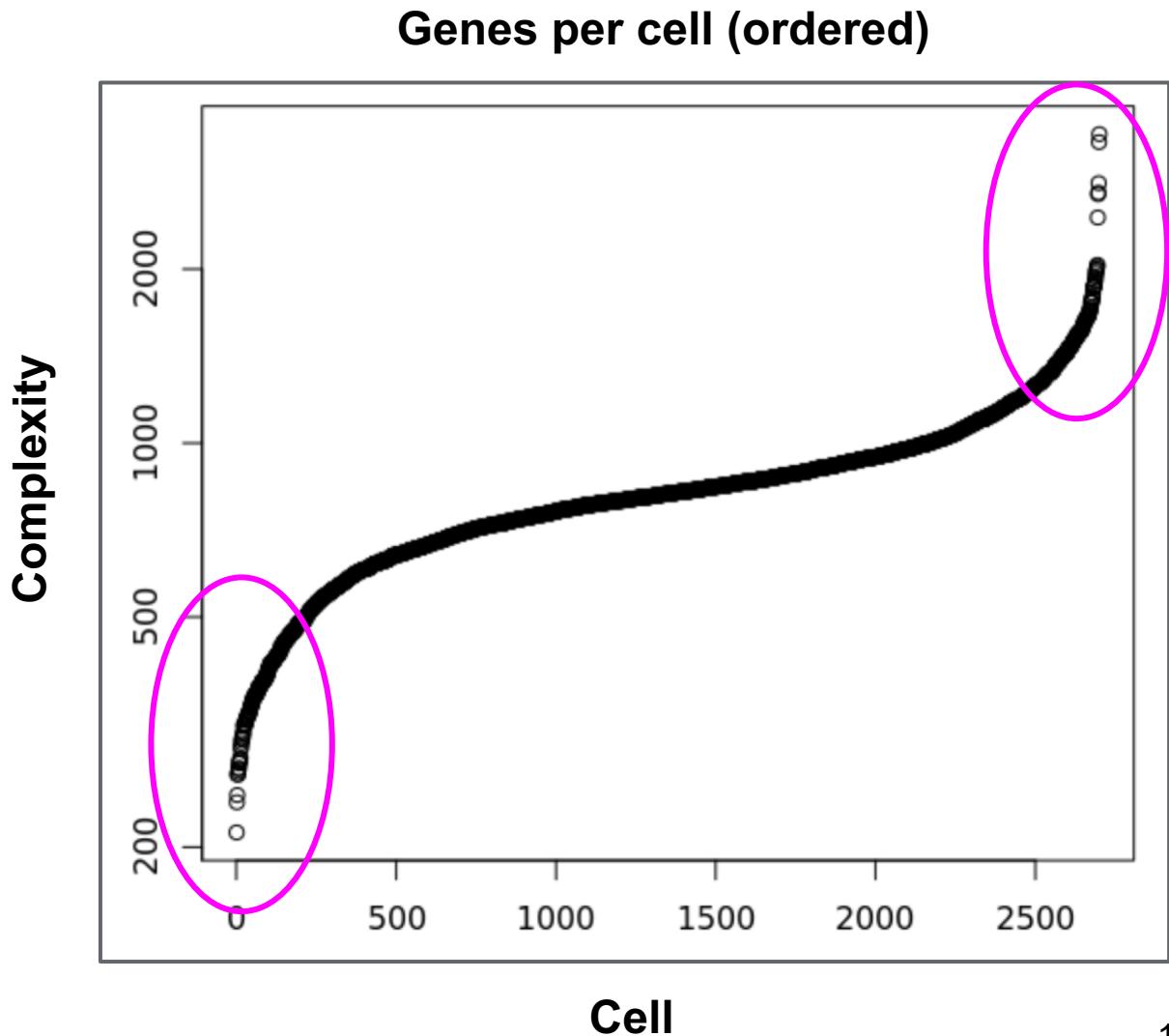
e-mail: [peter.kharchenko@post.harvard.edu](mailto:peter.kharchenko@post.harvard.edu)

office: Countway room 312, 10 Shattuck St., Boston, MA 02115

Peter received a PhD in [Biophysics at Harvard University](#), studying gene regulation and metabolic networks under the advisement of [George Church](#). He then completed a four-year postdoctoral fellowship in computational biology and genomics in the laboratory of [Peter Park](#).

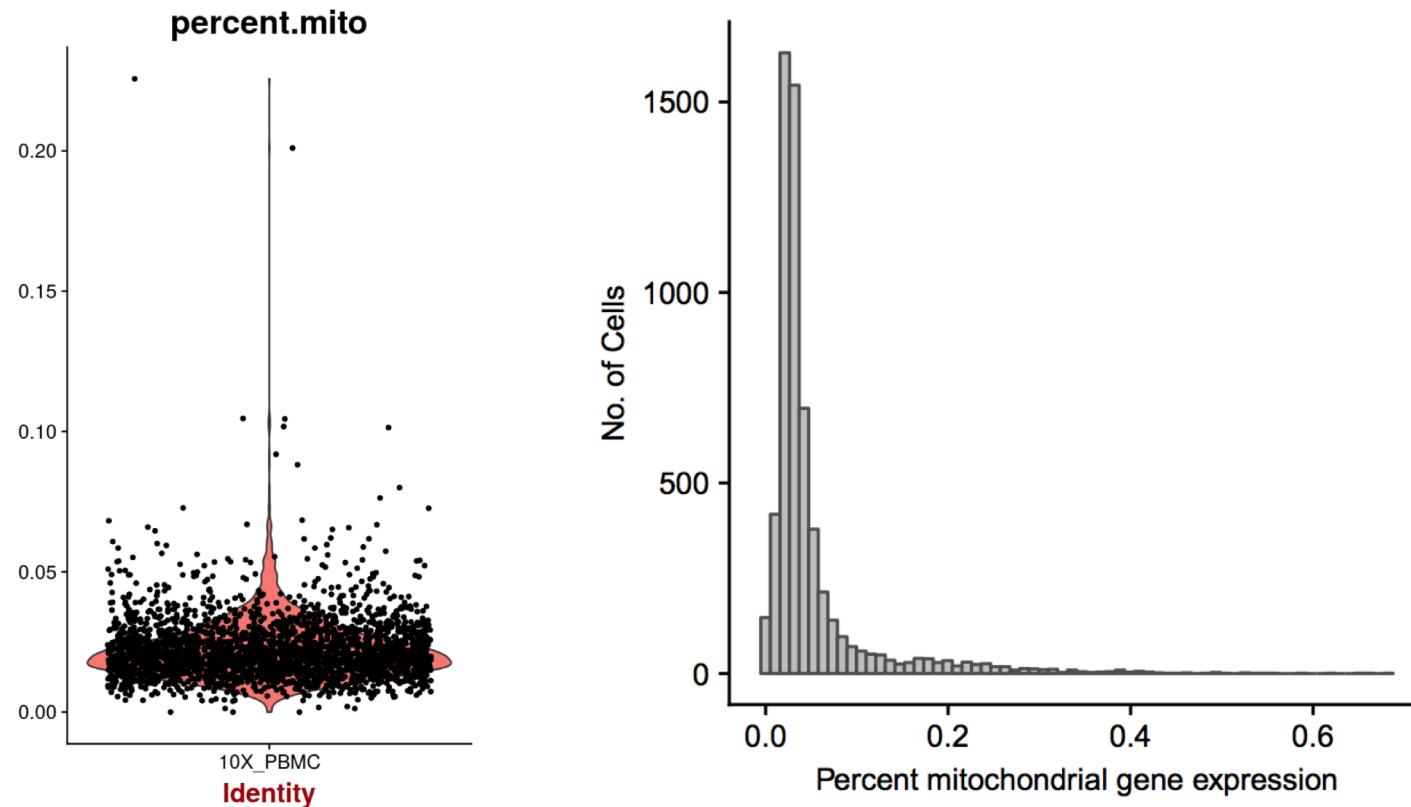
# Filtering and quality control: Number of genes per cell

complexity =  
number of genes  
detected in a cell



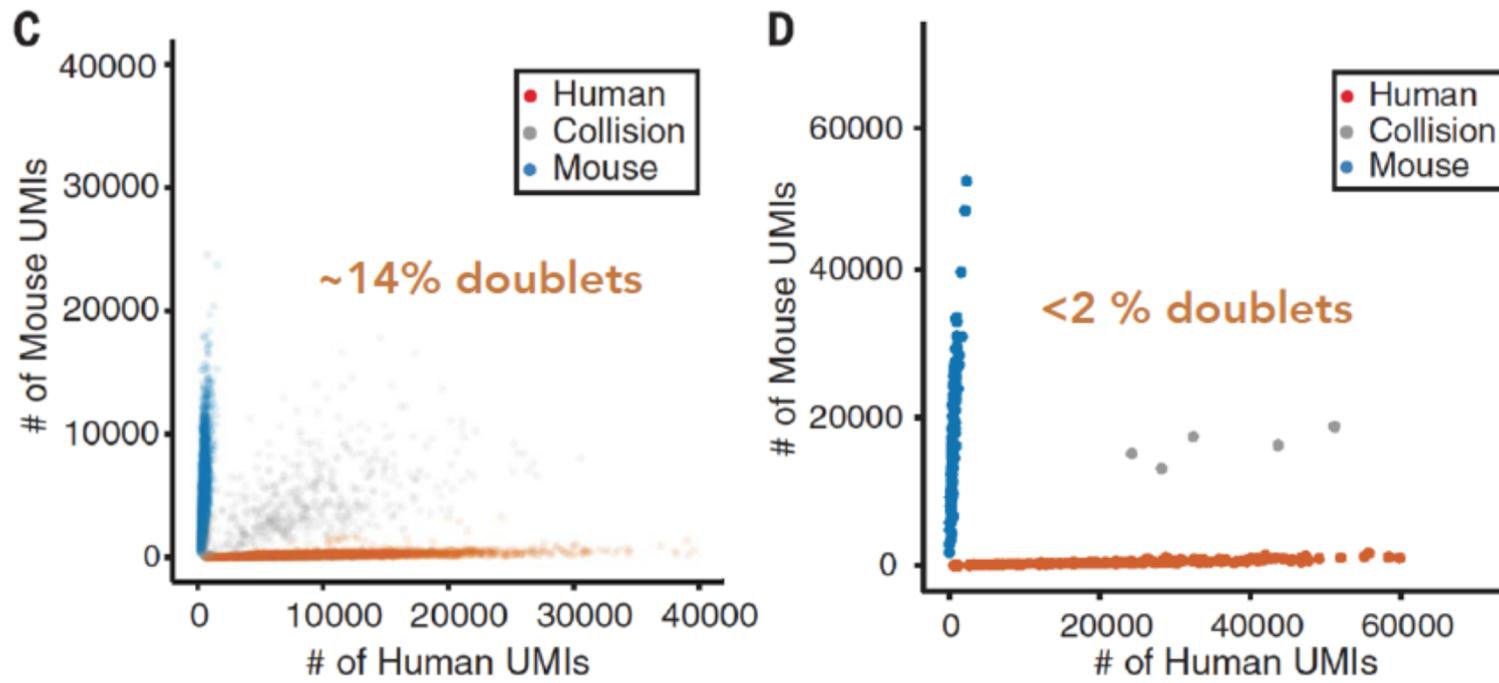
# Filtering and quality control: Mitochondrial gene expression

Percent of reads in a cell coming from mitochondrial genes is a good measure of cell quality - high mitochondrial gene expression indicates stressed cells (e.g., from damage during tissue dissociation)



# Filtering and quality control: Doublets - what are they?

Because of the setup, it is possible that two or more cells can enter the same droplet. Studies estimate doublet frequency through a “mixed-species” experiment



The doublet frequency is +vely correlated with throughput

# Filtering and quality control: Doublets - resources for identifying them

Most simple way to filter for doublets is to choose an upper threshold on the number of genes or counts per cell in your data - a doublet (which is two cells viewed as one) should in theory have a lot more genes and counts than other cells

More sophisticated way to remove doublets is to use a package for identifying doublets, such as:

<https://github.com/JonathanShor/DoubletDetection>

<https://github.com/AllonKleinLab/scrublet>

<https://www.biorxiv.org/content/early/2018/06/20/352484>

**DoubletFinder: Doublet detection in single-cell RNA sequencing data using artificial nearest neighbors**

 Christopher S McGinnis,  Lyndsay M Murrow,  Zev J Gartner

**doi:** <https://doi.org/10.1101/352484>

# Data normalization and scaling

**Typically, we:**

- Normalize gene expression for each cell by total expression and multiply by a scale factor

Objective is to have relative gene expression to eliminate technical factors that impact the variation in the number of molecules per cell

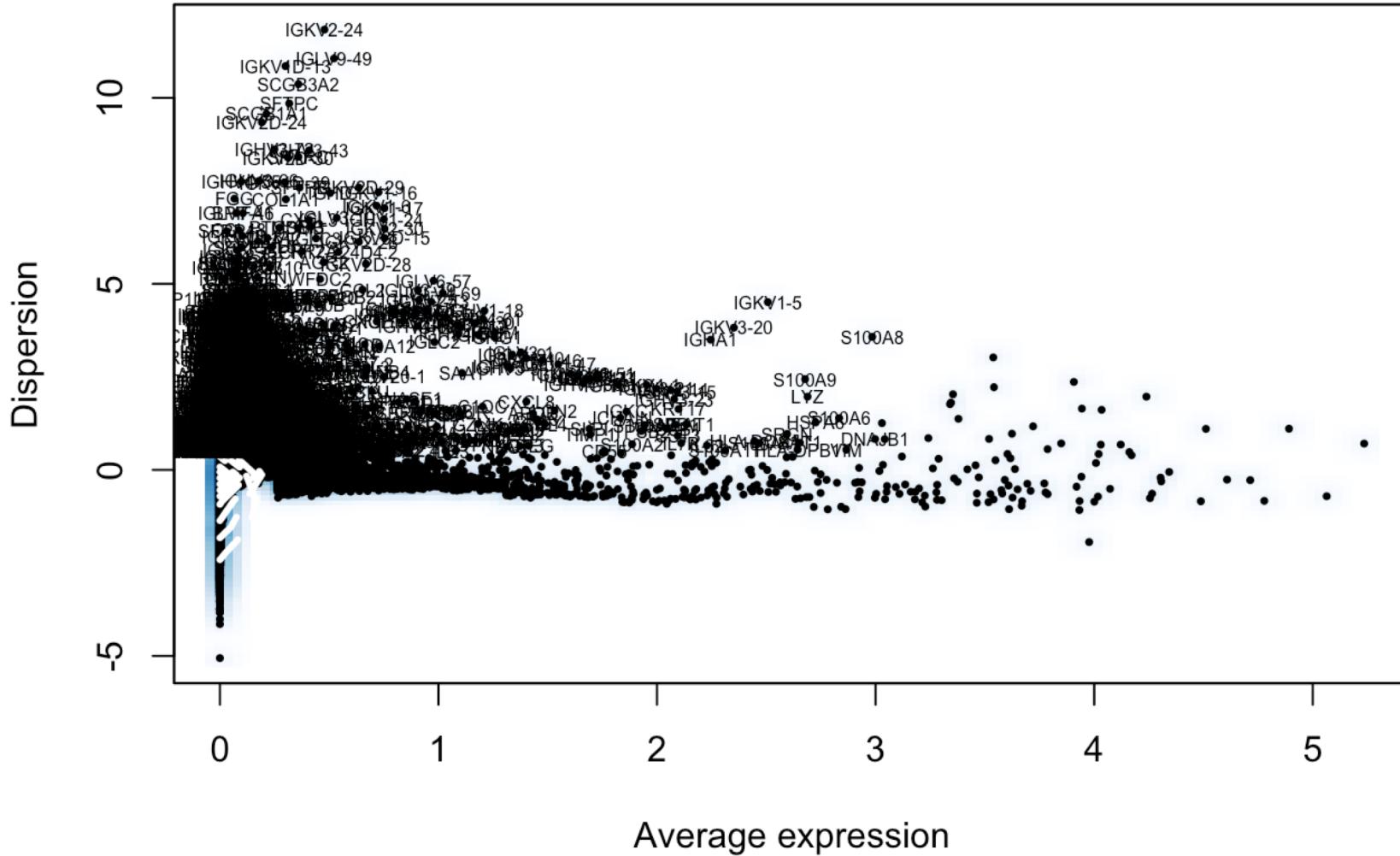
As a caution, there are biological factors that can impact this variation too

- Log transform the resulting normalized expression

Helps get rid of extreme values in the data

# Determining cell type, state, and/or function:

## 1. Identifying highly variable genes



# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

**Cells are in 20,000 dimensional space**

- many genes are lowly detected / noisy measurements
- genes are not independent of one another! rather they operate in coregulatory modules

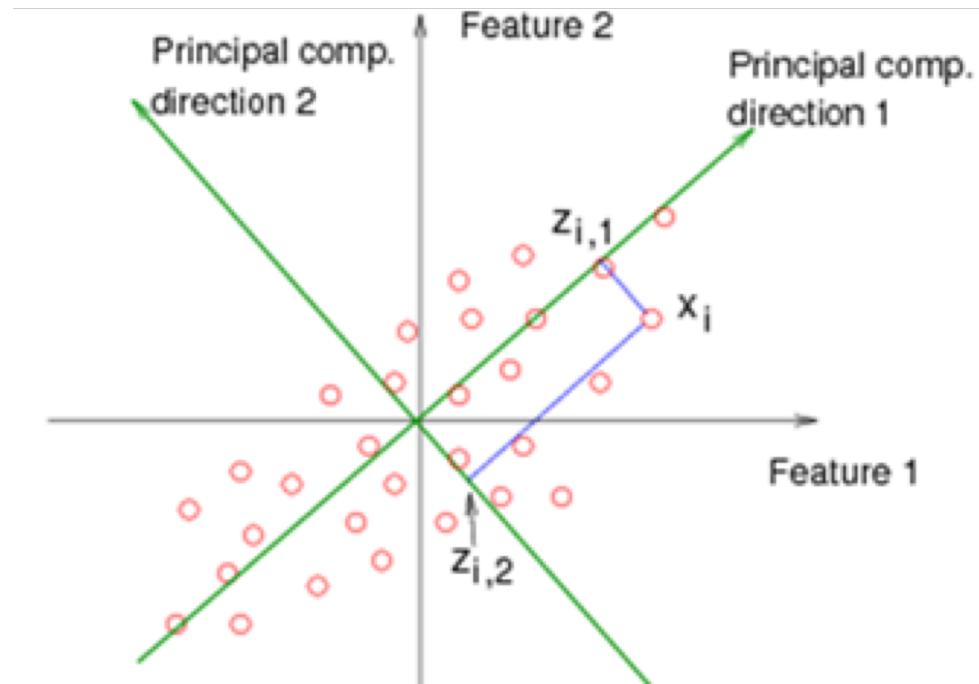
**Principal component analysis (PCA) moves us from describing cells with 20,000 gene expression values to 10-100 principal component scores**

*\*\* Note that the first principal component often captures technical variability*

# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

- PCA is a dimensionality reduction method that transforms a set of observations into a set of linearly uncorrelated variables called principal components
- The first principal component contains the most variance, and each component after contains as much variance while still being orthogonal to other components

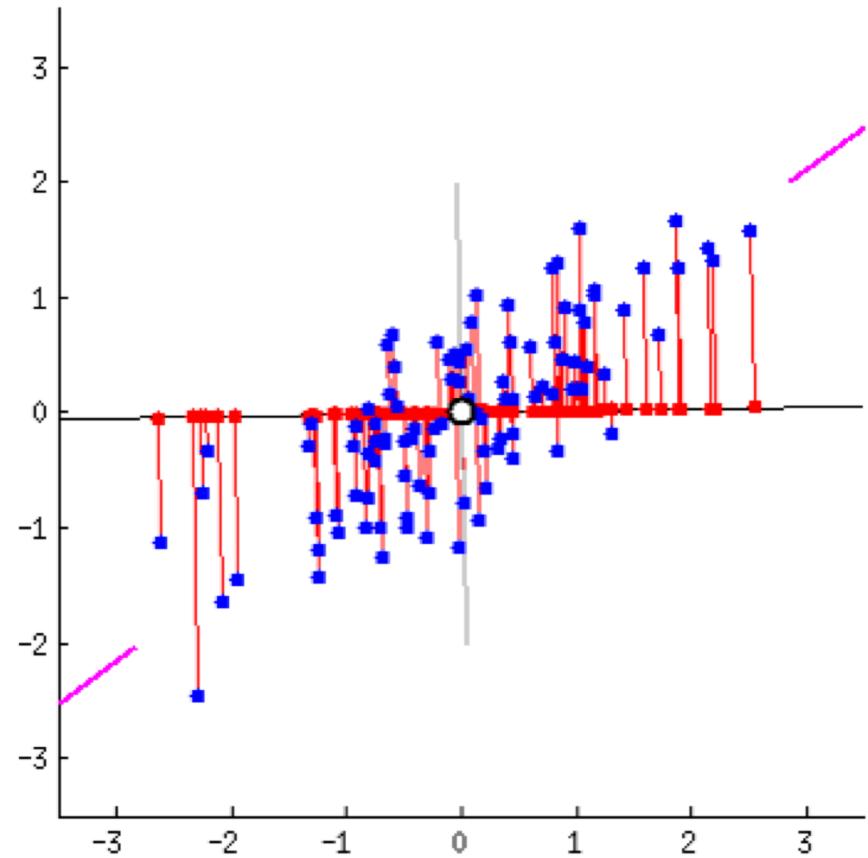


# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

- PCA is a dimensionality reduction method that transforms a set of observations into a set of linearly uncorrelated variables called principal components
- The first principal component contains the most variance, and each component after contains as much variance while still being orthogonal to other components

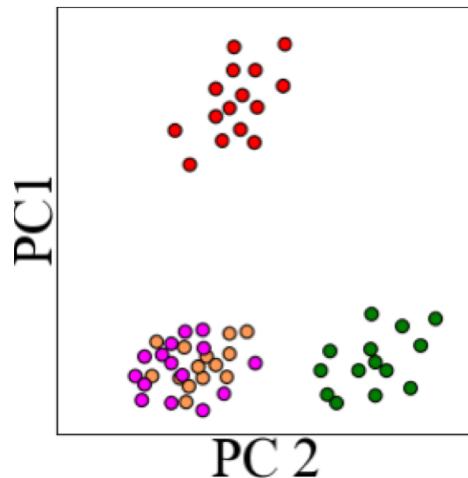
Identifying maximal orthogonal sources of variation



# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

PCA of  
single cell  
data

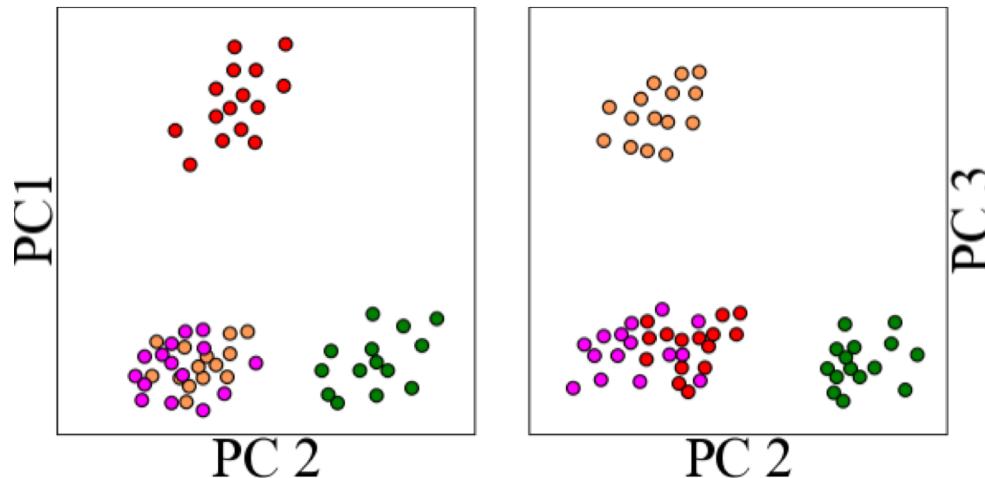


- PC1 separates the red cells from the pink, orange, and green cells
- PC2 separates the green cells from the red, pink, and orange cells

# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

PCA of  
single cell  
data

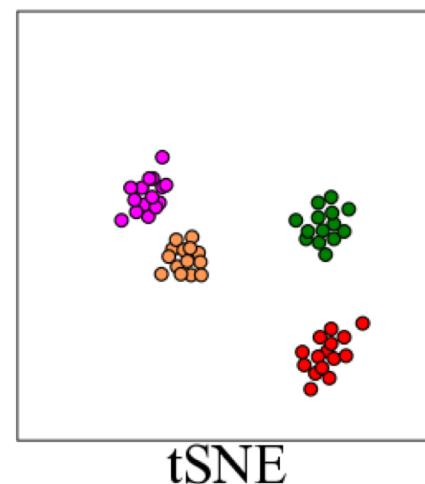
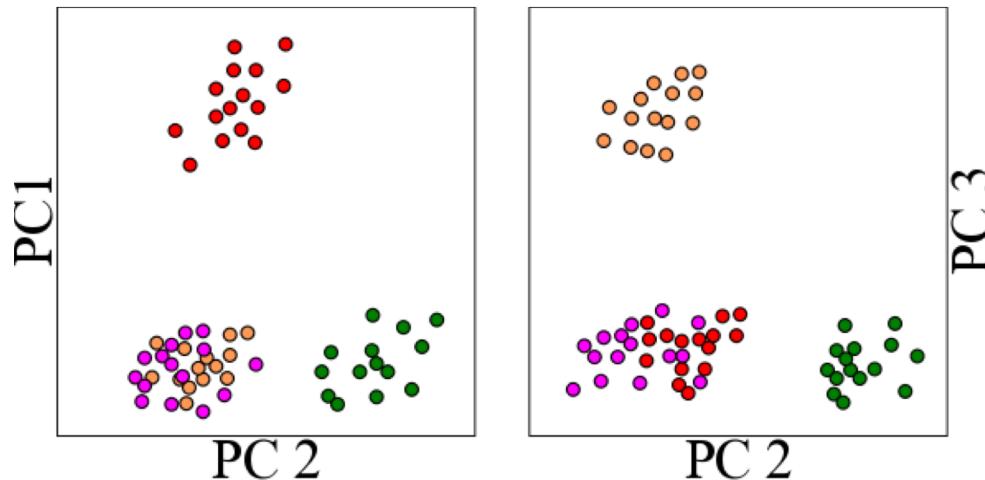


- PC3 further splits off the orange cells

# Determining cell type, state, and/or function:

## 2. Dimensionality reduction

PCA of  
single cell  
data

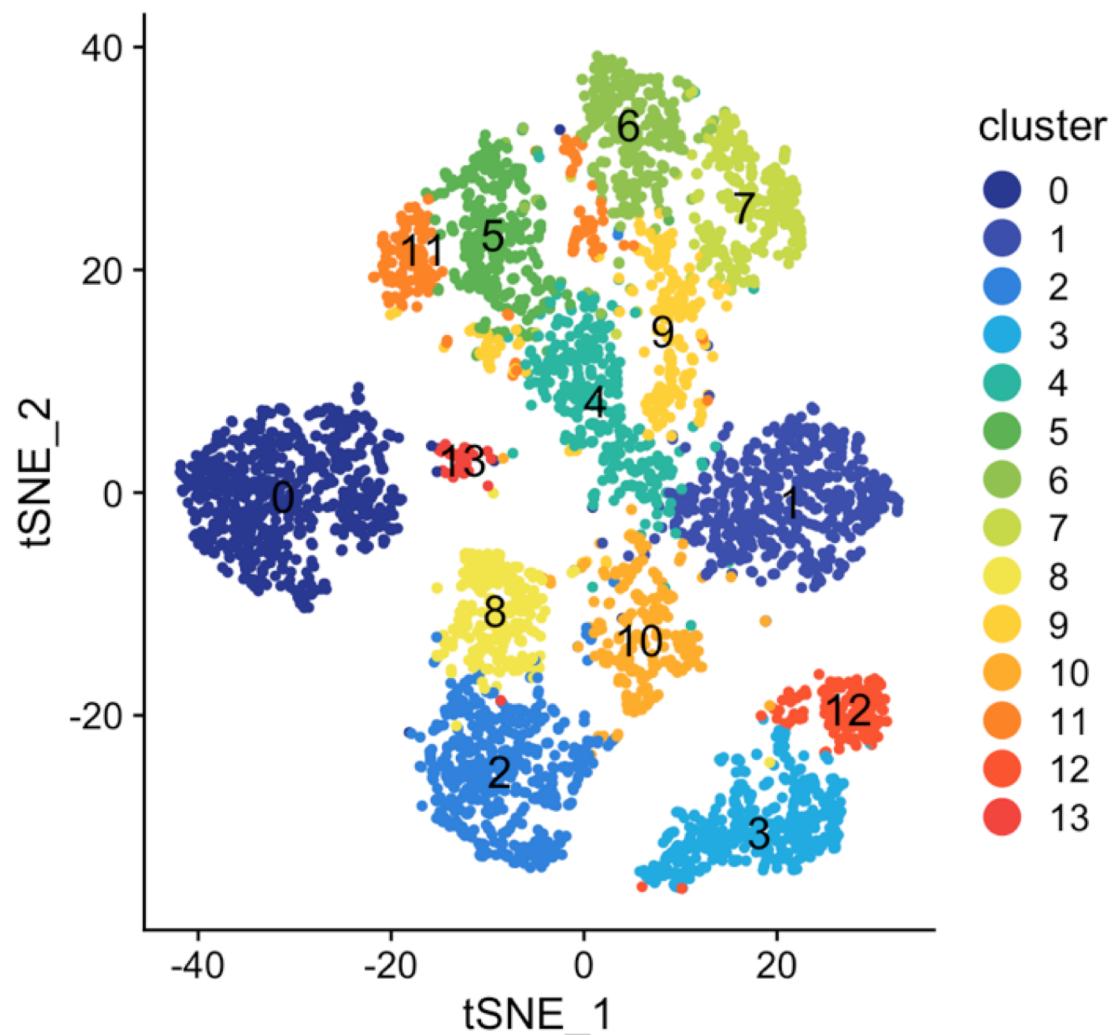


tSNE: t-  
distributed  
Stochastic  
Neighbor  
Embedding

- tSNE is nonlinear dimensionality reduction
- tSNE collapse the visualization to 2D

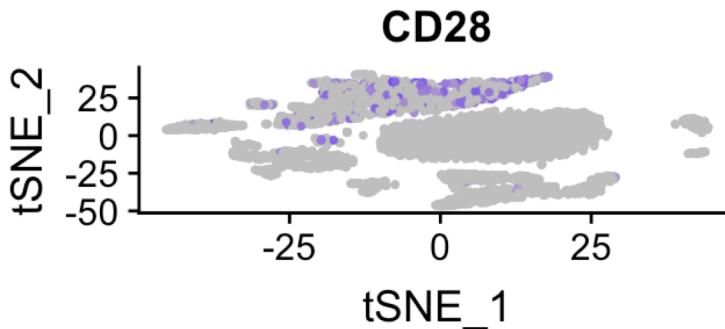
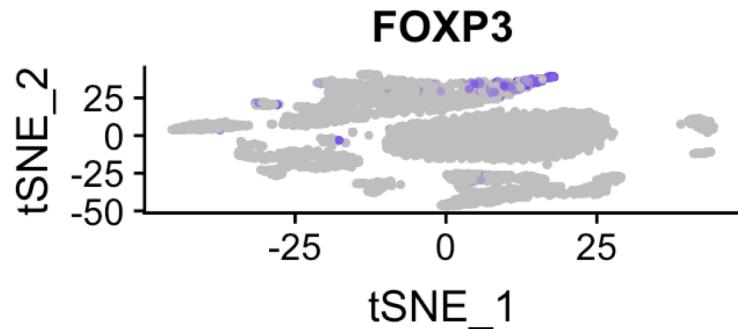
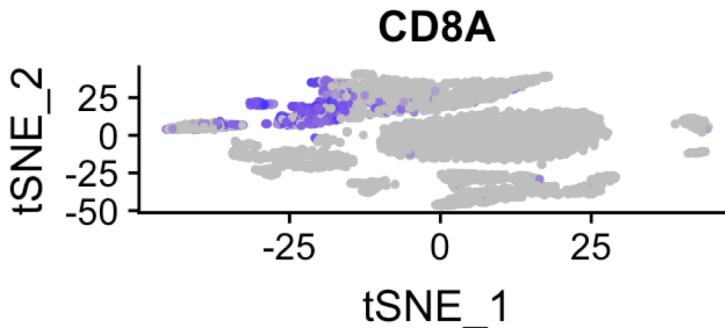
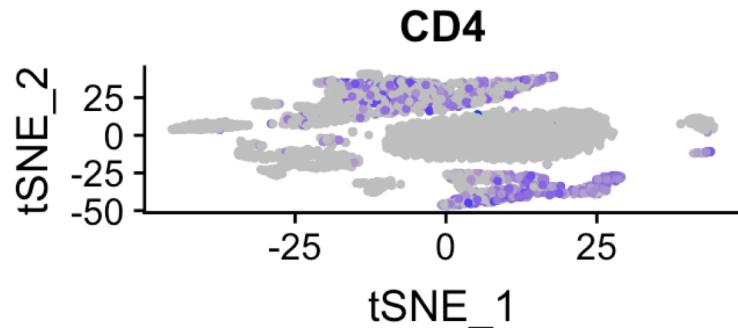
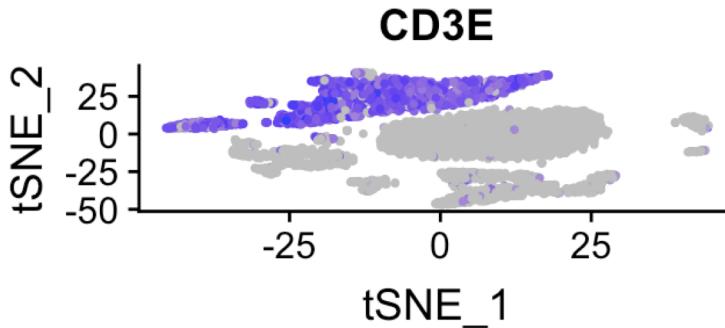
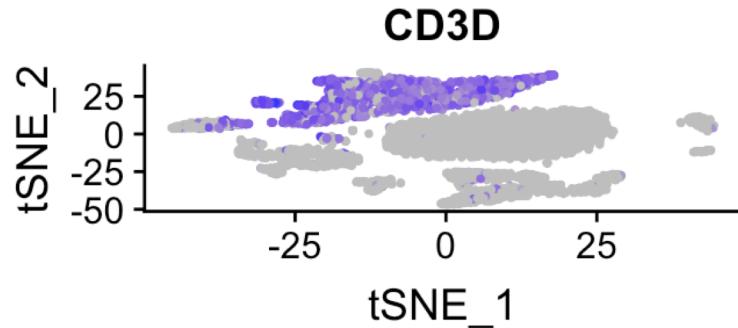
# Determining cell type, state, and/or function:

## 2. Dimensionality reduction



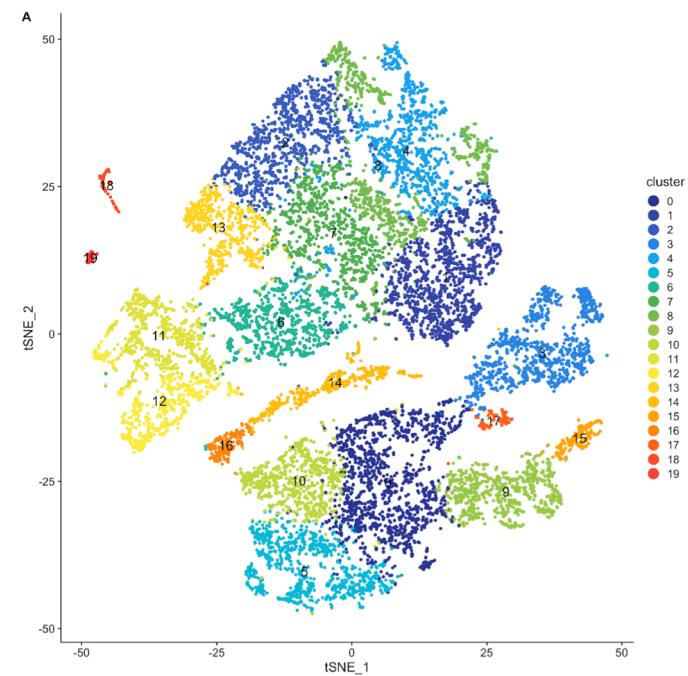
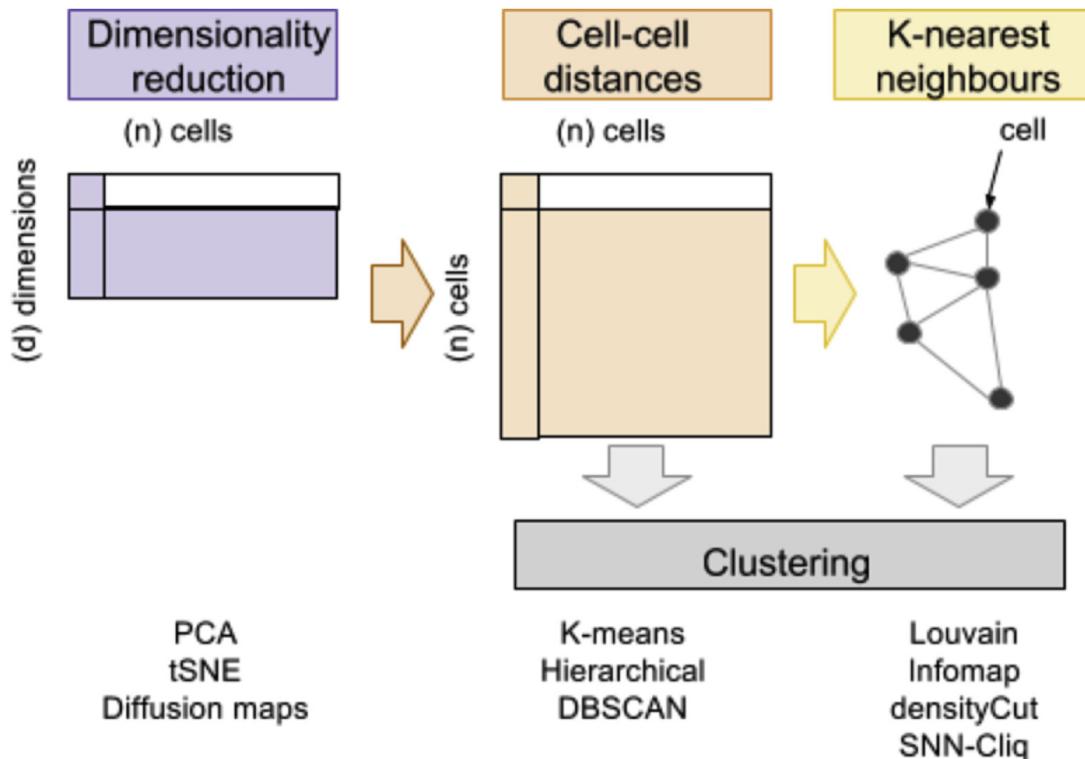
# Determining cell type, state, and/or function:

## 3. Exploring expression of marker genes



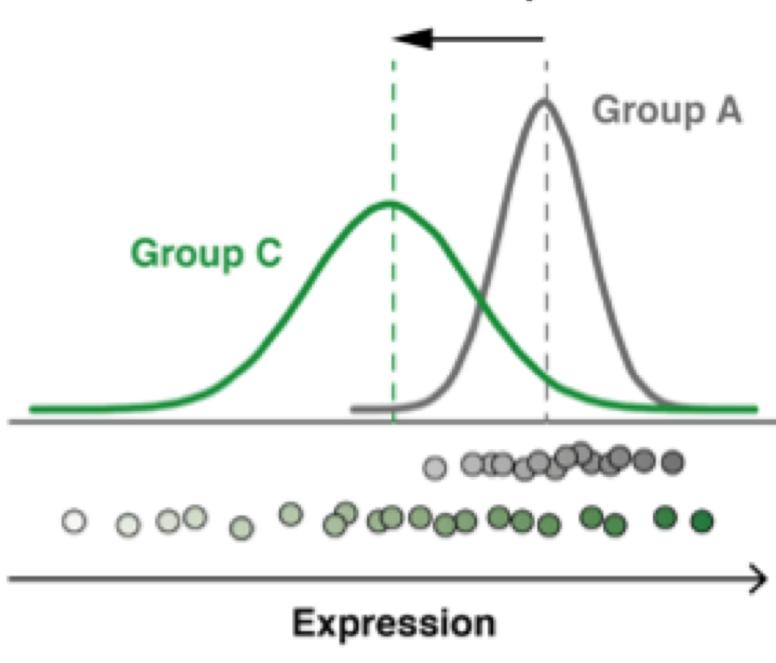
# Determining cell type, state, and/or function:

## 4. Clustering principal components



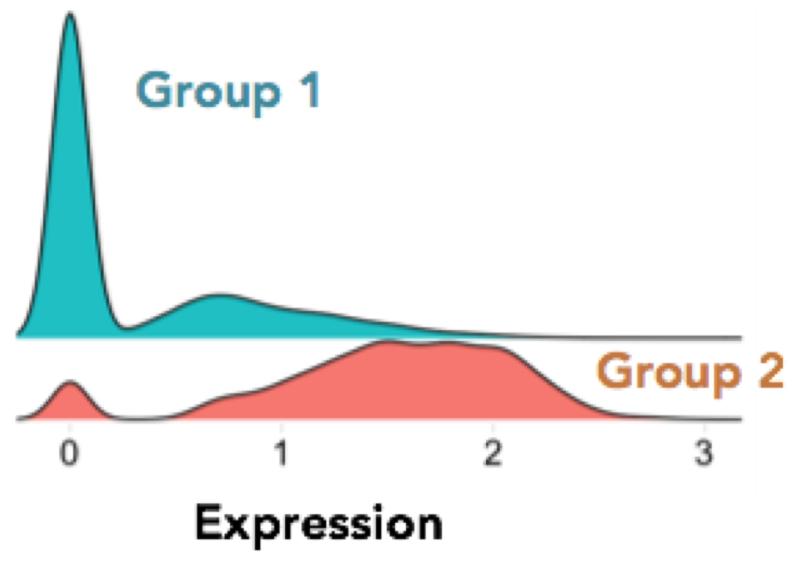
# Determining cell type, state, and/or function:

## 5. Identifying differentially expressed genes



Bulk

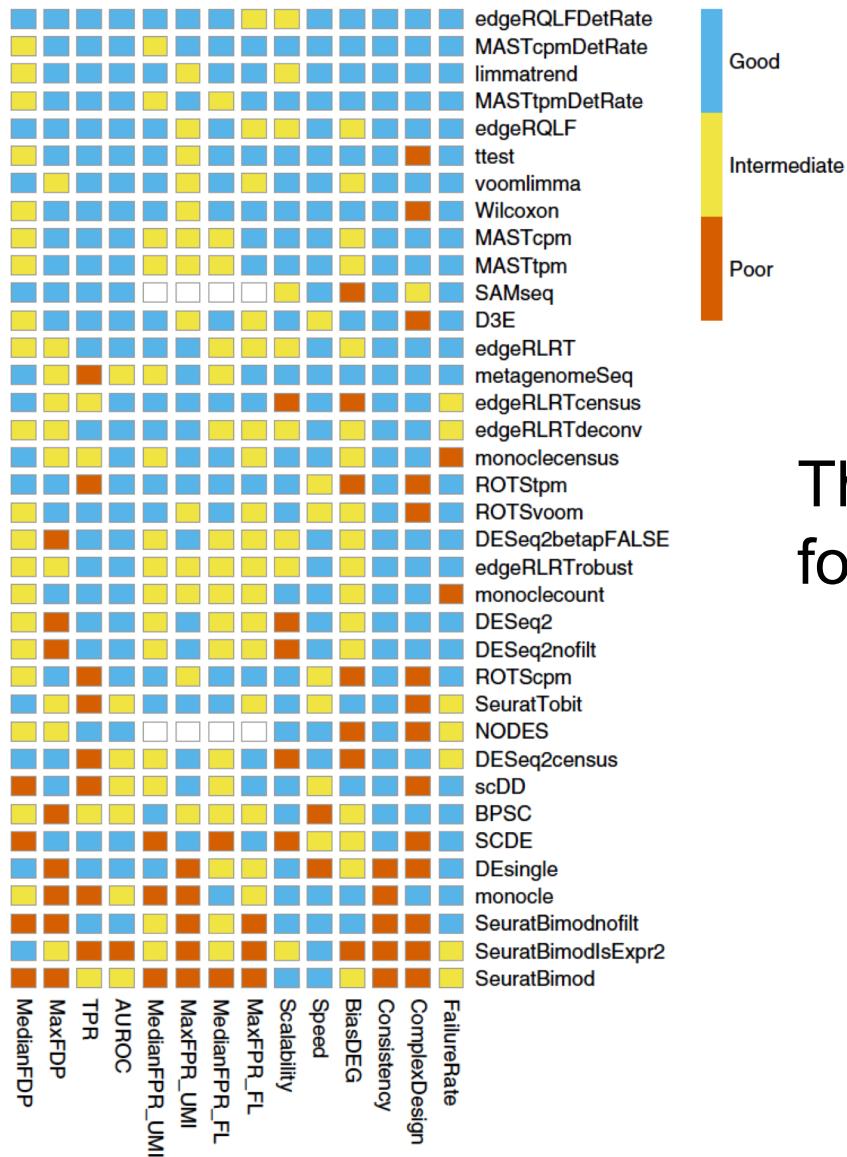
"Zero inflation" poses a challenge in single-cell data!



Single cell

# Determining cell type, state, and/or function:

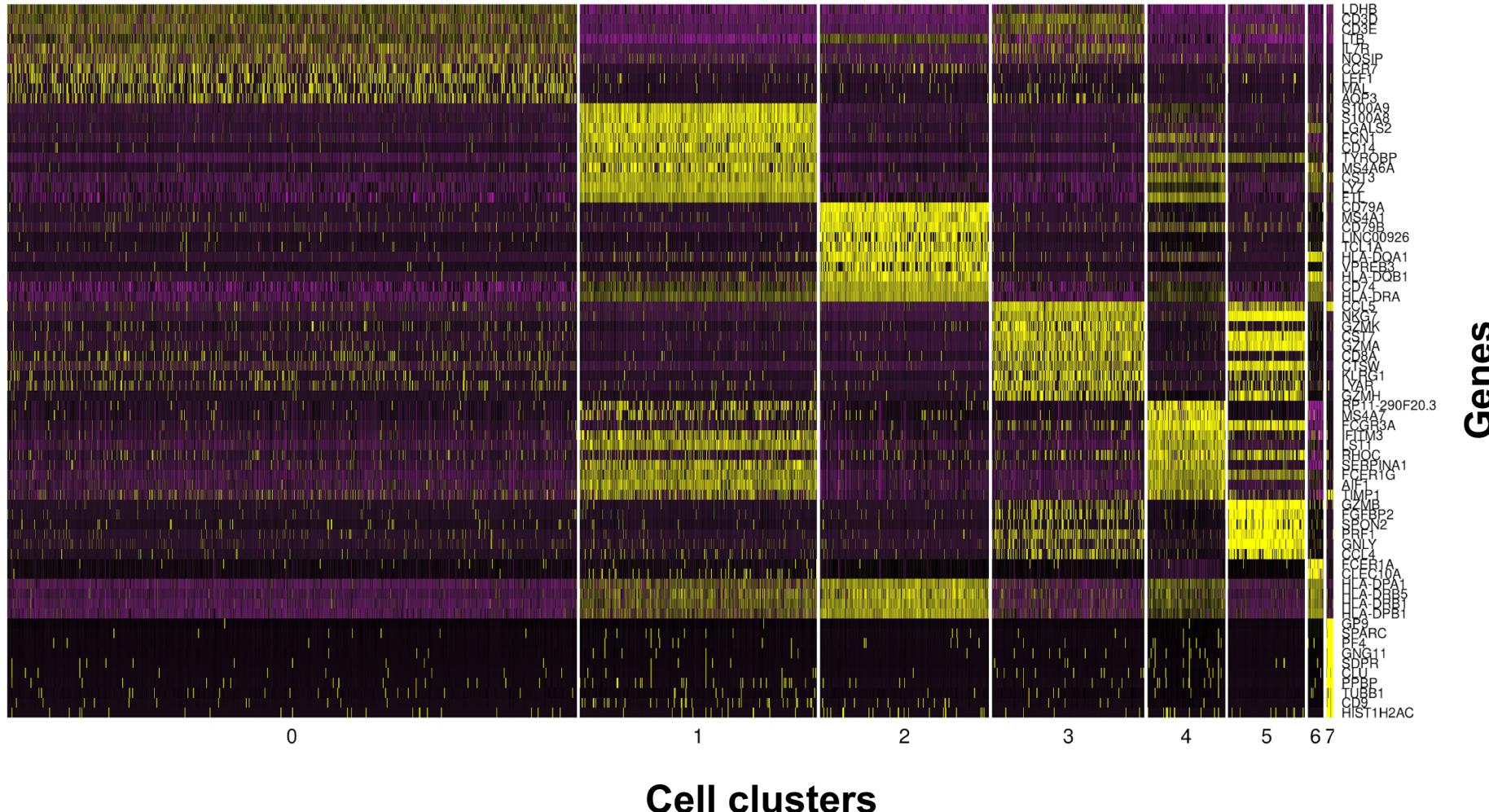
## 5. Identifying differentially expressed genes



There are many ways to test for differential expression!

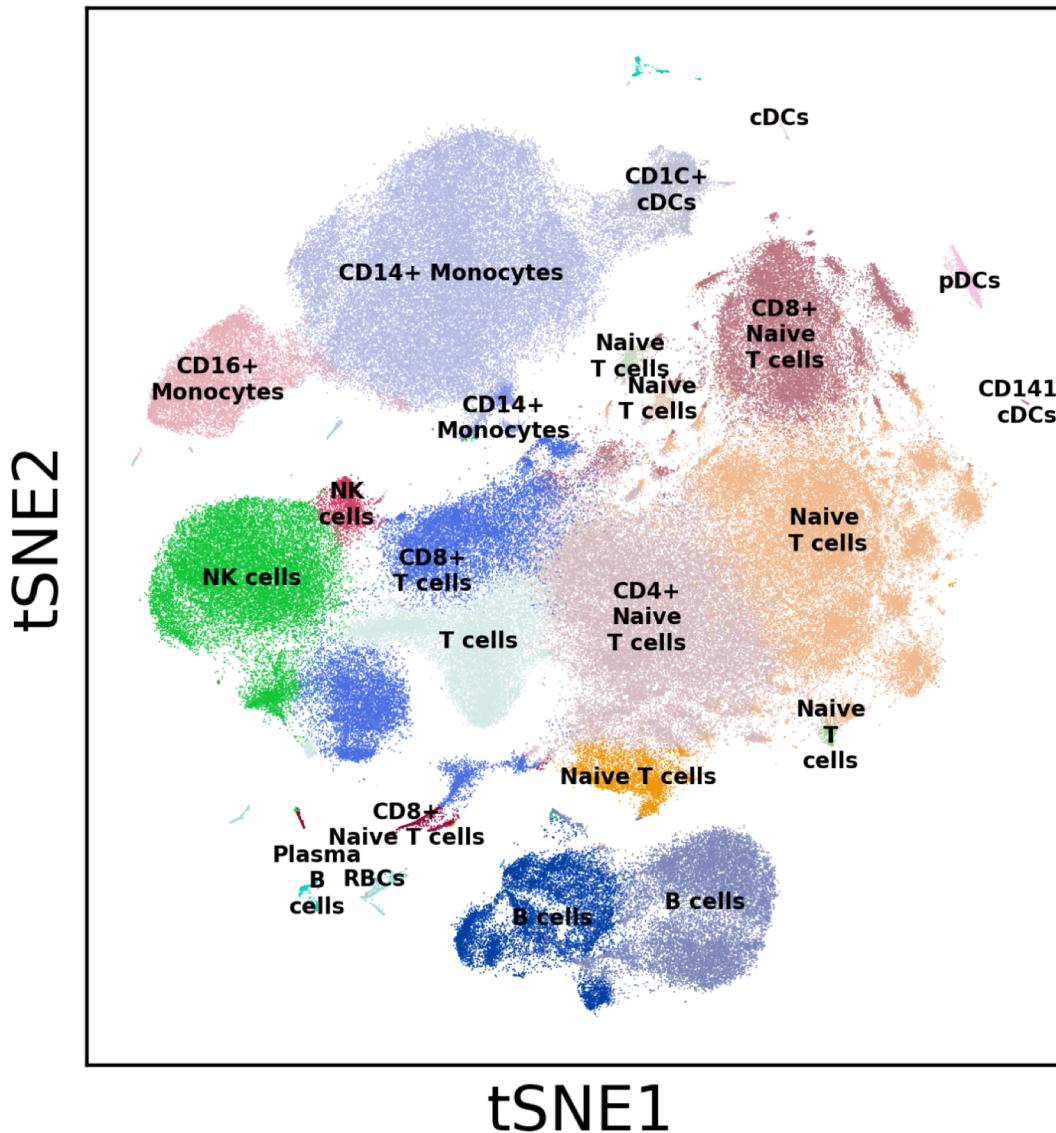
# Determining cell type, state, and/or function:

## 5. Identifying differentially expressed genes



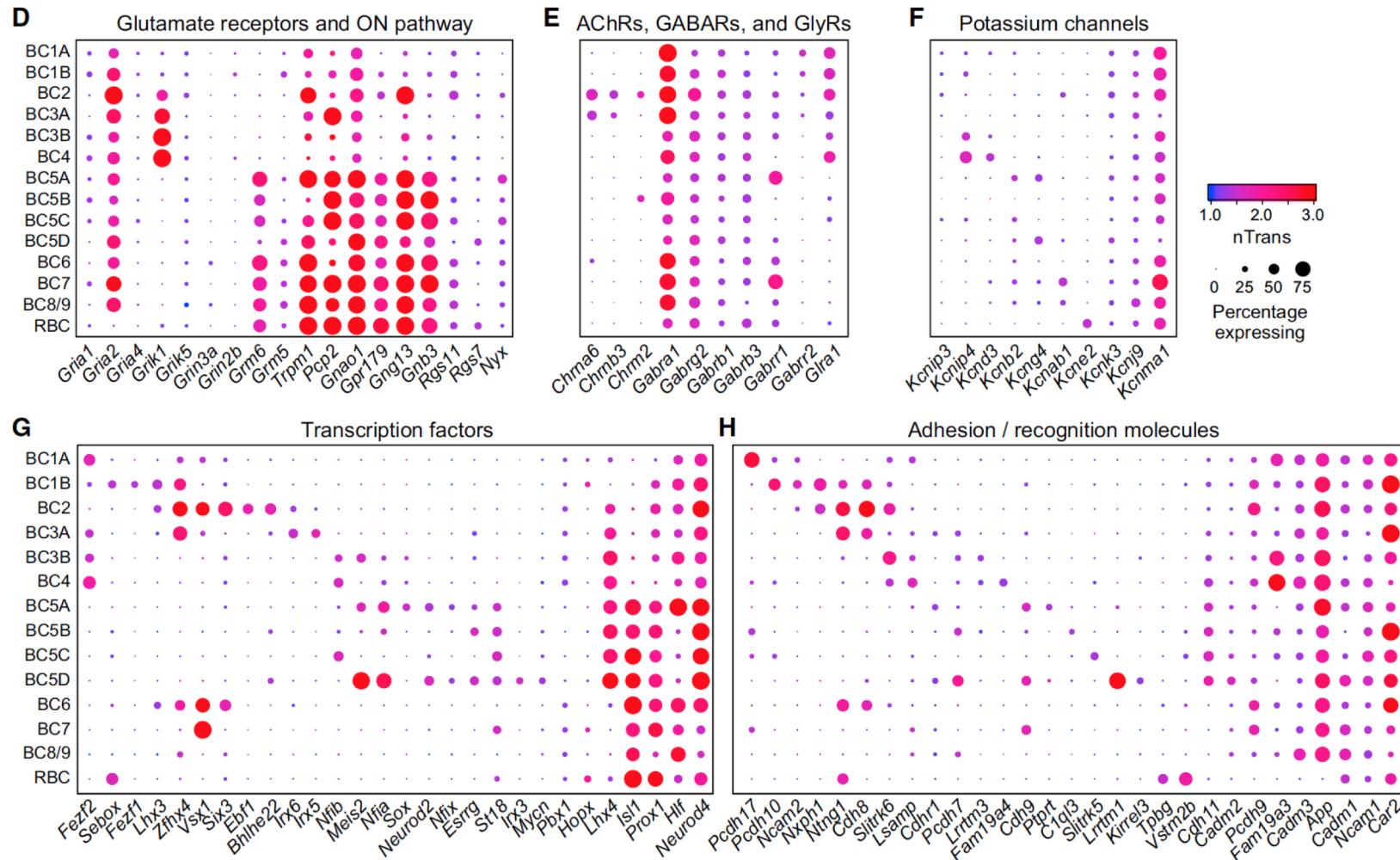
# Determining cell type, state, and/or function:

## 6. Assigning cell type



# Determining cell type, state, and/or function:

## 7. Functional annotation by pathway analysis and gene set enrichment analysis



# Visualizing genes of interest

## Dot plots, violin plots, feature plots

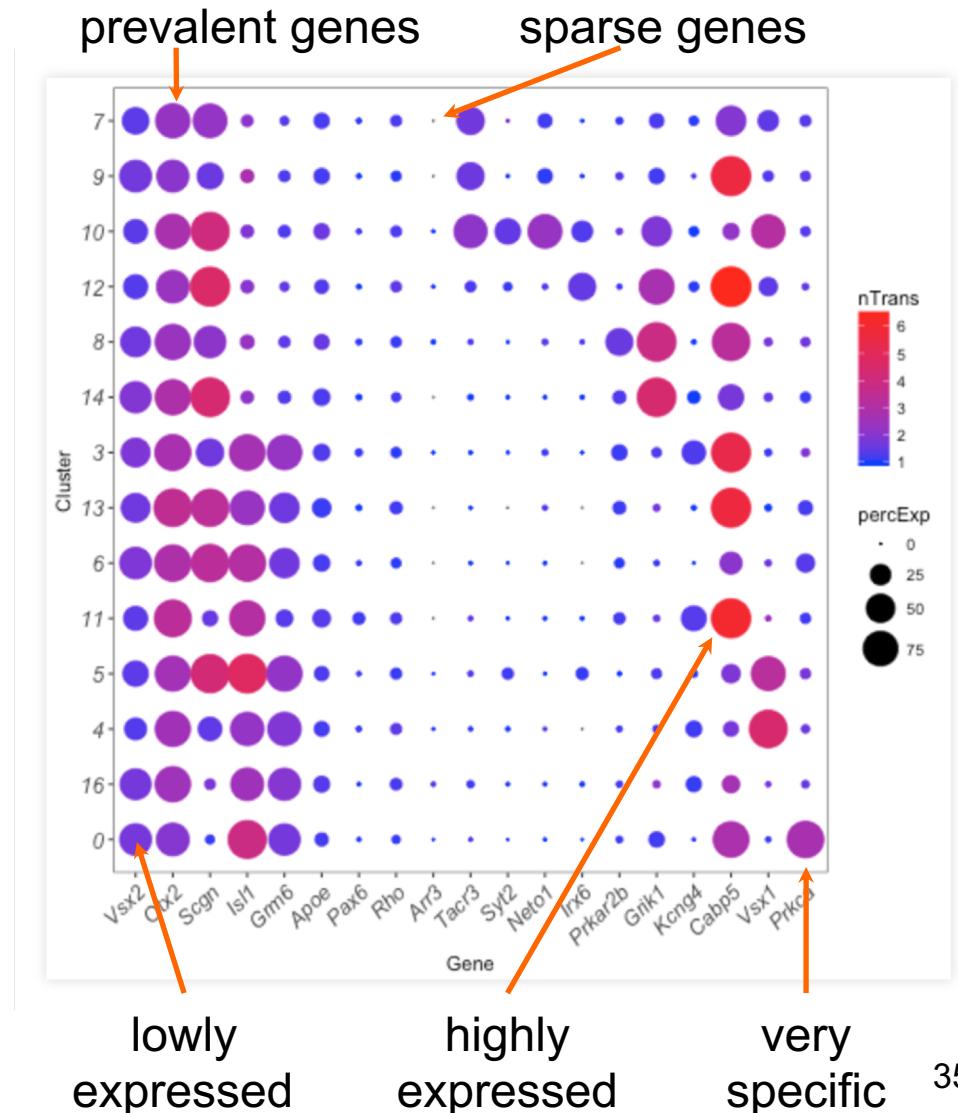
### Size of circle

- Gene prevalence in cluster

### Color of circle

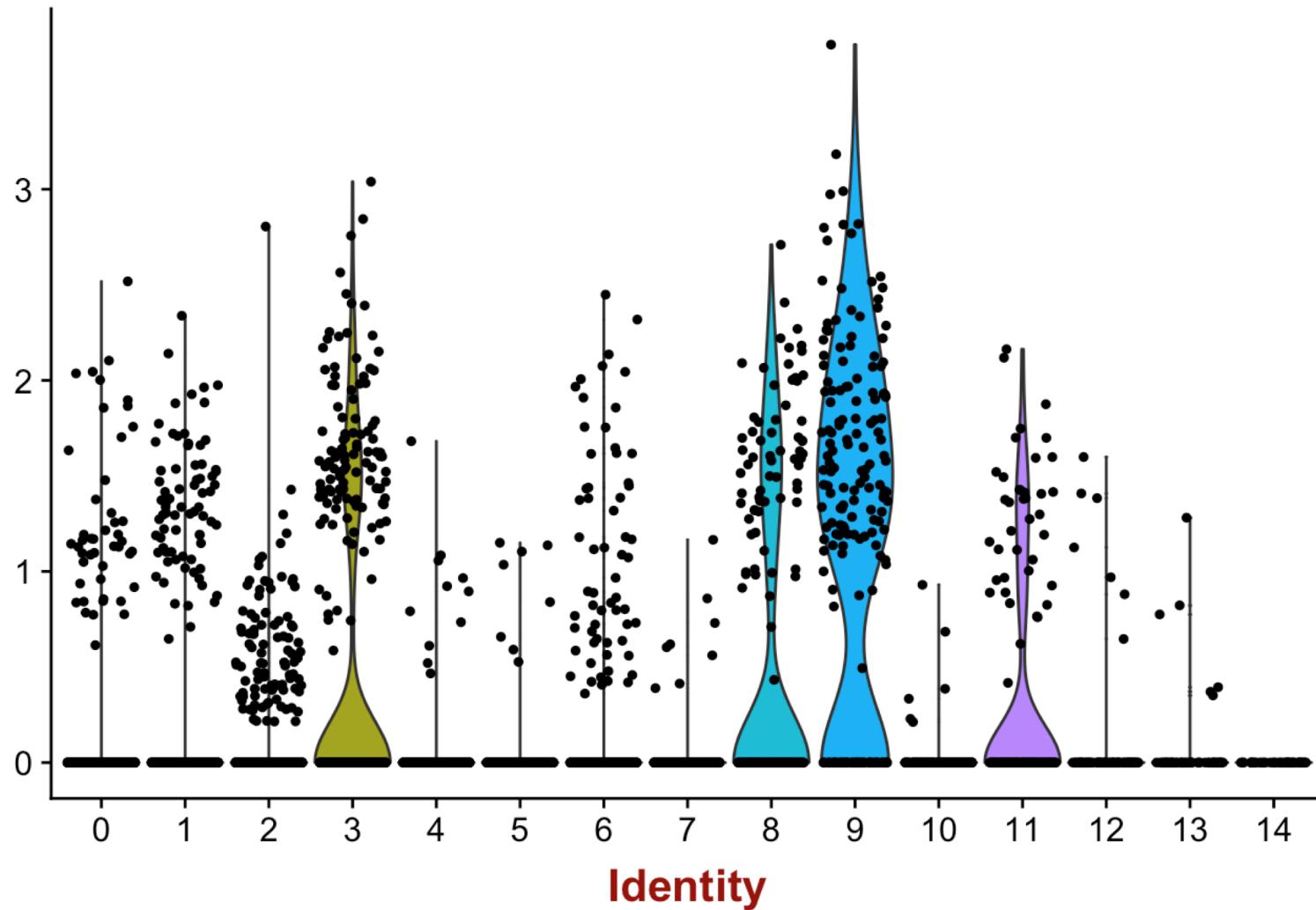
- More red, more expressed in cluster

### Scales well with many cells



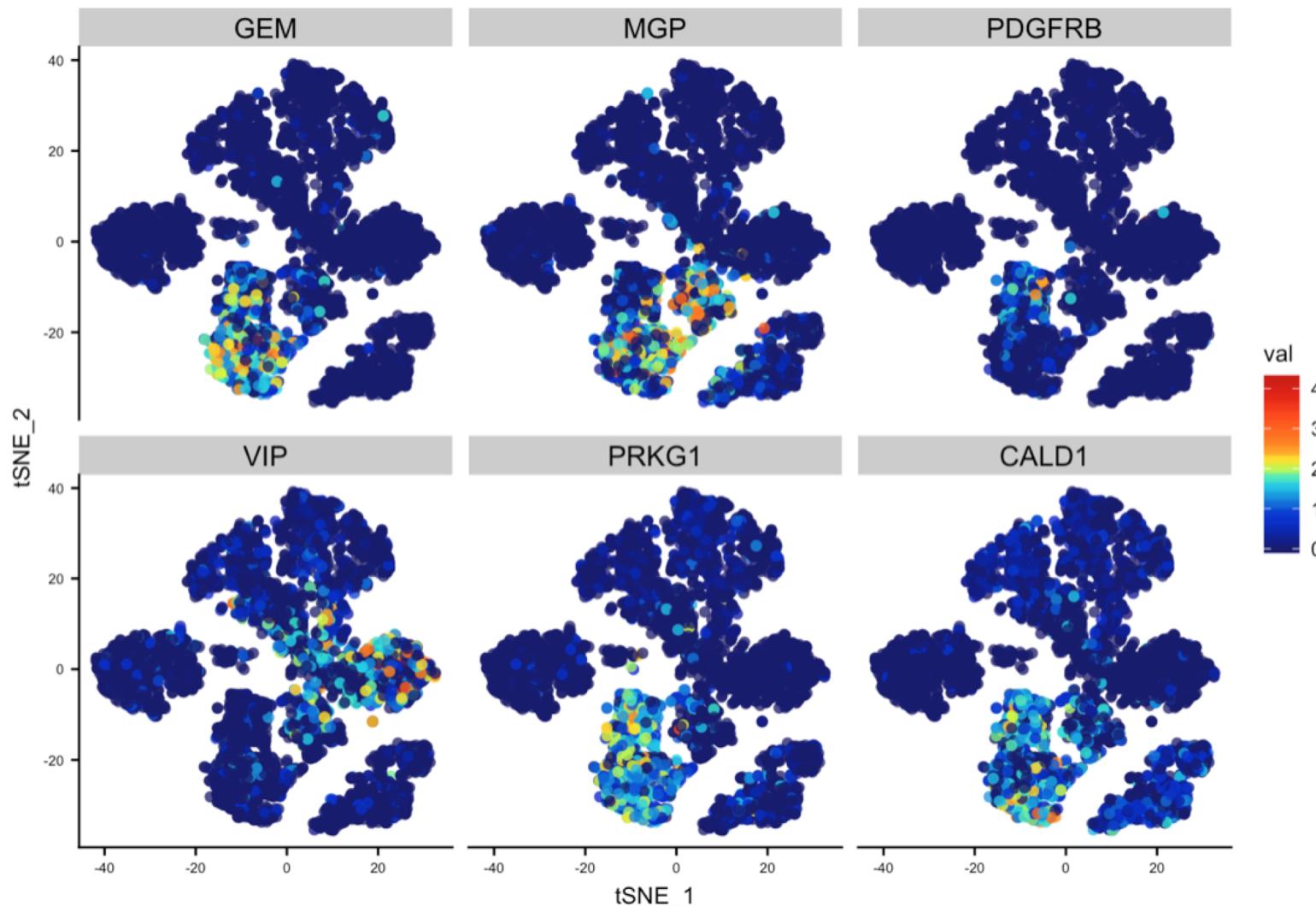
# Visualizing genes of interest

## Dot plots, **violin plots**, feature plots



# Visualizing genes of interest

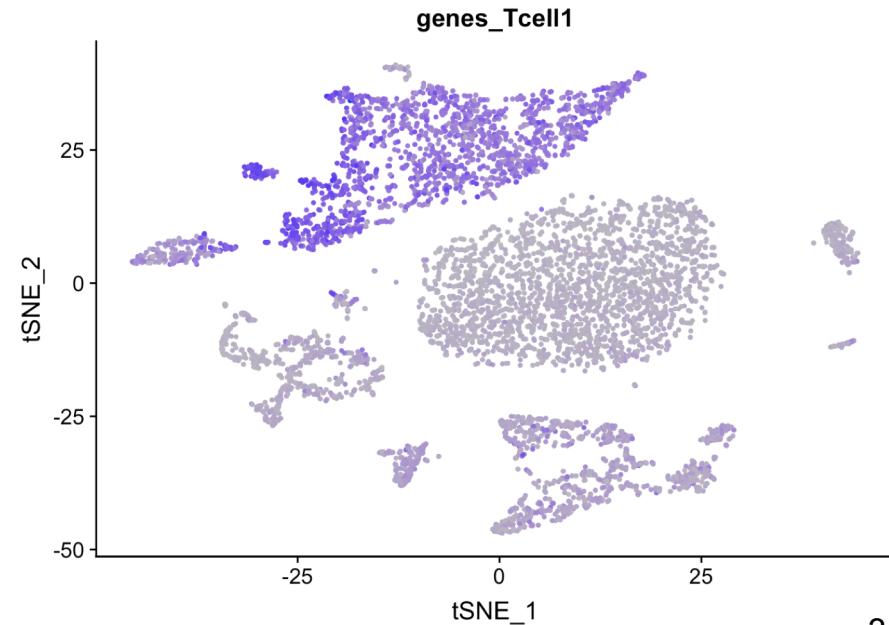
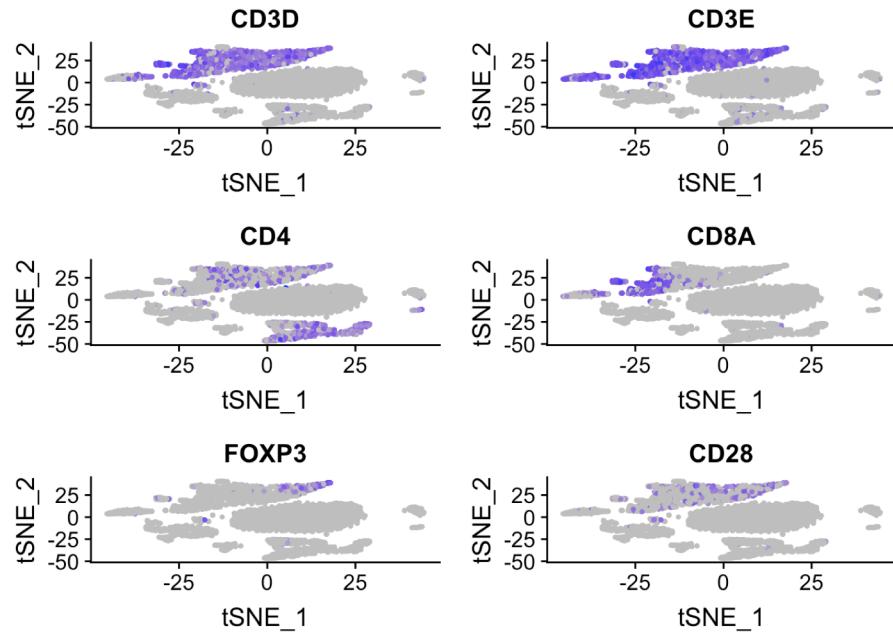
## Dot plots, violin plots, feature plots



# Gene signatures can be used to score each cell based on a set of genes

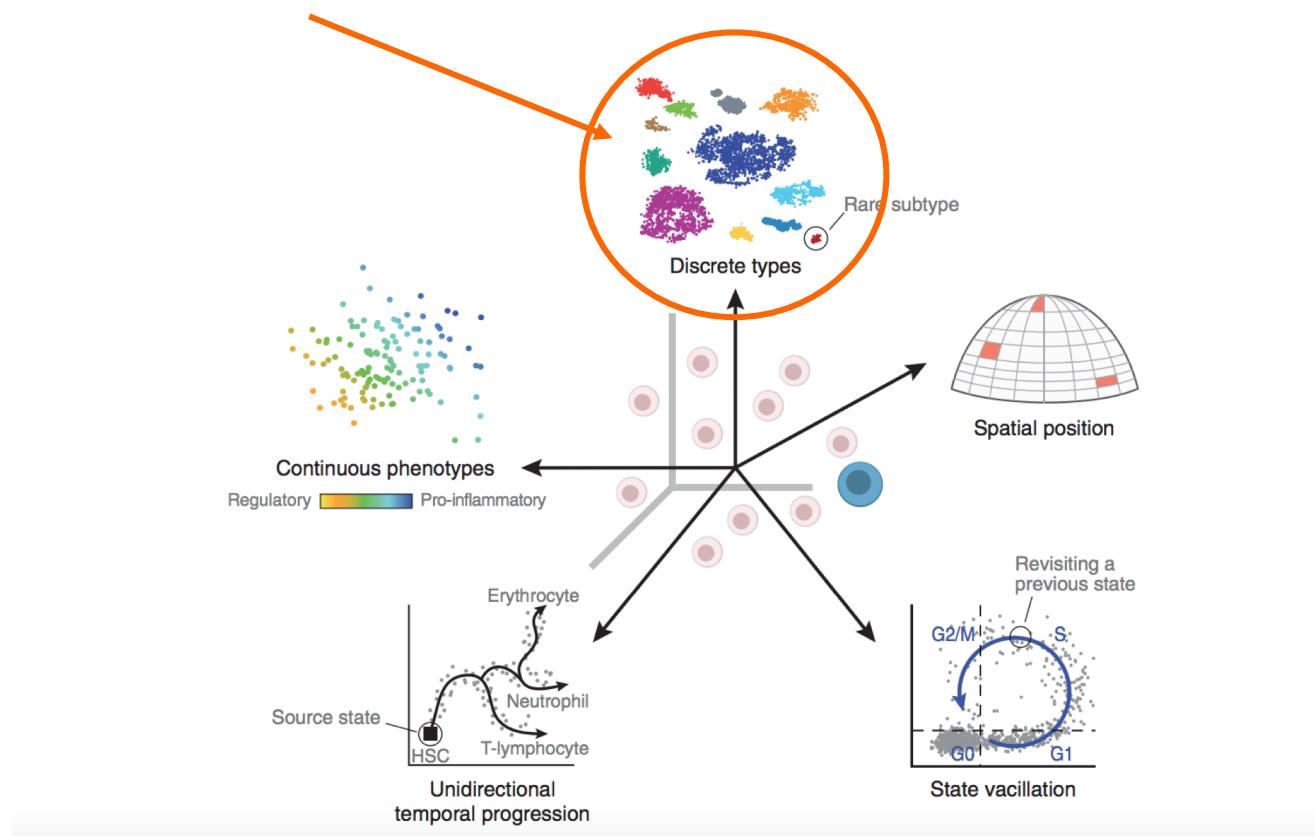
- Can visualize a score for each cell and look at multiple genes at once
- Done for a gene expression program of interest, e.g, cell-cycle, inflammation, cell type, dissociation
- Reduces the effects of dropouts

## Gene signature for T cells



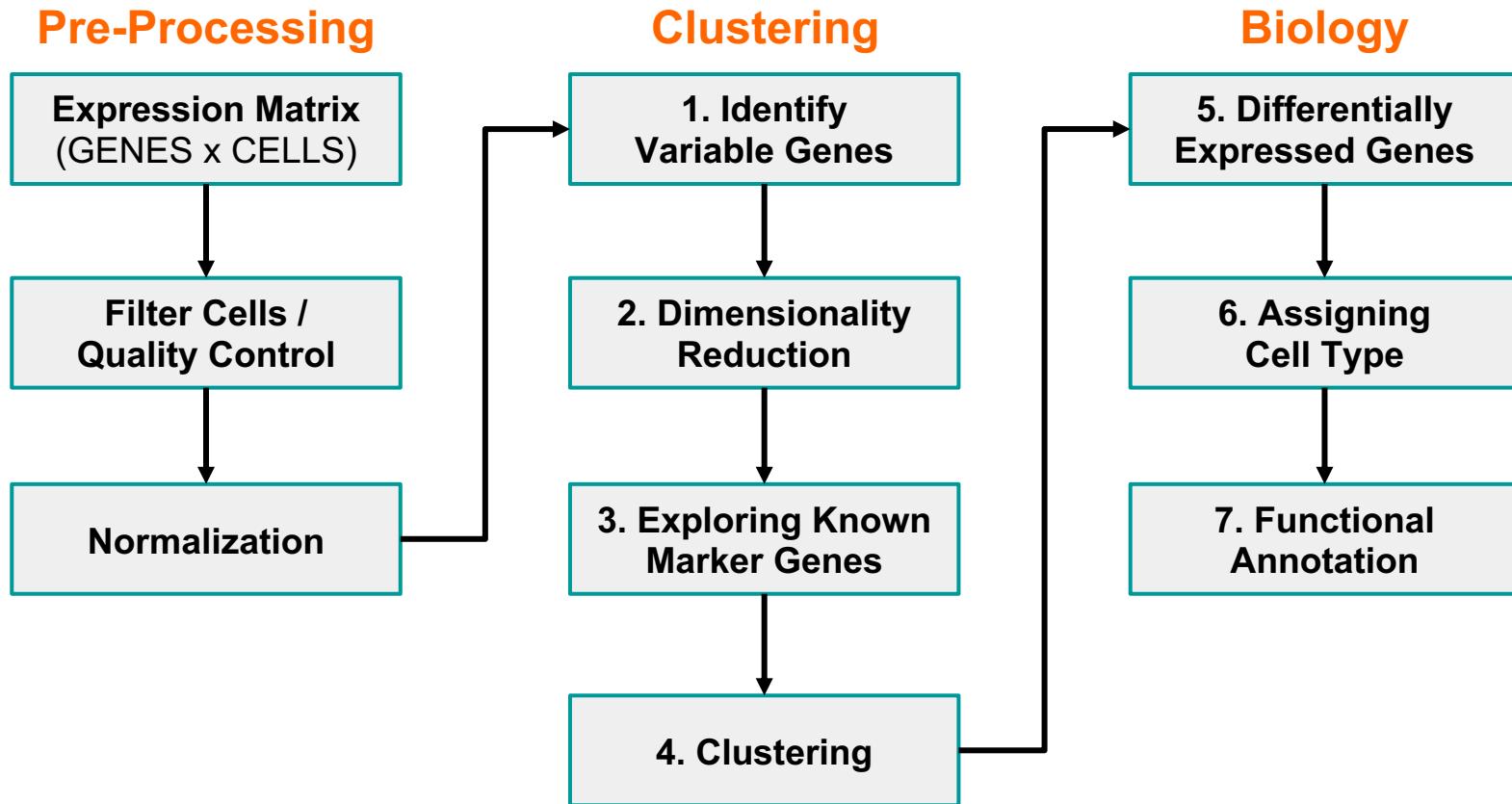
# Recap: what did we just cover?

We covered just this.  
So much more to learn!



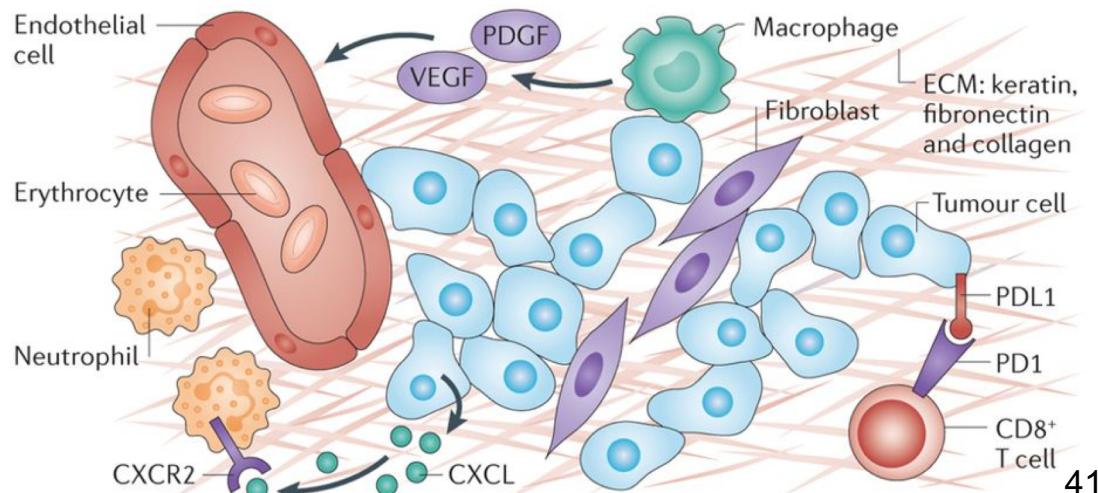
# Recap: what did we just cover?

Time to execute this pipeline in a hands on example!



# Today's dataset: Non-Small Cell Lung Cancer

- 5' gene expression data from a fresh surgical resection of a squamous non-small cell lung carcinoma tumor (publically available on 10X website).
- American Cancer Society: Squamous cell (epidermoid) carcinoma: About 25% to 30% of all lung cancers are squamous cell carcinomas. These cancers start in early versions of squamous cells, which are flat cells that line the inside of the airways in the lungs. They are often linked to a history of smoking and tend to be found in the central part of the lungs, near a main airway (bronchus).
- Tumor microenvironment consists of malignant cells, immune cells, stromal cells, vascular networks, and extracellular matrix.



41

**Tutorial is available here:**

[https://github.com/broadinstitute/CEGS-2018/blob/master/src/cegs\\_lab.Rmd](https://github.com/broadinstitute/CEGS-2018/blob/master/src/cegs_lab.Rmd)

Make sure your computer is connected to the “Broad” wireless network.

Open the Chrome Browser.

Access your gcloud Docker instance:

<http://35.196.189.242>:port

where port = 100xx, where xx is your assigned number

Username and pwd: training

# Single-cell portal: facilitates sharing and dissemination of data from single-cell studies

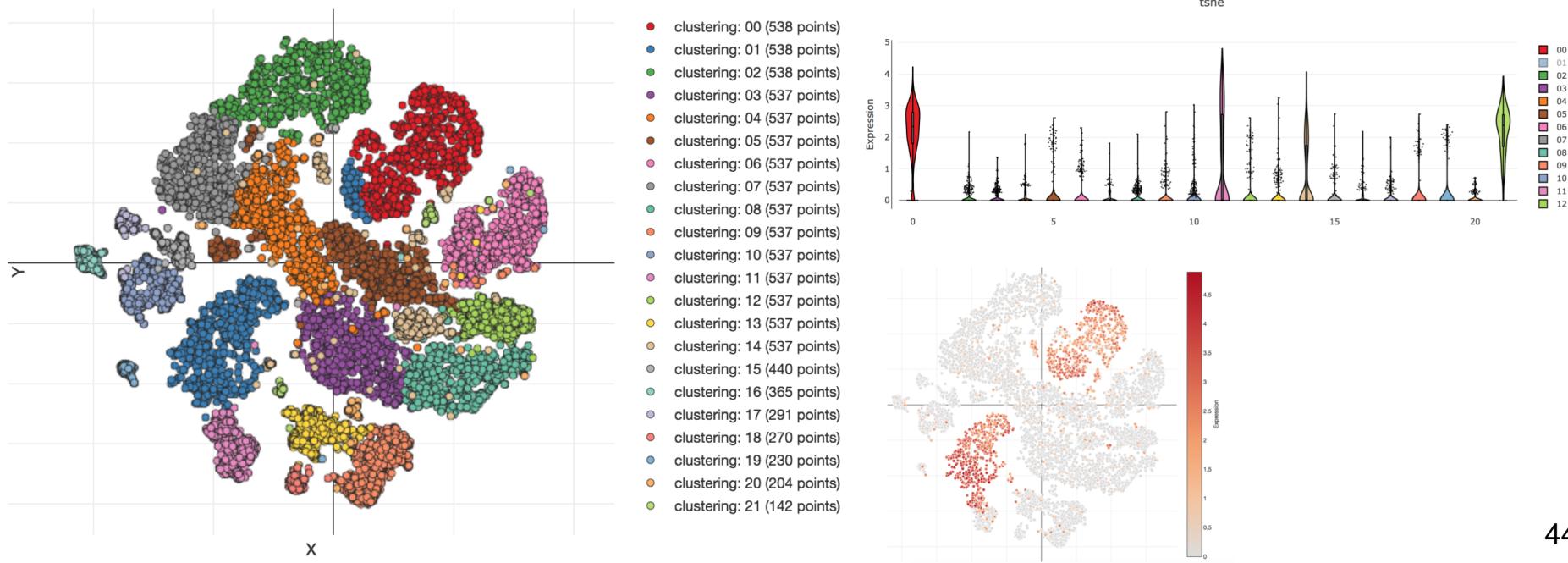
Single Cell Portal BETA

? Help ▾ Sign In

## Single Cell Portal BETA

Visualization portal for single cell RNA-seq data.

Now featuring **49** studies with **542,796** cells.



# Resources

## Learn more about tSNE

- Awesome Blog on t-SNE parameterization: <http://distill.pub/2016/misread-tsne>
- Publication: [https://lvdmaaten.github.io/publications/papers/JMLR\\_2008.pdf](https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf)
- Nice YouTube Video: <https://www.youtube.com/watch?v=RJVL80Gg3IA>
- Code: <https://lvdmaaten.github.io/tsne/>
- Interactive Tensor flow: <http://projector.tensorflow.org/>

## Computational packages for single-cell analysis

- <http://bioconductor.org/packages/devel/workflows/html/simpleSingleCell.html>
- <https://satijalab.org/seurat/>
- <https://scanpy.readthedocs.io/>

## Online courses

<https://hemberg-lab.github.io/scRNA.seq.course/>

<https://github.com/SingleCellTranscriptomics>

# Resources, cont.

Comprehensive list of single-cell resources:

[\*\*https://github.com/seandavi/awesome-single-cell\*\*](https://github.com/seandavi/awesome-single-cell)

The screenshot shows the GitHub repository page for 'seandavi / awesome-single-cell'. At the top, there are navigation links for Personal, Open source, Business, Explore, Pricing, Blog, and Support. On the right, there are buttons for 'Sign in' and 'Sign up'. Below the header, the repository name 'seandavi / awesome-single-cell' is displayed, along with a 'Watch' button (25), a 'Star' button (86), and a 'Fork' button. A navigation bar below the repository name includes 'Code' (selected), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Pulse', and 'Graphs'. The main content area contains the text: 'List of software packages for single-cell data analysis, including RNA-seq, ATAC-seq, etc.'

[\*\*www.singlecellnetwork.org\*\*](http://www.singlecellnetwork.org)

The screenshot shows the homepage of the Single Cell Network. It features a logo with the letters 'SCN' inside a blue circle. The text 'Single Cell Network' is prominently displayed, followed by the tagline 'Connecting people. Advancing science.' To the right, there is a 'Welcome, Guest' link, a 'Join' button, and a 'Log In' button. A search bar is located at the bottom right.

# Resources, cont.

## Data repositories: JingleBells

A repository of standardized single cell RNA-Seq datasets for analysis and visualization at the single cell level

Search

## Data repositories: Conquer

About conquer

The *conquer* (consistent quantification of external scRNA-seq data) repository is developed by Charlotte Soneson and Mark D Robinson at the University of Zurich, Switzerland. It is implemented in shiny and provides access to consistently processed public single-cell RNA-seq data sets. Below is a short description of the workflow used to process the raw reads in order to generate the data provided in the repository.

If you use *conquer* for your work, please cite

- C Soneson & MD Robinson: Bias, robustness and scalability in differential expression analysis of single-cell RNA-seq data. *BioRxiv* doi:10.1101/143289 (2017).

The information provided in the columns **Brief description**, **Protocol** and **Protocol type** was inferred and summarized from the information provided by the data generators in the public repositories. We refer to the original descriptions for more detailed information.

**Index building**

In order to use *Salmon* to quantify the transcript abundances in a given sample, we first need to index the corresponding reference transcriptome. For a given organism, we download the fasta files containing cDNA and ncRNA sequences from Ensembl, complement these with ERCC spike-in sequences, and build a Salmon quasi-mapping index for the entire catalog. Note that the *scater* report for a given data set (available in the *scater* report column) details the precise version of the transcriptome that was used for the quantification. For data sets with "long" reads (longer than 50 bp) we use the default `kr31`, while for "short reads" (typically around 25 bp) we set `kr15`.

We also create a lookup table relating transcript IDs to the corresponding gene IDs. This information is obtained by parsing the sequence names in the cDNA and ncRNA fasta files. From these names we also obtain the genomic coordinates for each feature.

**Sample list and run matching**

The first step is to determine the set of samples included in a given data set. We download a "RunInfo.csv" file for the data set from SRA and a Series Matrix file from GEO, in order to link samples both to individual runs and to phenotypic information. If the data set is not available from GED, we construct a phenotype data file from the information provided by the corresponding repository.

**Quality control**

For each sample in the data set, we find all the corresponding runs, and download and concatenate the corresponding FastQ files from SRA. There is also an optional step to trim adapters from the reads using *cutadapt*. Next, we run *FastQC* to generate a quality control file for each concatenated read file (one or two files per sample depending on whether it was processed with a single-end or paired-end sequencing protocol).

**Abundance quantification**

After the QC, we run *Salmon* to estimate the abundance of each transcript from the catalog described above in each sample. The Salmon output files are then compressed in an archive and can be downloaded from *conquer* (see the *salmon archive* column).

For data obtained with non full-length library preparation protocols (e.g. targeting only the 3' or 5' end of transcripts), we quantify transcript and gene abundances using the *umis* pipeline developed by Valentine Svensson. Briefly, we quasimap the reads to the transcriptome using *RapMap* and use the counting capabilities of *umis* to obtain feature counts.

**Summary report - MultiQC**

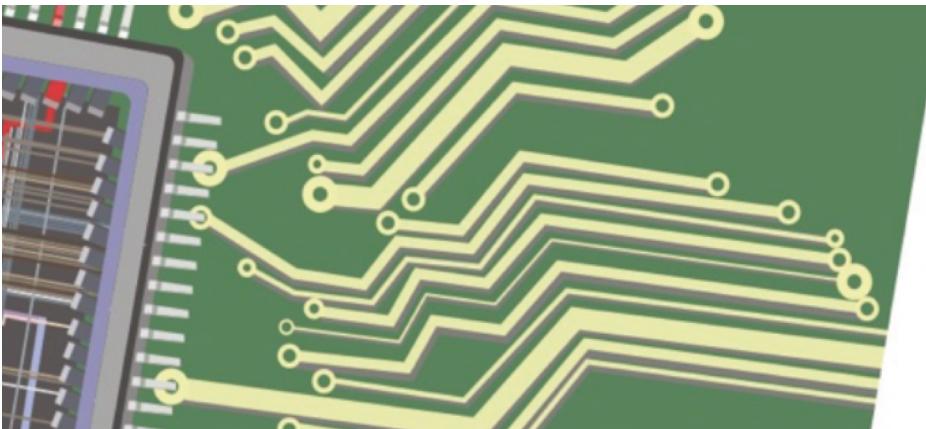
Once *FastQC* and *Salmon* (or *RapMap/umis*) have been applied to all samples in the data set, we run *MultiQC* to summarise all the information into one report. This can also be downloaded from *conquer* (see the *MultiQC report* column). This report contains quality scores for all the samples and can be used to determine if there are problematic samples and whether the data set is good enough for the purposes of the user or needs to be subsetted.

**Data summarisation**

The abundances estimated by *Salmon* are summarised and provided to the user via *conquer* in the form of a *MultiAssayExperiment* object. This object can be downloaded via the buttons in the *MultiAssayExperiment* column. To generate this object, we first use the *bimatrix* package to read the *Salmon* output into R. This returns both count estimates and TPM estimates for each transcript. Now, we summarise the transcript-level information to the gene level. The gene-level TPM is defined as the sum of the TPMs of the corresponding transcripts, and similarly for the gene-level counts. We also provide "scaled TPMs" (see [http://T200research.com/article/4\\_152/](http://T200research.com/article/4_152/) or the *scater* vignette for a discussion), that is, summarised TPMs scaled to a "count scale". In the summarisation step, we make use of the transcript-to-gene lookup table generated above.

The provided *MultiAssayExperiment* object contains two "experiments", corresponding to the gene-level and transcript-level values. The gene-level experiment contains four "assays":

- TPM
- counts
- count\_tpm (count scaled length\_weighted TPM)
- count\_tpm\_tissue (counts scaled length\_weight, which can be used to reflect in a model based on the [MultiAssayExperiment](https://CRAN.R-project.org/package=MultiAssayExperiment) package)



# Computational Genomics Workshop

## September 10 & 11, 2018

## Acknowledgements

Aviv Regev

Brian Haas

Kristine Schwenck

Elliot Boblitt

Orit Rosen

Vicky Horst

Karthik Shekhar

CEGS Workshop Team

**NIH CEGS (Center for  
Excellence in Genomic Science)**

# Questions?