

# Introduction to Single-cell Genomics

Research Informatics Solutions  
Minnesota Supercomputing Institute  
2022-04-28

# Overview & goal

1. Technical overview
2. Experimental Design and QA/QC
3. Software tools
4. Analysis Showcase

Goal:

Give an overview of single cell analysis from experimental design through analysis.

# For specific questions on library preparation and sequencing: consult UMGC!!!

Single cell library preparations and sequencing recommendations are

- Changing fairly rapidly
- Often specific to the cell type(s) being targeted and the biological questions being asked

As a result, the single cell folks at UMGC, who create the libraries and do the sequencing are the experts you want to talk for specific decisions about library preparation and sequencing.

# Single cell genomics technical overview

Christy Henzler, PhD

# What is single cell RNA-seq?

Cells are dissociated and individually sequenced

Allows the gene expression patterns of individual cells to be determined.

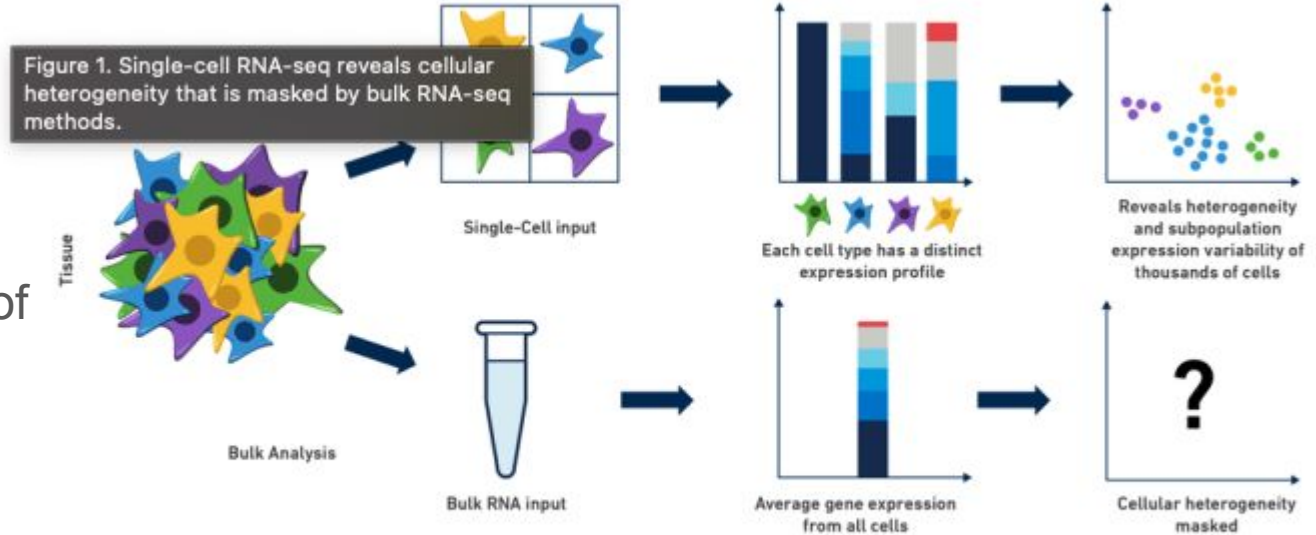


Figure 1. Single-cell RNA-seq reveals cellular heterogeneity that is masked by bulk RNA-seq methods.

In contrast, Low-input RNA-seq is a bulk RNA-seq method (one set of data produced per **sample**, not per **cell**) for small amounts of starting RNA

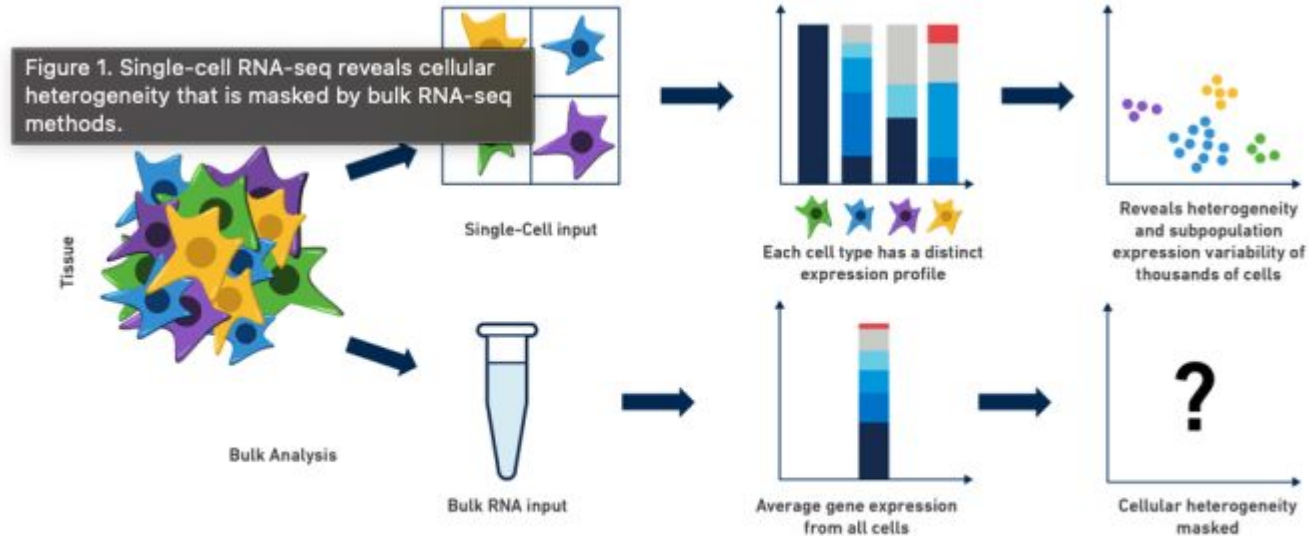
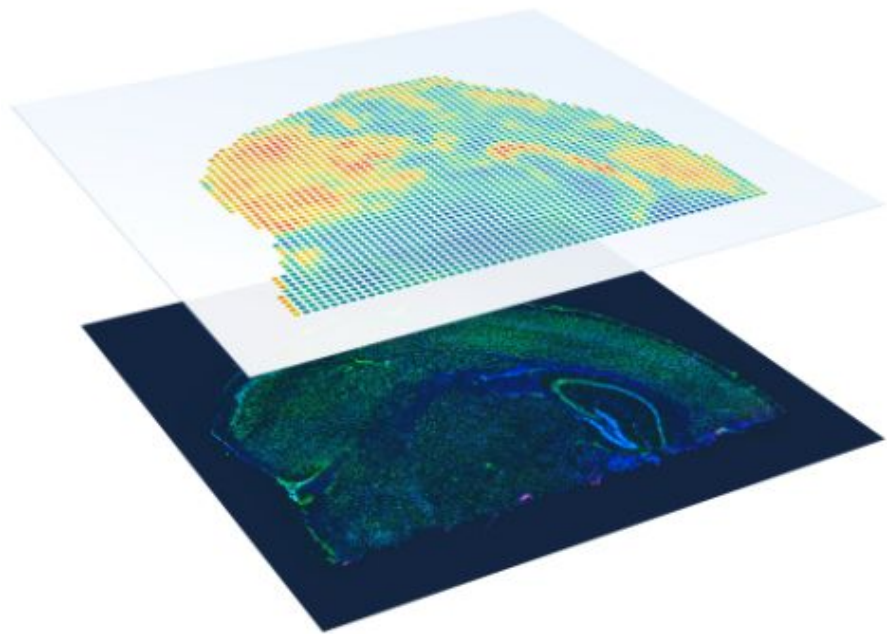
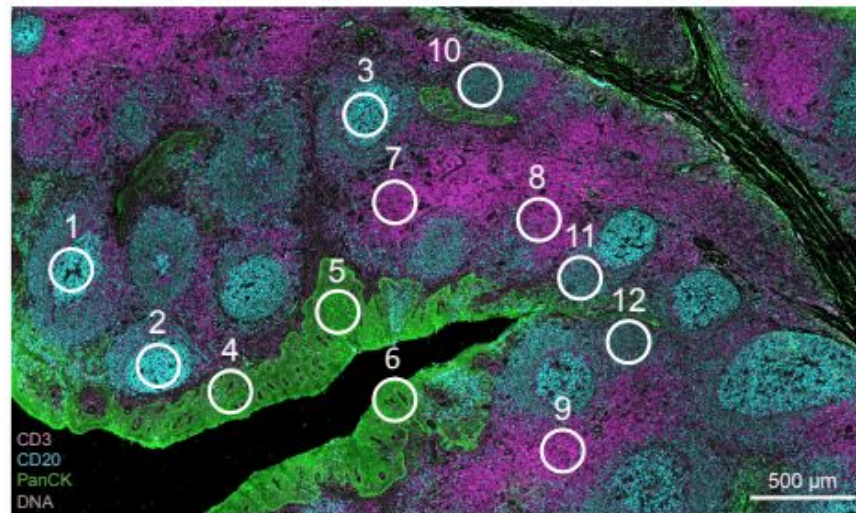


Figure 1. Single-cell RNA-seq reveals cellular heterogeneity that is masked by bulk RNA-seq methods.

Spatial transcriptomics is a method for spatially-resolved RNA-seq (can do protein, as well), though almost never at single-cell resolution



<https://www.10xgenomics.com/products/spatial-gene-expression>



[https://nanostring.com/wp-content/uploads/BR\\_MK0981\\_GeoMx\\_Brochure\\_r19\\_FINAL\\_Single\\_WEB.pdf](https://nanostring.com/wp-content/uploads/BR_MK0981_GeoMx_Brochure_r19_FINAL_Single_WEB.pdf)

# Droplet- vs plate-based scRNA-seq

	Droplet-based	Plate-based
Techniques	Drop-seq, inDrop, 10X Chromium	Smart-seq2, CEL-Seq2  Cell sorting, Fluidigm C1
How many cells?	Many (thousands of cells/sample, high-throughput)	Few (96/plate, low-throughput)
What part of gene is sequenced?	3' or 5' end	Full gene (but biased towards longer genes)
Sequencing depth	Thousands of reads	Millions of reads



# Droplet- vs plate-based scRNA-seq

	Droplet-based
Techniques	Drop-seq, inDrop, 10X Chromium
How many cells?	Many (thousands of cells/sample, high-throughput)
What part of gene is sequenced?	3' or 5' end
Sequencing depth	Millions of reads

**For MOST purposes, many cells (even at lower coverage) provide best results.**

**10X chromium is available at UMGC.**

**We will focus on droplet-based techniques today.**

# Single-cell RNA-seq vs single-nucleus RNA-seq

These methods are NOT equivalent, though the data types produced by them are the same!

	Single <b>cell</b> RNA-seq	Single <b>nucleus</b> RNA-seq
Quantity of RNA	More RNA	Less (only RNA that has been transported to the nucleus)
Types of RNA	All RNA in cell	RNA from some genes and types of genes are enriched or depleted in snRNA-seq studies
Tissue types & dissociation	Dissociation techniques can destroy fragile cells/not isolate harder to dissociate cells and/or create stress responses in cells	Can be used for frozen tissue, cells too large for the Chromium and tissue that's harder to dissociate, though there can still be biases
Enrichment/ depletion of cell types	Can enrich/deplete cells of interest using cell surface markers	Harder to enrich/deplete

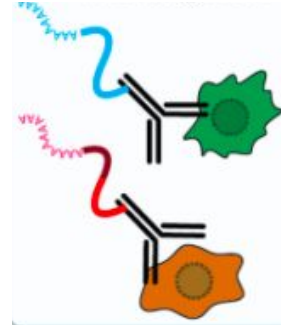
# Types of single cell 'omics

**Single cell RNA-seq: gene expression**

# Types of single cell 'omics

Single cell RNA-seq: gene expression

**Expression of cell surface proteins**

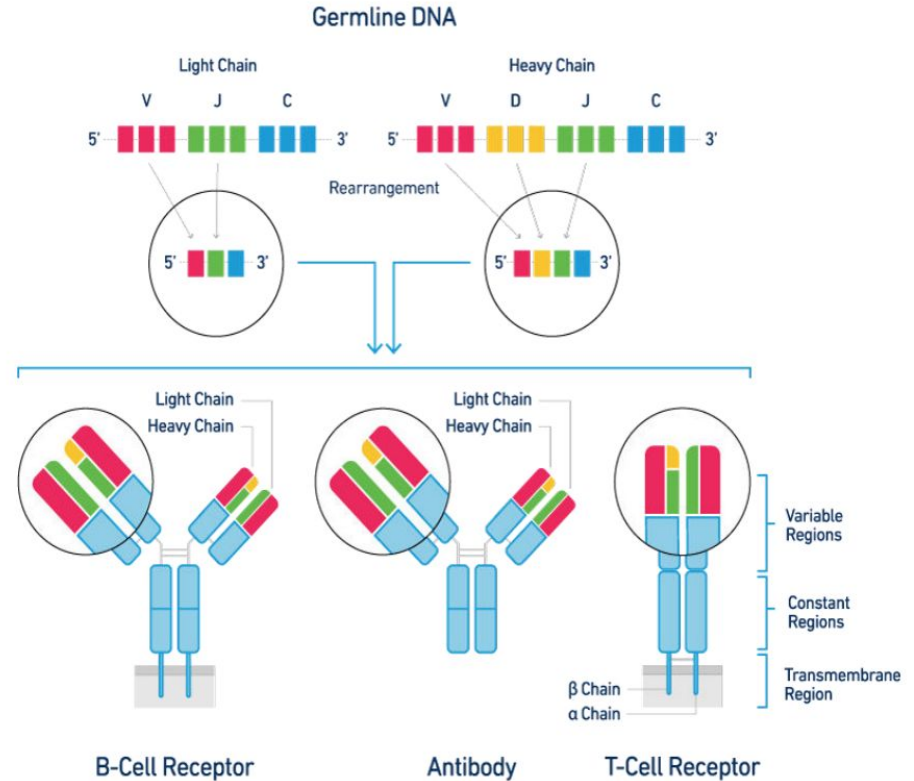


# Types of single cell 'omics

Single cell RNA-seq: gene expression

Expression of cell surface proteins

**V(D)J: immune cell profiling**



# Types of single cell 'omics

Single cell RNA-seq: gene expression

Expression of cell surface proteins

V(D)J: immune cell profiling

**ATAC (Assay for Transposase-Accessible Chromatin): chromatin accessibility**

# Types of single cell 'omics

Single cell RNA-seq: gene expression

Expression of cell surface proteins

V(D)J: immune cell profiling

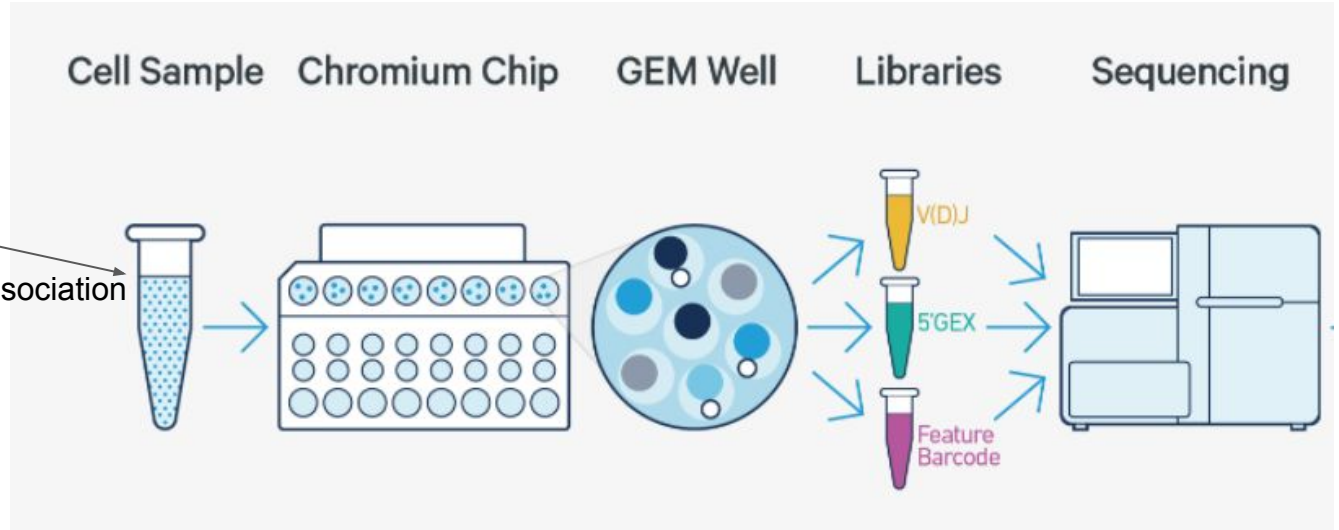
ATAC (Assay for Transposase-Accessible Chromatin): chromatin accessibility

**CRISPR guide screen: simultaneously assess CRISPR gene edits and gene or protein expression**

Tissue

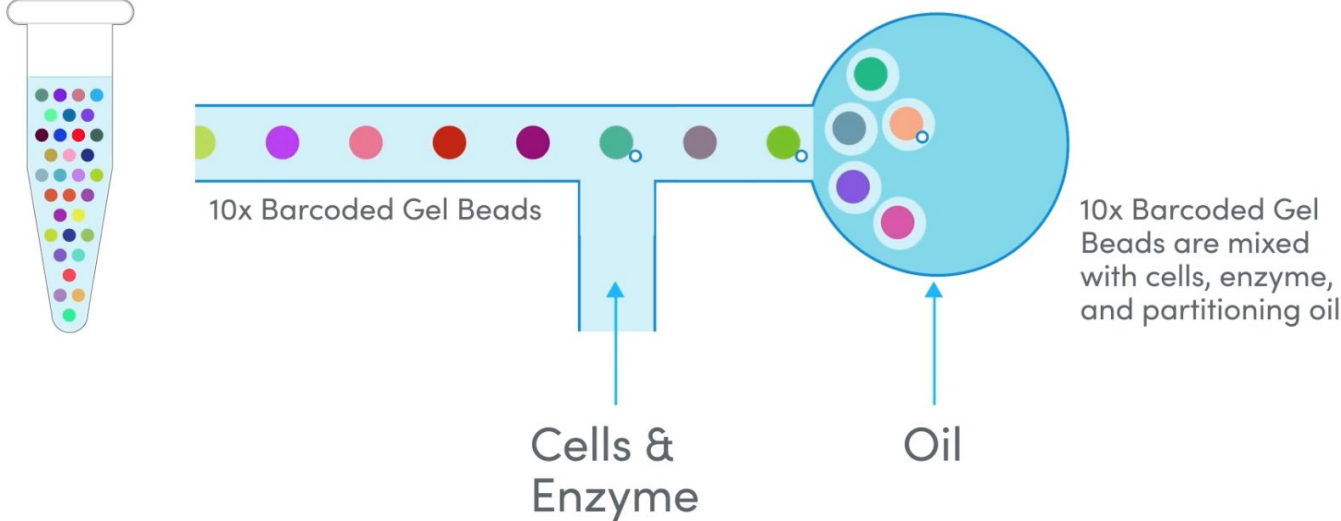


Dissociation

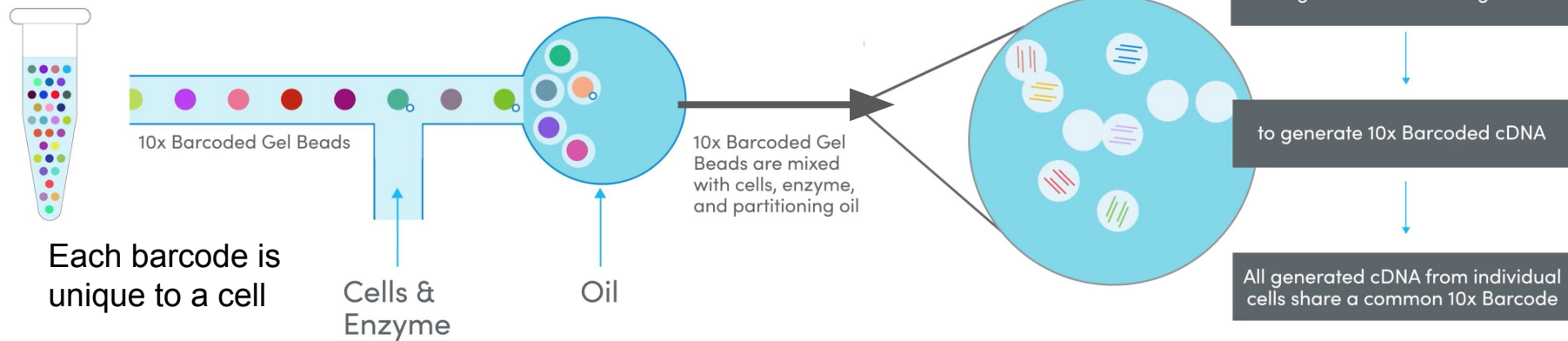




# 10x Next GEM Technology for Single Cell Partitioning

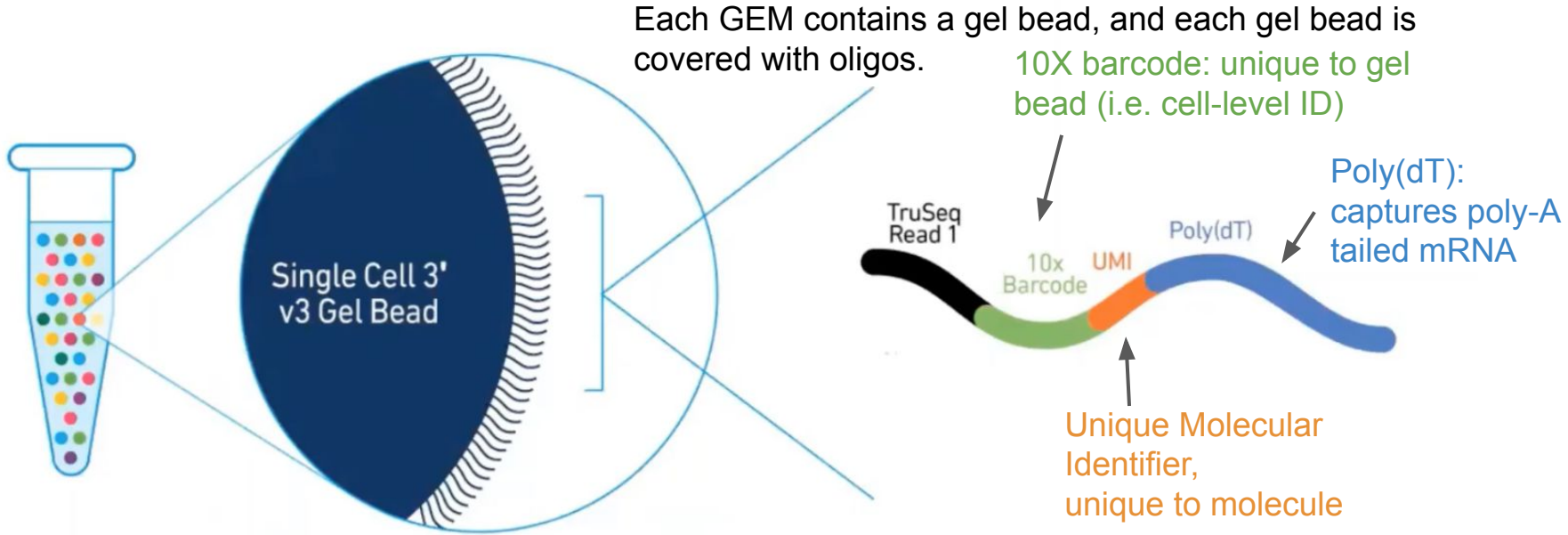


## 10x Next GEM Technology for Single Cell Partitioning



# Single Cell 3' v3 Gel Beads

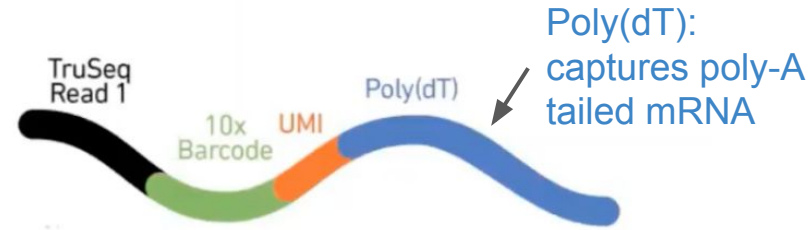
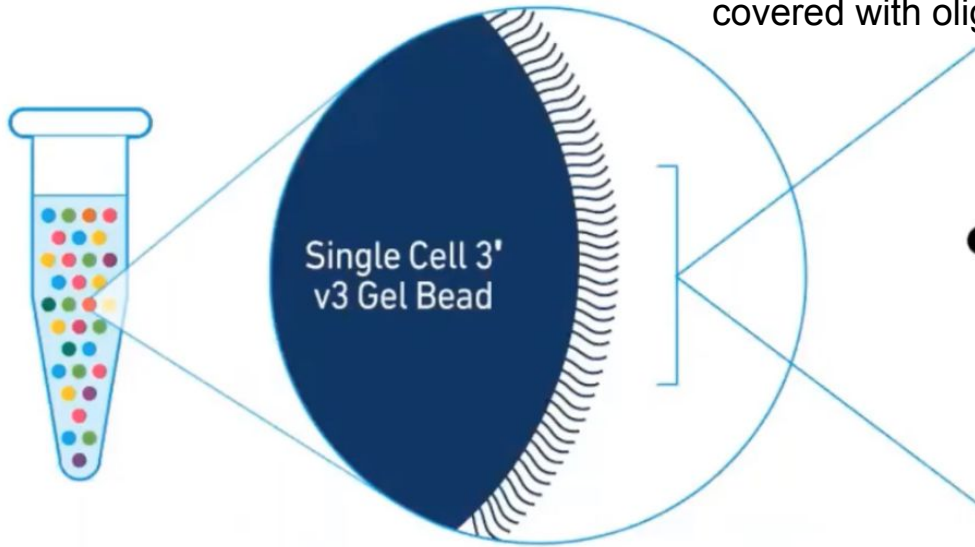
NOTE: This is an example of the gel bead capture structure for general explanatory purposes, and represents **single-indexed libraries**; UMGC has switched to **dual-indexed libraries** for gene expression, V(D)J and feature barcode assays. Check carefully with UMGC and 10x documentation for the exact structure of the oligos used for your experiments (which varies and affects processing steps); the general function is the same!



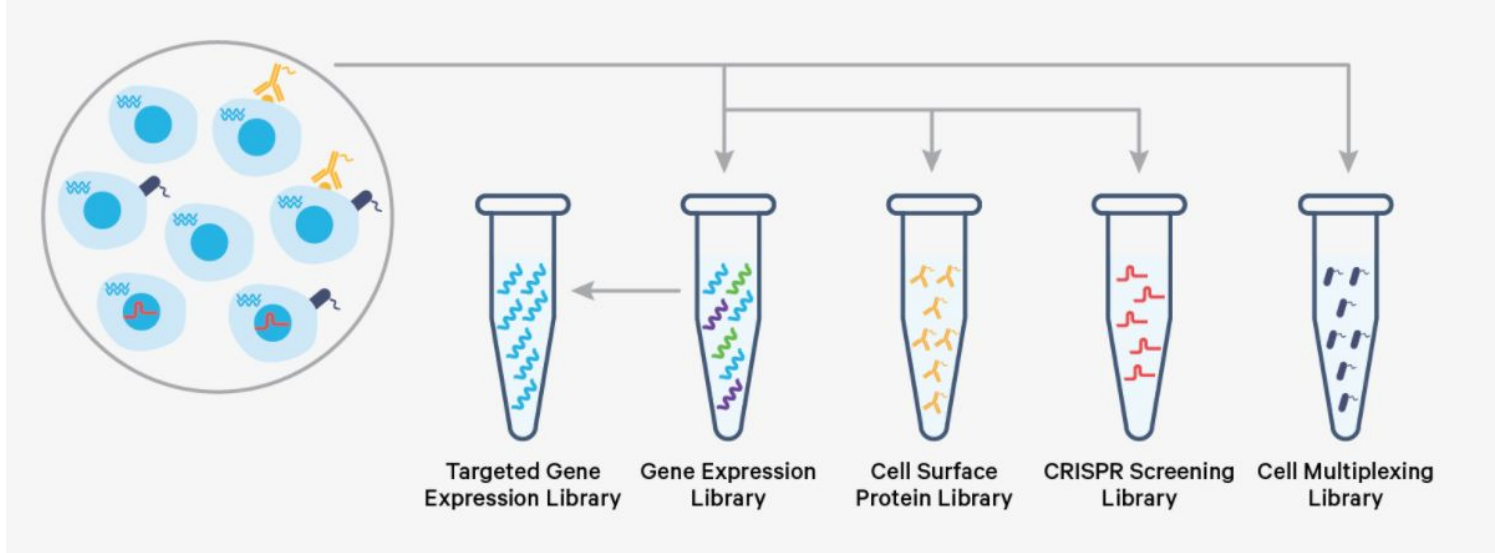
# Single Cell 3' v3 Gel Beads

NOTE: This is an example of the gel bead capture structure for general explanatory purposes, and represents **single-indexed libraries**; UMGC has switched to **dual-indexed libraries** for gene expression, V(D)J and feature barcode assays. Check carefully with UMGC and 10x documentation for the exact structure of the oligos used for your experiments (which varies and affects processing steps); the general function is the same!

Each GEM contains a gel bead, and each gel bead is covered with oligos.



The structure of this oligo changes for non-gene expression assays (i.e. V(D)J, CRISPR, ATAC), and for dual-indexed assays (see NOTE above).



# For single-indexed 3' gene expression library:

## Read 1:

10X barcode (cell-level): 16 bp

UMI (molecule-level): 12 bp

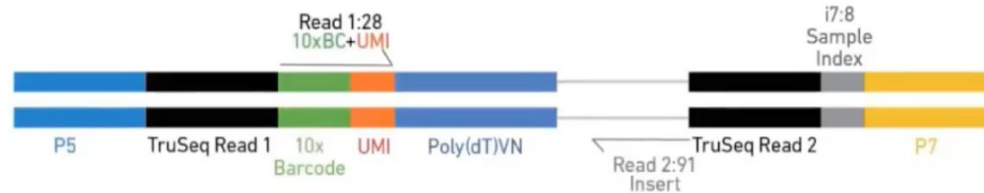
## Read 2:

Transcript: 91 bp

## Index 1:

i7 index 8 bp

**NOTE: This an example only, and UMGC has switched to dual indexing (slightly different read format) for gene expression, V(D)J and feature barcode libraries). Check carefully!!!**



20k - 50k reads per cell

	Read 1	i7 Index	i5 Index	Read 2
Purpose	10x Barcode & UMI	Sample Index	N/A	Transcript
Length	28	8	0	91*

# Limitations of the data

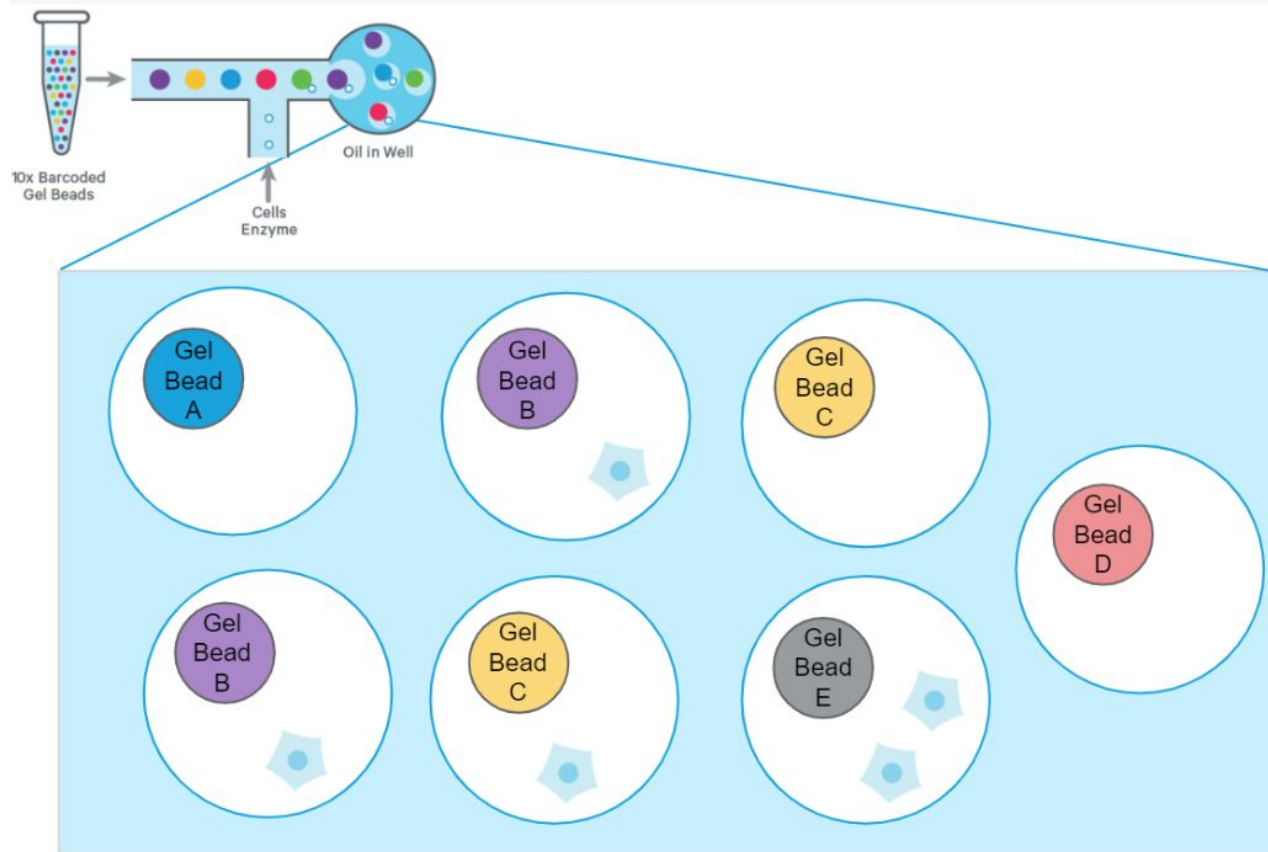
RNA handling and library prep – at any level (bulk, single cell, single nucleus) – are notorious for batch effects

Multiplets and empty droplets

Not all genes are detected:

sparse matrix (cell x gene matrix has a lot of zeros!)

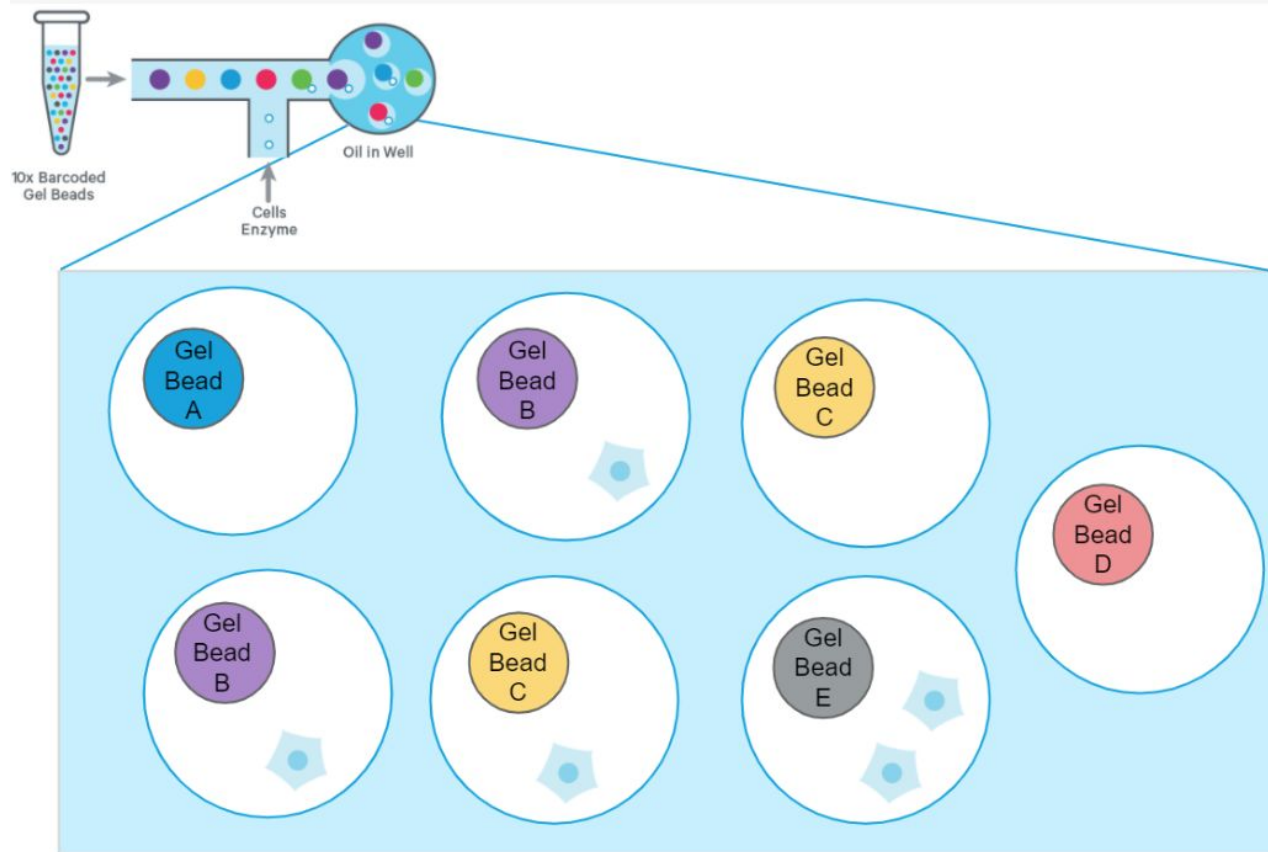
# Multiplets



<https://kb.10xgenomics.com/hc/en-us/articles/360059124751-Why-is-the-multiplet-rate-different-for-the-Next-GEM-Single-Cell-3-LT-v3-1-assay-compared-to-other-single-cell-applications->



# Multiplets



<https://kb.10xgenomics.com/hc/en-us/articles/360059124751-Why-is-the-multiplet-rate-different-for-the-Next-GEM-Single-Cell-3-LT-v3-1-assay-compared-to-other-single-cell-applications->

# Multiplexing samples

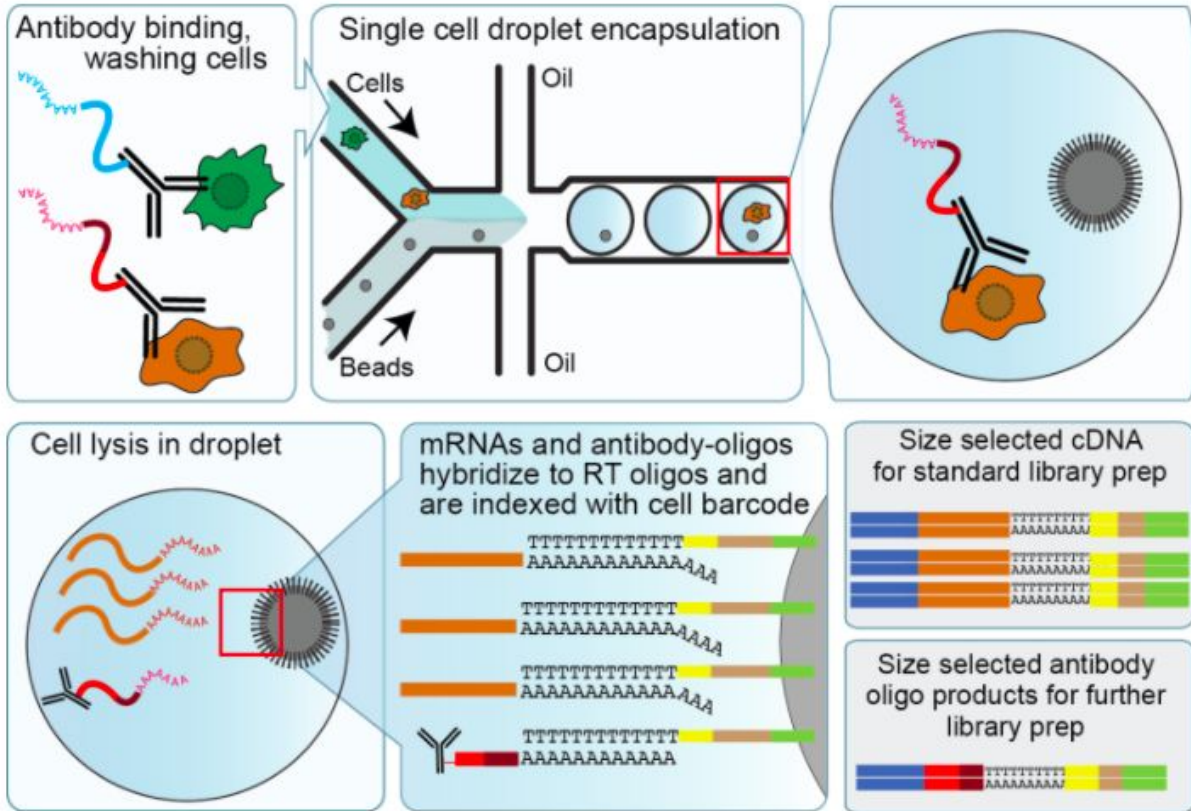
The first multiplexing technique is an oligo-tagged antibody method developed by the NY Genome Center and Satija lab as a method to target cell surface proteins of interest, and can be used for:

- Sample multiplexing (cell hashing): using ubiquitous cell surface proteins

- Quantifying protein expression (CITE-seq): using antibodies targeting cell-type specific cell surface proteins

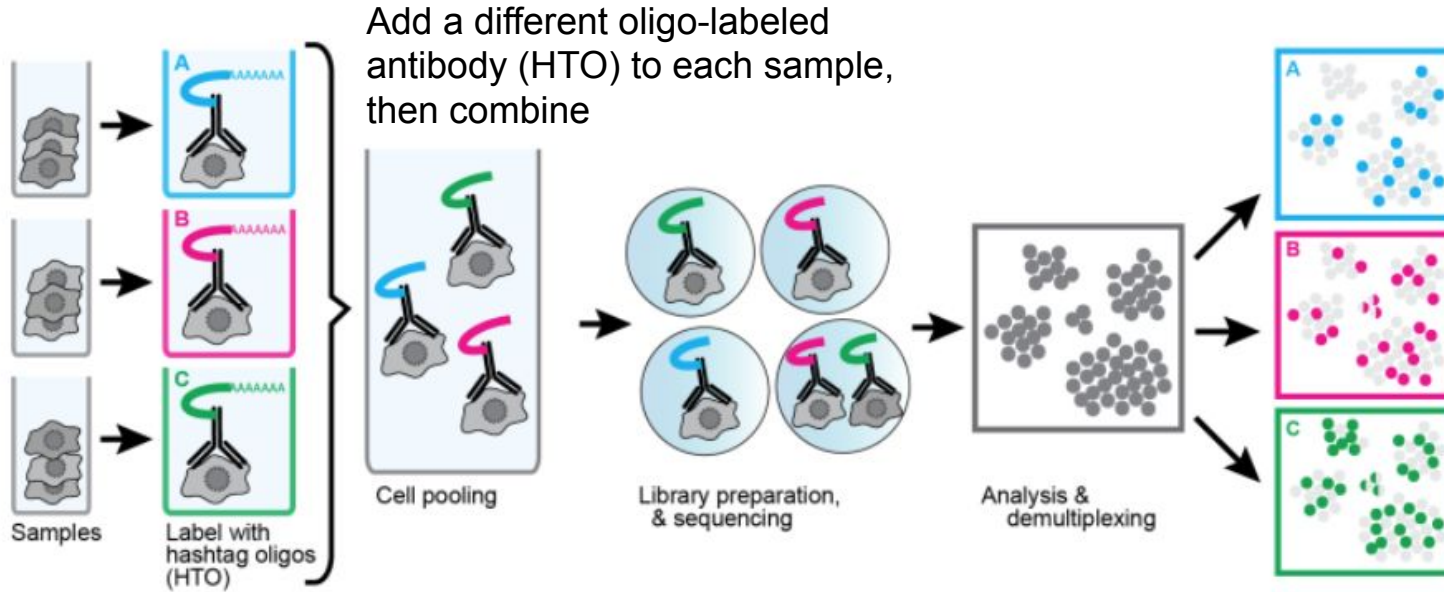
10X has developed a lipid-based method, 3' Cellplex

# CITE-seq: protein quantification



These libraries are called ADTs, for antibody-derived tags

# Multiplexing samples: cell hashing



Cellplex uses oligo-labeled lipids (CMOs), but the procedure is the same

<https://cite-seq.com/cell-hashing/>

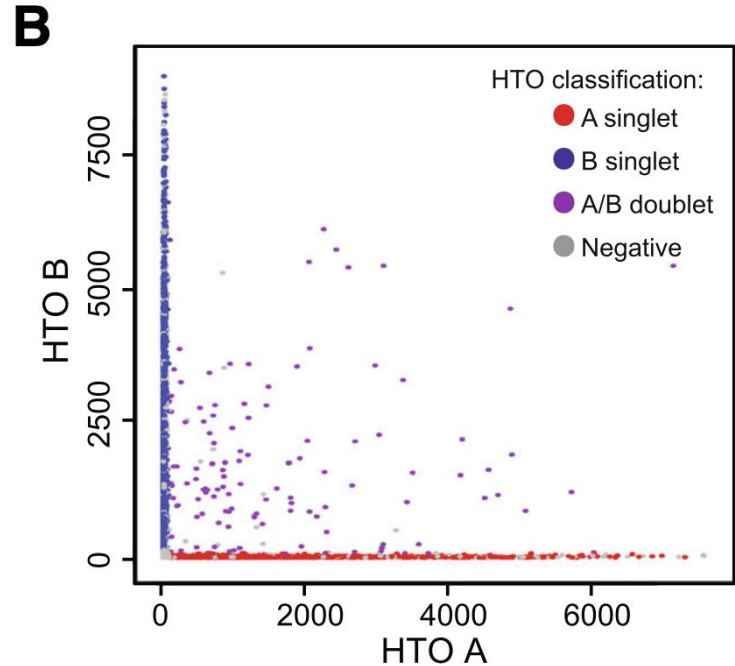
# Multiplet-detection with multiplexed samples

Cell loading into GEMs is a Poisson process

The more samples that are multiplexed, the more likely an GEMS with two or more cells contain cells from *different* samples.

Sample-specific labels make detecting these multiplets easy

This allows more cells to be loaded.



from the original cell hashing paper.

<https://genomebiology.biomedcentral.com/track/pdf/10.1186/s13059-018-1603-1.pdf>

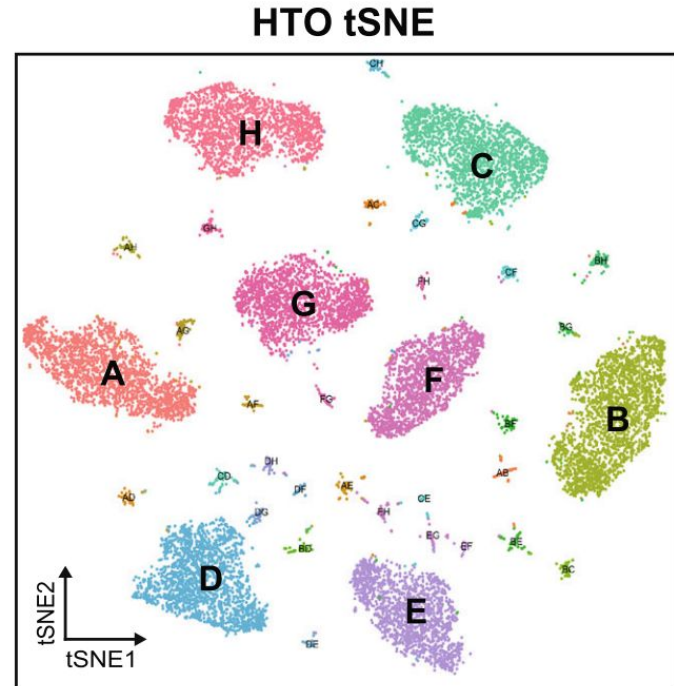
# Multiplet-detection with multiplexed samples

Sample HTOs or CMOs are produced in a separately-sequenced library, so are analyzed separately.

Clustering identifies large clusters corresponding to the single cells from each sample; much smaller clusters represent multiplets containing cells from >1 sample.

Still can't easily detect multiplets from the same sample (beyond higher expression), but the number of these should be low.

**D**



From the original cell hashing paper:  
<https://genomebiology.biomedcentral.com/track/pdf/10.1186/s13059-018-1603-1.pdf>

# Library types

Gene expression (3' or 5')

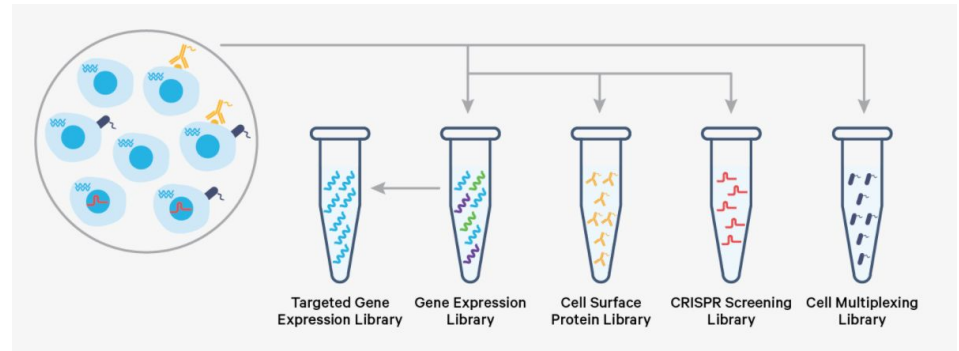
HTO/CMO (hash-tag oligos or cell multiplex oligos): Sample multiplexing

ADT (antibody-derived tags): Protein expression

V(D)J: immune cell profiling

ATAC (Assay for Transposase-Accessible Chromatin): chromatin accessibility

CRISPR guide screen: simultaneously assess CRISPR gene edits and gene or protein expression



# Single cell experiments require customized approaches

Single-cell RNA-seq analysis is still largely customized to the individual project

Choices for dissociation techniques, library preparation and sequencing are all influenced by the types of cells and biological questions – please talk to UMGC about the details!

Experimental design and data analysis also require careful thought, as you'll hear about in the next sections.

There is a growing list of user-friendly software for analysis, but you need to make thoughtful decisions at every step of this process that are specific to your analysis and the biological question you are trying to answer.



# Single-cell Experimental Design and QA/QC

Tom Kono, PhD

# Experimental Design is Still Important!

Single cell genomics is relatively new and exciting! It is also (for better or for worse) a “hot” technique in biomedical research.

A good experimental design will save you a lot of headaches:

- Less time to complete analysis
- More robust analysis
- *Less wasted money and effort!*

A good design can be *reanalyzed* if the initial analysis is done poorly. A poorly designed experiment is doomed from the start.

# General Experimental Design Principles

*Hypothesis*: a possible (and **testable**) explanation for a phenomenon. This is the goal of the experiment - test the hypothesis!

Even in the “novel discovery” phase, there should be some expectation or predicted outcome that can be measured or observed.

Examples:

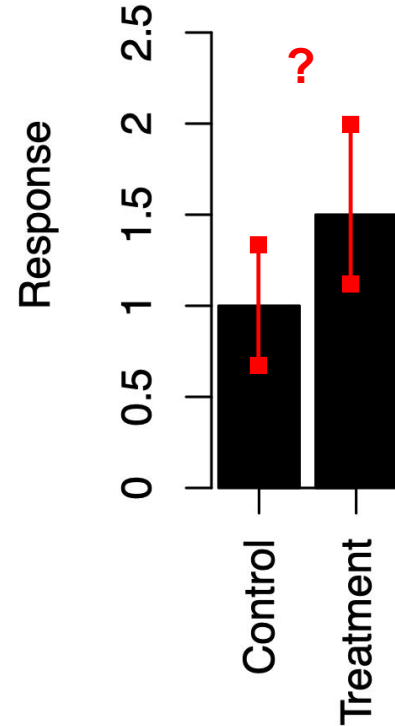
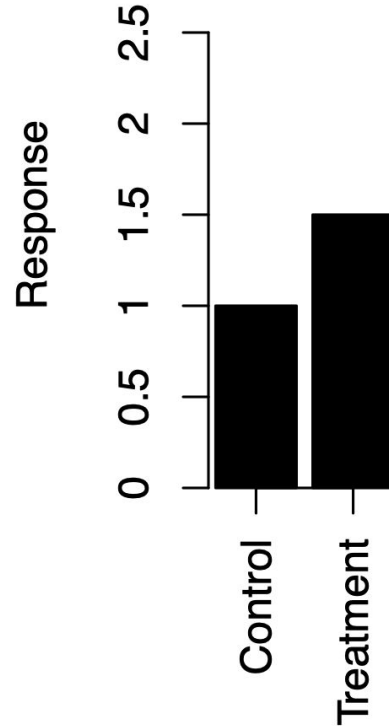
*Genes that are upregulated under stress relative to benign conditions in lung tissues are enriched for oncogenic functions.*

*Progression of a disease is associated with lower chromatin accessibility in pancreas cells.*

# General Experimental Design Principles

*Replication:* Multiple measures (animals or samples) of the same experimental condition. Allows estimation of **variability** and thus **statistical significance**.

*Randomization:* Assignment of samples to experimental groups independently (via random number generator, e.g.). Ensures that differences are due to treatment and not “accident.”

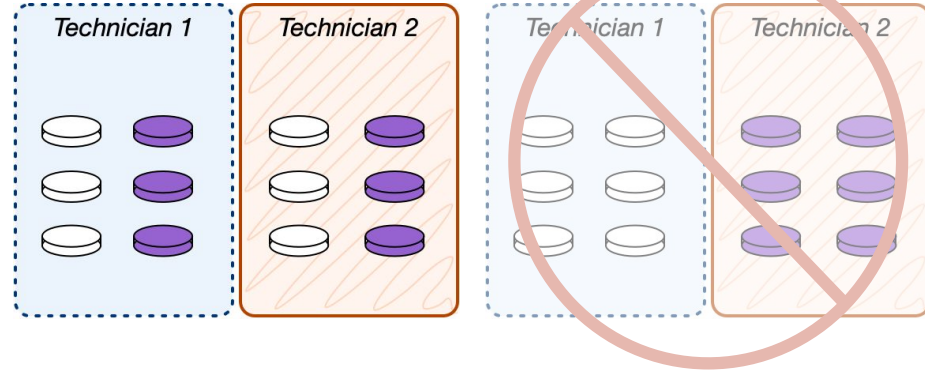


# General Experimental Design Principles

*Blocking:* Organizing samples into similar groups to reduce error.

**Randomize** or **balance** samples across groups to avoid **confounding**.

*Control:* A condition in which the experimental treatment is not applied, and used as a comparison to estimate the effect of the treatment.



**Saline**  
Control



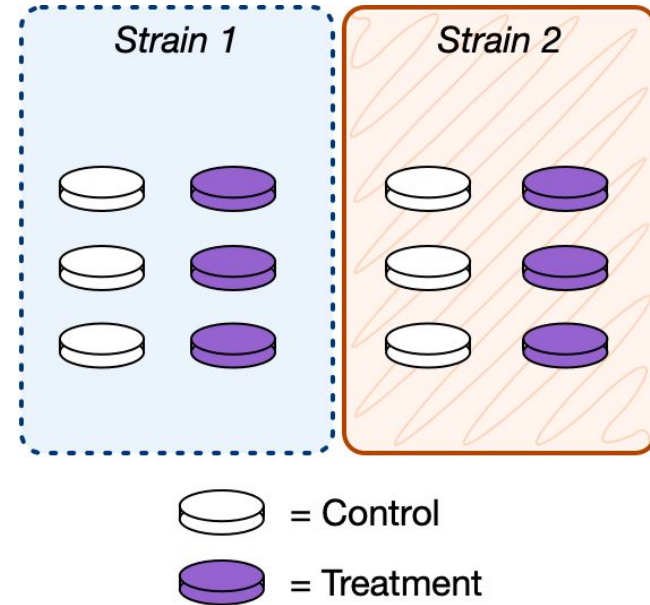
**Drug**  
Treatment

# “Block what you can control, randomize what you cannot.”

Use blocks to make groups of samples where a “nuisance variable” is constant and the experimental variable is not.  
Examples:

- Both female and male subjects in the experimental groups when sex differences are thought to contribute to observed response (“sex” is the blocking factor).
- Subjects of different strains/genotypes in the experimental groups (“strain” or “genotype” is the blocking factor)

This is possible for designed experiments! It is more difficult for clinical or patient samples.



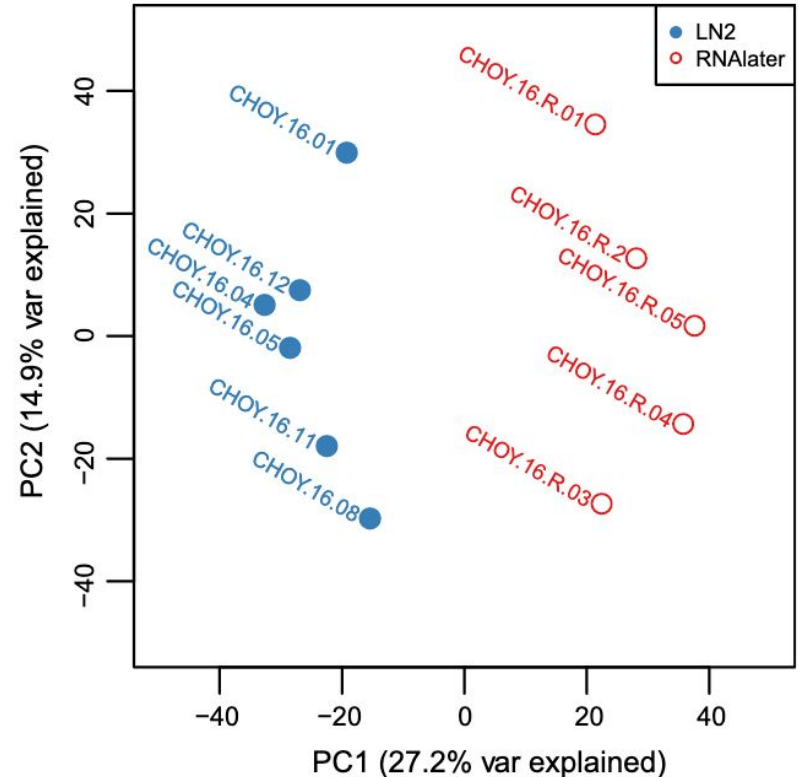
# Avoiding “Batch Effects”

Another term for the systematic effects of non-experimental variables. A large problem if they are **not properly handled by blocking**.

Example sources:

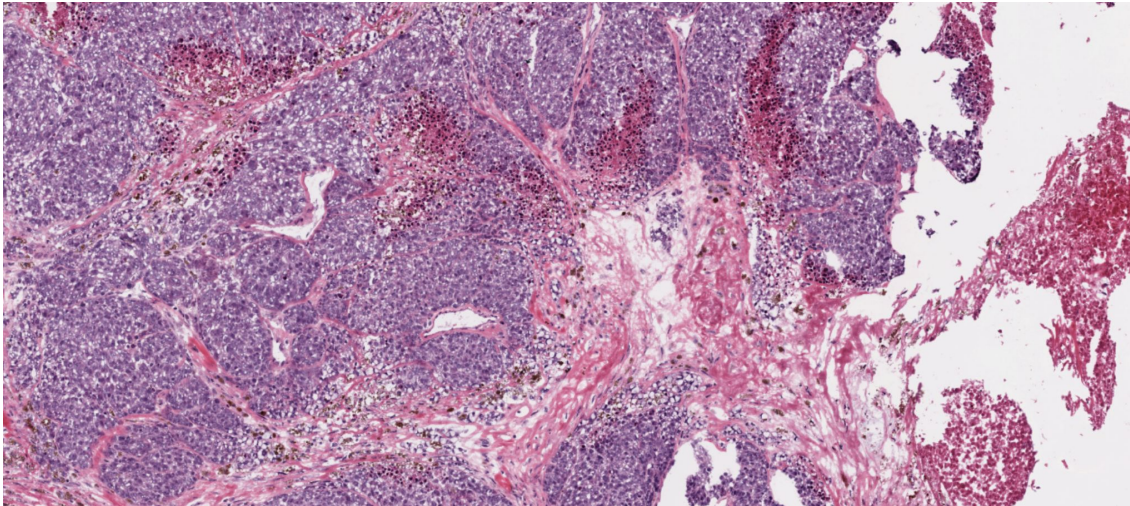
- Technician A processes\* “disease” samples and technician B processes “healthy” samples.
- “Drug” samples are processed\* on one day and “placebo” samples are processed on a different day.

\*: This includes sequencing!



# Single-cell Considerations: Biological

What types of cells are you interested in studying (and how different are they from each other)? How abundant are they expected to be in your sample?

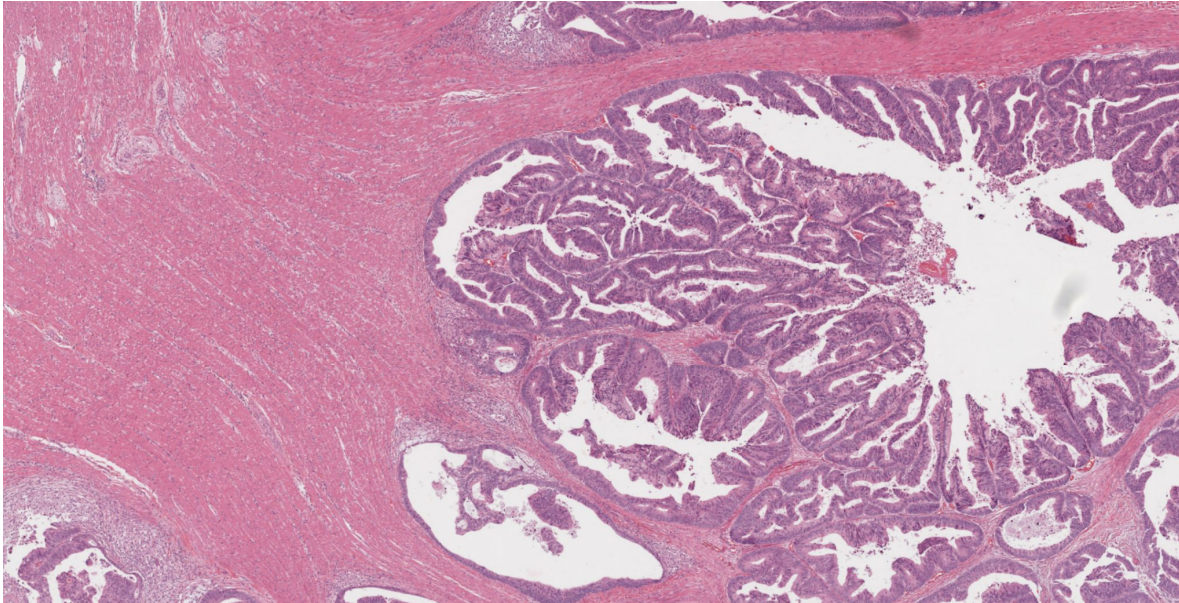


Consider: What types of cells would you expect to find in this skin sample? What is the most common type? Can you distinguish them based on gene expression or epigenetic profile?



# Single-cell Considerations: Biological

Are the cells in your sample challenging to handle (cell walls, irregular shapes, large variation in size, e.g.)? Are they preserved, or fresh?



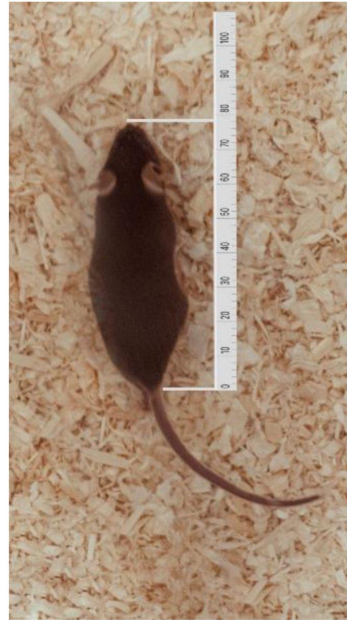
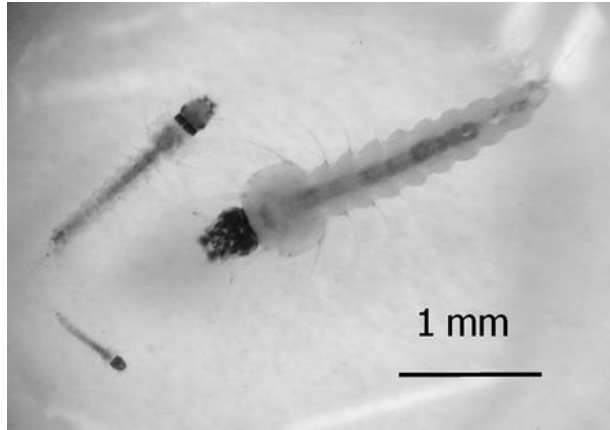
Compare epithelial cells and myocytes in this colon section.

Will myocytes dissociate cleanly?

Single-nucleus sequencing can help with difficult cell shapes!

# Single-cell Considerations: Biological

Are your samples very small (e.g., from young/embryonic individuals)?



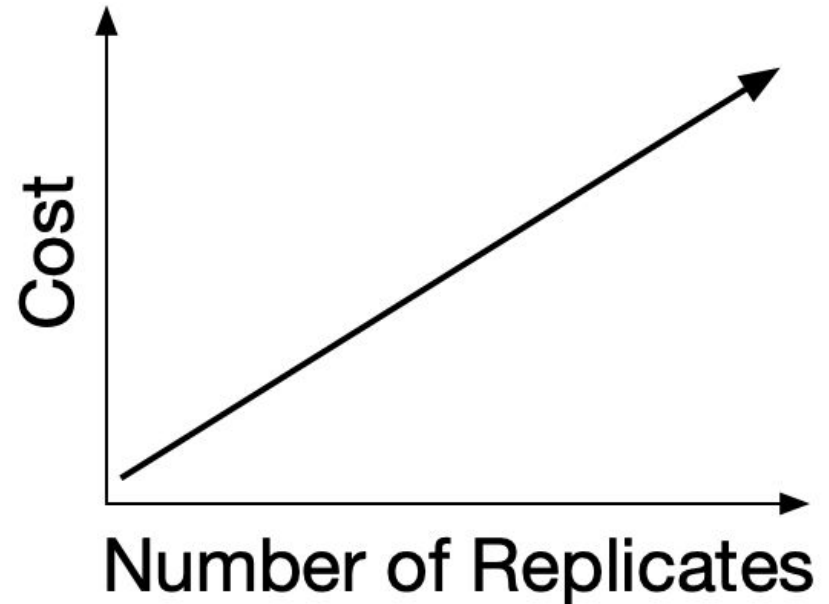
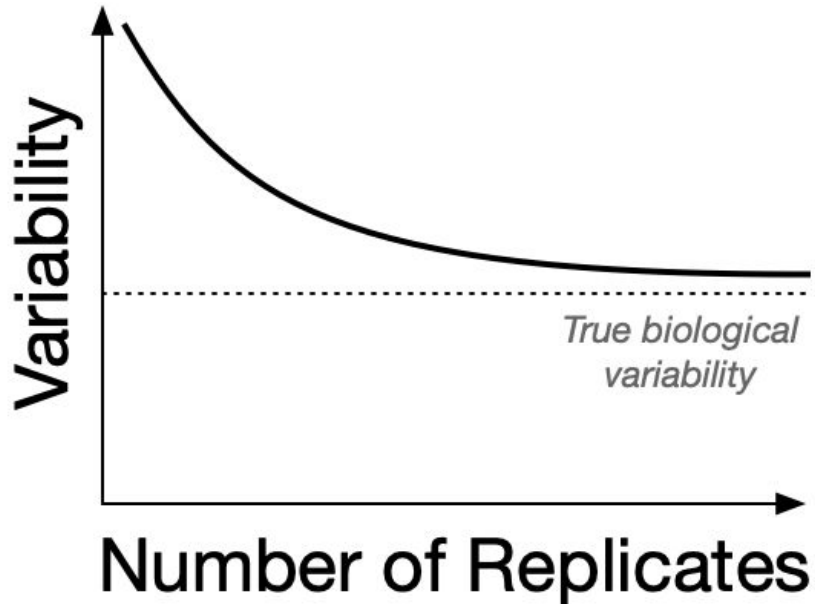
(A)

If you have very small samples, you may have to *pool* multiples together (tag them if you can!!)

Be aware of your sample handling protocols and avoid batch effects!

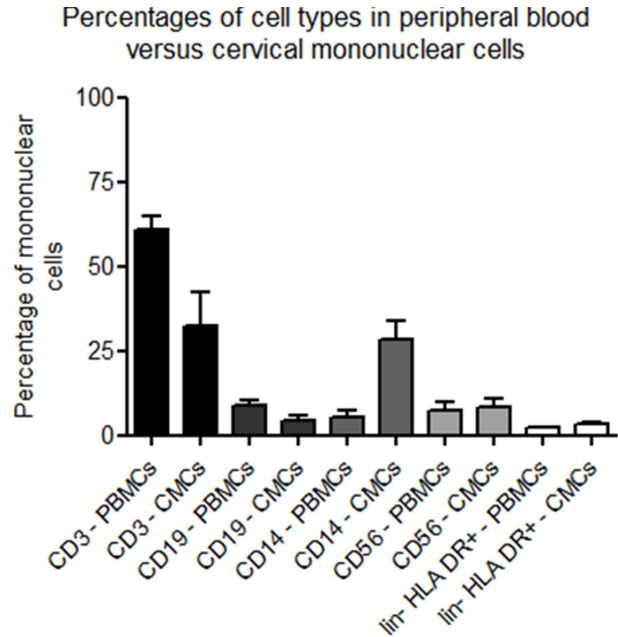
# Single-cell Considerations: Technical

How many replicates are you planning to collect?



# Single-Cell Considerations: Technical

How many total cells are you planning to study? Are you studying a rare type?

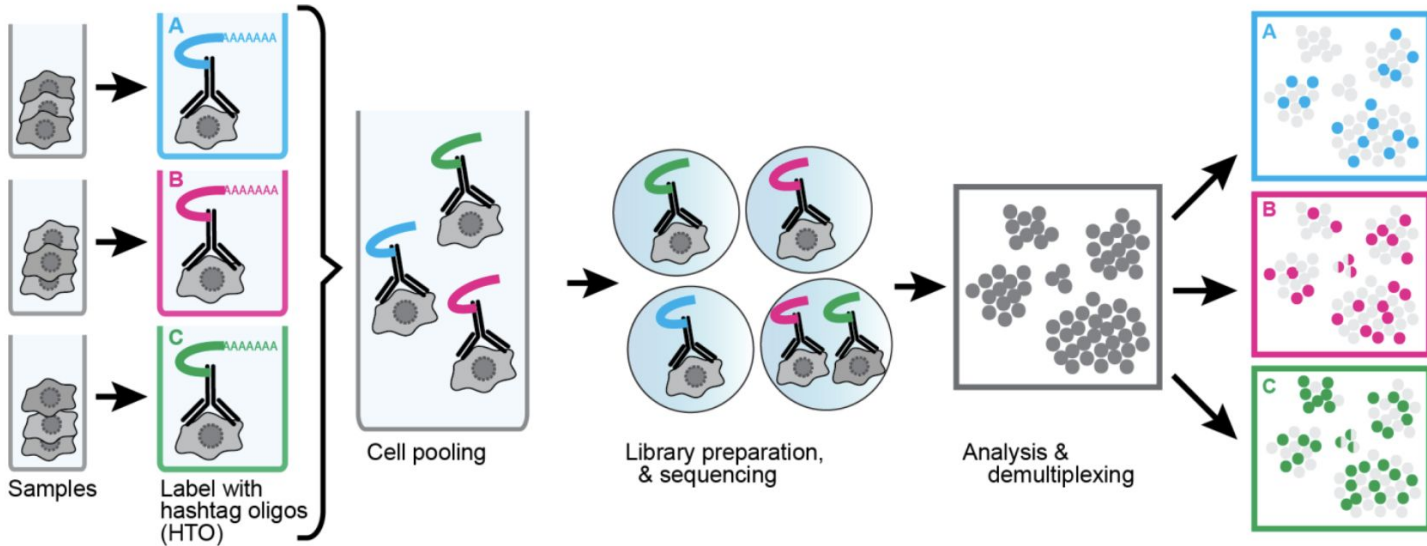


The expectation is that your sample will have cells in proportion to their relative abundance.

Flow sorting can help enrich for specific types, if they can be tagged!

# A Special Call-out to Cell Tagging (Hashing)

This is coming up multiple times in this tutorial: please use tagging!



# Deciding on Cell Number

Target number of cells depends on the factors we have discussed in the previous slides! There are some tools to help, though:

- SCOPIT: sample size calculator and power estimation tool  
[https://alexdavisscs.shinyapps.io/scs\\_power\\_multinomial/](https://alexdavisscs.shinyapps.io/scs_power_multinomial/)
- scPower: multi-sample power estimation for scRNAseq:  
<http://scpower.helmholtz-muenchen.de/>
- Tagging cells with oligos for multiplexing:  
<https://www.10xgenomics.com/blog/answering-your-questions-about-sample-multiplexing-for-single-cell-gene-expression>

# Let's Discuss an Example

Your colleague has skin biopsy samples from patients with an inflammatory disorder. They would like to study the immune cells that are found at skin lesions. The rarest type of cell they are interested in sampling occurs at **1%** of cells in skin tissue.

How many cells should they capture if they would like to sample **at least 100** of the rarest cell?

# Let's Discuss an Example

Your colleague has skin biopsy samples from patients with an inflammatory disorder. They would like to study the immune cells that are found at skin lesions. The rarest type of cell they are interested in sampling occurs at **1%** of cells in skin tissue.

How many cells should they capture if they would like to sample **at least 100** of the rarest cell?

Suppose they want to multiplex 12 patients because the CellPlex reagents support up to 12 samples. Would this experiment be feasible to run?



# Let's Discuss an Example

Your colleague also thinks there will be sex differences in immune cells from the biopsies, so they sample **six female** and **six male** individuals. If **up to 8** samples can be prepared at a time (i.e., in a single batch), how would you advise your colleague to handle the samples to minimize batch effects?

# QA/QC of Single-Cell Data

*Quality assurance (QA)*: Techniques and processes that *avoid* errors and defects in the data or results. These are usually performed in the sample handling and data generation steps of the experiment.

*Quality control (QC)*: Techniques and procedures that *remove* errors and defects in the data or results. These are usually performed once the data have been delivered to you, before you begin analysis.

# Quality Assurance of Single-cell Data

In brief, use good laboratory technique and sample handling practices! These somewhat depend on the material and protocol you are working with. But:

- Use the best-quality samples you can get
- Minimize freeze-thaw cycles
- Isolate samples to minimize cross-contamination

On the protocol steps you can, include a “blank” or “control” sample. **Note that is is different from an experimental control!**

# MiSeq Run vs. Full NovaSeq Run

The UMGC may collect data from a MiSeq run on your sample before running it on the NovaSeq.

These QC metrics are accurate for a shallow sequencing run:

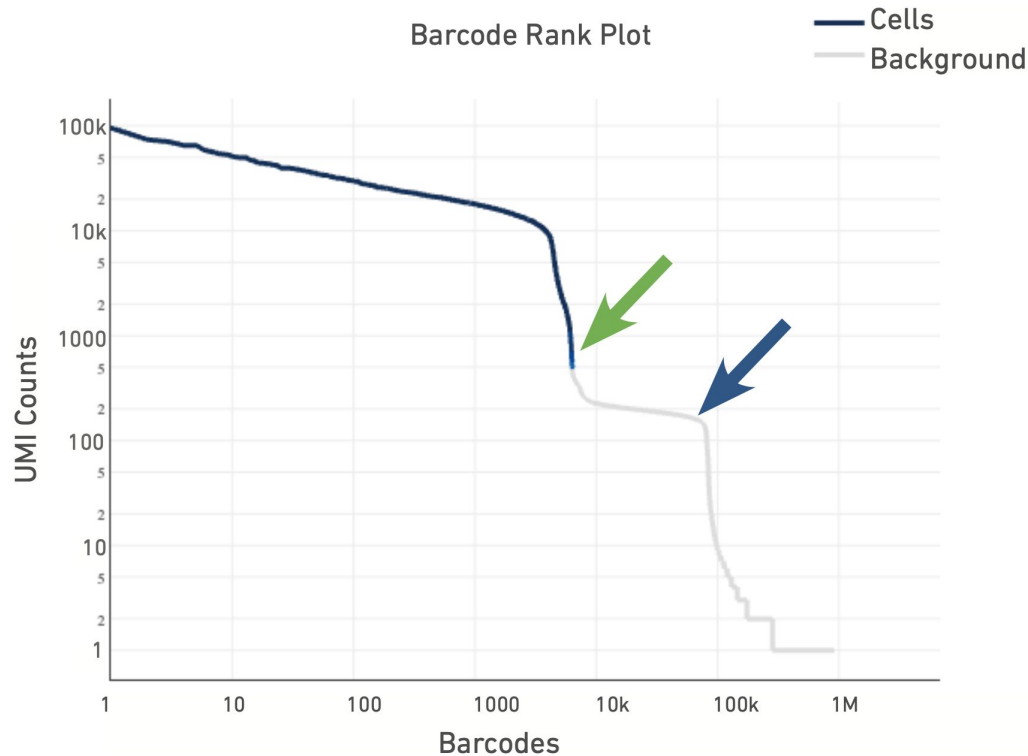
- Valid barcodes
- Valid UMIs
- Reads mapped to transcriptome

Metrics involving cells (reads in cells, cell numbers, etc) are **not** accurate at shallow depths.

See article from 10X (also in handout):

<https://kb.10xgenomics.com/hc/en-us/articles/360054613831-Can-I-perform-shallow-sequencing-to-assess-the-quality-of-Single-Cell-3-Gene-Expression-libraries->

# 10X QC: Cell Ranger

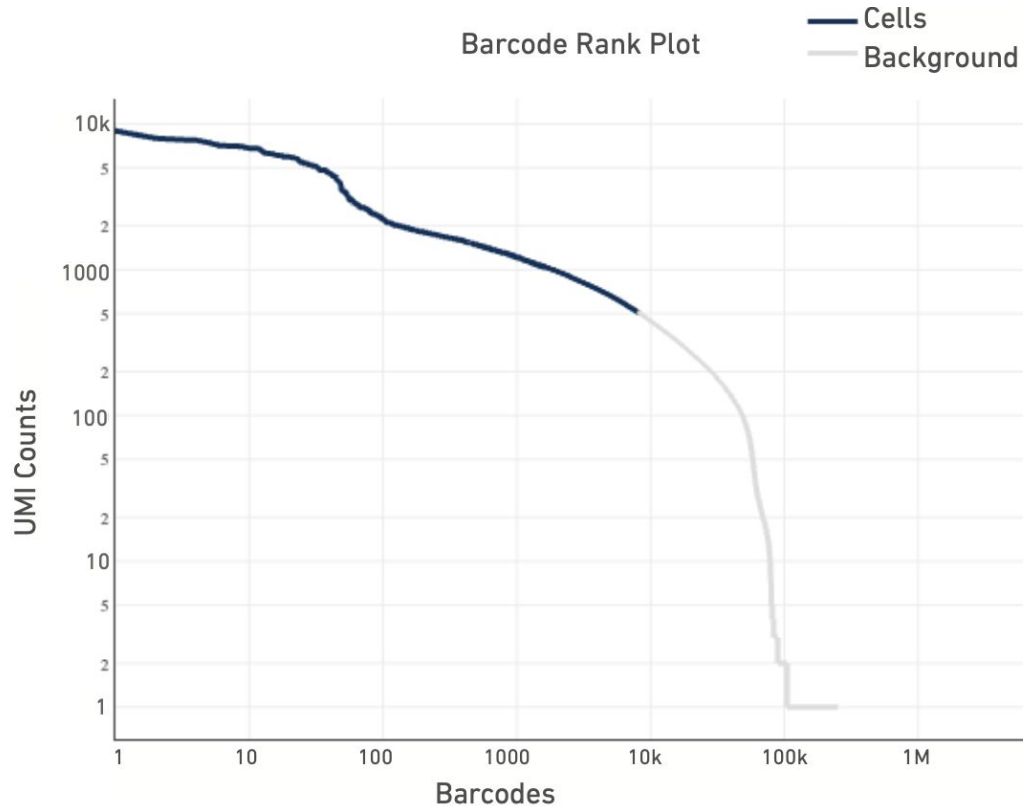


“Cliff and Toe” plot from Cell Ranger. Barcodes ranked from largest UMI count to smallest along X-axis.

A “good” sample will have a sharp drop-off between “cells” and “background.”

Indicates a clean separation between cell-containing droplets and empty droplets.

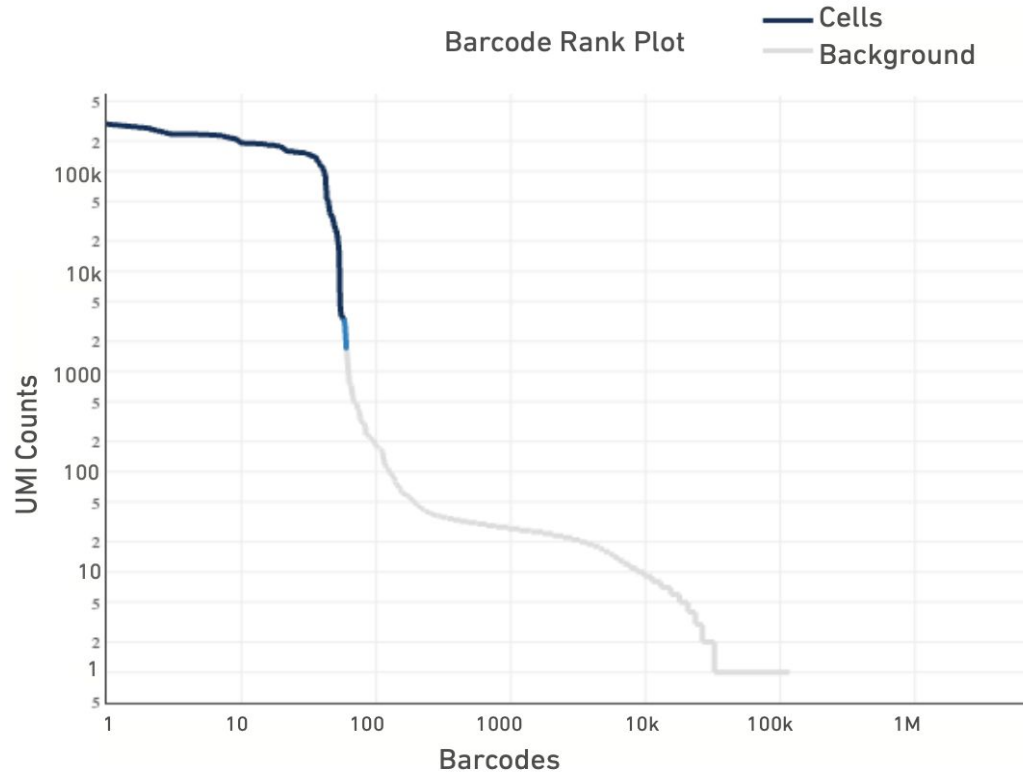
# 10X QC: Cell Ranger



A “bad” sample will have a smooth drop-off between “cells” and “background.”

Indicates no clear separation between cell-containing droplets and empty droplets.

# 10X QC: Cell Ranger



A “bad” sample can also have a sharp drop-off, but much fewer barcodes than expected.

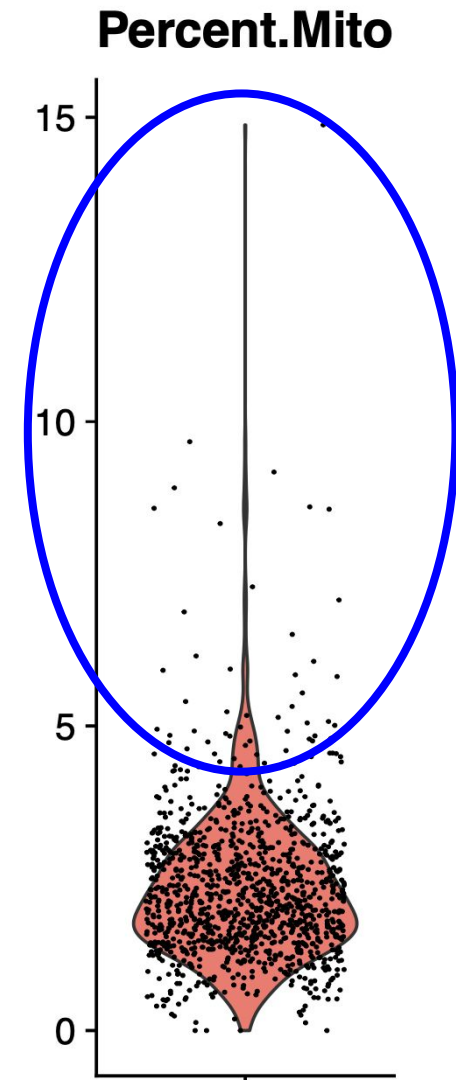
Indicates many fewer cells than expected were in the sample.

# 10X QC: After Cell Ranger

How many cells are “dead?”

Look at mitochondrial signal! Dead cells or cellular debris will mostly yield mitochondrial sequence.

Remove cells that have a relatively high expression of mitochondrial genes (scRNAseq) or where most reads come from the mitochondrial genome (scATAC-seq).



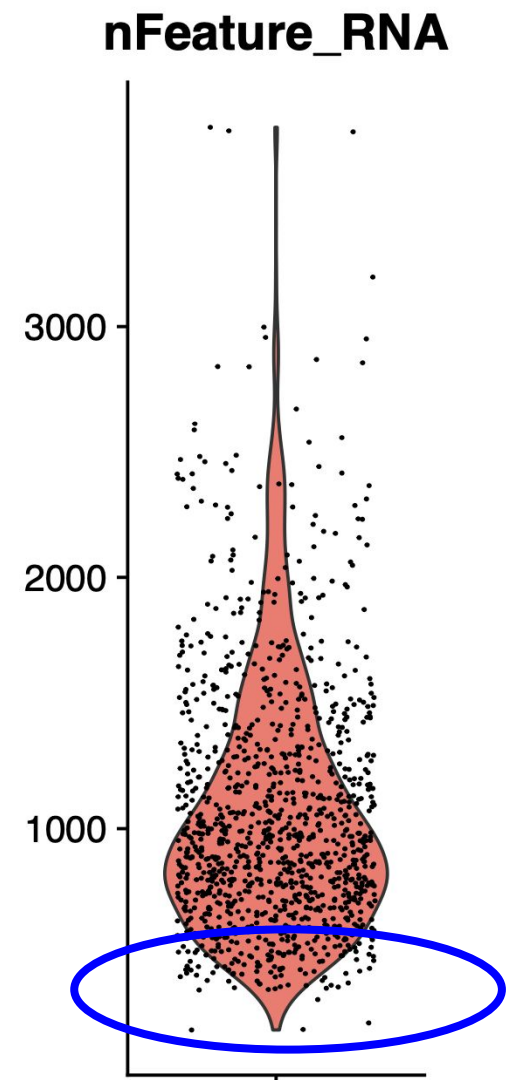


# 10X QC: After Cell Ranger

How many cells are “empty?”

Look at the number of genes detected per-cell. Empty reaction vessels will yield very small quantities of sequencing reads or yield mostly non-biological sequences.

Remove cells that have very low gene expression overall (scRNAseq) or very low mapping rate to genome (scATAC-seq).



# 10X QC: After Cell Ranger

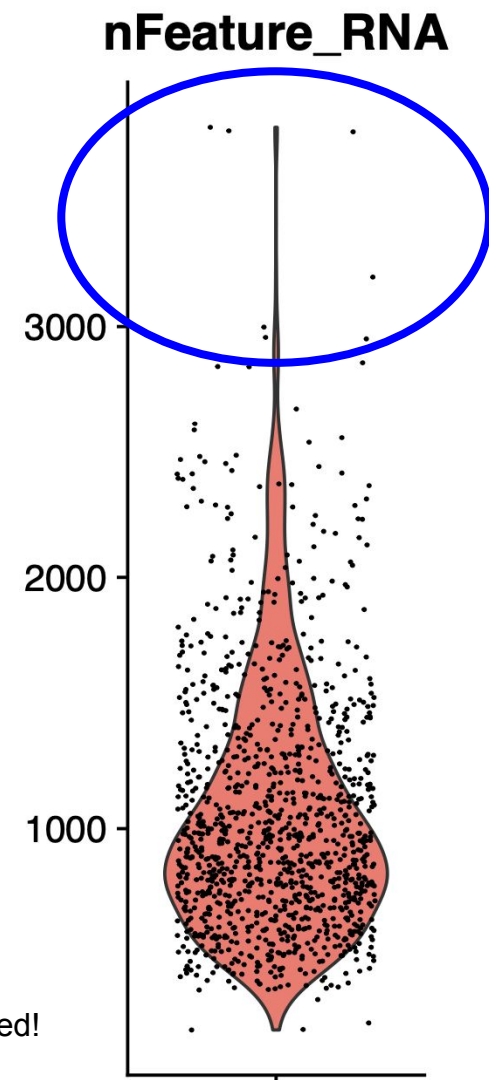
How many of my cells are “multiplets?”

Look at number of genes detected per-cell\*, too! Multiplets will have an unusually high number of expressed genes or quantity of sequencing data.

Remove cells that have very high gene expression overall (scRNAseq) or very high sequencing yield relative to the other cells (scATAC-seq).

If you have tagged your cells, you can look at the distribution of cell hash tags, too.

\*: If your data are hash-tagged, there are other explicit methods to identify multipliers. Stay tuned!

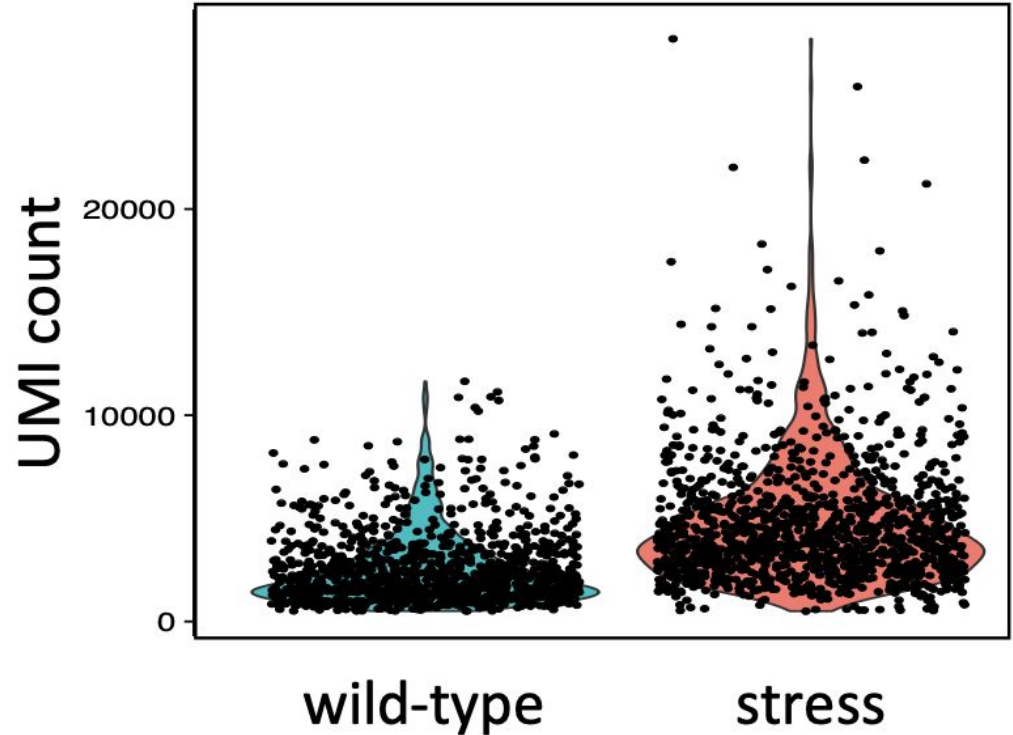


# 10X QC: After Cell Ranger

UMI counts plots from Seurat.

Cells should have a high UMI count.

Indicates that each cell had sufficient tagging of unique transcripts/fragments.

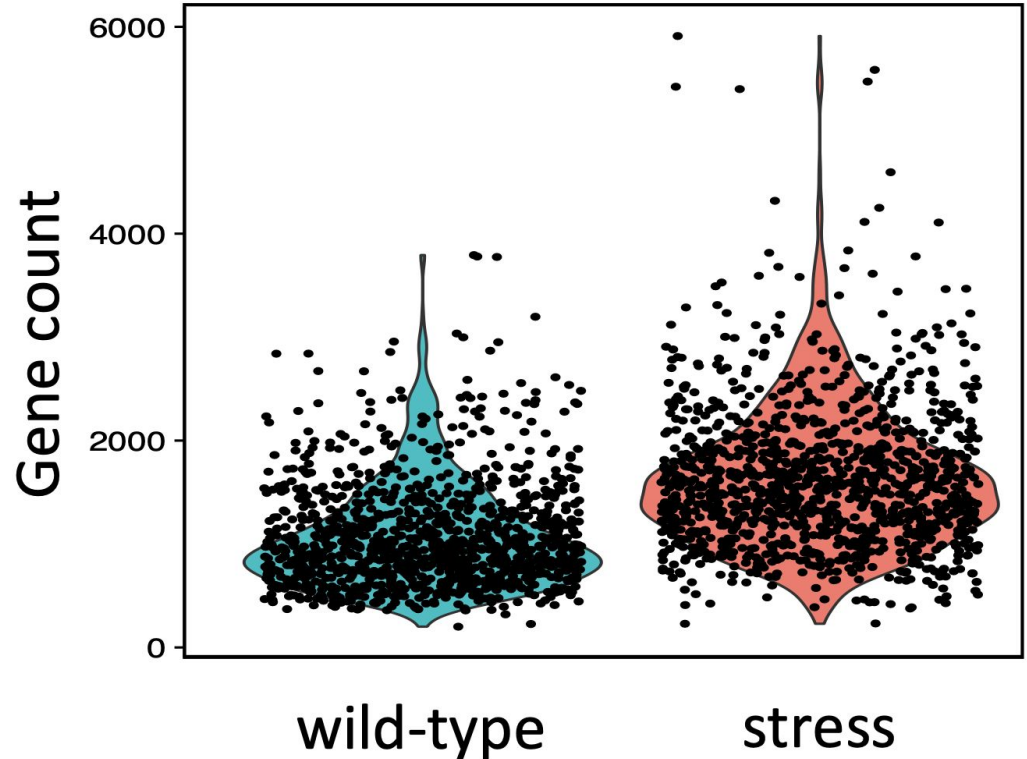


# 10X QC: After Cell Ranger

Gene counts plots from Seurat.

Cells should have a high\* gene count.

Indicates that each cell had a broad sampling of genes that it is expressing, and not dead.



\*: depending on the types of cells!

# “So, are my data good? Or, are they bad?”

Depends on the biology of the study system and the hypotheses being tested!  
Single-cell methods are not as codified as other fields of genomics.

But, there are a few general metrics that apply to all single-cell datasets:

- There should be few “dead,” “empty,” or “multiplet” cells.
- All of your hash tags should be detected, and they should have relative abundances proportional to their input.
- The cell count should be as close to your expected input number as possible.
- Most of your reads should map to the genome or transcriptome.

# “So, are my data good? Or, are they bad?”

For scientific purposes, it is harder to call a dataset “good” or “bad.”

Instead, ask:

- Do you see the patterns of gene expression or sequencing coverage that you would expect from the input material?
- Do the cell types you identify align with expectations based on the input material?
- Are the genes or genomic regions of interest covered by the sequencing data?

These are based on your ***hypotheses!***

# Software Tools for Analyses

Adam Herman, PhD

## Color key



**srun**  
(interactive mode)



**sbatch**  
(batch mode)



# Color key

**srun**

(interactive mode)

```
[10:08:17 riss] [aherman@ln0004] (~) $ srun --nodes=1 --ntasks-per-node=24 --cpus-per-task=1 --time=03:00:00 --mem=50GB --partition=interactive --account=riss --x11 --pty bash
srun: job 146726397 queued and waiting for resources
srun: job 146726397 has been allocated resources
[10:08:48 riss] [aherman@acn17] (~) $
```

**sbatch**

(batch mode)

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=64
#SBATCH --cpus-per-task=1
#SBATCH --time=6:00:00
#SBATCH --mem=600GB
#SBATCH --account=riss
#SBATCH --mail-type=ALL
#SBATCH --mail-type=END
#SBATCH --mail-user=aherman@umn.edu
#SBATCH --partition=ag2tb
#SBATCH -o "run_cr_small-%j.out"
#SBATCH -e "run_cr_small-%j.err"

module load umgc
module load cellranger/6.0.0
```

[10:12:10 riss] [aherman@ln0004] (~) \$ sbatch cr\_example.slurm

# Workflow overview

reads  $\implies$  counts  $\implies$  summaries  $\implies$  analyses  $\implies$  sharing

# From reads to counts



cellranger

The diagram features a light purple rounded rectangle on the left side. Inside it, there are two smaller rounded rectangles. The top one is light gray and contains the text 'cellranger'. The bottom one is light yellow and contains the text 'kallisto | bustools'.

Used the most by far

Can count GEX, HTO, ADT, VDJ, ATAC at once

kallisto | bustools

EXTREMELY FAST

Counting of HTO needs to be done with other software

# From reads to counts

cellranger



kallisto | bustools

## cr\_run

Summary Analysis

27,871

Estimated Number of Cells

50,090

Mean Reads per Cell

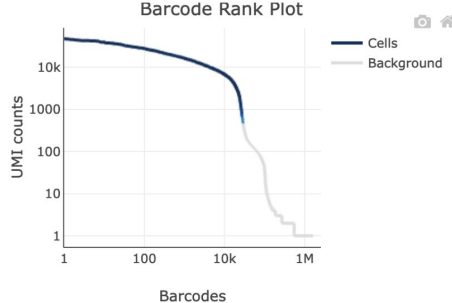
1,930

Median Genes per Cell

### Sequencing

Number of Reads	1,396,068,925
Number of Short Reads Skipped	0
Valid Barcodes	96.3%
Valid UMIs	99.9%
Sequencing Saturation	78.2%
Q30 Bases in Barcode	96.4%
Q30 Bases in RNA Read	94.9%
Q30 Bases in UMI	95.8%

### Cells



Estimated Number of Cells	27,871
Fraction Reads in Cells	94.0%
Mean Reads per Cell	50,090
Median Genes per Cell	1,930
Total Genes Detected	23,317
Median UMI Counts per Cell	5,321

# From counts to clean data



Seurat

Like cellranger, the most commonly used (in RIS!)

Very good vignettes

scanpy

More software-principles oriented

In my experience, a bit more efficient in terms of speed and memory

# From counts to clean data



Seurat

The diagram consists of a large, light green rounded rectangle on the left side of the slide. Inside this green rectangle, there are two smaller rounded rectangles stacked vertically. The top one is light blue and contains the text 'Seurat'. The bottom one is light yellow and contains the text 'scanpy'.

There is no right answer here

Do you like R or Python

Regardless, there's some prep you're going to need to do

# From counts to clean data

A diagram on the left side of the slide. It features a large, light green rounded rectangle in the background. Overlapping this are two smaller rounded rectangles: a blue one on top and a yellow one on the bottom. The word 'Seurat' is centered in the blue rectangle, and 'scanpy' is centered in the yellow rectangle.

Seurat

scanpy

X-forwarding ([Windows](#), [Mac](#))

Local package installs (CRAN, Bioconductor)

[notebooks.msi.umn.edu](https://notebooks.msi.umn.edu)

[conda environments](#)

# From clean data to whatever you like!



Seurat

Slingshot

Monocle

scanpy

scvi-tools

scVelo

Seurat and scanpy have absorbed many functions

There may or may not be some degree of wrangling involved

While errors are frustrating, beware some just working



# From clean data to whatever you like!

Seurat

Slingshot

Monocle

More R package installation

scanpy

scvi-tools

scVelo

More conda installation

# Sharing your data



VISION

Rmarkdown

Jupyter  
notebooks

Gene  
Expression  
Omnibus  
GEO

# Sharing your data



VISION

Rmarkdown

Jupyter  
notebooks

In my experience VISION is the most useful collaboration tool

Rmarkdown is super for reports

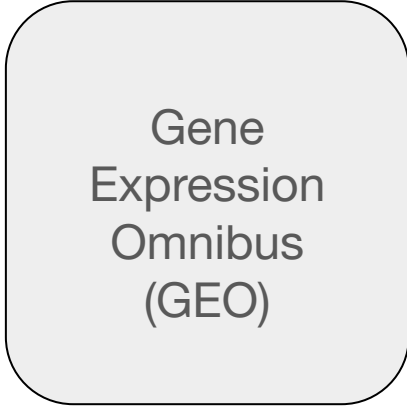
Jupyter notebooks are great for sharing

# Sharing your data

You'll need to adjust your conception of your experiment

It's not awfully painful, though!

See [here](#) for a great overview

A light gray rounded square with a thin black border containing the text "Gene Expression Omnibus (GEO)".

Gene  
Expression  
Omnibus  
(GEO)

# Single-cell Analysis Showcase

Marissa Macchietto, PhD

# Why single-cell analysis?

- Bulk methods average cell profiles in a sample. Often, this is a good first step to finding markers of interest
- In certain experiments, bulk RNA-seq methods can be problematic (due to signal masking or can cause difficulty in interpreting results). A few examples include when:
  - The samples are very heterogeneous (e.g. tumor)
  - A treatment affects a rare cell population in your sample
  - Sample cell compositions change as a result of treatment

# Why single-cell analysis?

- Provides detailed information about individual cells (gene expression, chromatin accessibility, SNPs, copy number variation, protein expression, etc)
- Allows us to study cellular heterogeneity

This can help us:

- discover new cell types/states
- uncover mechanisms of action for drugs (when they target a particular cell type)
- better predict how a disease will evolve (e.g. tumor cells acquiring new phenotypes)
- uncover novel enhancers and promoters + characterize regulatory networks

# Outline of Topics

Clustering and Cell Typing

General Single Cell Analysis

Differential Feature Identification

VISION

Single-cell Data Set Integration (Batch correction, Multimodal Data)

CITE-seq

VDJ Analysis

Trajectory Inference (Pseudotime, RNA velocity)

inferCNV



# Outline of Topics

Clustering and Cell Typing

Differential Feature Identification

VISION

Single-cell Data Set Integration (Batch correction, Multimodal Data)

CITE-seq

Multi Sample/ Multitimodal

VDJ Analysis

Trajectory Inference (Pseudotime, RNA velocity)

inferCNV

# Outline of Topics

Clustering and Cell Typing

Differential Feature Identification

VISION

Single-cell Data Set Integration (Batch correction, Multimodal Data)

CITE-seq

VDJ Analysis

Specific use cases

Trajectory Inference (Pseudotime, RNA velocity)

inferCNV

# General Single-Cell Analysis Steps

- Remove low quality cells (very low feature counts, high mitochondrial content, multiplets)
- Normalize feature expression
- Find variable features
- Reduce data set dimensionality (i.e. PCA, UMAP) using variable features
- Find nearest cell neighbors (i.e. SNN graph construction)

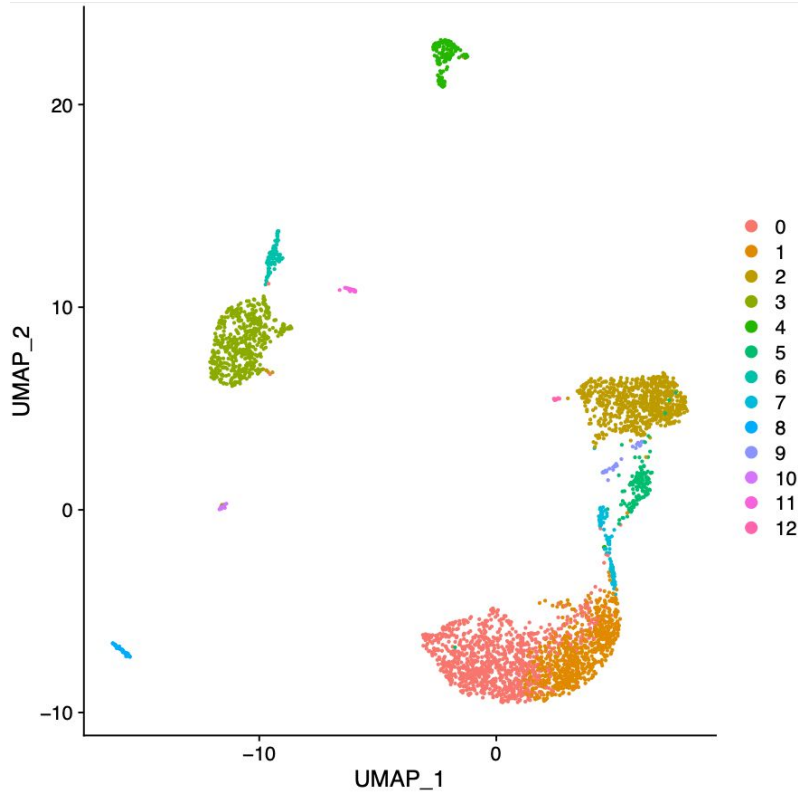


Cell clusters



**Differential Feature Expression**

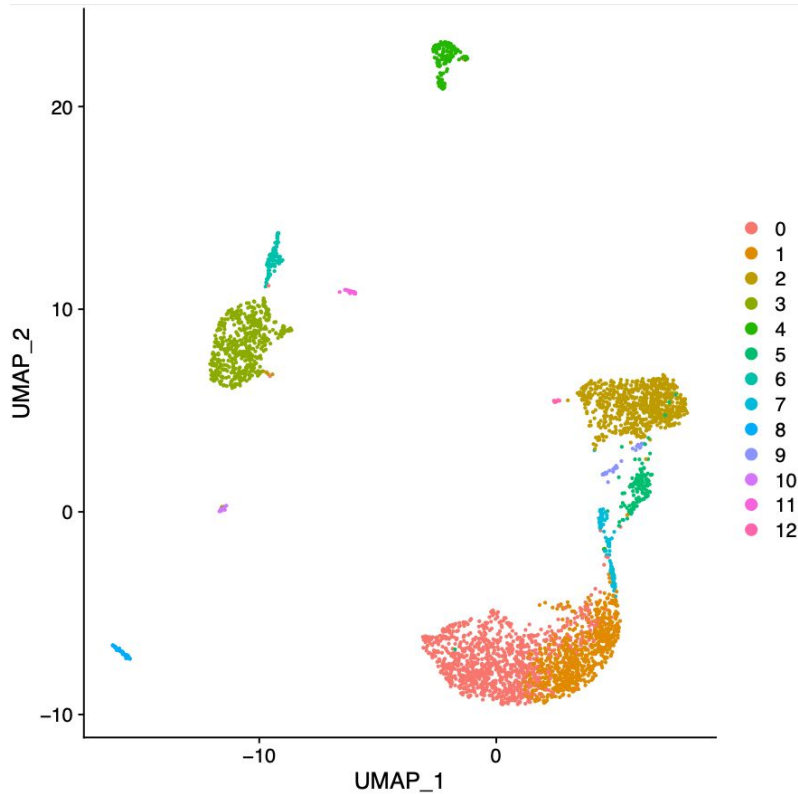
# Cell Clustering



PBMCs (from a patient's serum)  
clustered on transcriptomes using  
Seurat R package

Visualize cell clustering results on  
**nonlinear** dimensional reduction plot  
(UMAP or tSNE)

# Cell Clustering

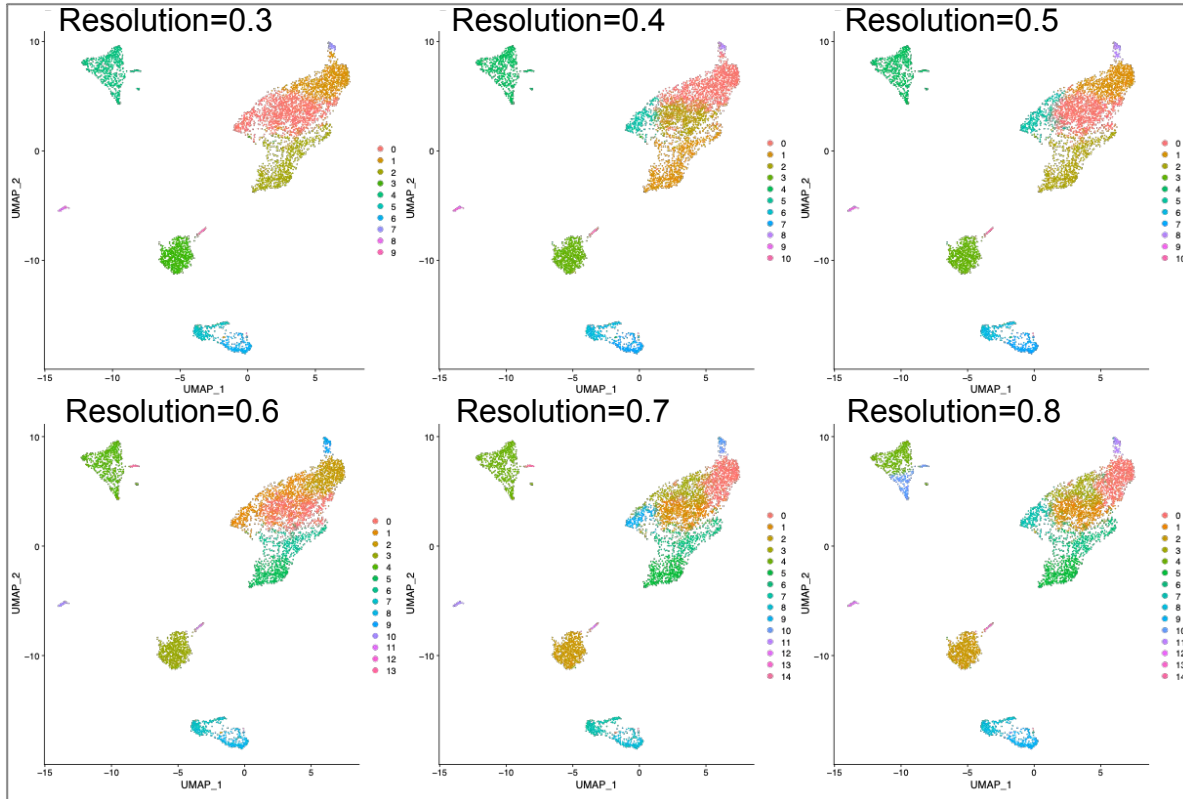


PBMCs (from a patient's serum)  
clustered on transcriptomes using  
Seurat R package

Visualize cell clustering results on  
**nonlinear** dimensional reduction plot  
(UMAP or tSNE)

**Do not be tempted to read too much  
into cell or cluster distances on this  
plot – it may not mean anything due  
to non-linear representation!**

# Cell Clustering



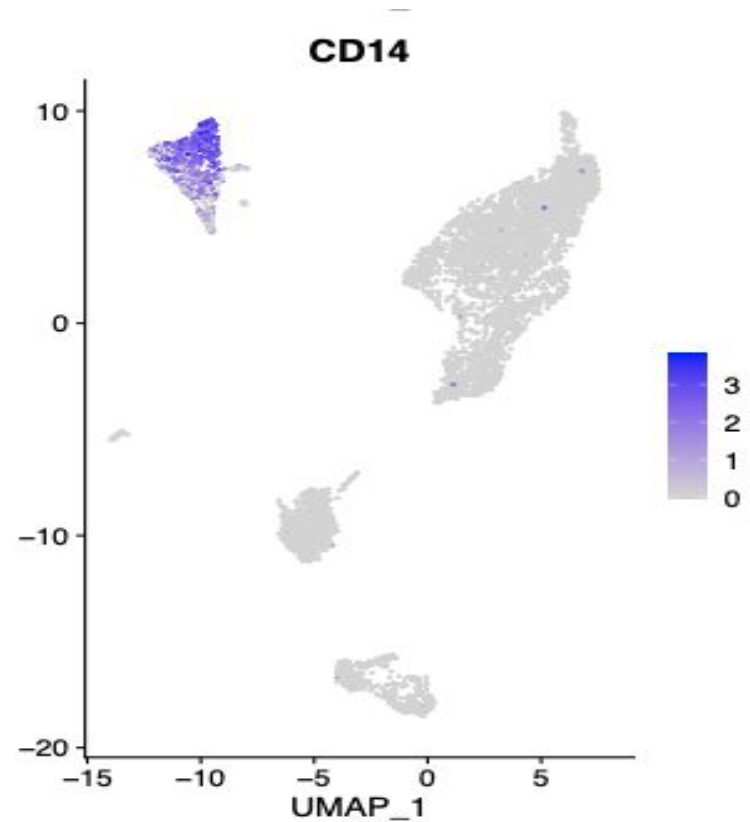
Clustering resolutions are adjustable

Adjust resolution to achieve clustering results that make sense for your experimental objectives

How do I adjust the clustering resolution to get the most accurate result?

# Manual Cell Type Annotations

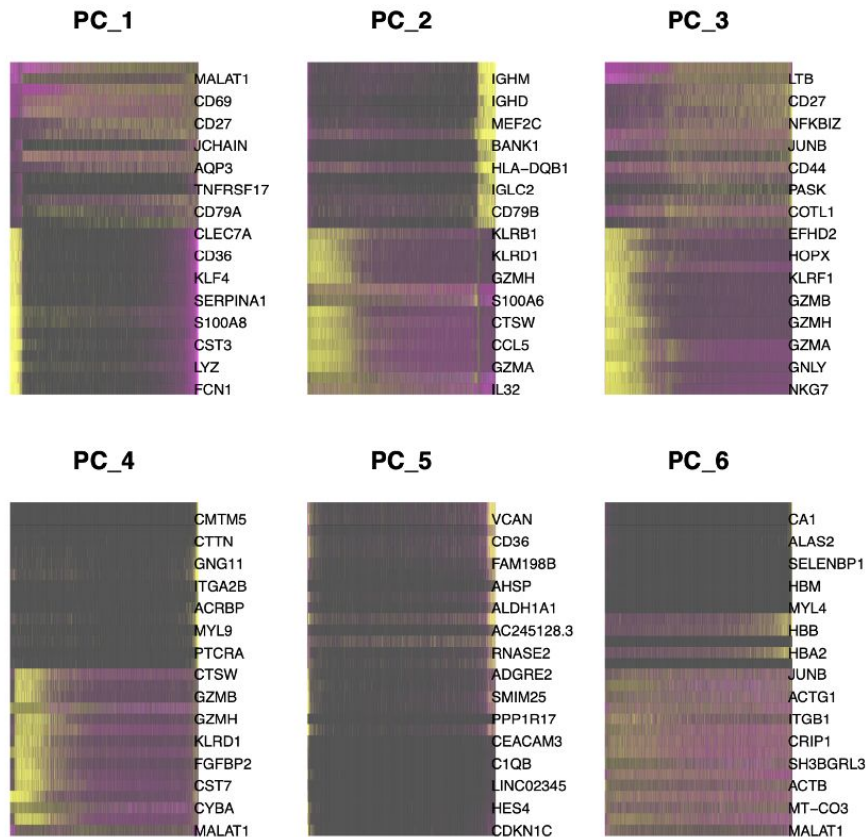
- Use known marker genes





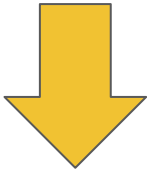
# Manual Cell Type Annotations

- Use known marker genes
- Use PCA loadings

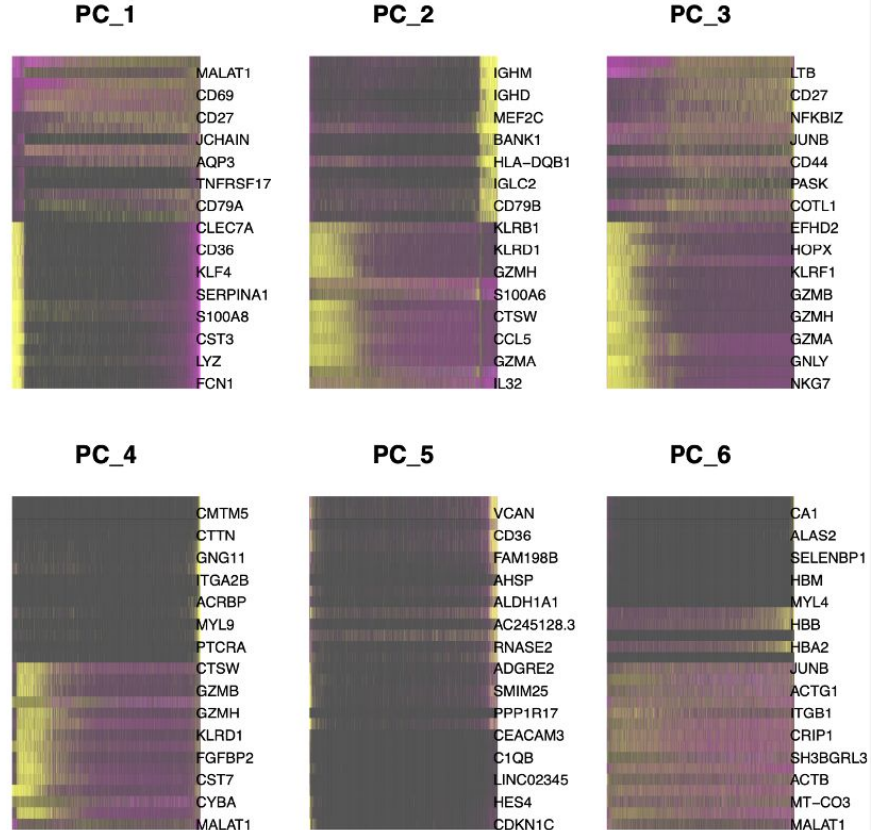


# Manual Cell Type Annotations

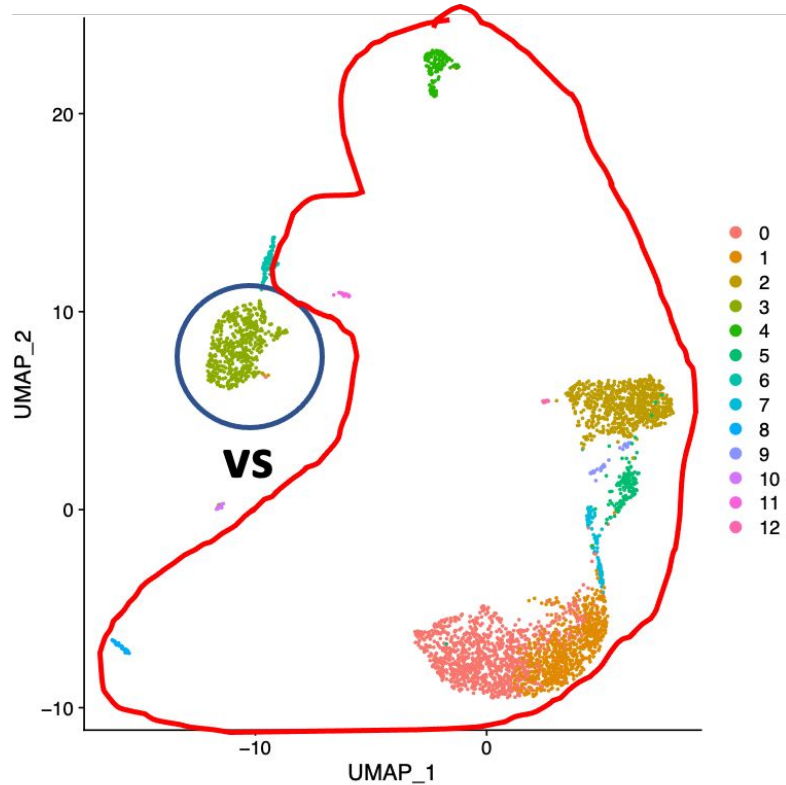
- Use known marker genes
- Use PCA loadings
- Use Differentially Expressed Genes/ Features



All of these can help us distinguish cell types!

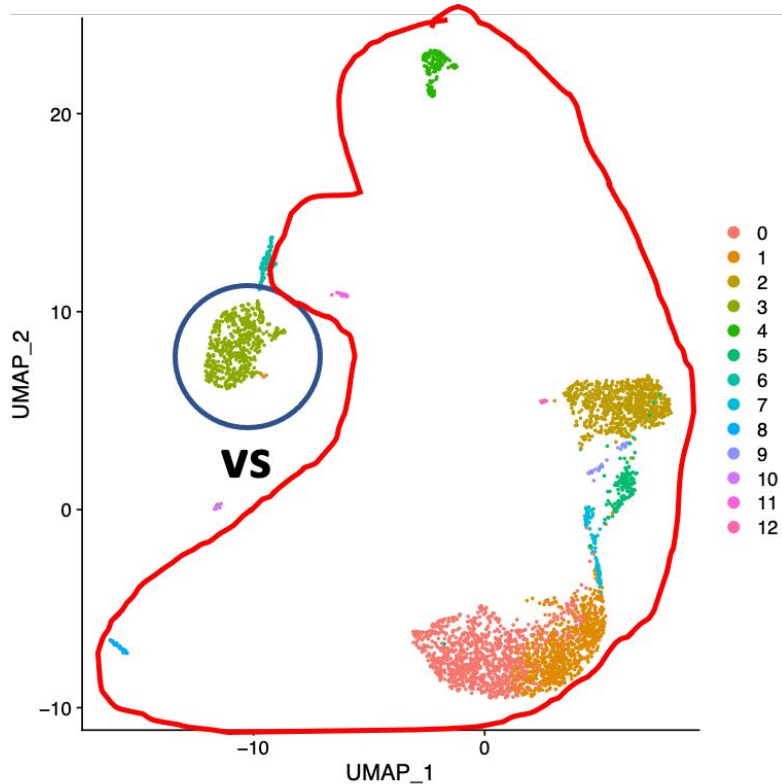


# Differentially Expressed Feature (Gene) Analysis

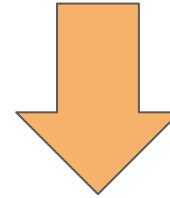


How is the yellow-green cluster different from **all** of the other cells?

# Differentially Expressed Gene Analysis



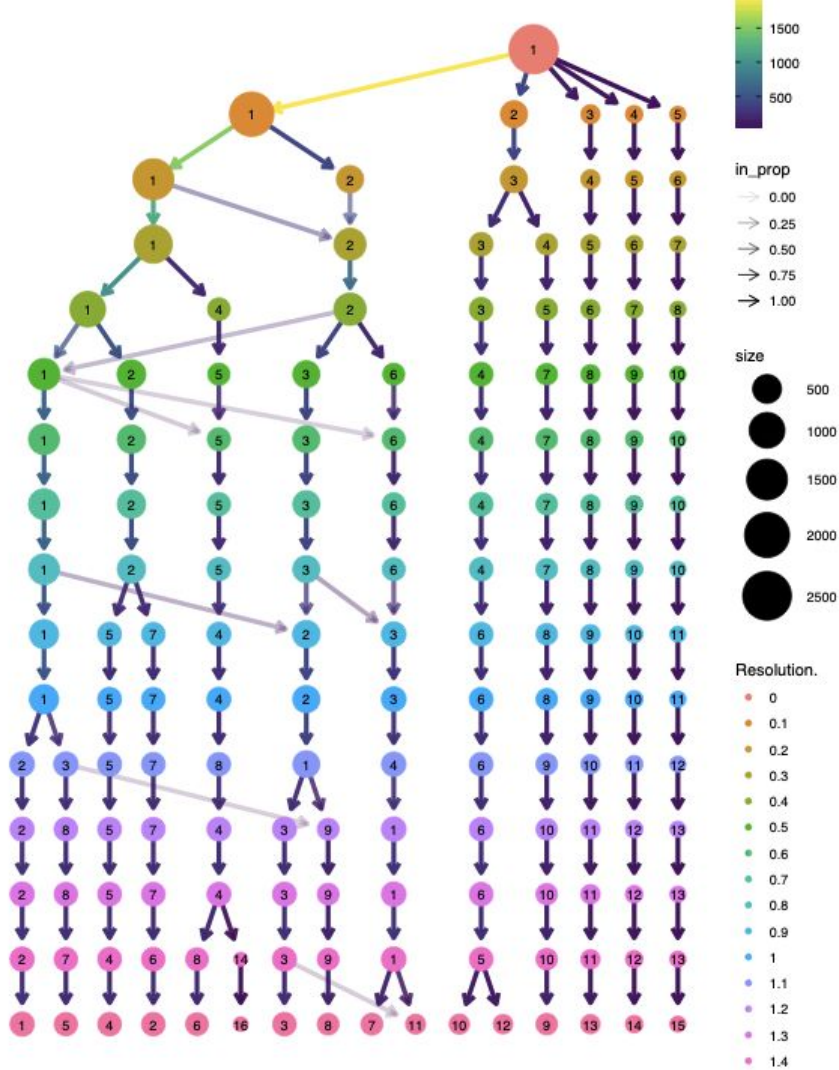
How is the yellow-green cluster different from **all** of the other cells?



This could help us determine that the identity of yellow-green cluster is monocyte-like

# How do I adjust the clustering resolution to get the most accurate result?

- Use the **Clustree** R package (<https://github.com/lazappi/clustree>) to iterate through resolutions and find stable clustering solutions using fun, tree visualizations



# Clustree R package

- Can produce tree visualizations that show cell cluster compositions at different cluster resolutions
- Can easily add right into your single-cell pipeline – uses *Seurat* or *SingleCellExperiment* objects

# How do I adjust the clustering resolution to get the most accurate result?

- Use the **Clustree** R package (<https://github.com/lazappi/clustree>) to iterate through resolutions and find stable clustering solutions using fun, tree visualizations
- Use an **automated single-cell annotation** tool that will annotate individual cells against a reference RNA-seq databases (e.g. SingleR R package or CHETAH R package)

# Automated Cell Type Classification

Many tools available for annotating cell identities and come in different types:



# Automated Cell Type Classification

Many tools available for annotating cell identities and come in different types:

- **Reference-based: Uses RNA-seq reference datasets**
  - SingleR
  - CHETAH
- **Marker-gene based: Uses marker gene lists to probabilistically assign cell types**
  - Cellassign

# Automated Cell Type Classification

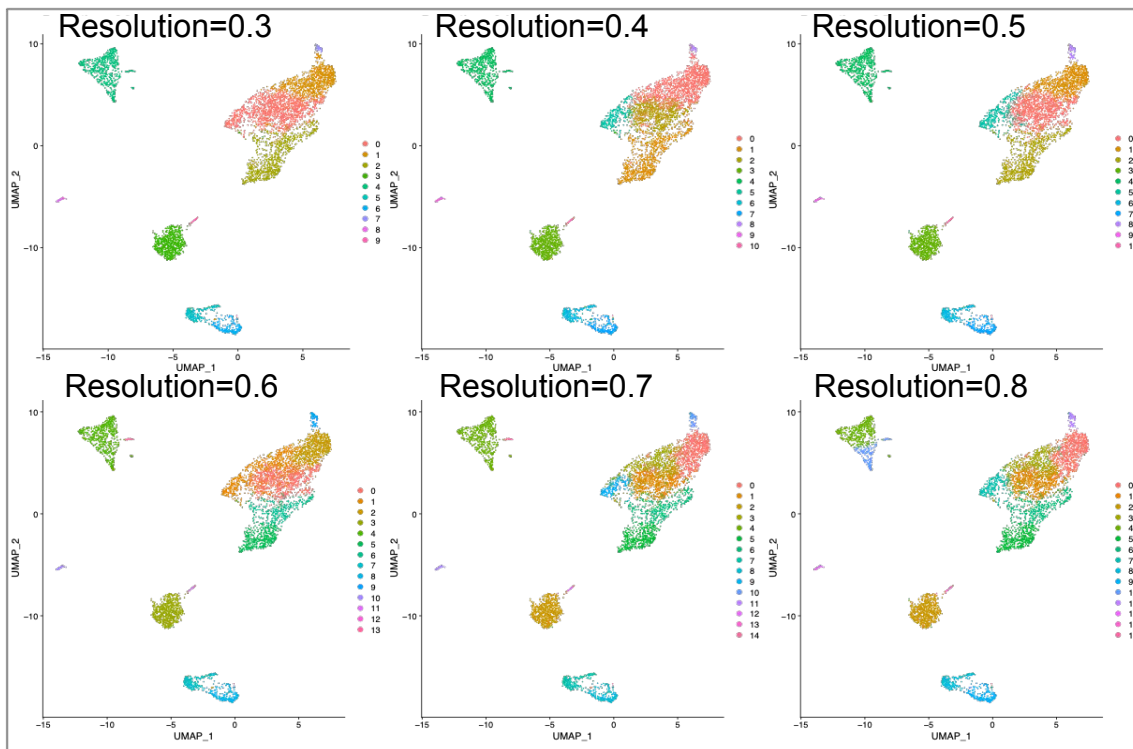
Many tools available for annotating cell identities and come in different types:

- **Reference-based: Uses RNA-seq reference datasets**
  - SingleR
  - CHETAH
- **Marker-gene based: Uses marker gene lists to probabilistically assign cell types**
  - Cellassign

# Reference-based Cell Type Classification

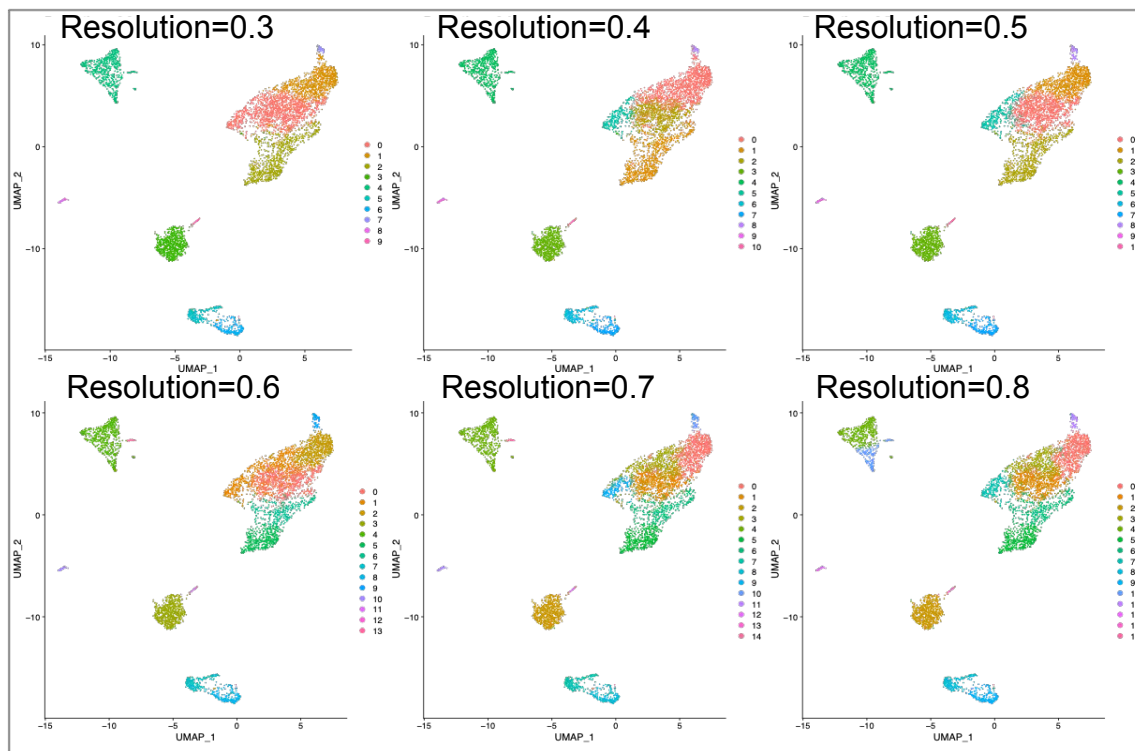
6 Different Clustering Resolutions:

Tool: SingleR  
Reference: Blueprint + ENCODE (bulk RNA-seq)

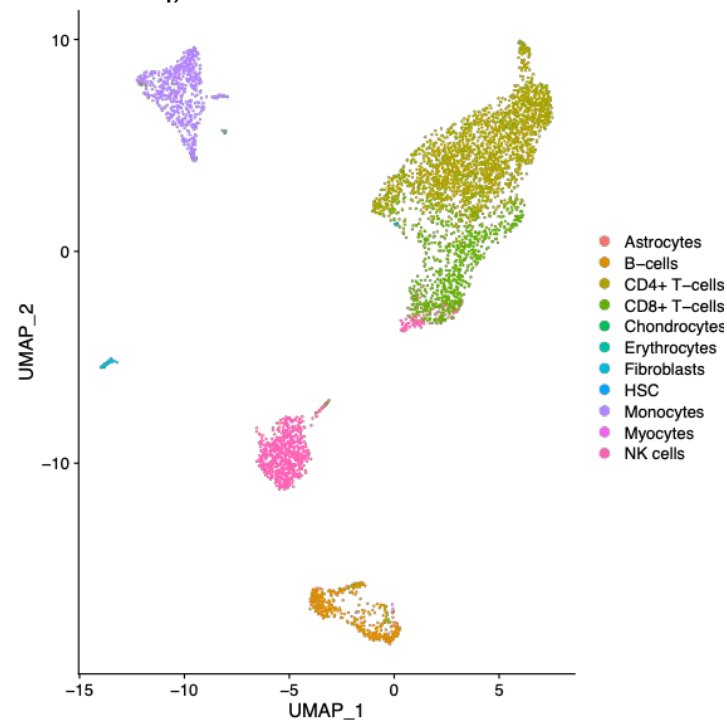


# Reference-based Cell Type Classification

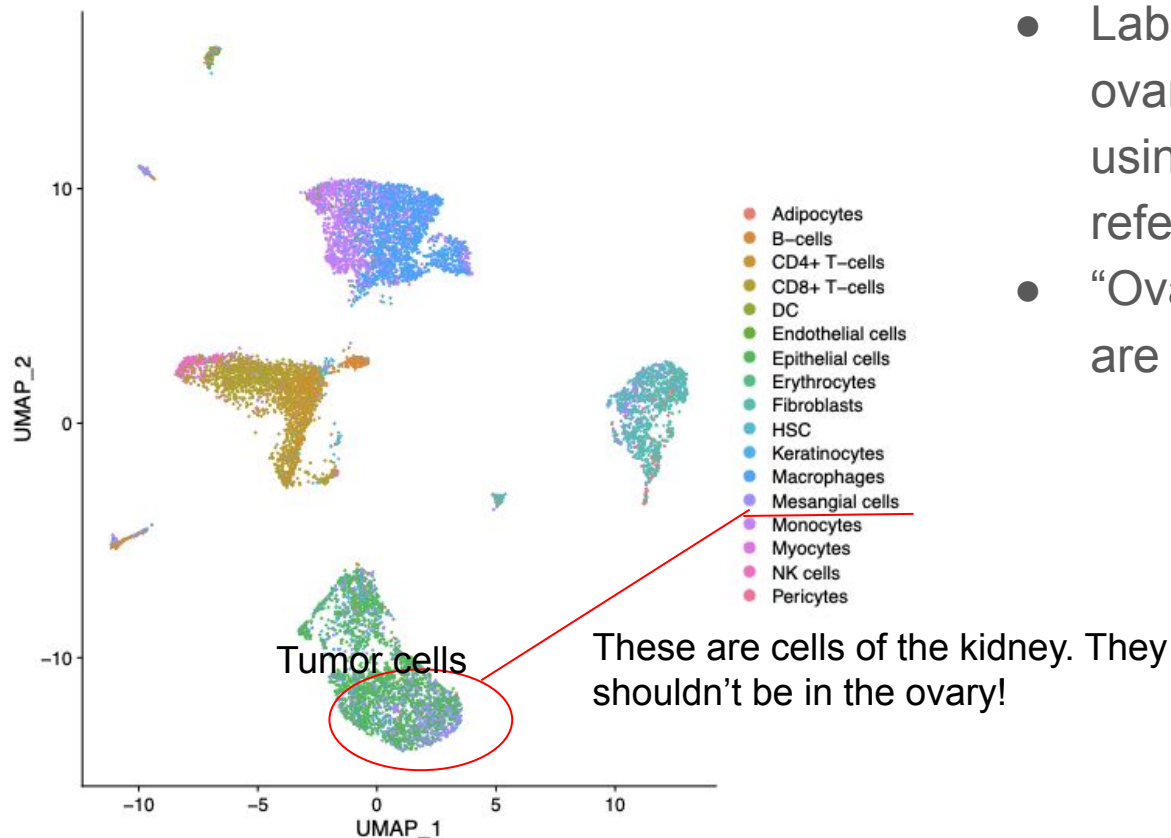
6 Different Clustering Resolutions:



Tool: SingleR  
Reference: Blueprint + ENCODE (bulk RNA-seq)



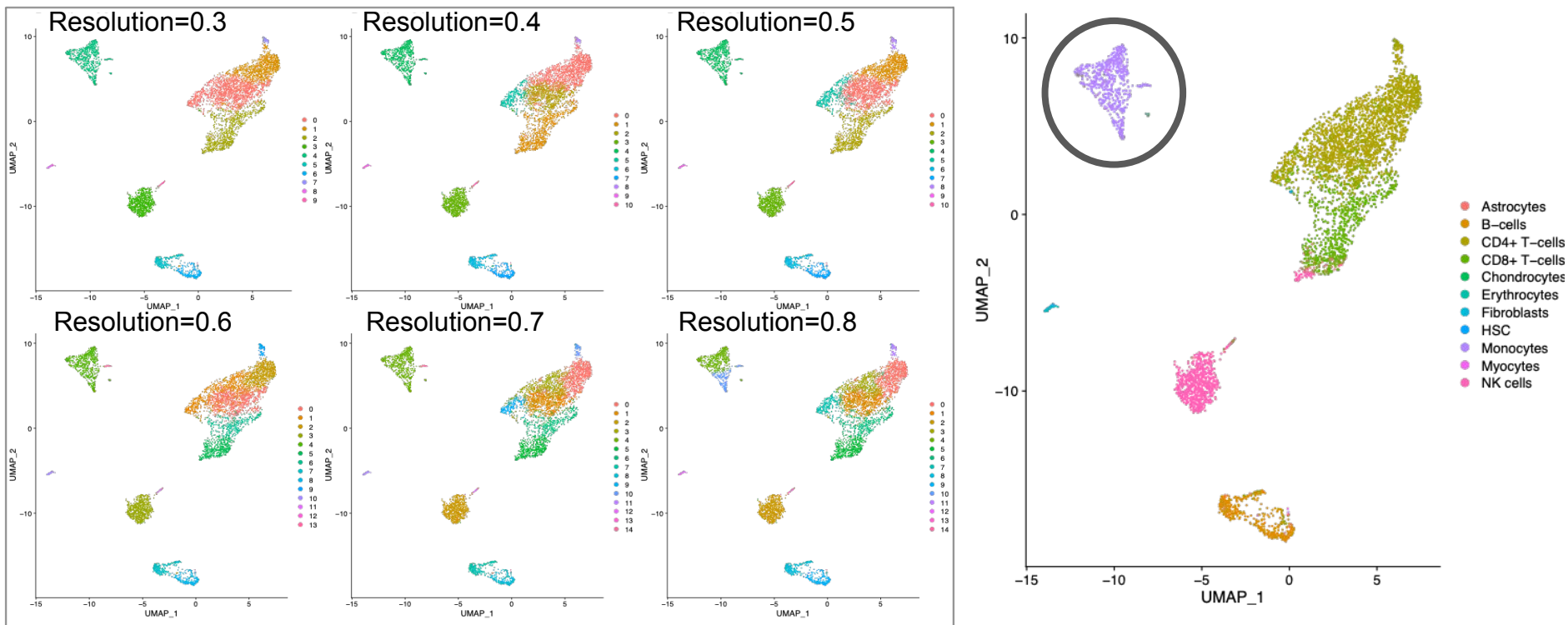
# Cell type mislabeling



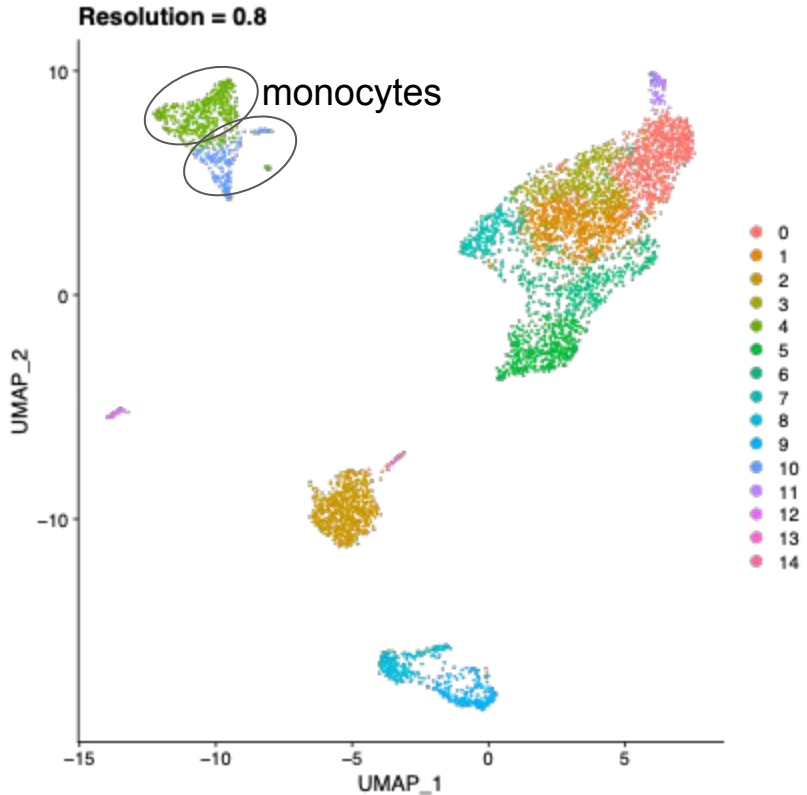
- Labeled individual cells of an ovarian tumor with SingleR tool using Blueprint+ ENCODE as a reference
- “Ovarian tumor epithelial cells” are not in the reference

Good example of cell type mislabeling!

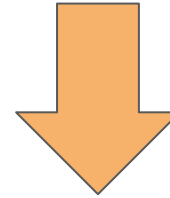
# What if I am interested in CD14+ and CD16+ monocytes?



# What if I am interested in CD14+ and CD16+ monocytes?



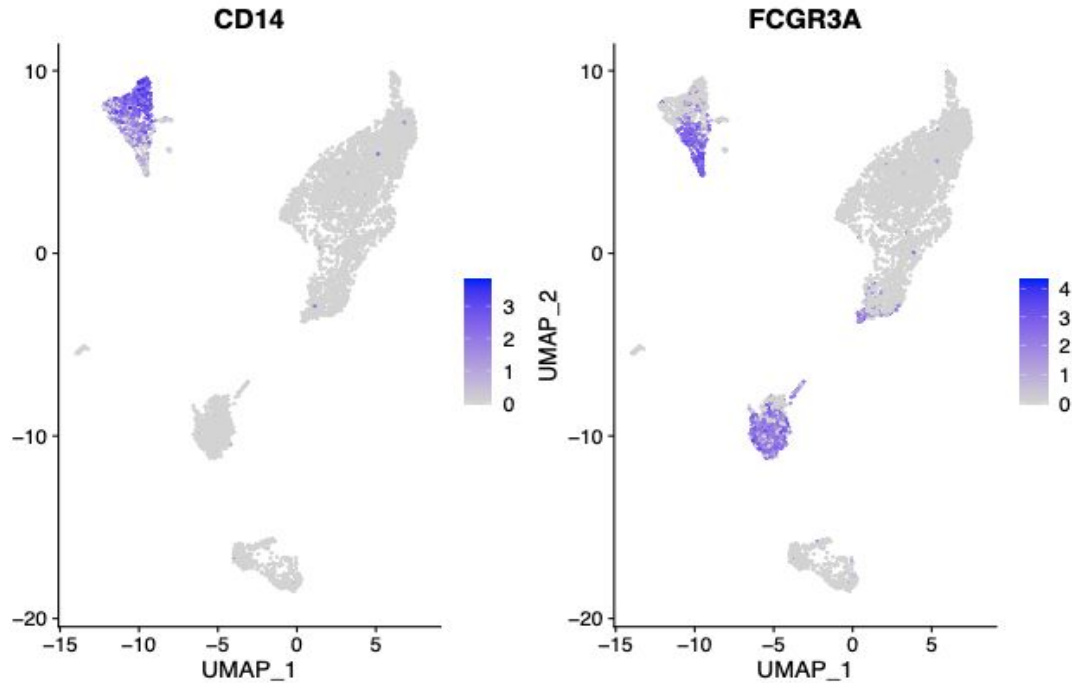
How is the blue monocyte cluster (10) different from green monocyte cluster (4)?



DEG analysis

Note: Ideally, it would be great to have **> 100 cells** in a cluster for proper DEG comparisons because of the signal dropout that happens a lot with single-cell data

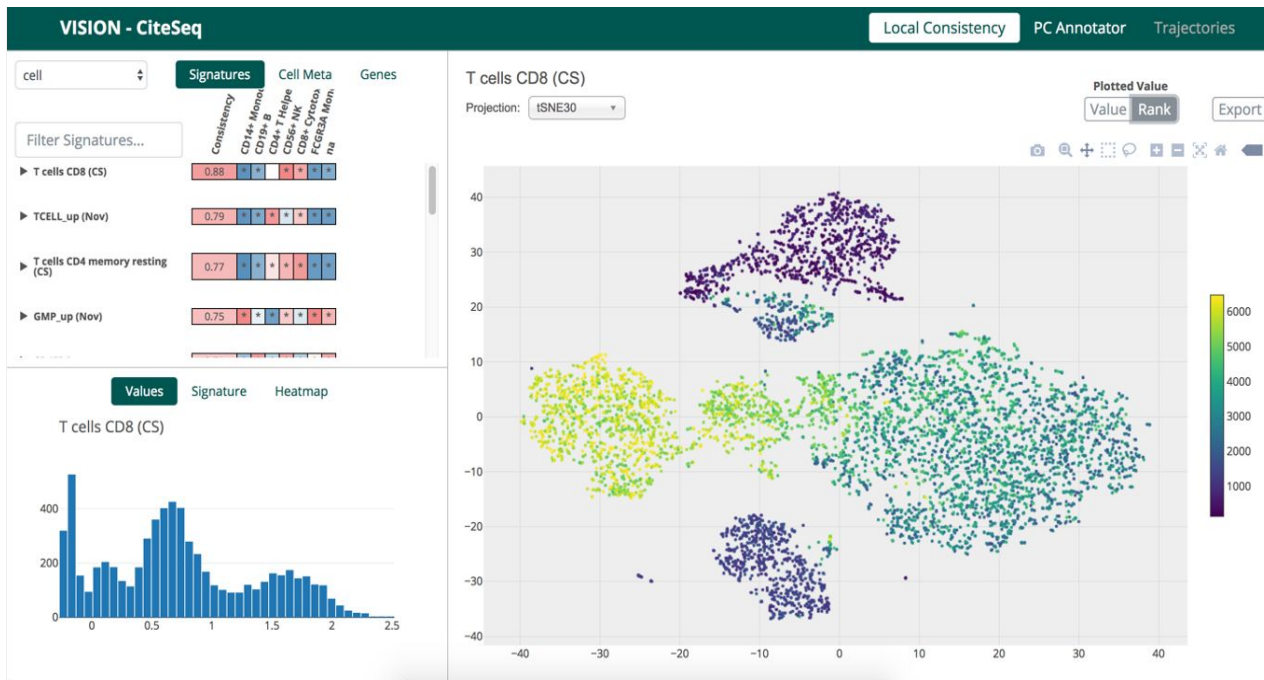
# What if I am interested in CD14+ and CD16+ monocytes?



Look at DEG marker genes to distinguish the two subtypes



# VISION



webapp for exploring gene signatures in scRNA-seq data

Load data from scRNA-seq analysis (precomputed dimensionality reductions, clustering, trajectory inference)

Share easily with collaborators

To see all of what VISION has to offer:

<https://github.com/YosefLab/VISION>  
[ON](#)

(“Tour of the output report user interface (PDF)”)

# 10X Cellranger ATAC Analysis

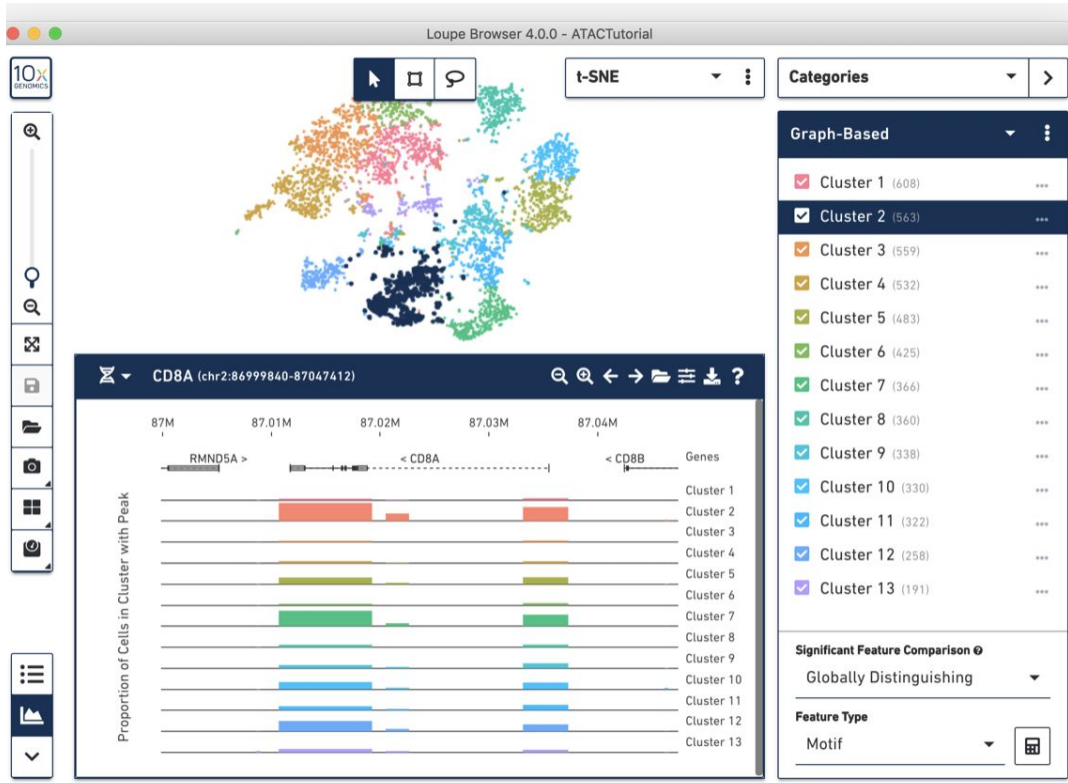
**10X single cell  
ATAC fastqs**

**cellranger-atac count**

(10X software)

- Read filtering and alignment
- Barcode counting
- Identification of transposase cut sites
- Detection of accessible chromatin peaks
- Cell calling
- Count matrix generation for peaks and transcription factors
- Dimensionality reduction
- Cell clustering
- Cluster differential accessibility

# Loupe Browser for Visualization of 10X Cellranger ATAC Data



Interactive desktop application for Windows and MacOS designed for quick visualization of 10X data (either ATAC or RNA)

Browser uses “.loupe” file created by Cellranger run

Can be used to find significant open chromatin regions and transcription factor motifs, identify and compare cell types, and explore substructure

Export tables/screenshots for sharing

Multisample/ multimodal data

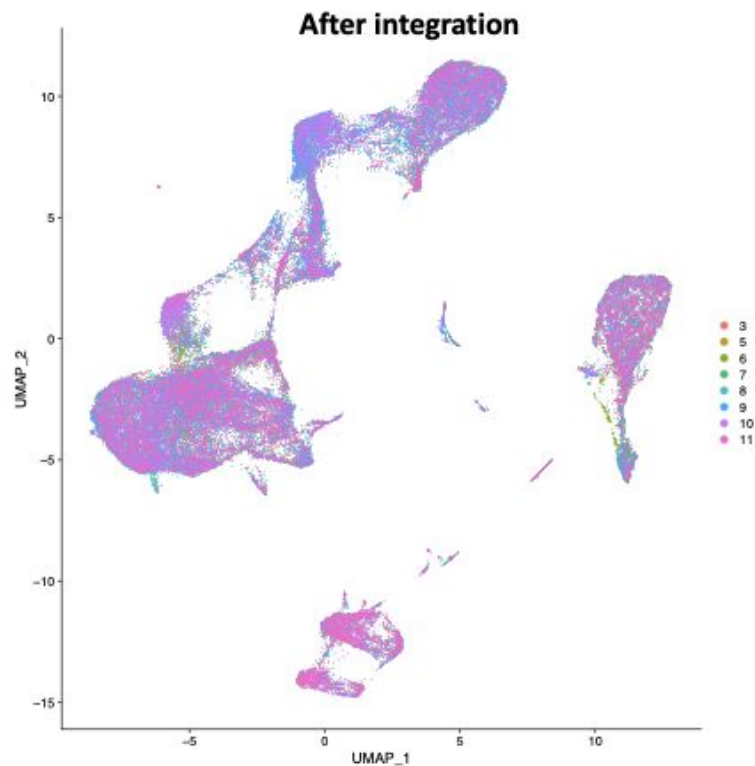
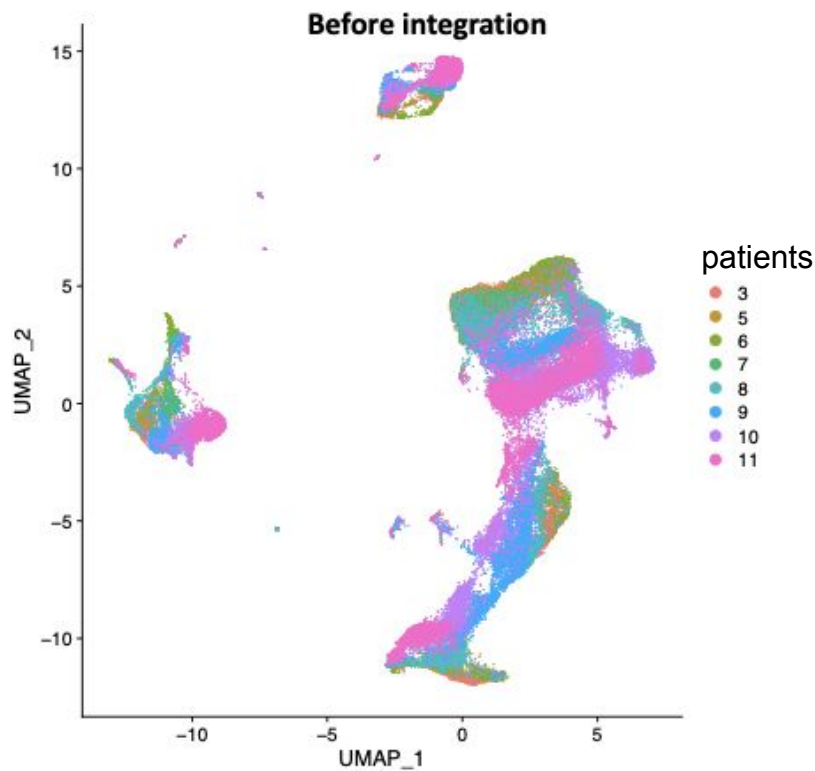
# Single-cell Data Set Integration

- Is a **batch correction** technique you can use to combine and compare data sets from different:
  - Subjects
  - Conditions (Sick vs Healthy, Treated vs Untreated)
  - Technologies (e.g. 10X, Fluidigm, Dropseq, CEL-seq)
  - Functional genomics profiles (e.g. RNA-seq, ATAC-seq, methylation, spatial RNA)
  - Species

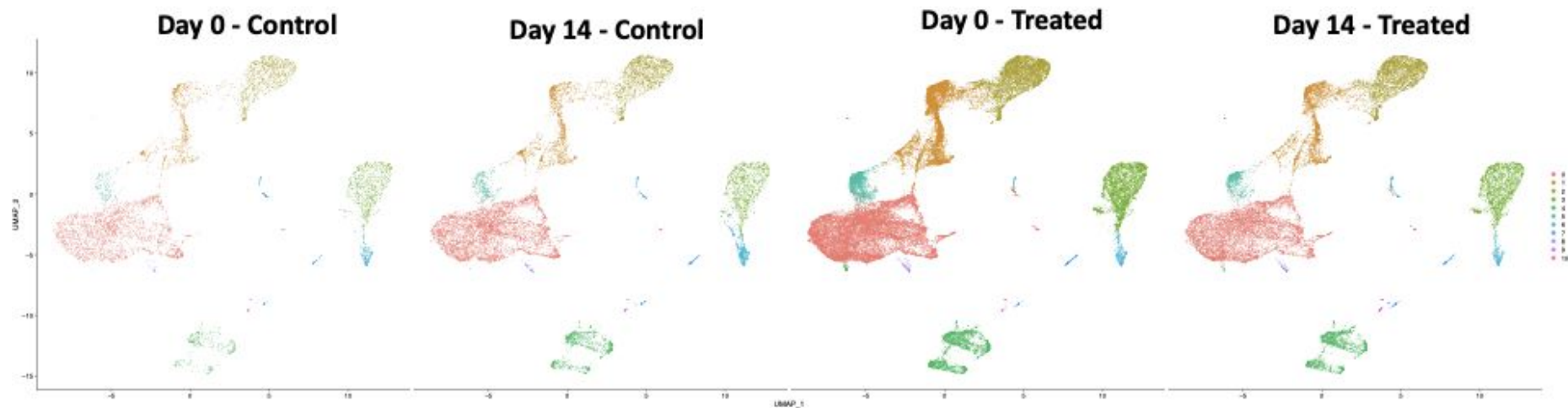
Vignette:

[https://satijalab.org/seurat/articles/integration\\_introduction.html](https://satijalab.org/seurat/articles/integration_introduction.html)

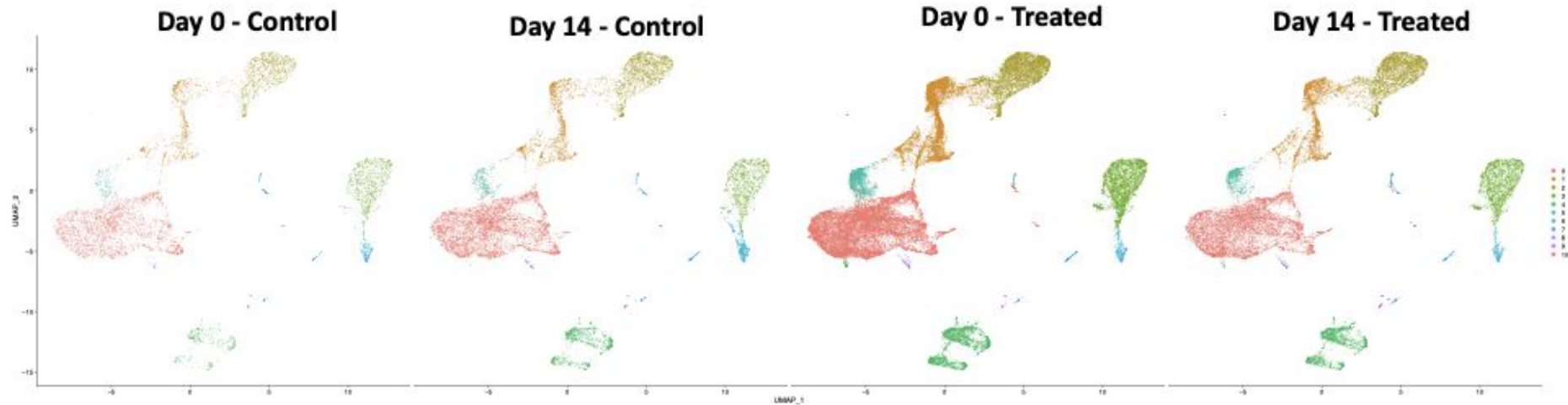
# Example 1: Integration of single-cell RNA-seq samples from multiple 10X Captures



# Can perform DEG analysis on integrated data now



# Can perform DEG analysis on integrated data now



DEGs in control patients over time

DEGs in in treated patients over time



Instead of integrating data, it would be better to design the experiment to use CITE-seq!

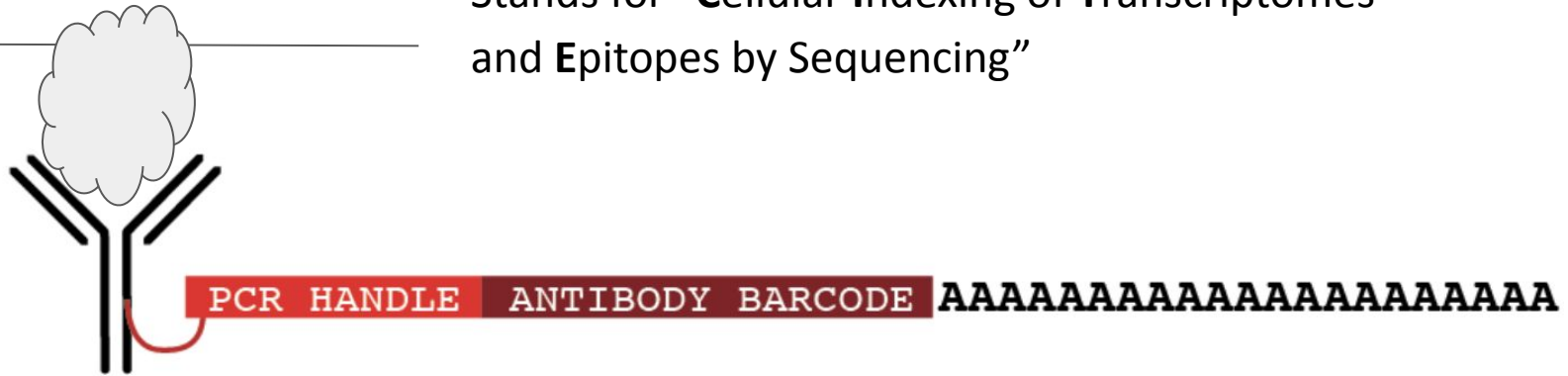
Stands for “**C**ellular **I**ndexing of **T**ranscriptomes and **E**pitopes by **S**equencing”



**Method that allows you to perform RNA-seq + quantify surface protein marker**

Instead of integrating data, it would be better to design the experiment to use CITE-seq!

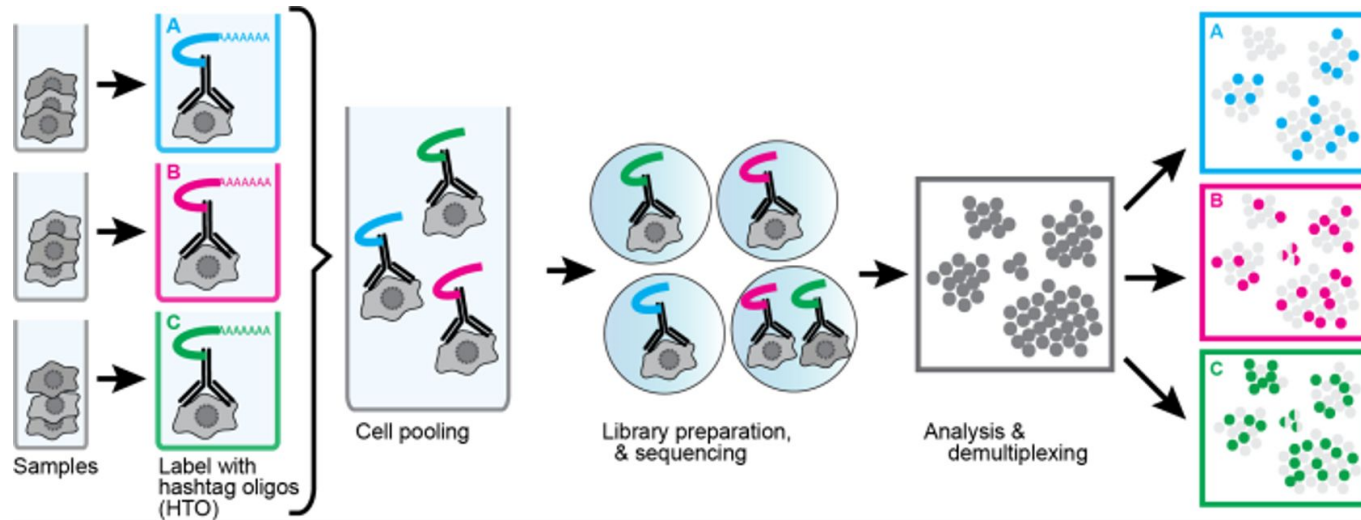
Stands for “**C**ellular **I**ndexing of **T**ranscriptomes and **E**pitopes by **S**equencing”



Method that allows you to perform RNA-seq + quantify surface protein marker

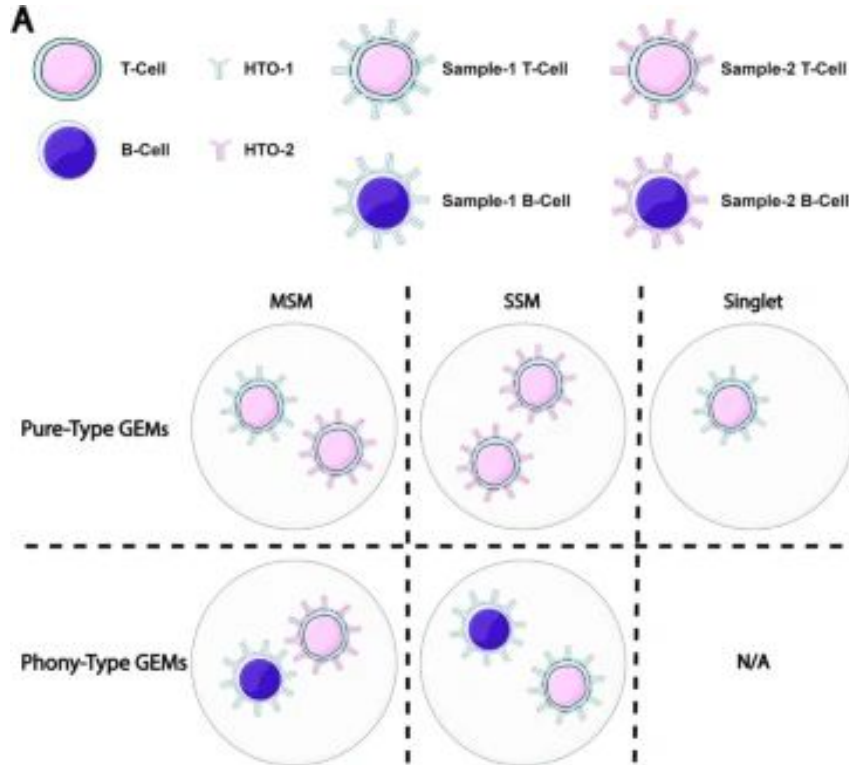
# CITE-seq - “cell hashing” (HTO library)

A way to label cells from different samples, so they can be pooled, prepped, and sequenced together.



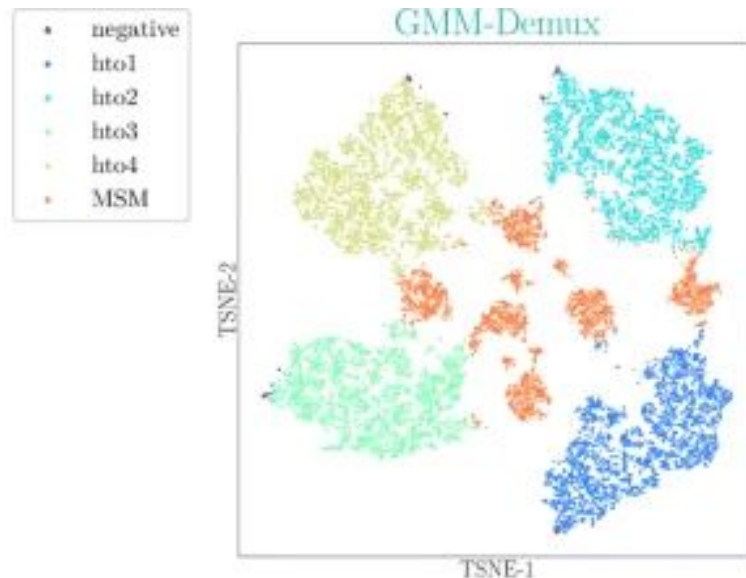
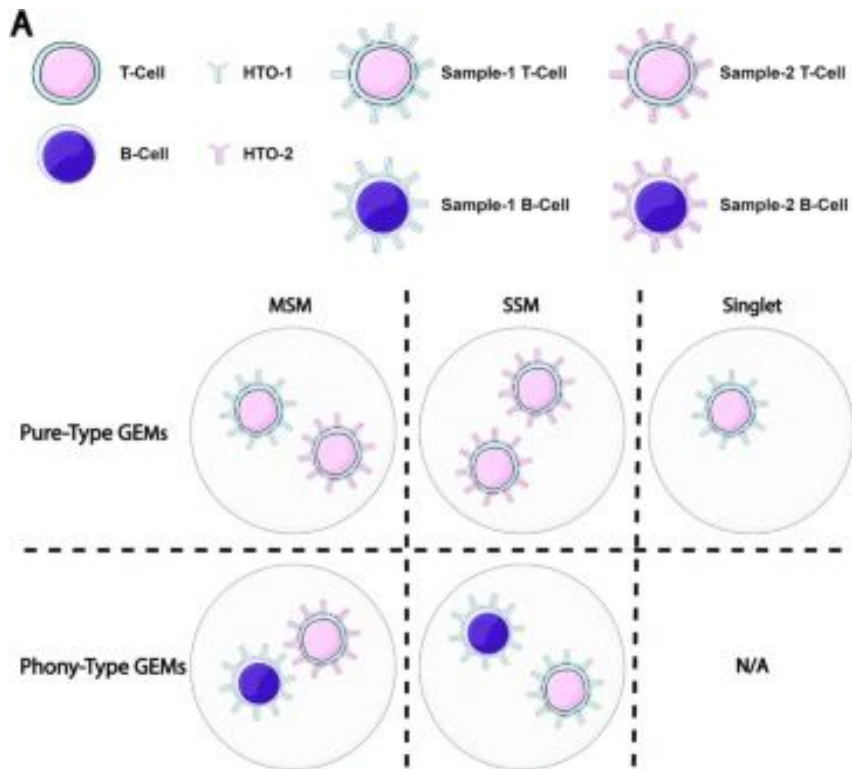
# Multiplet detection

- Multiplets become an issue in super-loaded 10X runs (> 20k cells)
- Tools like GMM Demux can identify multiplets in single-cell CITE-seq (“cell hashing”) data for removal



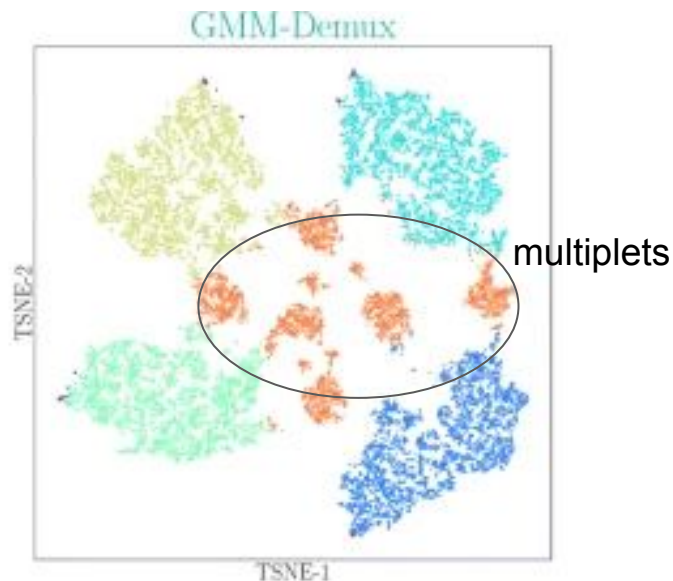
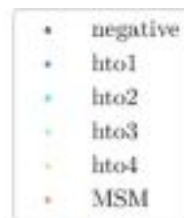
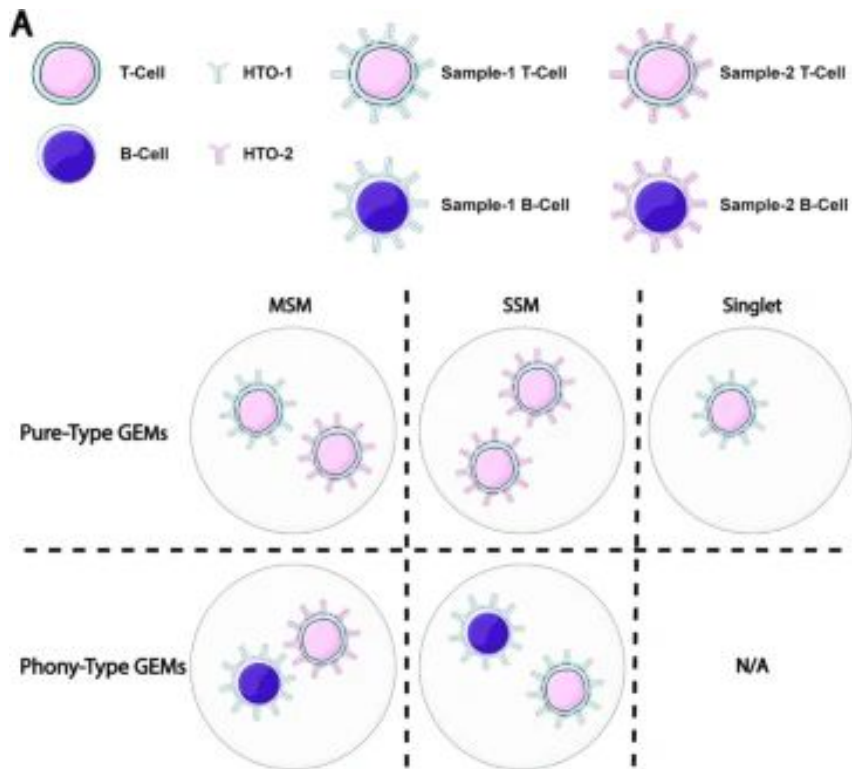
# Multiplet detection

- Multiplets become an issue in super-loaded 10X runs (> 20k cells)
- Tools like GMM Demux can identify multiplets in single-cell CITE-seq (“cell hashing”) data for removal

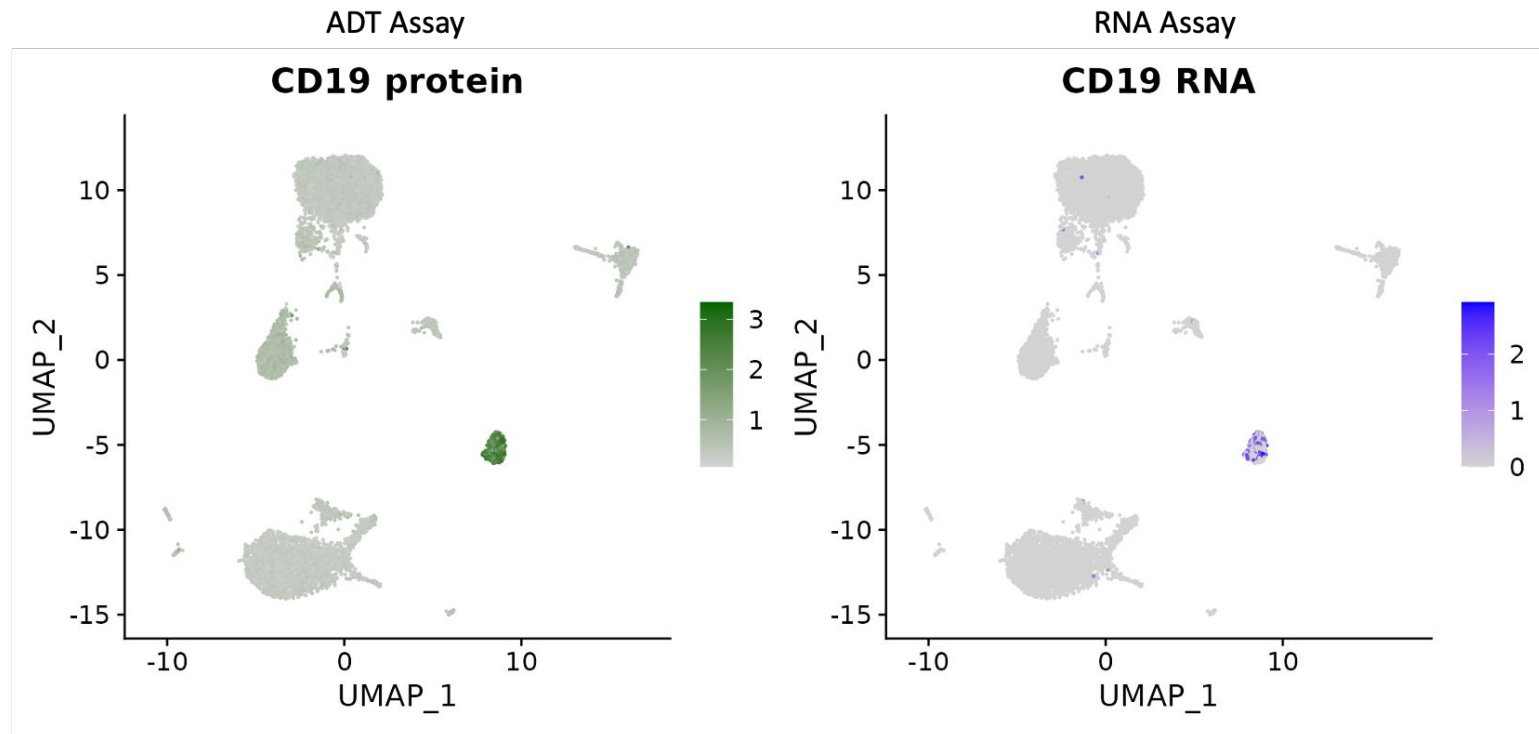


# Multiplet detection

- Multiplets become an issue in super-loaded 10X runs (> 20k cells)
- Tools like GMM Demux can identify multiplets in single-cell CITE-seq (“cell hashing”) data for removal



# Example of Data Set Tagged with CD19 (ADT library)



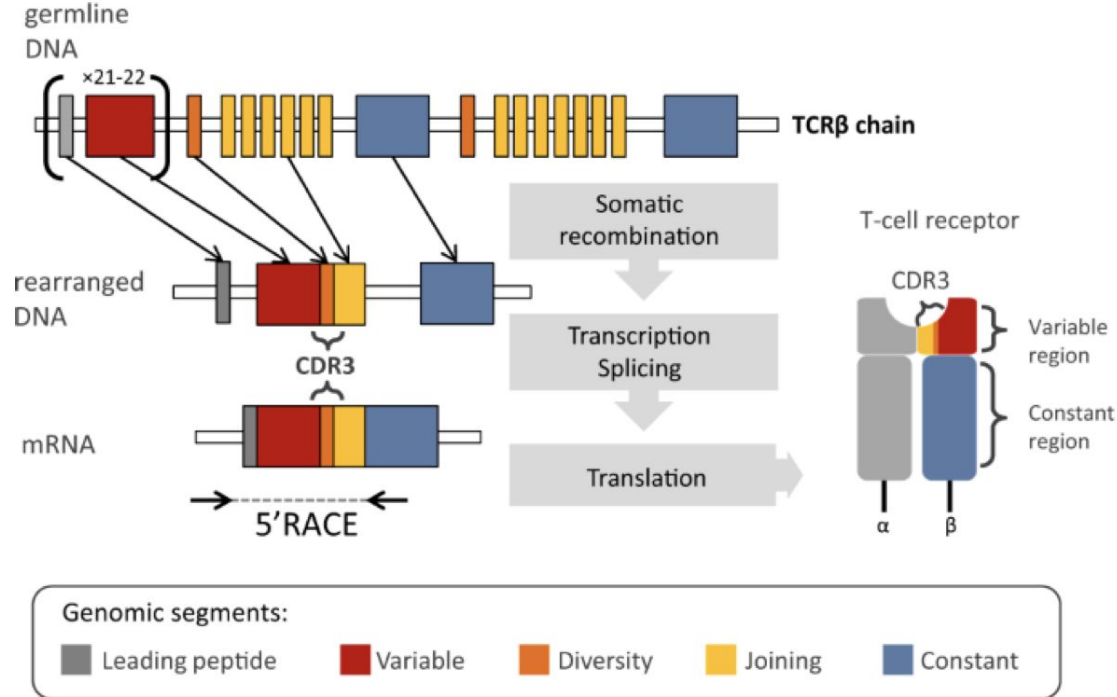
Software for specific use cases



# Variable-Diversity-Joining (VDJ) Analysis

Use case: Immune cell (T-cell and B-cells) profiling!

# Variable-Diversity-Joining (VDJ) Analysis



- T-cell and B-cells undergo recombination of their somatic genomes at T-cell and B-cell receptor loci
- These regions can be sequenced to determine which genome segments were recombined to produce the final receptor chains → “immune profiling”

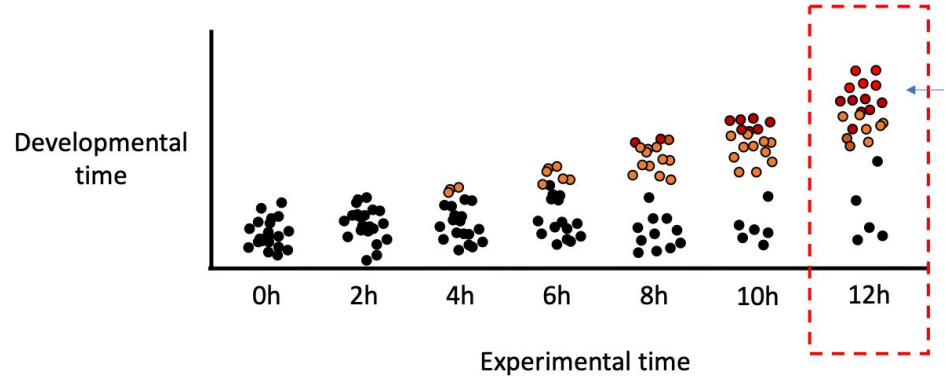
# VDJ analysis with 10X Genomics “cellranger vdj”

Example output:

	A	B	C	D
1	clonotype_id	frequency	proportion	cdr3s_aa
2	clonotype1	182	0.015041322	TRB:CASSYTGNEQYF;TRA:CAMVGSAGNKLTF
3	clonotype2	31	0.002561983	TRB:CASPWDRYNSPLYF;TRB:CASSDEGGQNTLYF;TRA:CATDENNTGKLTF
4	clonotype3	29	0.002396694	TRB:CASSGQGAGEQYF;TRA:CAIVAPGGSNAKLTF
5	clonotype4	29	0.002396694	TRB:CASSLRQSSYEQYF;TRA:CALRWDAGAKLTF
6	clonotype5	24	0.001983471	TRB:CASSLGYNNSPLYF;TRA:CAAASSGSWQLIF
7	clonotype6	20	0.001652893	TRB:CASSGTAETLYF;TRA:CALSEG TNAYKVIF
8	clonotype7	20	0.001652893	TRB:CASGETLYF;TRA:CAAEANQGGRALIF
9	clonotype8	19	0.001570248	TRB:CTCSADSSSQNTLYF;TRA:CAVRNQGGRALIF
10	clonotype9	17	0.001404959	TRB:CASSLGLGGQEYF;TRA:CAIERTNAYKVIF;TRA:CAVRTGFASALTF
11	clonotype10	17	0.001404959	TRB:CASSIKGSGNTLYF;TRA:CAAVRTGGNNKLTF
12	clonotype11	15	0.001239669	TRB:CASSLGLGYEQYF;TRA:CALSTEGADRLTF

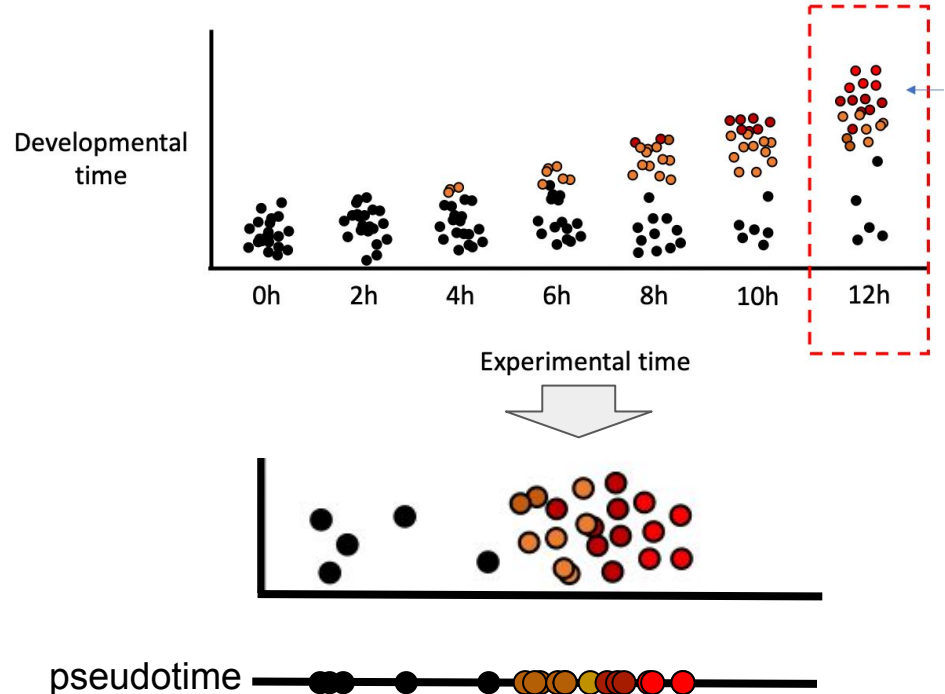
# Trajectory Inference Analysis (aka Pseudotime)

- Method for extracting temporal information from a (static) single-cell sample by ordering cells along a uni-dimensional trajectory
- Type of data sets that they can be used on:
  - Cell differentiation
  - Embryonic development
  - Cell Treatment
  - Cell polarization

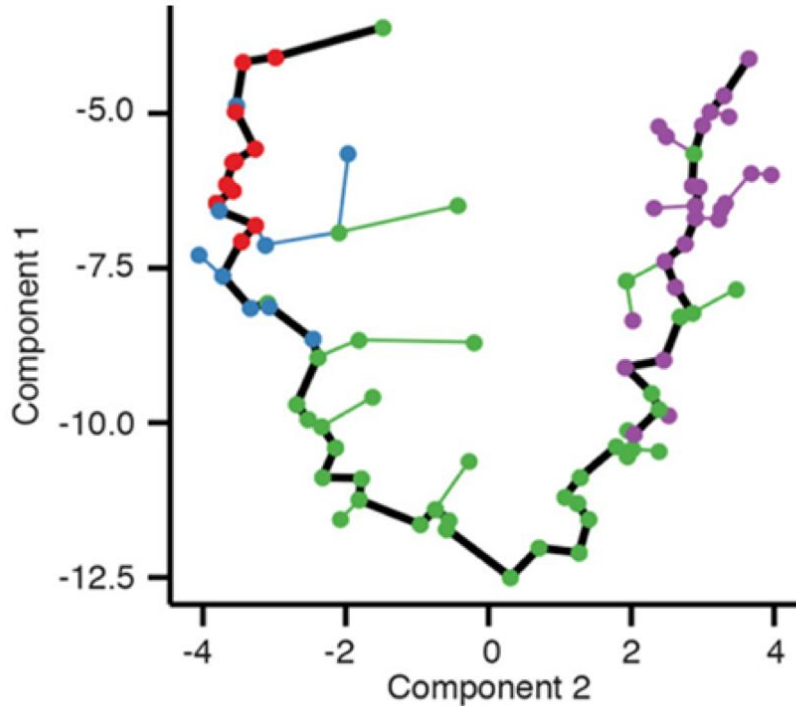


# Trajectory Inference Analysis (aka Pseudotime)

- Method for extracting temporal information from a (static) single-cell sample by ordering cells along a uni-dimensional trajectory
- Type of data sets that they can be used on:
  - Cell differentiation
  - Embryonic development
  - Cell Treatment
  - Cell polarization

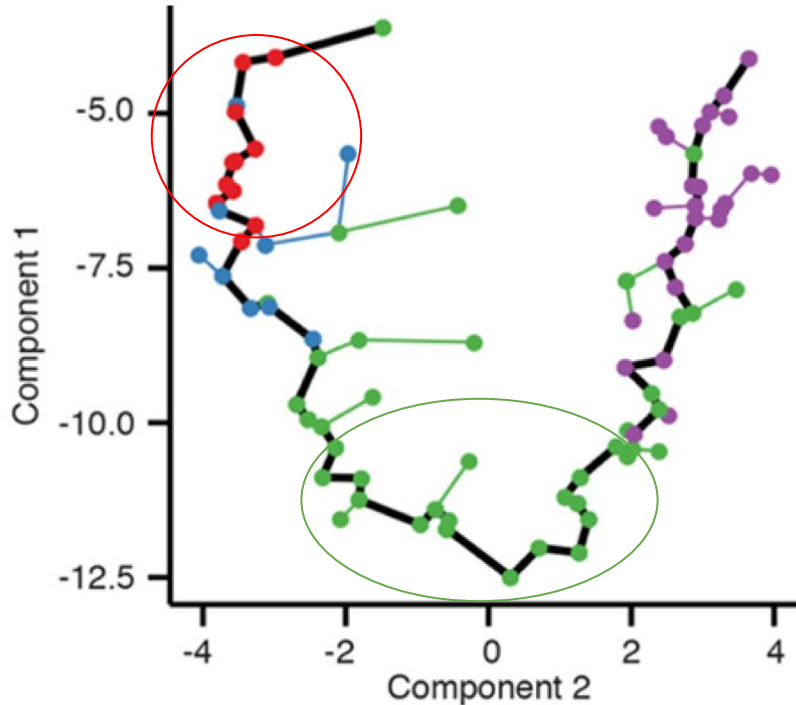


# Trajectory Inference Analysis (aka Pseudotime)



- Software tools will apply pseudotime trajectory (**thick line**) over existing visualizations

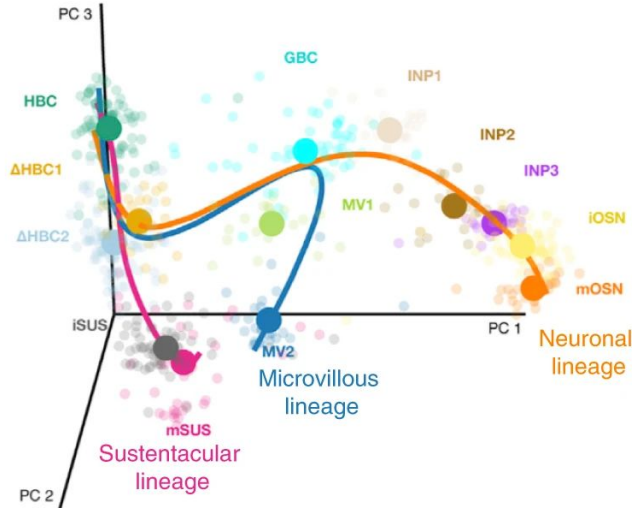
# Trajectory Inference Analysis (aka Pseudotime)



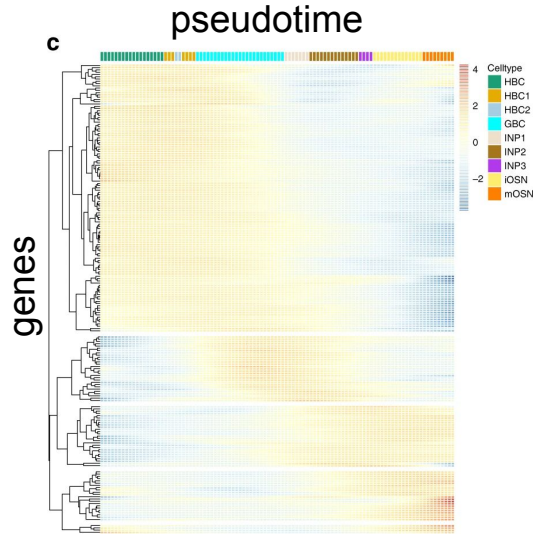
- Software tools will apply pseudotime trajectory (**thick line**) over existing visualizations
- Can compare cell clusters along the trajectory to determine genes that change over pseudotime

# Monocle 3 Pseudotime Analysis Example

a



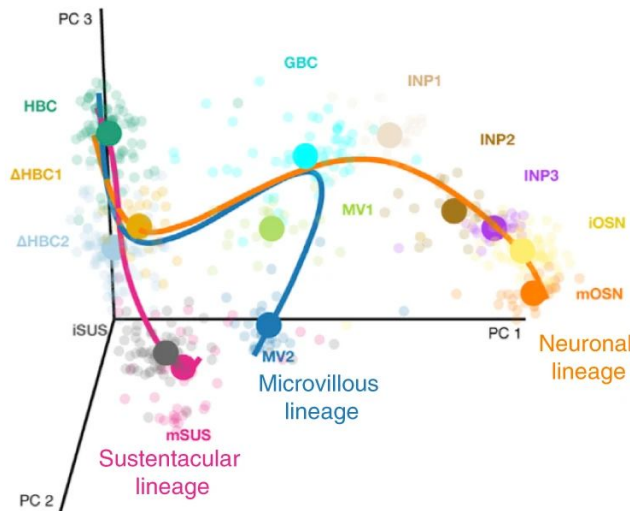
Mouse olfactory epithelium  
(Software: Monocle 3)





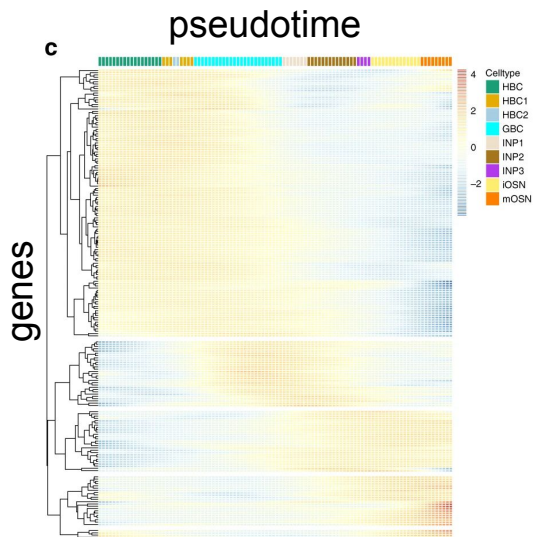
# Monocle 3 Pseudotime Analysis Example

**a**

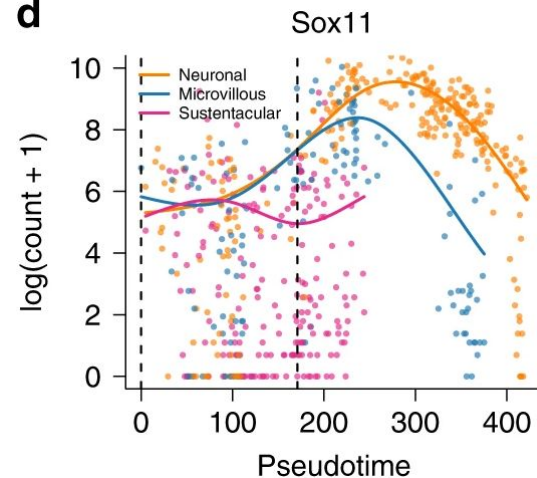


Mouse olfactory epithelium  
(Software: Monocle 3)

**c**



**d**



# Pseudotime Software

Some software tools can handle cyclical and disconnected trajectories

Method	Inferable trajectory types										Aggregated scores per experiment					
	Priors required	Wrapper type	Platform	Topology inference	Cycle	Linear	Bifurcation	Multifurcation	Tree	Connected	Disconnected	Overall	Accuracy	Scalability	Stability	Usability
<b>Graph methods</b>																
PAGA	x	Direct	Python	Free	△	→	→	→	→	→	→	→	→	→	→	→
RaceID / StemID		Proj	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
SLICER	x	Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
<b>Tree methods</b>																
Slingshot		Direct	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
PAGA Tree	x	Direct	Python	Free	△	→	→	→	→	→	→	→	→	→	→	→
MST		Proj	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
pCreode		Proj	Python	Free	△	→	→	→	→	→	→	→	→	→	→	→
SCUBA		Cluster	Python	Free	△	→	→	→	→	→	→	→	→	→	→	→
Monocle DDRTree		Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
Monocle ICA	x	Cell	R	Param	△	→	→	→	→	→	→	→	→	→	→	→
cellTree maptpx		Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
SLICE		Direct	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
cellTree VEM		Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
EIPiGraph		Direct	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
Sincell		Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
URD	x	Direct	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
CellTrails		Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
Mpath	x	Cluster	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
CellRouter	x	Cell	R	Free	△	→	→	→	→	→	→	→	→	→	→	→
<b>Multifurcation methods</b>																
STEMNET	x	Prob	R	Param	△	→	→	→	→	→	→	→	→	→	→	→
FateID	x	Prob	R	Param	△	→	→	→	→	→	→	→	→	→	→	→
MFA	x	Prob	R	Param	△	→	→	→	→	→	→	→	→	→	→	→
GPfates	x	Prob	Python	Param	△	→	→	→	→	→	→	→	→	→	→	→
<b>Bifurcation methods</b>																
DPT		Direct	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
Washbone	x	Direct	Python	Param	△	→	→	→	→	→	→	→	→	→	→	→
<b>Linear methods</b>																
SCORPIUS		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
Component 1		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
Embeddr		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
MATCHER		Linear	Python	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
TSCAN		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
Wanderlust	x	Linear	Python	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
PhenoPath		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
topslam	x	Linear	Python	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
Waterfall		Linear	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
EIPiGraph linear		Direct	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
ouijaflow		Linear	Python	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
FORKS		Linear	Python	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
<b>Cyclic methods</b>																
Angle		Cycle	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
EIPiGraph cycle		Direct	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→
reCAT		Cycle	R	Fixed	△	→	→	→	→	→	→	→	→	→	→	→

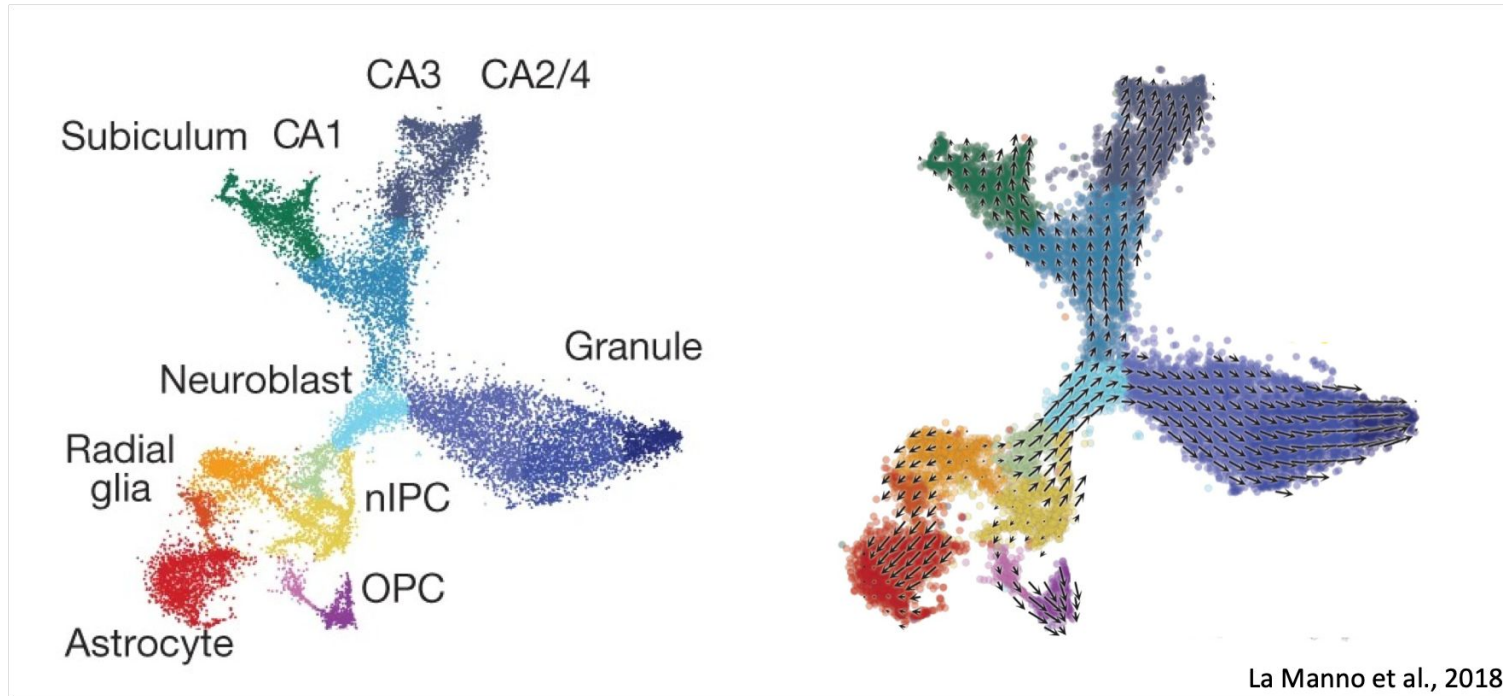
Prior information required: None, x Weak: Start or end cells, x Strong: Cell grouping or time course  
 Not shown, insufficient data points: ouija, cellTree Gibbs, pseudogp, GrandPrix, SCIMITAR, MERLoT, SCOUF

# RNA Velocity

- Uses transcript splicing rates of genes as a way to predict a cell's fate (on a short time scale)

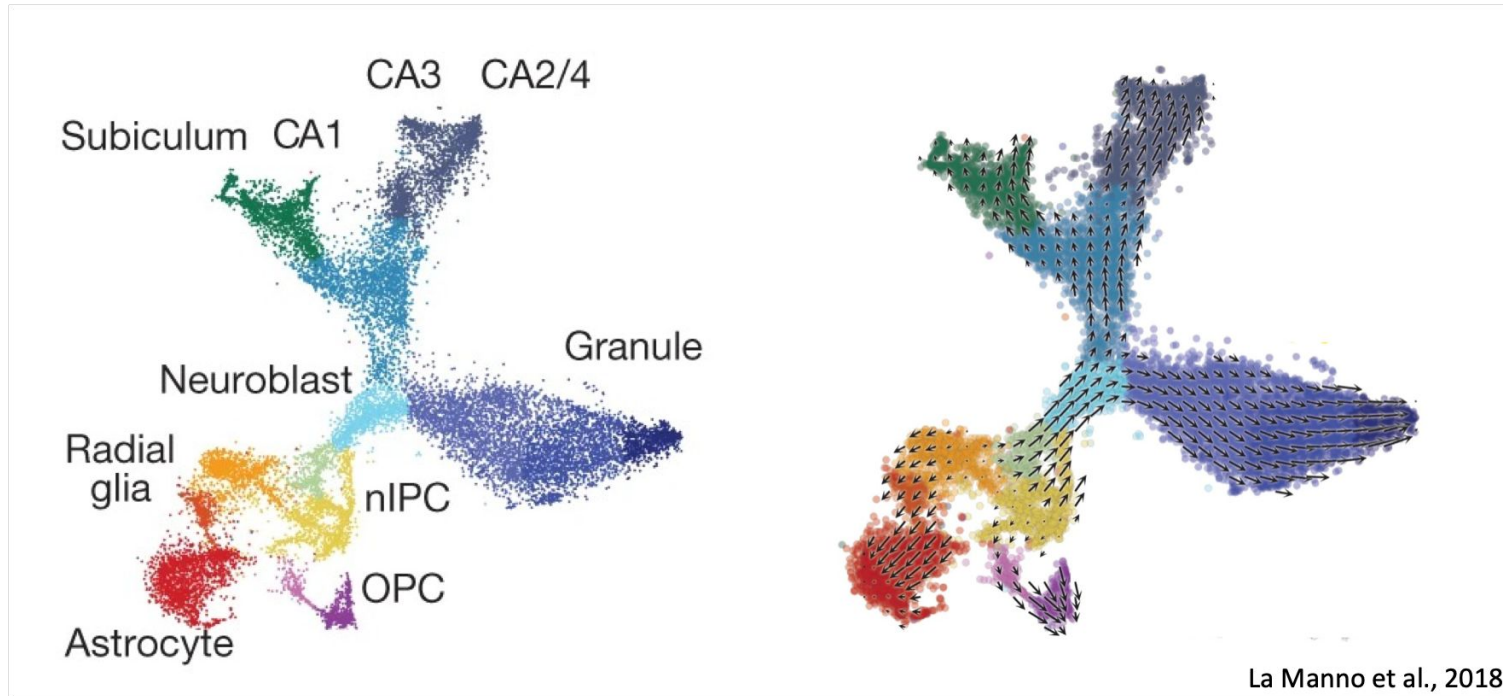
# RNA Velocity

- Uses transcript splicing rates of genes as a way to predict a cell's fate (on a short time scale)

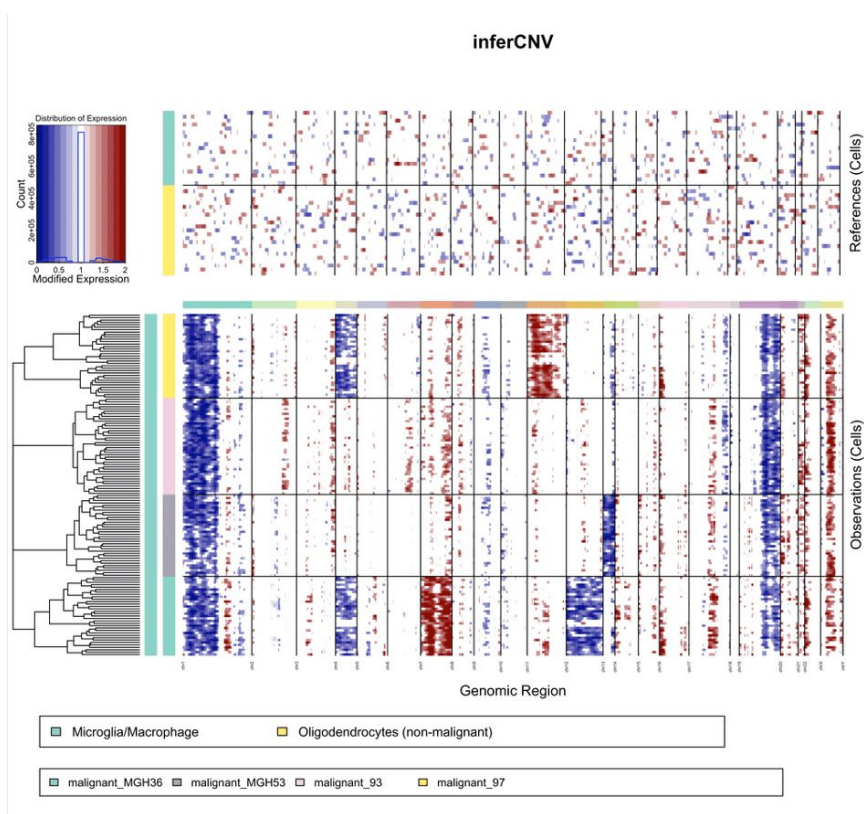


# RNA Velocity

Use case: to determine if a gene of interest is being induced or repressed in a cell population of interest



# Copy Number Detection in Single-cell RNA-seq Data



Infers copy number variations by exploring gene expression intensities across positions of the genome in “abnormal” cells and comparing them to “normal” cells

Use case: tumor single-cell RNA-seq sample

Tools: InferCNV, Casper, Copykat

# Supplemental Slides



# Differentially Expressed Gene Analysis

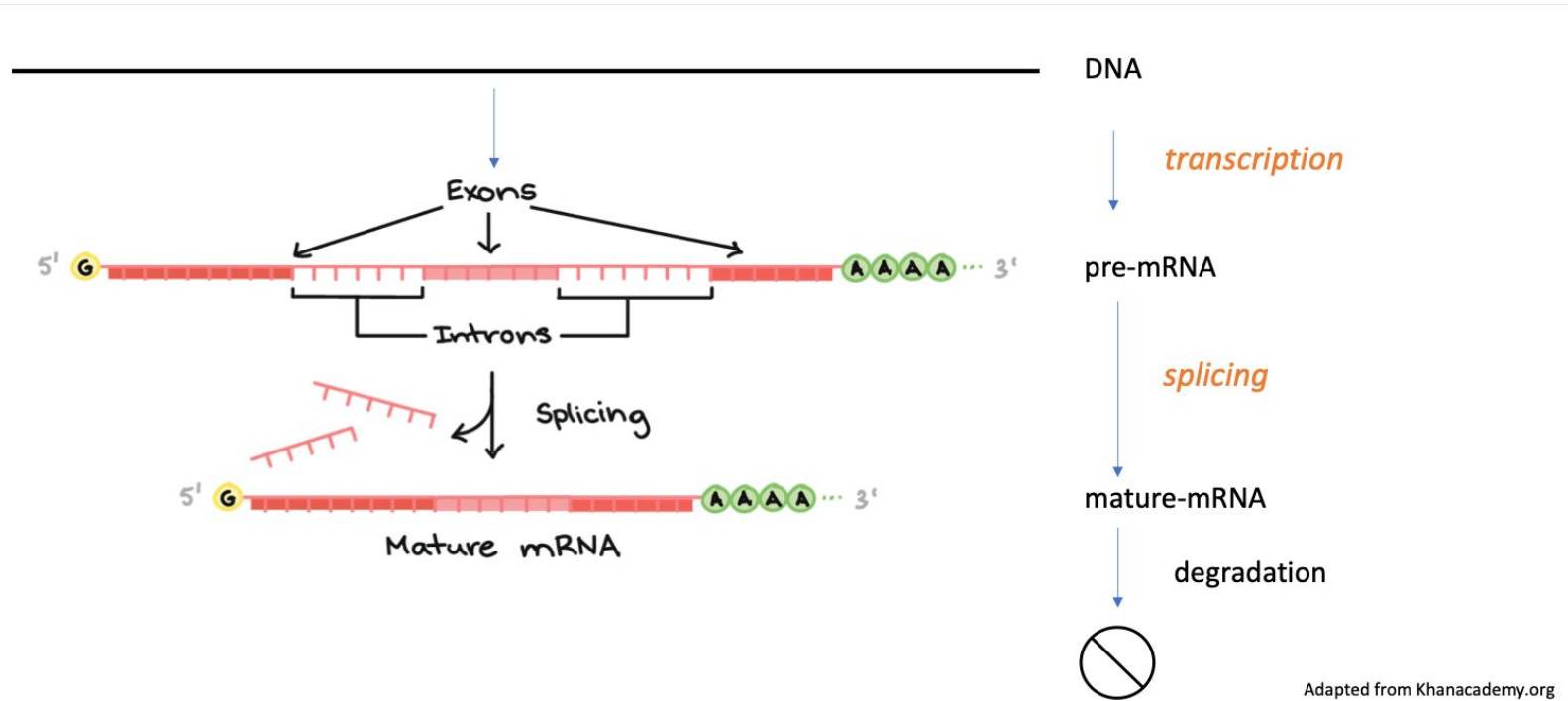
Ideally, it would be great to have **> 100 cells** in a cluster for proper DEG comparisons because of the signal dropout that happens a lot with single-cell data

Side note: When designing an experiment, it is good to think about how many cells will be acquired, how many will be lost through filtering, and how many cells of your cell types of interest you expect to have at the end for DEG analysis. If you predict too few, you may want to reconsider your experimental design (maybe perform cell sorting to enrich for populations of interest?)

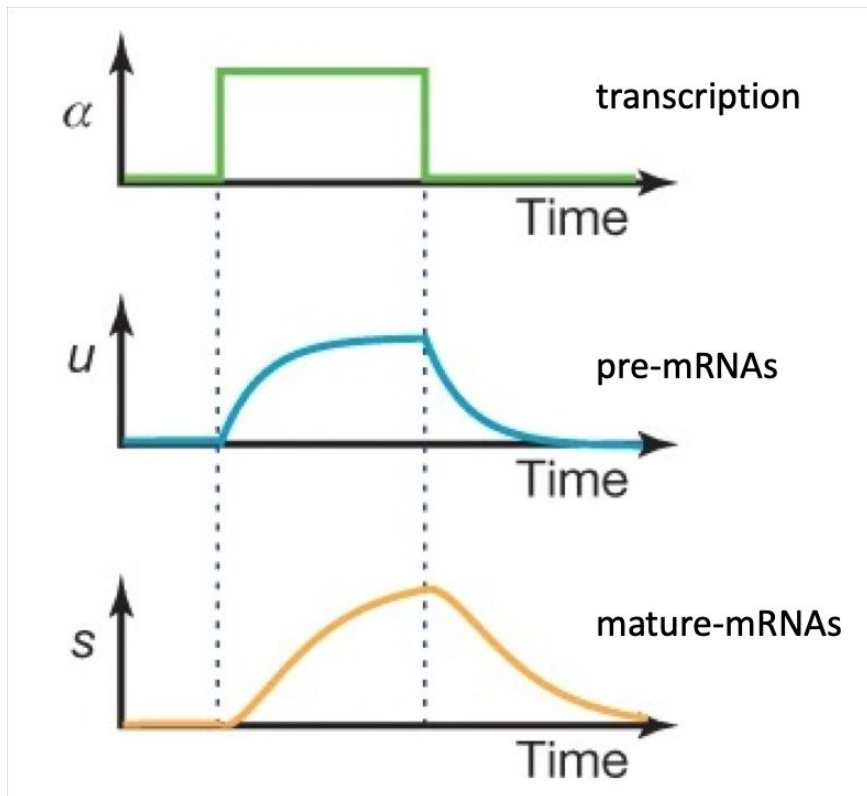


# RNA Velocity

- ~23% of UMIs are from unspliced molecules (Le Manno et al., 2018)



# RNA Velocity Model



transcription (state-dependent)

splicing

$$\frac{du(t)}{dt} = \alpha_k(t) - \beta u(t),$$
$$\frac{ds(t)}{dt} = \beta u(t) - \gamma s(t),$$

RNA velocity

degradation

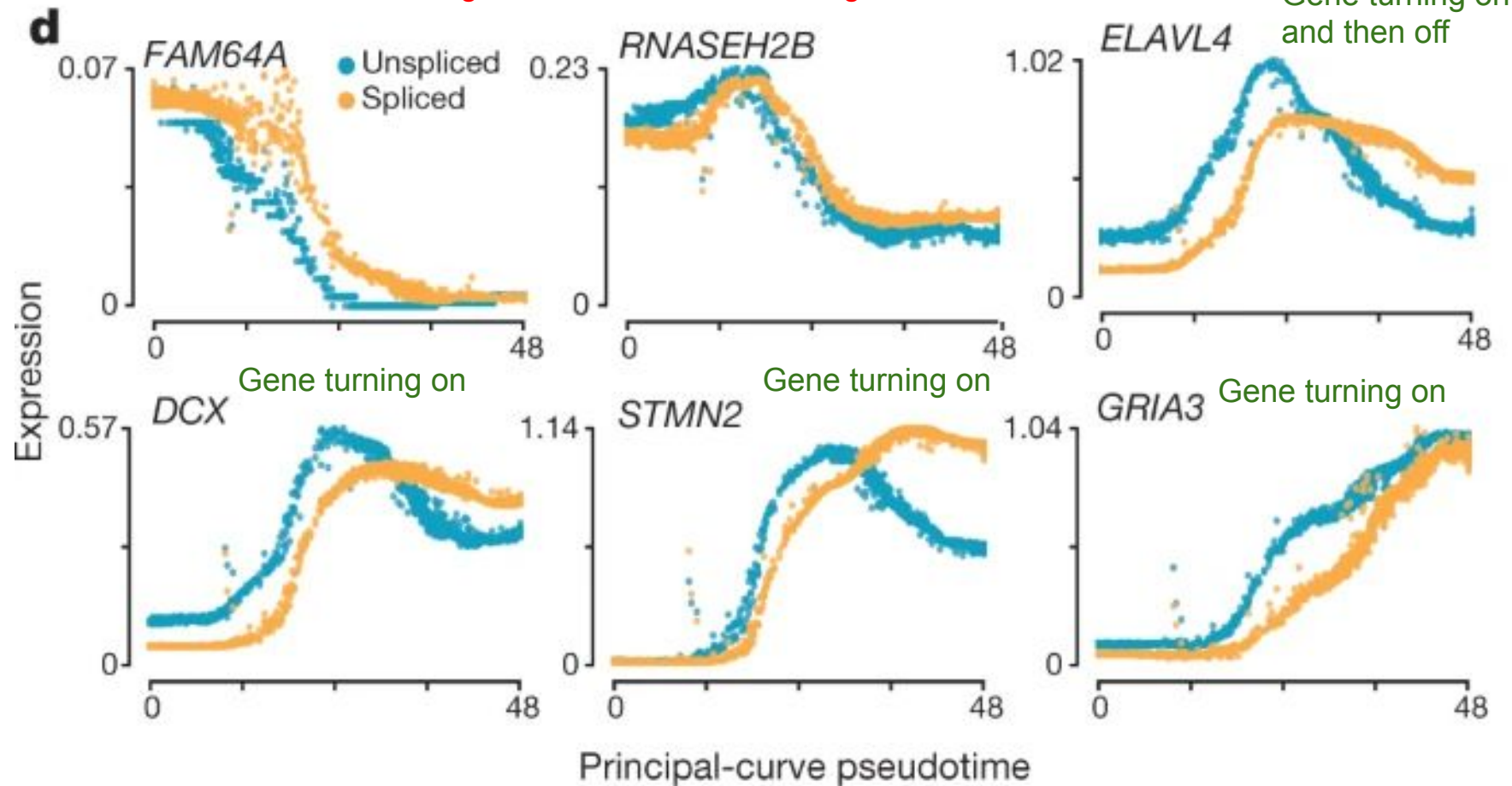
Detailed description: This diagram illustrates the RNA velocity model with two differential equations. The first equation,  $\frac{du(t)}{dt} = \alpha_k(t) - \beta u(t)$ , shows the rate of change of pre-mRNAs ( $u$ ). The term  $\alpha_k(t)$  is labeled "transcription (state-dependent)" and is enclosed in a red box. The term  $\beta u(t)$  is labeled "splicing" and is enclosed in a blue box. The second equation,  $\frac{ds(t)}{dt} = \beta u(t) - \gamma s(t)$ , shows the rate of change of mature-mRNAs ( $s$ ). The term  $\beta u(t)$  is labeled "splicing" and is enclosed in a blue box. The term  $\gamma s(t)$  is labeled "degradation" and is enclosed in a green box. The entire second equation is enclosed in a red box labeled "RNA velocity".

# RNA Velocity

Gene turning off

Gene turning off

Gene turning on and then off



# RNA Velocity Software

## **Velocyto (La Manno et al. 2018)**

- Implemented in R and python
- Uses BAM files generated from Cellranger (10X Genomics)
  - Uses uniquely mapped reads (multimapping and reads in repeat-masked regions are discarded).

Assumes splicing rate = 1 for all genes/all cells

## **scVelo (newer – 2020)**

- Install for Python 3.6 with Pypi
- Input for scVelo:
  - Counts matrix of unspliced pre-mature mRNA abundances
  - Counts matrix of spliced mature mRNA abundance
  - Both can be produced with loompy/kallisto pipeline or velocyte
- Estimates betas (splicing rates) and gammas (degradation rates) for each gene

# Reference-based Cell Type Classification

## SingleR/Cheetah

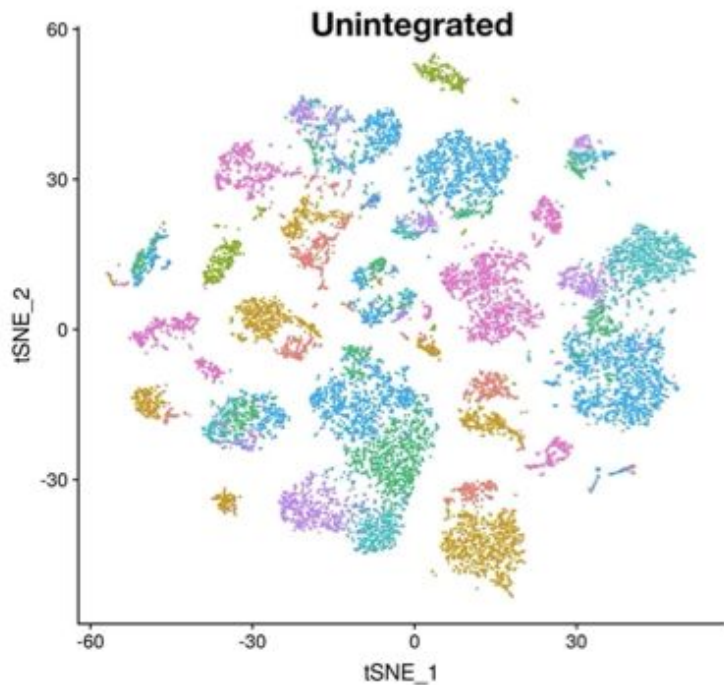
- Performs Spearman rank correlation of individual single cells (or cell clusters) against samples in a reference bulk RNA-seq, single-cell RNA-seq, or microarray database (correlations are done using variable single-cell genes)
- Performs filtering to get the most accurate cell type calls

## IMPORTANT NOTES:

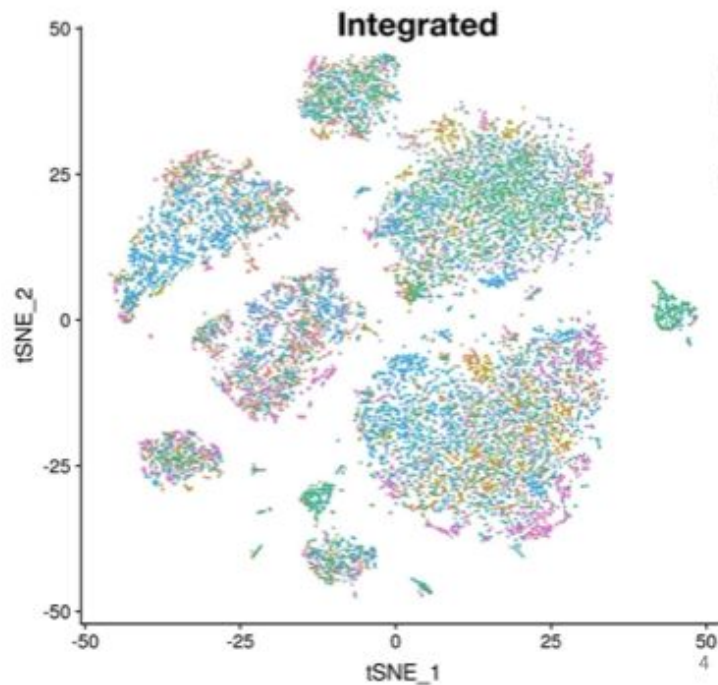
- Be aware of the cell types in the reference data; the reference may not contain all of the cell types in your query dataset
- Some classification tools (e.g. SingleR) **force** cell type annotation labels onto cells. If a cell type in your query data set is not in the reference, it will still be annotated with the best match. So need to be careful not to take the results at face value. Other tools (CHETAH) can annotate with “in-between” cell type labels if it cannot place it

# Integration of Pancreas Data Sets

8 pancreas data sets: CELseq, CELseq2, FluidigmC1, SMARTseq2, InDrop#1, InDrop#2, InDrop#3, InDrop#4



**30+ clusters**

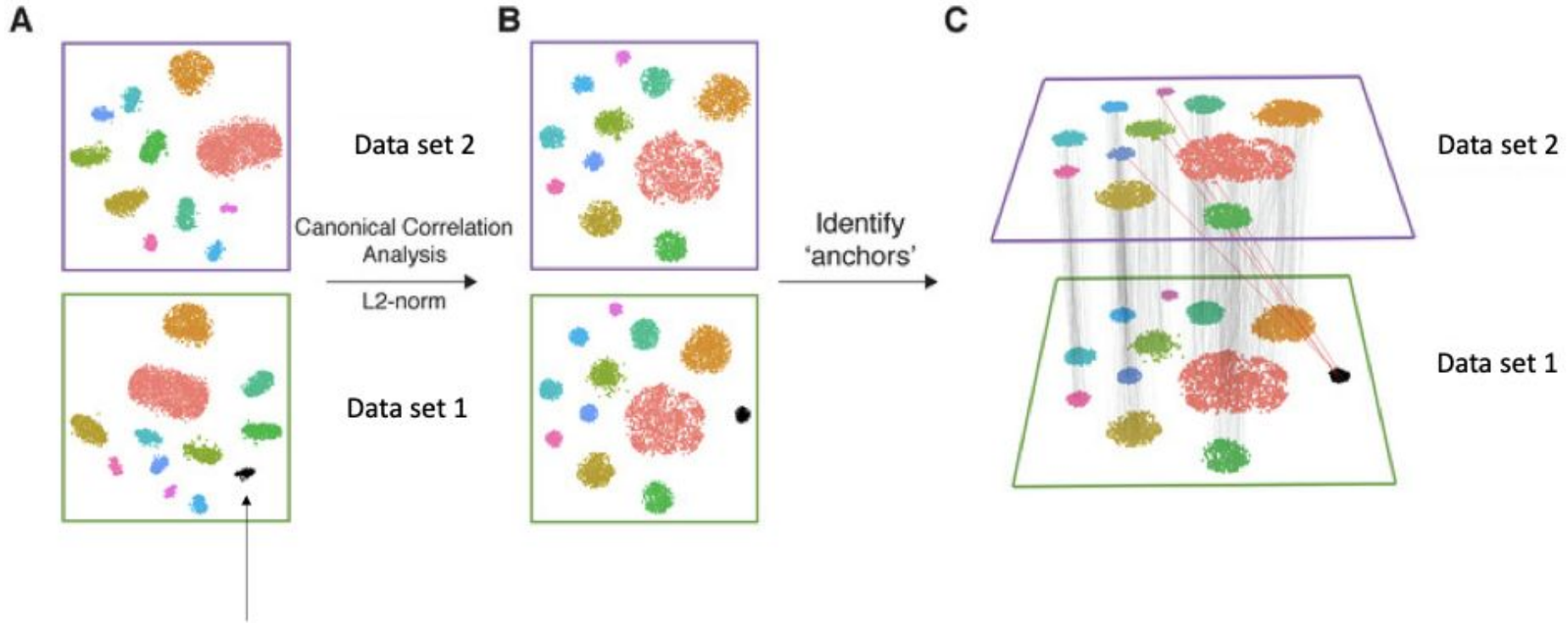


**~9 clusters**

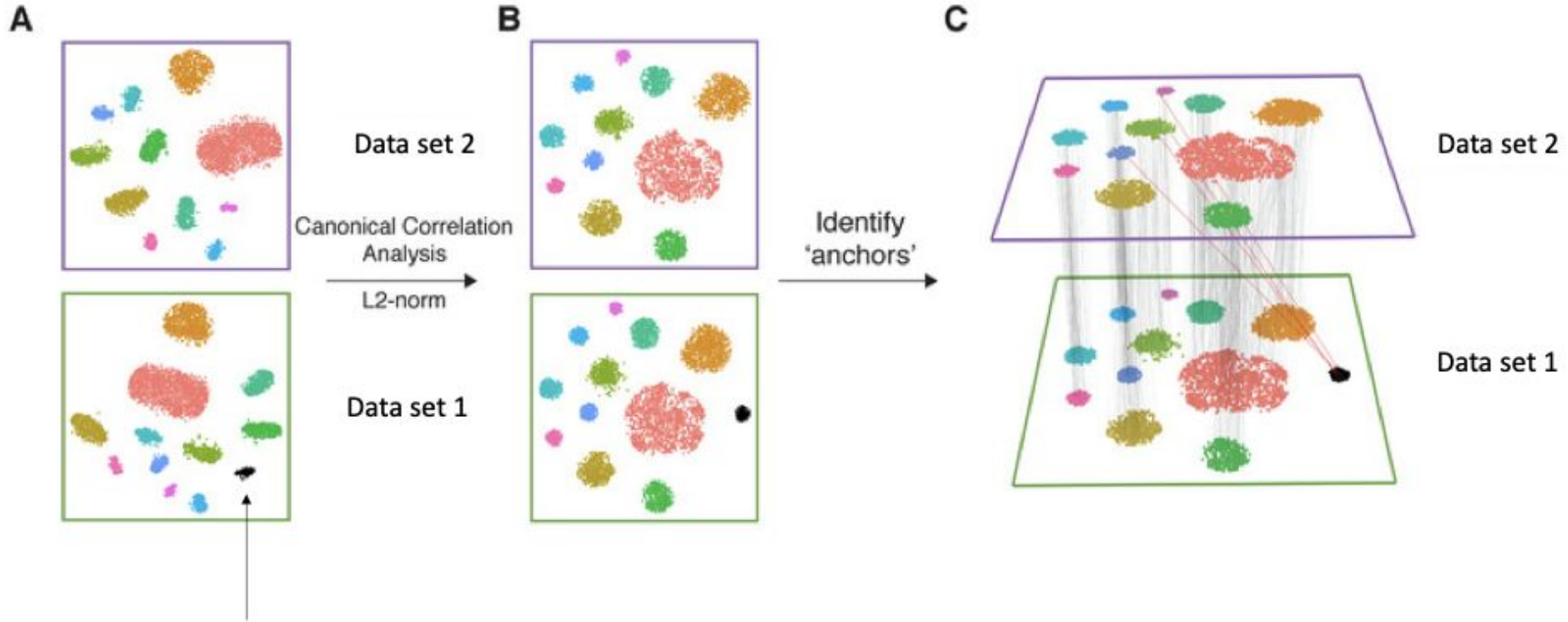
Baron et al. 2016, *Cell Syst.*  
Lawlor et al. 2017, *Genome Res.*  
Grun et al. 2016, *Cell Stem Cell*  
Muraro et al. 2016, *Cell Syst.*

Adapted from Ahmed Mahfuoz

# Single-Cell Data Set Integration (Using Seurat)

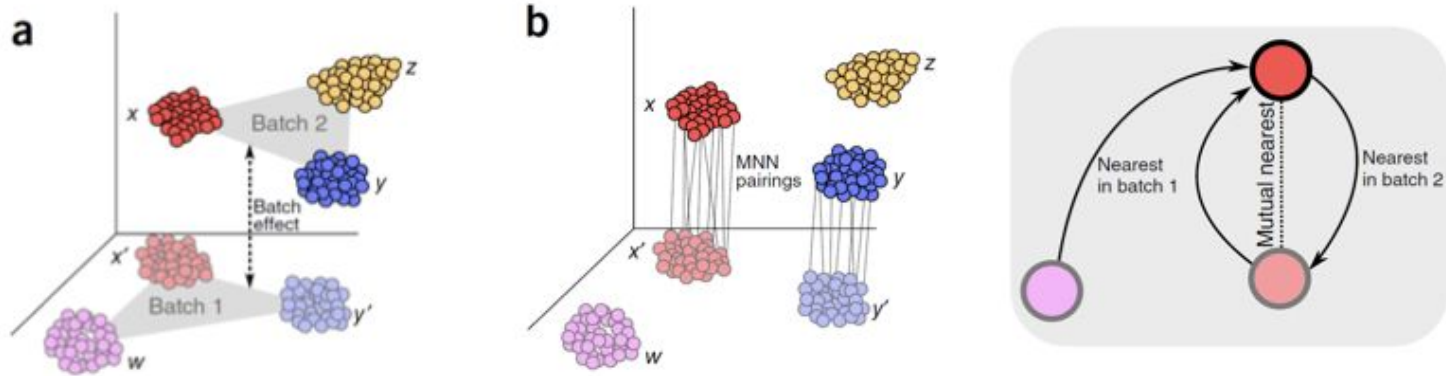


# Single-Cell Data Set Integration (Using Seurat)

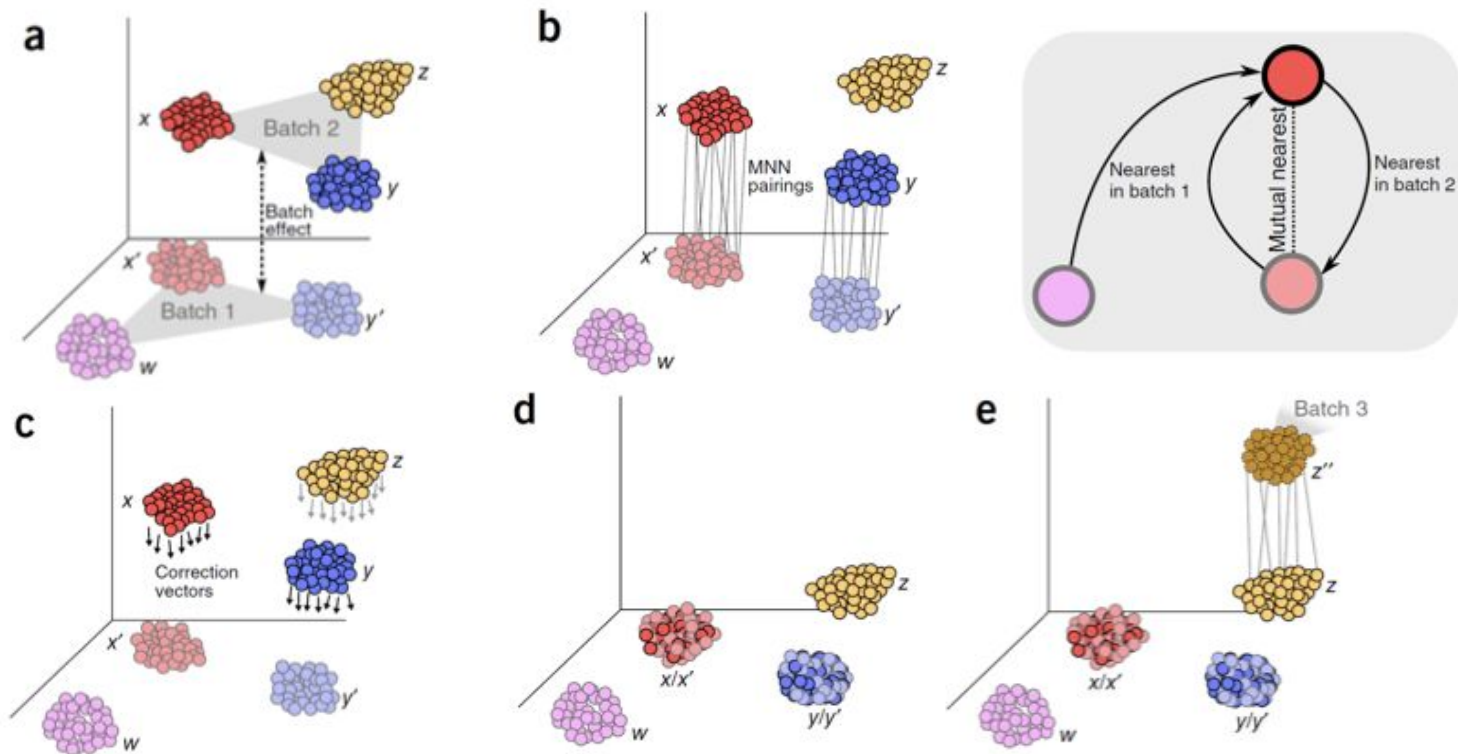




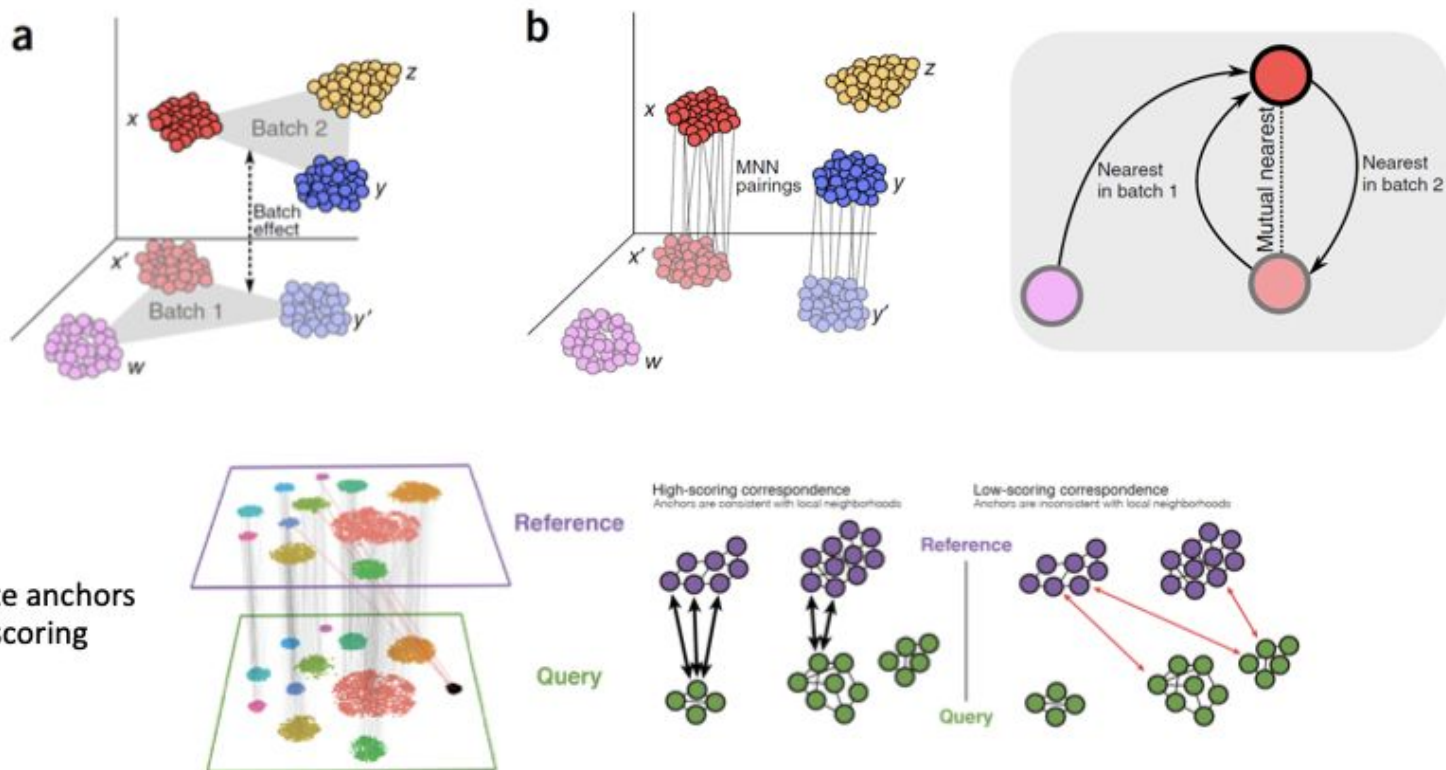
# Single-Cell Data Set Integration (Using Seurat)



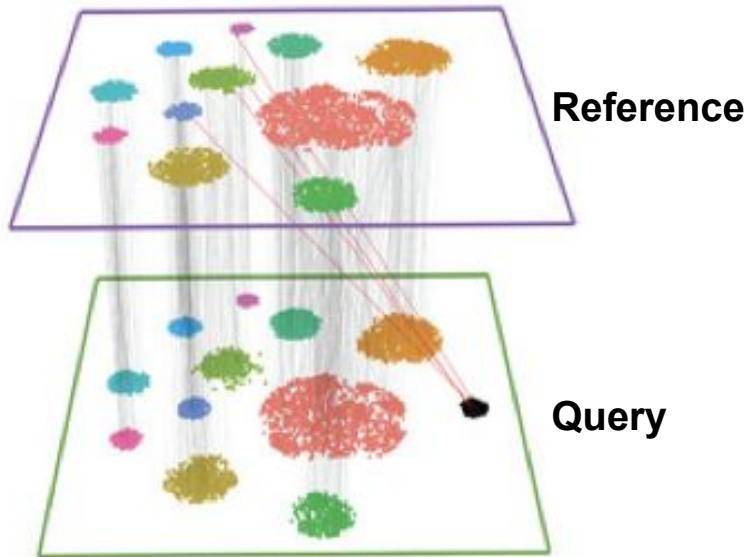
# Single-Cell Data Set Integration (Using Seurat)



# Single-Cell Data Set Integration (Using Seurat)



What if I spent hours delineating cell clusters and their identities in a single sample? Will I have to do that for all of the samples like it?

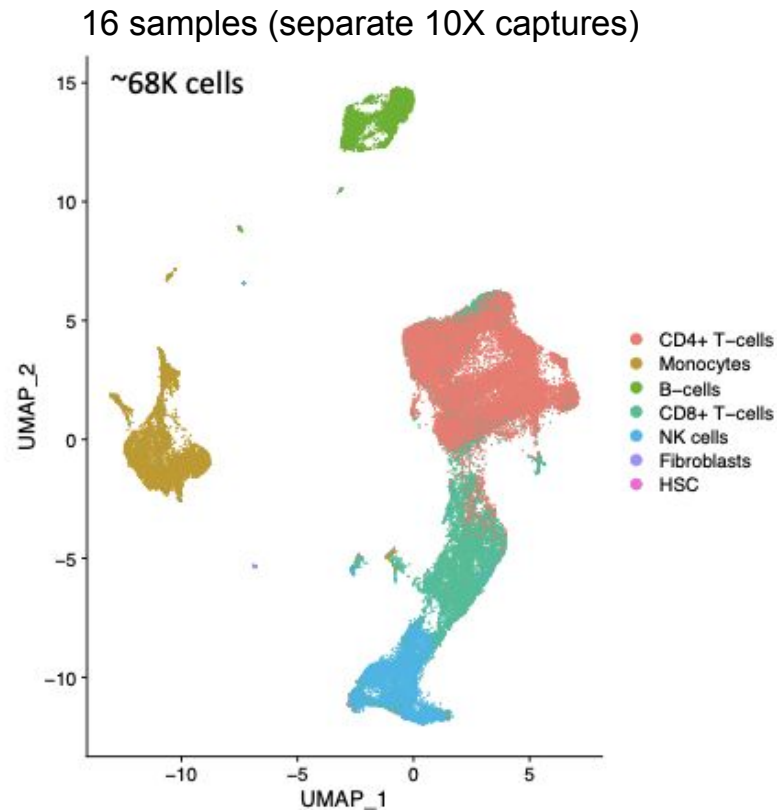
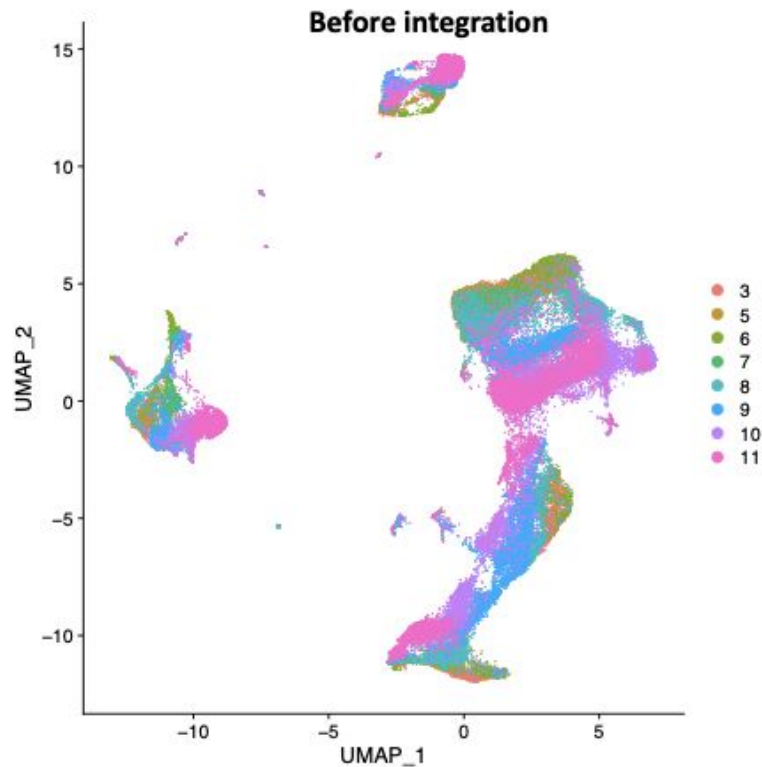


You could also use the integration technique to transfer cell type labels from a **reference** data set to a **query** data set!

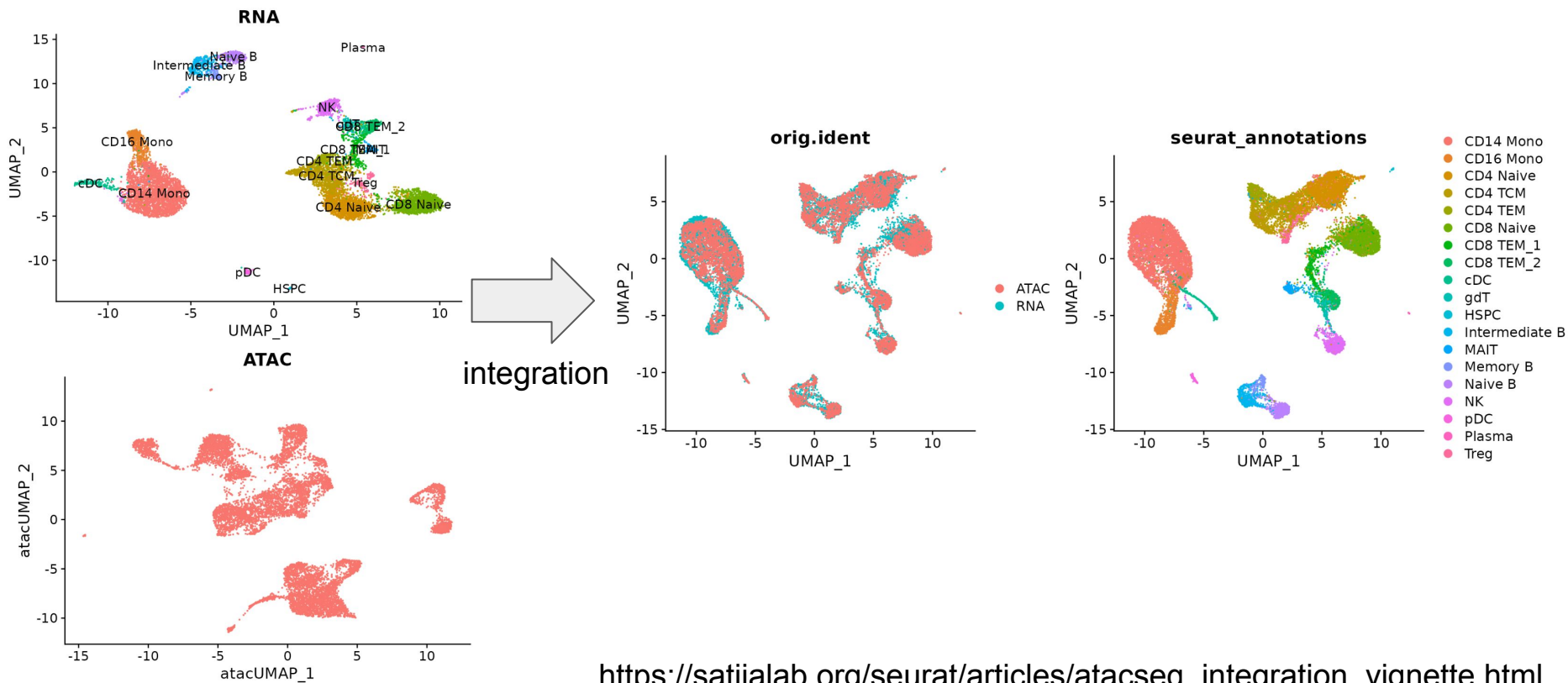
Use “integration” as a cell type classification method.

Always good to double-check your results.

# Example 1: Integration of single-cell RNA-seq samples from multiple 10X Captures



# Example 2: Integration of RNA and ATAC single-cell data



# VDJ Analysis (Multimodal Data)

- VDJ libraries will tell us information about the type of T-cell or B-cell receptor a cell has (clone)
- Then, we can use GEX libraries to tell us what those T-cells are doing transcriptionally and ADT libraries can help us confirm expression of proteins on their cell surfaces
- Using these complimentary data sets (via CITE-seq) will allow us to study immune cell clonotypes in greater depth through their normal development and disease progression

# VDJ analysis with 10X Genomics “cellranger vdj”

Performs four steps:

- 1) Assembly
- 2) VDJ calling (B-cell or T-cell)
- 3) Annotation of contigs with VDJ segments and **locates CDR3 region**
- 4) Clonotype grouping



# VDJ analysis with 10X Genomics “cellranger vdj”

## Analysis step one: cellranger vdj

### Output files:

#### Annotation Files

File	Description
<a href="#">clonotypes.csv</a>	High-level descriptions of each clonotype.
<a href="#">consensus_annotations.csv</a>	High-level and detailed annotations of each clonotype consensus sequence.
<a href="#">filtered_contig_annotations.csv</a>	High-level annotations of each high-confidence, cellular contig. This is a subset of <code>all_contig_annotations.csv</code> .
<a href="#">all_contig_annotations.csv,bed,json</a>	High-level and detailed annotations of each contig.
<a href="#">airr_rearrangement.tsv</a>	Annotated contigs and consensus sequences of VDJ rearrangements in the AIRR format.

	A	B	C	D
1	clonotype_id	frequency	proportion	cdr3s_aa
2	clonotype1	182	0.015041322	TRB:CASSYTGNEQYF;TRA:CAMVGSAGNKLTF
3	clonotype2	31	0.002561983	TRB:CASPWDRYNSPLYF;TRB:CASSDEGGQNTLYF;TRA:CATDENNTGKLTFF
4	clonotype3	29	0.002396694	TRB:CASSGQGAGEQYF;TRA:CAIVPGGSNAKLTF
5	clonotype4	29	0.002396694	TRB:CASSLRQSSYEYQYF;TRA:CALRWDAGAKLTF
6	clonotype5	24	0.001983471	TRB:CASSLGYNSPLYF;TRA:CAAASSGSWQLIF
7	clonotype6	20	0.001652893	TRB:CASSGTAETLYF;TRA:CALSEGNTAYKIVF
8	clonotype7	20	0.001652893	TRB:CASGETLYF;TRA:CAAEANQGGRALIF
9	clonotype8	19	0.001570248	TRB:CTCSADSSSQNTLYF;TRA:CAVRNQQGGRALIF
10	clonotype9	17	0.001404959	TRB:CASSLGLGGQEYF;TRA:CAIERTNAYKIVF;TRA:CAVRTGFASALTF
11	clonotype10	17	0.001404959	TRB:CASSIKSGNTLYF;TRA:CAAVRTGGNNKLTF
12	clonotype11	15	0.001239669	TRB:CASSLGLGYYEQYF;TRA:CALSTEGADRLLTF

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	barcode	is_cell	contig_id	high_confide	length	chain	v_gene	d_gene	j_gene	c_gene	full_length	productive	fwr1	fwr1_nt	cdr1	cdr1_nt	fwr2	fwr2_nt	cdr2	cdr2
2	AAACCTGAG	TRUE	AAACCTGAG	TRUE	540	TRB	TRBV13-3		TRBJ2-7	TRBC2	TRUE	TRUE	EAAVTQSPR:GAGGCTGCA NNHDY		AATAACCAT:MYWYRQDT:ATGTA CTGG SYVADS					TCA1
3	AAACCTGAG	TRUE	AAACCTGAG	TRUE	606	TRB	TRBV30	TRBD1	TRBJ2-3	TRBC2	TRUE	TRUE	SVLLYQKPNR:AGTGTCTCTC SQVVS		AGTCAAGTT:MFWYQQFQ ATGTTTGG ANEGSEA					GCA
4	AAACCTGAG	TRUE	AAACCTGAG	TRUE	606	TRA	TRAV14D-3-DV8		TRAJ32	TRAC	TRUE	TRUE	QQQVRQSP:CAGCAGCAG NSAFDY		AACAGTGTCT:FPWYQQFP:TTCCCATGG1 ILSVSDK					ATA
5	AAACCTGAG	TRUE	AAACCTGAG	TRUE	516	TRB	TRBV31		TRBJ2-1	TRBC2	TRUE	TRUE	AQTIHWPPV:GCTCAGACT: GKSSPN		GGGAAATCA:LYWYQQAT:CTCTACTGG1 SITVG					TCTA
6	AAACCTGAG	TRUE	AAACCTGAG	TRUE	541	TRA	TRAV14D-1		TRAJ57	TRAC	TRUE	TRUE	QQQVRQSP:CAGCAGCAG DSTFNY		GACAGACT:FPWYQQFP:TTCCCATGG1 IRSVSDK					ATA
7	AAACCTGAG	TRUE	AAACCTGAG	TRUE	565	TRB	TRBV1		TRBJ2-5	TRBC2	TRUE	TRUE	VTLEQNPRV:GTGACTTTG: NSQYPW		AATTCACAG1:MSWYQQDL:ATGAGCTGG1 LRSPGD					CTG