

# Working material for this workshop

1. Download the material
2. Pre-workshop instructions

# Single cell RNA-sequencing analysis

Gladstone Institutes

Ayushi Agrawal, Michela Traglia  
Reuben Thomas

Gladstone Bioinformatics Core

October 16-17 2023

# Introductions

Reuben Thomas

Associate Core Director

Michela Traglia

Statistician III

Ayushi Agrawal

Bioinformatician II

# Goals

- To give an overview of current practices
- Highlight assumptions and limitations underlying the methods
- Understand the relevance and impact of intermediate steps
- Experience how to analyze scRNA-seq data in R



# Workshop outline

- **Session 1 (Mon, 9am-12pm)**

Load the data, quality control, normalization, feature selection, dimensionality reduction.

- **Session 2 (Mon, 1-4 pm)**

Clustering, finding marker genes and Demo

- **Session 3 (Tue, 1-4 pm)**

Advanced discussion on normalization, differential analysis, and batch-correction, Q&A.

December - scATAC-seq and multi-omics workshops

# Outline of Sessions 1-2

## Session 1

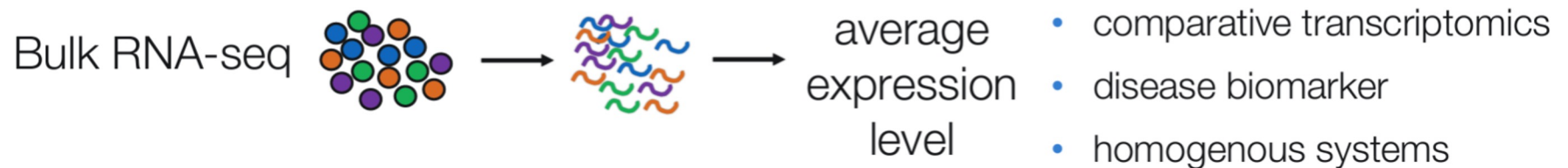
1. Introduction
2. Pre-processing raw sequencing data to get counts
3. Loading data in R using Seurat
4. Quality control
5. Break (10 min)
6. Normalization
7. Feature selection
8. Dimensionality reduction

## Session 2

1. Clustering
2. Finding marker genes + Demo

# 1. Introduction

# Bulk vs Single Cell RNA Sequencing (scRNA-seq)

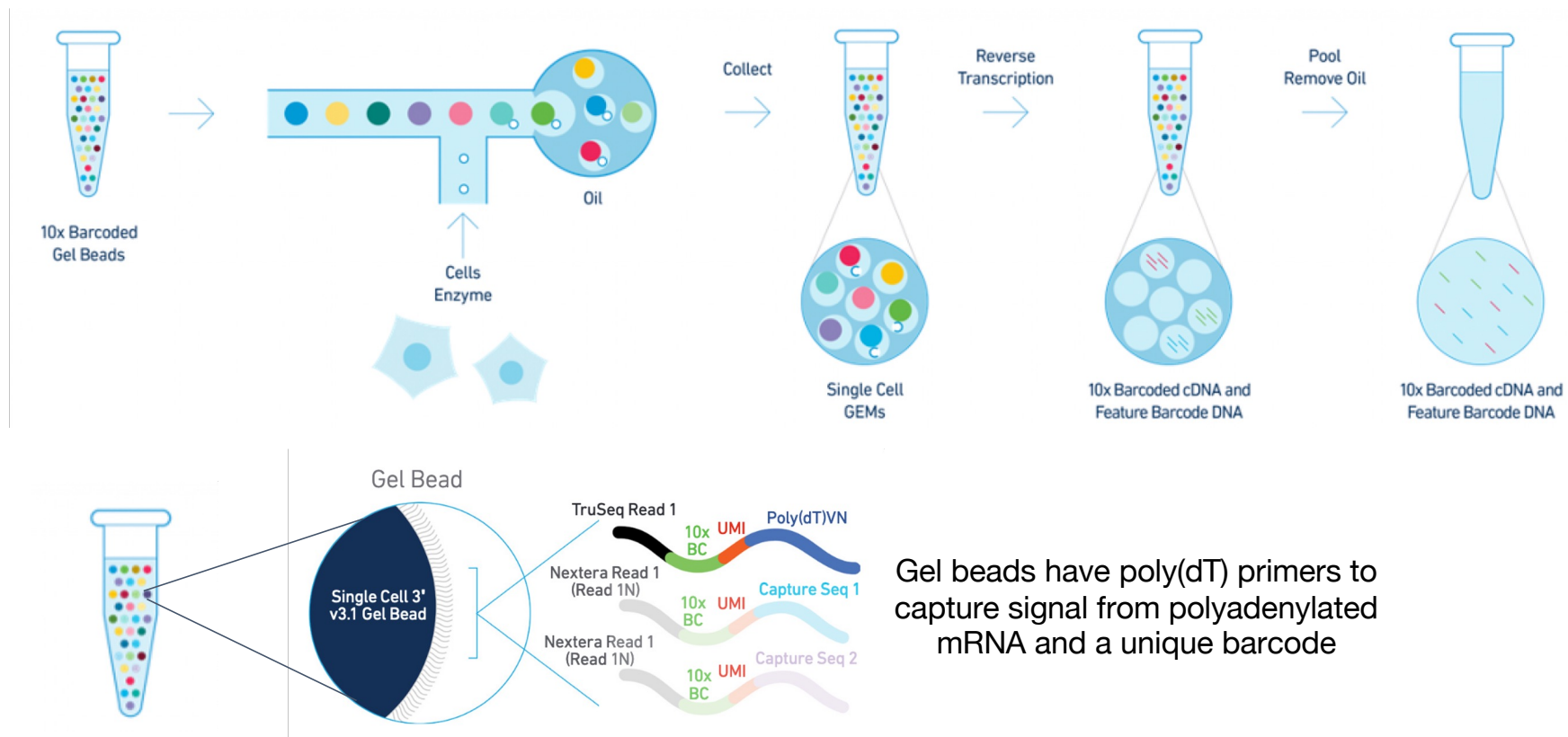


*In combination with spatial transcriptomics and single-cell multi-omics, applications of single-cell technologies in new areas are emerging rapidly*

It might be possible to achieve project goals with experiment designs that avoid scRNA-seq

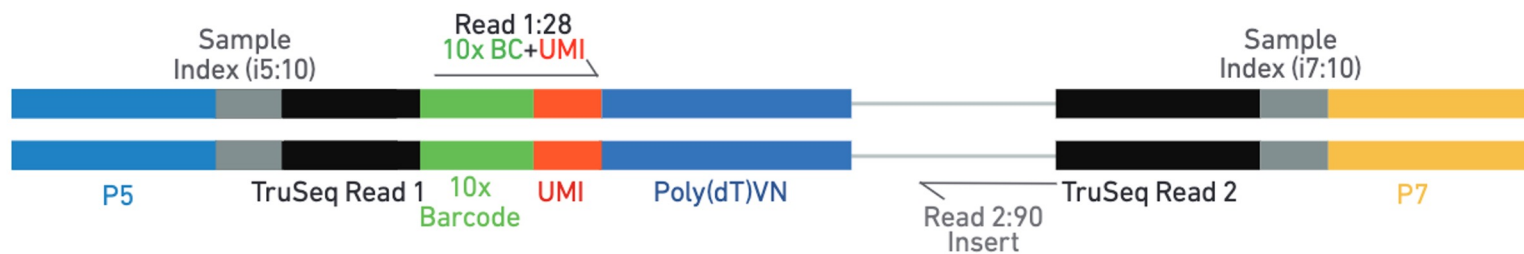
- Testing for existence of new cell types?  
Studying a diverse population with no known markers?  
=> scRNA-seq may be useful
- Can sort cells by FACS into sub-populations of interest?  
=> bulk RNA-seq may suffice
- ...

Single-cell technologies allow tagging cell-of-origin of reads with barcodes, which makes single-cell RNA-seq possible



# Read structure and sequencing

## Chromium Single Cell 3' Gene Expression Dual Index Library



Sequencing read	Description	Example file name
Read R1	Cell and UMI read	MySample_S1_L001_R1_001.fastq.gz
i7 index	Sample index read	MySample_S1_L001_I1_001.fastq.gz
i5 index	Sample index read	MySample_S1_L001_I2_001.fastq.gz
Read 2	Insert read (transcript)	MySample_S1_L001_R2_001.fastq.gz

# Current limitations of scRNA-seq

- Expensive
    - Typical studies are under-powered
    - Computational challenges
  - Technological
    - Not recommended for alternative splicing studies, low expressed genes, etc.
    - Low sample size
- => Data used to generate hypotheses that require testing

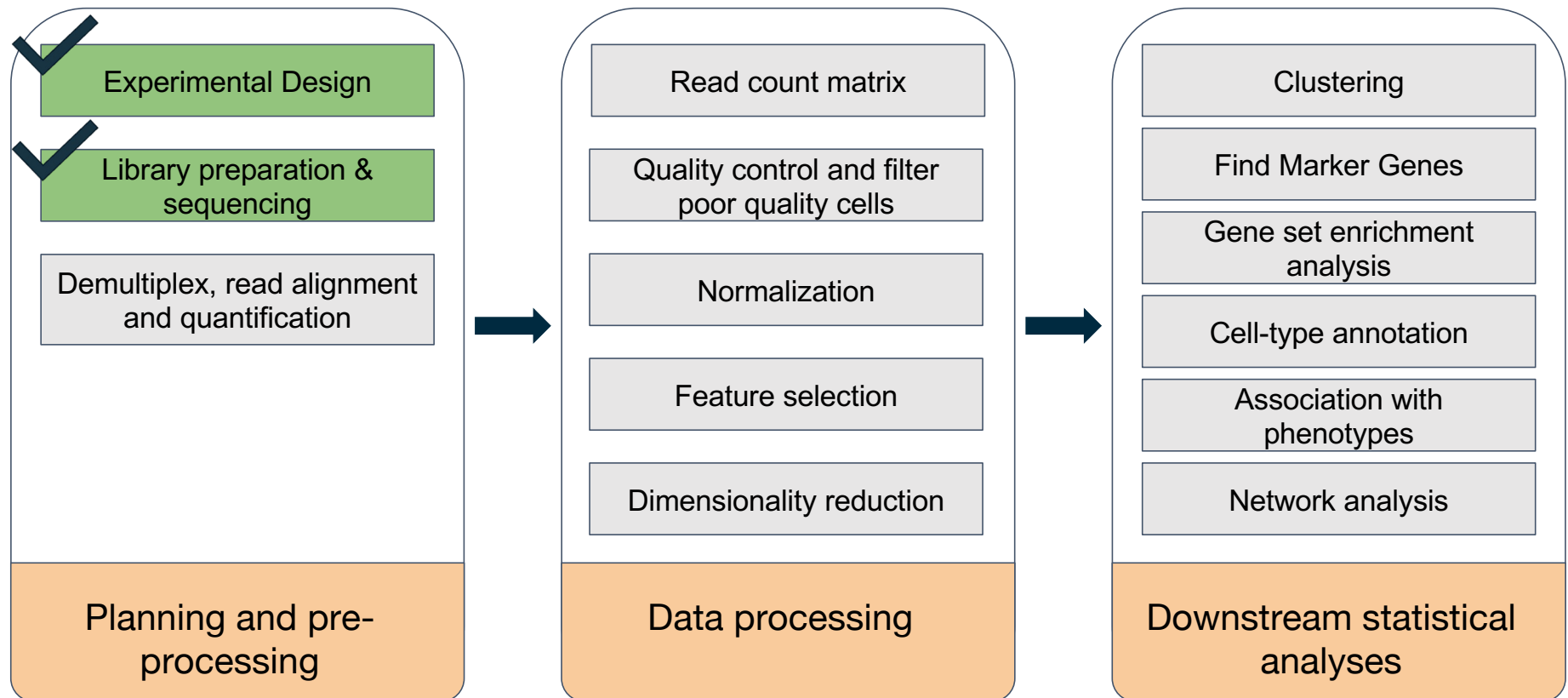


## scRNA-seq analysis = Challenges with bulk RNA-seq analysis + More

- Data is sparser than bulk RNA-seq
- High dimensionality
- Few replicates, if any
- Technical noise, batch effects, ...
- Nonlinear gene expression dynamics

Gene	Cell #1	...	Cell #10X
Gene #1	7	0	0
Gene #2	0	0	3
...	0	0	0
...	1	0	0
Gene #N	4	0	1

# scRNA-seq workflow



## 2. Pre-processing raw sequencing data to count matrix

# Pre-processing steps



1. Single cell RNA-seq library preparation (consult with the Genomics Core)



1. Deep sequencing (e.g., by UCSF CAT Core)

 [ucsf-wynton / tutorials](#)

1. [Wynton tutorial](#) to move files from CAT Core (link in slide notes)

**WYNTON**  
UCSF Research Computing at Scale

1. Pre-process raw data (e.g., using Cell Ranger from 10x Genomics pre-installed on Wynton)

 [sdparekh / zUMIs](#)

**10x** GENOMICS

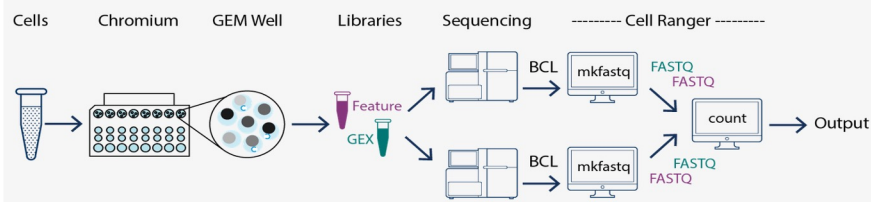
[kallisto](#) | [bustools](#)

 [dpeerlab / seqc](#)  
forked from [ambrosejarr/seqc](#)

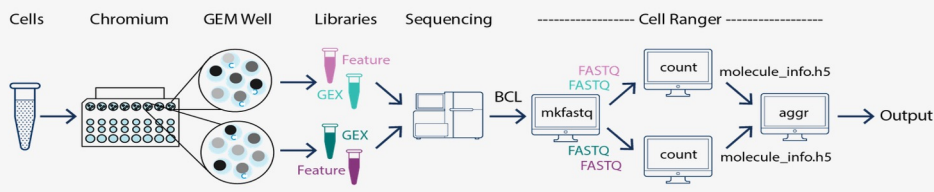
1. QC runs and initial inspection (e.g., using Loupe Browser from 10x Genomics)

# Cell Ranger pipeline available on Wynton from the CBI software repository

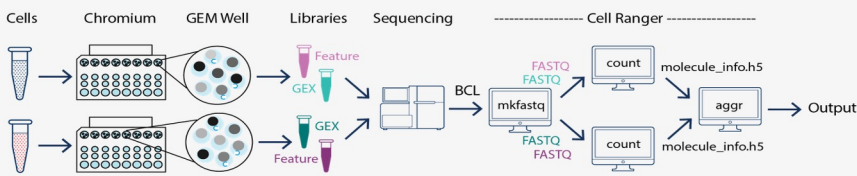
One Sample, One GEM well, Multiple Flowcells



One Sample, Multiple GEM Wells, One Flowcell



Multiple Samples, Multiple GEM Well, One Flowcell



1. Demultiplex (BCL to FASTQ)
  - ***cellranger mkfastq***
2. Alignment, filtering, barcode counting, and UMI counting
  - ***cellranger count***
3. Aggregate counts from multiple runs
  - ***cellranger aggr***

(Images from 10x Genomics website)

# Cell Ranger outputs include count matrices

- BAM files with aligned reads
- **web\_summary.html**
- **Count matrices:** ~20000 genes (rows) and ~6000 cells (columns)
  - Unfiltered matrices (all known-good barcodes, i.e., both cells and empty drops)
  - Filtered matrices (only cellular barcodes)
  - Matrix data organized in three files (see next slide)
- Secondary analysis (e.g., dimensionality reduction, clustering, etc.)
  - Often ignored, e.g., perform secondary analysis using Seurat instead
- Loupe file to interactively view secondary analysis results with Loupe Browser from 10x Genomics
- Molecule info

# Cell Ranger outputs

(more detail on HTML output)

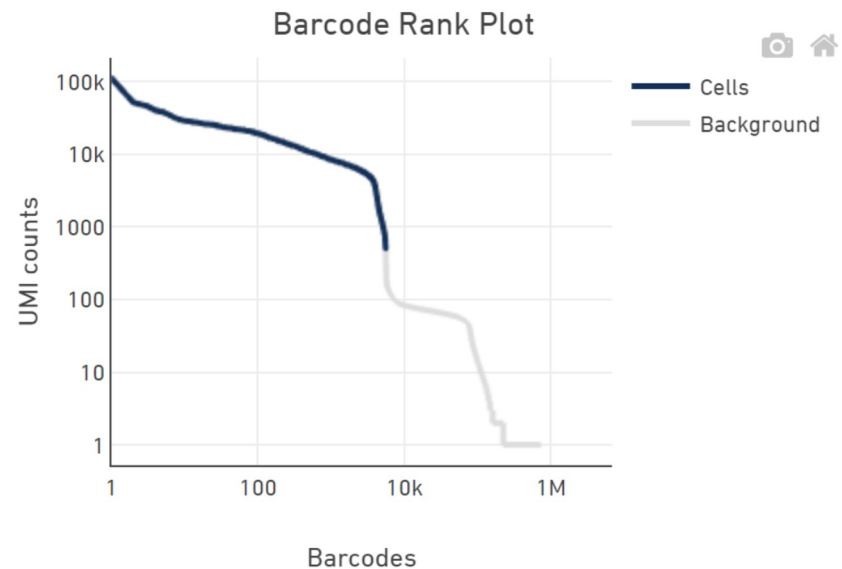
- web\_summary.html
- Barcode rank plot shows distribution of UMI counts, which is used to identify empty drops
- Some drops have cells; many have ambient RNA (=> classification problem)



# HTML output shows distribution of UMI counts, which is used to identify empty drops

- Some drops have cells; many have ambient RNA (=> classification problem)
- Assumption: UMI counts for cell-associated barcodes >> UMI counts for background-associated barcodes
- Assumption: Gene-wise counts for cell-associated barcodes resemble expression profiles of real cells (based on the EmptyDrops method)

## Cells ?

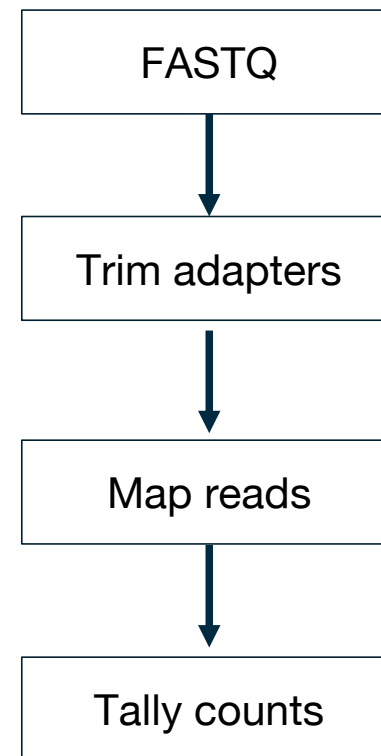


(Image from 10x Genomics website)



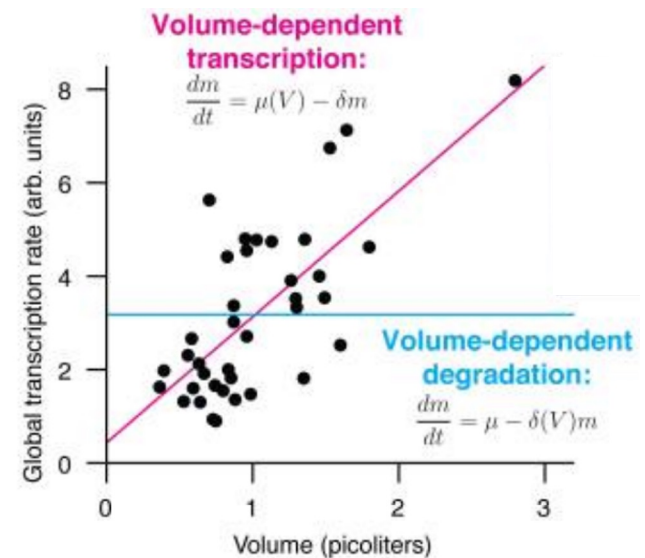
# Cell Ranger outputs count matrices before and after filtering empty drops

- Tools in common with bulk RNA-seq
  - Illumina's bcl2fastq
  - STAR



## Additional consideration 1: Cell Ranger outputs filtered and unfiltered count matrices. Which one to use?

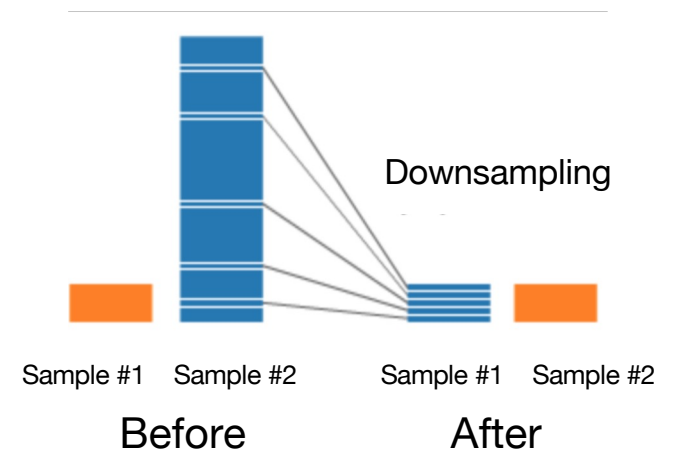
- Say, there are cells of different sizes in the sample
- Small cells might have low total RNA leading to low total UMI counts
- Filtering all barcodes with less than certain UMI counts might filter small cells
- EmptyDrops by Lun et al., 2019 compares count profile for each barcode with ambient RNA profile (similar implementation in Cell Ranger 3)
  - => Small cells retained but empty drops removed
  - => Alternative to Cell Ranger filtering



(Image from Padovan-Merhar et al, 2015)

## Additional consideration 2: Should we downsample reads when aggregating batches?

- Say, multiple batches with different sequencing depths
- By default, Cell Ranger downsamples deeper sequenced batches to equalize mapped read counts
- DropletUtils allows to downsample UMI counts or read counts
- Hafemeister and Satija, 2019 developed a count normalization method that recommends skipping downsampling (*sctransform*)



## Additional consideration 3: R vs Python vs Galaxy for the next steps

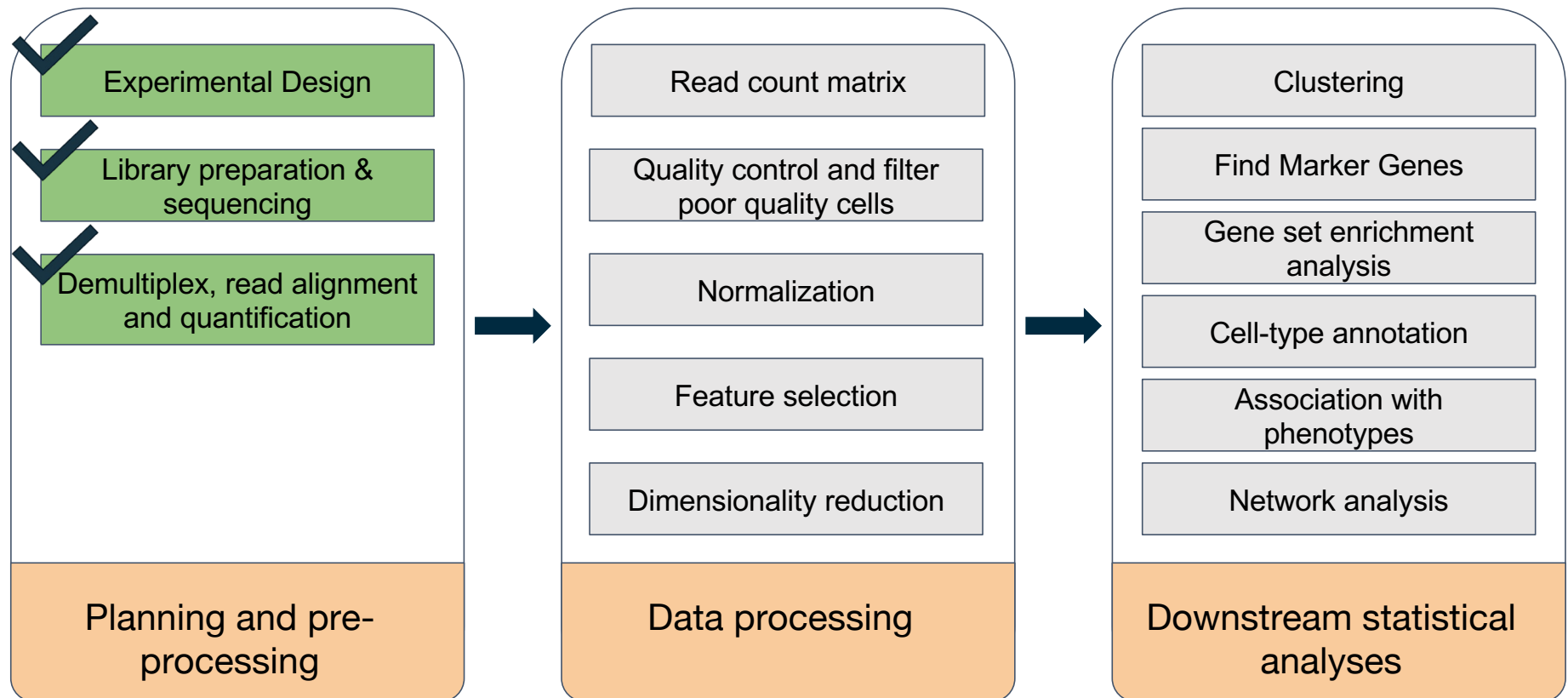
- R (Seurat, Bioconductor, Monocle3, etc.)
  - R offers a rich collection of libraries for statistics and data science
  - Popular in academia
  - Bioconductor has a lot more than single-cell RNA-seq
- Python (scanpy and other)
  - Much faster than R packages (5-90 times speedup for steps)
  - Easier access to advanced machine-learning packages (e.g., tensorflow)
- Galaxy
  - R or Python tools can be installed
  - Provides GUI-based solution

## Knowledge check

Cell Ranger output includes

1. Count matrices
2. Web summary
3. Loupe file with secondary analysis results
4. All of the above
5. None of the above

# scRNA-seq workflow



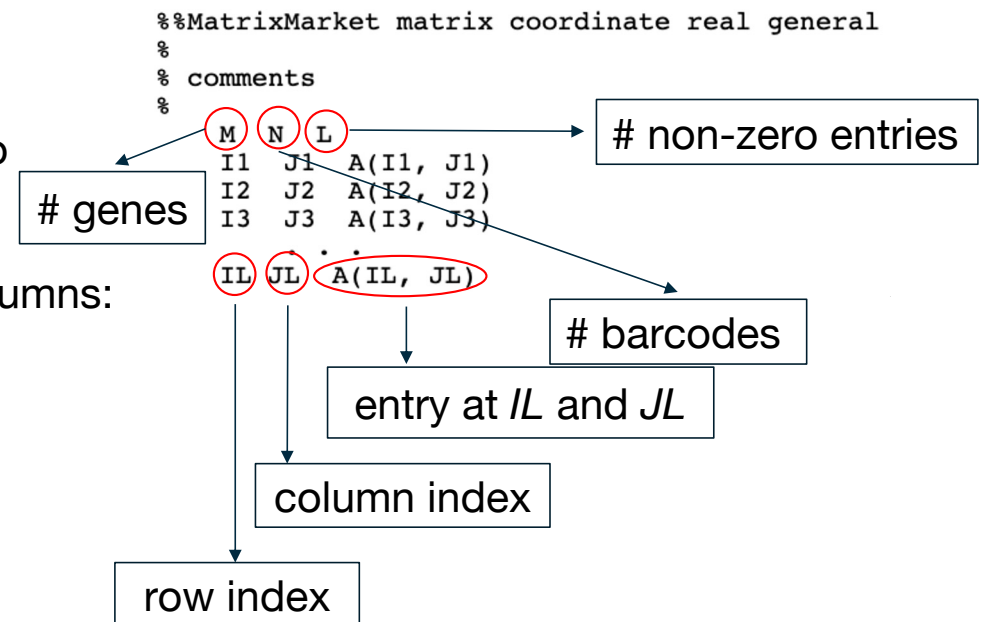
### 3. Loading data in R using Seurat

# Count data is organized in three files (1/3)

matrix.mtx.gz

.mtx format

- .gz :- Compression format
- .mtx :- Matrix Market Exchange format
- Favored for sparse matrices (mostly zero entries)
- Consider table with 20k rows and 6k columns:
  - # of genes ( $\sim 10^4$ )
  - # of cellular barcodes ( $\sim 10^4$ )
  - # non-zero entries ( $\sim 10^6$ )
  - => sparse matrix





## Count data is organized in three files (2/3)

- ‘Feature’ is data science jargon for any measurable property
- In scRNA-seq, feature = expression level of a gene
- Gene identifiers are stored in features.tsv.gz
- Order is the same as order of rows in matrix.mtx.gz
- Example:

<b>ENSMUSG00000051951</b>	<b>Xkr4</b>	<b>Gene Expression</b>
<b>ENSMUSG00000089699</b>	<b>Gm1992</b>	<b>Gene Expression</b>
<b>ENSMUSG00000102331</b>	<b>Gm19938</b>	<b>Gene Expression</b>
...	...	...
...	...	...
...	...	...

# Count data is organized in three files (3/3)

- Each cell is identified by a barcode
- List of barcodes is saved in barcodes.tsv.gz
- Order is same as order of columns in matrix.mtx.gz
- Numeric suffix after barcode sequence gives sample number in aggregated matrix
- Example:

```
AAAGCCAAGCATCCTA-1  
AAAGCCAAGCGTGT-1  
...  
...  
TTTGAGCAAATCGTC-24  
TTTGAGGTCAAGCCC-24
```

## Load the data in R using the Read10X function from the Seurat package and create Seurat object

- Seurat is an R package designed for QC, analysis, and exploration of single-cell RNA-seq data.
- Functions are available to convert Seurat objects to different formats required by Bioconductor and scanpy

# Seurat object contains all the relevant information

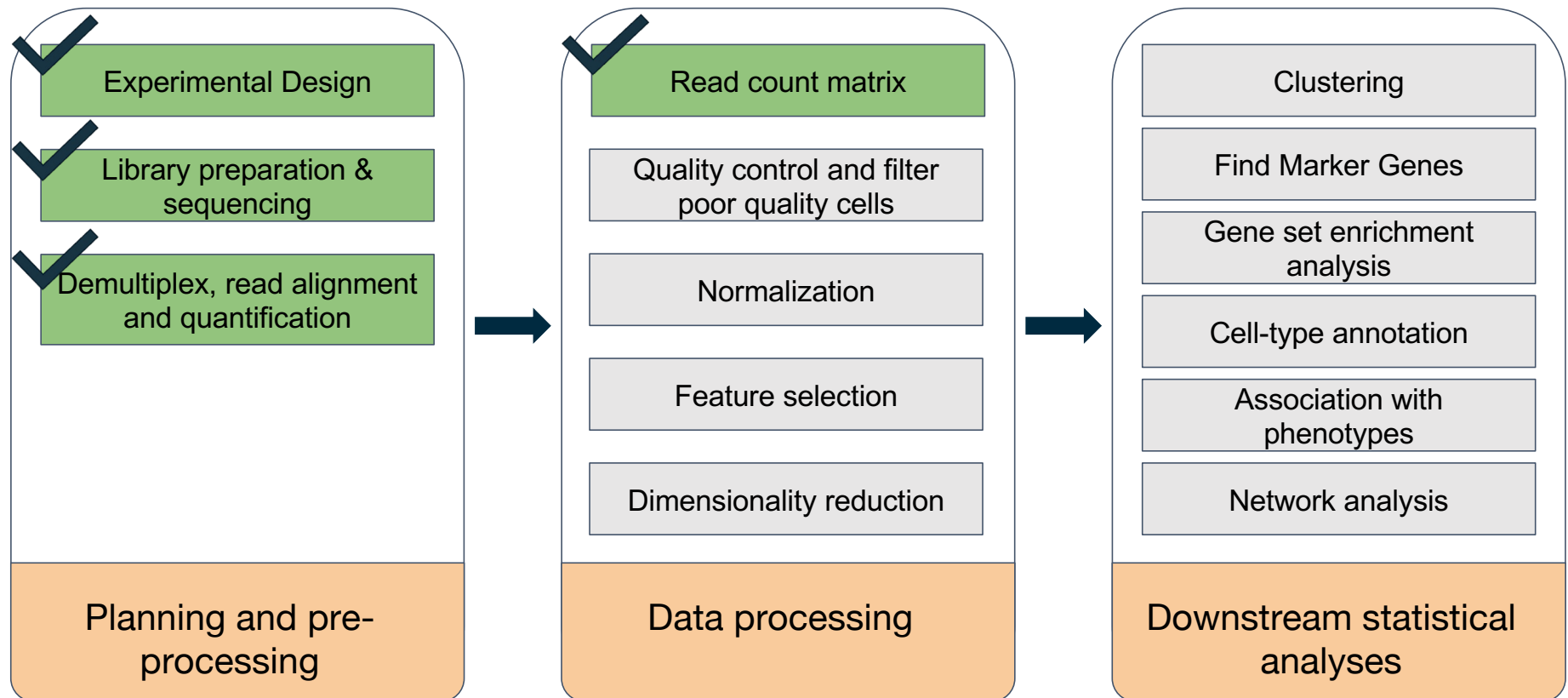
Name	Type	Value
▼ data	S4 [21352 x 5438] (SeuratObject::Seurat)	S4 object of class Seurat
▶ assays	list [1]	List of length 1
▶ meta.data	list [5438 x 6] (S3: data.frame)	A data.frame with 5438 rows and 6 columns
active.assay	character [1]	'RNA'
▶ active.ident	factor	Factor with 18 levels: "0", "1", "2", "3", "4", "5", ...
▶ graphs	list [2]	List of length 2
neighbors	list [0]	List of length 0
▶ reductions	list [3]	List of length 3
images	list [0]	List of length 0
project.name	character [1]	'Hello_scWorld'
misc	list [0]	List of length 0
▶ version	list [1] (S3: package_version, numeric_version)	List of length 1
▶ commands	list [10]	List of length 10
tools	list [0]	List of length 0

## Knowledge check

Seurat object contains only the raw counts from Cell Ranger.

1. True
2. False

# scRNA-seq workflow

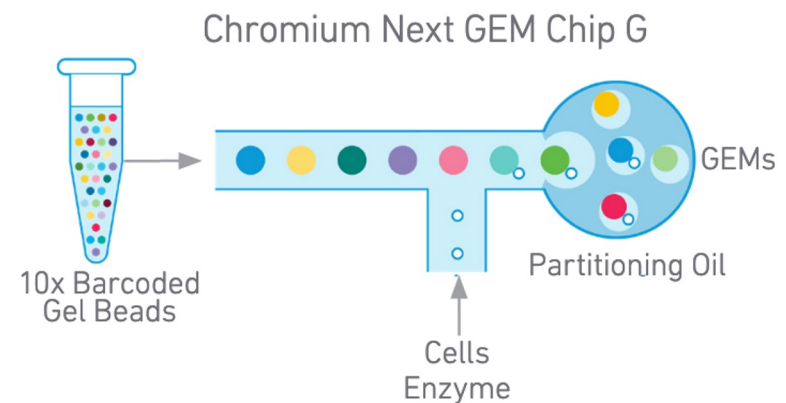


## 4. Quality control

Not all barcodes represent good quality cells  
Not all genes are detected in all cells  
=> Filtering of poor quality or uninteresting data needed

Common filtering criteria:

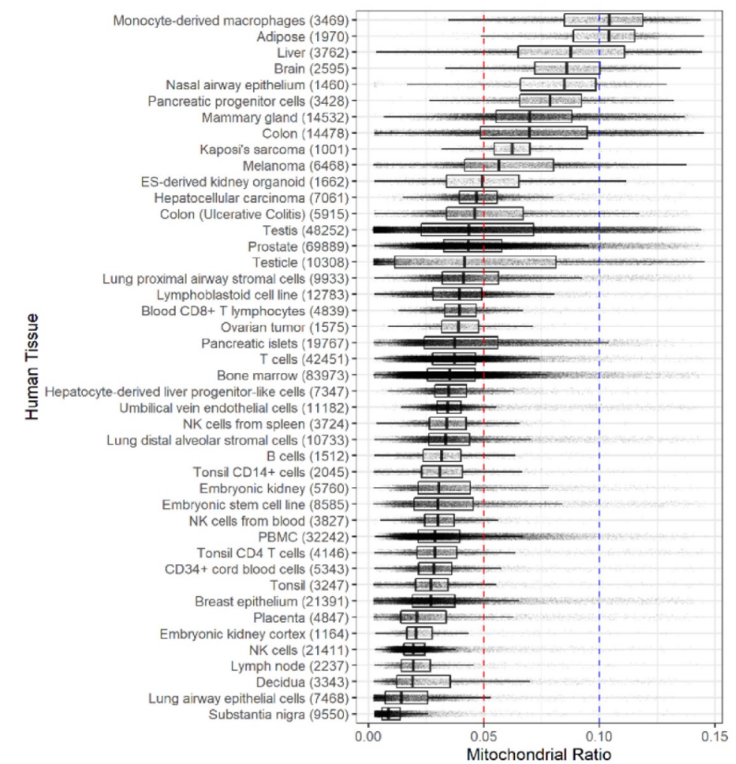
- # genes detected in a cell
  - If too few, empty drops or dead cells?
  - If too many, doublets or multiplets?
- # unique molecules in a cell
- Percent of mitochondrial genes
  - If too high, low quality/dying cells?
- See the R script





## Additional consideration 1: Careful selection of mitochondrial % cutoff

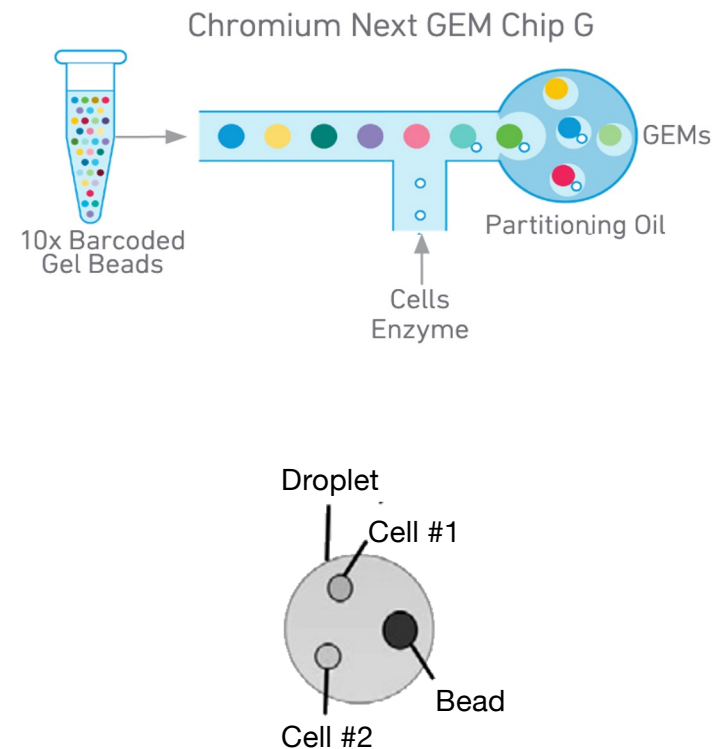
- Mitochondrial transcripts indicate cell stress
  - “Normal” value depends on the cell type
  - Ex. 1: 30% maybe normal for cardiomyocytes
  - Ex. 2: 30% in lymphocyte => stress
- Osorio and Cai, 2020 studied published data
  - 5% cutoff for mouse cells performed well
  - for humans, a higher cutoff required (proposed 10%)
  - provided reference values for range of human/mouse cell types and tissues
- *scater* provides methods that adapt cutoffs to data
  - any cell with QC metrics >3 MAD from median across all cells is outlier



(Image from Osorio and Cai, 2020)

## Additional consideration 2: What causes doublets and how to identify them

- Ideal world => all drops have a single cell
- Real world =>
  - some drops have multiple cells
    - there is chance at play
    - some cell types are harder to separate
  - many drops have no cells

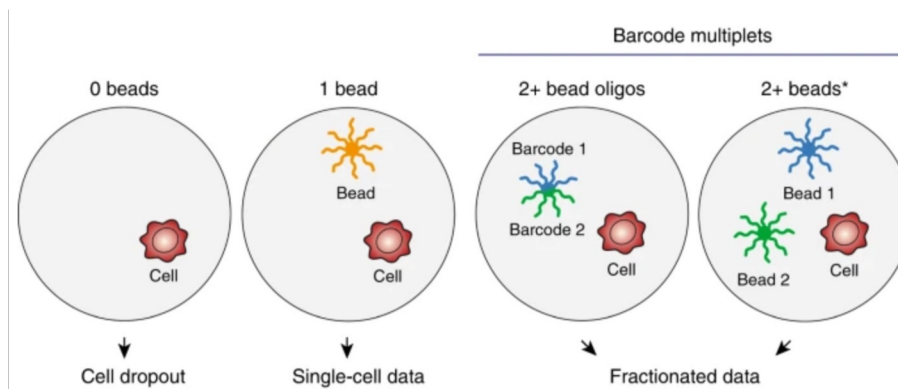


Count-based filtering assumes all cells have equal total mRNAs  
May not be true => Other tools to detect doublets

- DoubletFinder (McGinnis et al, 2019) and Scrublet (Wolock et al., 2019)
  - Simulate doublets by averaging expression profile of random pair of cells
  - Barcodes with profiles close to simulated doublets represent doublets
  - Limitation: Mixed-lineage cell states may be present with hybrid transcriptomes
- DoubletDecon, scds, Solo, ...

## Additional consideration 3: Barcode multiplets vs cell multiplets

- 10x Genomics does “super-loading” of beads
- Some drops may have one cell but multiple beads  
=> leads to barcodes with fractionated single-cell data (Lareau et al, 2020)
- Can lead to mischaracterizing artifacts as new cell types
- Not part of typical analysis currently

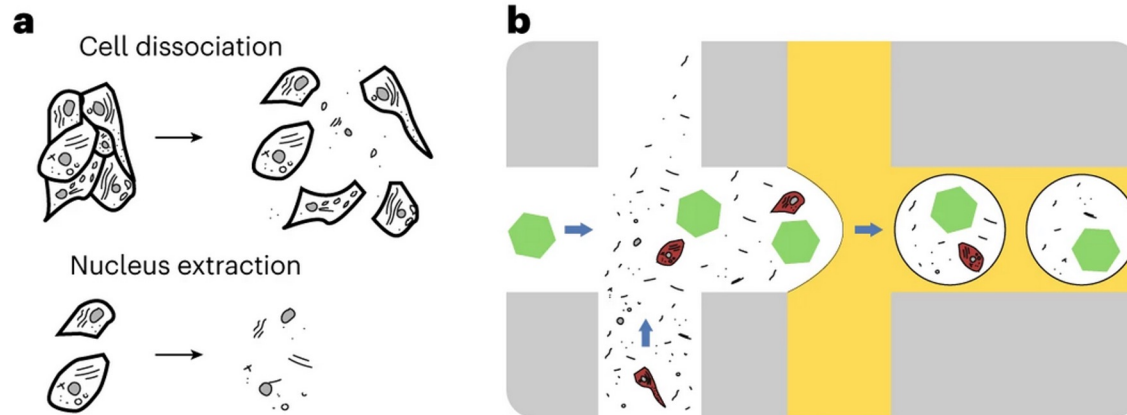


(Image from Lareau et al., 2020)

## Additional consideration 4: Ambient RNA contamination

- Ambient RNA, from damaged cells, introduces noise and confounds signal.
- Removing ambient RNA improves the accuracy of gene expression profiles by reducing this interference.
- CellBender, SoupX, ..

Phenomenology of ambient RNA



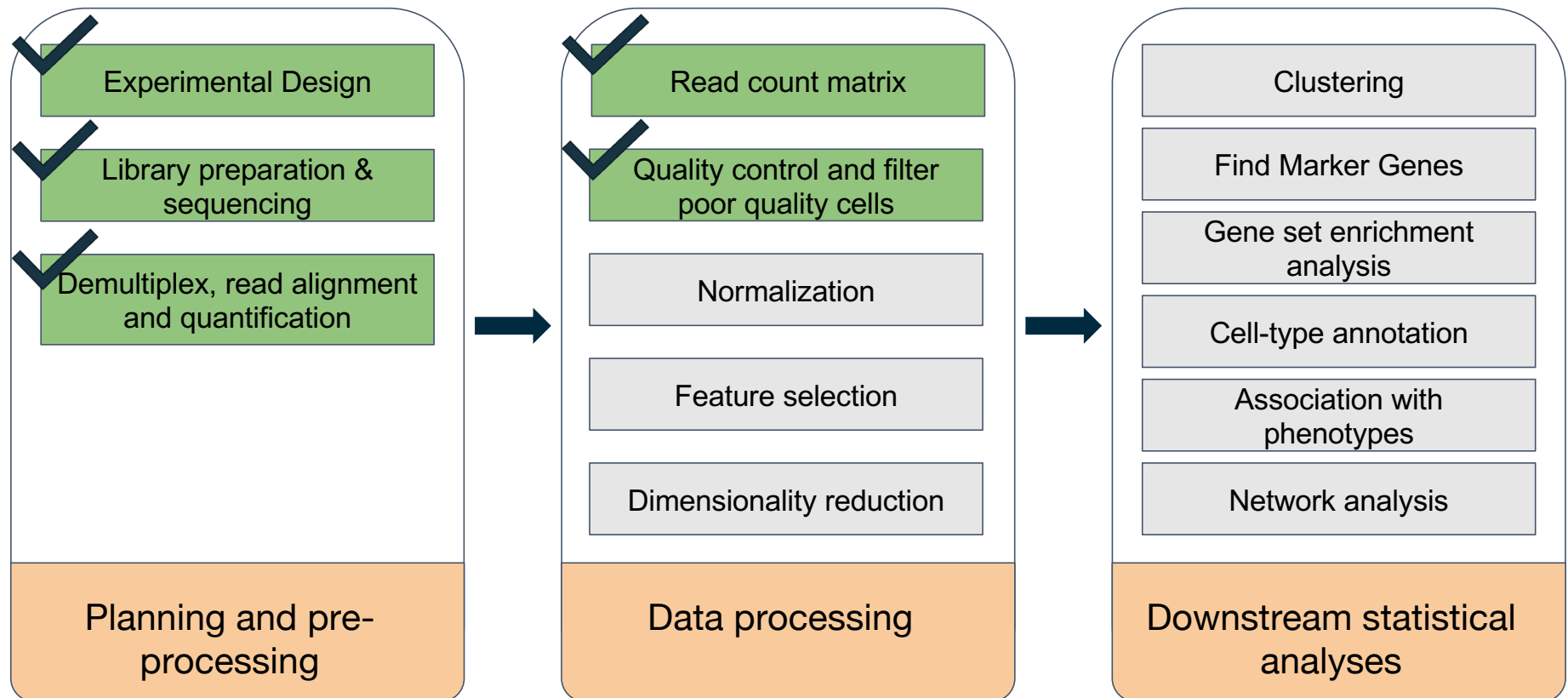
(Image from Fleming et al., 2023)

## FAQ: How to identify the mitochondrial genes?

- Mitochondrial gene symbols start with
  - “mt-” for mouse
  - “MT-” for humans

Break

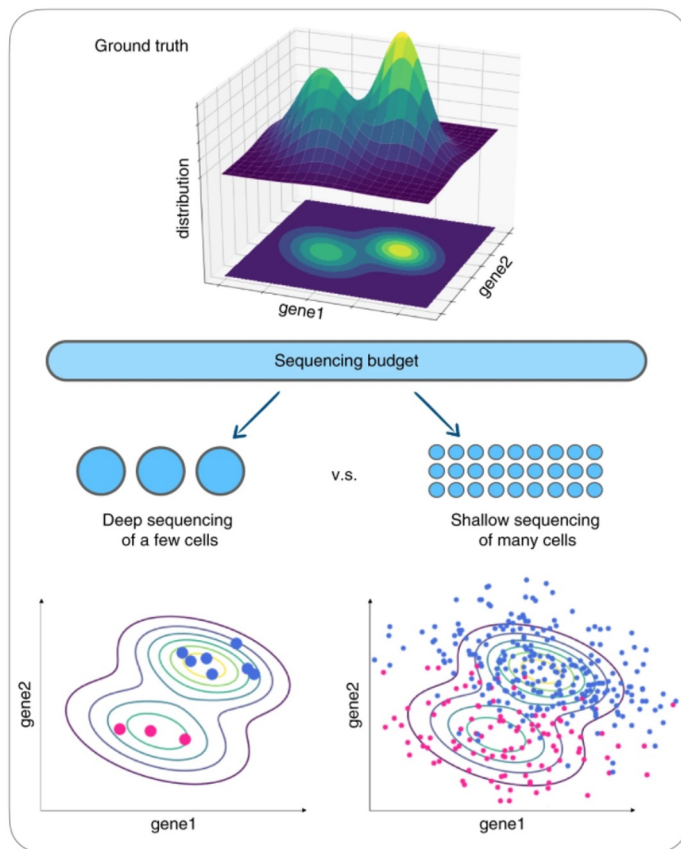
# scRNA-seq workflow





## 6. Normalization

# Motivation



The observed sequencing depth  
(number of molecules detected per cell)  
can vary significantly between cells

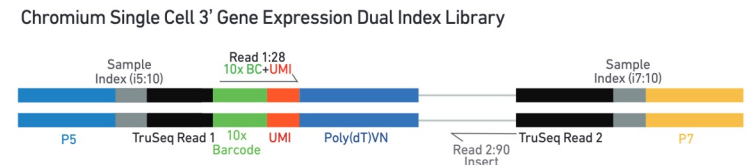
Remove the influence of technical  
effects while preserving true biological  
variation

Differences in cellular sequencing depth do  
not influence downstream analytical tasks

<https://www.nature.com/articles/s41467-020-14482-y>

# UMI counts

- Unique molecular identifiers (UMIs) are short sequences to tag each molecule in a library
- More duplicates of singular molecule of the original cell
- Filter out duplicate reads and PCR errors
- UMI counts vs Reads
- >60-90% of the data are zeros
- UMI counts represent the absolute number of observed transcripts (per gene, cell or sample) -> sequencing depth



# Duplicates after PCR

	Cellular barcode	UMI		
Cell 1	TTGCCGTGGTGT	GGCGGGGA	CGGTGTTA	DDX51
	TTGCCGTGGTGT	TATGGAGG	CCAGCACC	NOP2
	TTGCCGTGGTGT	TCTCAAGT	AAAATGGC	ACTB
Cell 2	CGTTAGATGGCA	GGGCCGGG	CTCATAGT	LBR
	CGTTAGATGGCA	ACGTTATA	ACGCGTAC	ODF2
	CGTTAGATGGCA	TCGAGATT	AGCCCTTT	HIF1A
Cell 3	AAATTATGACGA	AGTTTGTA	GGGAATTA	ACTB ← 2 reads, 1 molecule
	AAATTATGACGA	AGTTTGTA	AGATGGGG	
	AAATTATGACGA	TGTGCTTG	GACTGCAC	RPS15
Cell 4	GTTAAACGTACC	CTAGCTGT	GATTTTCT	GTPBP4
	GTTAAACGTACC	GCAGAAAGT	GTTGGCGT	GAPDH
	GTTAAACGTACC	AAGGCTTG	CAAAGTTC	ARL1 ← 2 reads, 2 molecules
	GTTAAACGTACC	TTCCGGTC	TCCAGTCG	

(Thousands of cells)

- Biological duplicates: Reads with **different UMIs** mapping to the same transcript were derived from **different molecules** - each read should be counted.
- Technical duplicates: Reads with the **same UMI** originated from the **same molecule** - the UMIs should be collapsed to be counted as a single read.

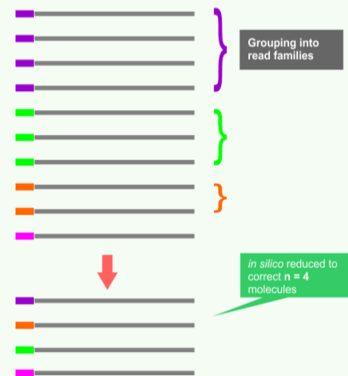
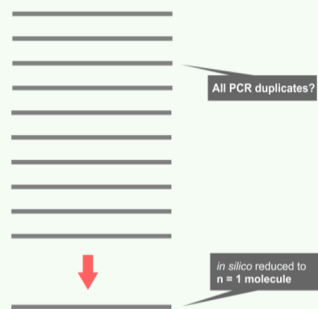
modified from Macosko EZ et al. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets, Cell 2015

# UMI applications

PCR duplicate removal without UMIs

PCR duplicate removal with UMIs

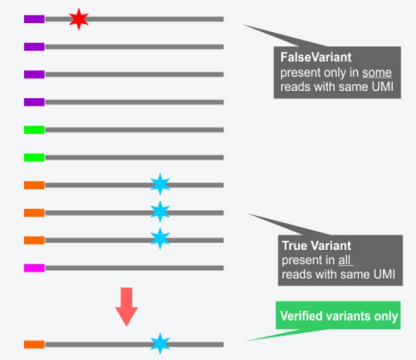
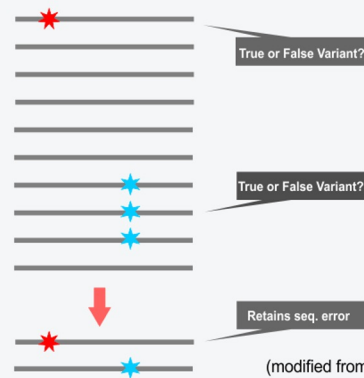
reference sequence



Variant calling without UMIs

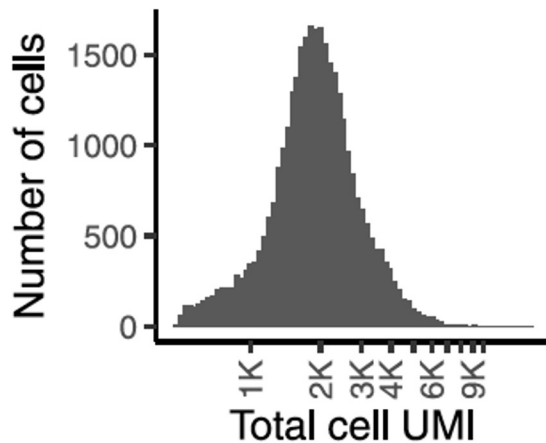
Variant calling with UMIs

reference sequence

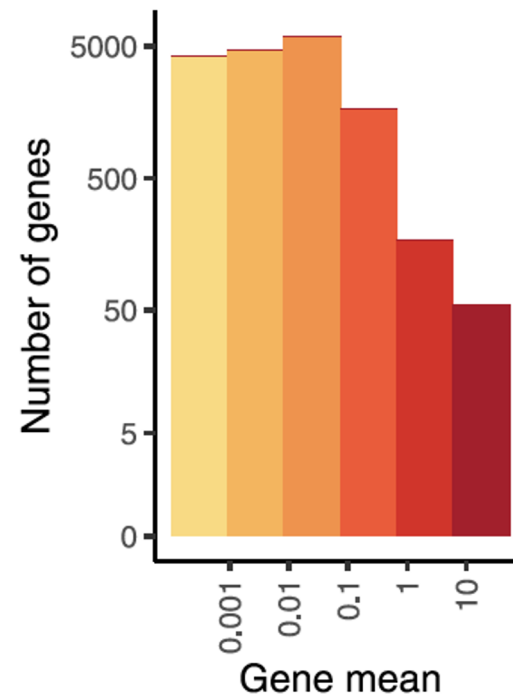


# Technical variability: Sequencing depth variation across cells

N= 33,148 human peripheral blood mononuclear cells (PBMC) and ~17,000 genes in at least 5 cells.



sequencing depth variation

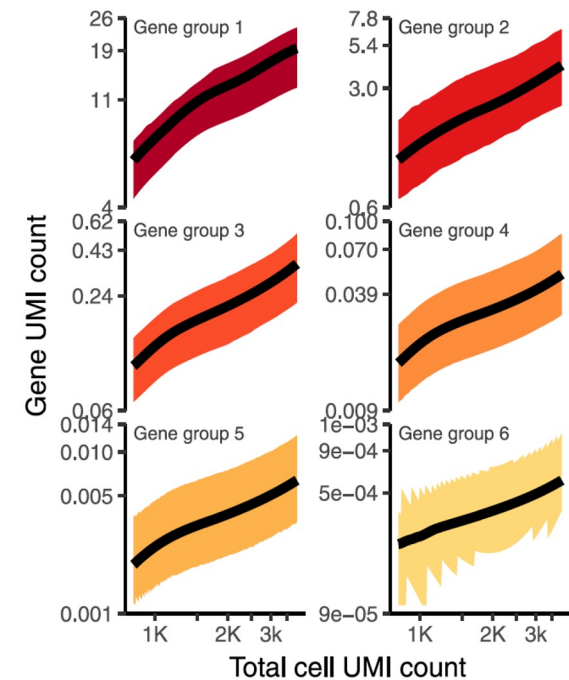


6 groups of genes - average expression

(Image from Hafemeister and Satija, 2019)

# Technical variability: Sequencing depth variation across cells

- ✦ Variation in sequencing depth across single cells  
⇒ 10 out of 10k in Cell #1  $\neq$  10 out of 200k in Cell #2
- ✦ True signal does not depend on sequencing depth  
⇒ Normalized expression of gene should not depend on sequencing depth  
⇒ Not true for raw data, i.e., raw data needs to be normalized



(Image from Hafemeister and Satija, 2019)

## Knowledge check

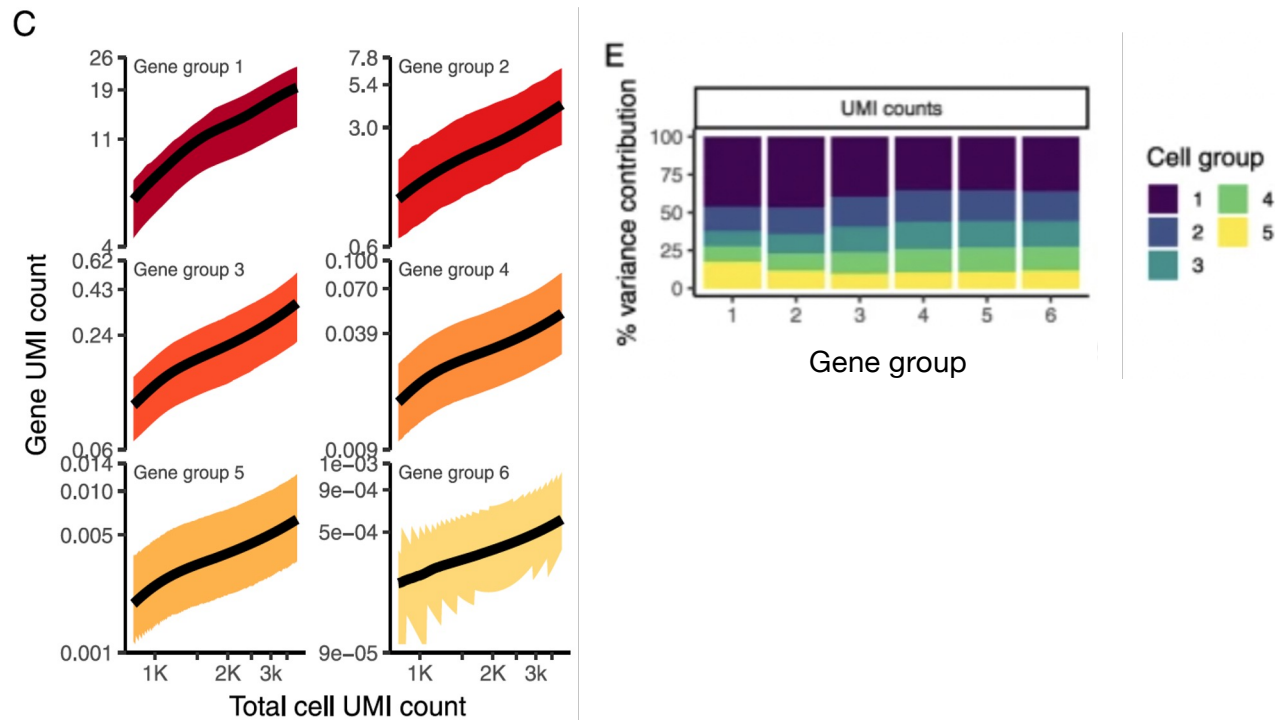
Are UMI counts comparable between samples?

1. Yes
2. No



# Raw counts should be adjusted by total sequencing depth

Higher sequencing depth, higher counts



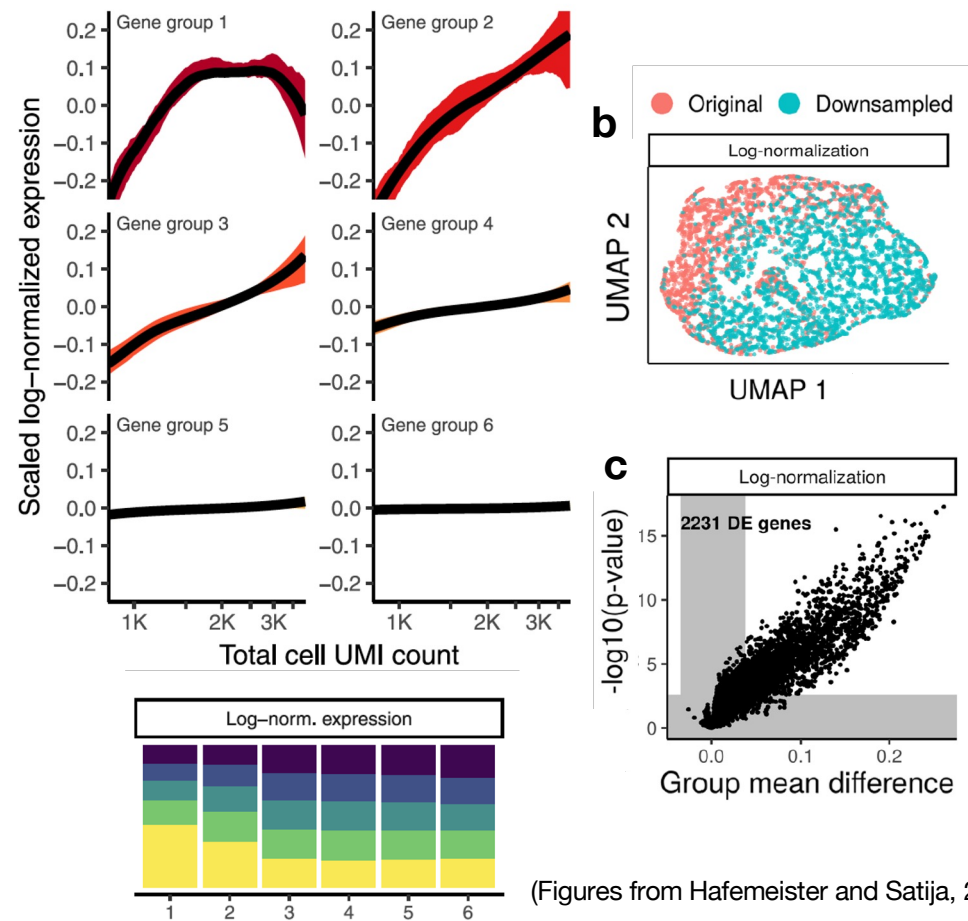
(Image from Hafemeister and Satija, 2019)

Standard workflows in Seurat and scanpy utilize log-normalization  
Steps: 1. Divide by total expression, 2. Scale and add pseudo-count, 3. log-transform

- Inspired by bulk RNA-seq data
  - Reason 1:
    - gene counts range over several orders of magnitude in bulk data
    - direct comparison of large numbers not ideal
    - log-normalization enables interpretation of differences as log of fold change
  - Reason 2:
    - variance of a gene's counts is proportional to its mean
    - log-normalization prevents highly expressed genes from dominating analysis
- Before UMI usage in scRNA-seq
  - single-cell data had counts ranging over several orders of magnitude  
=> log-normalization was adopted
  - UMI data have lots of zeros and do not have large counts but log-normalized

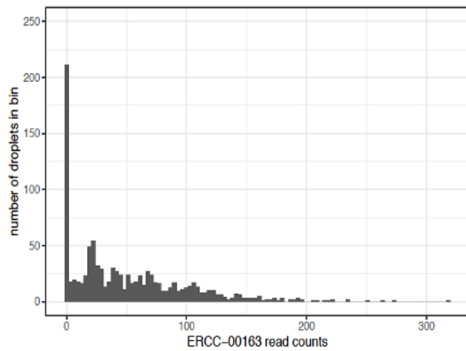
# Log-normalized counts not independent of total count

- Identical scaling for all genes => Different effects for genes with overall different expression levels
- Cells with low total UMI counts -> higher variance for high-abundance genes
- Comparison of a dataset with its down-sampled sequencing depth identified >2000 DE genes at FDR<0.01!

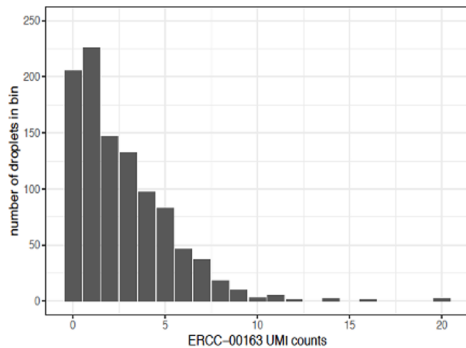


(Figures from Hafemeister and Satija, 2019)

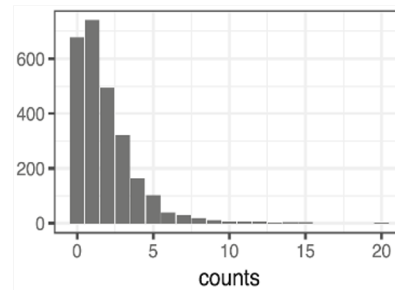
# log-transformation distorts the count distributions



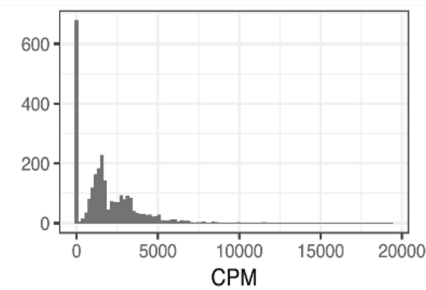
(a) Read counts



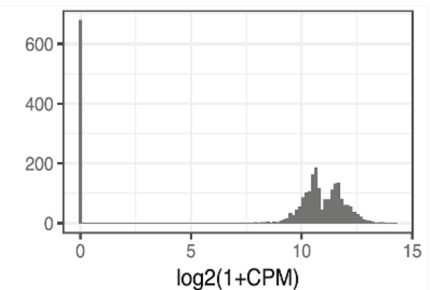
(c) UMI counts



(a) UMI counts



(b) counts per million (CPM)



(c) log of CPM

“... the log transformation is not necessary and in fact detrimental for the analysis of UMI counts” – Townes et al., 2019

# GLM designed for UMI by Seurat team inspired by bulk RNA-seq (edgeR , DESeq)

## ♦ Three steps:

1. Gene-wise GLM regression

$$\log(E(x_{ij})) = \beta_0 + \beta_1 \log_{10} m_j$$

2. Learning global trends

3. Normalization (deduct contribution of sequencing depth and scale variance)

$$z_{ij} = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}},$$

$$\mu_{ij} = \exp(\beta_{0i} + \beta_{1i} \log_{10} m_j),$$

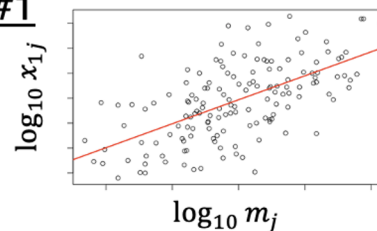
$$\sigma_{ij} = \sqrt{\mu_{ij} + \frac{\mu_{ij}^2}{\theta_i}},$$

UMI counts -> Pearson residuals

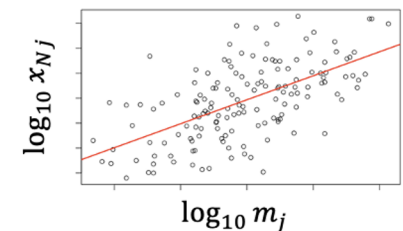
( $x_{ij} \equiv$  raw UMI counts of gene  $i$  in cell  $j$ )

( $m_j \equiv$  total UMI counts in cell  $j$ )      sequencing depth

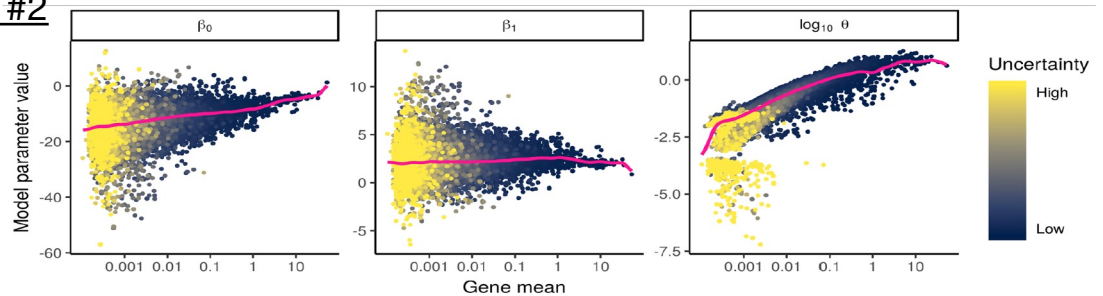
### Step #1



...

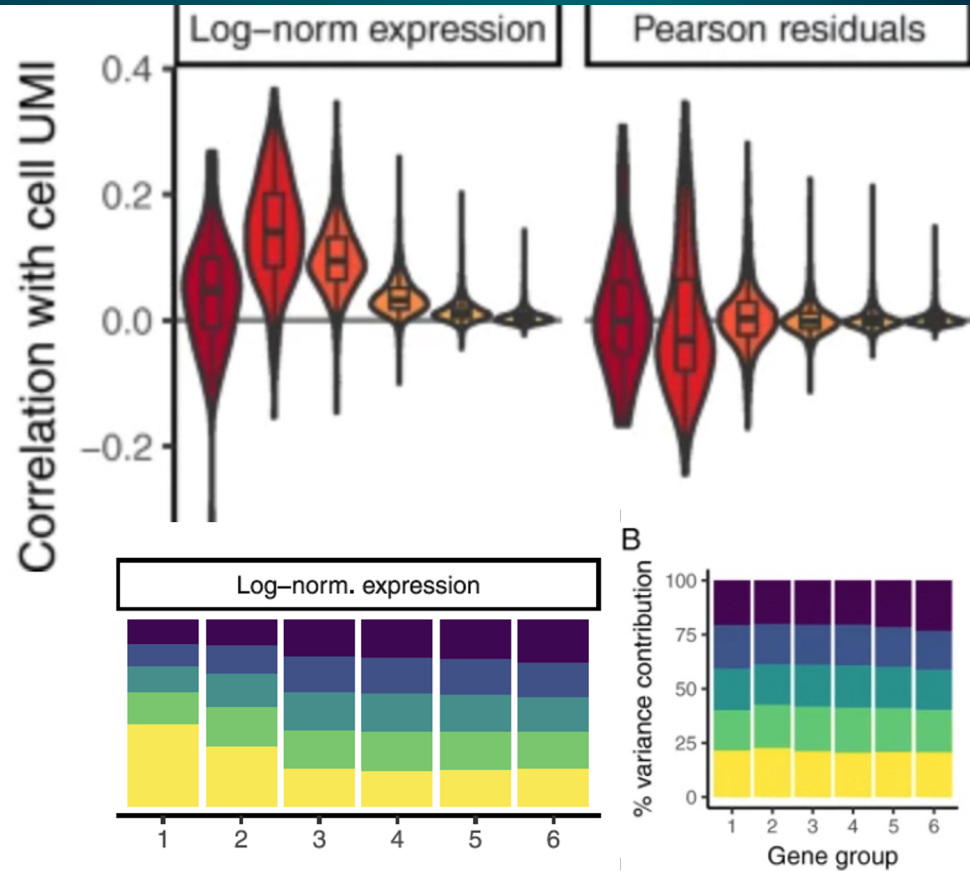
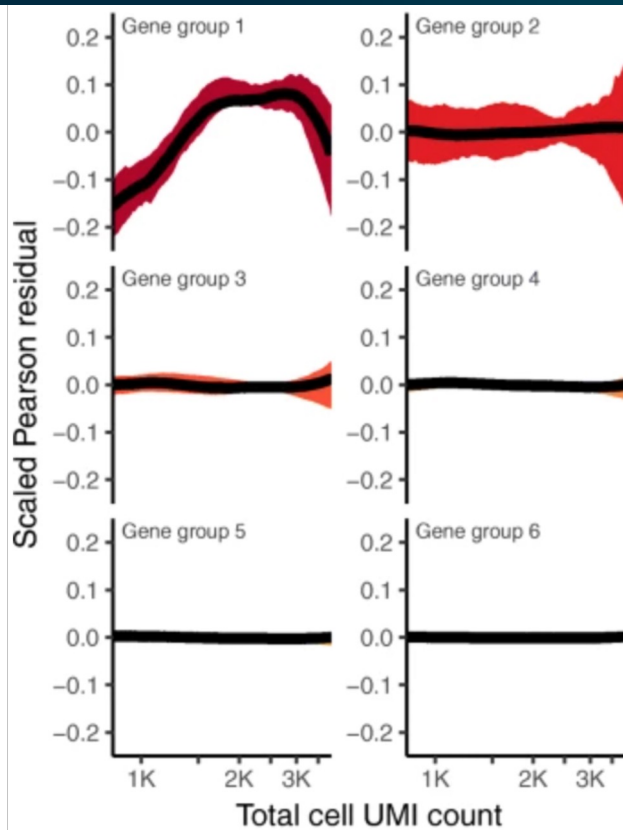


### Step #2



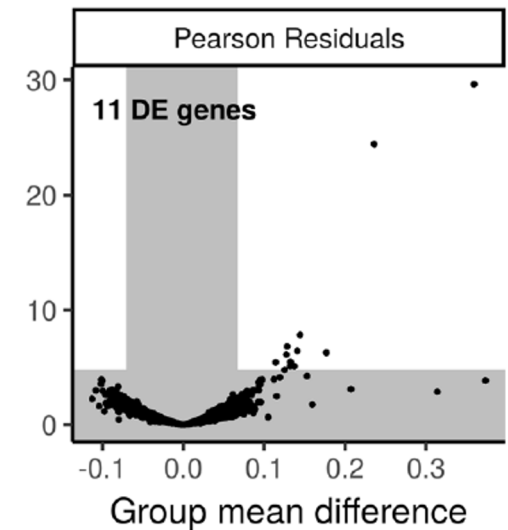
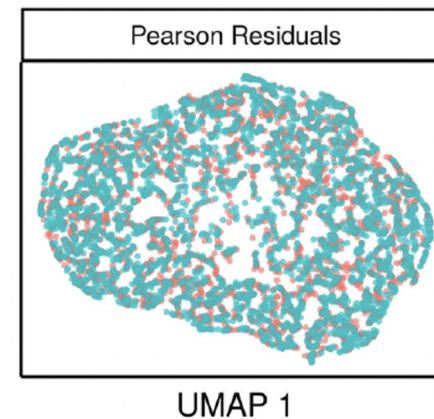
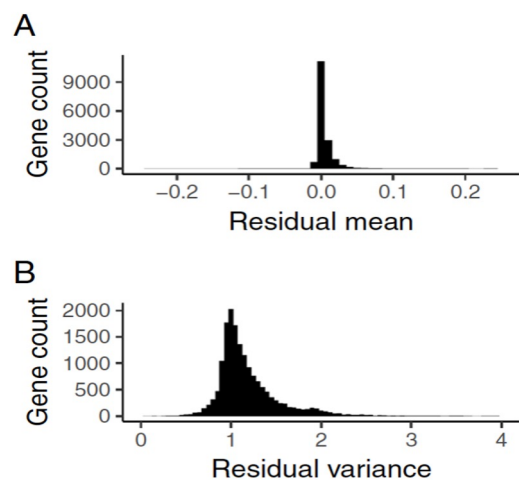
(Figure from Hafemeister and Satija, 2019)

# Pearson residuals and cellular sequencing depth



# Characteristics of *sctransform*-normalized data

- ✦ Normalized values should have mean 0 and variance 1 for non-DE genes
- ✦ Genes with variance  $\gg 1$  verified as real signal
- ✦ Comparison of a dataset with its downsampled version returned only 11 DE genes (vs >2000 genes for log-normalized data)



**sctransform** - [github.com/ChristophH/sctransform](https://github.com/ChristophH/sctransform) - **Seurat >=4** - updated version (Figure from Hafemeister and Satija, 2019)

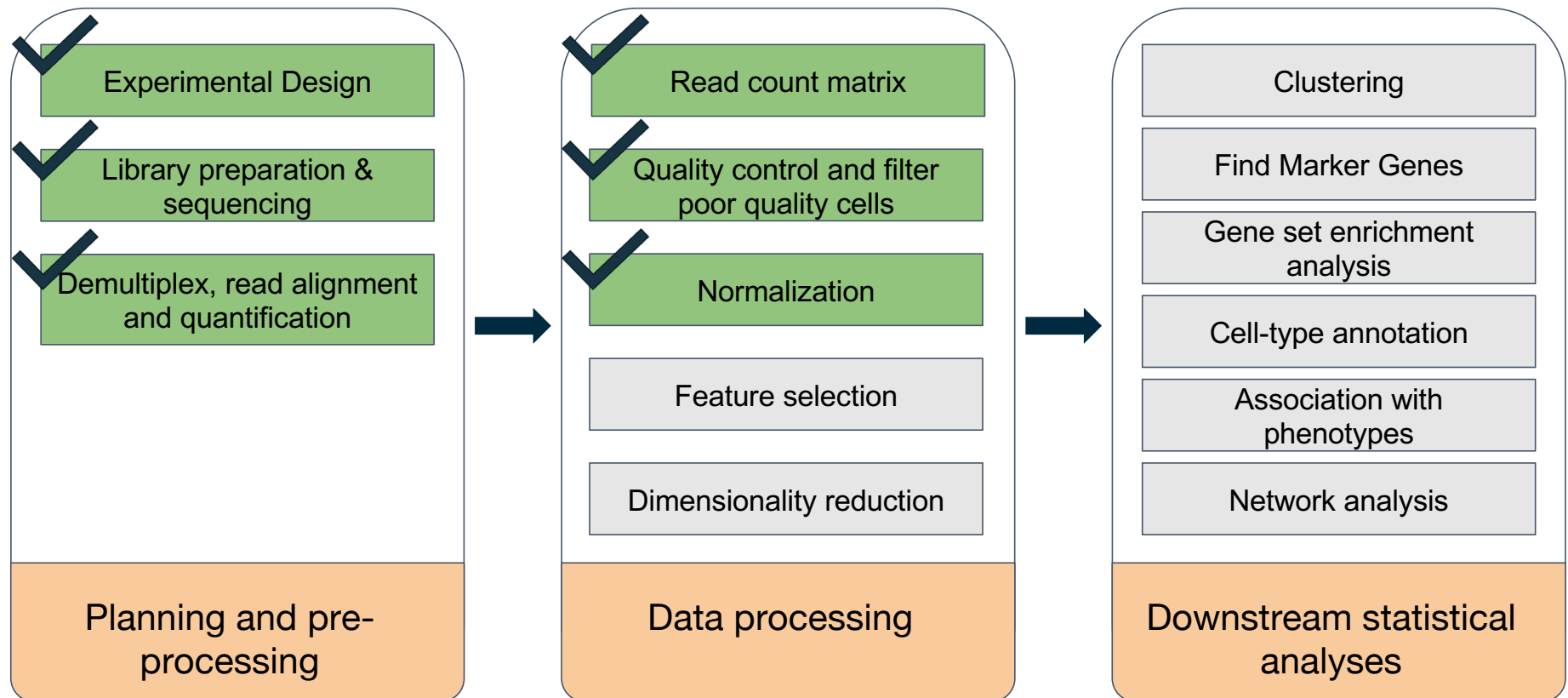
# Other methods to normalize

- Use size-factors that account for compositional differences in transcriptomes of cells
  - Total UMI counts is representative of sequencing depth if total mRNA content of all cells is identical
  - *scrn* pools cells with similar library sizes and uses the summed expression values to estimate pool-based size factors
- Normalize using spike-ins
- Sanity (SAmpling-Noise-corrected Inference of Transcription activityY)
- CLR-normalization in Seurat for CITE-seq (see references)

**Session 3: Advanced discussion on normalization, differential analysis, and batch-correction. (Tomorrow)**



# scRNA-seq workflow



# 7. Feature selection

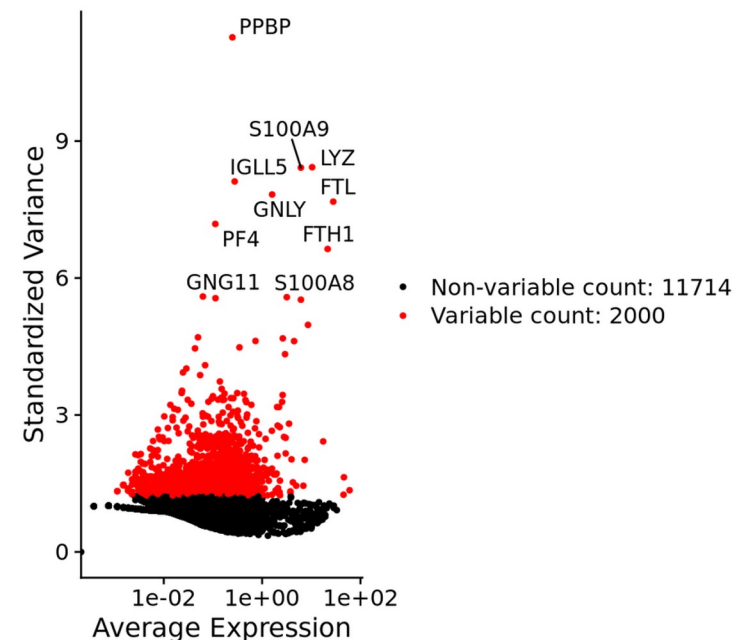
# Why work with select features (genes)?

- In a typical study, most genes may not be relevant
- Many genes are biologically variable only across different tissues.  
=> Variation in most genes is technical or inherent biological noise
- Feature selection may improve signal-to-noise ratio
- Feature selection may improve computational efficiency

**Excluding uninformative genes such as those which exhibit no meaningful biological variation across samples.**

# Most common approach currently: To select highly variable genes

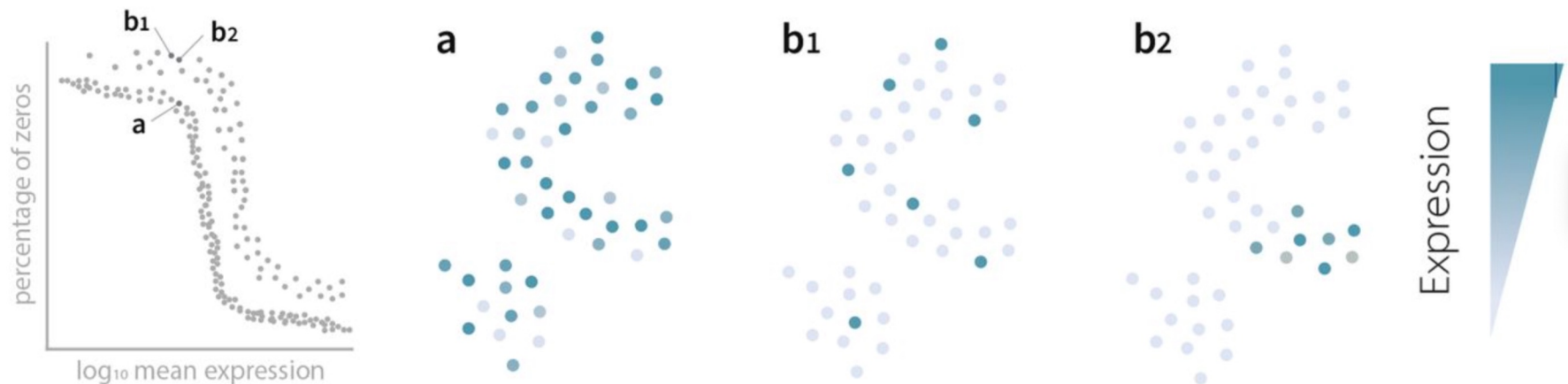
- Data must be properly normalized before selection
- **Standard workflow of Seurat** uses log-normalization followed by a variance stabilizing transform to account for mean-variance relationship - ranking for selection
- **SCTransform** normalizes and selects highly variable genes **in one step**
- May miss genes relevant only to rare cell types



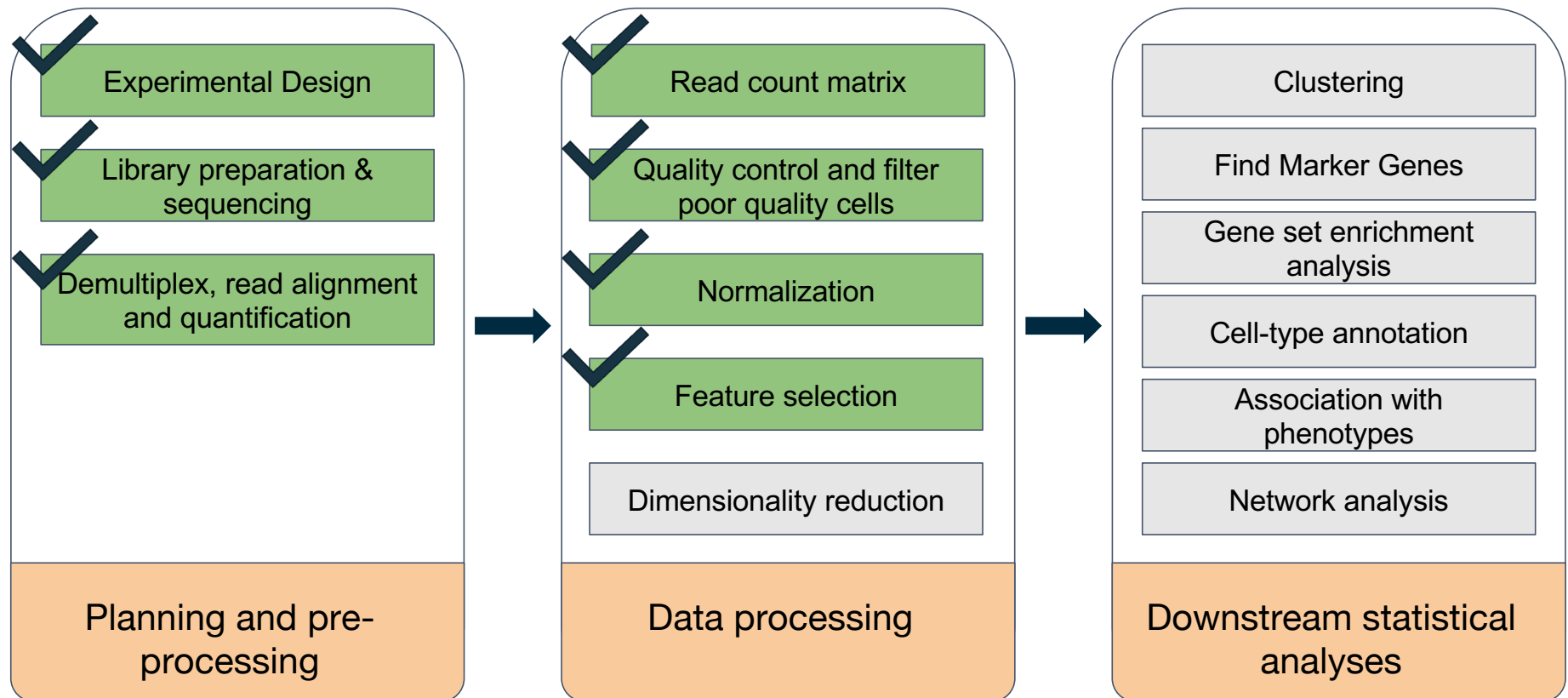
(Figure from the Satija lab website)

# Other approaches

- Select highly *deviant* genes from null (uninformative) (Townes et al., 2019)
- Select genes with high *dropout* rates (Andrews and Hemberg, 2019)
- Triku - selects genes that show an unexpected distribution of zero counts and whose expression is localized in cells that are transcriptomically similar



# scRNA-seq workflow



## 8. Dimensionality reduction

# Motivation

scRNA experiments aim to separate cell types  
exploiting the expression levels **across multiple  
highly variable genes** that maximize the cell-to-cell  
variation



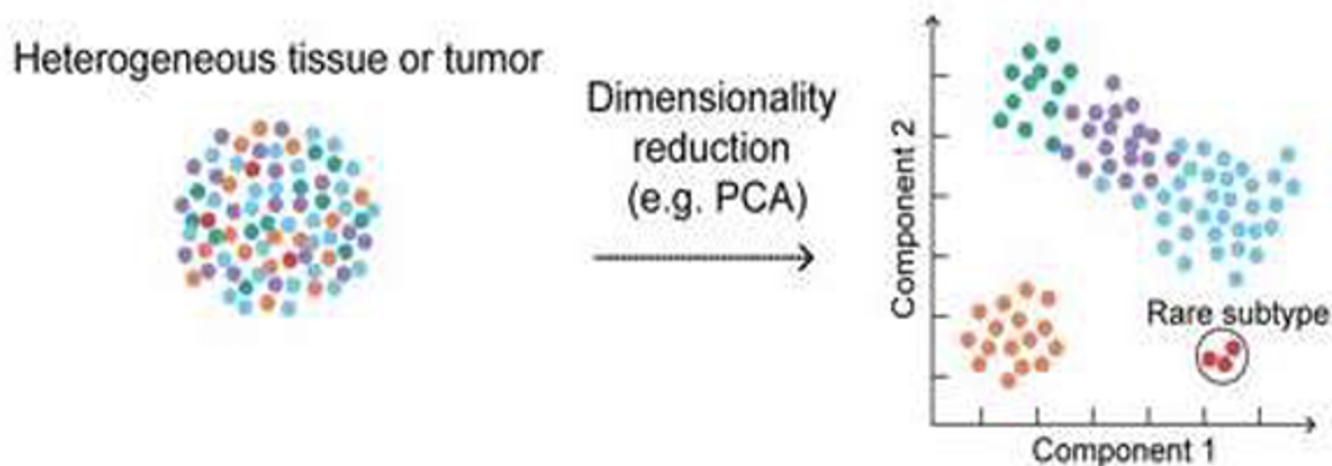
```
graph TD; A[variation] --> B[high complexity]; A --> C[visualization];
```

high complexity

visualization



# Dimensionality reduction for single-cell analysis

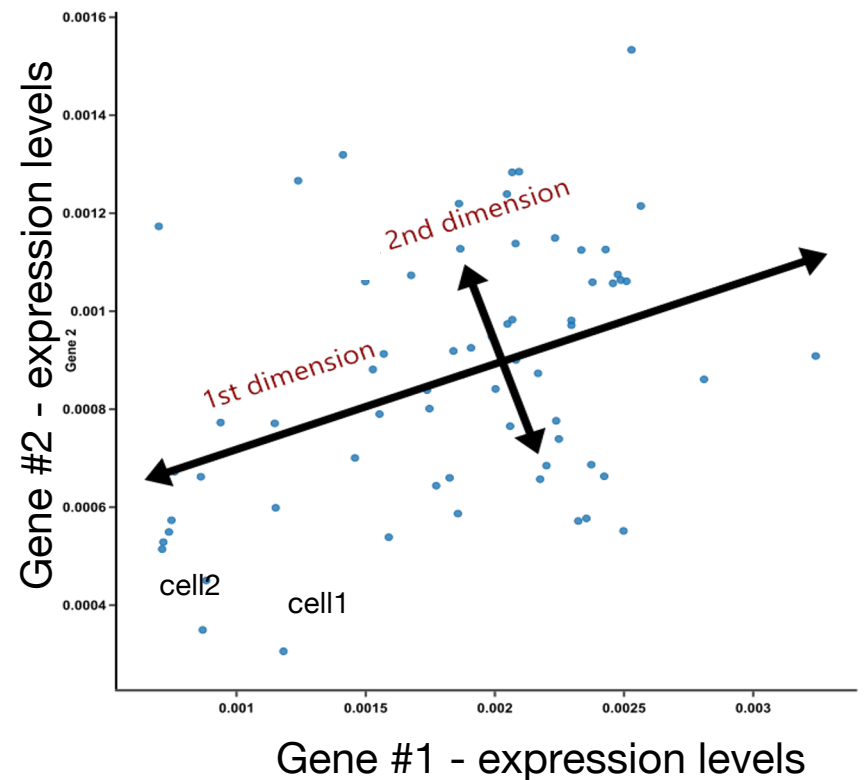


Each single cell involves huge numbers of features (each gene expression is one dimension)

Project  $n$ -dimensional data onto a  $k$ -dimensional space ( $k \ll n$ ) to reduce noise and help with data explorations

# How do we visualize multiple genes?

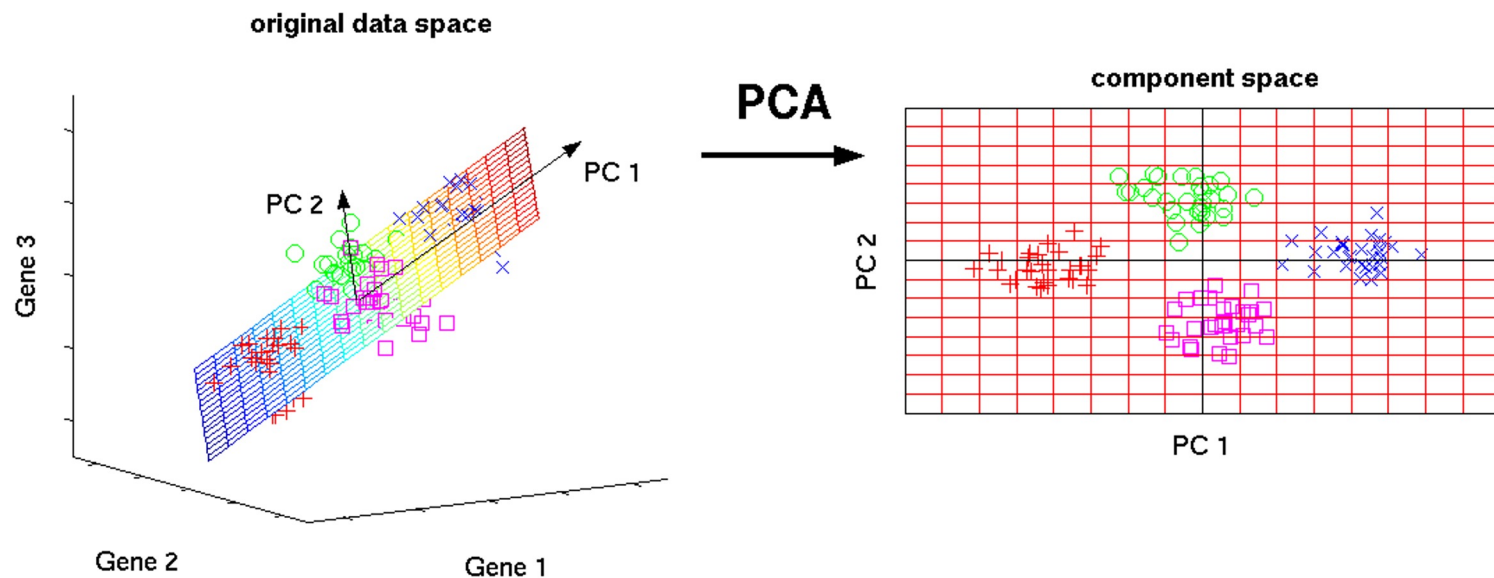
- Biological phenomena more likely driven by sets of genes than individual genes
- Dimensions = Number of coordinates of data points = Number of genes  
=> Too many to visualize
- Cells may be separated in the direction of hidden “factors”  
=> 1<sup>st</sup> dimension: disease-associated pathway  
=> 2<sup>nd</sup> dimension another hidden factor, e.g., batch effect



# Data embedding

- Map existing features into smaller number of new features
- Linear combination methods (projection), e.g, Principal Component Analysis (PCA), Multidimensional Scaling (MDS)
- Nonlinear combination methods, e.g., t-SNE, UMAP

# PCA finds directions with largest amount of variation



Orthogonal transformation of the original dataset -> new uncorrelated variables or principal components

PC1 is the projection direction that maximizes the variance of the projected data

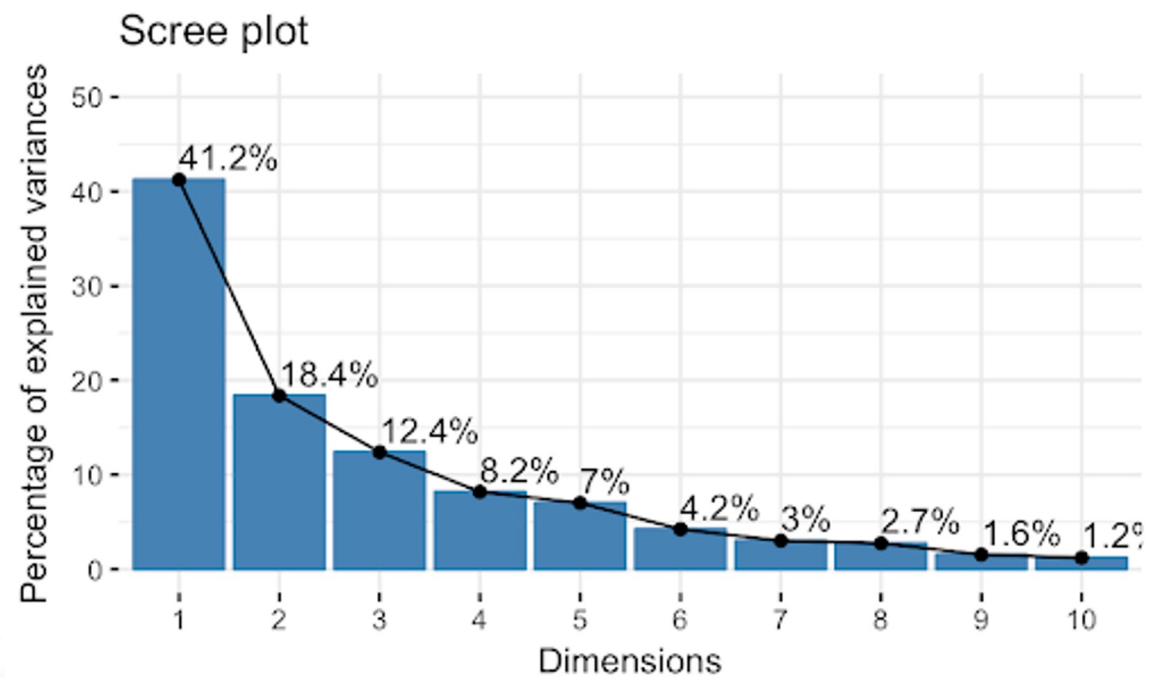
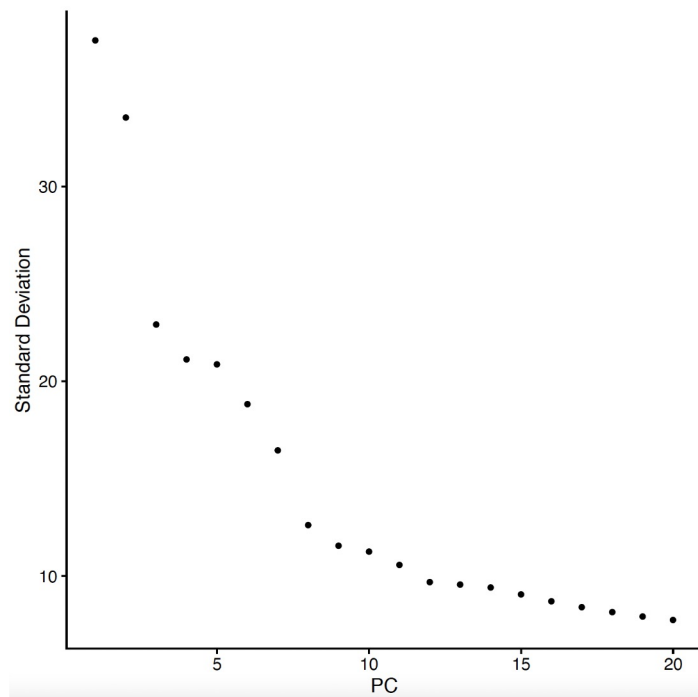
PC2 is the projection direction that is orthogonal to PC1 and maximizes variance of the projected data

# PCA approximate the information from relevant genes

- Linear combination of original data that re-express the original information
- Dimensions associated with only the “relevant” factors
  - => Useful approximation of complete data
  - => Computational efficiency improved
  - => Signal-to-noise ratio increased
- Consider exploratory visualizations to determine how many PCs (elbow plots, etc)

Many algorithms to perform PCA (for benchmarking comparisons, Tsuyuzaki et al., 2020)

# PCA: how many components?

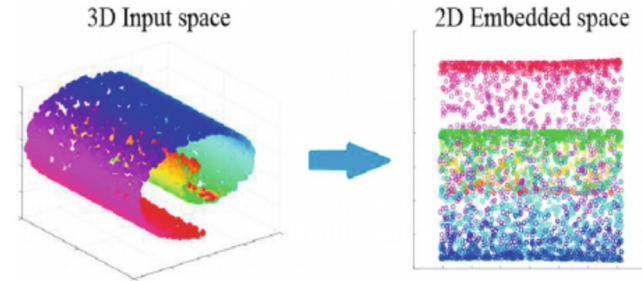


- Common to work with top ~10-15 PCs for downstream clustering

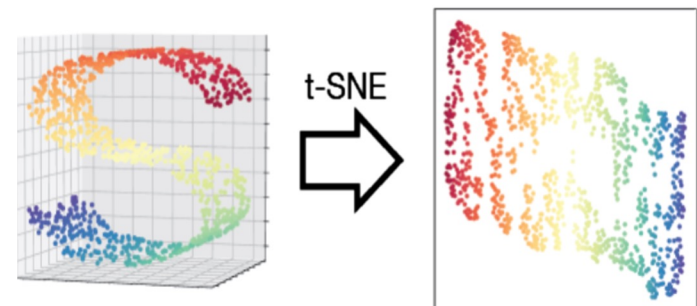
# Visualization of single-cell gene expression data commonly utilizes nonlinear dimensionality reduction

- PCA denoises data but may not meet requirements for visualization
- Cells may lie along nonlinear low dimensional surface (manifolds)  
⇒ linear projection may mix up different cell types
- How to visualize depends on the goal of visualization
  - Goal is view dimensions with largest variance  
⇒ PCA may be good
  - Goal is to preserve local structure  
⇒ nonlinear may be projection required

Mixing of cell types on linear projection



t-SNE preserves local structures



(Images from Salazar-Castro et al., 2018 (top) and Cooley et al, 2020 (bottom))

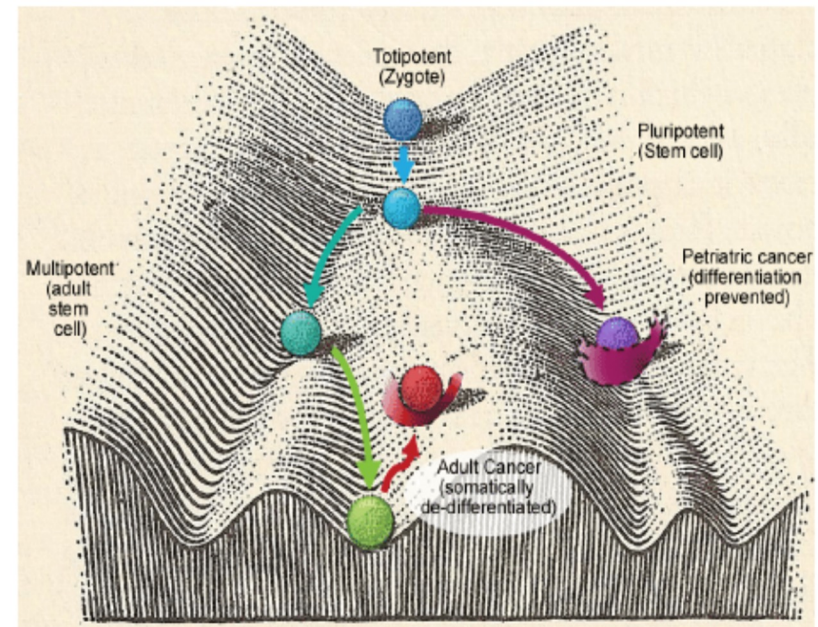
# Nonlinear dynamics are commonplace in biological systems

- Gene regulatory networks contain feedback loops
- Nonlinear protein-protein and protein-DNA interactions
- Cells are transitioning from one state to another along uneven landscapes



Human B Cell Development

Waddington landscape

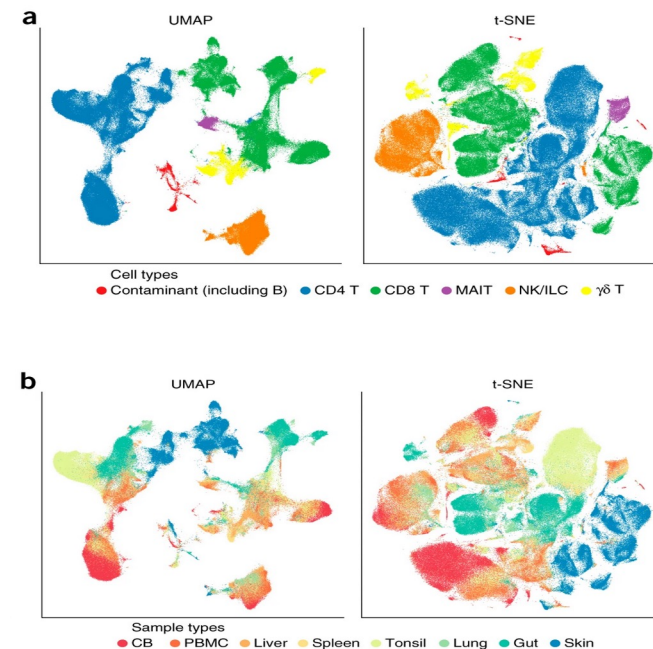


(Image from Fels et al., 2015 and Bendall et al., 2014)



# Dimensionality reduction: FAQs

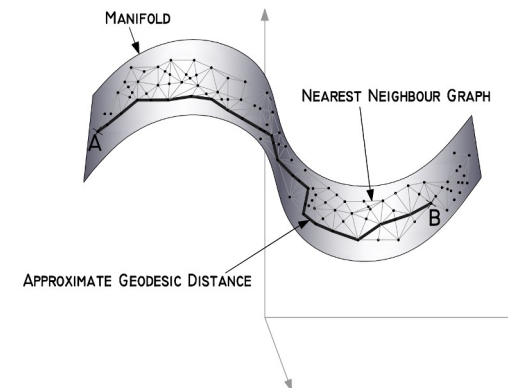
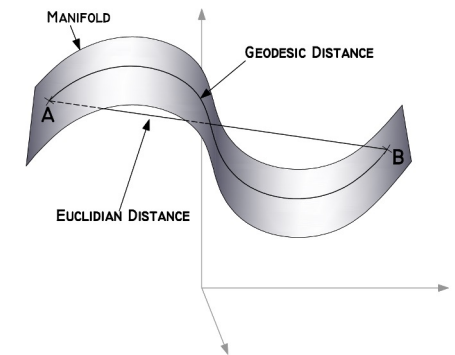
- PCA vs tSNE vs UMAP?
- Is it art or science or both?
- Why the output from yesterday doesn't match the one from today?
- ...



(Image source: Becht et al, 2018)

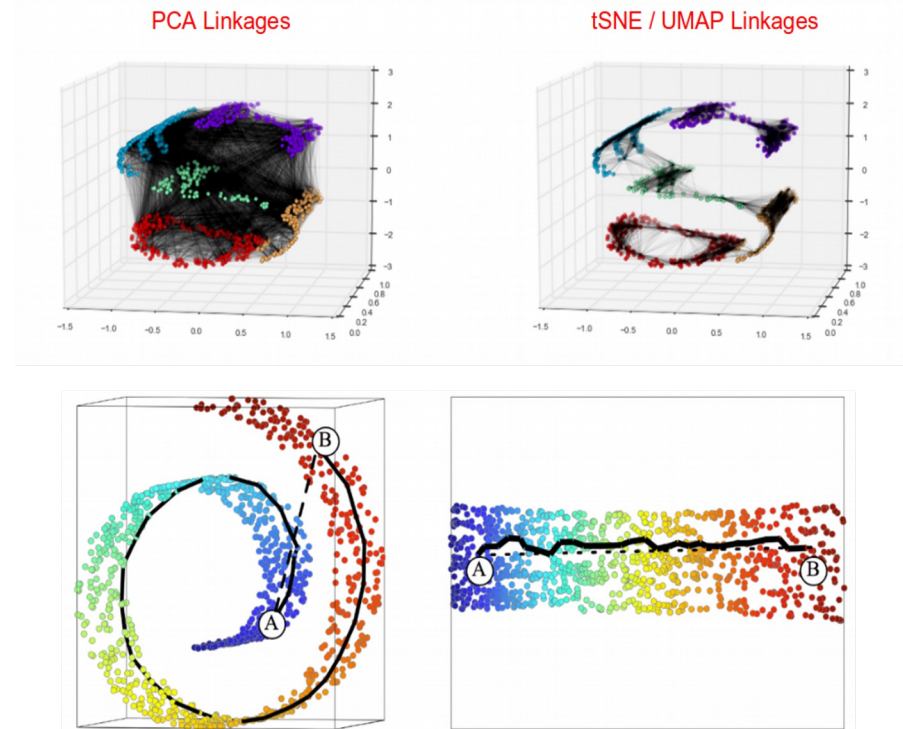
# Principle underlying t-SNE and UMAP

- Given normalized counts, Euclidean distance gives the straight line distance
- Transform Euclidean distances into “probability of being neighbor”
- Initialize 2D or 3D embedding - gradient descent
- Optimize embedding to best preserve relative distances (after transformation) - local structures in low dimensions
- Various algorithms available, e.g., SNE, t-SNE (t-Distributed Stochastic Neighbor Embedding), UMAP (Uniform Manifold Approximation and Projection), etc



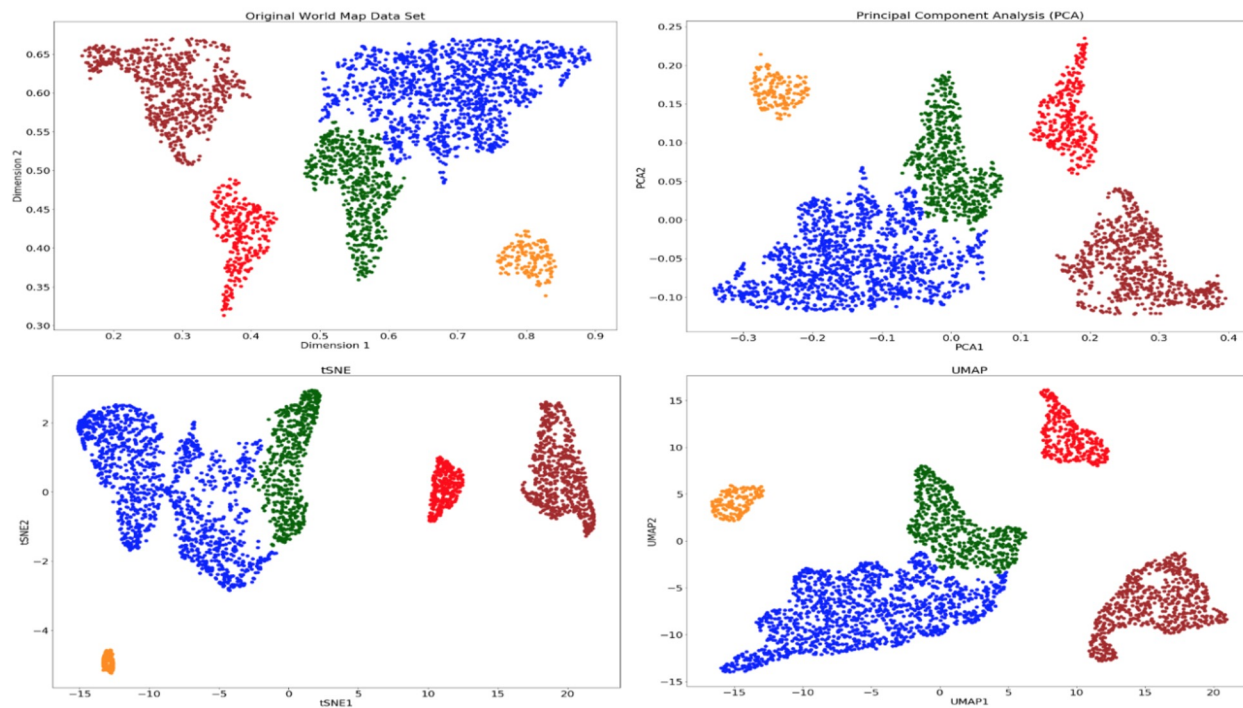
# t-SNE and UMAP prioritize preservation of distances between neighbors

- PCA considers all pairwise distances equally
- t-SNE and UMAP prioritize distances between neighbors



(Image from blog by Nikolay Oskolkov on [towardsdatascience.com](https://towardsdatascience.com))

# PCA / t-SNE / UMAP: 2D world map



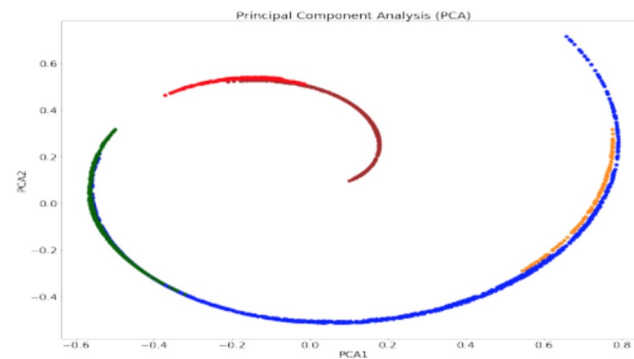
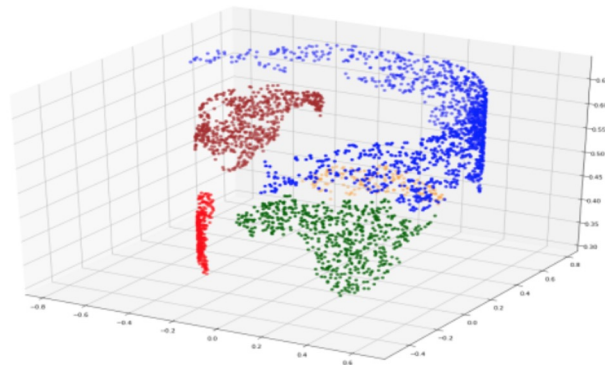
PCA

UMAP

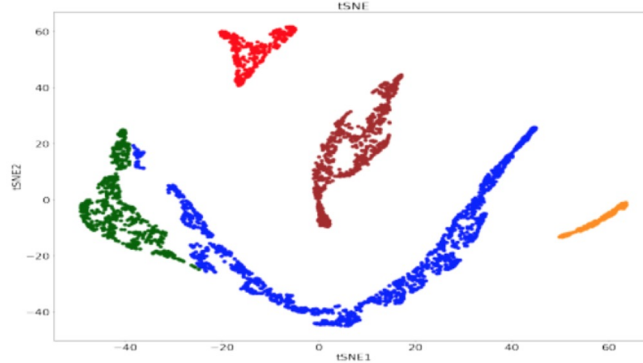
t-SNE

Reconstruction of the World Map by PCA, tSNE (perplexity = 500) and UMAP (n\_neighbor = 500)

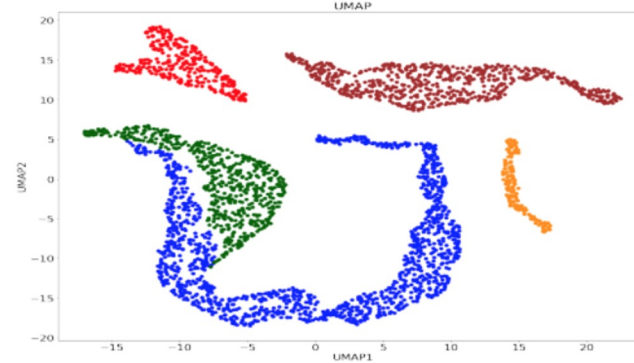
# PCA / t-SNE / UMAP: 3D world map (Swiss roll)



PCA

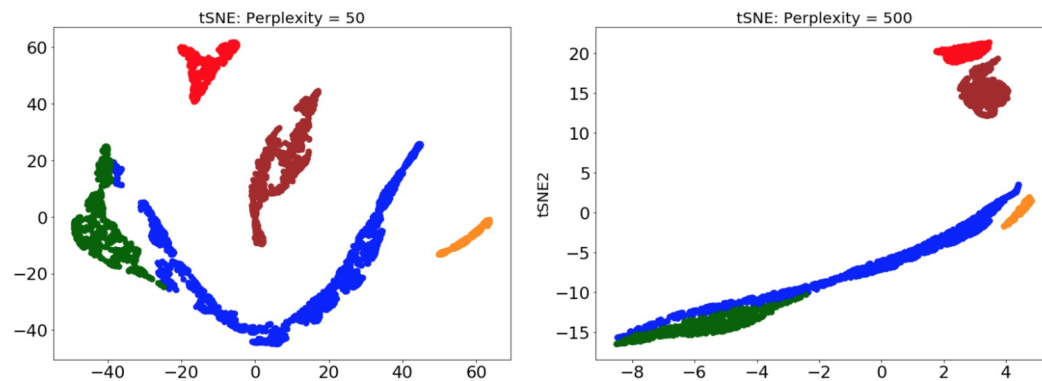


t-SNE



UMAP

# t-SNE: changing parameters (perplexity)



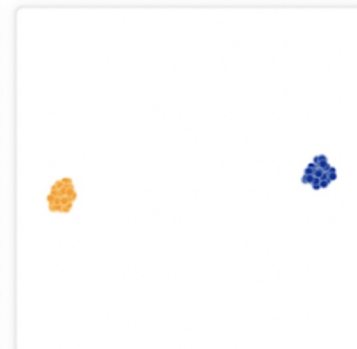
*Original*



Perplexity: 2



Perplexity: 5

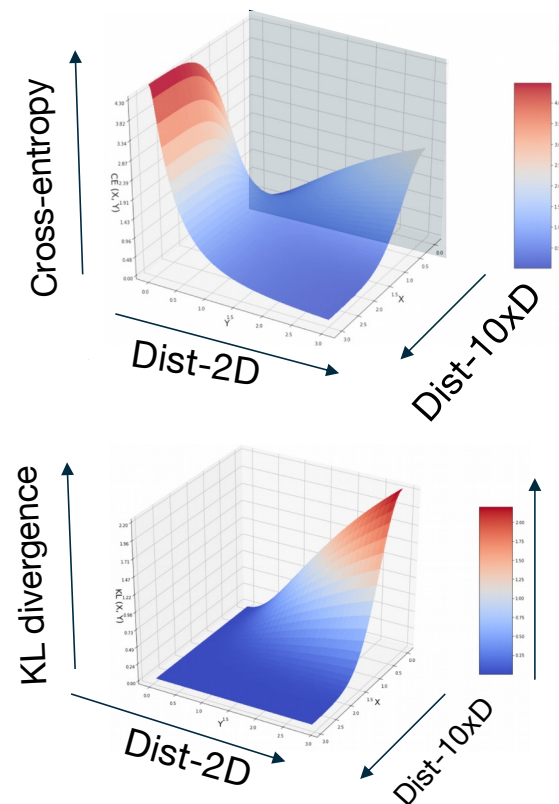


Perplexity: 30

# UMAP is better at preserving global structure than t-SNE (?)

- ✦ t-SNE and UMAP share the same philosophy but differ in design
- ✦ UMAP uses cross-entropy as cost function while t-SNE uses KL divergence
  - => UMAP focuses on both local and global structure
  - => t-SNE only preserves local structure

**Choice of cost function by UMAP** ensures that points far away from each other in high dimensions will remain far away in low dimensions.

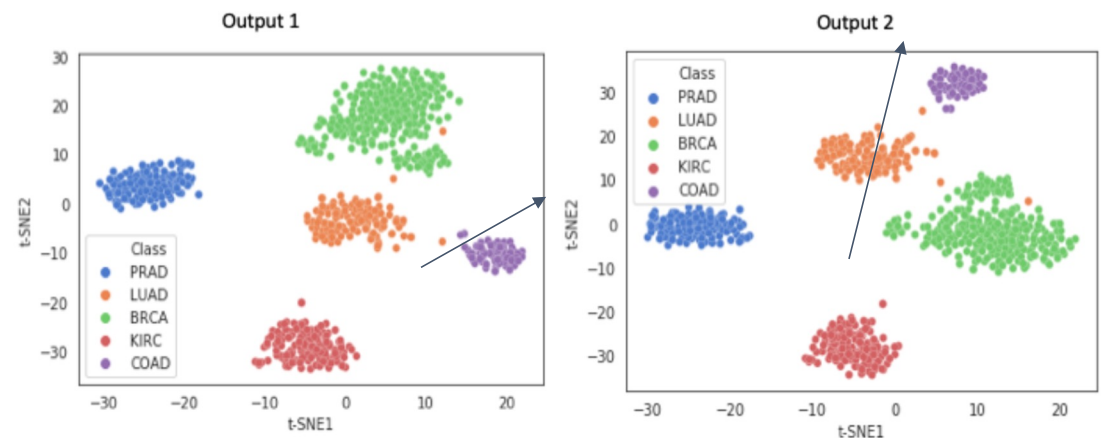


(Image from blog by Nikolay Oskolkov on [towardsdatascience.com](https://towardsdatascience.com))



# t-SNE vs UMAP and choice of parameter values

- Kobak and Berens, 2019
- Default UMAP is not necessarily better than t-SNE at preserving global structure
- Both depend on hyperparameters and initialization - randomness
- t-SNE is a stochastic method, there is a random initialisation involved and it doesn't produce similar outputs on successive runs



(Image from blog by Nikolay Oskolkov on [towardsdatascience.com](https://towardsdatascience.com))

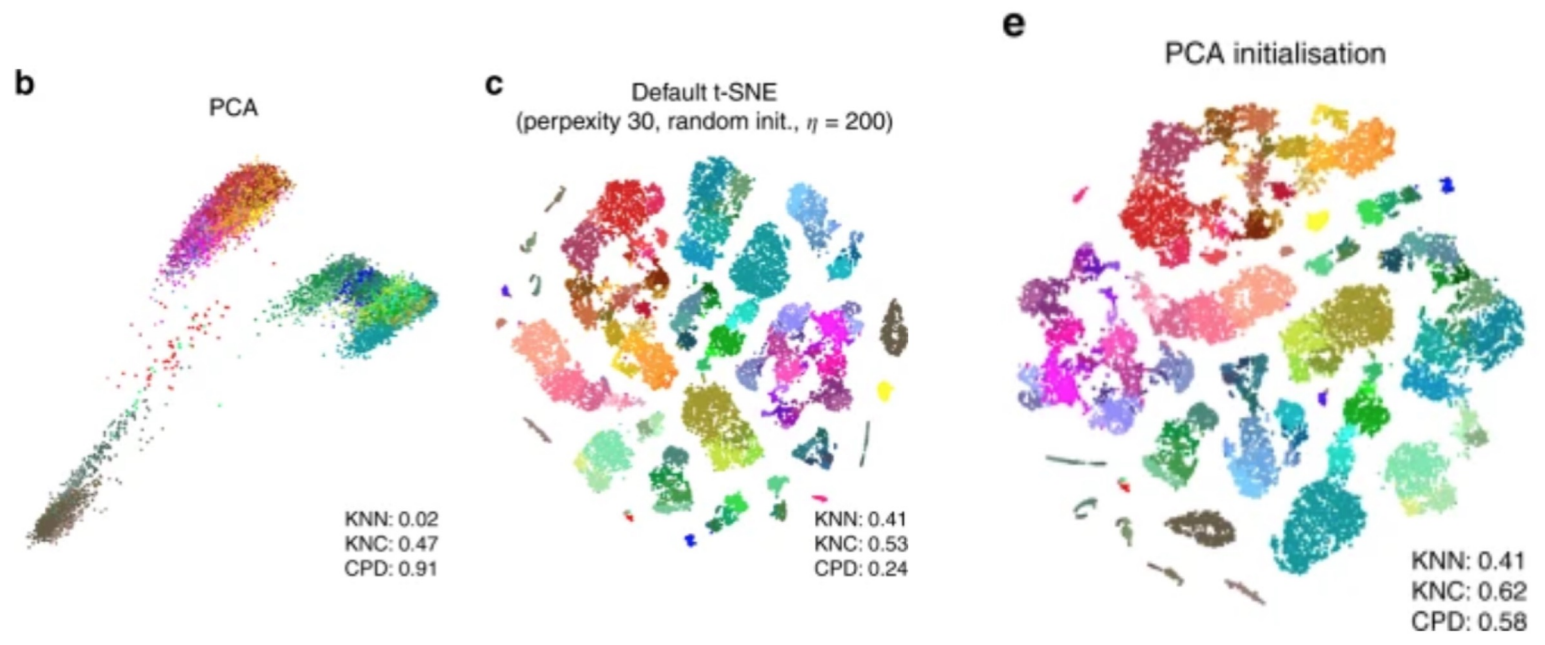


# UMAP advantages and limitations

- + Non linear dataset
- + Computational efficiency
- + Global structure preservation - control the balance between local and global structures
- + Downstream applications
- The lower dimension embeddings of UMAP lack strong interpretability
- UMAP can tend to find manifold structure within the noise of a dataset, better with large datasets
- Accuracy of global structure

(Image from blog by Nikolay Oskolkov on [towardsdatascience.com](https://towardsdatascience.com))

# Recommendation 1: Consider PCA initialization for t-SNE and UMAP



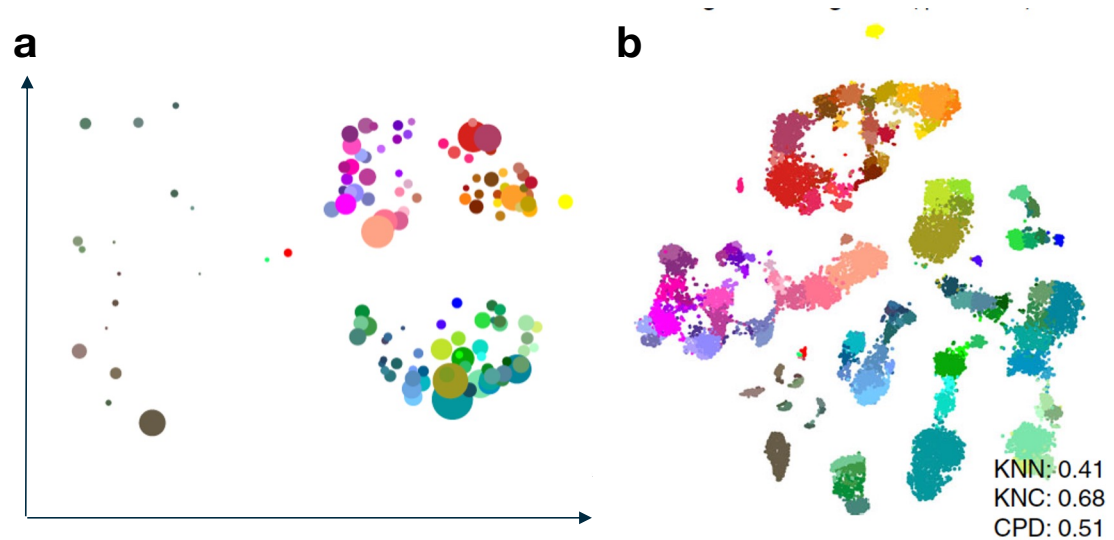
- Should save the seed used to initialize the pseudo-random number generator
- Should save the output after running analysis

(Images from Kobak and Berens, 2019)

Recommendation 2: Consider using an MDS plot of class means in conjunction with t-SNE or UMAP plots (Kobak et al, 2019)

a. Multi Dimensional Reduction of class means highlights the global structure

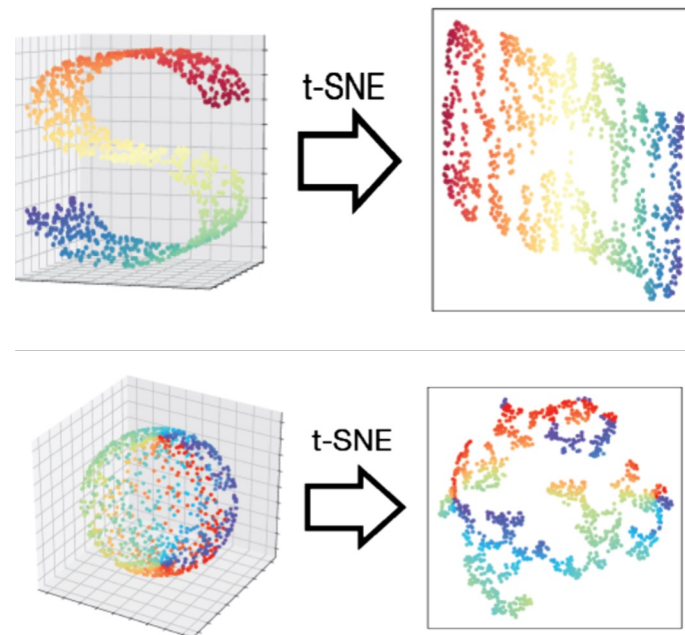
a. t-SNE or UMAP highlights the local structures



(Images from Kobak and Berens, 2019)

## Recommendation 3: Consider quantifying distortion in low-dimensional embedding

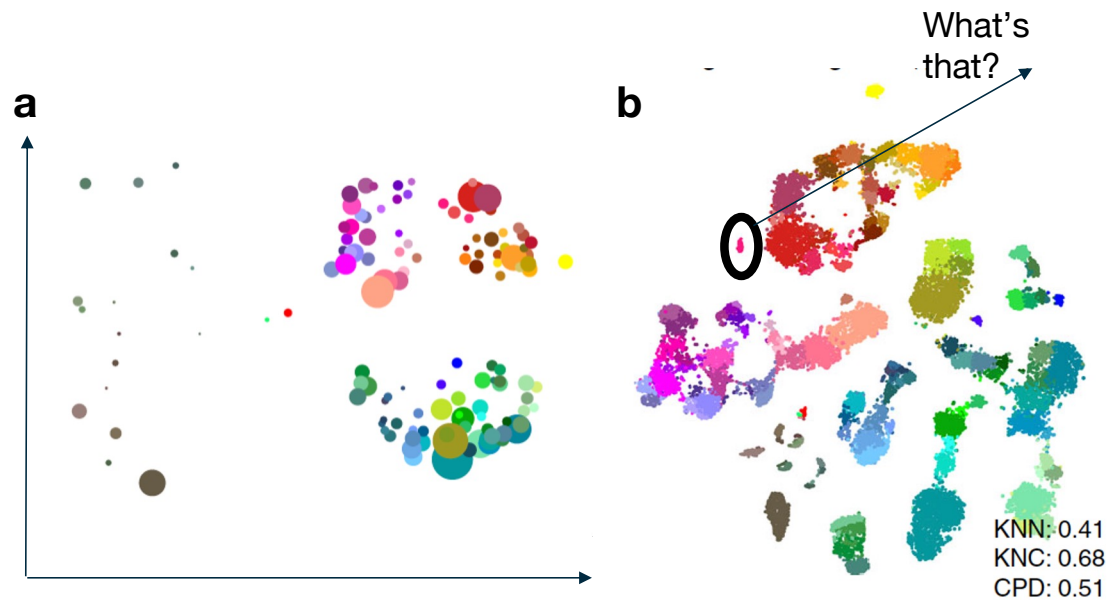
- Consider quantifying distortion
- comparing the local neighborhoods of points before and after dimensionality reduction
- => clustering not done in t-SNE space



(Image from Cooley et al, 2020)

# Patterns that we see in low-dimensional embedding can be new biology or artifacts

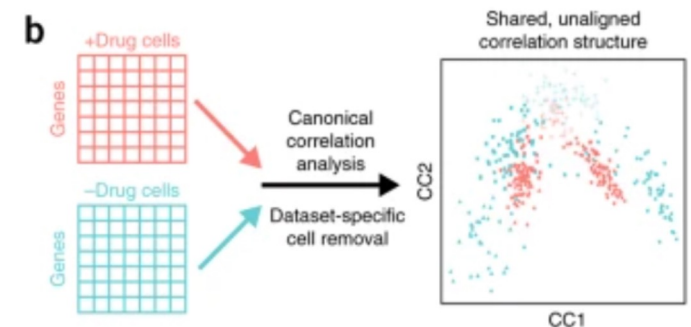
- Rare cell type?
- Distortion?
- Doublets?
- Dead cells?
- Empty drops?
- ...



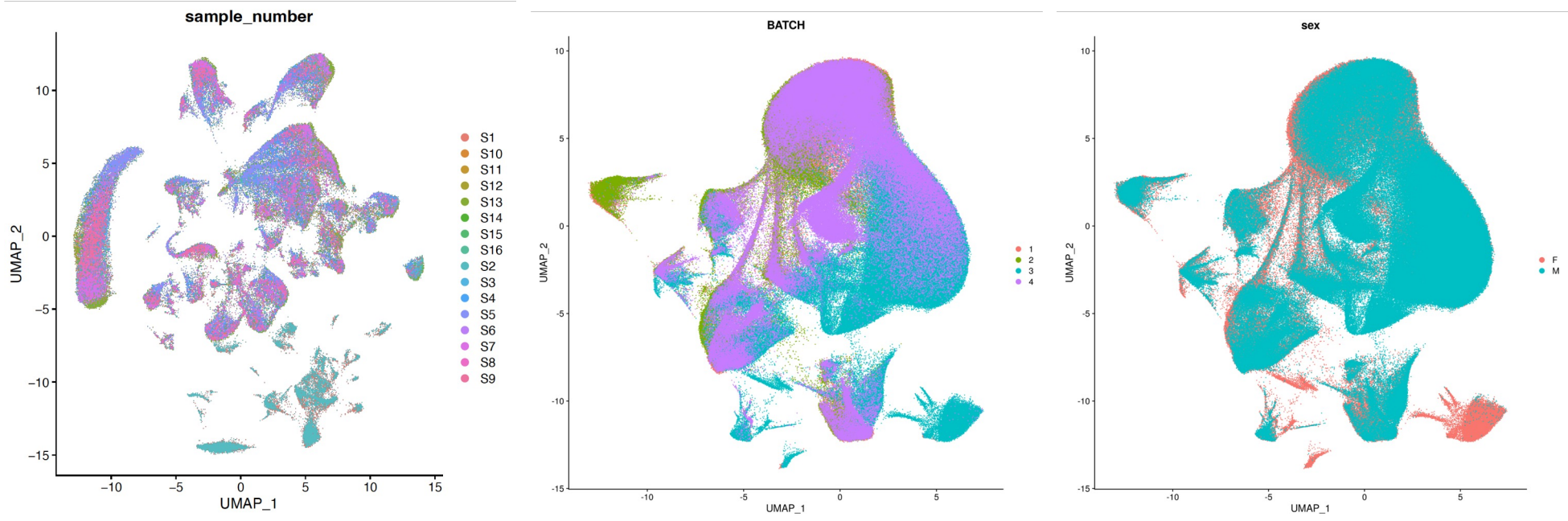
(Images from Kobak and Berens, 2019)

# Dimensionality reduction of scRNA-seq data is an active area of research

- GLM-PCA by Townes et al., 2019
  - log normalized data and feature selection by highly variable genes -> false variability in dimension reduction.
  - Simple multinomial methods for non-normal distributions and feature selection using deviance
- Ge et al., 2020 report a method that can handle batch effects
- CCA (Canonical correlation analysis) - shared gene–gene correlations across multi data sets



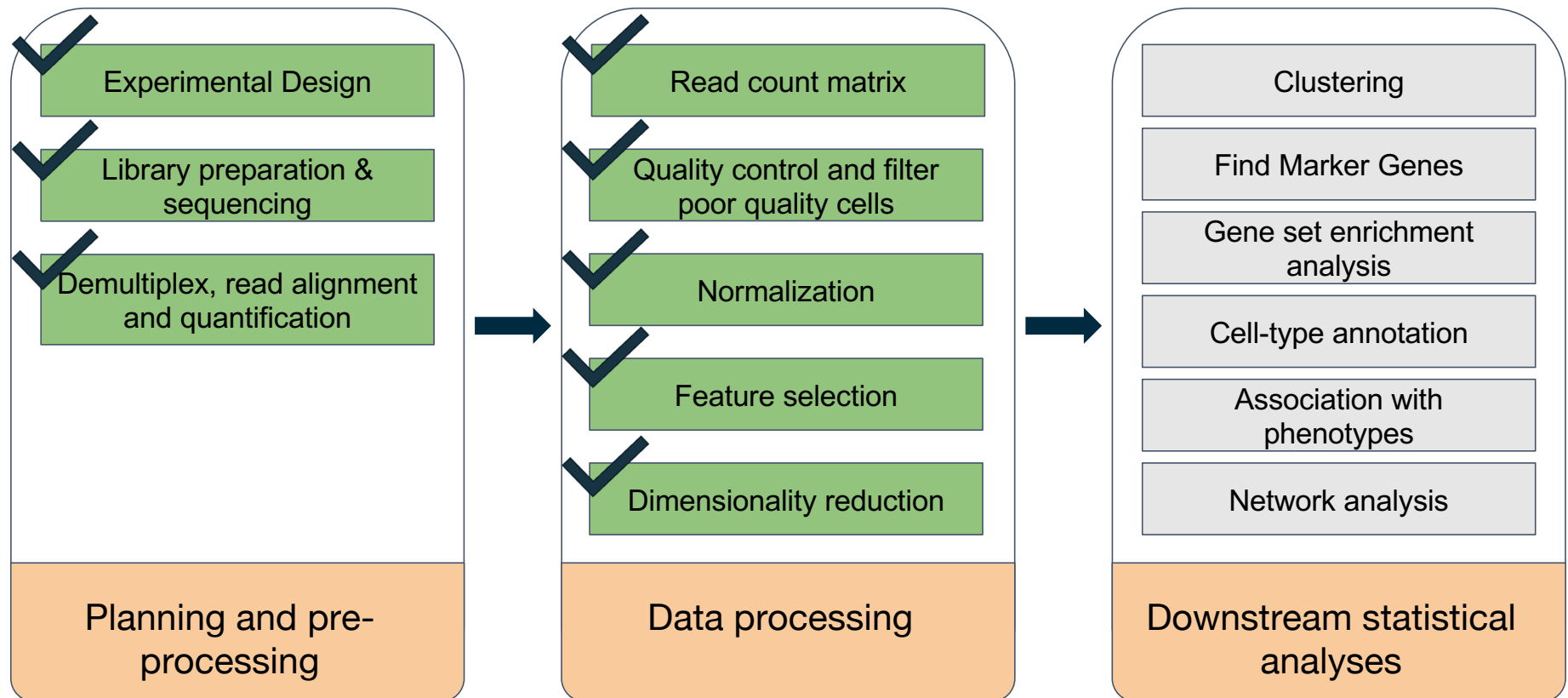
# Visualization of batch effects in scRNA-seq data using the 2D embedding



Use the metadata collected during the experiment to visualize potential batch effects!

**Session 3: advanced discussion on batch-correction.**

# scRNA-seq workflow





# To summarize where we are...

- Using scRNA-seq we want to quantify for gene expression changes in particular cell-types or in the differences of the proportion of cell-types between conditions
- Biology complex!  $\sim O(10,000)$  genes and  $O(\sim 1000)$  cells
- We are trying to understand the system empirically
- There is a hidden logic behind the changes we observe - this is what we are trying to uncover
  - Not all genes act independently and not all cells distinct in terms of their functional significance and consequence
- Our goals are to be able quantify the changes in an unbiased and in an efficient manner

# Unbiased quantification of changes

- Filter out dying cells, doublets
- Perform batch correction of read counts assayed by different individuals, or at different times or with different reagents

# Efficient quantification of changes

- Normalize the read counts between cells to account for differences in sequencing depth between cells
  - Variance of gene expression should be independent of sequencing depth
- We focus on the most variable genes to minimize “noise” associated with genes that may not play a role in defining
  - Cell type
  - Visualizing the HIGH dimensional data
- Differential expression performed on all genes on the identified clusters/cell-types

# Session 2

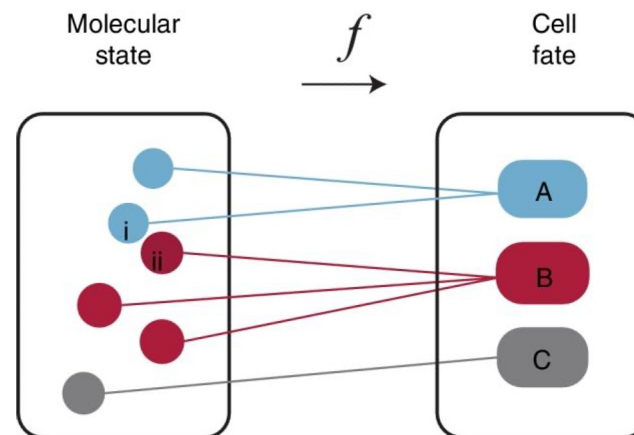
# Topics for Session 2

- Clustering
- Find marker genes
- Demo

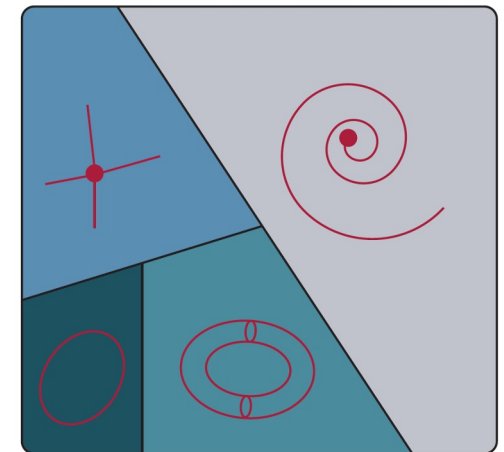
# 9. Clustering

# Clusters of cells represent cell types

- Cells are limited to basin of attraction of their attractor states  
  
=> Cells of the same type form clusters in gene expression space



Different expression state space map to the same fates

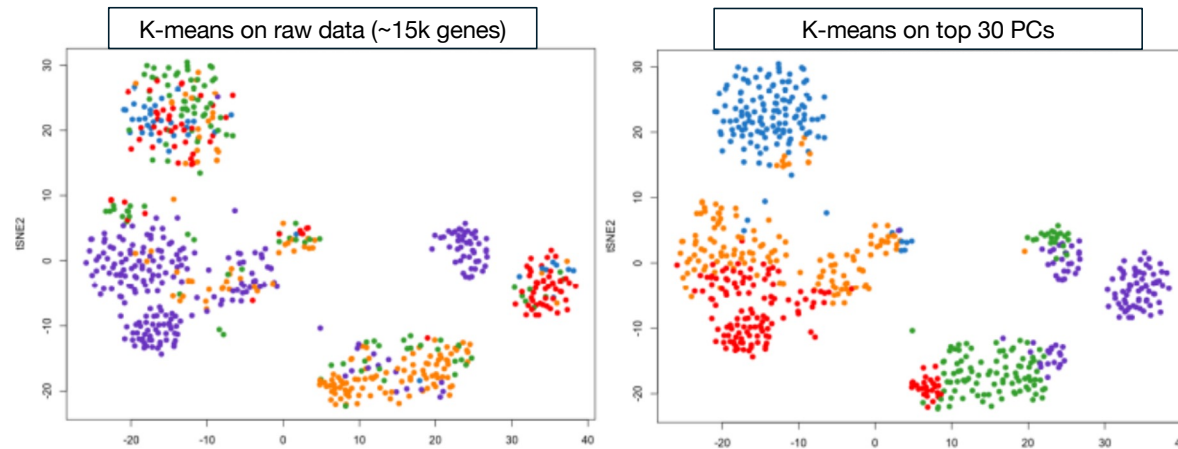


(Image from Casey et al, 2020)

# Commonly used clustering algorithms may not be appropriate

- Assumptions in common algorithms typically do not apply to scRNA-seq
- Ex: k-means generates clusters of similar sizes and ~ spherical in shape
  - but basins of attraction can have different sizes and complex geometries

Nearby points in t-SNE plot are not always in the same cluster from k-means



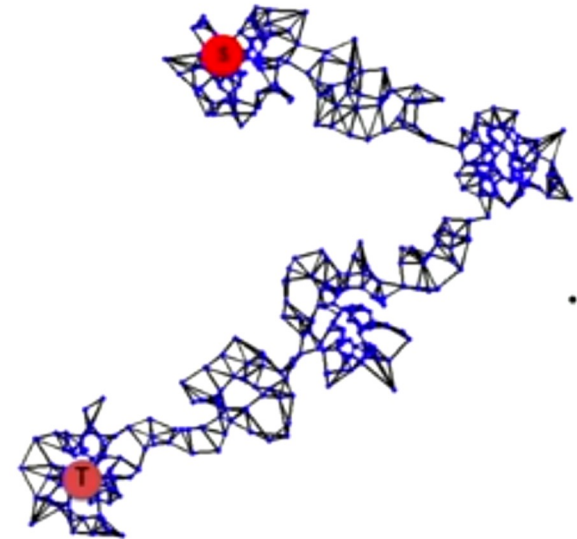
(Image from blog by Nikolay Oskolkov on [towardsdatascience.com](https://towardsdatascience.com))



# Seurat finds clusters in two steps:

## 1. Build shared nearest neighbor graph

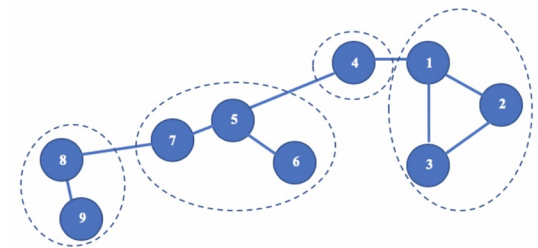
- Build a k-nearest neighbor graph (KNN)
- Keep only edges between cells that share a neighbor because similar feature expression patterns
- Adjust the edge weights between any two cells based on similarity  
= Shared nearest neighbor graph
- Use #PCs as input



## Seurat finds clusters in two steps:

### 2. Community detection (e.g., Louvain algorithm)

- "community" = a cluster of cells that are more connected among themselves than with cells of other communities
- graph-based clustering detects clusters of arbitrary structures
- optimizes modularity - **resolution**
  - Large modularity => most edges are within clusters
  - Resolution between 0.4-1.2 typically returns good results for 3K single-cell datasets
- Steps:
  1. Initialize: Every cell is its own community
  2. Take cell from its community and add to another
  3. Evaluate modularity gain
  4. Place cell in community with highest modularity gain
  5. Do for all cells and repeat until no improvement



(Image from blog on medium.com)

# Graph-based methods alternatives

- Leiden algorithm
- Walktrap
- Label propagation
- Fast-greedy
- ...

# Challenges:

- Control resolution parameter in FindClusters to tune number of clusters -
- Without a gold-standard this process of identifying cell-types in snRNA-seq turns out to be quite subjective
  - One can define as many cell-types as one wants
- Cells belonging to the same cluster should co-localize on the t-SNE and UMAP plots
- Cell types annotation

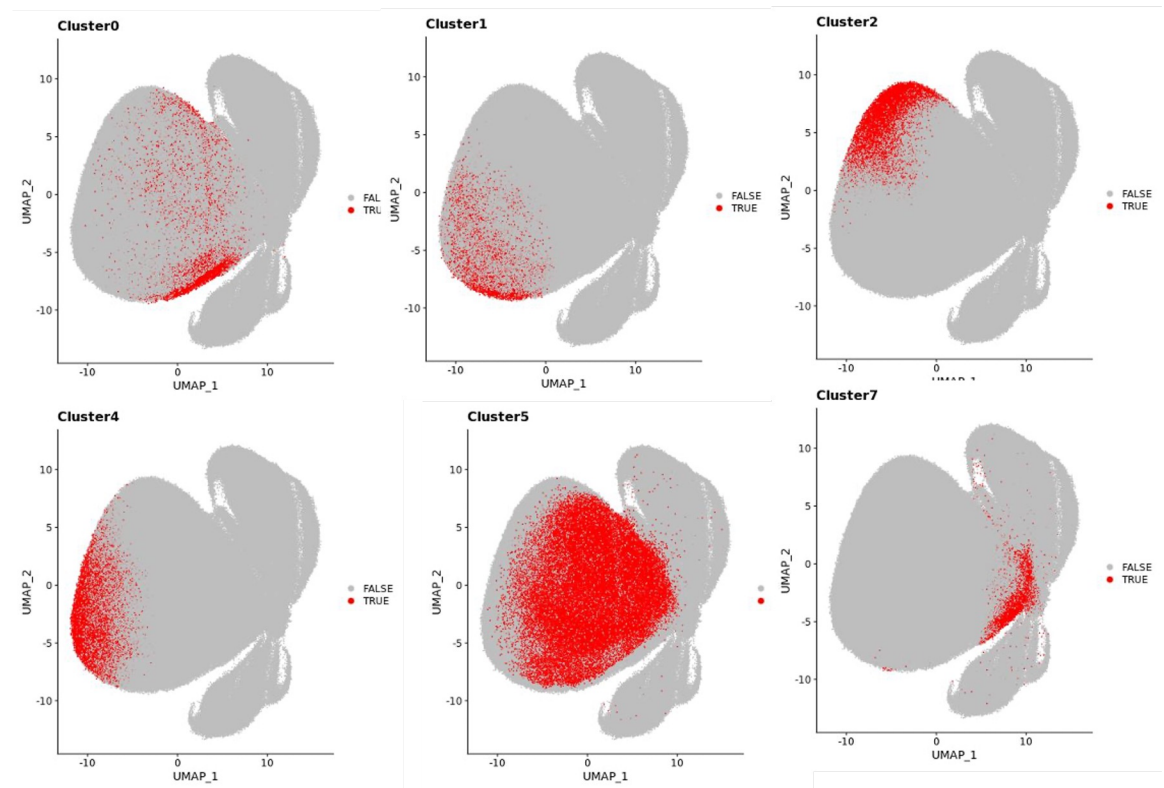
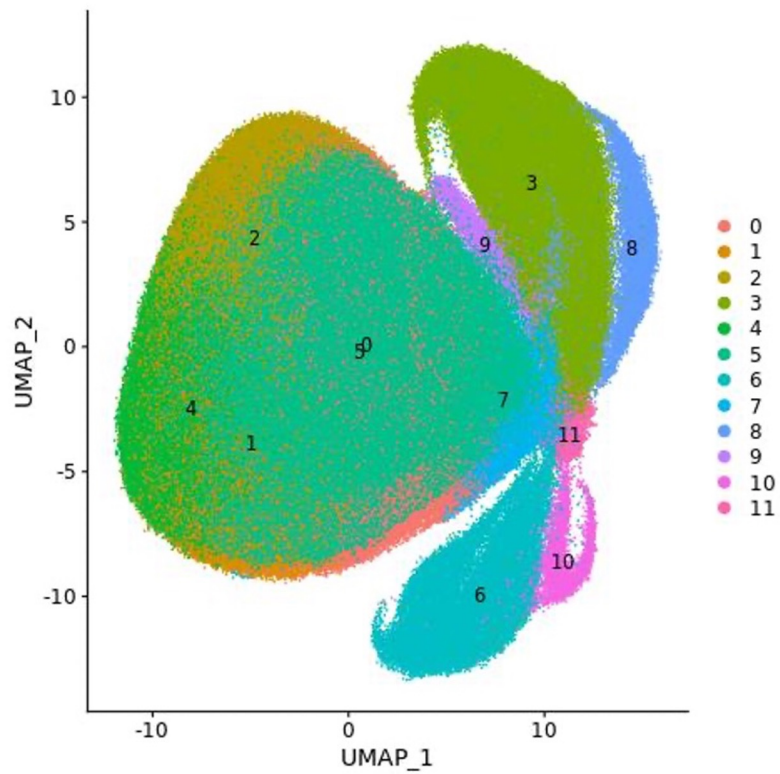
# How to select the appropriate parameter?

- Higher resolution / more clusters
- Some clusters may be redundant - merging
- Some clusters are “*unknown*”
- Rare clusters vs Sub-clustering big clusters

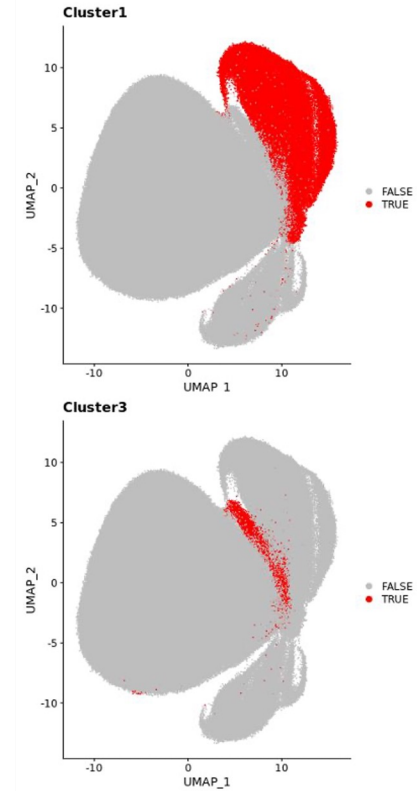
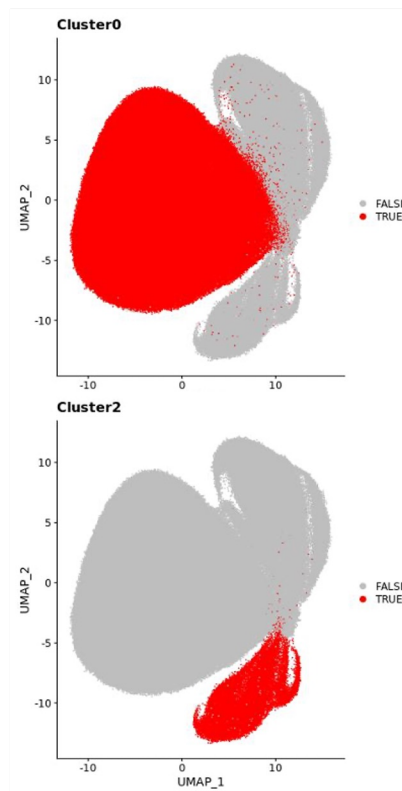
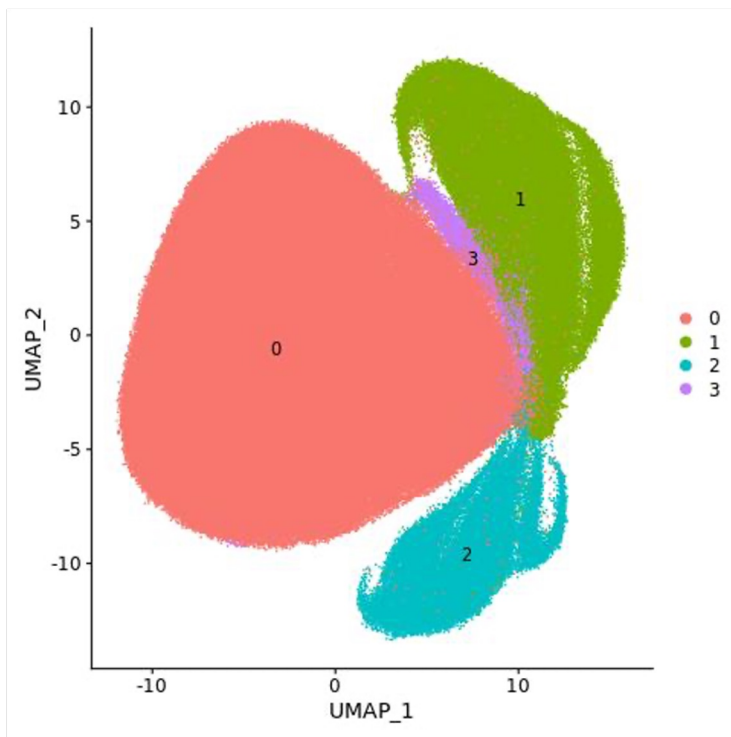
Is there a more objective method to pick the appropriate number of clusters?

- under vs over clustering
- reproducible clustering across datasets

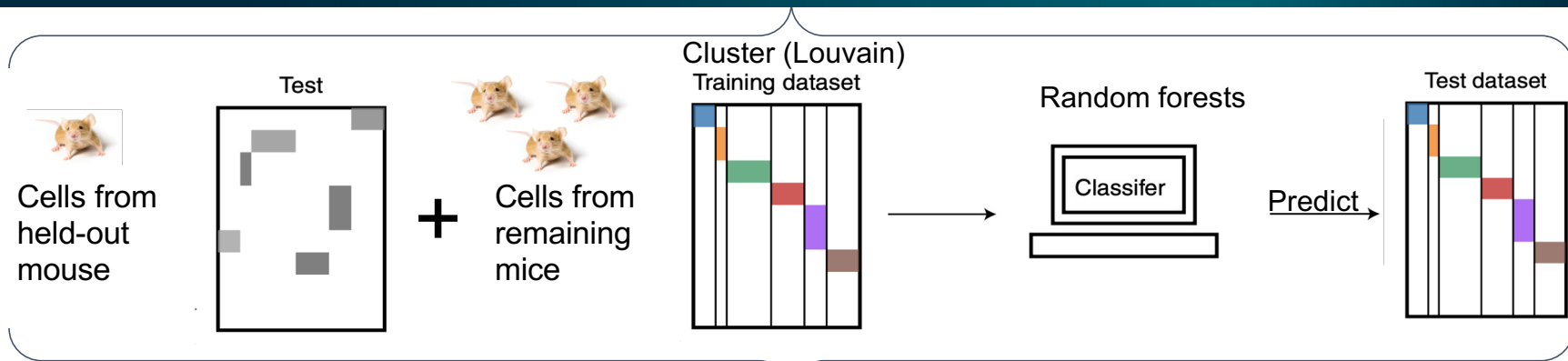
# Too many clusters?



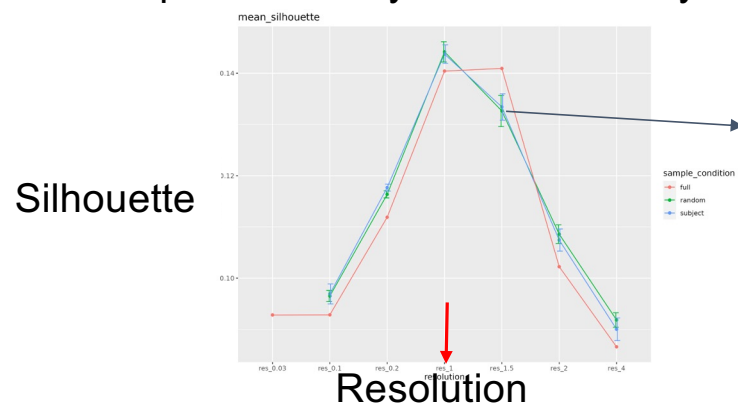
# Too few clusters?



# ML method based on cross-validation and silhouette score (Core: Min & Reuben)



Repeat for every mouse at every resolution



Error bars correspond to variation across different mice  
Can be viewed as a **measure of reproducibility**

Pick the resolution that maximizes Silhouette Width!



# Cluster -> cell type assignment

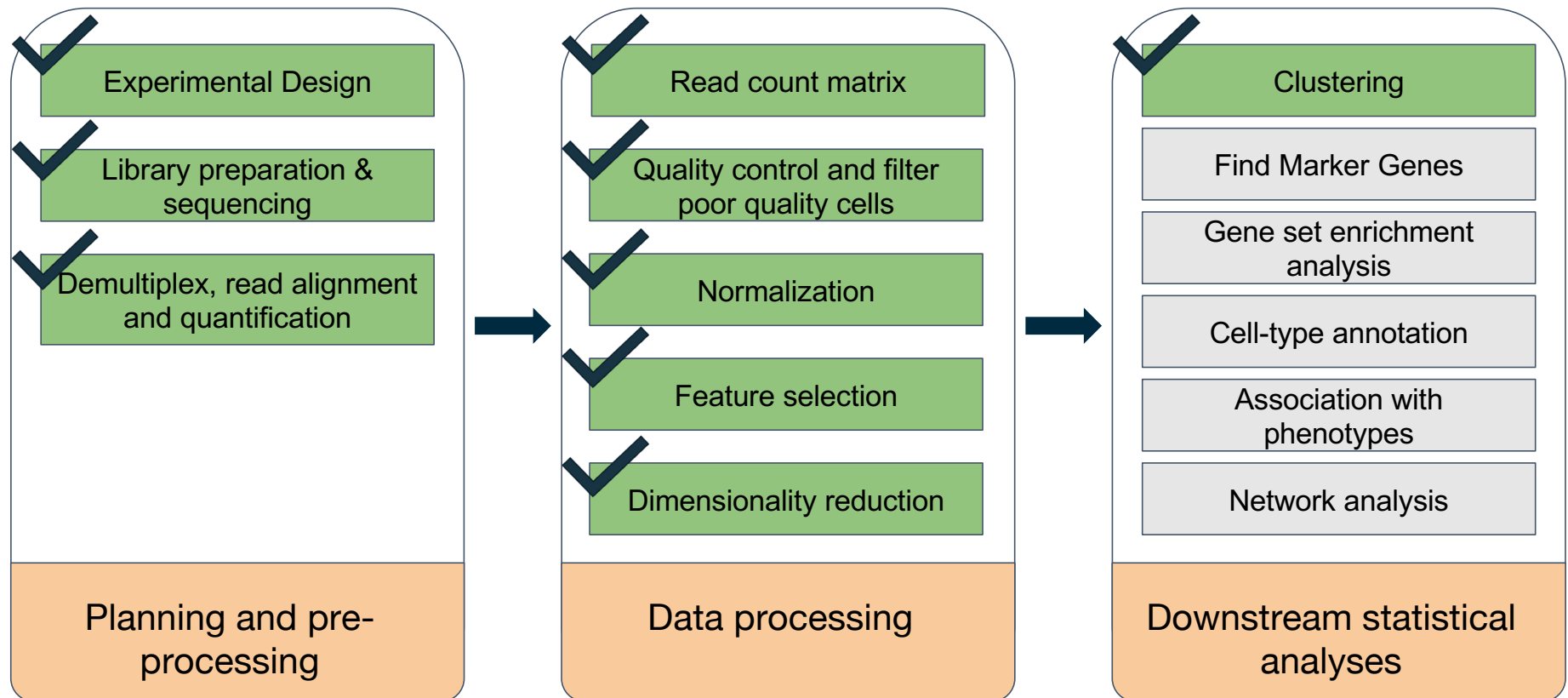
- Cluster annotation is challenging
- Based on feature markers expression
- Database of known markers per each cell type

ScType on github for unsupervised cell type annotation

- db of markers for each cell type (CellMarker and PanglaoDB)
- positive and negative markers - customizable
- Specificity score - marker genes show specificity both across clusters and cell types

We verify the overlap with findmarkers and sctype per cluster

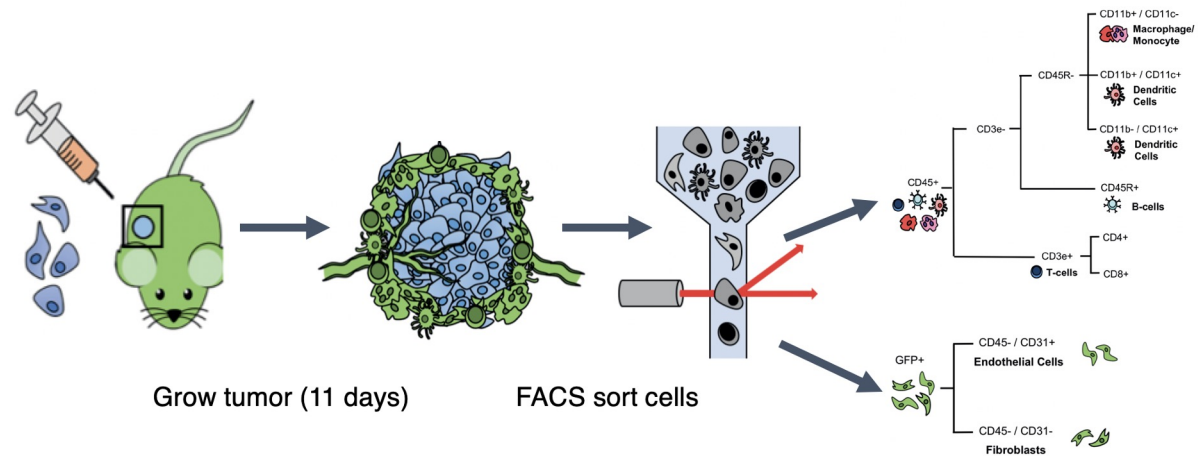
# scRNA-seq workflow



# 10. Hands-on

# Data for this workshop

- Two groups
- Multiple time points
- Replicates
- Analysis led to new cell types



(Image from Davidson et al, 2020)

# Find marker genes

- See the R script

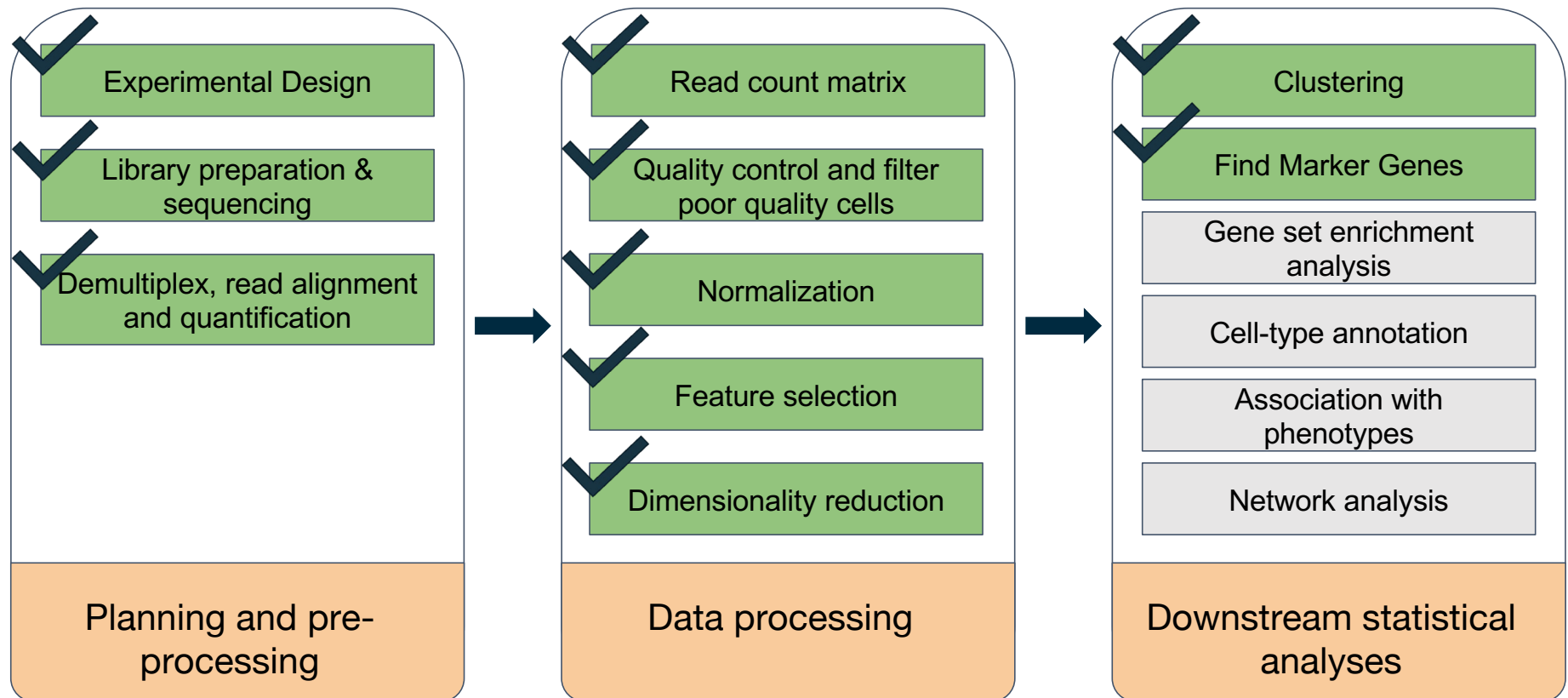
# Seurat options to visualize marker expression

- See the R script

# Many options to interact with the Seurat object

- Visit:  
[https://satijalab.org/seurat/articles/interaction\\_vignette.html](https://satijalab.org/seurat/articles/interaction_vignette.html)

# scRNA-seq workflow





# Conclusion

- Multiple algorithms might be required in the same project
- One-size-fits-all solution not available
- Observed patterns in data may be new biology or artifacts

## Upcoming sessions

- *Session 3*: Advanced discussion on normalization, differential analysis, and batch-correction. (Tue 1-4pm)

# Helpful resources

- Wynton Slack channel
  - [ucsf-wynton.slack.com](https://ucsf-wynton.slack.com)
- Gladstone Bioinformatics Core slack channel
  - <https://gladstoneinstitutes.slack.com/archives/C0145F1L7QS>
- Wynton tutorials
  - <https://github.com/ucsf-wynton/tutorials/wiki>

# Your feedback is important to us!

- <https://www.surveymonkey.com/r/F75J6VZ>
- ~3 min.



# GLADSTONE INSTITUTES