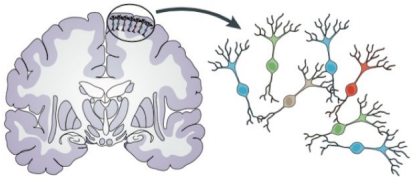
Single-cell and spatial omics

Stephanie Hicks
Department of Biomedical Engineering
Department of Biostatistics
https://stephaniehicks.com

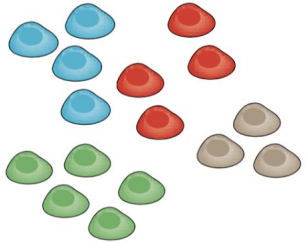
Lecture 15 -- Applied Comparative Genomics

Single-cell RNA-sequencing (scRNA-seq)

Tissue (e.g. tumor)



Isolate and sequence individual cells

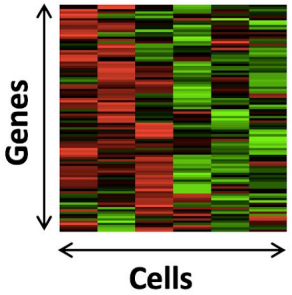


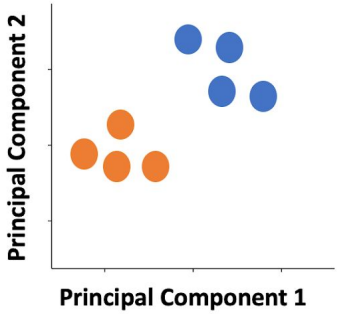
Gene 1
Cell 1

Read Counts

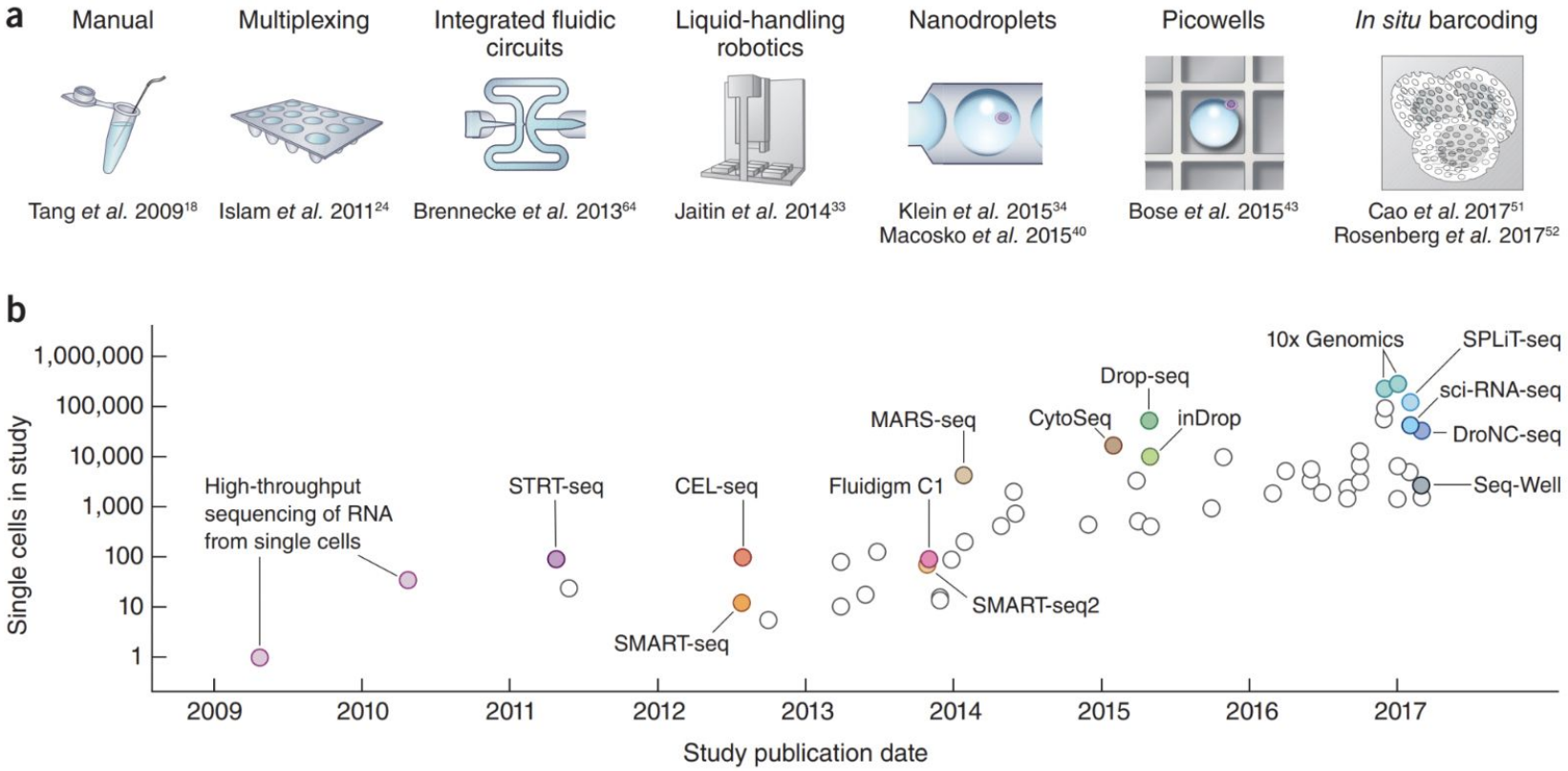
	Cell 1	Cell 2	…
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
…			

Compare gene expression profiles of single cells





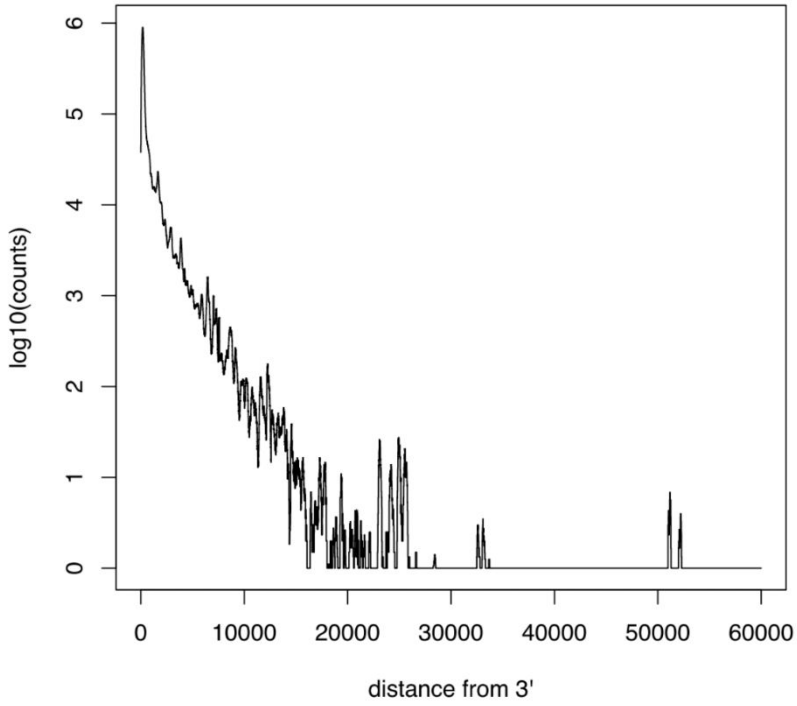
Evolution of single-cell technology



Svensson et al. (2018) *Nature Protocols*

Ntranos et al. 2019. DOI:10.1038/s41592-018-0303-9

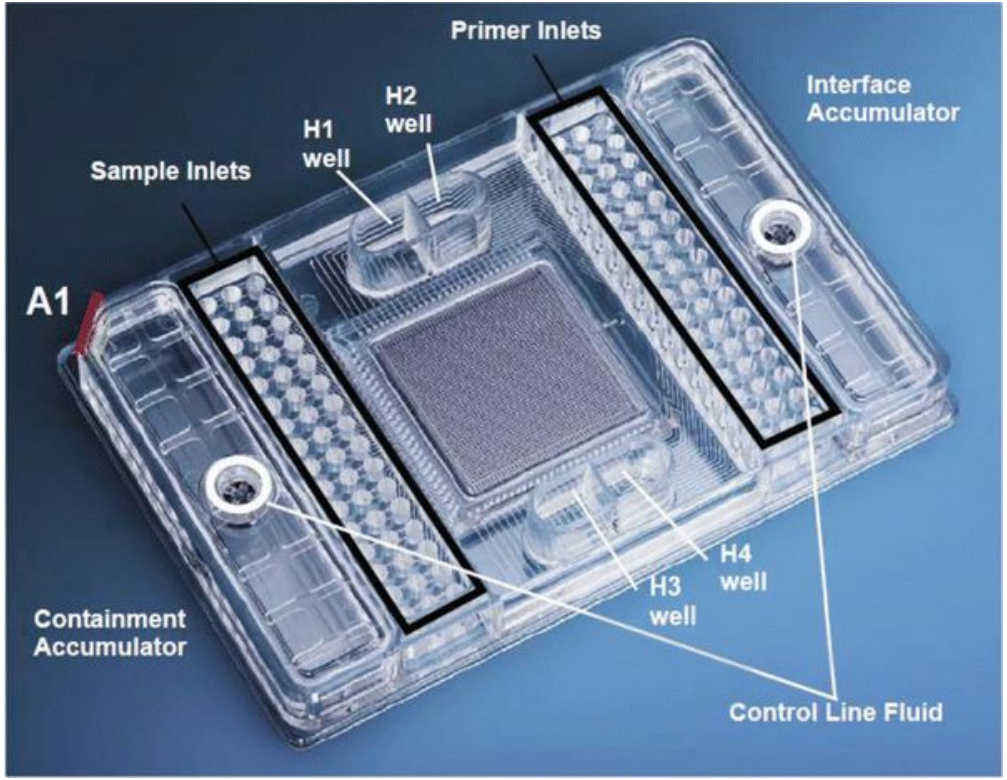
Capture full-length transcript vs 3’ end only



Isoform-level analyses only possible with full-length scRNA-seq or long-read RNA-seq

Plate-based

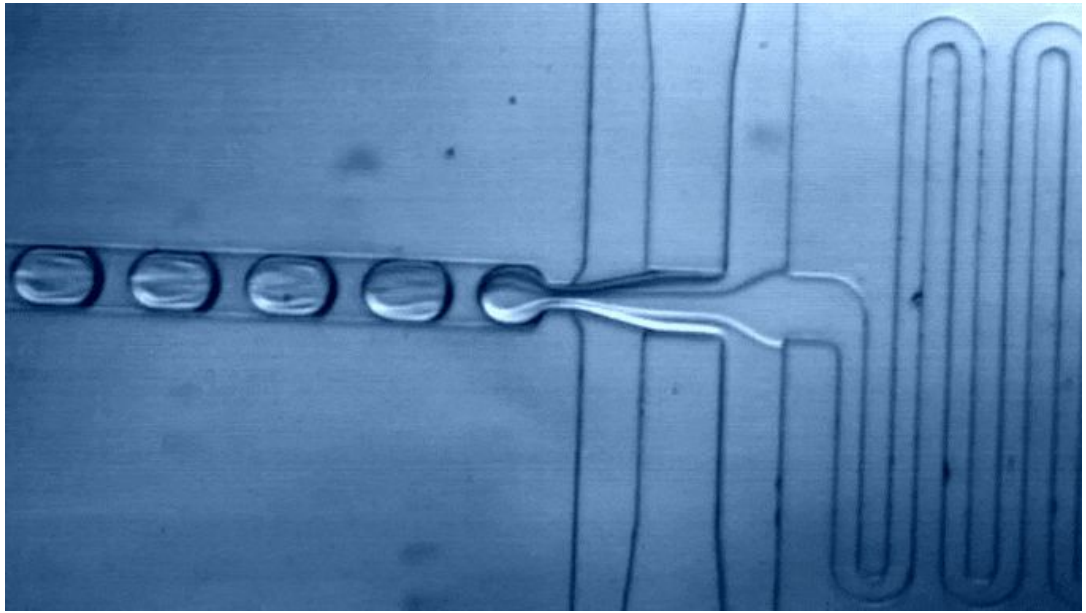
- lower throughput
- higher capture rate (% of input cells measured)



Adamowicz, Maratou & Aitman 2017 (https://doi.org/10.1007/978-1-4939-7481-8)

Droplet-based

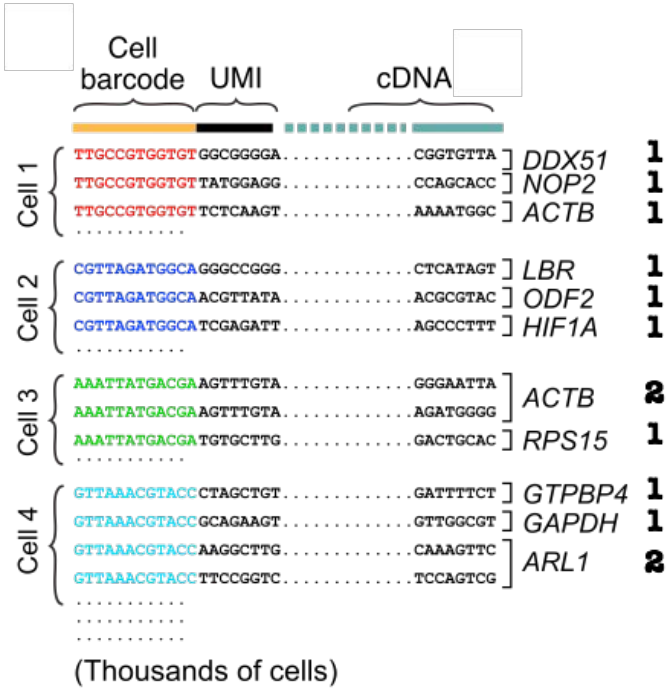
- higher throughput
- lower capture rate (% of input cells measured)



Macosko et al. 2015 (https://doi.org/10.1016/j.cell.2015.05.002)

Unique Molecular Identifiers (UMIs)

- PCR introduces **nonlinear** amplification bias
- UMIs are a way to tag each unique molecule in the sequencing library (before PCR)
- Number of possible UMIs = 4^L, where L is the length of the UMI
- Low L or sequencing errors can cause barcode collisions
- Afterward, count up only the number of distinct UMIs (collapse reads)



https://dnatech.genomecenter.ucdavis.edu/faqs/what-are-umis-and-why-are-they-used-in-high-throughput-sequencing

Quality control for scRNA-seq

Motivation for Quality Control

Low-quality scRNA-seq libraries can come from many places:

- Cell damage during dissociation
- Failure in library preparation (e.g., inefficient reverse transcription or PCR amplification)

These manifest as “cells” with

- low total counts
- few expressed genes
- high mitochondrial or spike-in proportions, etc

Motivation for Quality Control

Low-quality libraries are problematic as they can contribute to misleading results in downstream analyses:

- Often form their own distinct cluster(s) (complicating interpretation of results)
  - Often driven by increased mitochondrial proportions or enrichment for nuclear RNAs after cell damage
  - In the worst case, low-quality libraries generated from different cell types can cluster together based on similarities in the damage-induced expression profiles, creating artificial intermediate states or trajectories between otherwise distinct subpopulations

Motivation for Quality Control

Low-quality libraries are problematic as they can contribute to misleading results in downstream analyses:

- Interfere with characterization of population heterogeneity during variance estimation or principal components (PCs) analysis.
    - The first few PCs will capture differences in quality rather than biology, reducing the effectiveness of dimensionality reduction.
    - Similarly, genes with the largest variances will be driven by differences between low- and high-quality cells. The most obvious example involves low-quality libraries with very low counts where scaling normalization inflates the apparent variance of genes that happen to have a non-zero count in those libraries.

Motivation for Quality Control

Need to remove the problematic cells at the start of the analysis!

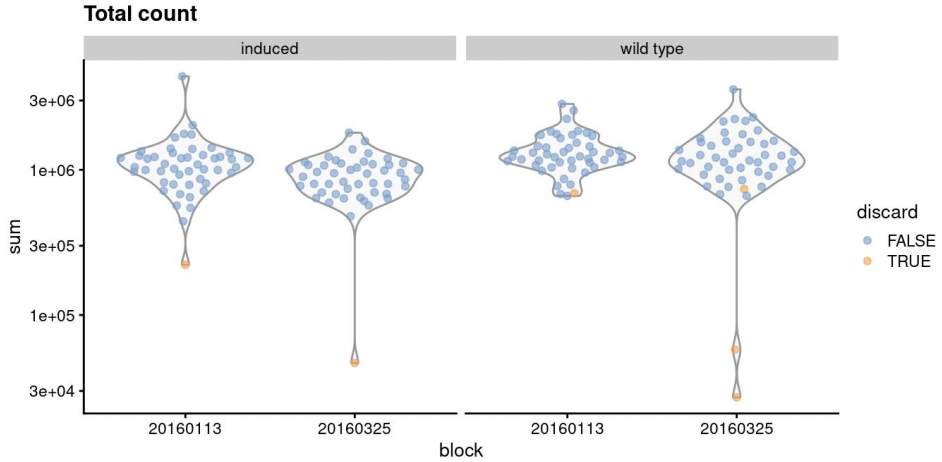
This step is commonly referred to as quality control (QC) on the cells.

(We will use “library”, “barcode”, and “cell” rather interchangeably here, though the distinction will become important when dealing with droplet-based data.)

Common choices for QC metrics

Library size (= total number of reads/UMIs across relevant features per cell)

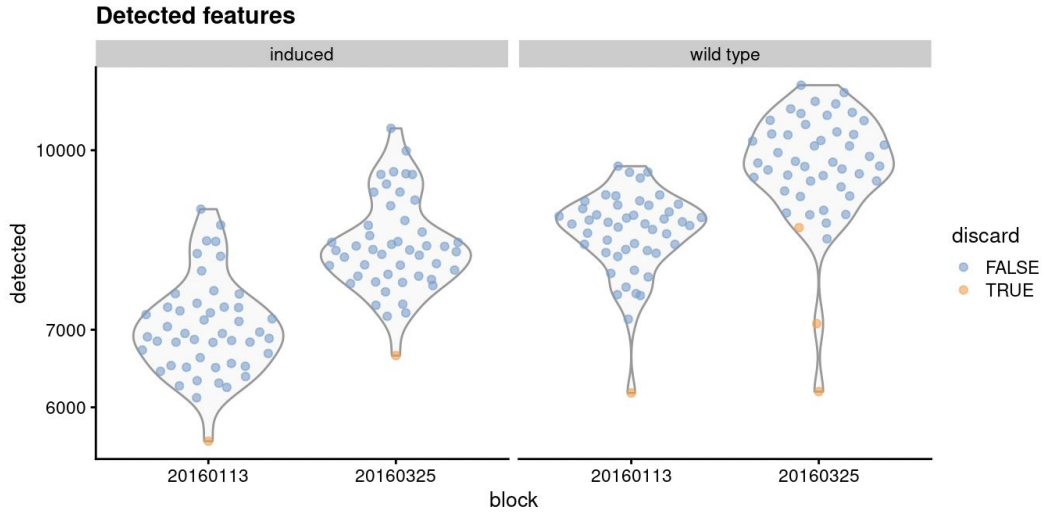
- Typically, we will consider the relevant features to be the endogenous genes
- Cells with small library sizes are of low quality as the RNA has been lost at some point during library preparation, either due to cell lysis or inefficient cDNA capture and amplification



Common choices for QC metrics

Number of expressed features per cell (sometimes “detection rate” or “dropout rate”)

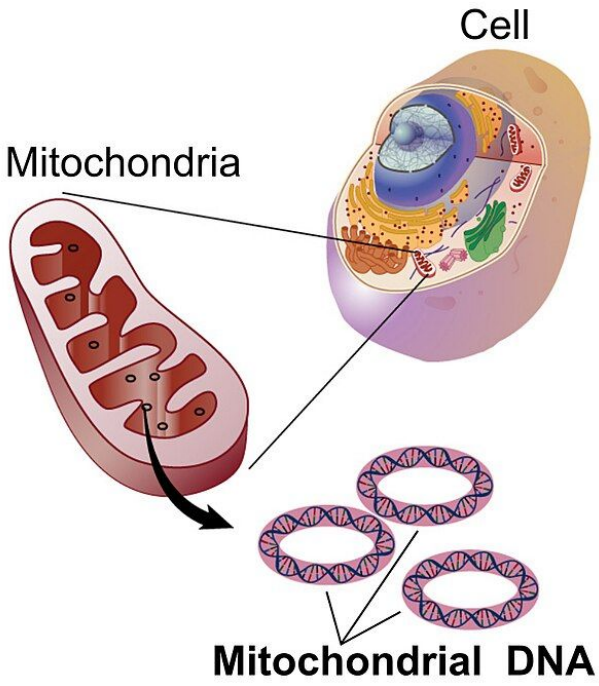
- Defined as the number of endogenous genes with non-zero counts for that cell
- Any cell with very few expressed genes is likely to be of poor quality as the diverse transcript population has not been successfully captured.



Common choices for QC metrics

Proportion of reads mapped to genes in the mitochondrial (mt) genome

- High proportions are indicative of poor-quality cells
  - Presumably because of loss of cytoplasmic RNA from perforated cells
- **Idea**: in the presence of modest damage, the holes in the cell membrane permit efflux of individual transcript molecules, but are too small to allow mitochondria to escape, leading to a relative enrichment of mitochondrial transcripts
- For **single-nuclei RNA-seq** experiments, high proportions are also useful as they can mark cells where the cytoplasm has not been successfully stripped.



How to choose thresholds?

Fixed thresholds. Remove if e.g.

- **Lib size** < 1e5
- **Num detected genes** < 100

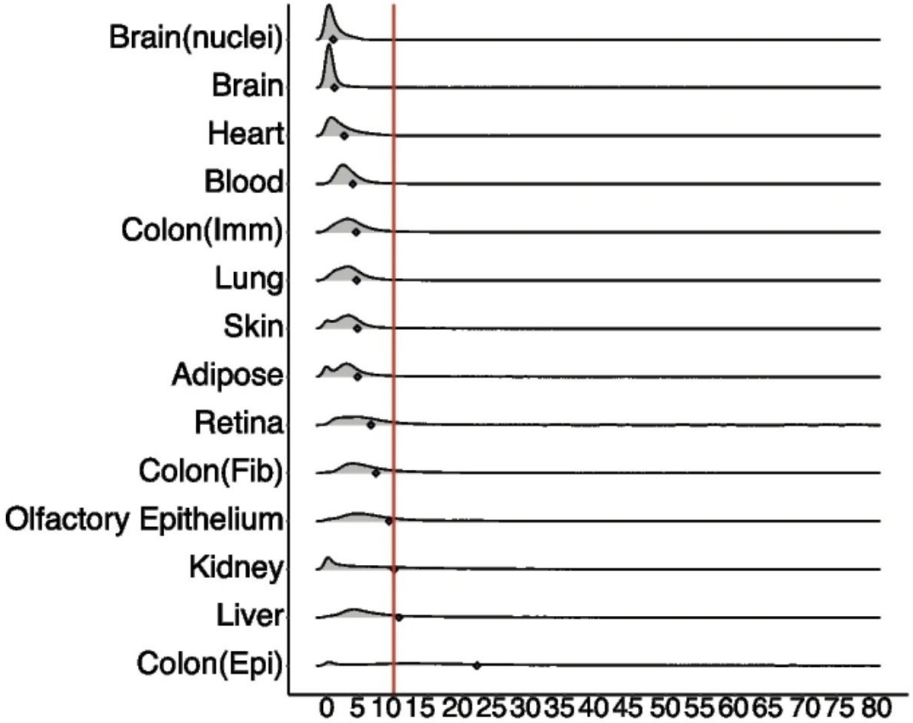
Adaptive thresholds.

- **Median absolute deviations**. For a data set X_1 X_2, ..., X_n, the MAD is defined as the median of the absolute deviations from the data's median $\tilde{X} = \text{median}(X)$

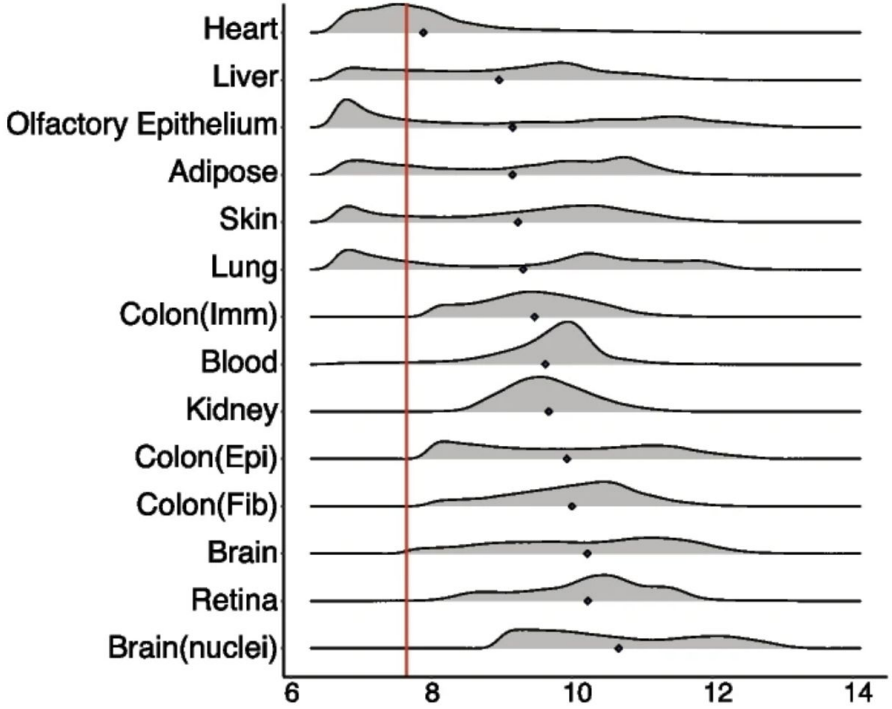
$$\text{MAD} = \text{median}(|X_i - \tilde{X}|)$$

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-022-02820-w

QC metrics vary by tissue



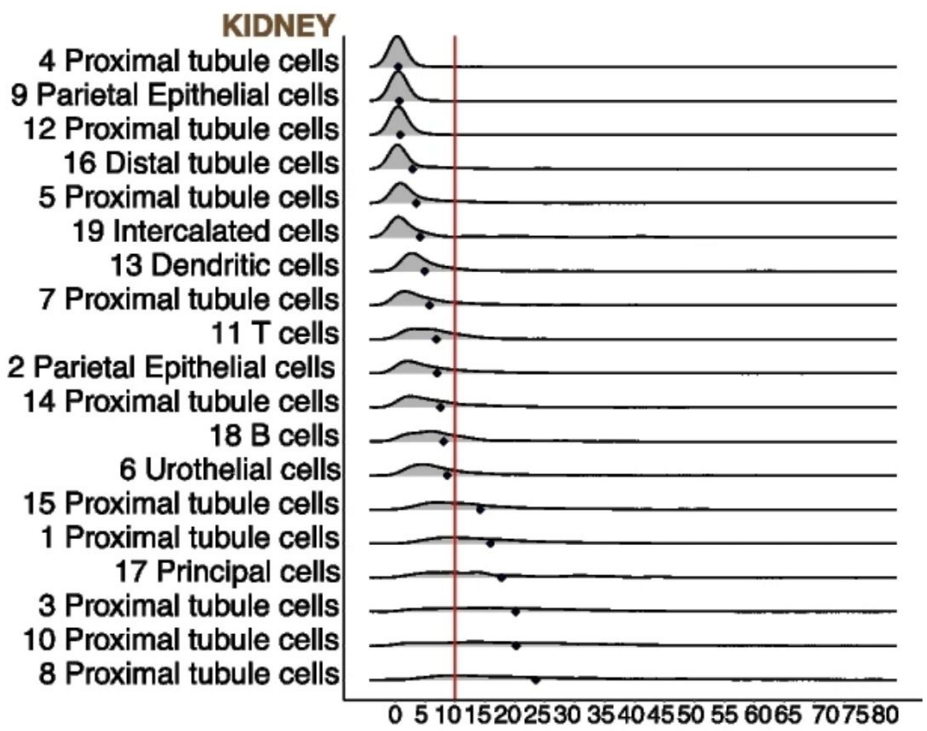
Fraction of mitochondria reads



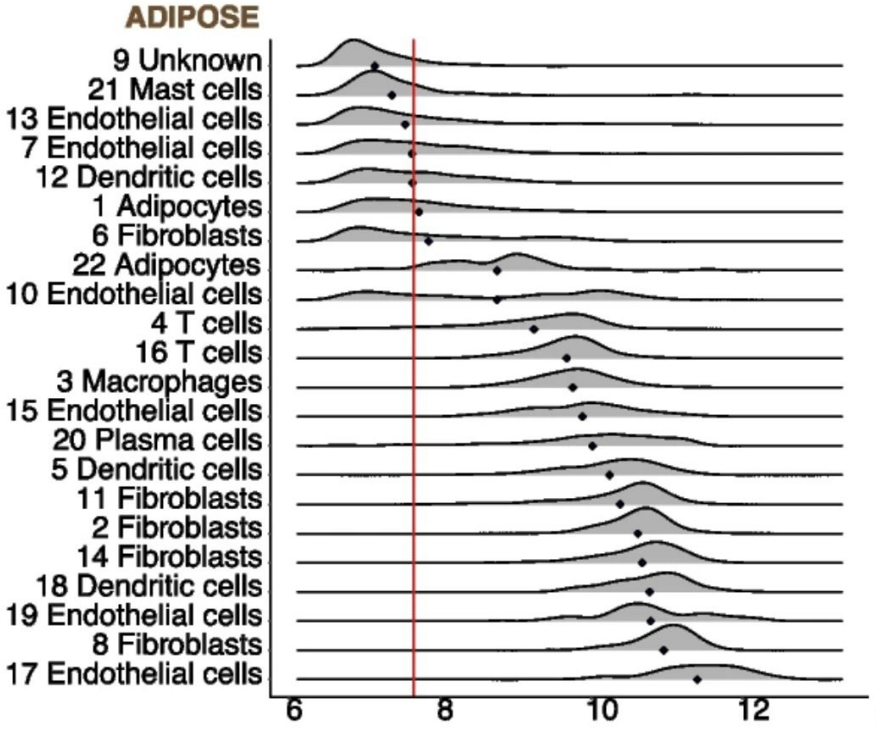
log2(num of detected genes)

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-022-02820-w

QC metrics vary by cell types



Fraction of mitochondria reads



log2(num of detected genes)

Normalization for scRNA-seq

Normalization and log-transformation

Now that we've removed problematic cells, need to put counts on a comparable scale before/during downstream analysis

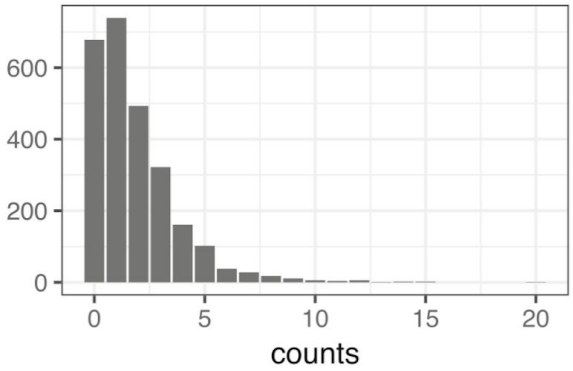
Main factors to account for:

- sequencing depth / library size (the tot sum of counts across all genes for each cell)
- batch effects (known or unknown)

Most common approaches:

- Scaling normalization e.g. CPM (either cell- OR cell- and gene-specific) followed by
- Log-transformation (and variance stabilization broadly see this post)

Library size normalization and log2 transformation



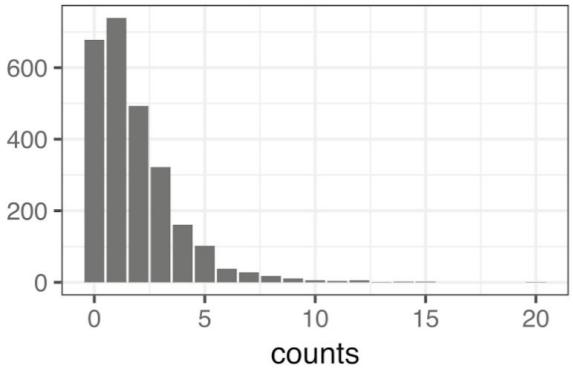
(a) UMI counts

$$y_{ij}$$

(for a given gene *j* and across all cells)

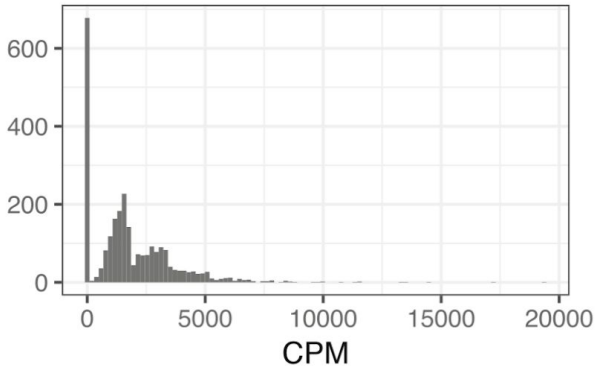
Townes et al. 2019, *Genome Biology*

Library size normalization and log2 transformation



(a) UMI counts

$$y_{ij}$$



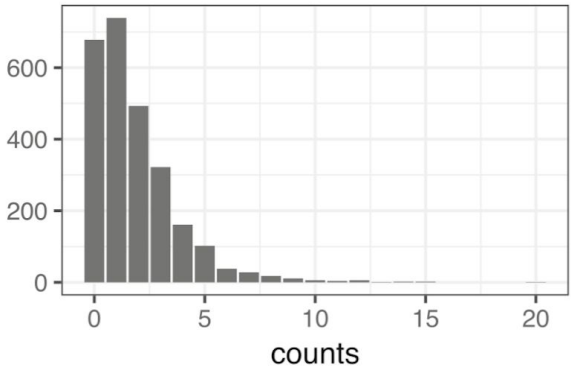
(b) counts per million (CPM)

$$(y_{ij}/n_i) \times 10^6$$

where $n_i = \sum_j y_{ij}$

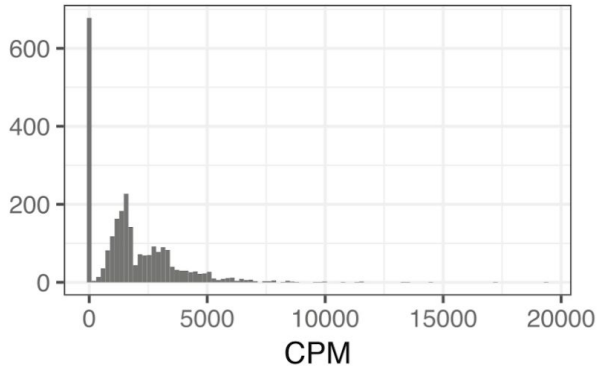
Townes et al. 2019, *Genome Biology*

Library size normalization and log2 transformation



(a) UMI counts

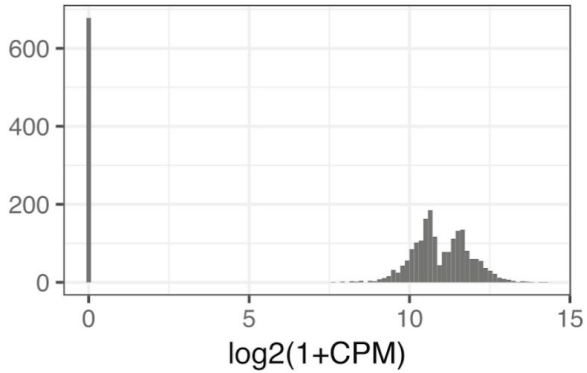
$$y_{ij}$$



(b) counts per million (CPM)

$$(y_{ij}/n_i) \times 10^6$$

where $n_i = \sum_j y_{ij}$

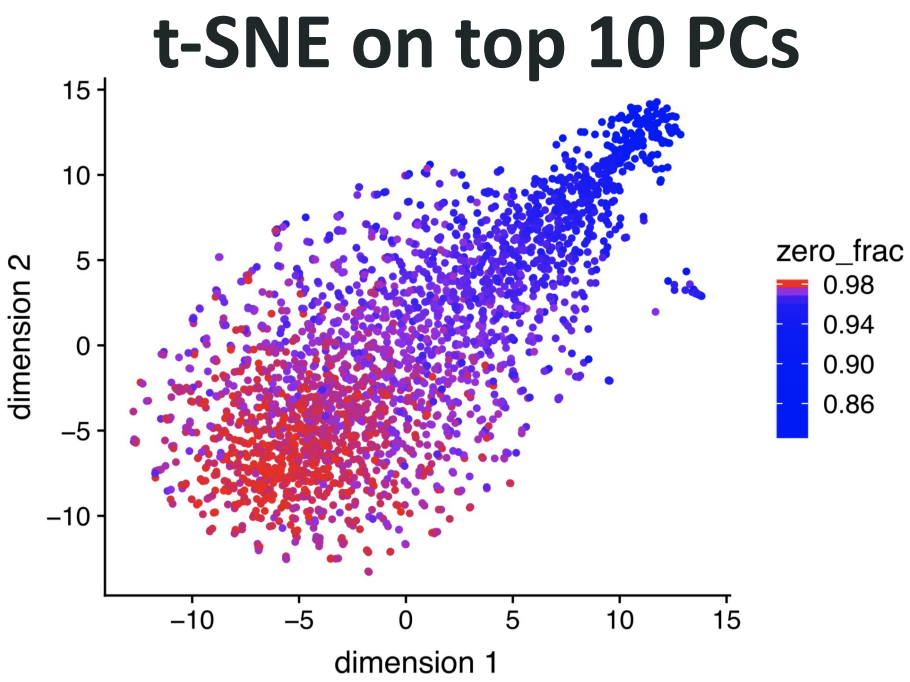


(c) log of CPM

Townes et al. 2019, *Genome Biology*

Artificial zero-inflation from log-transformation

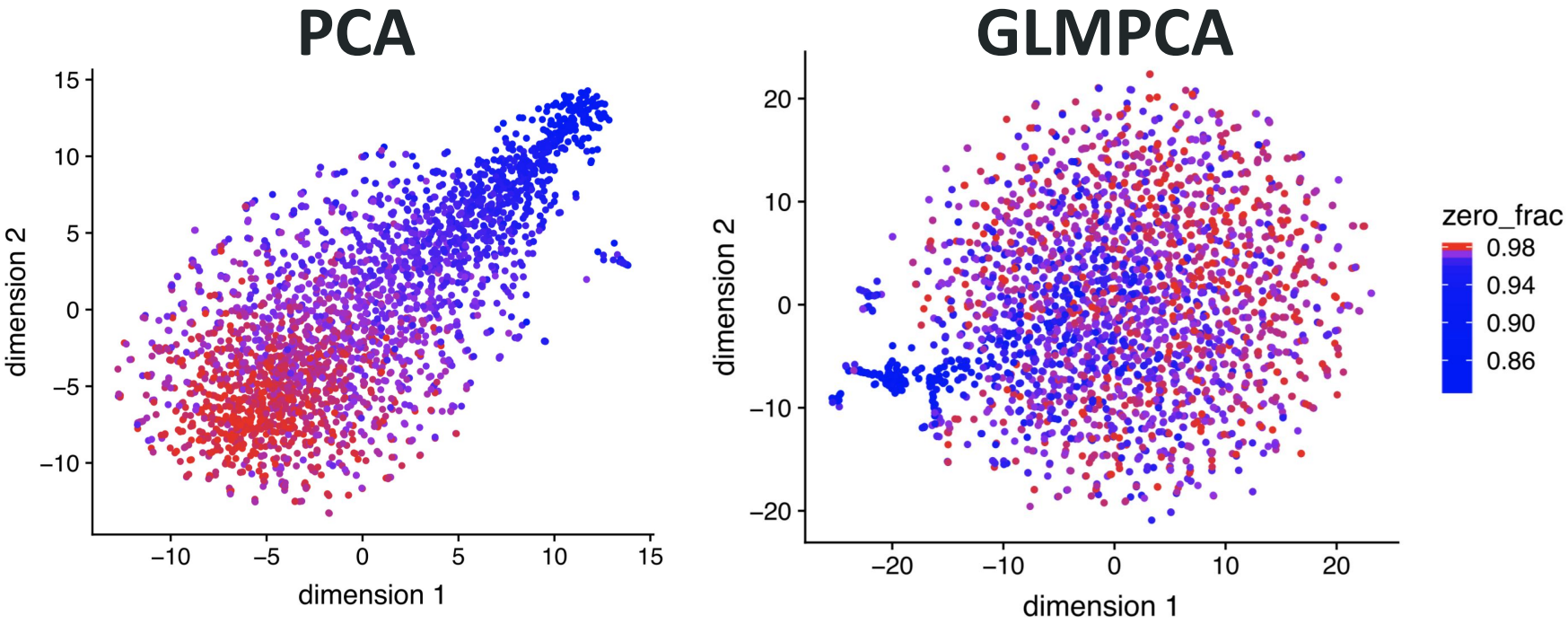
Negative control data



Townes et al. 2019, *Genome Biology*

Alternatives to CPM + log2 transformation

- glmpca (Townes et al. 2019) and scTransform: (Hafemeister & Satija 2019)
  - Key idea: Fit generalized linear model (e.g. poisson or negative binomial, etc) and the residuals of the model (i.e. after “regressing out” library size) are used for normalized counts



Feature selection

Motivation for feature selection

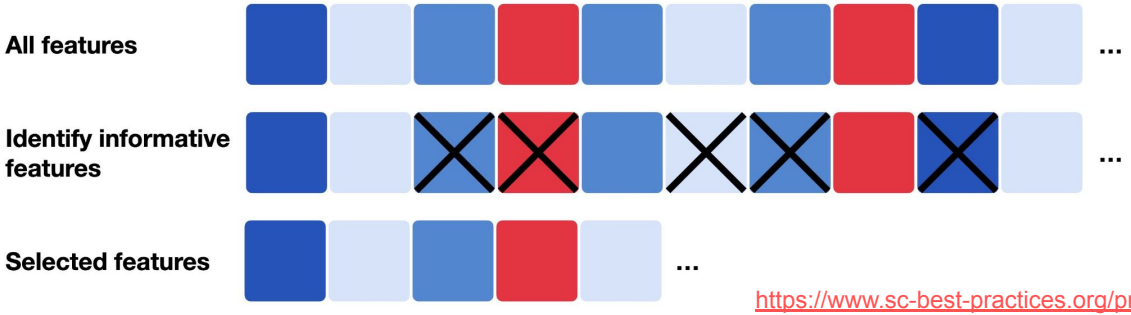
Often a primary goal of scRNA-seq analysis is to characterize heterogeneity across cells

Dim reduction & clustering combine per-gene diffs into a similarity metric between cells

The choice of ***which genes to use*** can heavily impact downstream analyses

1. Keep: genes with information about the biology of the system
2. Remove: genes that contain random noise

Feature selection aims to keep #1 and remove #2 to reduce the size of the data



https://www.sc-best-practices.org/preprocessing_visualization/feature_selection.html

Motivation for feature selection

Select most variable genes (e.g. out of 20K) based on expression across the population

This assumes genuine bio differences will manifest as increased variation in the affected genes (compared to genes only affected by tech noise or “uninteresting” biological variation (e.g. transcriptional bursting)

Goal: quantify variation per gene to select a set of “highly variable genes” (HVGs)

Also, you might be interested in “highly expressed”, “highly dropout”, “highly deviant”, etc

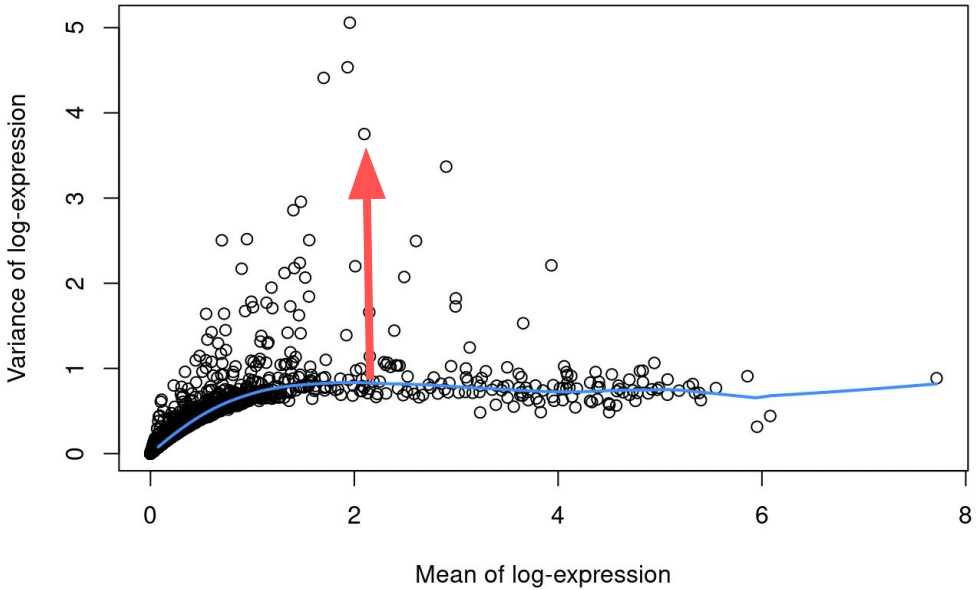
Mean-variance of log-normalized expression values

Simplest approach: Quantify per-gene variation by computing the variance of the log-normalized expression values (i.e. “log-counts”) for each gene across all cells

- **Rank genes by the empirical variance** (most to least) and pick top *n* genes
- Genes with the largest variances in log-values will contribute most to the Euclidean distances between cells during procedures like clustering and dimensionality reduction.
- But this **doesn’t capture the mean-variance relationship** in seq data

Mean-Variance of log-normalized expression values

Total variance = biological var (interesting) + technical var (uninteresting)



- Typically select top *n* genes with the largest biological components
- Nice because a user can directly control the number of genes retained (and hence the computational complexity of downstream analyses)

Other approaches to feature selection

- Tree-based approaches
- Shrinkage / regularization (e.g. LASSO)
- SVMs
- Perturbation-based deep learning (e.g. see this review)

While they might pick up on non-linearities, In practice, these approaches are not used due to being too slow, too memory-intensive, etc

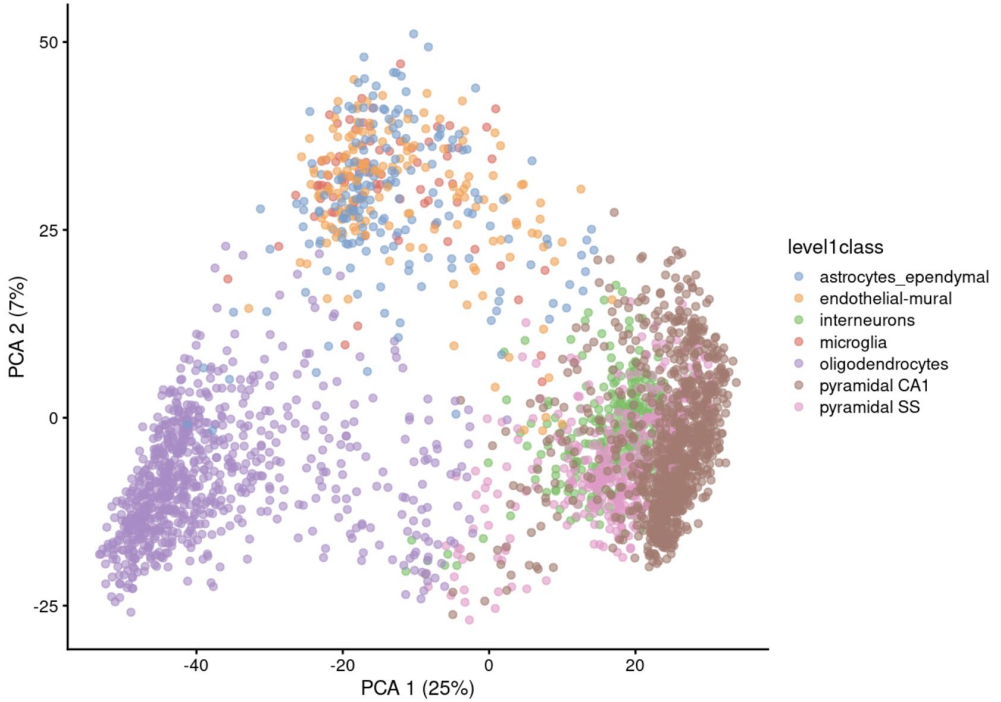
Also, simpler approaches tend to work well!

Dimensionality reduction

Principal Components Analysis (PCA)

Using top features selected, PCA finds orthogonal vectors (linear combinations of features in original dataset) that capture the largest amount of variation

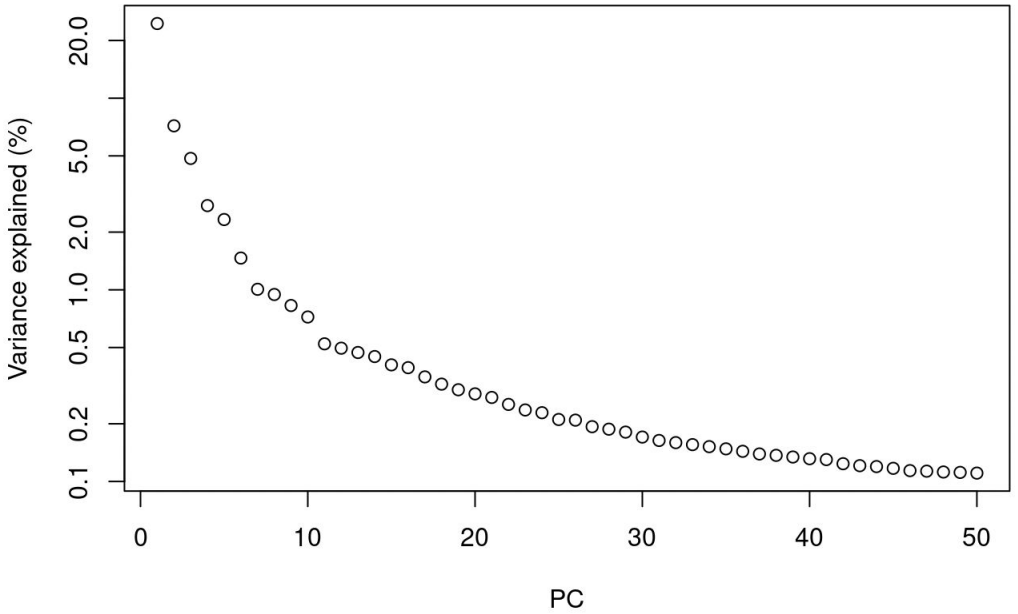
- Highly interpretable and computationally efficient
- PCs with the lowest variance are discarded to effectively reduce the dimensionality of the data without losing information
- Top PCs are typically used for downstream (e.g. clustering)



How to choose the number of PCs?

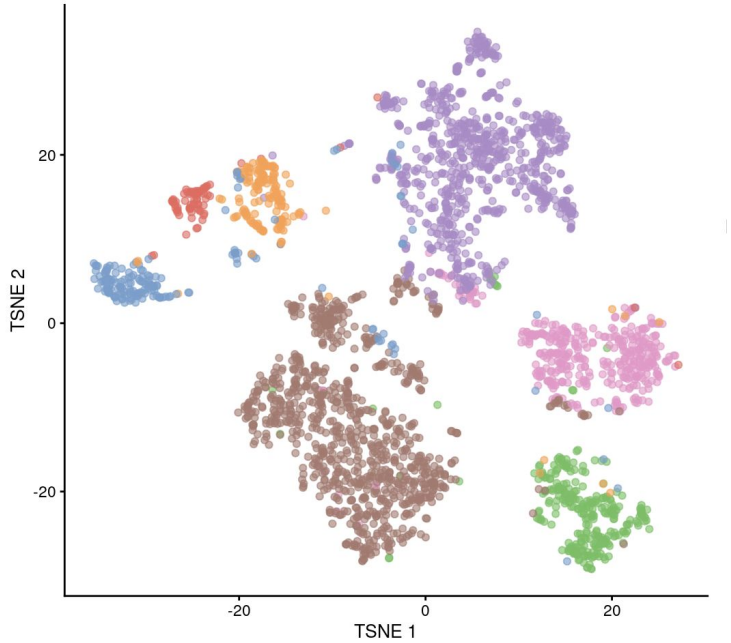
More top PCs retains more biological information, but too much introduces noise and takes longer to run methods

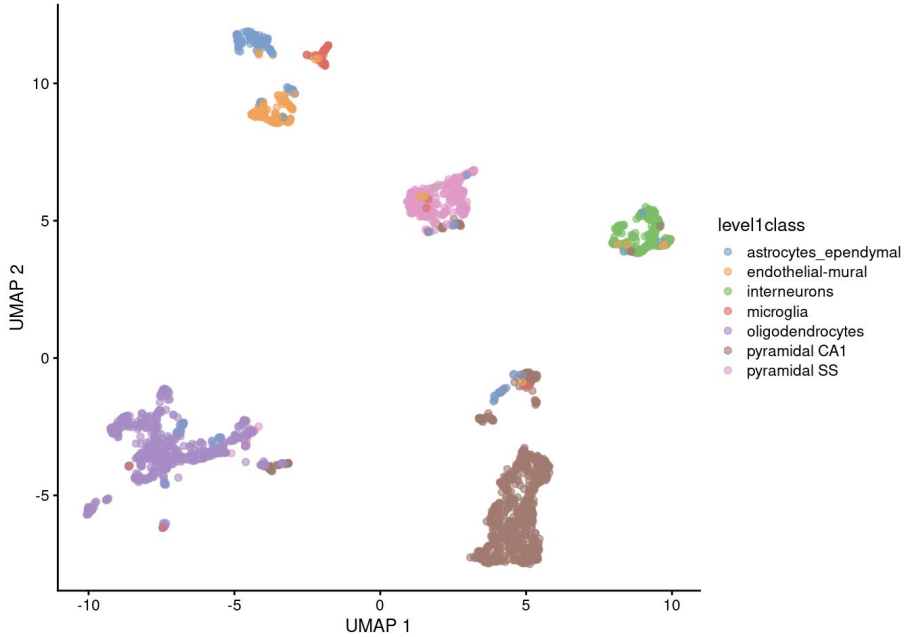
Often arbitrary # of PCs (e.g. 10-100) is chosen



Non-linear methods ***for data visualization***

- *t*-stochastic neighbor embedding
- Uniform manifold approximation and projection (UMAP)





Other approaches to dimensionality reduction

- Independent Components Analysis (ICA)
- Zero-inflated factor analysis (ZIFA) (link)
- PHATE (link)
- scvi-tools (link)

Clustering

Motivation for clustering

Goal: empirically define groups of cells with similar expression profiles.

Purpose: summarize complex scRNA-seq data into a digestible format for human interpretation.

This allows us to describe population heterogeneity in terms of discrete labels that are easily understood, rather than attempting to comprehend the high-dimensional manifold on which the cells truly reside.

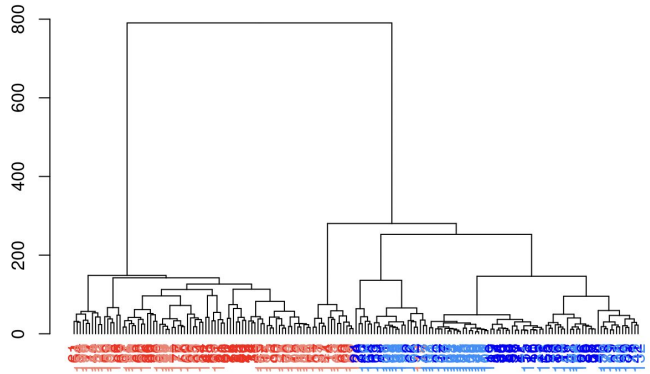
After annotation based on marker genes, the clusters can be treated as proxies for more abstract biological concepts such as cell types or states.

Hierarchical clustering

1. Calculate a pairwise cell-to-cell distance matrix
2. Arrange cells into a hierarchy based on relative similarity to each other
3. Build and cut dendrogram based on pairwise distance matrix

Pros: Visually fantastic to quantitatively capture pop relationships at various resolutions

Cons: Way too slow (*quadratic* in time for increasing # of cells); greedy algorithm (i.e. likely a suboptimal partition)

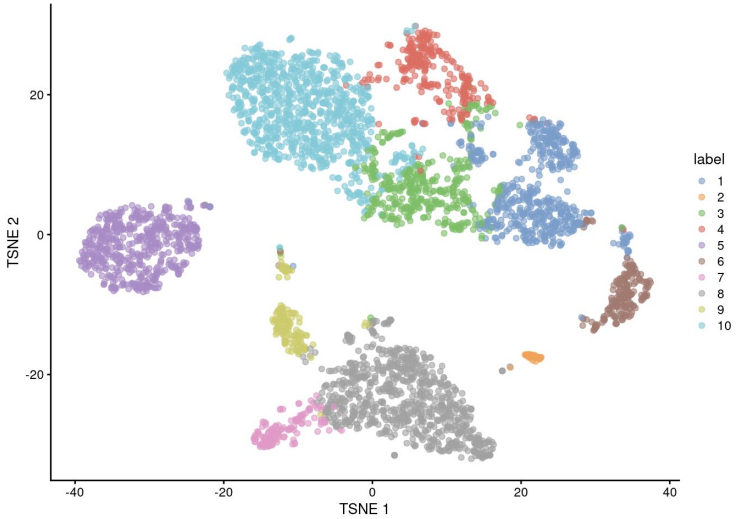


k-means clustering

1. Assignment: Given centroids, assign a cell to the cluster with the closest centroid
2. Update: Compute new centroid profiles for each cluster

Pros: Super simple and easy algorithm to understand; scalable to thousands of cells

Cons: Assumption that the data have spherical variance



Graph-based clustering

Idea:

1. Build a graph structure where each node is a cell that is connected to its neighbs. Edges are weighted based on the similarity between the cells
2. Apply algorithms to identify “communities” of cells that are more connected to cells in the same community than they are to cells of different communities. Each community represents a cluster that we can use for downstream interpretation.

Pros: (1) Super scalable (ie only needs *k*-NN search log-linear time vs quadratic) in # of cells, and (2) Graph avoids making strong assumptions about the shape of the clusters

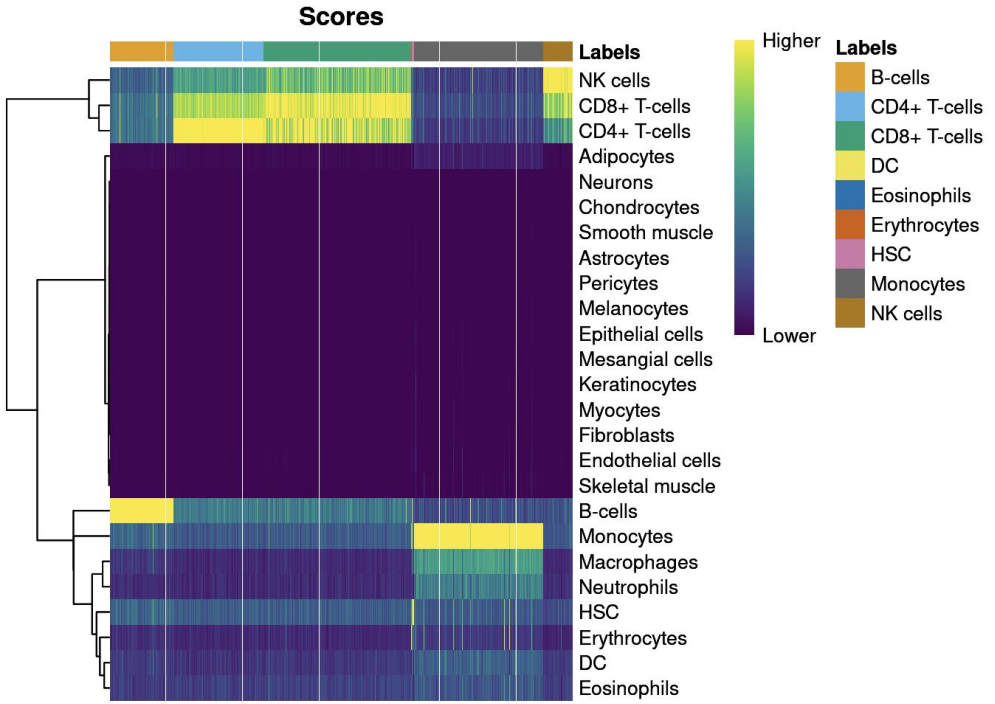
Cons: after graph, no information is retained about relationships beyond cells

Cell type annotation

Cell type annotation / labeling

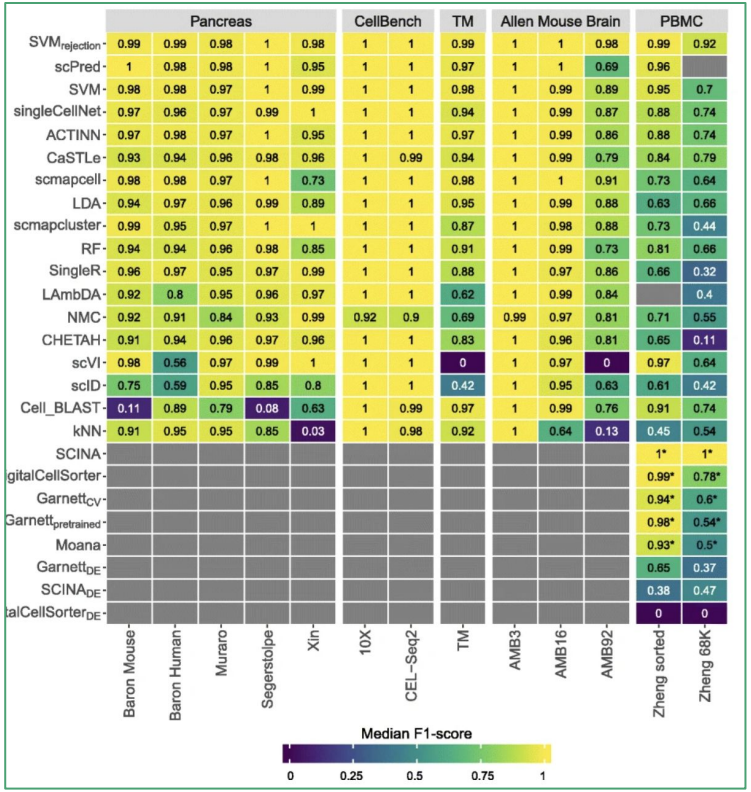
Assign cell labels using reference datasets

- Correlation to training data
- Or simply plot gene markers



Cell type annotation / labeling

Name	Version	Language	Underlying classifier	Prior knowledge
Garnett	0.1.4	R	Generalized linear model	Yes
Moana	0.1.1	Python	SVM with linear kernel	Yes
DigitalCellSorter	GitHub version: e369a34	Python	Voting based on cell type markers	Yes
SCINA	1.1.0	R	Bimodal distribution fitting for marker genes	Yes
scVI	0.3.0	Python	Neural network	No
Cell-BLAST	0.1.2	Python	Cell-to-cell similarity	No
ACTINN	GitHub version: 563bcc1	Python	Neural network	No
LAmbDA	GitHub version: 3891d72	Python	Random forest	No
scmapcluster	1.5.1	R	Nearest median classifier	No
scmapcell	1.5.1	R	kNN	No
scPred	0.0.0.9000	R	SVM with radial kernel	No
CHETAH	0.99.5	R	Correlation to training set	No
CaSTLe	GitHub version: 258b278	R	Random forest	No
SingleR	0.2.2	R	Correlation to training set	No
scID	0.0.0.9000	R	LDA	No
singleCellNet	0.1.0	R	Random forest	No
LDA	0.19.2	Python	LDA	No
NMC	0.19.2	Python	NMC	No
RF	0.19.2	Python	RF (50 trees)	No
SVM	0.19.2	Python	SVM (linear kernel)	No
$SVM_{rejection}$	0.19.2	Python	SVM (linear kernel)	No
kNN	0.19.2	Python	kNN ($k=9$)	No



A comparison of automatic cell identification methods for single-cell RNA sequencing data

Data integration

Data integration

Challenge: remove batch effects and integrating two or more datasets

- Lots of active research in this area!
- Batch effects in scRNAseq can arise at different levels (e.g. samples, donors, datasets, experimental labs

Batch-effect removal methods can vary in each of these three steps e.g.

- May use various linear or non-linear dimensionality reduction approaches, linear or non-linear batch effect models
- May output different formats of batch-corrected data

Data integration

Global models

- Fit regression model with batch effect covariate

Residuals (often using linear regression):

$$\hat{n}_{gc} = f_D(B_c, \dots)$$

$$r_{gc} = n_{gc} - \hat{n}_{gc} = n_{gc} - (\beta_0 + \beta_1 B_c)$$

in linear model case

Example:
sc.tl.regress_out()

Correct for fitted batch effect:

$$n_{gcb} = \alpha_g + X\beta_g + \gamma_{gb} + \delta_{gb}\epsilon_{gcb}$$

bio design matrix

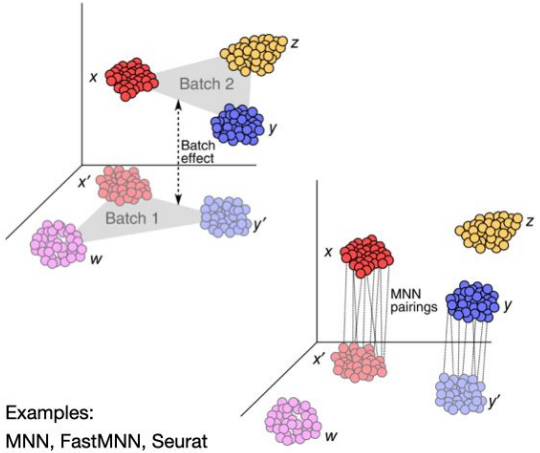
additive batch effect

multiplicative batch effect

Example:
ComBat - ***scanpy.pp.combat()***

Linear embedding models

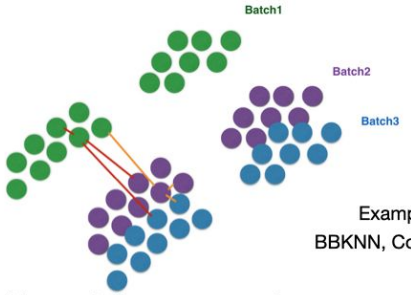
- Project cells into low dimensional embedding
- find most similar cells in other batch e.g., using mutual nearest neighbours (MNNs)
- Use MNNs as anchors to calculate a correction vector



Examples:
MNN, FastMNN, Seurat v3, Scanorama

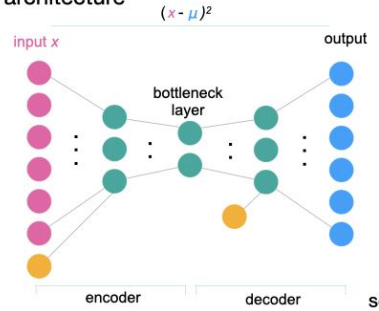
Graph-based methods & Deep learning

Enforce graph connections between different batches



Examples:
BBKNN, Conos

Add condition node into auto-encoder architecture

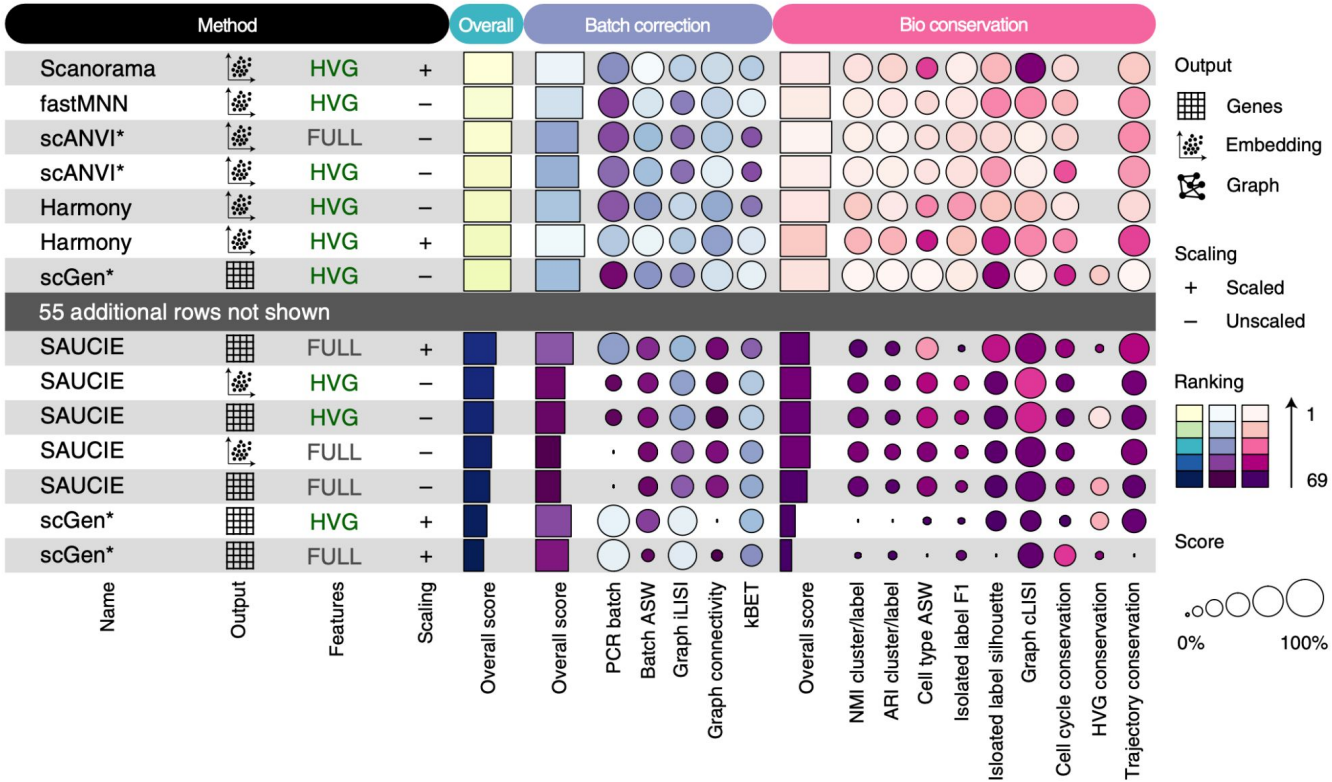


Examples:
scVI, trVAE, SAUCIE

https://www.sc-best-practices.org/cellular_structure/integration.html

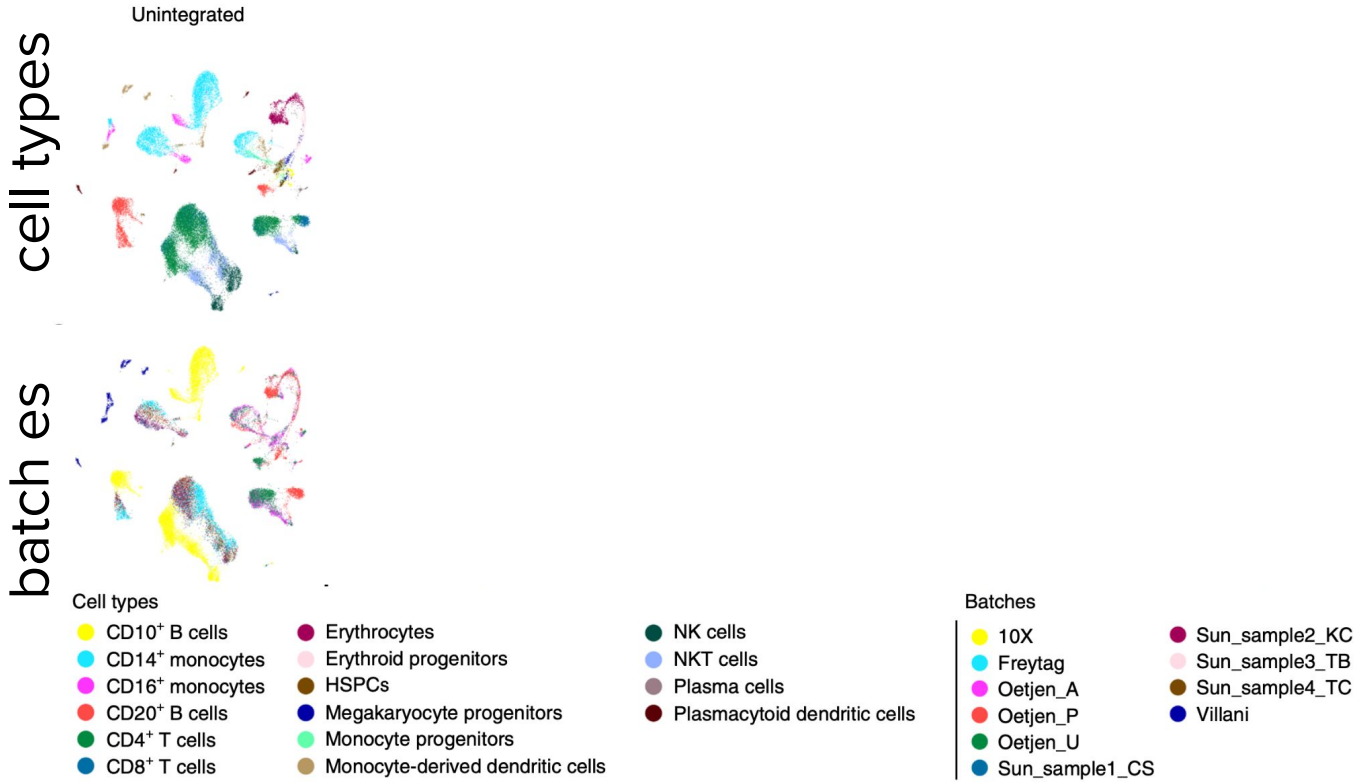
Data integration: benchmarking

Not quite an independent assessment though!



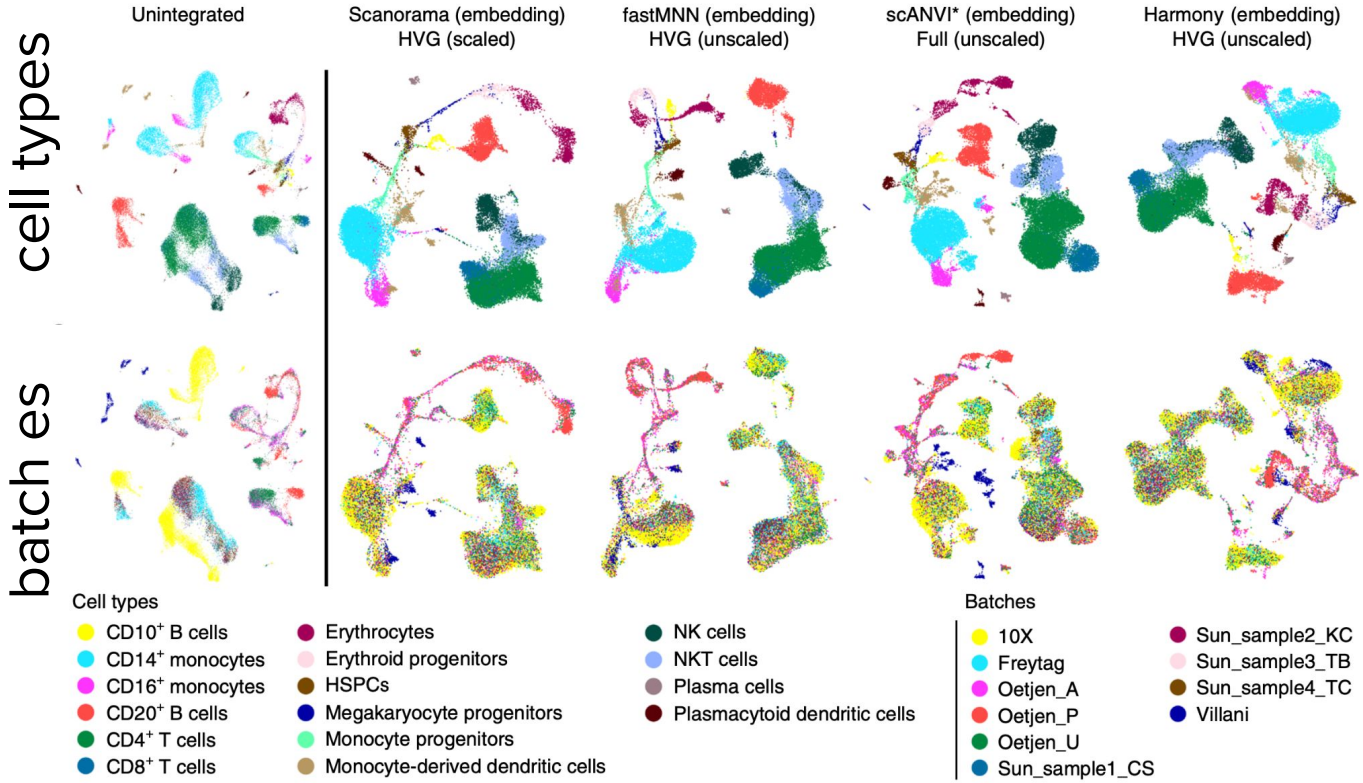
https://www.nature.com/articles/s41592-021-01336-8

Data integration: benchmarking



https://www.nature.com/articles/s41592-021-01336-8

Data integration: benchmarking



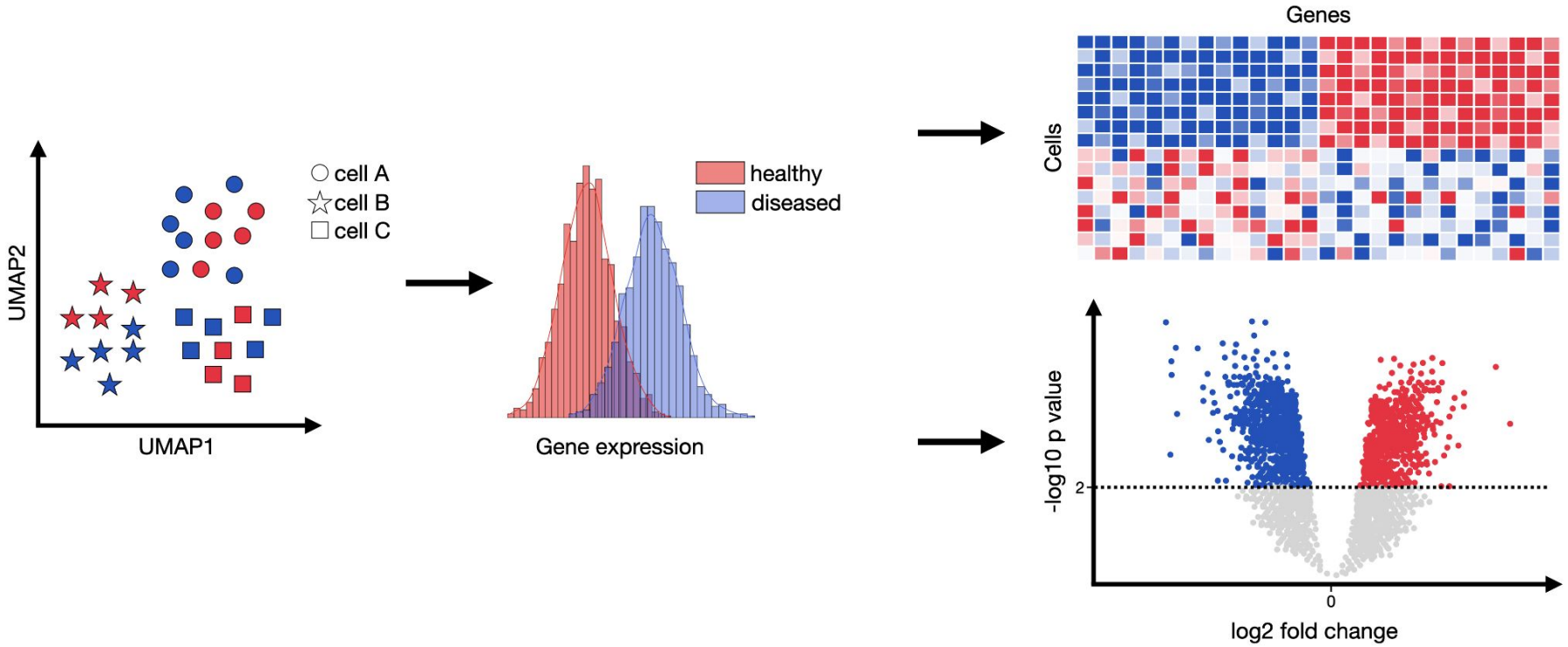
https://www.nature.com/articles/s41592-021-01336-8

Differential expression analysis

Differential expression (DE) analysis

Differential gene expression (DE / DGE) analysis attempts to infer genes that are statistically significantly over- or under-expressed between any one or more conditions such as diseases, genetic knockouts or drugs (commonly between healthy and condition per cell type)

Differential expression (DE) analysis



https://www.sc-best-practices.org/conditions/differential_gene_expression.html

Differential expression (DE) analysis

Common DE approaches:

- Cell level:
  - *t*-test (comparing the differences in mean expression between two groups)
  - Wilcoxon Rank Sum Test (basically same as *t*-test, but nonparametric)
  - Differences in distributions (link)
  - Generalized mixed effect models such as MAST (link) or glmmTMB (link)

Differential expression (DE) analysis

Common DE approaches:

- Cell level:
  - *t*-test (comparing the differences in mean expression between two groups)
  - Wilcoxon Rank Sum Test (basically same as *t*-test, but nonparametric)
  - Differences in distributions (link)
  - Generalized mixed effect models such as MAST (link) or glmmTMB (link)
- Sample/donor-level
  - Aggregate cell counts to create “pseudobulks” and then analysed with methods originally designed for bulk expression samples such as edgeR (link) or DEseq2 (link)

Differential expression (DE) analysis

Common DE approaches:

- Cell level:
  - *t*-test (comparing the differences in mean expression between two groups)
  - Wilcoxon Rank Sum Test (basically same as *t*-test, but nonparametric)
  - Differences in distributions (link)
  - Generalized mixed effect models such as MAST (link) or glmmTMB (link)
- Sample/donor-level
  - Aggregate cell counts to create “pseudobulks” and then analysed with methods originally designed for bulk expression samples such as edgeR (link) or DEseq2 (link)

How to choose? My general two cents...

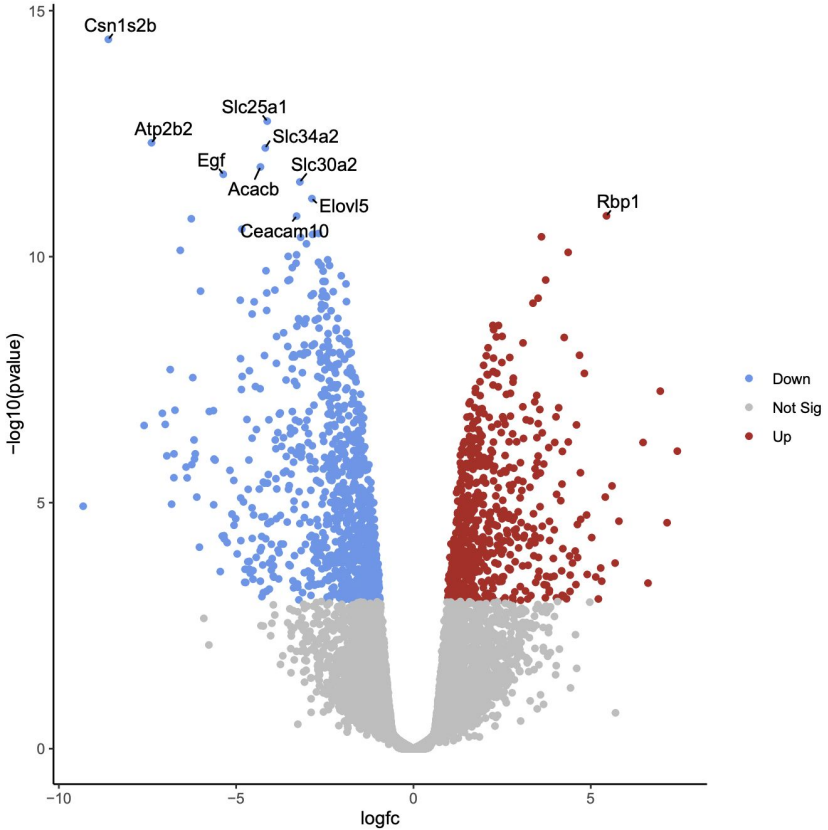
- Pseudobulk approaches tend to be more robust as single-cell data is noisy and sparse!

Differential expression (DE) analysis

Typical output from a DE analysis:

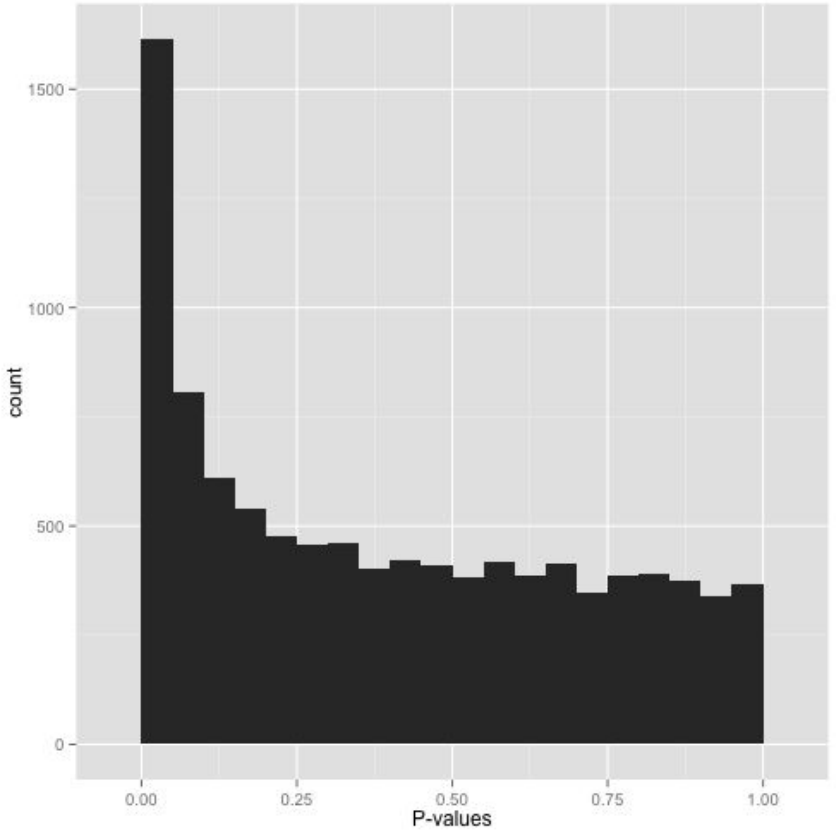
- log2 fold-change and
- adjusted *p*-value per compared genes per compared conditions

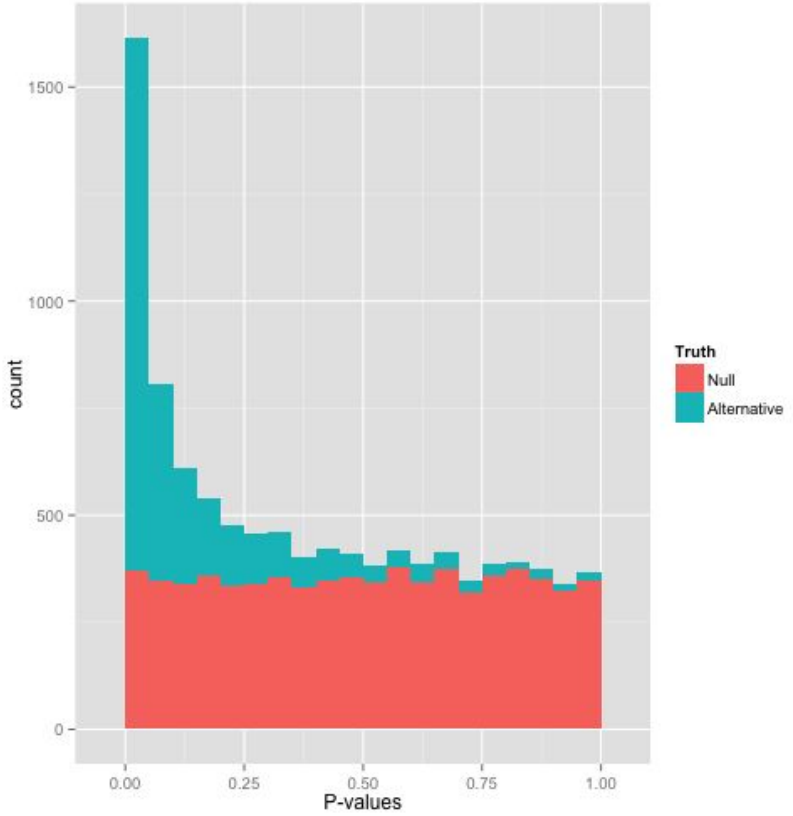
This list can then be sorted by p-value and investigated in more detail with volcano plots



Check your p-values!

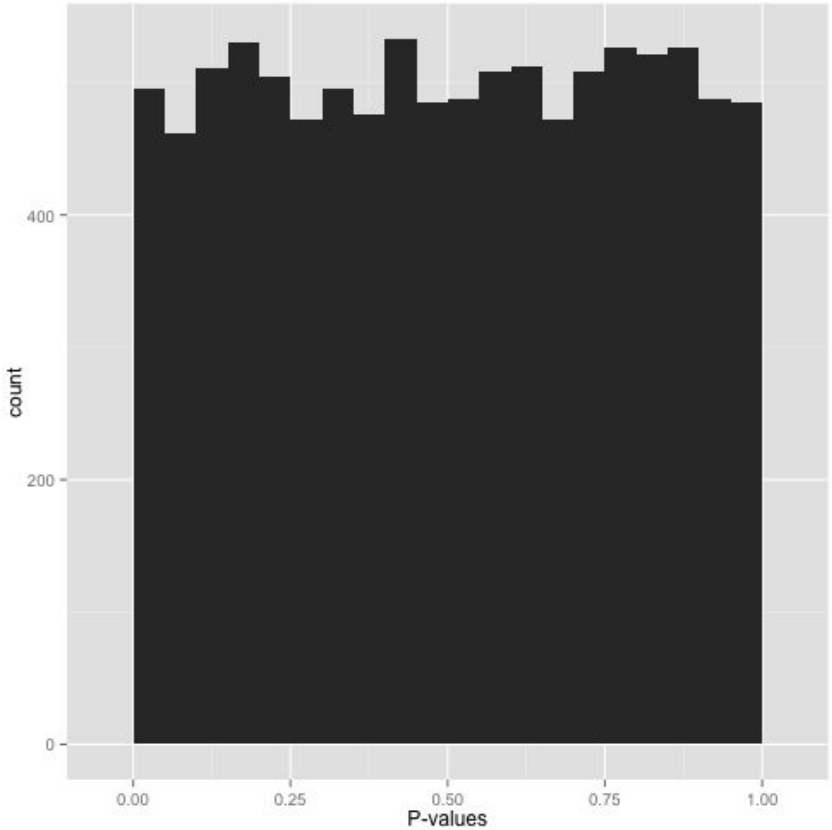
Anti-conservative p-values
Hooray!





http://varianceexplained.org/statistics/interpreting-pvalue-histogram

Check your p-values!

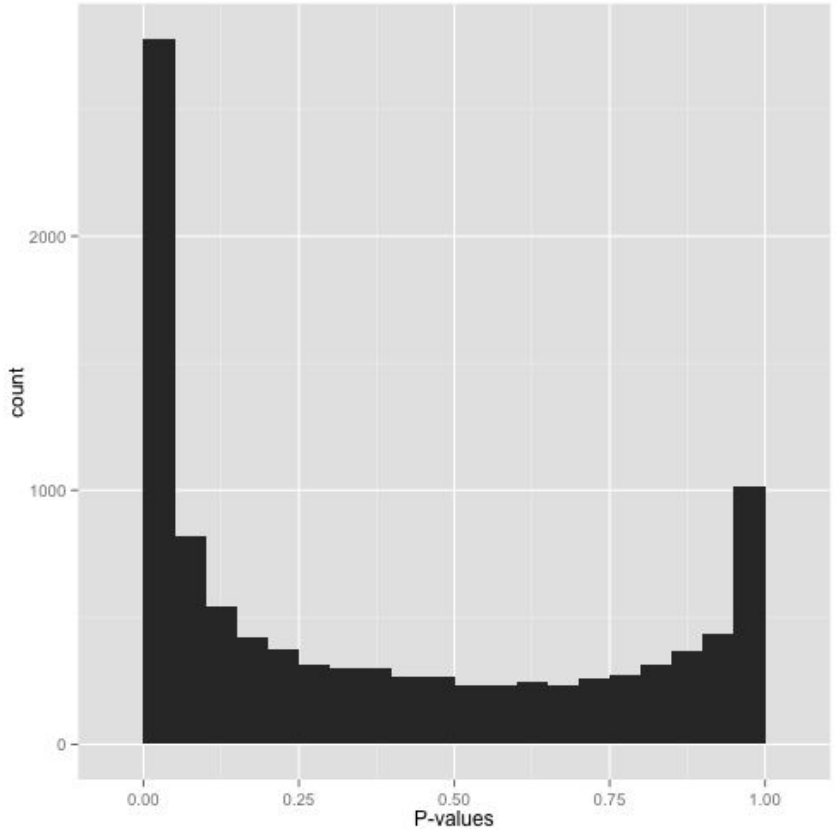


Uniform p-values

awww (sad face)

http://varianceexplained.org/statistics/interpreting-pvalue-histogram

Check your p-values!



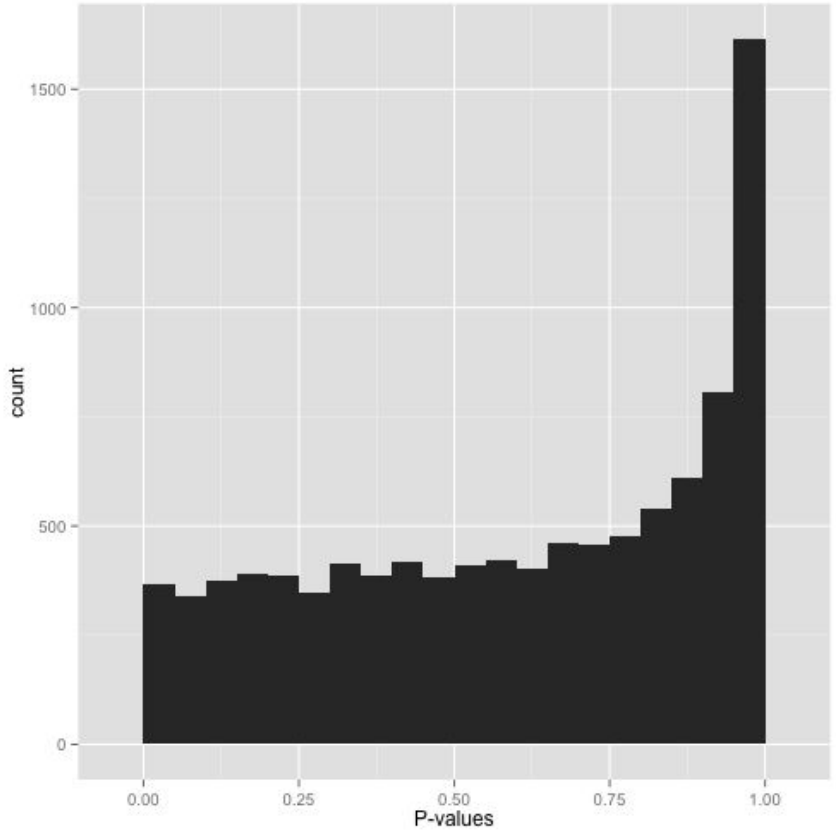
Bimodal p-values

Hmm…

Don’t apply FDR methods yet!
P-val dist should be uniform; explore your p-values

http://varianceexplained.org/statistics/interpreting-pvalue-histogram

Check your p-values!



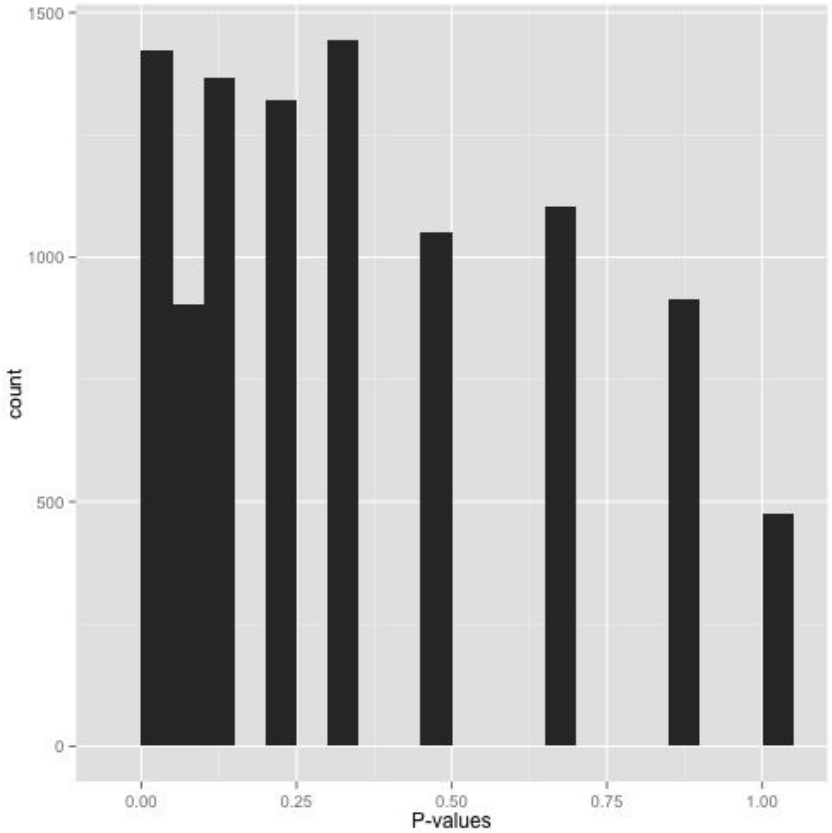
Conservative p-values

Whoops!

P-val dist should be uniform; something is likely wrong with your test!

http://varianceexplained.org/statistics/interpreting-pvalue-histogram

Check your p-values!



Sparse p-values

Hold on...

Gaps mean you have only a small # of distinct p-value

http://varianceexplained.org/statistics/interpreting-pvalue-histogram

Korthauer *et al. Genome Biology* (2019) 20:118
https://doi.org/10.1186/s13059-019-1716-1

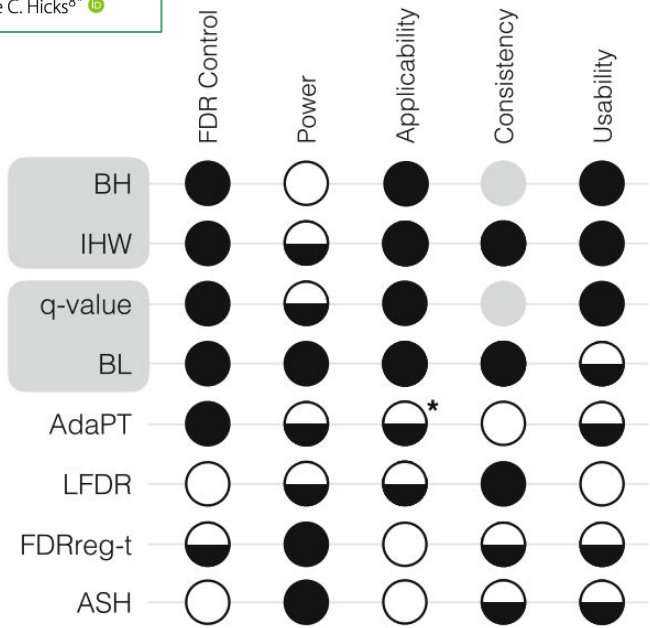
Genome Biology

RESEARCH Open Access

A practical guide to methods controlling false discoveries in computational biology

Keegan Korthauer[1,2†], Patrick K. Kimes[1,2†], Claire Duvallet[3,4†], Alejandro Reyes[1,2†], Ayshwarya Subramanian[5†], Mingxiang Teng[6], Chinmay Shukla[7], Eric J. Alm[3,4,5] and Stephanie C. Hicks[8*]

	Input	Assumptions	Output	R package
BH	*p*-values	exchangeability	adjusted *p*-values	`stats`
IHW	(1) *p*-values (2) independent & informative covariate	exchangeability within covariate groups		`ihw`
q-value	*p*-values	exchangeability	*q*-values	`qvalue`
BL	(1) *p*-values (2) independent & informative covariate	exchangeability conditional on covariate(s)	adjusted *p*-values	`swfdr`
AdaPT			*q*-values	`adaptMT`
LFDR		exchangeability within covariate groups	adjusted *p*-values	none
FDRreg	(1) *z*-scores (2) independent & informative covariate	exchangeability conditional on covariate(s); normal test statistics	Bayesian FDRs	`FDRreg`
ASH	(1) effect sizes (2) standard errors of (1)	effects are unimodal; test statistics have normal or *t* mixture components	*q*-values	`ash`



https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1716-1

Want to learn more about scRNAseq data analysis?

- Book in Python/scanpy (https://www.sc-best-practices.org)
- Book in R/Bioconductor (https://bioconductor.org/books/OSCA) (paper)
  - 14 end-to-end scRNAseq data analyses in R/Bioconductor (https://bioconductor.org/books/release/OSCA.workflows)
- Book in R/Seurat (https://satijalab.org/seurat/articles/get_started_v5_new)

Spatially-resolved transcriptomics

Spatially-resolved transcriptomics

2000 µm

High-throughput spatially resolved transcriptomics

Can view as extension of single-cell RNA sequencing with added spatial information

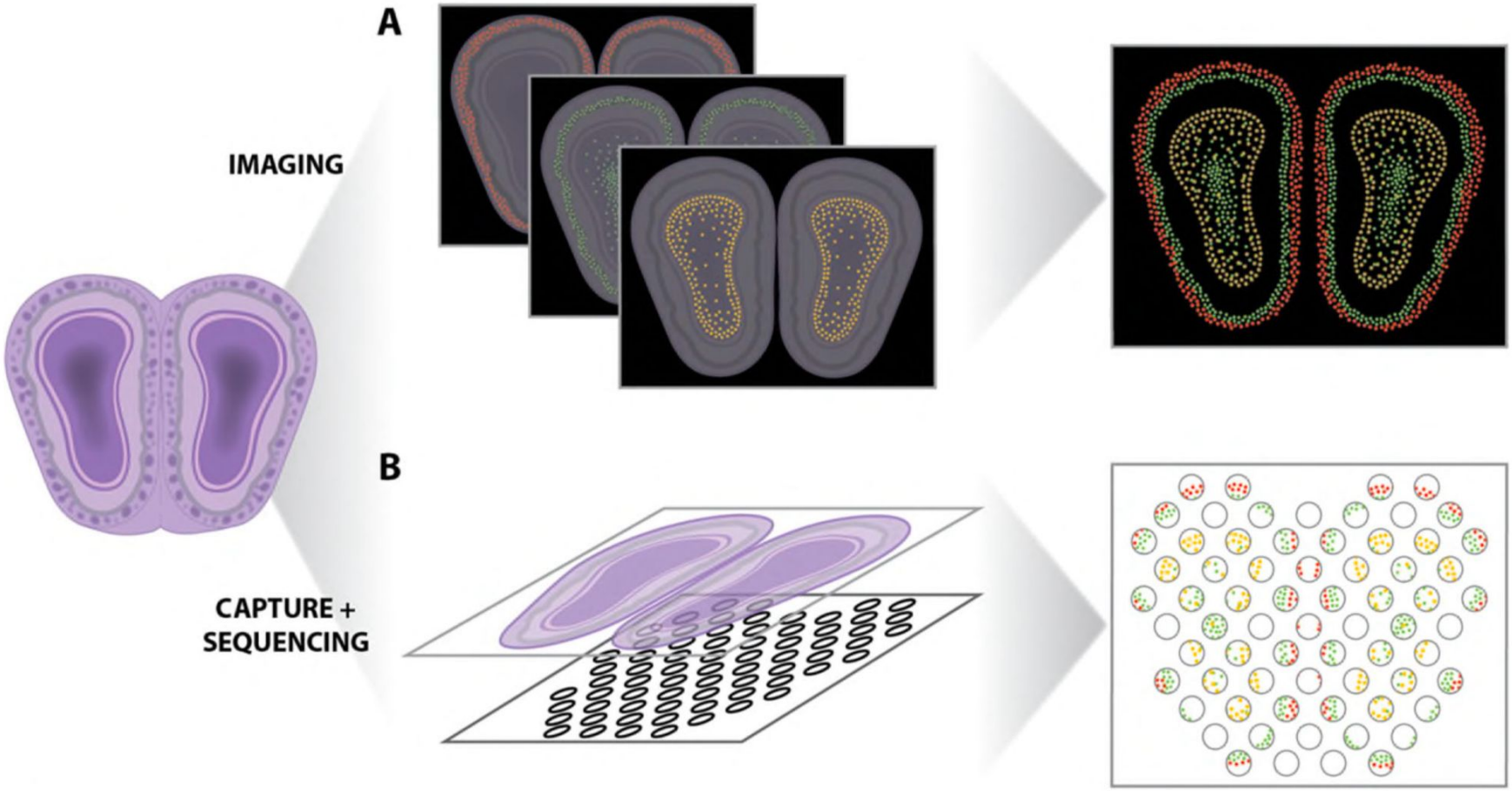
Example: cell populations within the brain

bulk RNA-seq

single-cell RNA-seq

spatially-resolved transcriptomics

Image credit: @BoXia7

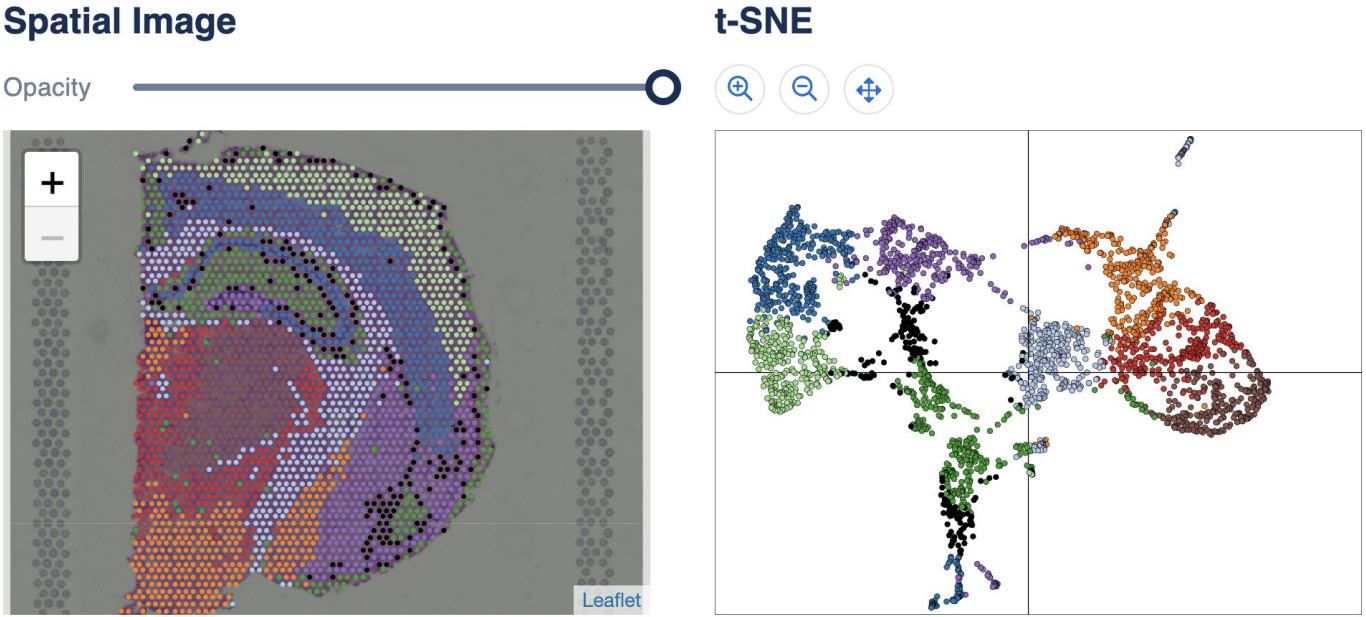


Atta and Fan (2021). *Nature Communications*

Examples of spatially-resolved transcriptomics data

Spatially-resolved transcriptomics

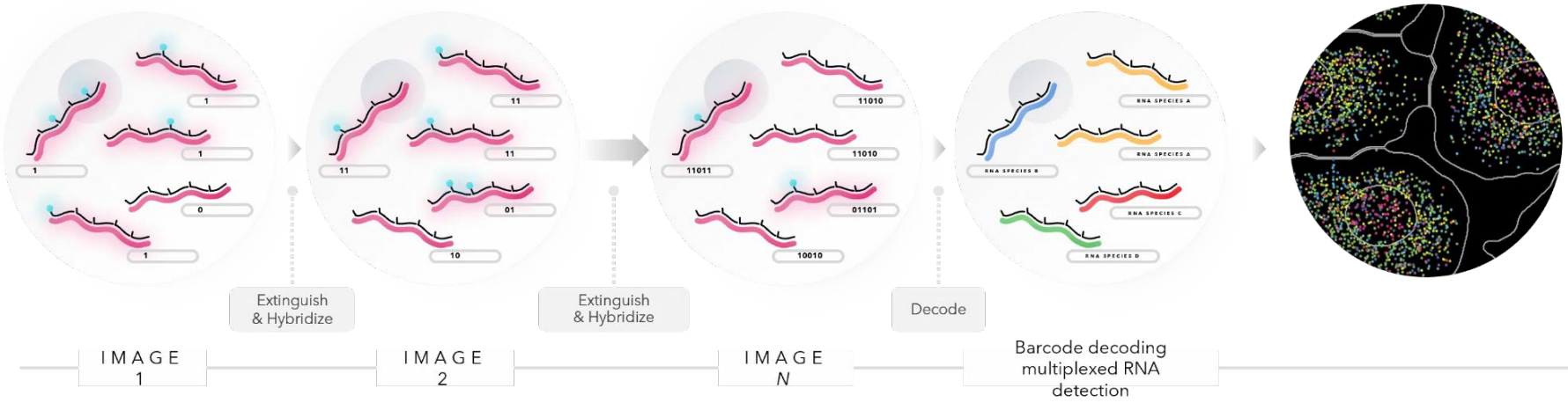
Example: 10x Genomics Visium platform



https://www.10xgenomics.com/spatial-transcriptomics

Spatially-resolved transcriptomics

Example: MERFISH

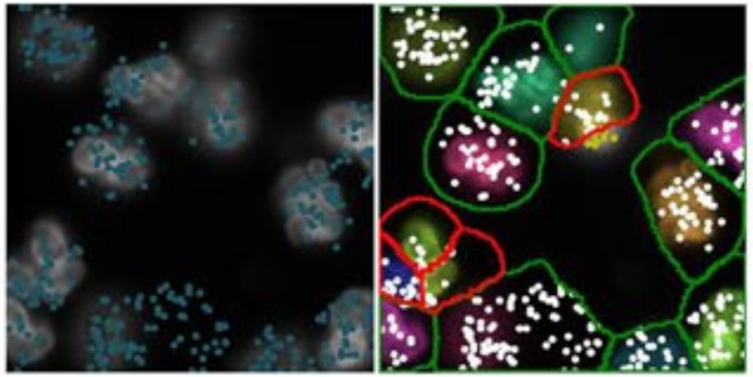


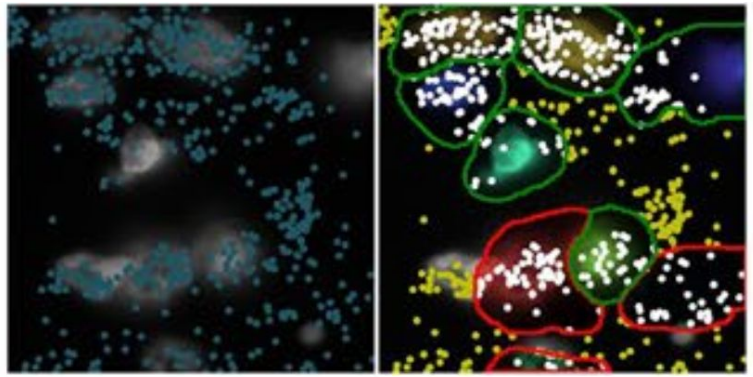
https://vizgen.com/technology

Preprocessing spot-based (and image-based) data

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-023-03054-0

Preprocessing steps: cell / nuclei segmentation

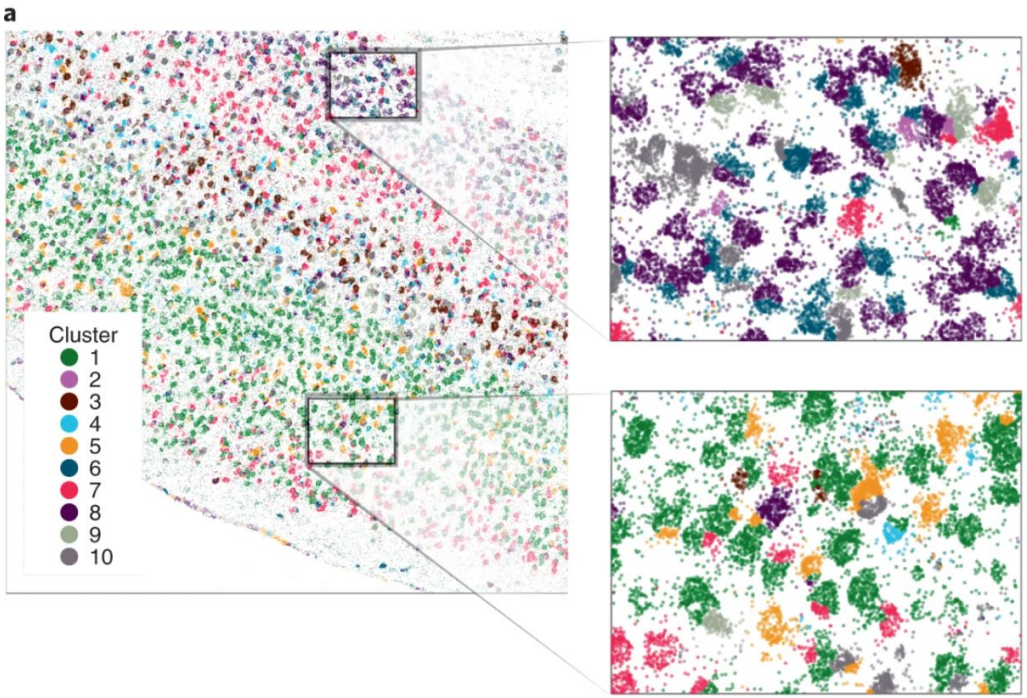


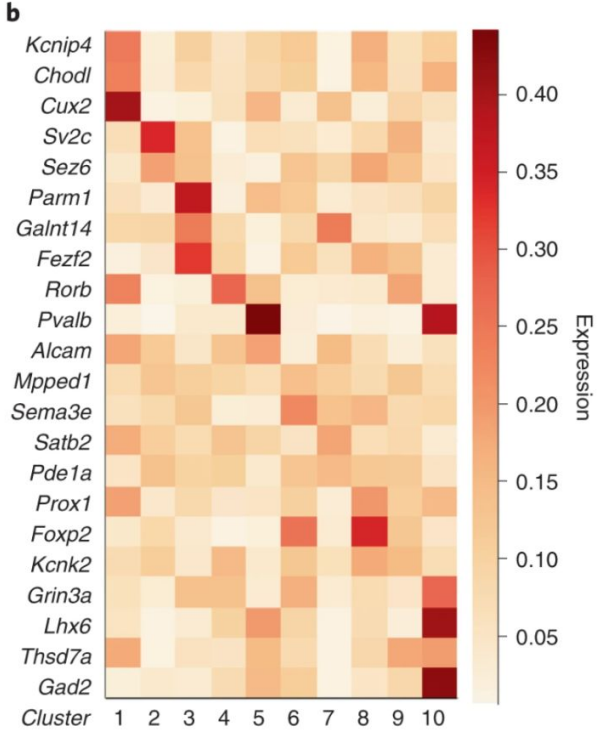


Broad approaches:

1. **Designed for biomedical images** (e.g. Watershed, U-Net, Deepcell, and Cellpose) -- often the image doesn't contain enough info to segment cell boundary (mostly stains for just nucleus)
2. **Uses spatial locations of RNA reads** to infer cell boundaries (e.g. Baysor) -- often ignores the image or makes unrealistic assumptions
3. **Integrates both** images and spatial locations of RNA reads

Segmentation-free cell type inference

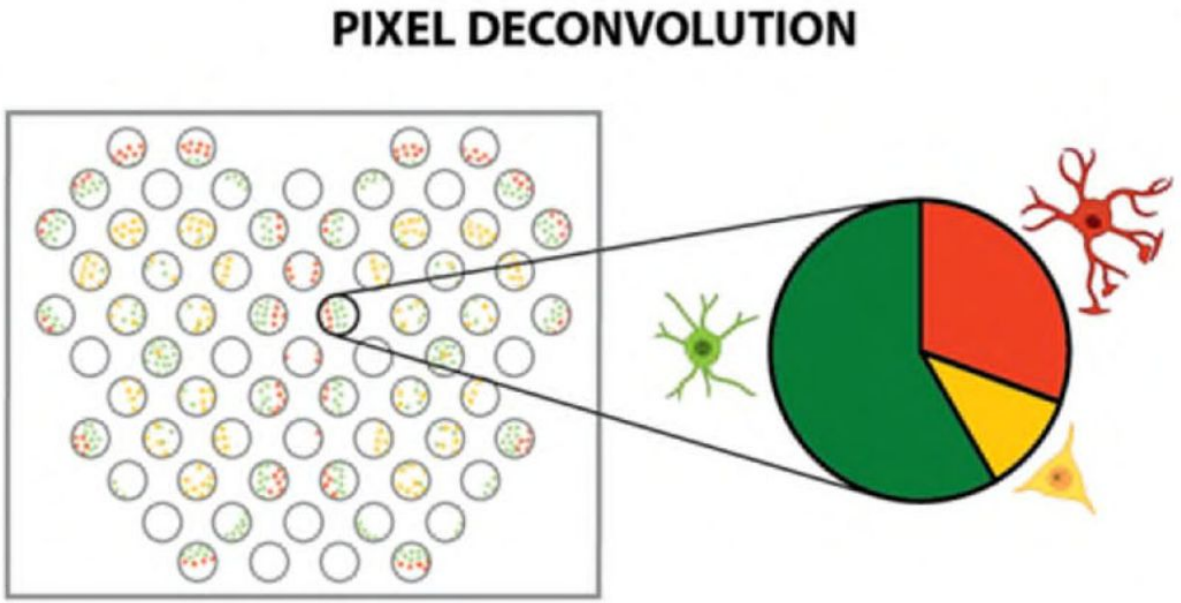




https://www.nature.com/articles/s41587-021-01044-w

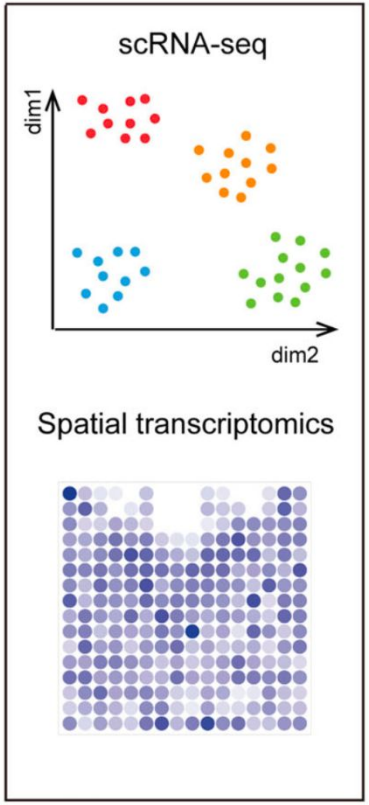
Types of questions?

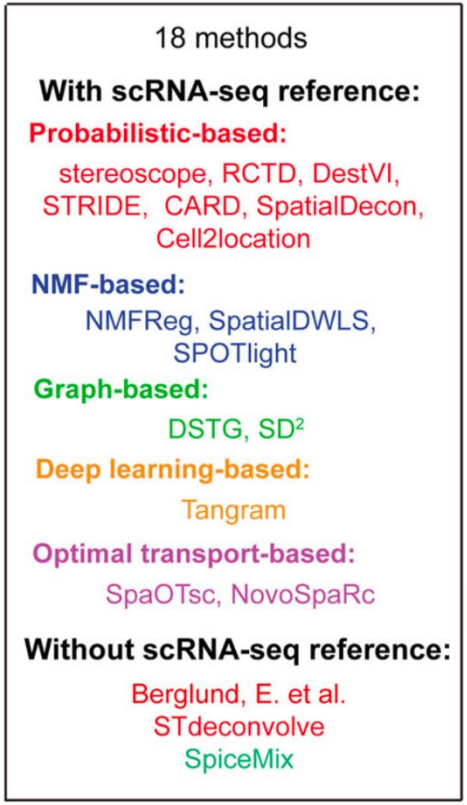
Deconvolve multi-cellular pixel-resolution data to determine pixel cell-type composition at each observed spatial location (sometimes called “deconvolution”)



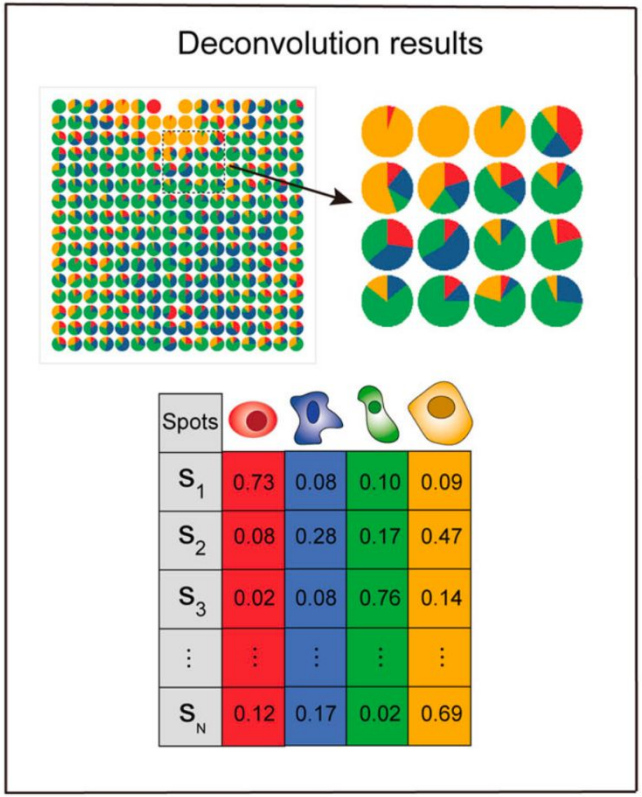
Atta and Fan (2021)

Spot-level deconvolution benchmark evaluation





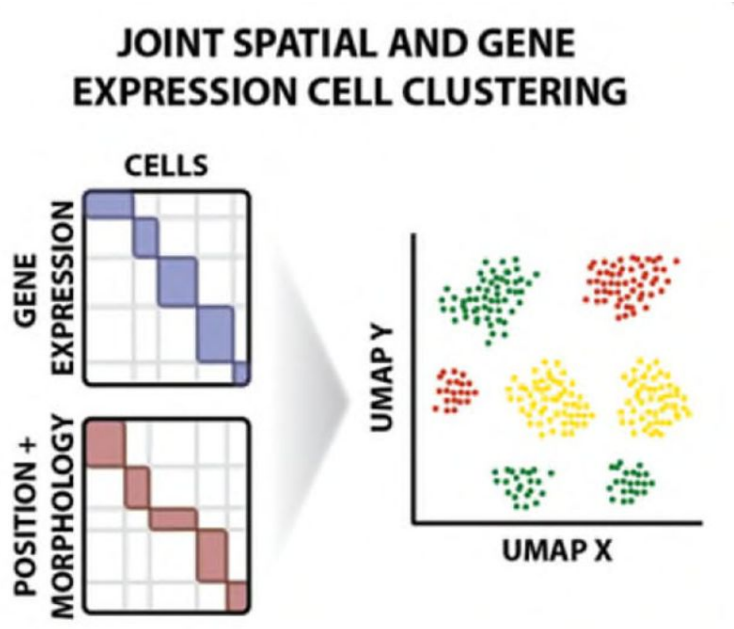
50 datasets



https://www.nature.com/articles/s41467-023-37168-7

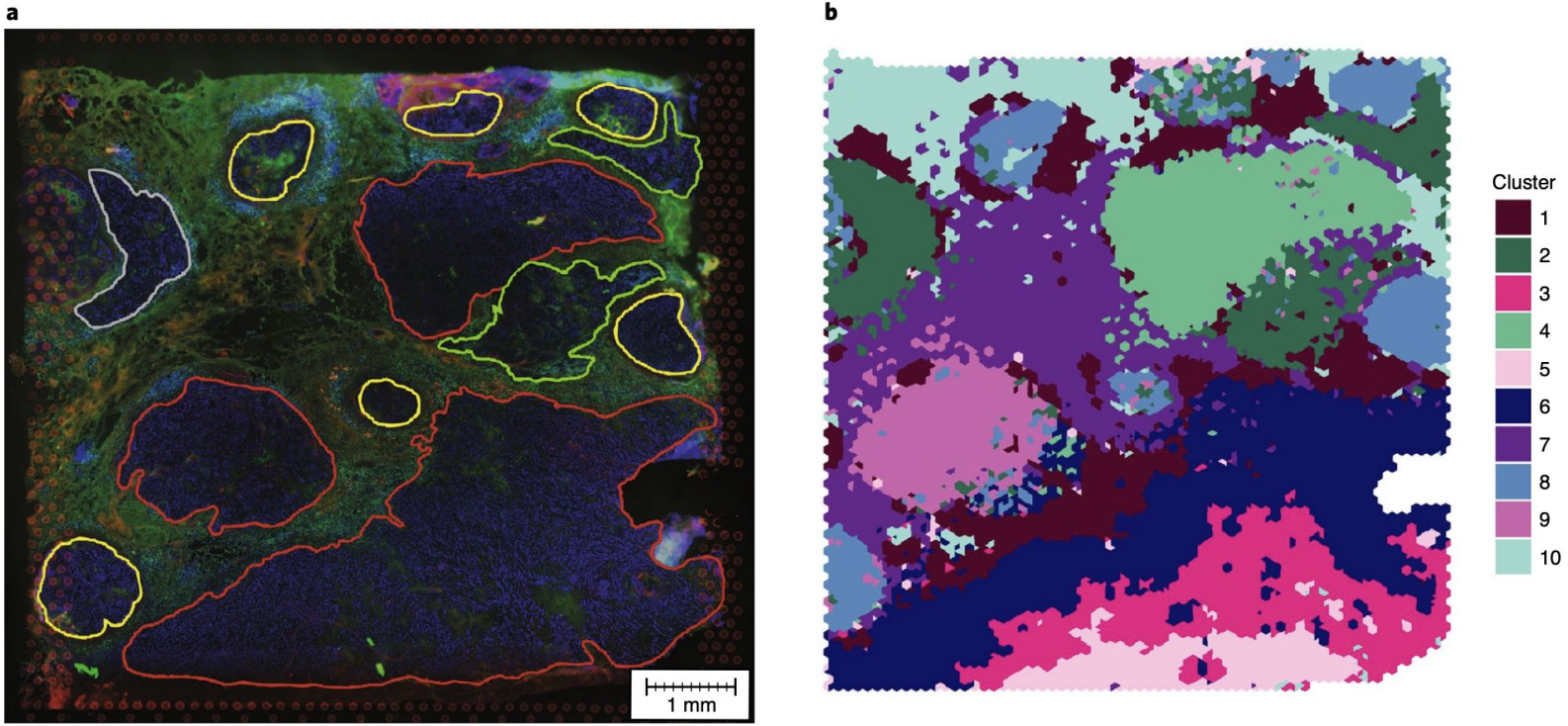
Types of questions?

Spatial clustering = combine gene expression, position, and morphological information to cluster cell populations



Atta and Fan (2021)

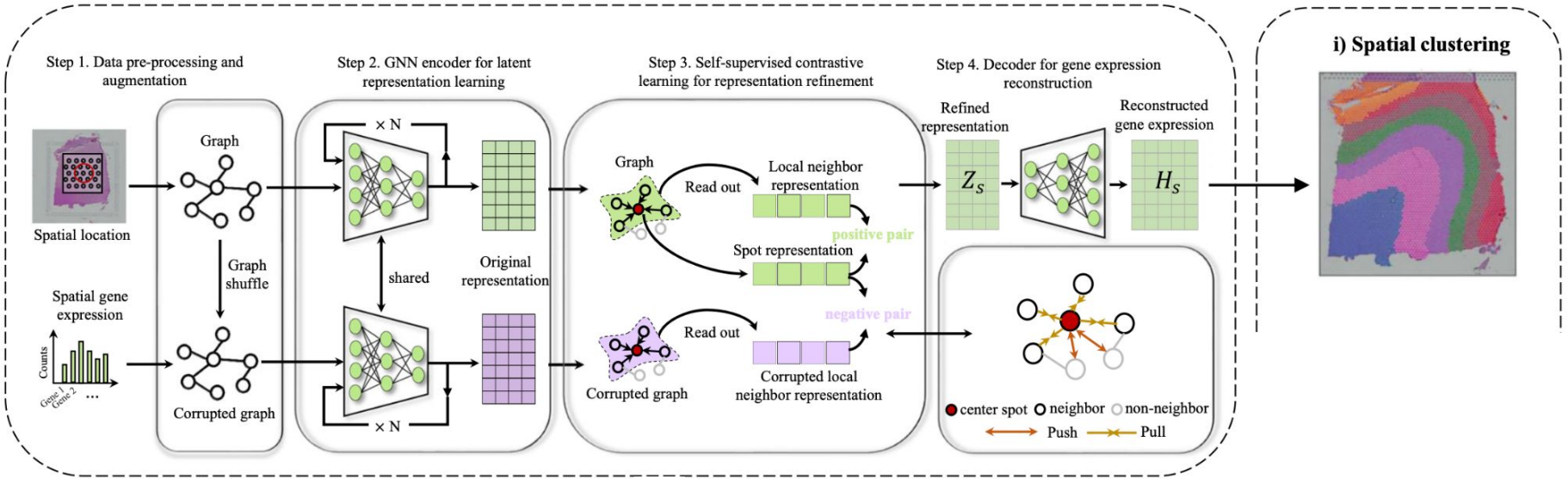
Spatial clustering with BayesSpace



Fully Bayesian statistical approach with a Markov random field

https://www.nature.com/articles/s41587-021-00935-2

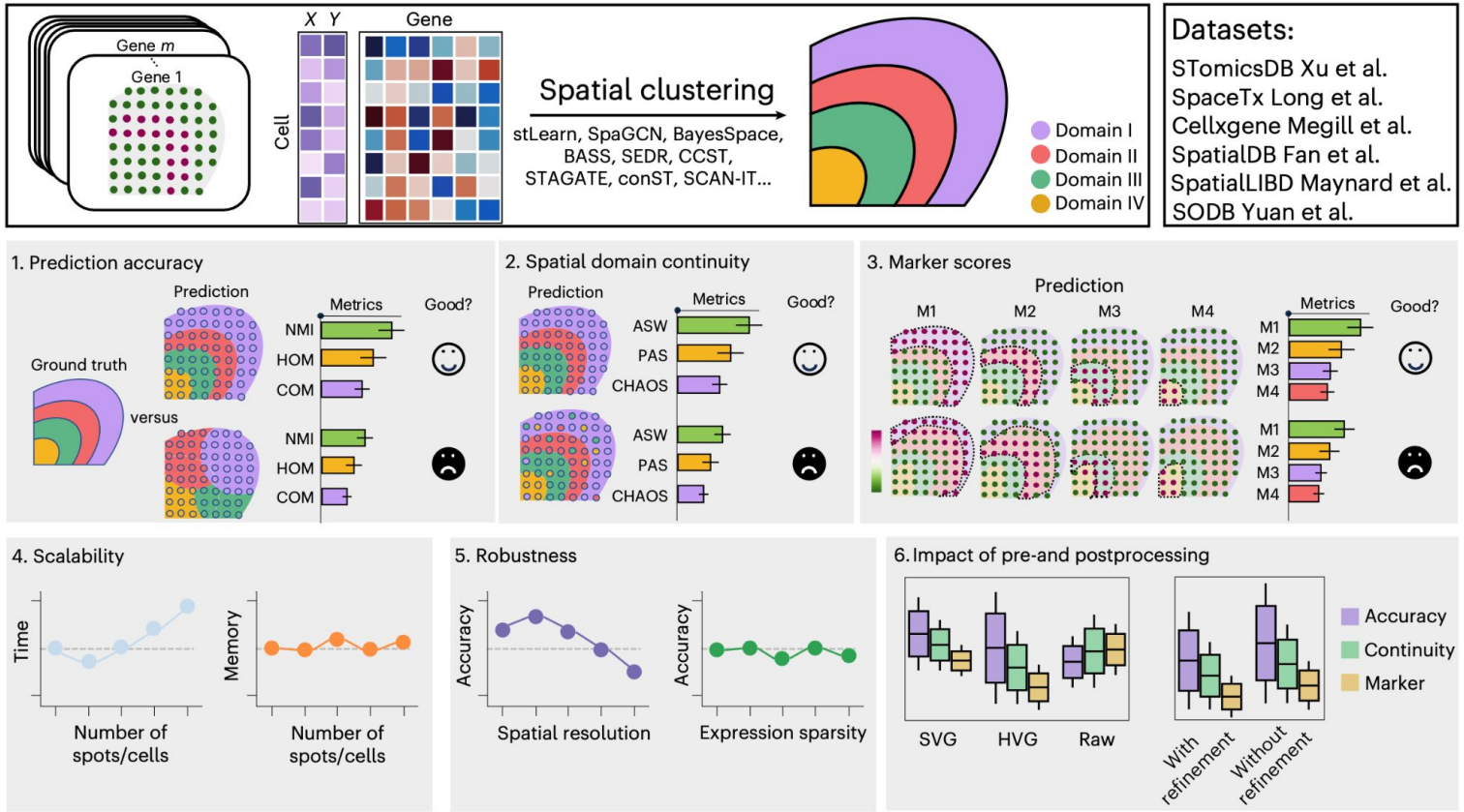
Spatial clustering with GraphST



GraphST = graph-based convolutional autoencoder (and refines latent embeddings with contrastive self-supervised learning)

https://www.nature.com/articles/s41592-024-02215-8

Spatial clustering benchmark evaluation



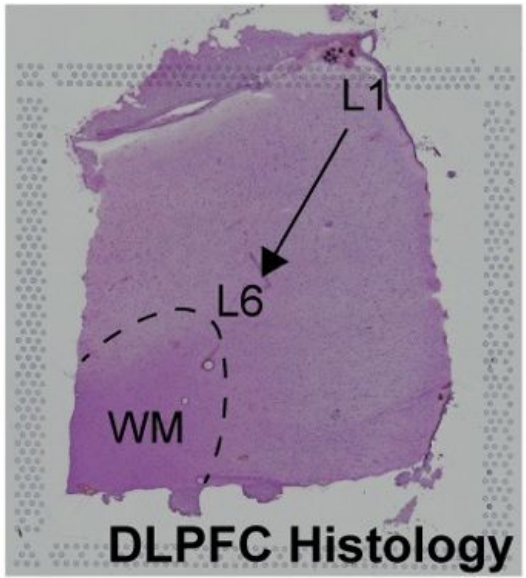
https://www.nature.com/articles/s41592-024-02215-8

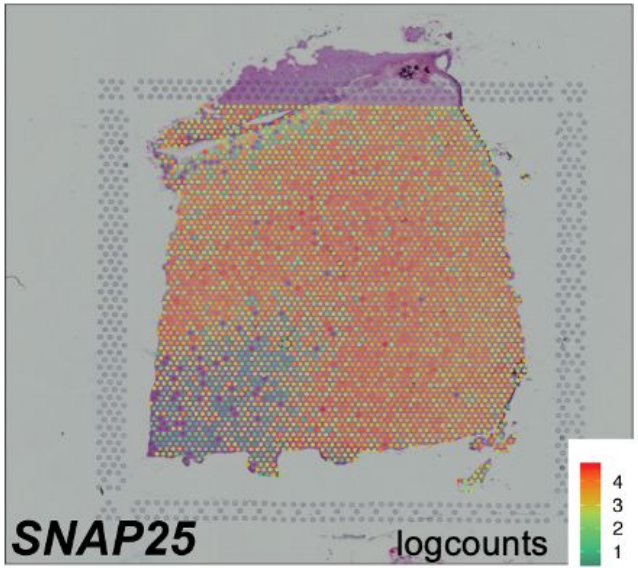
Other important challenges for spatial omics

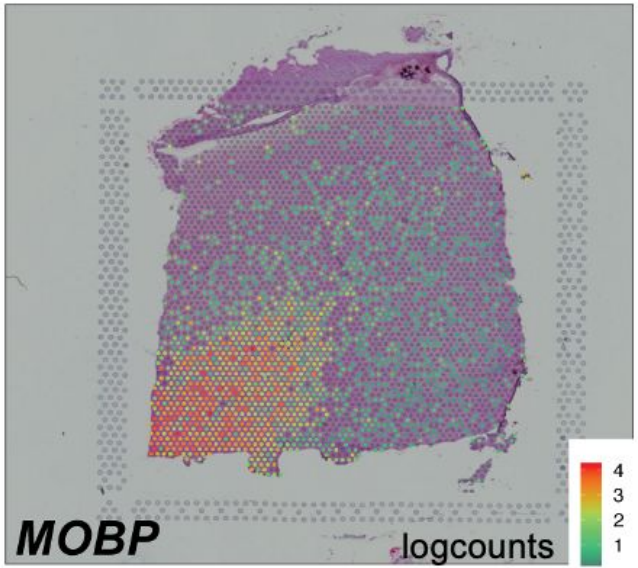
- Cell segmentation remains one of the biggest challenges for spatial transcriptomics (e.g. neurons are not round and can be up to a meter in length in humans)
- QC and normalization (e.g. 0, 1, or multiple cells per spatial location)
- Cell type-specific differential expression (e.g. DE could be due to changes in cell type composition across space and the fact that measurement units often detect transcripts from more than one cell type)
- Enhancement of resolution (e.g. predict spatial expression at a finer resolution)
- Spatial registration / alignment (e.g. align spatial replicates OR align scRNAseq to spatial maps)
- Measuring and predicting different data modalities (RNA and protein) in 2D
- ML model to predict gene expression from images (e.g. brightfield H&E images)

https://arxiv.org/abs/2306.01859

Relationship between H&E brightfield image and gene expression? Yes!







Can we train ML models to predict gene expression from a H&E image (these images are widely used in pathology and get info for free)?


Thank you and wishing you all a great rest of the semester! If you want to a local place to explore this time of year, check out Patapsco Valley State Park -- one my favorites!