

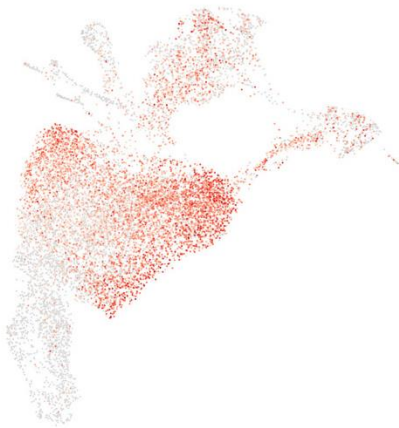
# Single Cell Data Analysis

CMM262-2020

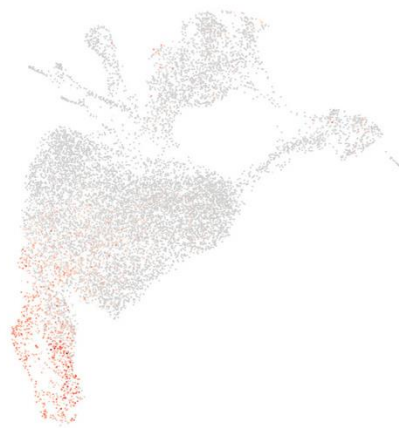
Rob Morey

remorey@eng.ucsd.edu

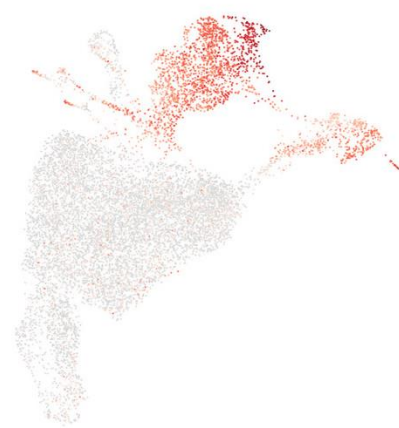
**Slc12a2**



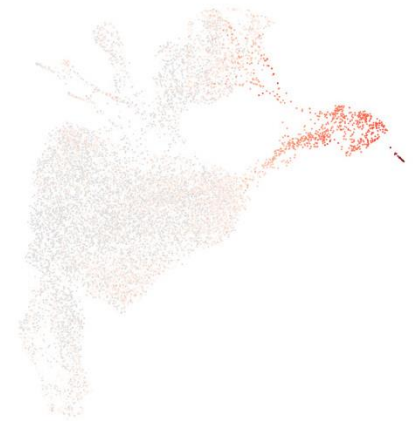
**Arg2**



**Tff3**



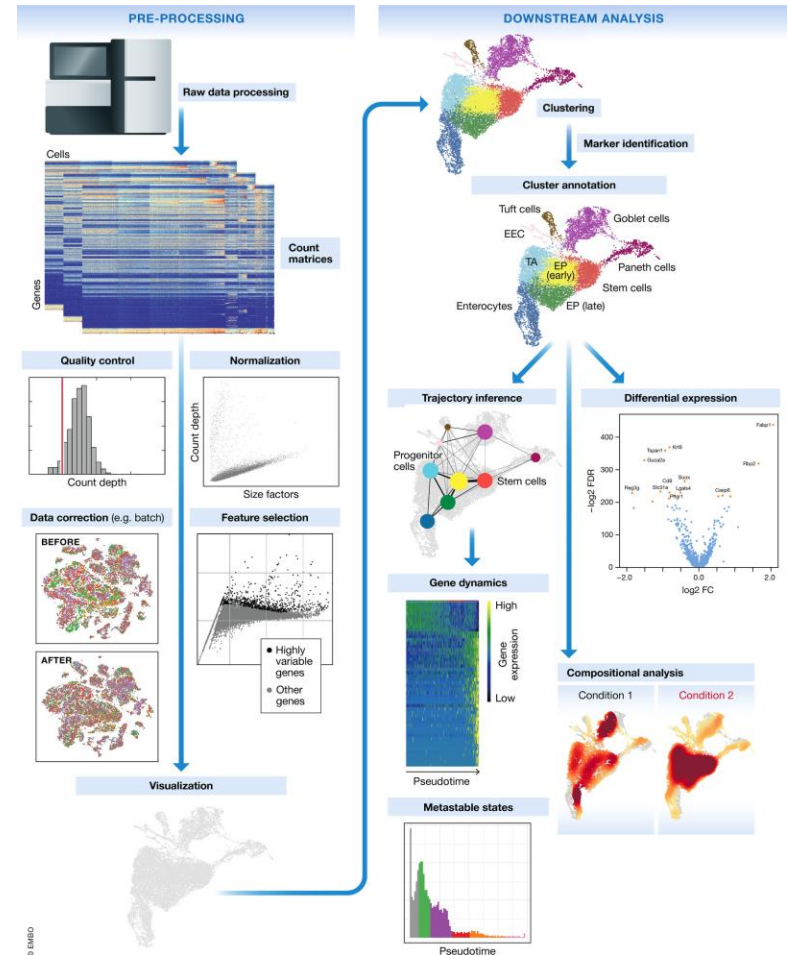
**Defa24**



# Typical Single-cell RNA-seq Analysis Workflow

## Workflow:

- Filter for good cells and detected genes (arbitrary cutoffs)
- Normalize/Scale Data
- Remove unwanted sources of variation (batch/cell cycle)
- Feature selection, dimensionality reduction and visualization
- Feature selection (highly variable genes)
- Dimensionality Reduction – summarization (describe data in as few dimensions as possible for downstream)
- Dimensionality Reduction – visualization (describe data in 2D or 3D)
- Clustering (grouping cells based on expression profiles)
- Define Cell-Type specific signatures through cluster annotation.
- Trajectory analysis (transitions between cell identities)
- Unification between clustering and trajectory inference - (partition-based graph abstraction - PAGA)



SOFTWARE

Open Access



# SCANPY: large-scale single-cell gene expression data analysis

F. Alexander Wolf<sup>1\*</sup> , Philipp Angerer<sup>1</sup> and Fabian J. Theis<sup>1,2\*</sup>

*Review*




molecular  
systems  
biology

## Current best practices in single-cell RNA-seq analysis: a tutorial

Malte D Luecken<sup>1</sup>  & Fabian J Theis<sup>1,2,\*</sup> 

<https://github.com/theislab/single-cell-tutorial/>

<https://scanpy.readthedocs.io/en/latest/index.html>

 Scanpy  
latest

[Tutorials](#)  
[Usage Principles](#)  
[Installation](#)  
[API](#)  
[References](#)

Docs » Scanpy – Single-Cell Analysis in Python [Edit on GitHub](#)


pypi v1.3.7

docs passing

build passing

install with bioconda

## Scanpy – Single-Cell Analysis in Python



Scanpy is a scalable toolkit for analyzing single-cell gene expression data. It includes preprocessing, visualization, clustering, pseudotime and trajectory inference and differential expression testing. The Python-based implementation efficiently deals with datasets of more than one million cells.

Report issues and see the code on [GitHub](#). If Scanpy is useful for your research, consider citing [Genome Biology \(2018\)](#).

# Usage Principles

Import the Scanpy API as:

```
import scanpy.api as sc
```

## Workflow

The typical workflow consists of subsequent calls of data analysis tools in `sc.tl`, e.g.:

```
sc.tl.tsne(adata, **tool_params) # embed the data using tSNE
```

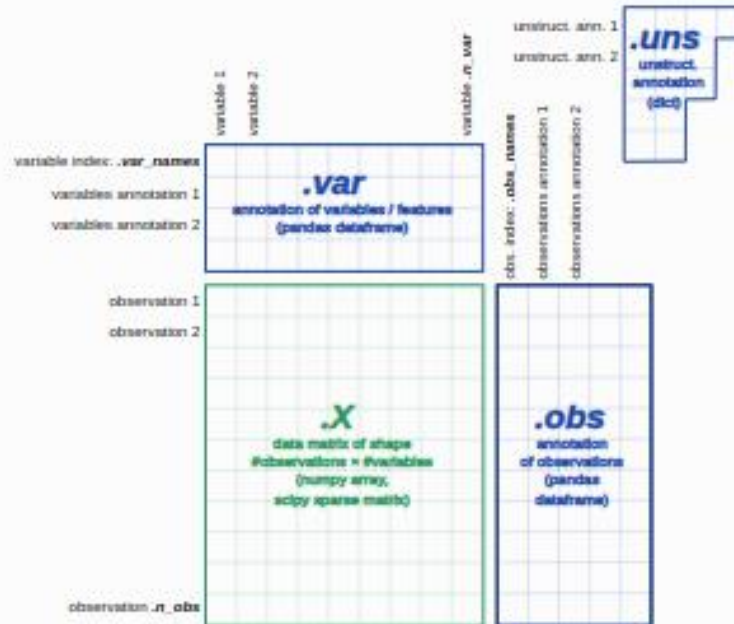
where `adata` is an `AnnData` object. Each of these calls adds annotation to an expression matrix  $X$ , which stores  $n\_obs$  observations (cells) of  $n\_vars$  variables (genes). For each tool, there typically is an associated plotting function in `sc.pl`:

```
sc.pl.tsne(adata, **plotting_params)
```

If you pass `show=False`, a `matplotlib.axes.Axes` instance is returned and you have all of matplotlib's detailed configuration possibilities.

# AnnData

Scanpy is based on `anndata`, which provides the `AnnData` class.



At the most basic level, an `AnnData` object `adata` stores a data matrix (`adata.X`), dataframe-like annotation of observations (`adata.obs`) and variables (`adata.var`) and unstructured dict-like annotation (`adata.uns`). Values can be retrieved and appended via `adata.obs['key1']` and `adata.var['key2']`. Names of observations and variables can be accessed via `adata.obs_names` and `adata.var_names`, respectively. `AnnData` objects can be sliced like dataframes, for example, `adata_subset = adata[:, list_of_gene_names]`.

[http://falexwolf.de/blog/171223\\_AnnData\\_indexing\\_views\\_HDF5-backing/](http://falexwolf.de/blog/171223_AnnData_indexing_views_HDF5-backing/)



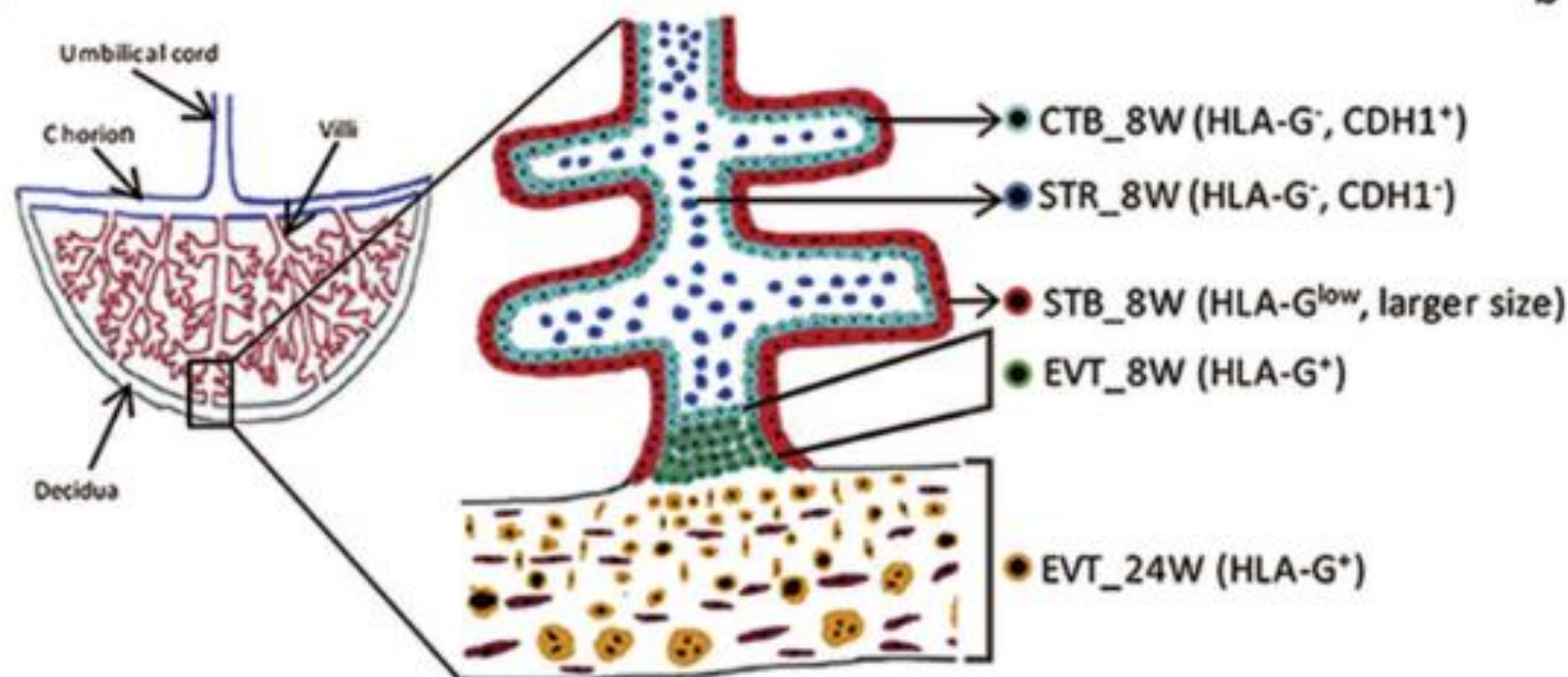


## ARTICLE OPEN

## Single-cell RNA-seq reveals the diversity of trophoblast subtypes and patterns of differentiation in the human placenta

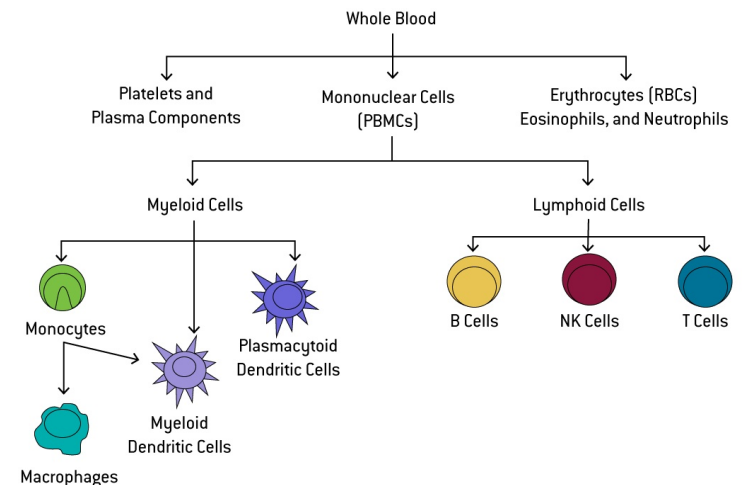
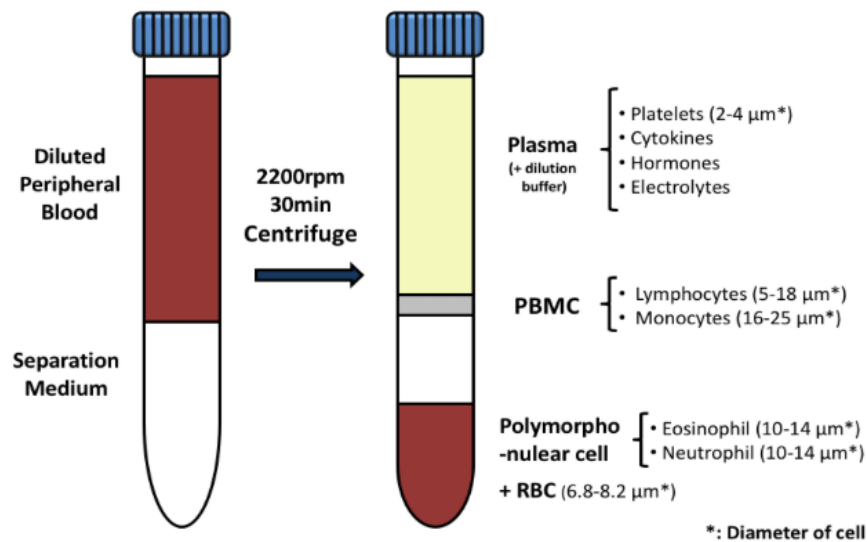
Yawei Liu<sup>1</sup>, Xiaoying Fan<sup>2</sup>, Rui Wang<sup>2</sup>, Xiaoyin Lu<sup>1,3</sup>, Yan-Li Dang<sup>4</sup>, Huiying Wang<sup>5</sup>, Hai-Yan Lin<sup>1</sup>, Cheng Zhu<sup>1</sup>, Hao Ge<sup>1,2</sup>, James C. Cross<sup>6</sup> and Hongmei Wang<sup>1</sup>

b



<https://support.10xgenomics.com/single-cell-gene-expression/datasets/>

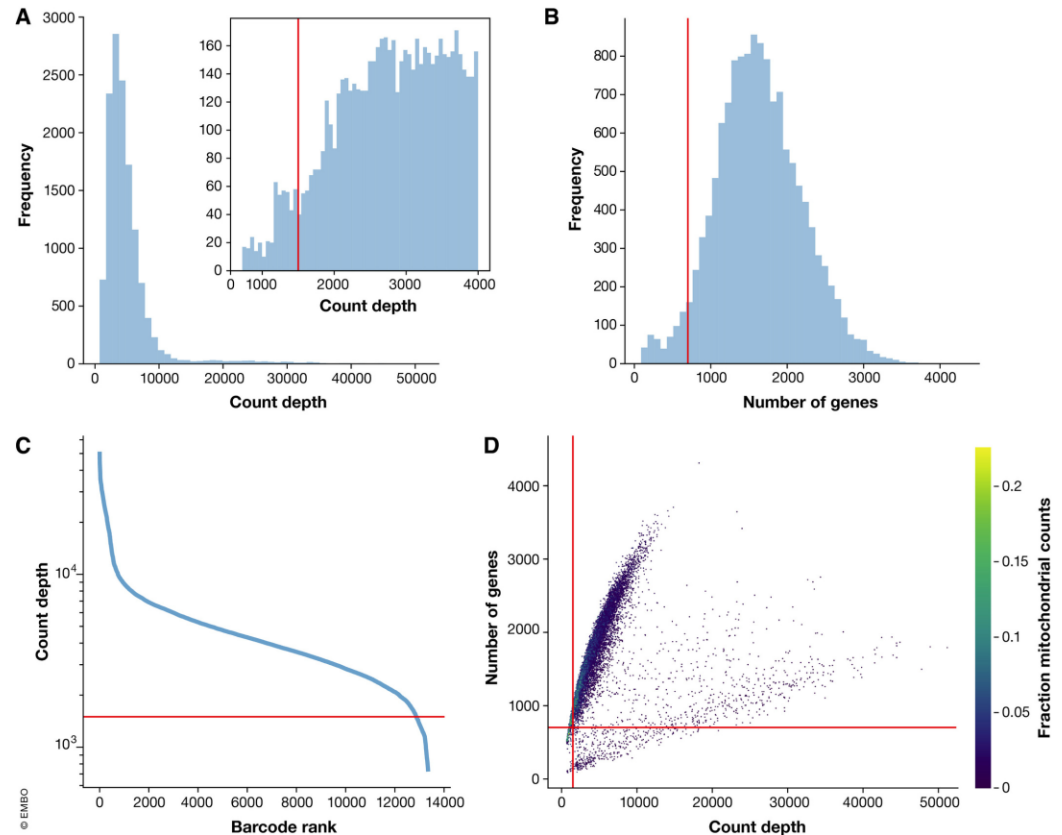
## Single Cell Gene Expression Datasets





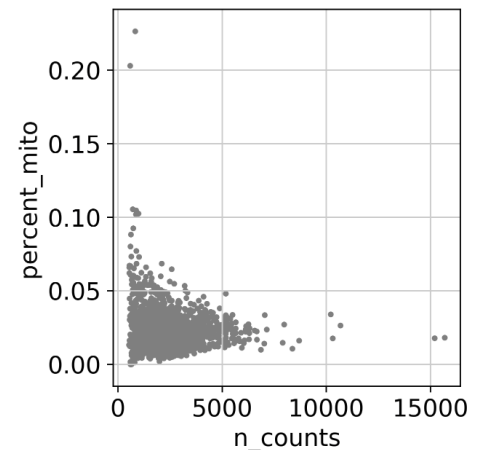
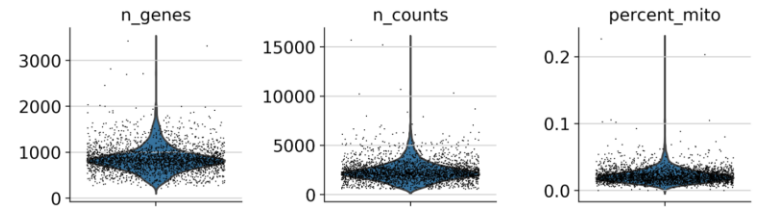
# Preprocessing: Raw data processing and filtering

- Raw data processing pipelines like Cell Ranger assign reads to cells, align reads to genome, and create count matrices.
- Cell QC performed on three QC covarites:
  - 1) Count Depth (number of reads per cell barcode)
  - 2) Number of genes per barcode
  - 3) Fraction of counts from mitochondrial genes



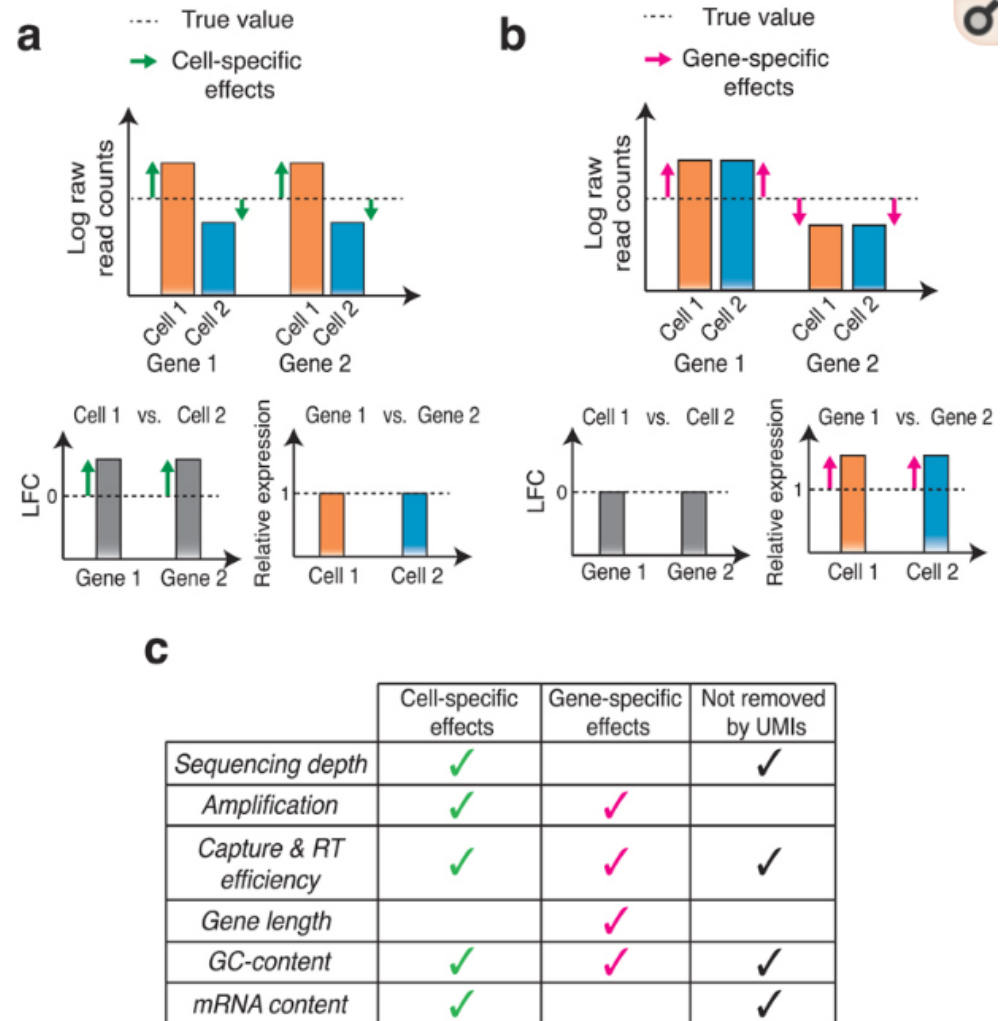
# Remove outlier cells

- A guideline to setting thresholds is to use the minimum cell cluster size that is of interest and leaving some leeway for dropout effects.
- Be permissive and revisit QC after clustering



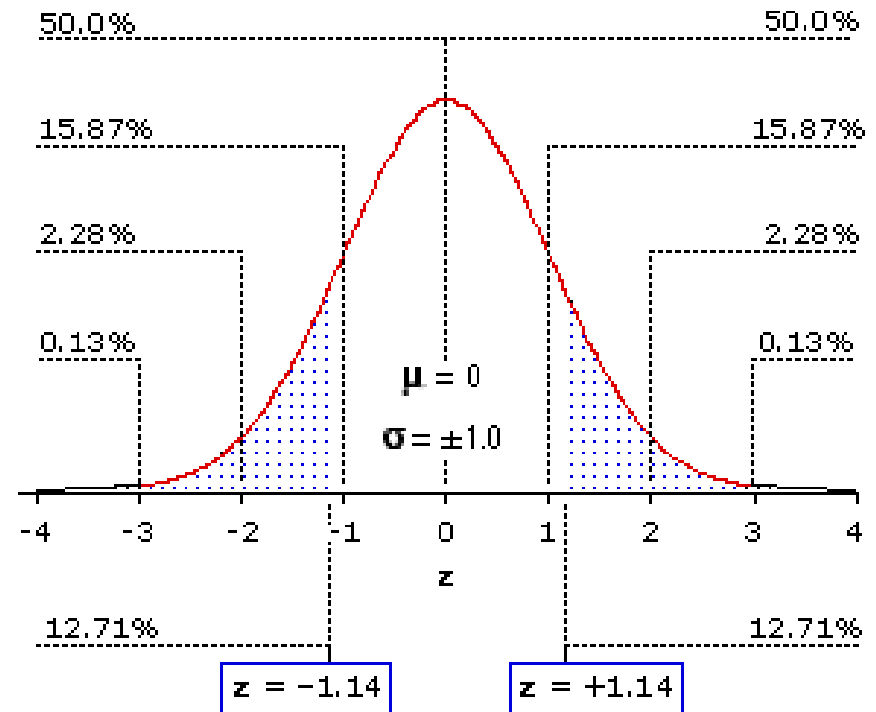
# Normalization

- How do you normalize for sequencing depth?
- Should you normalize for sequencing depth in single cell data?
- Is Length an important normalization step here?
- Should you perform gene scaling (weight genes equally)?
- Log transformation? - allows for normal dist. assumption



## Z-score scaling for filtering in variable genes:

- Z-score is the number of standard deviations away from the mean
- What is the expected sum of all z-scores for a given gene?
- Purpose is to scale expression of each gene relative to all the cells
- The preference between the two choices revolves around whether all genes should be weighted equally for downstream analysis, or whether the magnitude of expression of a gene is an informative proxy for the importance of the gene.



# Overcoming systematic errors caused by log-transformation of normalized single-cell RNA sequencing data

Aaron Lun<sup>1,\*</sup>

**1 Cancer Research UK Cambridge Institute, University of Cambridge, Li Ka Shing Centre, Robinson Way, Cambridge CB2 0RE, United Kingdom**

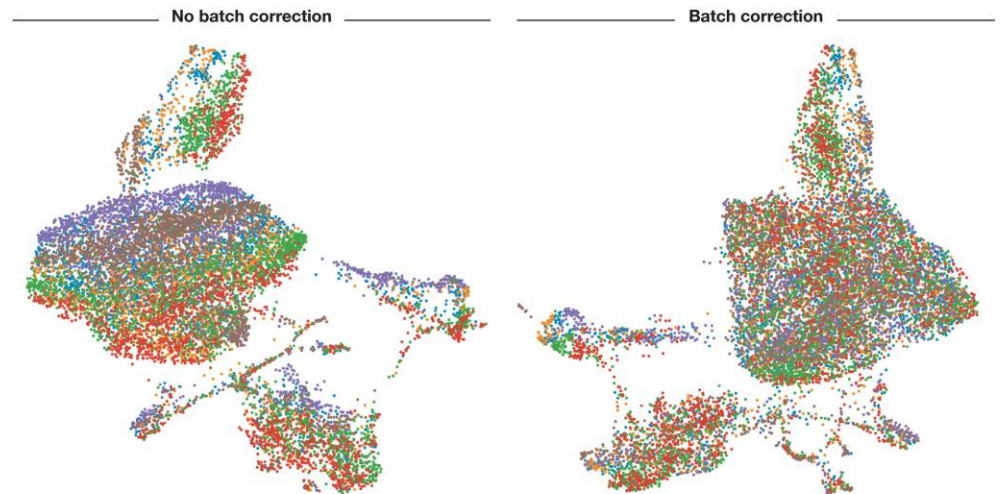
**\* Email: [aaron.lun@cruk.cam.ac.uk](mailto:aaron.lun@cruk.cam.ac.uk)**

## Abstract

Applying a log-transformation to normalized expression values is one of the most common procedures in exploratory analyses of single-cell RNA sequencing (scRNA-seq) data. Normalization removes systematic biases in sequencing coverage between cells, while the log-transformation ensures that downstream computational procedures operate on relative rather than absolute differences in expression. We show that the log-transformation can introduce systematic errors when cells vary in sequencing coverage, leading to spurious non-zero differences in expression and artificial population structure in simulations. We observe similar effects in real scRNA-seq data where the difference in transformed values between groups of cells is not an accurate proxy for the log-fold change. We provide some practical recommendations to overcome this effect and analytically derive an expression for a larger pseudo-count that controls the transformation-induced error to a specified threshold.

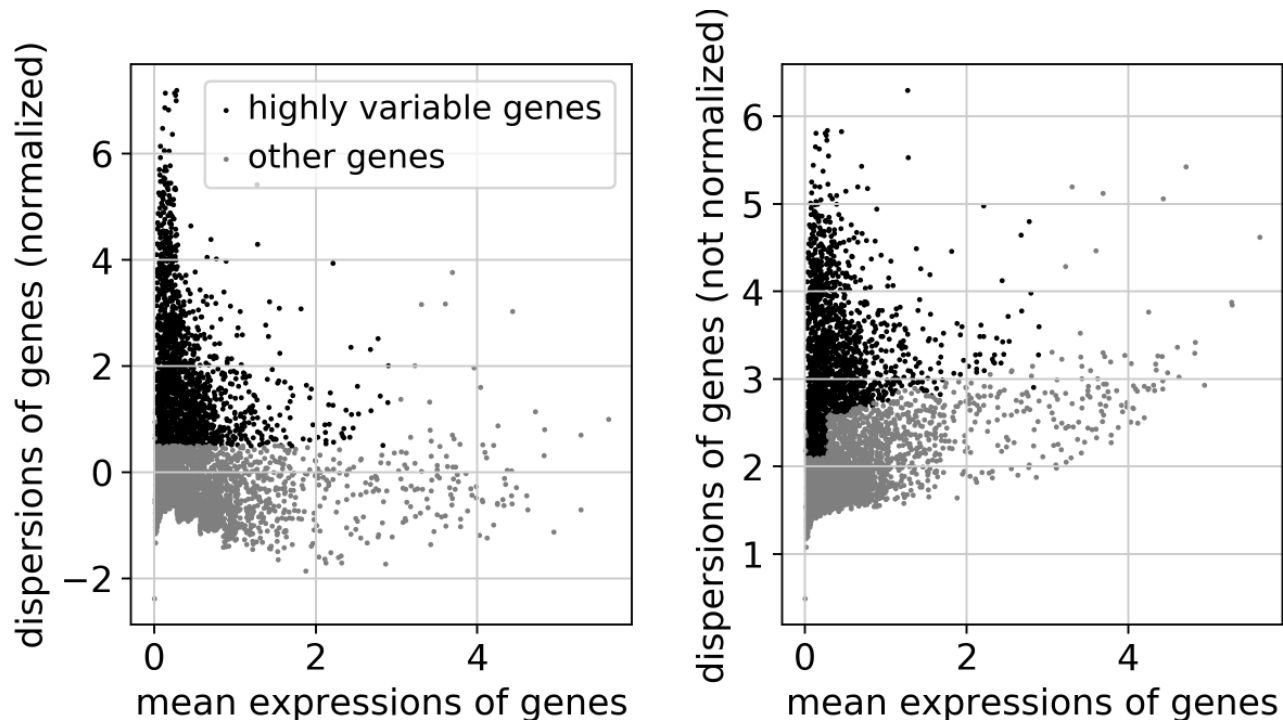
# Data Correction and Integration

- Correct for biological covariates (e.g. cell cycle) covariates –  
*warning: Regress out biological covariates only for trajectory inference and if other biological processes of interest are not masked by the regressed out biological covariate*
- Correct for technical covariates (e.g. batch effects – data integration)



# Feature Selection: Filter for variable genes to control the relationship between variability and average expression

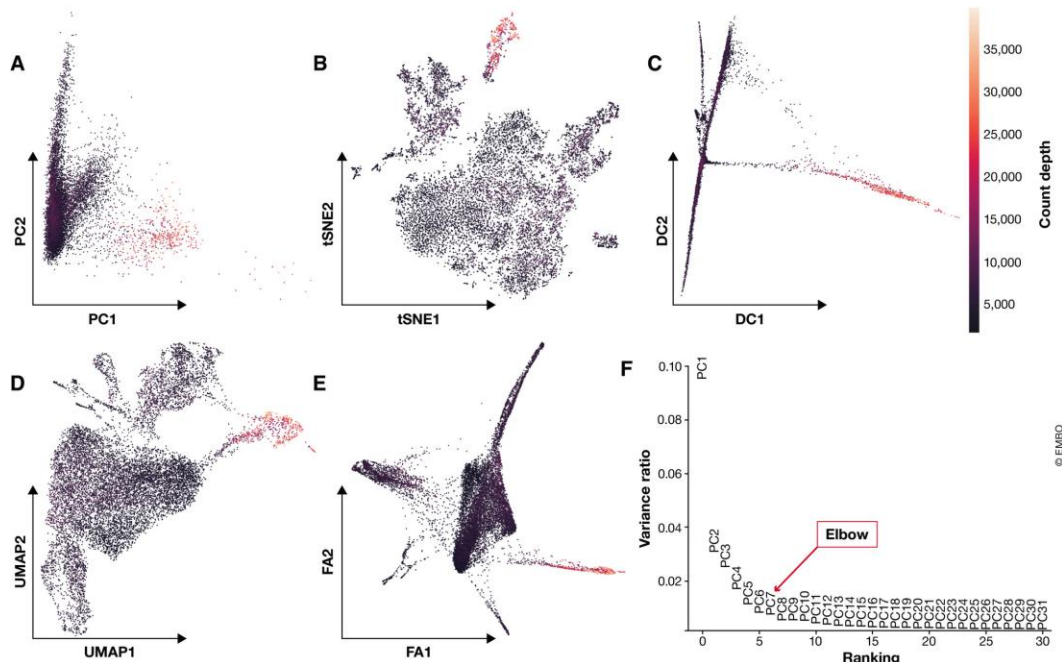
---



Preliminary results from Klein et al (2015) suggest that downstream analysis is robust to the exact choice of the number of HVGs (between 200-2400).



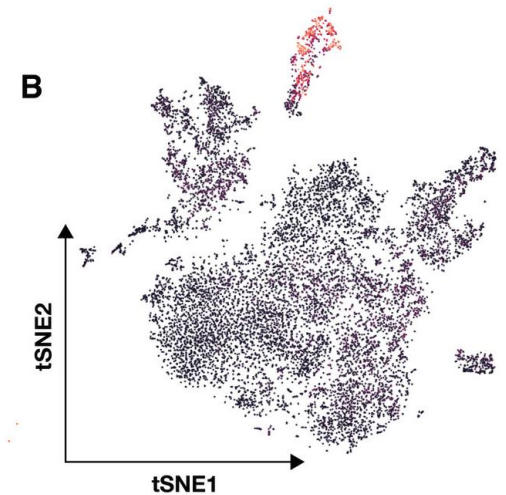
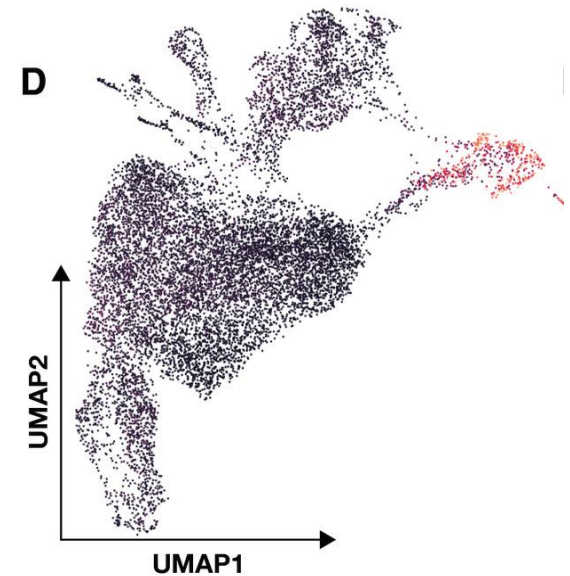
# Dimensionality Reduction



- Embed expression matrix into low-dimensional space that still captures the underlying structure of the data
- Two Goals:
  - 1) Visualization – describe data in 2D or 3D
  - 2) Summarization – reduce data to essential components - used downstream

# Visualization

- Standard Practice – non-linear dimensionality reduction methods
- t-SNE dimensions focus on capturing local similarity at the expense of global structure. Thus, these visualizations may exaggerate differences between cell populations and over-look potential connections between these populations.
- Uniform Approximation and Projection method (UMAP; preprint: McInnes & Healy, 2018) - arguably represent the best approximation of the underlying topology



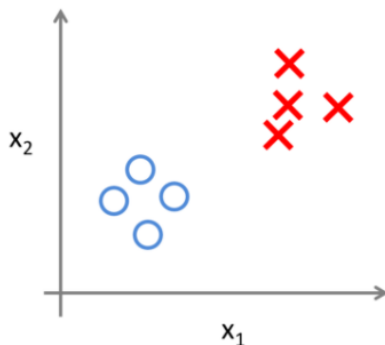
# Stages of pre-processed data

**Table 1. Stages of data processing and appropriate downstream applications.**

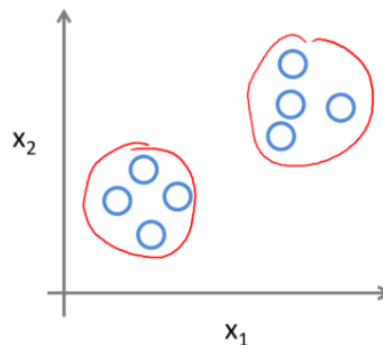
Pre-processing layer	Stage of data processing	Appropriate applications
Measured	1) Raw	Statistical testing (Differential expression: marker genes, genes over condition, genes over time)
	2) Normalized (+ log transformed)	
Corrected	3.1) Corrected (technical correction)	Visual comparison of data (plotting)
	3.2) Corrected (biological correction)	Pre-processing for trajectory inference
Reduced	4) Feature selected	Visualization, trajectory inference
	5) Dimensionality reduced (summarized)	Visualization, clustering, KNN graph inference, trajectory inference

# Cluster Analysis

Supervised Learning

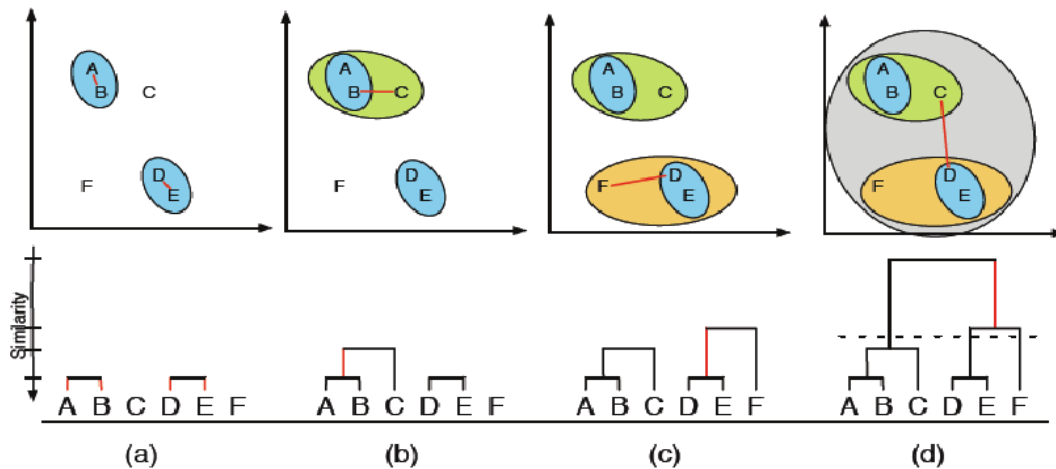


Unsupervised Learning

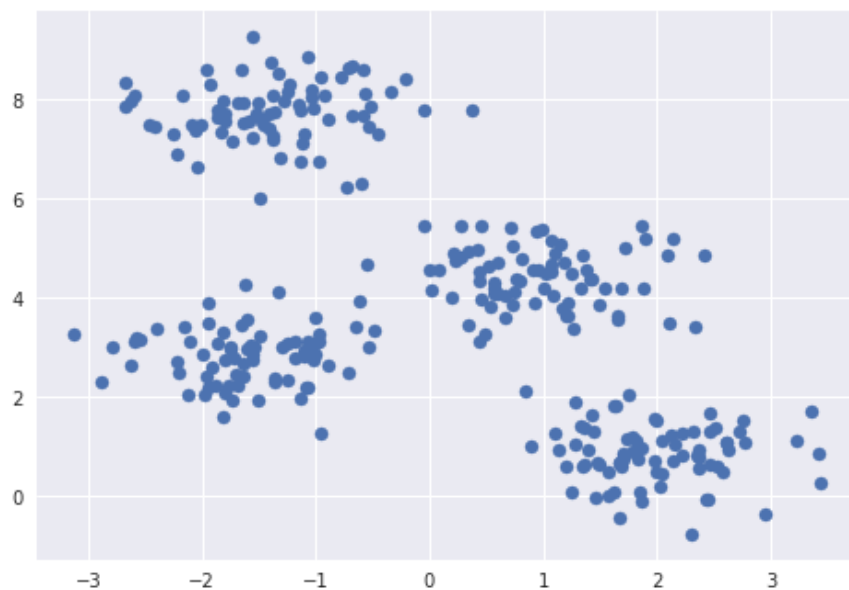
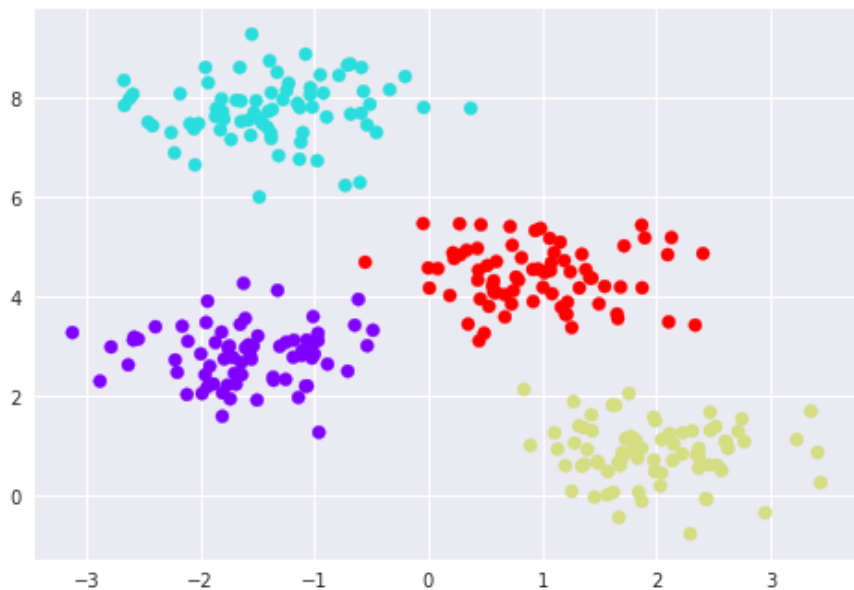


- **Clustering** – group cells based on similarity of expression profiles determined by distance metrics
  - Supervised:
    - Have prior knowledge of the groups.
  - Unsupervised:
    - Have no priors. Looking for substructure to find new patterns
- Cells are assigned to clusters by minimizing intracluster distances or finding dense regions in the reduced expression space.
- Louvain algorithm detects communities as groups of cells that have more links between them than expected from the number of links the cells have in total.

# Hierarchical Clustering:



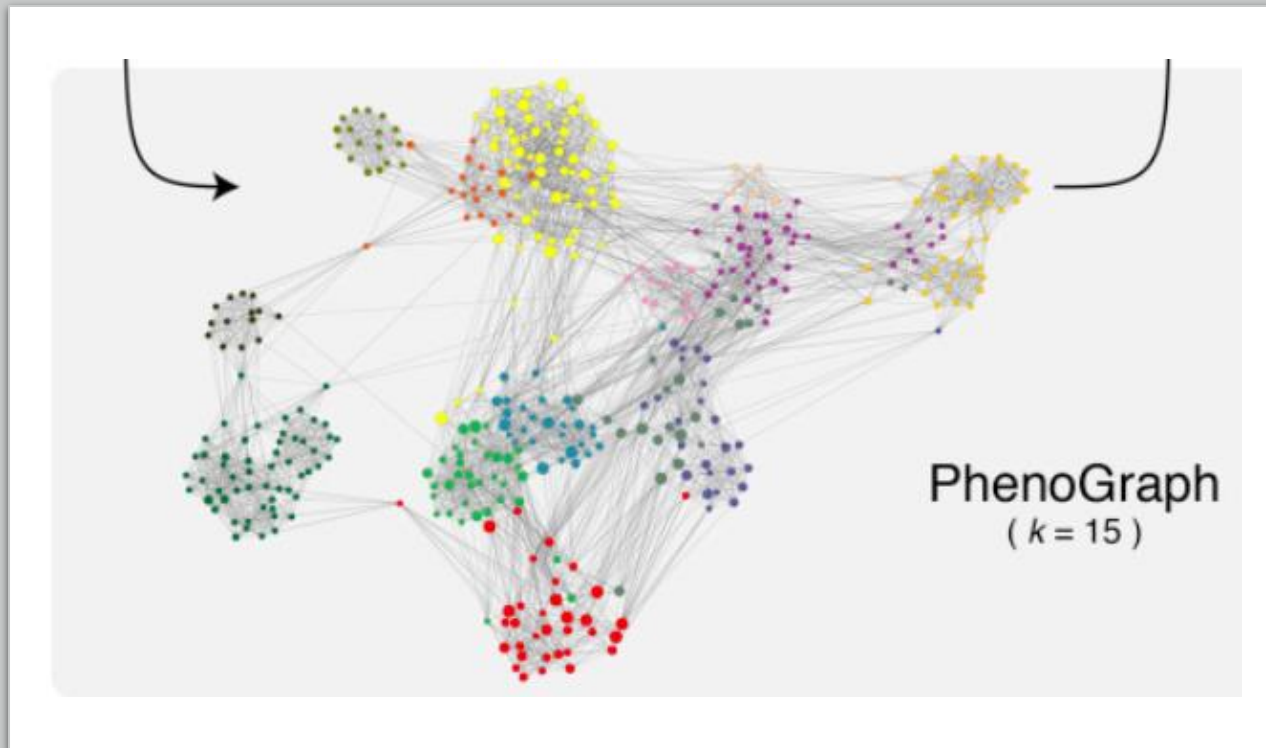
- Grouping cells together based on similarity
- Bottom up: Start with individual points and add most similar data points
- Top down: Start with one big group, and drop out one at a time based on similarity
- Cluster groups are defined based on your chosen similarity metric



How many clusters exist in this dataset?

## K-means clustering

- Randomly assign points to guess the cluster centers (number of points =  $k$ )
- Assign data points to the nearest cluster center.
- Move the cluster assignment to the mean of each group
- Repeat



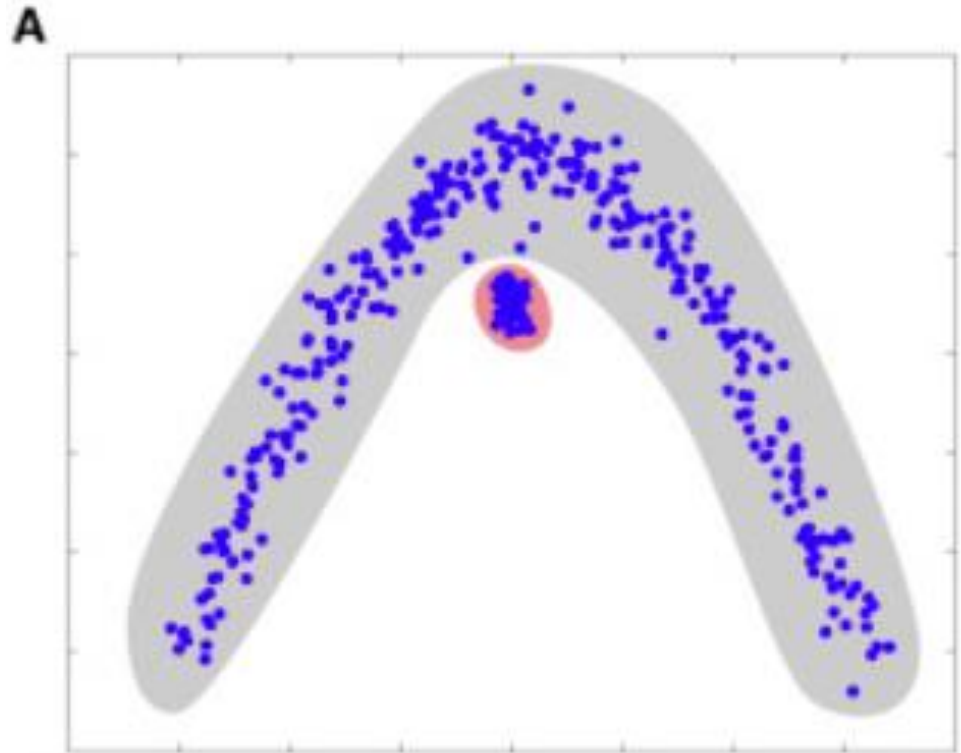
## Graph-based clustering (k-nearest neighbors)

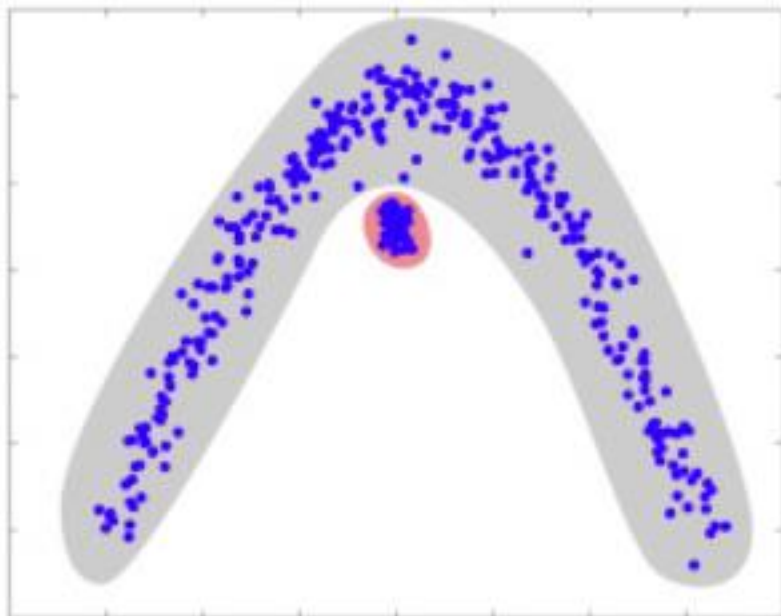
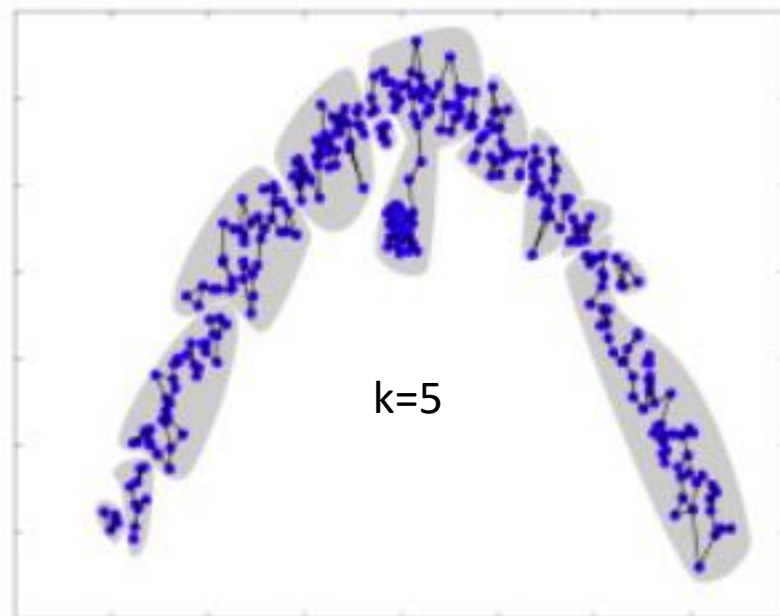
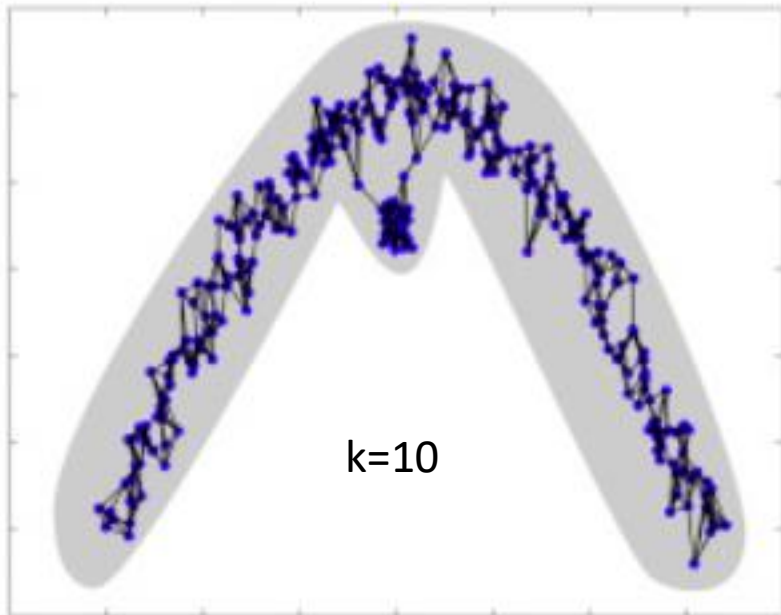
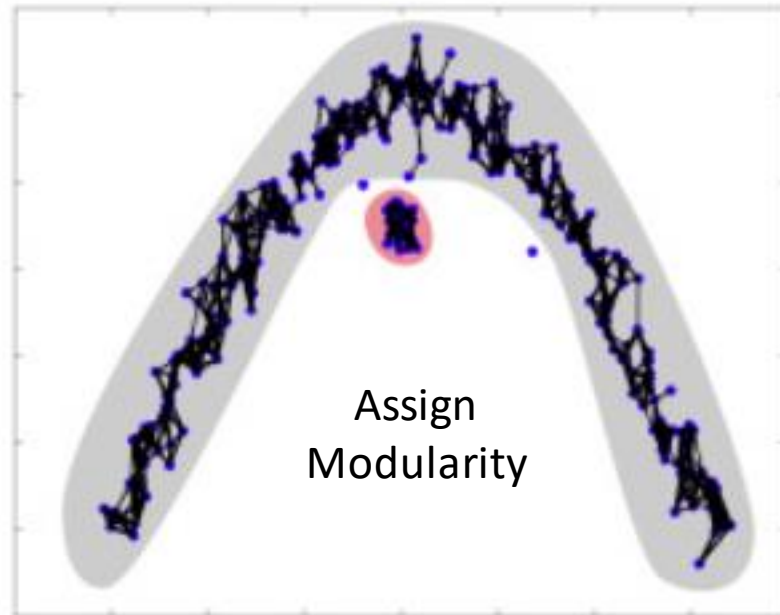
Cells are represented as nodes in the graph. Each cell is connected to its  $K$  most similar cells, which are typically obtained using Euclidean distances on the PC-reduced expression space. Depending on the size of the dataset,  $K$  is commonly set to be between 5 and 100 nearest neighbors. The resulting graph captures the underlying topology of the expression data



# Graph-based clustering (k-nearest neighbors)

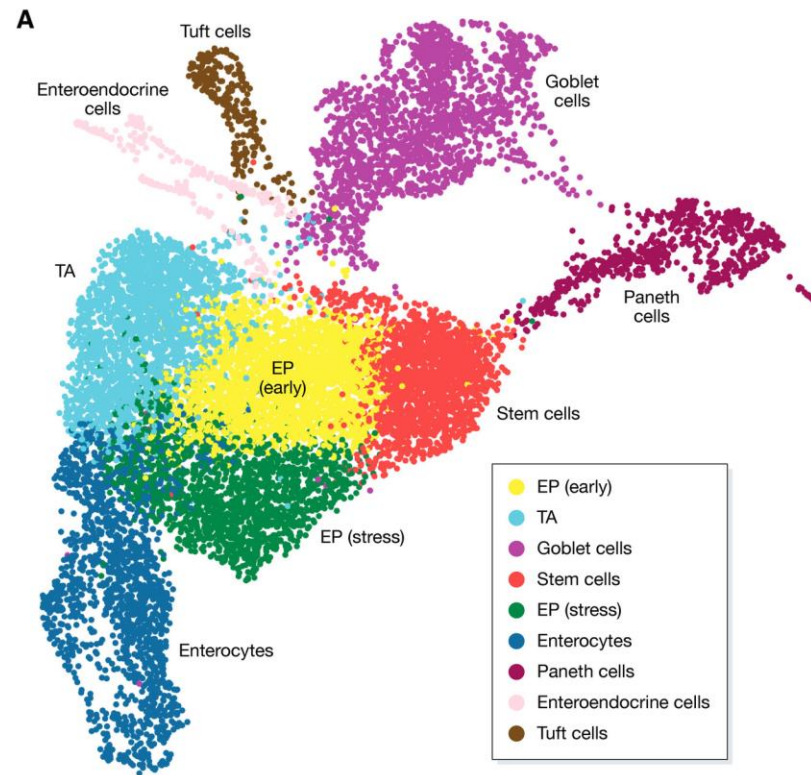
- Measure Euclidian distances between cells up to K nearest neighbors (edges)
  - Weight of edges scales with similarity metric (Euclidian distance)
- Calculate “connectedness” of nodes (cells) by the number of shared neighbors
- Refine densely connected modules as communities



**A****B****C****D**

# Cluster Annotation

Compare marker genes in data to marker genes in reference dataset



# Trajectory analysis

Captures transitions between cell identities, differentiation processes, or changes in biological function

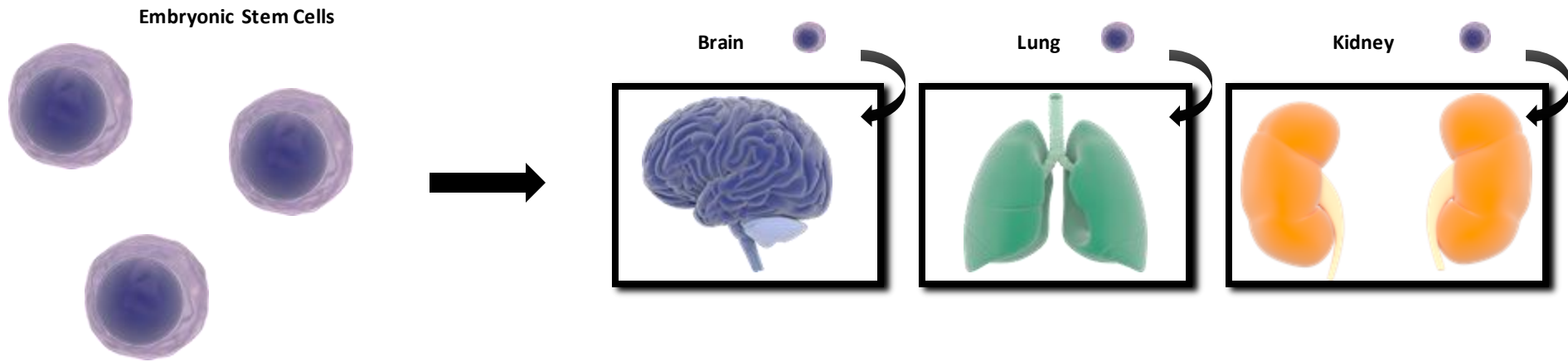
Recent comparisons revealed that Slingshot (Street et al, 2018) outperformed other methods for simple trajectories that range from linear to bi- and multifurcating models. If more complex trajectories are expected, PAGA (Wolf et al, 2019) was recommended by the authors.

## LETTER

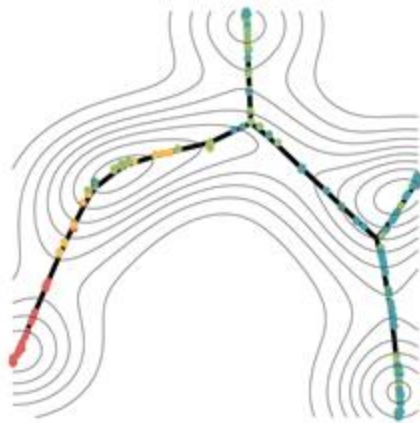
<https://doi.org/10.1038/s41586-018-0414-6>

## RNA velocity of single cells

Gioele La Manno<sup>1,2</sup>, Ruslan Soldatov<sup>3</sup>, Amit Zeisel<sup>1,2</sup>, Emelie Braun<sup>1,2</sup>, Hannah Hochgerner<sup>1,2</sup>, Viktor Petukhov<sup>3,4</sup>, Katja Lidschreiber<sup>5</sup>, Maria E. Kastriti<sup>6</sup>, Peter Lönnerberg<sup>1,2</sup>, Alessandro Furlan<sup>1</sup>, Jean Fan<sup>3</sup>, Lars E. Borm<sup>1,2</sup>, Zehua Liu<sup>3</sup>, David van Bruggen<sup>1</sup>, Jimin Guo<sup>3</sup>, Xiaoling He<sup>7</sup>, Roger Barker<sup>7</sup>, Erik Sundström<sup>8</sup>, Gonçalo Castelo-Branco<sup>1</sup>, Patrick Cramer<sup>5,9</sup>, Igor Adameyko<sup>6</sup>, Sten Linnarsson<sup>1,2\*</sup> & Peter V. Kharchenko<sup>3,10\*</sup>



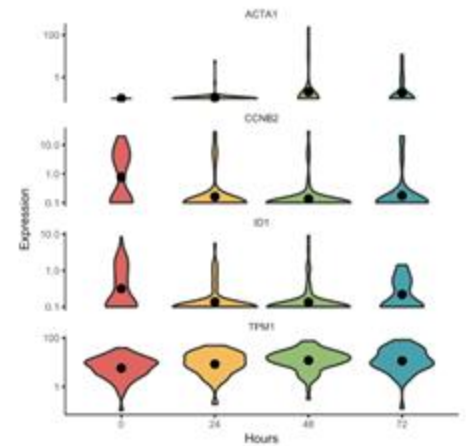
**How can one reconstruct cellular dynamics using static snapshots of the cellular state from single cell RNA-seq (scRNA-seq)?**



Pseudotime



Clustering

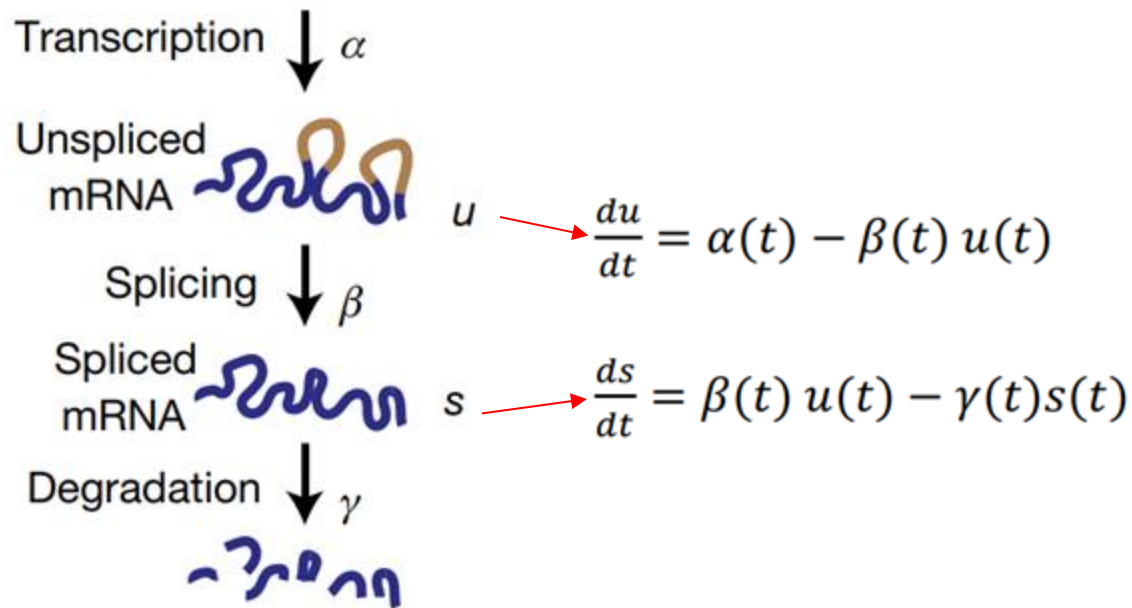


Differential expression

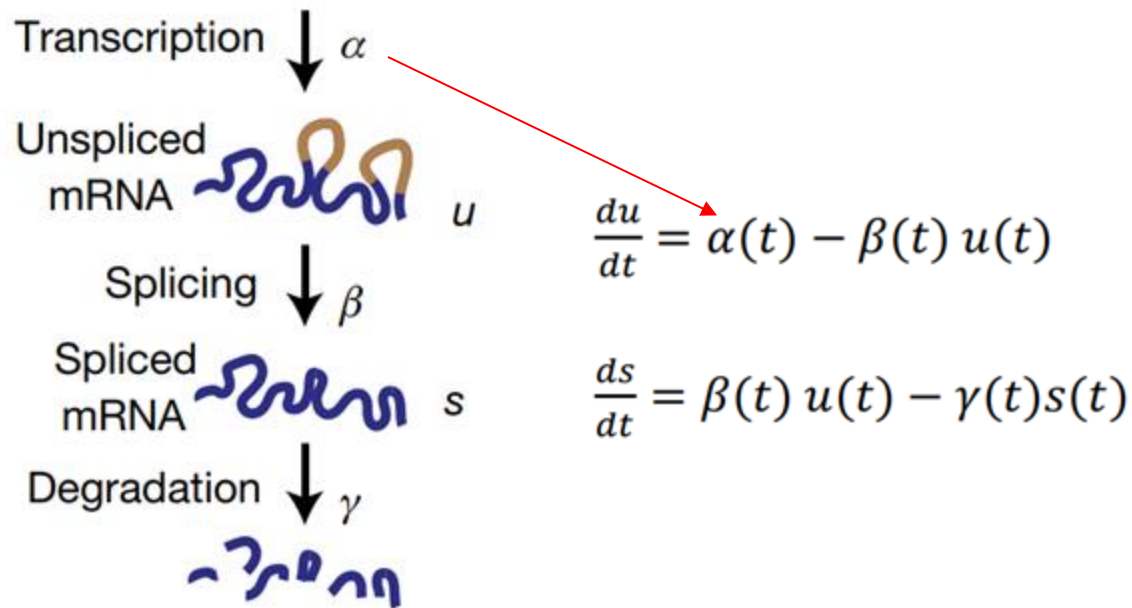
**The relative abundance of **unspliced (nascent)** and **spliced (mature)** mRNA can be used to reveal the **rate** and **direction** of transcriptomic changes**



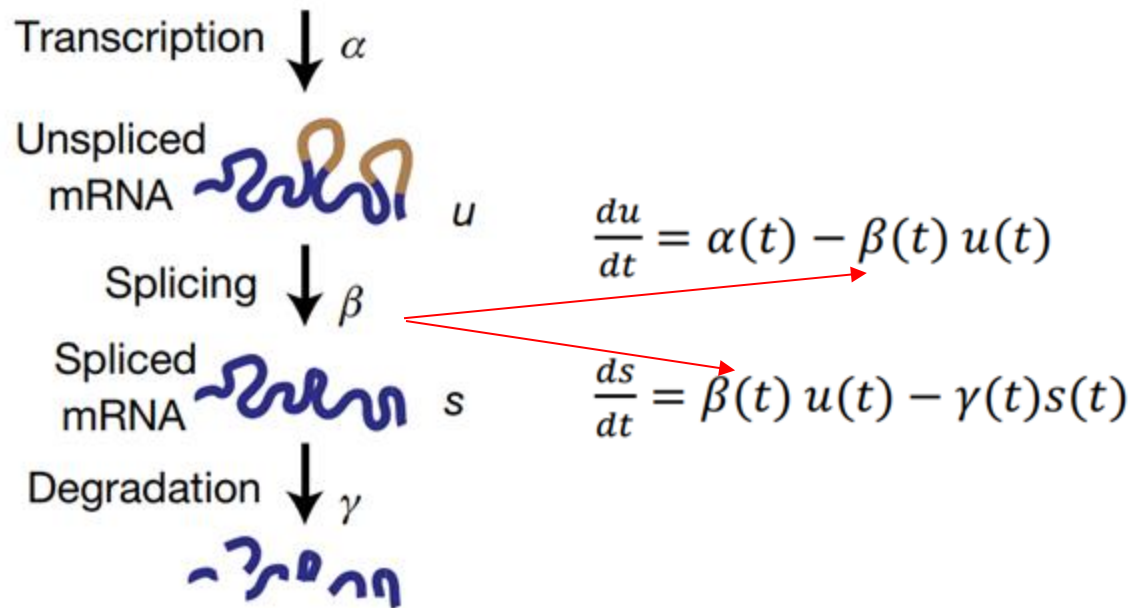
# Unspliced/Spliced model



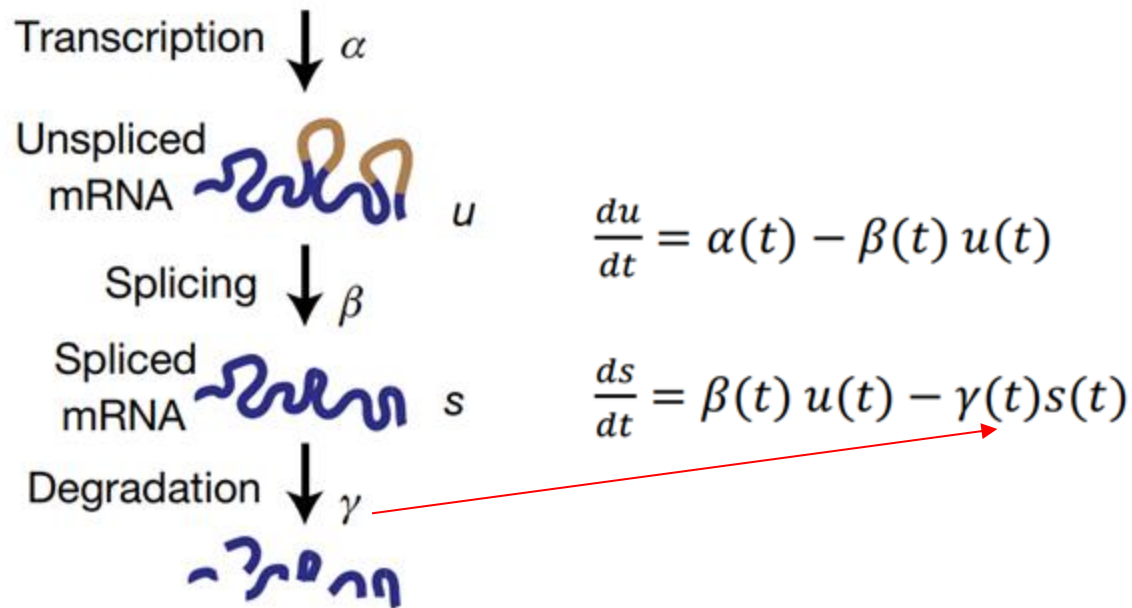
# Unspliced/Spliced model



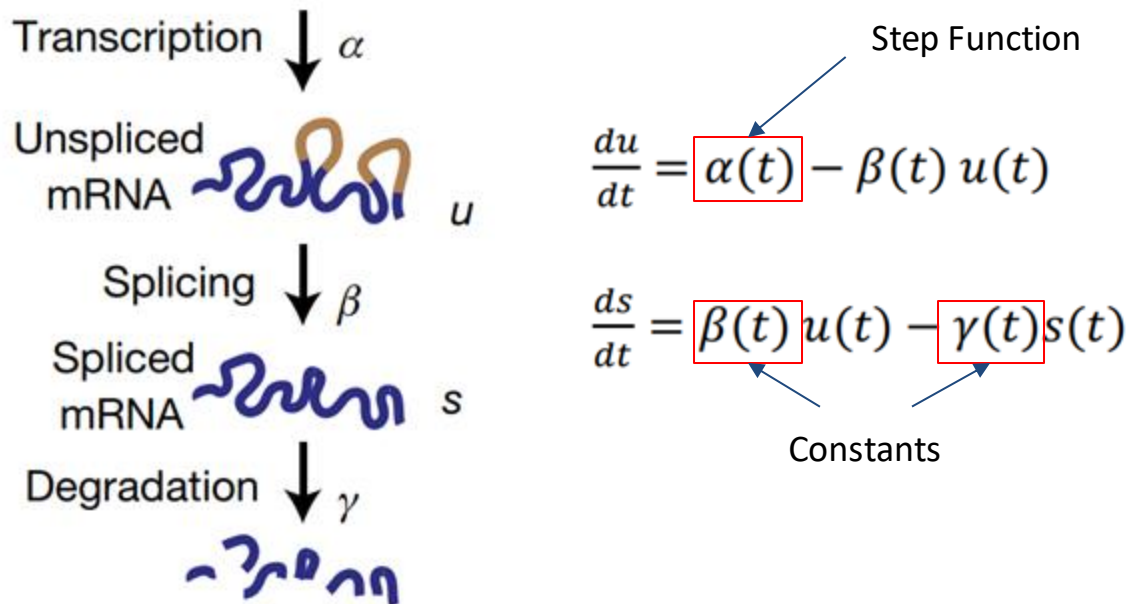
# Unspliced/Spliced model



# Unspliced/Spliced model

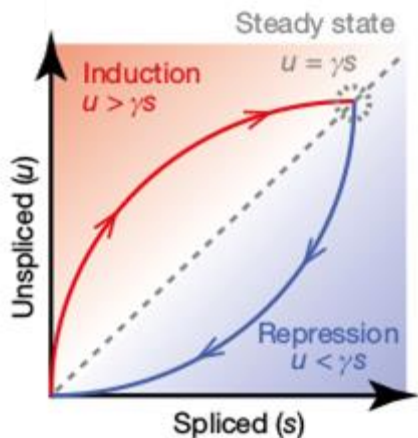


# Assumption: Time independent transcription, splicing, and degradation rates



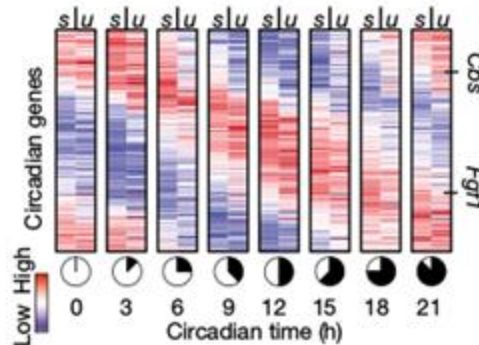
# Circadian Cycle in Mouse Liver Demonstrates Model

Proposed Model



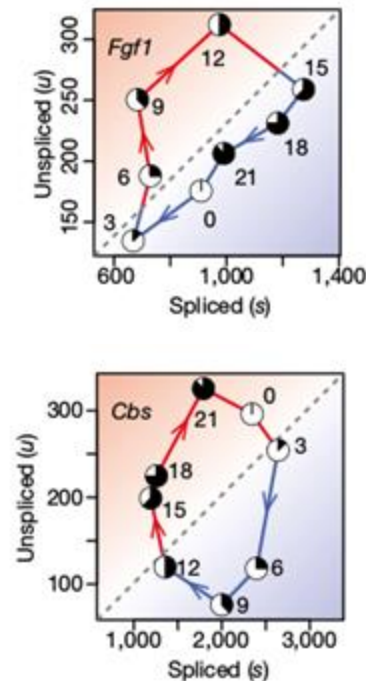
$$\frac{ds}{dt} = u - \gamma s$$

Spliced and Unspliced mRNA Levels

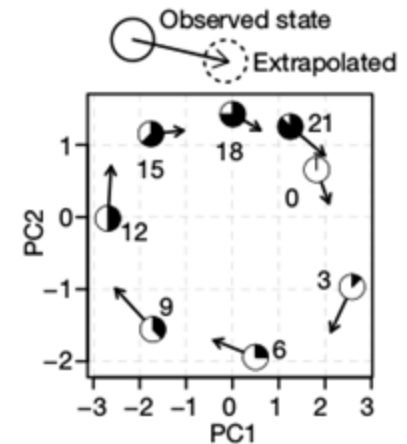


Circadian-associated genes (Fgf1 and Cbs) show excess of unspliced mRNA in upregulation and deficit in downregulation

Applied Model

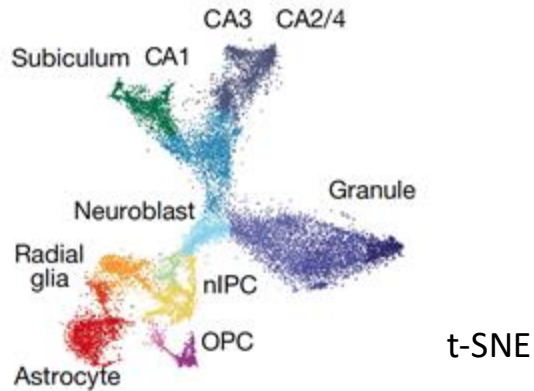


Predictive Model

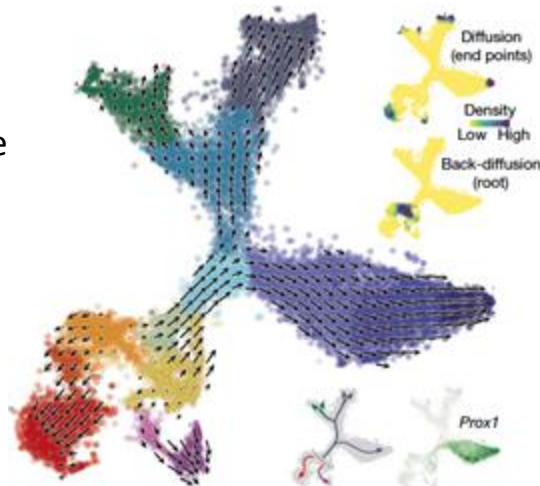


Solving differential equation of each gene allows prediction of the expected direction of progression of the circadian cycle

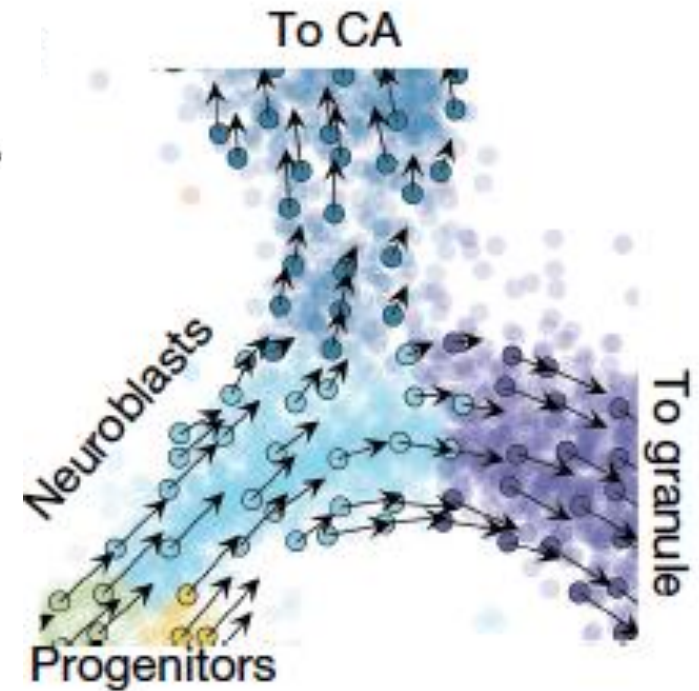
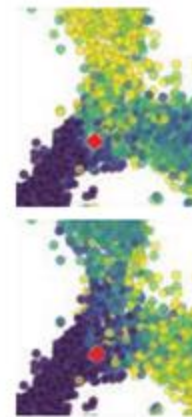
# Fate Decisions of Major Neural Lineages



RNA Velocity can orient a lineage tree without prior knowledge of the developmental process



transition prob.  
 $10^{-6}$   $6 \times 10^{-4}$





# Conclusions

- Reveals temporal dynamics of single-cell gene expression on a timescale of hours
- Matches the unfolding of developmental, regenerative and reactive processes in both human and other mammals
- Can be used to model commitment, fate choice and precise kinetics of transcription in vivo
- Can be used to estimate cellular state 2.5-3.8 hours into the future

# Issues

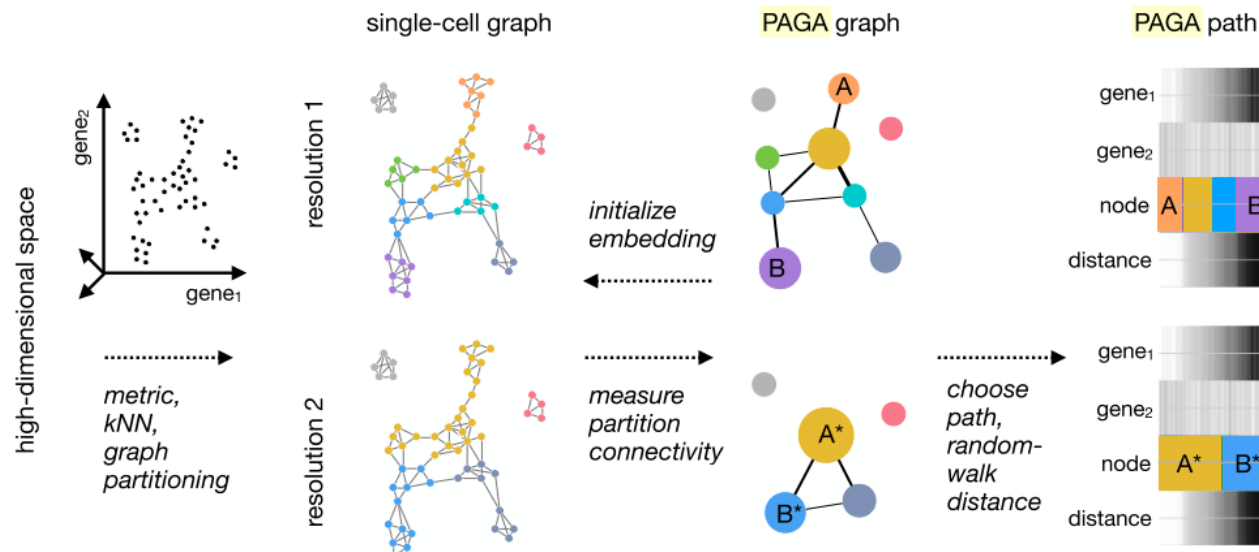
- Doesn't work on some models of differentiation (such as in vitro differentiation, which is over several weeks)
- The exact time-points of when you take your samples becomes incredibly important
- Genetic variation within sampling

# PAGA (Partition-based graph abstraction)

- Current computational approaches for scRNA-seq:
  - **Clustering**: assumes that data is composed of biologically distinct groups such as discrete cell types or states and labels these with a discrete variable
  - **Inferring pseudotemporal orderings or trajectories of cells**: assumes that data lie on a connected manifold and labels cells with a continuous variable
- PAGA is a statistical model for the connectivity of groups of cells whose nodes correspond to cell groups and whose edge weights quantify the connectivity between groups. Groups are connected if their number of inter-edges exceeds a fraction of the number of inter-edges expected under random assignment.
- By quantifying the connectivity of partitions (groups, clusters) of the single-cell graph, PAGA generates a much simpler abstracted graph (PAGA graph) of partitions
- By averaging over single-cell paths, it becomes possible to trace a putative biological process from a progenitor to fates

# Partition-based graph abstraction (PAGA)

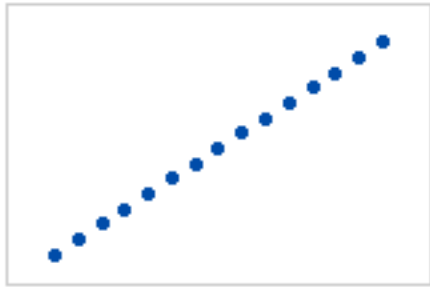
- PAGA generates graph-like maps of cells that preserve both the continuous and disconnected structure in single cell mRNA-seq data, reconciling clustering and pseudotemporal ordering algorithms



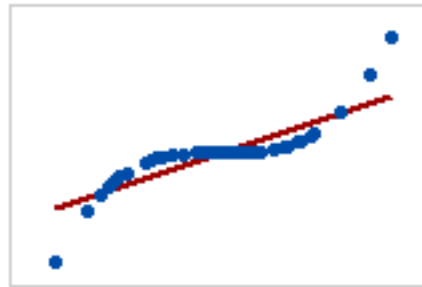
# Pearson vs Spearman correlation

Pearson Correlation: For every increase in  $x$ , is there a constant, proportional increase in  $y$ ?

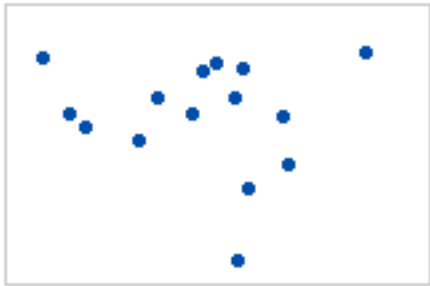
Spearman Correlation: Rank based. If  $x$  increases, does  $y$  also increase  
(magnitude doesn't matter)



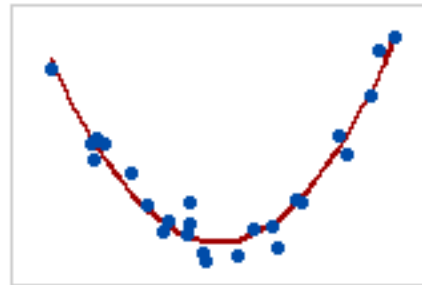
Pearson = +1, Spearman = +1



Pearson = +0.851, Spearman = +1



Pearson = -0.093, Spearman = -0.093

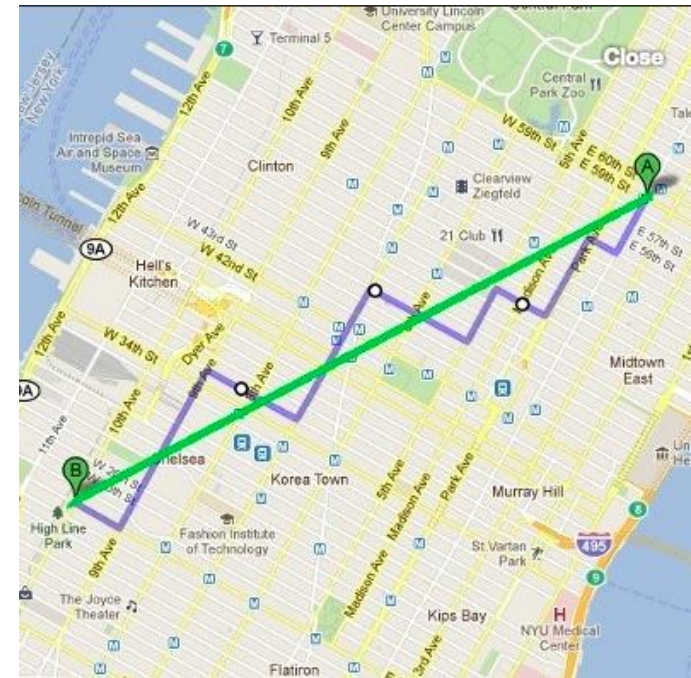
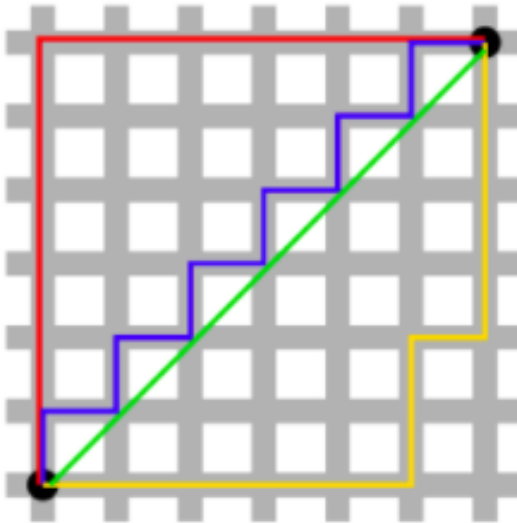


Coefficient of 0

# Distance metrics for “similarity” (Euclidean and Manhattan)

Euclidean: shortest distance between two points

Manhattan: distance when only moving along x or y axis



- Distance is scaled to N dimensional vectors for lots of cells and genes
- Two cells with 500 genes in each contain 500 distances to measure for overall similarity

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

# Dimensionality Reduction (PCA)

## PCA:

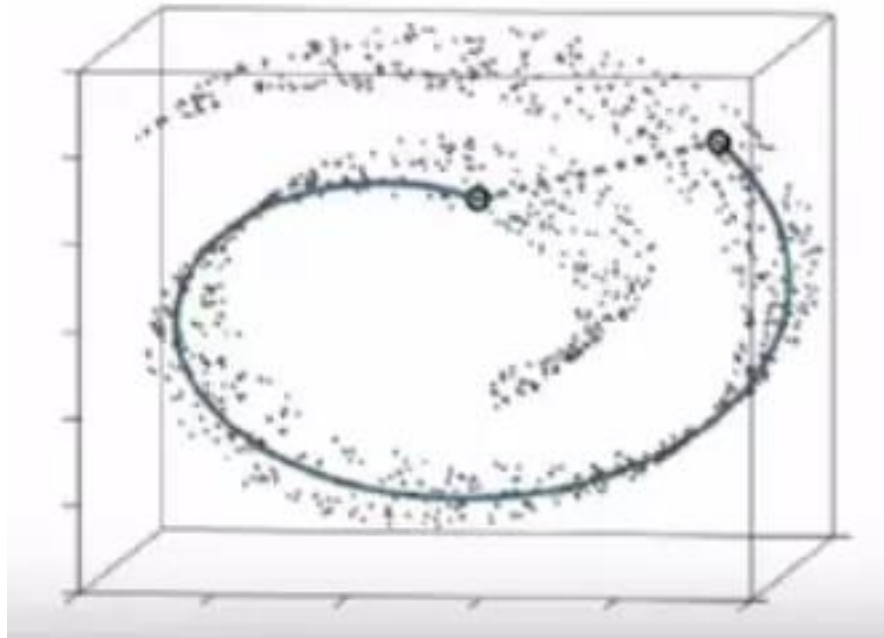
Identifies linear components of variation within the dataset



# Dimensionality Reduction (tSNE)

## tSNE:

- Handles non-linear and sparse data very well
- Preserves local relationships in high dimensional space



PCA and tSNE application to digits:

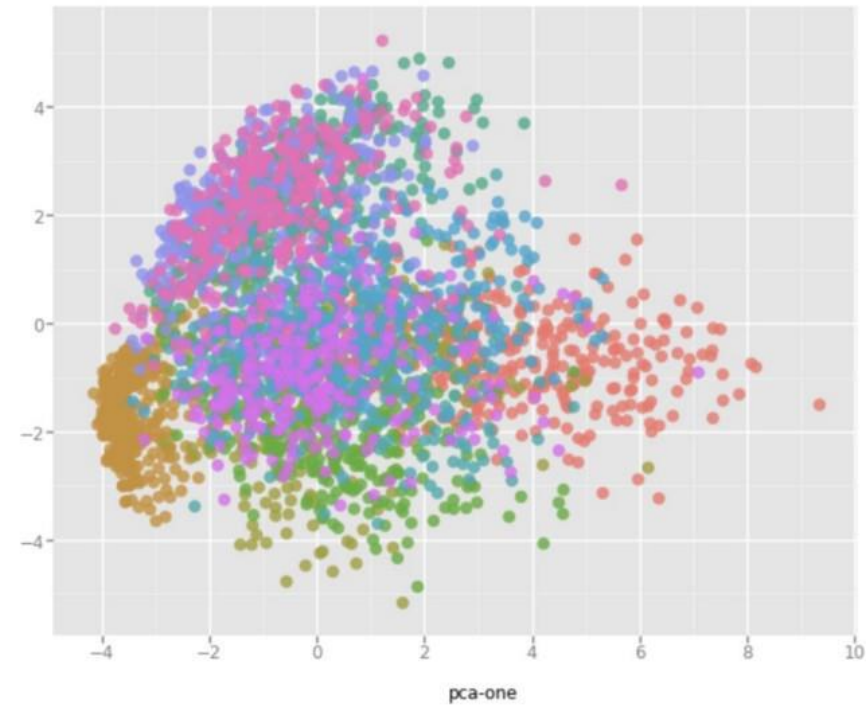




# PCA and tSNE application to digits

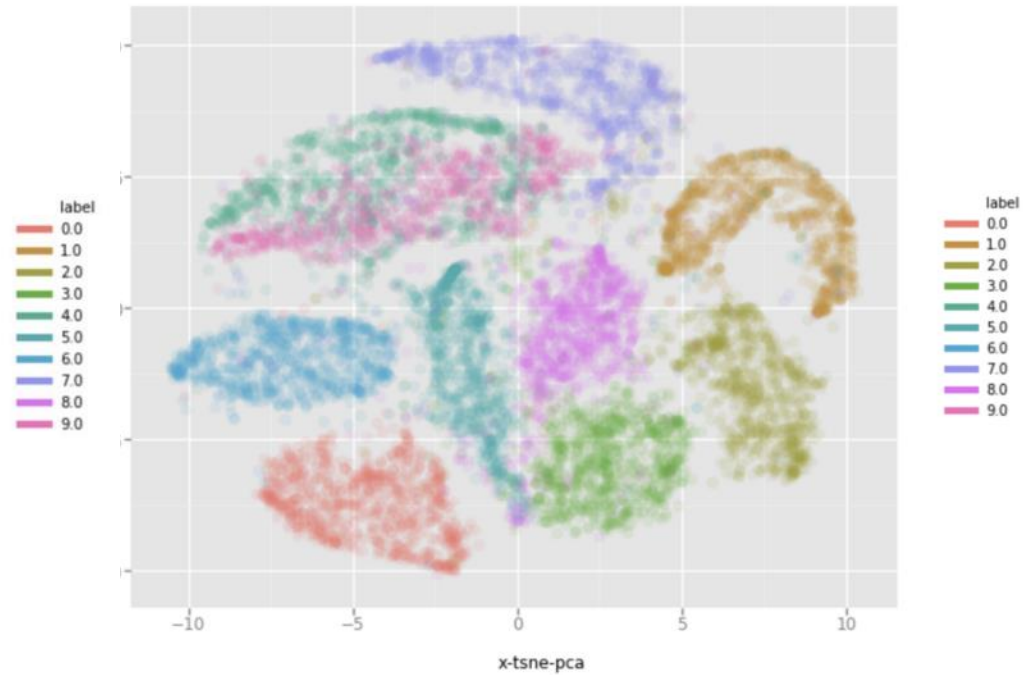
## PCA

First and Second Principal Components colored by digit



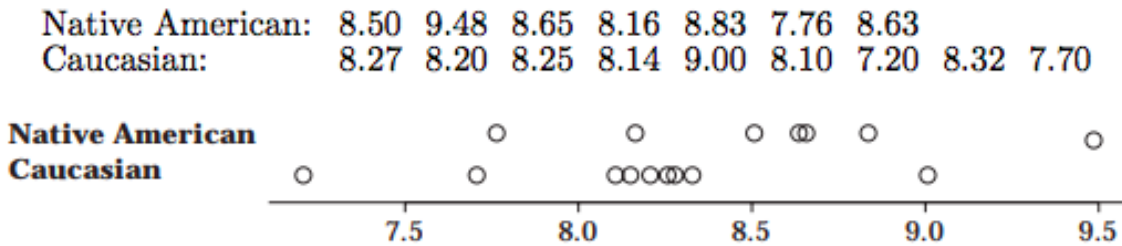
## tSNE

tSNE dimensions colored by Digit (PCA)



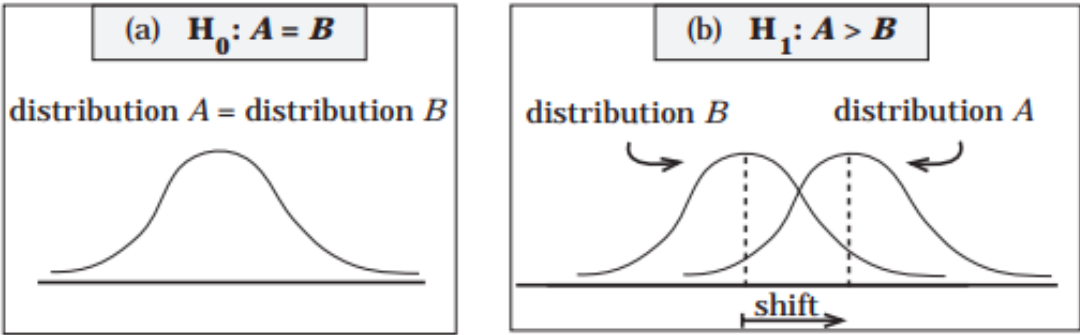
# Wilcoxin rank-sum test (Mann Whitney U test)

- Rank-based t-test (data is not normally distributed)



**Figure 1 :** Comparing MSCE measurements.

	7.20	7.70	7.76	8.10	8.14	8.16	8.20	8.25	8.27	8.32	8.50	8.63	8.65	8.83	9.00	9.48
Race	Ca	Ca	NA	Ca	Ca	NA	Ca	Ca	Ca	Ca	NA	NA	NA	NA	Ca	NA
<b>Rank:</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16



**Figure 2 :** Illustration of  $H_0 : A = B$  versus  $H_1 : A > B$ .

# Bonferroni correction for multiple hypothesis testing

- If we perform 1000 tests (1000 genes) and accept a p-value  $< 0.05$ , how many tests will be incorrect?
- Bonferroni correction (very stringent)
  - Multiply the resulting p-values by the number of tests performed