

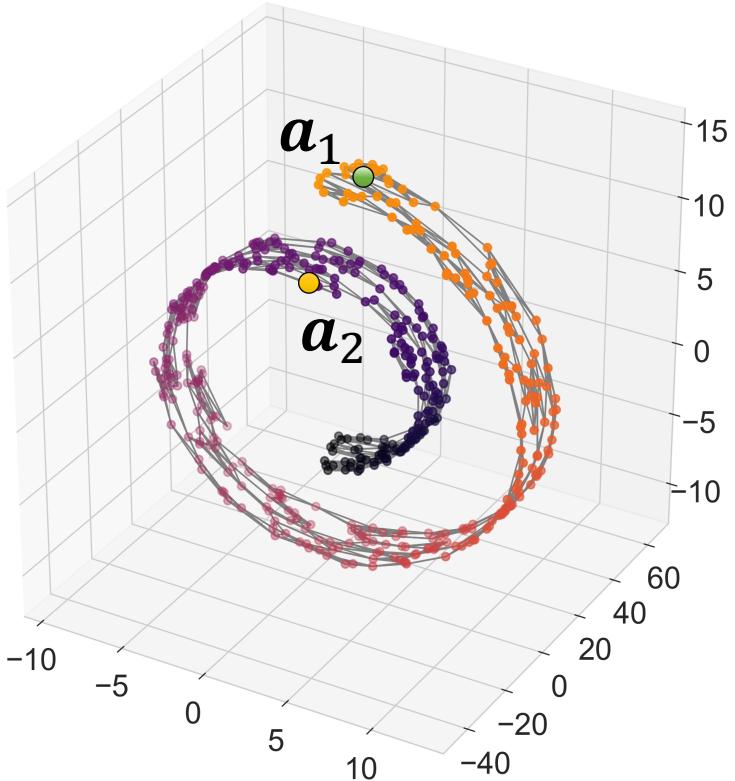
When poll is active, respond at **PollEv.com/yaleml**

Text **YALEML** to **22333** once to join

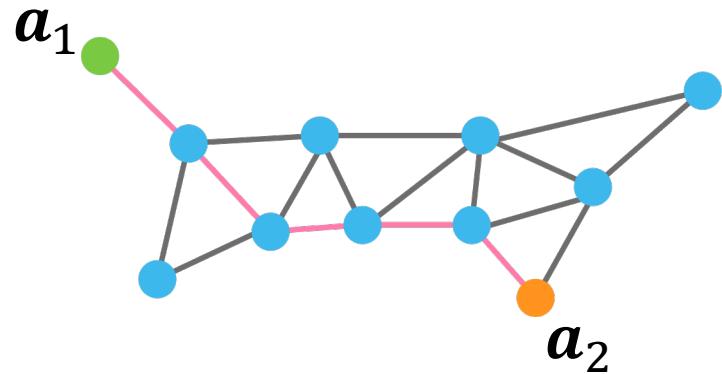
What is a passion you've discovered during quarantine?

Recap from Day 2

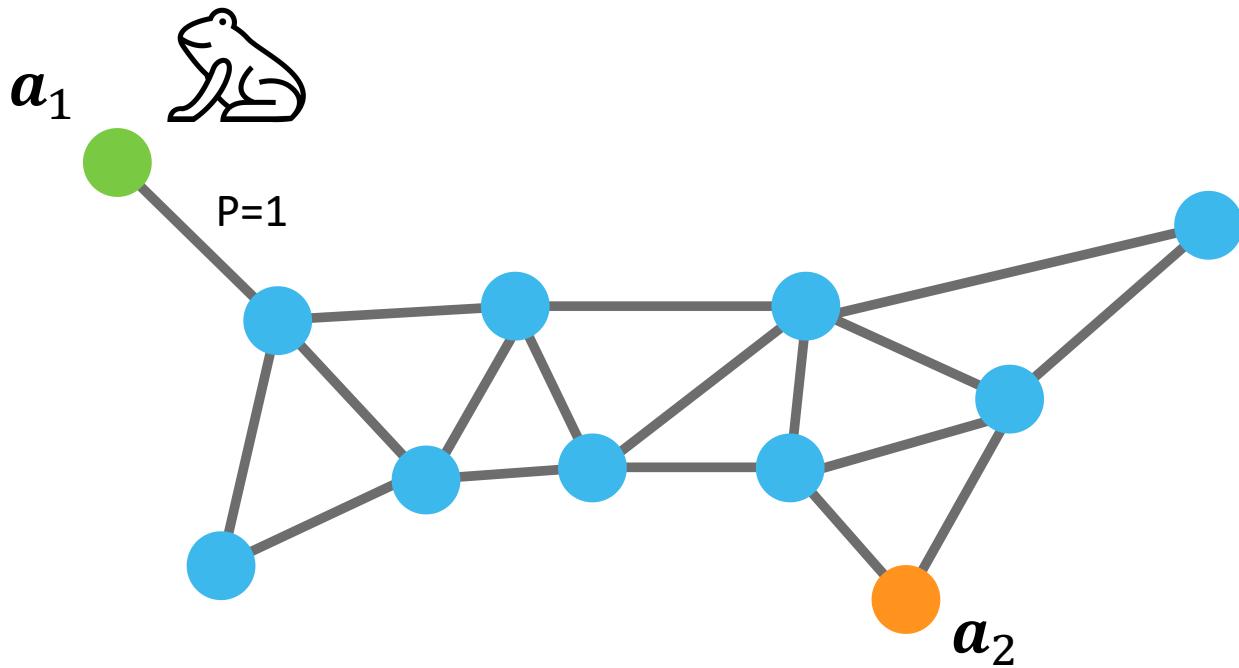
Paths or “walks” on graphs reveal dominant data manifold directions



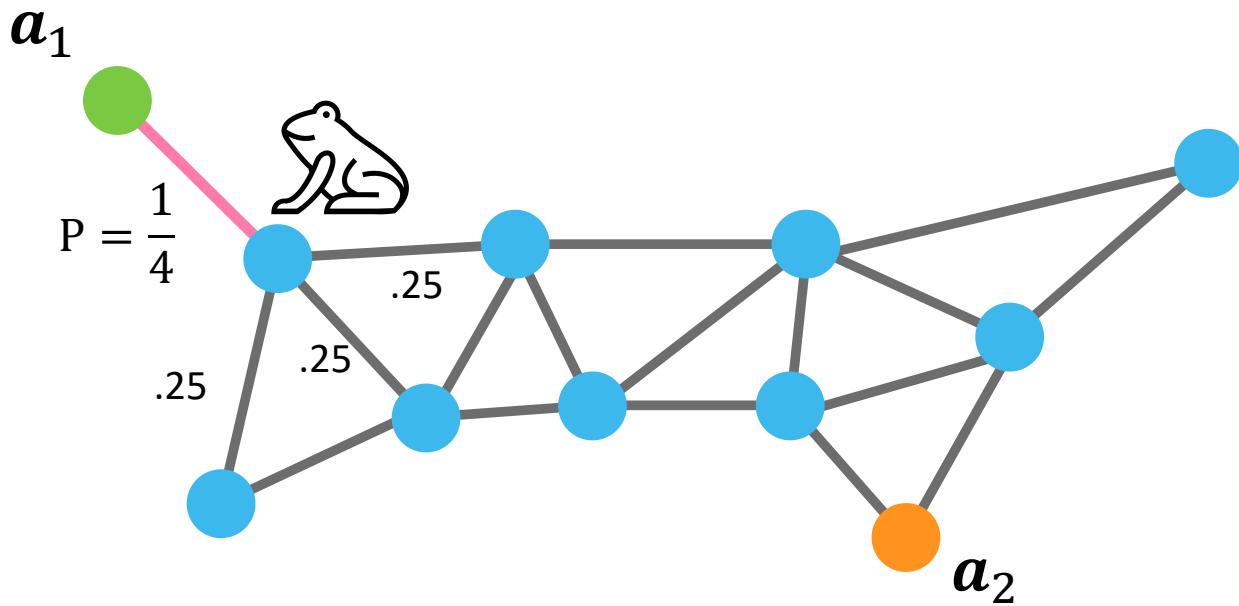
Shortest path between points



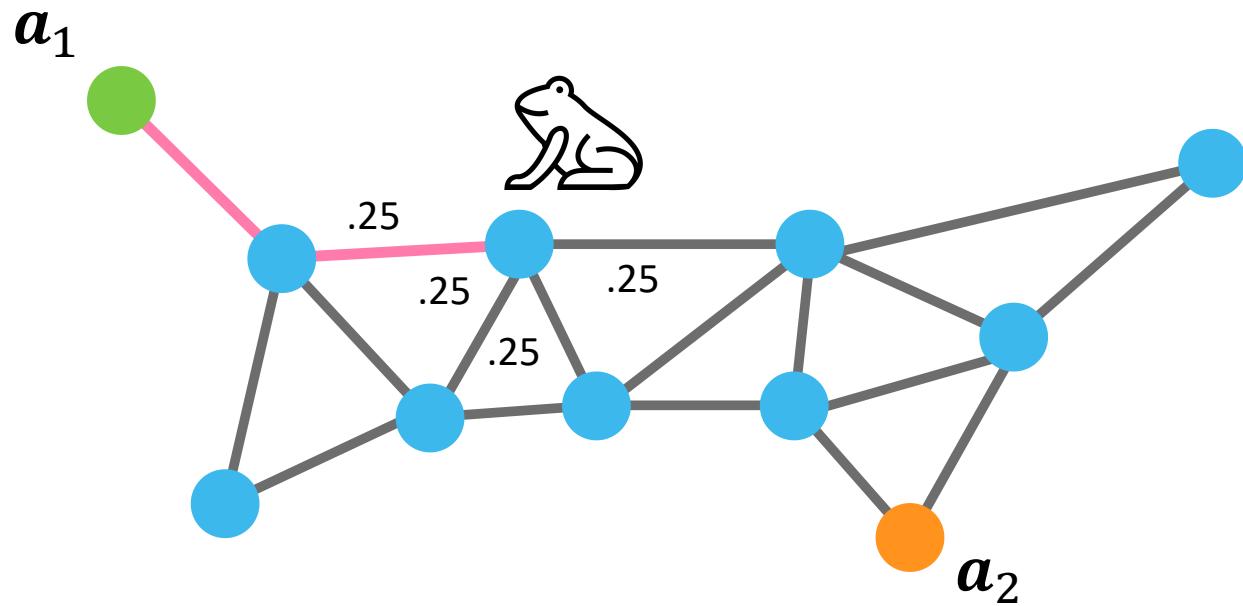
Paths or “walks” on graphs reveal dominant data manifold directions



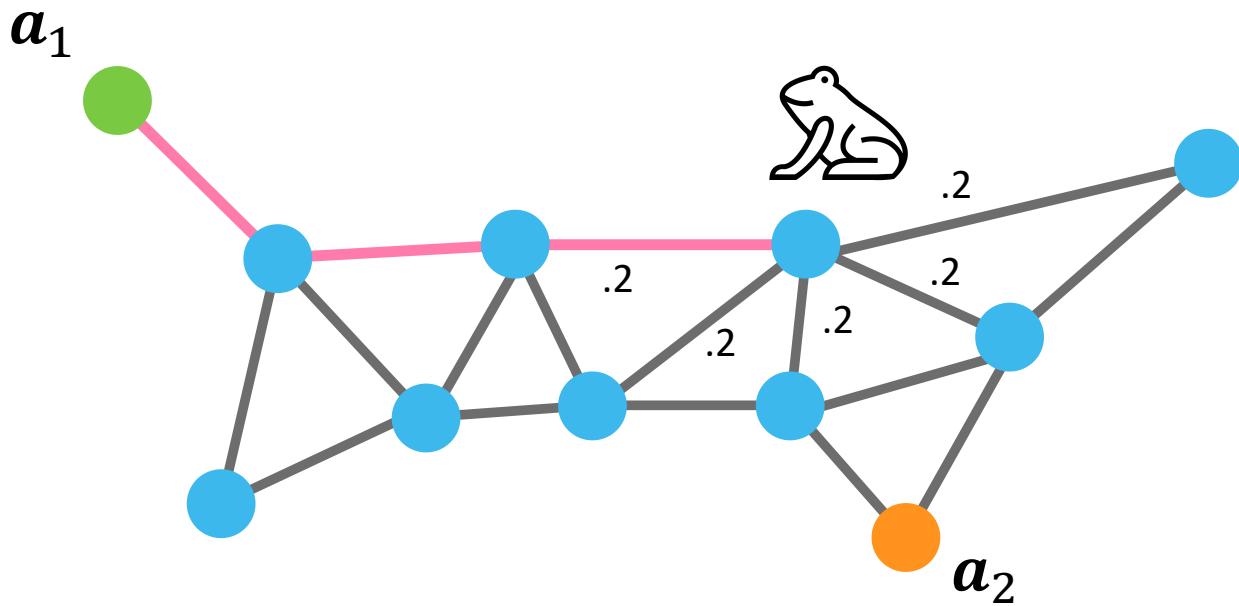
Paths or “walks” on graphs reveal dominant data manifold directions



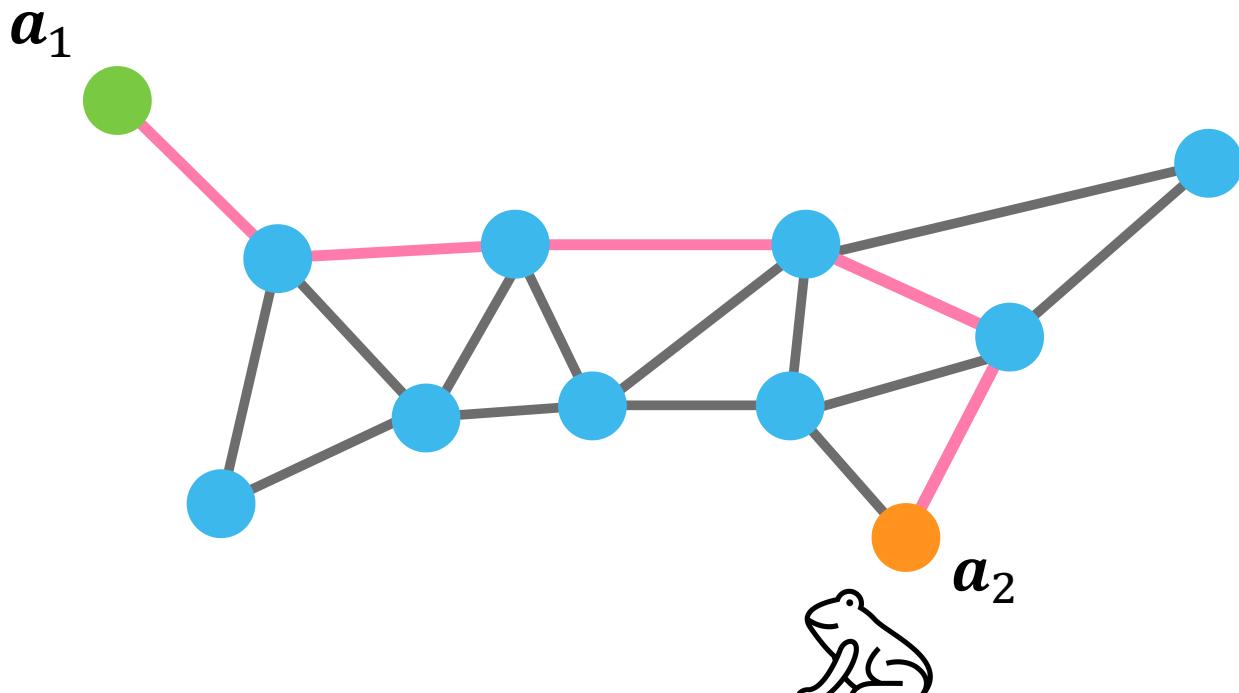
Paths or “walks” on graphs reveal dominant data manifold directions



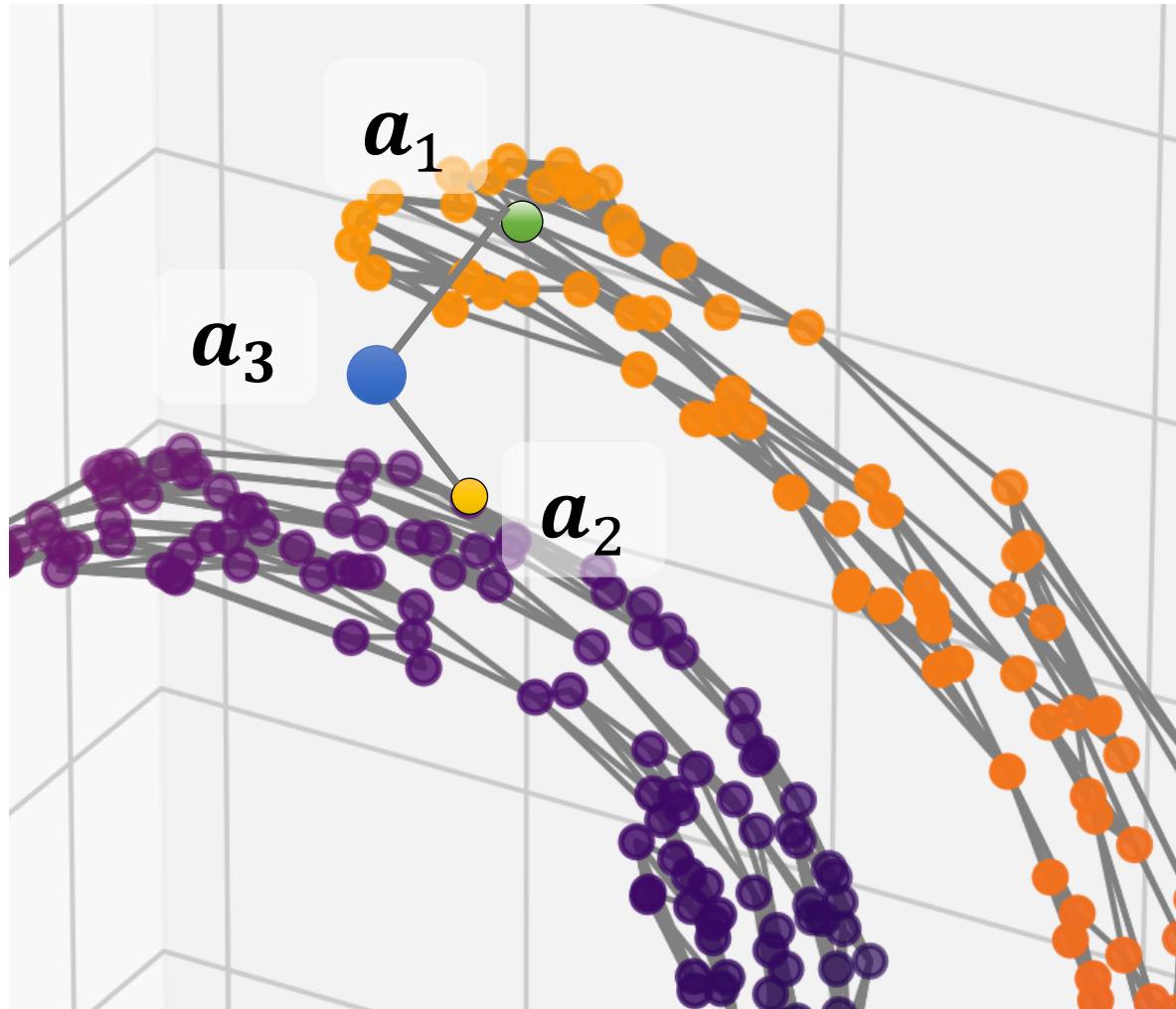
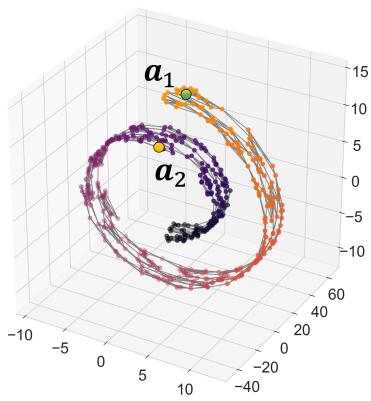
Paths or “walks” on graphs reveal dominant data manifold directions

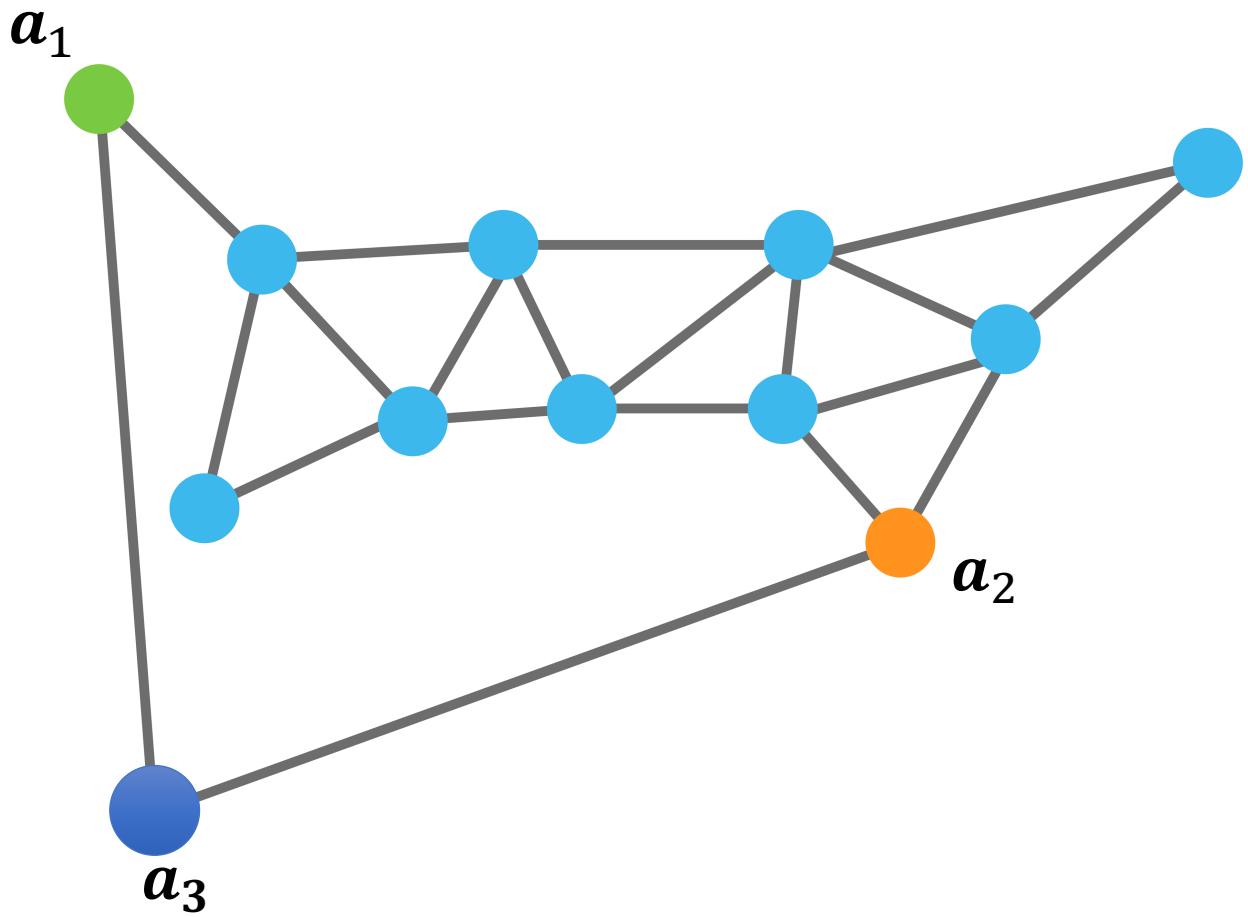
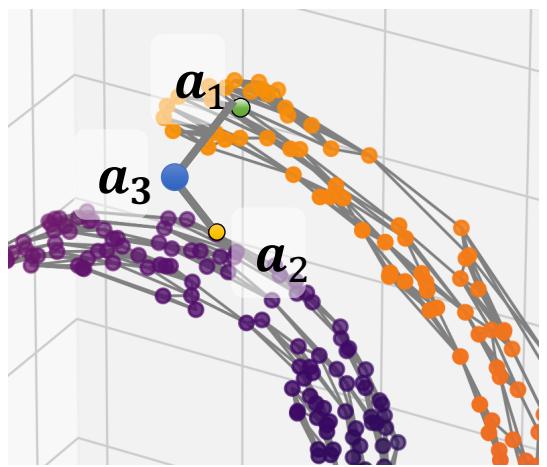


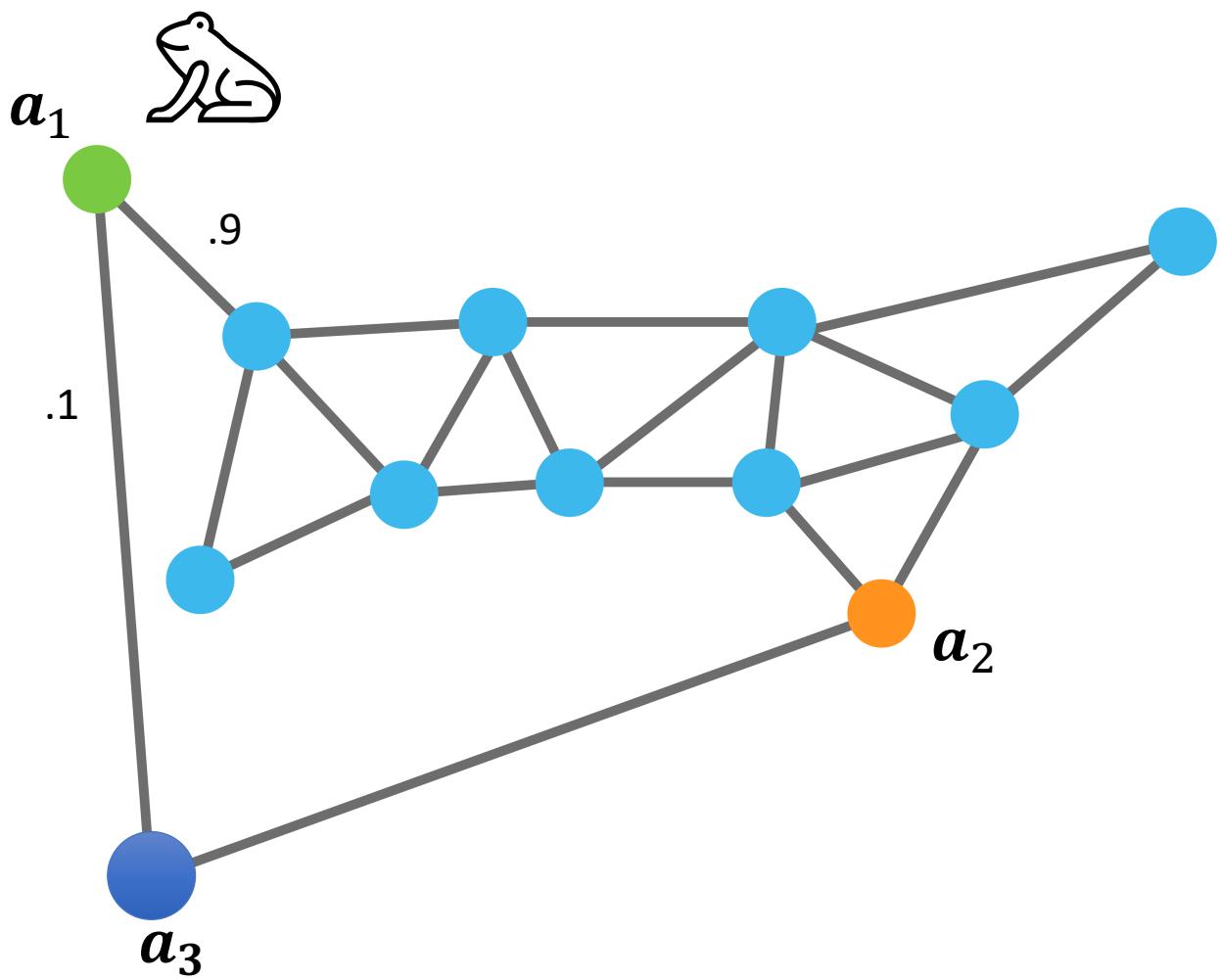
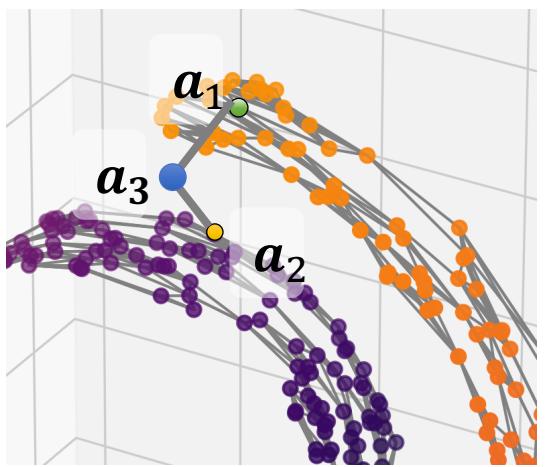
Paths or “walks” on graphs reveal dominant data manifold directions



Steps = 5



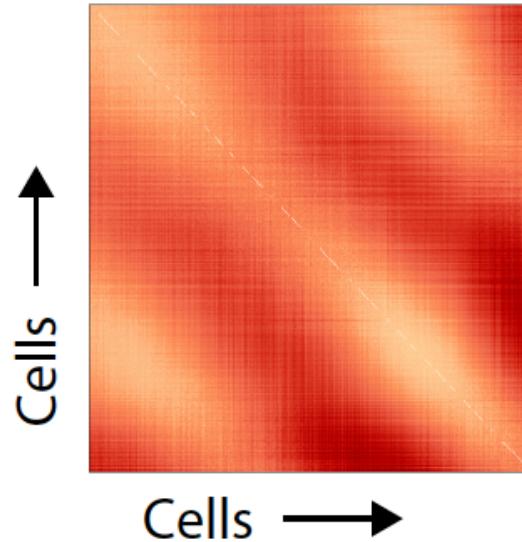




Distance Matrix



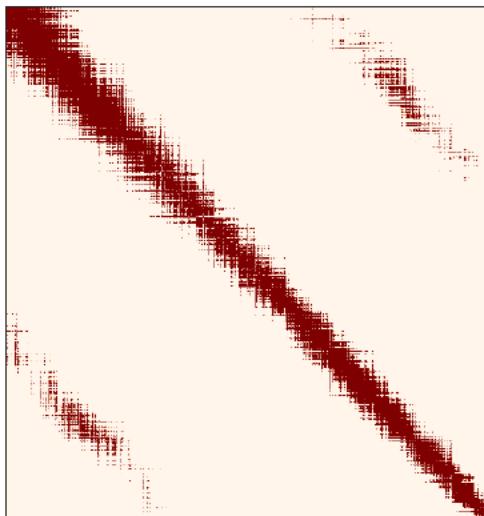
Distance Matrix



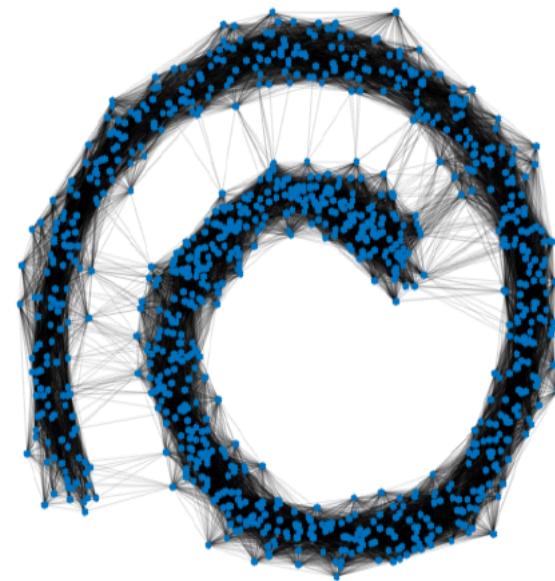
Entries are distances

$$D(x_i, x_j) = \sqrt{||x_i - x_j||}$$

Affinity Matrix



(Gaussian Kernel)



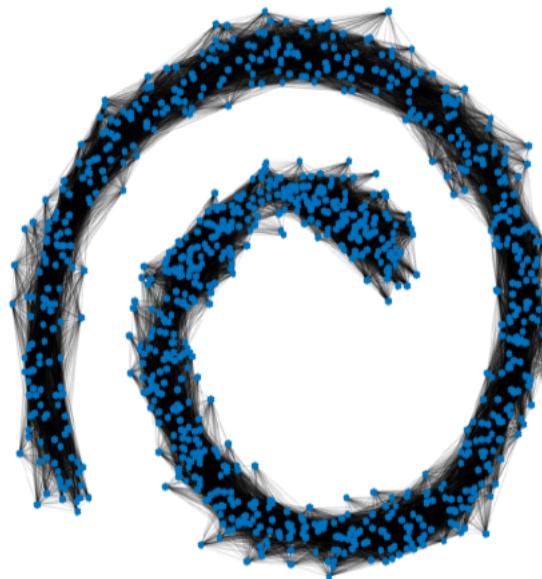
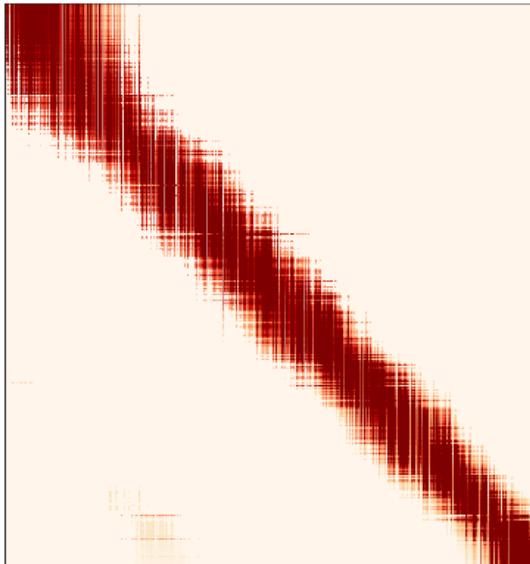
Graph Representation

Entries are affinities

$$A(x_i, x_j) = \exp\left(-\frac{D(x_i, x_j)^2}{\sigma}\right)$$

Powered Markov Matrix— Denoises the Affinities (relationship between points)

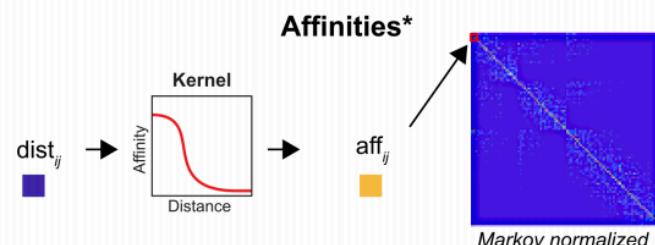
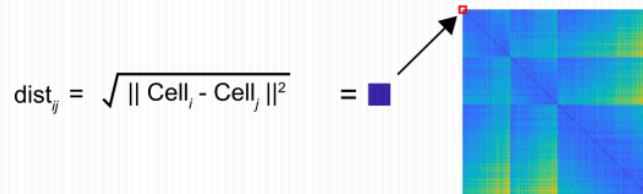
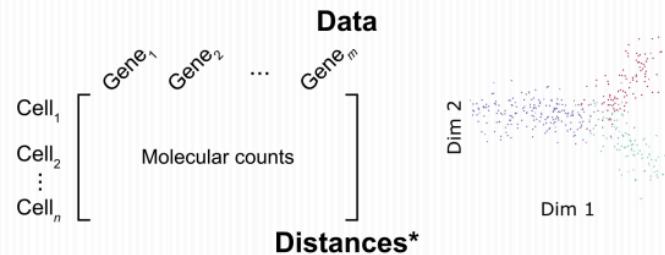
Powered Markov Matrix



Entries are row normalized affinities

$$M(x_i, x_j) = \frac{A(x_i, x_j)}{\sum_j A(x_i, x_j)}$$

PHATE Steps



Powered Diffusion Operator

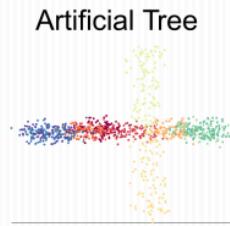
$$A = \left[\begin{matrix} \text{Affinities} \\ \vdots \\ \text{Affinities} \end{matrix} \right]^t$$

Potential Distances

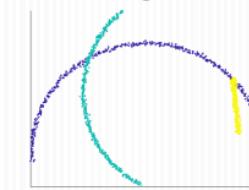
$$dist_{pot,ij} = \sqrt{\| \log(A_j) - \log(A_i) \|^2}$$

Visualization

Artificial Tree



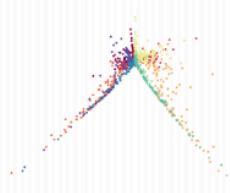
Intersecting Curves



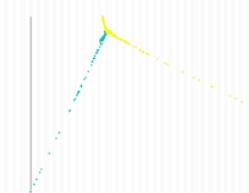
(PCA)



(MDS)

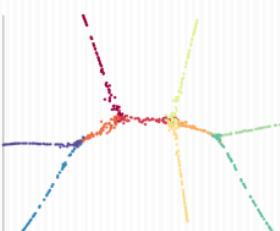


(tSNE)



(DMs)

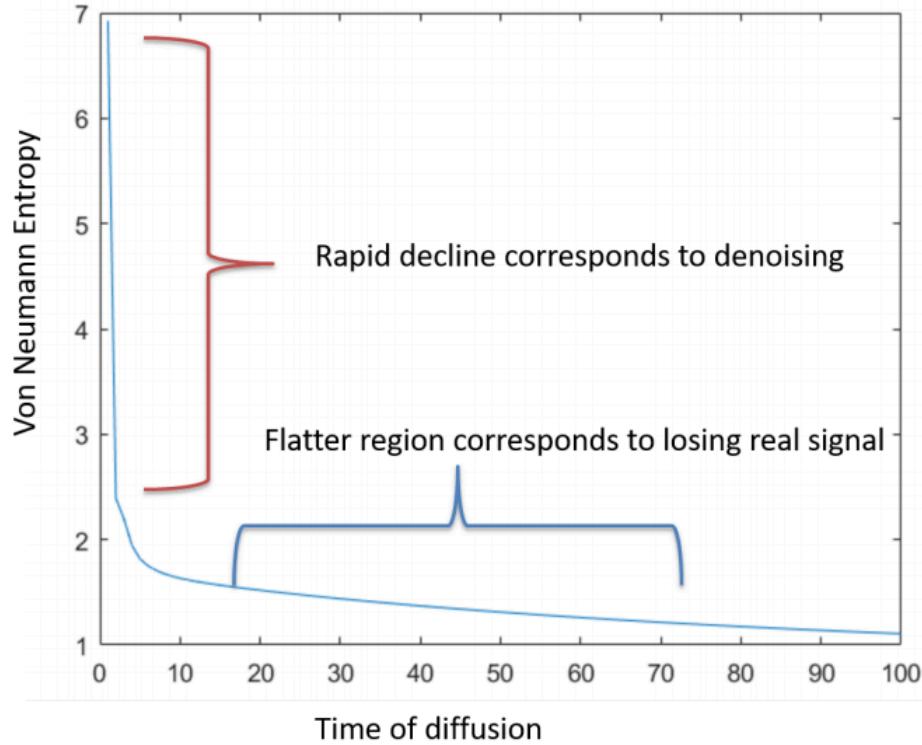
PHATE



PHATE



Time of Diffusion t



Workshop — Krishnaswamy Lab X | Google Search X | +

https://www.krishnaswamylab.org/workshop

Course Schedule

Day 1 – Wednesday, May 20th

Lecture	Download from GitHub	Introduction to scRNA-seq and Preprocessing
Exercise	Run in Google Colab	1.0. Preprocessing Embryoid Body Data (Beginner)
	Run in Google Colab	1.0. Preprocessing Embryoid Body Data (Advanced)
	Run in Google Colab	1.0. Preprocessing Embryoid Body Data (Answer Key)
	Run in Google Colab	1.1. Loading and pre-processing your own data (optional)

Day 2 – Thursday, May 21st

Lecture	Download from GitHub	Manifold Learning and Dimensionality Reduction
Exercise	Run in Google Colab	2.0. Plotting UCI Wine Data
Exercise	Run in Google Colab	2.0. Plotting UCI Wine Data (Answer Key) 
	Run in Google Colab	2.1. Learning Graphs from Data
	Run in Google Colab	2.1. Learning Graphs from Data (Answer Key)
	Run in Google Colab	2.2. Visualizing UCI Wine Data
	Run in Google Colab	2.2. Visualizing UCI Wine Data (Answer Key)
	Run in Google Colab	2.3. PCA on Retinal Bipolar Data
	Run in Google Colab	2.3. PCA on Retinal Bipolar Data (Answer Key)
	Run in Google Colab	2.4. Visualizing Retinal Bipolar Data
	Run in Google Colab	2.4. Visualizing Retinal Bipolar Data (Answer Key)
	Run in Google Colab	2.5. Visualizing Embryoid Body Data (Advanced)
	Run in Google Colab	2.5. Visualizing Embryoid Body Data (Answer Key)

Day 3 – Friday, May 22nd

Lecture	Download from GitHub	Clustering and Data Denoising
Exercise	Run in Google Colab	3.0 Clustering Toy Data (Beginner)
	Run in Google Colab	3.0 Clustering Toy Data (Advanced)
	Run in Google Colab	3.1 Clustering & Denoising Embryoid Body Data (Advanced)
	Run in Google Colab	3.2 Batch correction in PBMCs
Survey	Take on Google Forms	Week 1 Survey 

<https://www.krishnaswamylab.org/workshop>

The screenshot shows a web browser window with the following details:

- Title Bar:** Workshop Materials — Krishnas X
- Address Bar:** https://www.krishnaswamylab.org/workshop-materials#links
- Toolbar:** Includes standard browser icons for back, forward, search, and file operations.

The main content area displays a section titled "Useful links to outside resources".

Useful links to outside resources

Day 1 - Introduction and preprocessing

- [Current best practices in single-cell RNA-seq analysis: a tutorial](#)
 - A useful paper and associated tutorial for analysis of single cell datasets. Remember this is still a set of opinions that were compiled in 2019.
- [Orchestrating Single-Cell Analysis with Bioconductor](#)
 - Probably one of the best Single Cell Analysis Tutorials in R

Day 2 - Dimensionality reduction and manifold learning

- [Selecting the Right Tool for the Job: A comparison of dimensionality reduction algorithms](#)
 - An interactive comparison of tools like PHATE, TSNE, UMAP on real and synthetic data
- [How to Use t-SNE Effectively](#)
 - Interactive explainable demonstrating the effect of changing parameters on t-SNE
- [Understanding UMAP](#)
 - Same as "How to use t-SNE Effectively" but for UMAP
- Resources on Eigenvectors / Eigenvalues
 - ["Eigenvectors and eigenvalues"](#) from the Essence of Linear Algebra Series by 3Blue1Brown - Consider checking out this whole series as it provides some nice animated explanations of complex topics
 - ["Introduction to eigenvalues and eigenvectors"](#) by Khan Academy - Also consider watching the whole ["Alternate coordinate Systems"](#) series if you want to dive deep.
- [Can a Chess Piece Explain Markov Chains? | PBS Infinite Series](#) on YouTube
 - An accessible explanation of random walks and Markov chains

<https://www.krishnaswamylab.org/workshop-materials#links>



Smita Krishnaswamy Daniel Burkhardt



Scott Gigante



Kofi Ansong



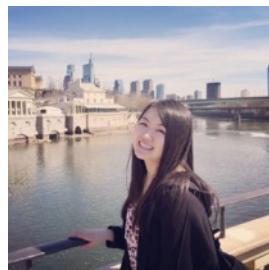
Andrew Benz



Egbert Castro



Pranik Chainani



Joanna Chen



Annie Gao



Michał Gerasimiu



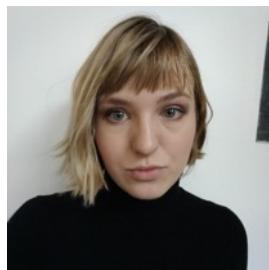
Wes Lewis



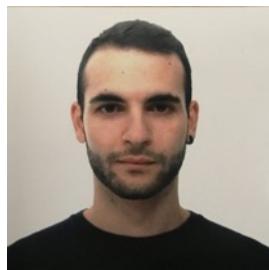
Francesc Lopez



Sameet Mehta



Sasha Safanova



Giacomo Scanavin



Alexander Tong



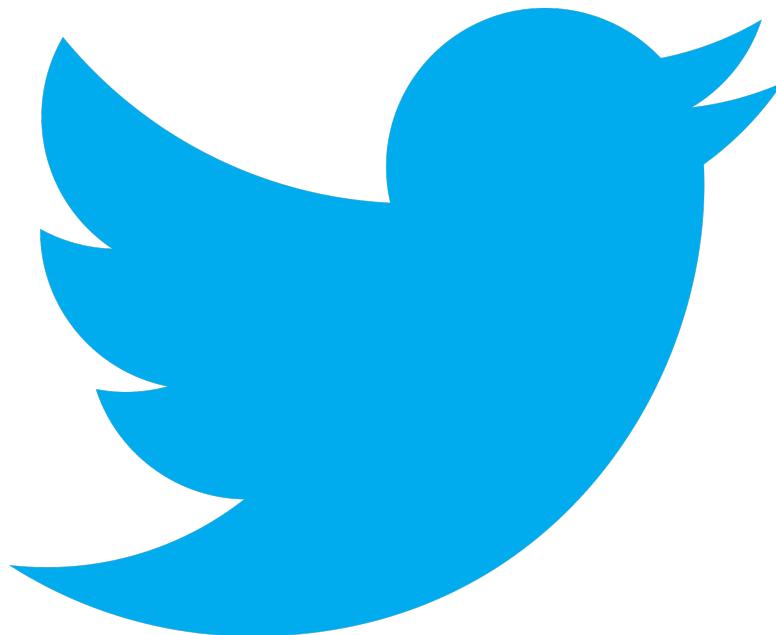
Aarthi



Max Yuan



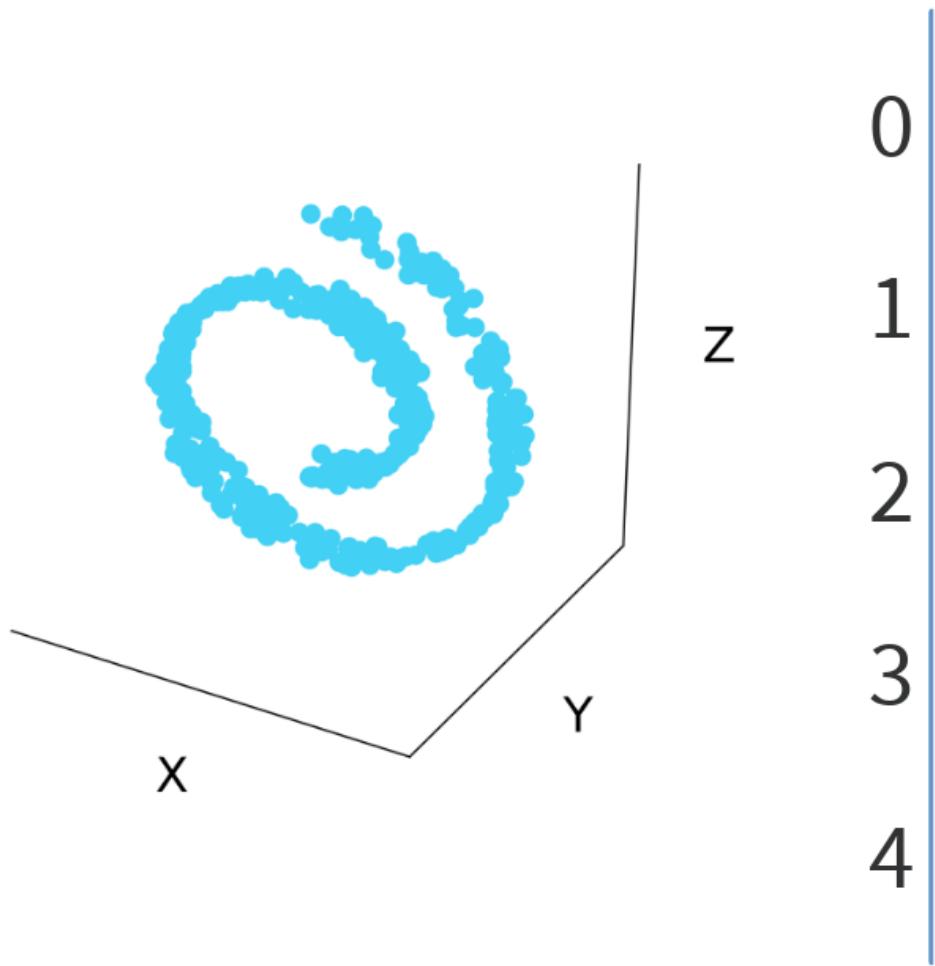
Jeffrey Zhou



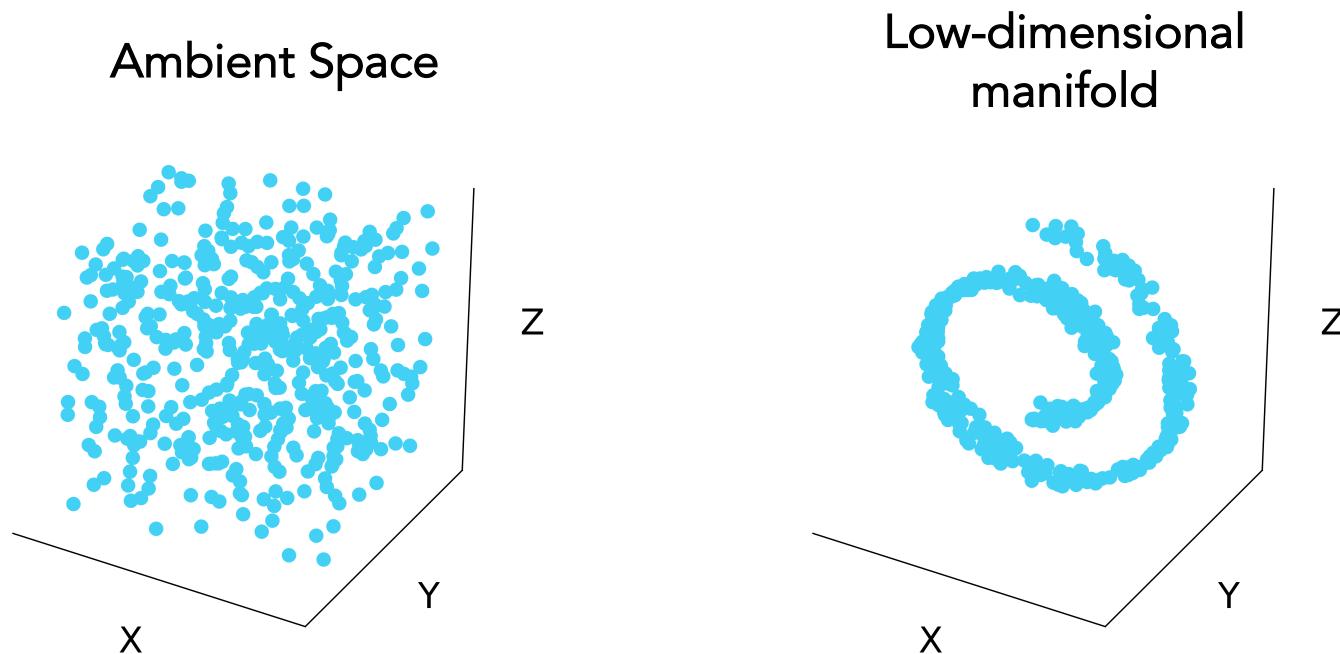
@KrishnaswamyLab
#YaleML

Day 3: Denoising, Batch Correction, and Clustering

What is the intrinsic or latent dimensionality of this data?



Latent structure in high dimensional data

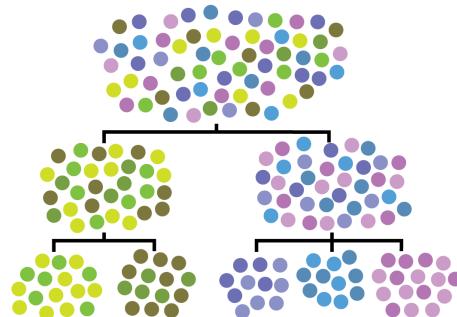


Other uses of affinity matrices, eigenvectors

Data denoising and batch normalization



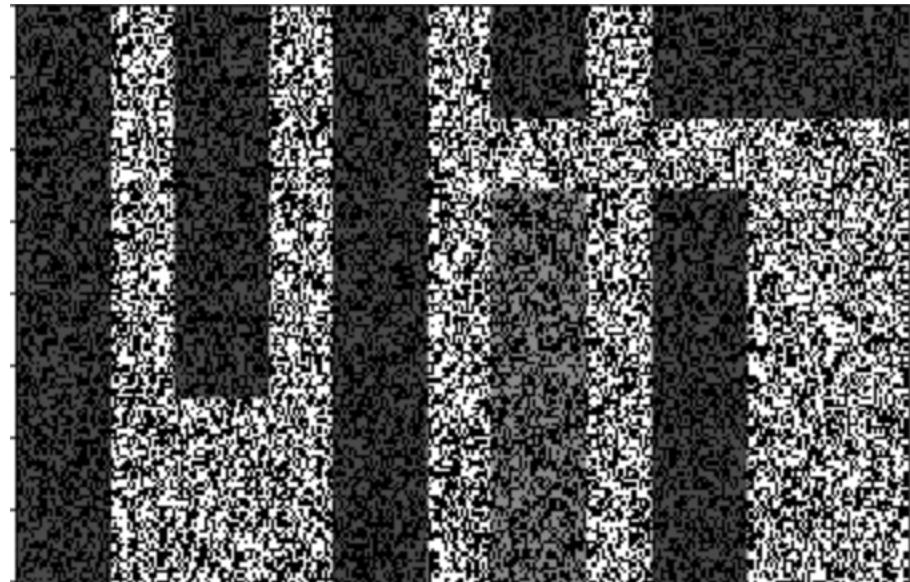
Clustering



Using the manifold model to denoise data

Denoising by eliminating dimensions

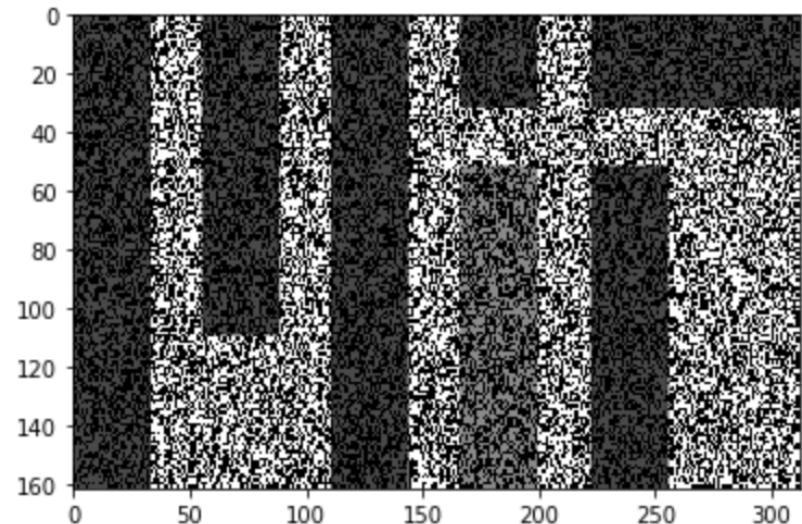
- The number of dimensions in a data that are independent are called RANK
- You can denoise data by lowering rank



Noisy image

Recreate the data without noise

- For this to work you have to have your dimensions be broken up into ***data dimensions and noise dimensions***
- Noisy image on the right is broken up into columns and rows instead
- What do we do?



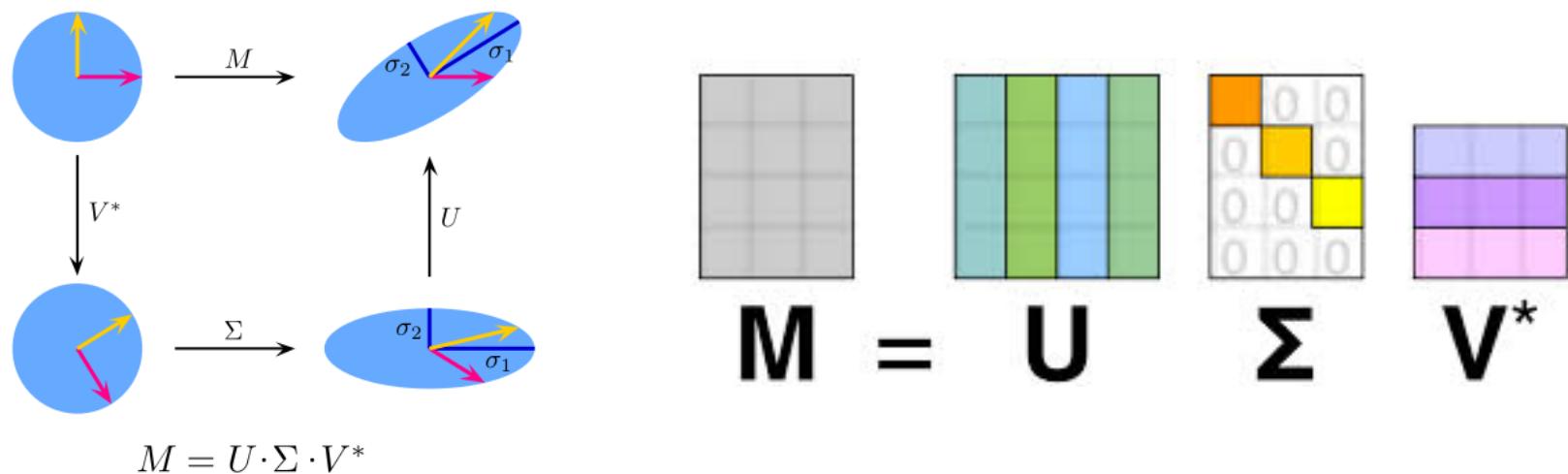
- When poll is active, respond at **PollEv.com/yaleml**
- Text **YALEML** to **22333** once to join

How do we get dimensions that correspond to data "signal" and "noise" ?

PCA splits signal and noise

- The axes with a lot of variation are likely to correspond to data
- Axes with little variation correspond to noise directions
- The amount of variation captured in each eigenvector is given by its eigenvalue
- PC1 will have the highest eigenvalue, captures the most variation ..
 - PC2 next most
 - And so on...

SVD: Process similar to Eigendecomposition on non-square matrices



Singular values are eigenvalues of MM^* or M^*M

Singular vectors are eigenvectors of MM^* or M^*M

For a mean centered feature matrix the singular vectors U are also PCs

Low Rank approximation

- Eliminate the singular vectors with low singular values and recreate the matrix

$$A_{n \times d} = \hat{U}_{n \times r} \Sigma_{n \times d} V^T_{d \times d}$$

The diagram illustrates the Singular Value Decomposition (SVD) of a matrix A . The matrix A is shown as a pink rectangle labeled $n \times d$. It is equal to the product of three matrices: \hat{U} , Σ , and V^T . \hat{U} is a pink rectangle labeled $n \times r$, where r is the rank of the matrix. Σ is a light blue rectangle labeled $n \times d$, with its top-left $r \times r$ block labeled $\hat{\Sigma}$ in pink. V^T is a light blue rectangle labeled $d \times d$, divided into two horizontal sections: a pink section labeled \hat{V}^T and a light blue section below it.

Low Rank Approximation on MIT matrix

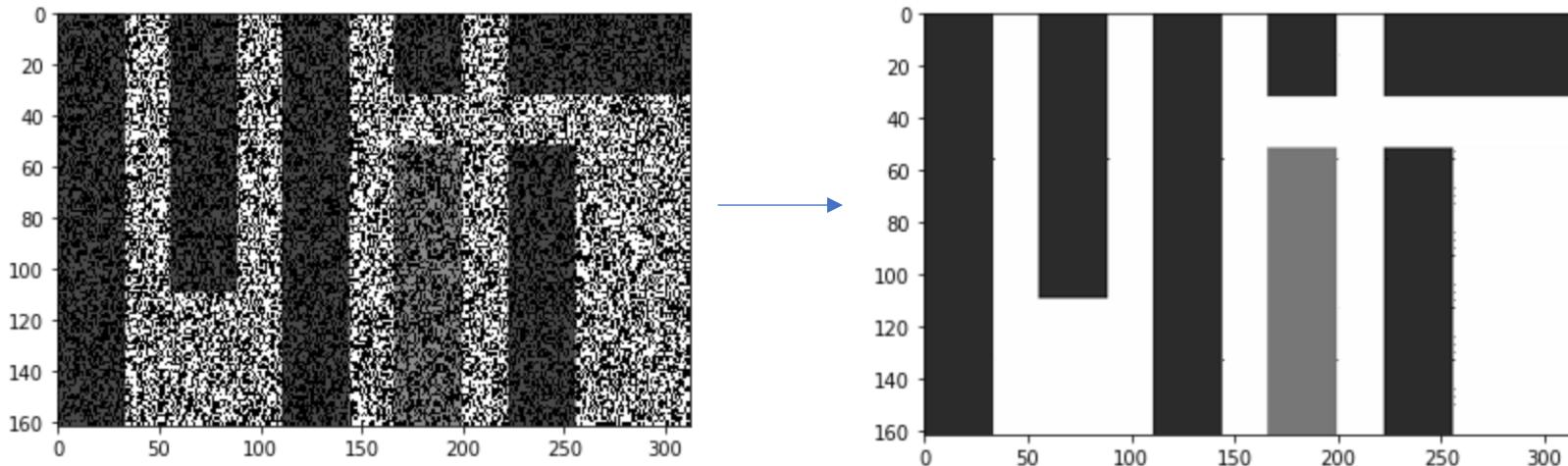
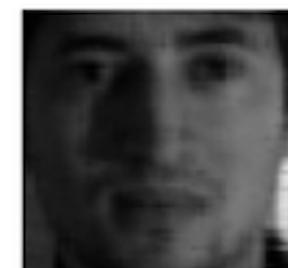
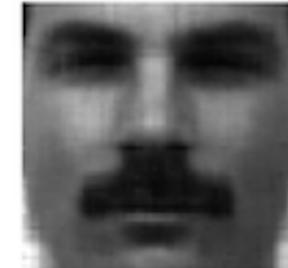
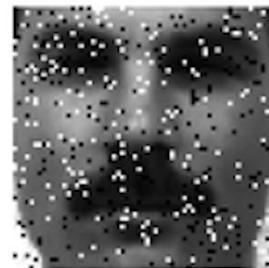
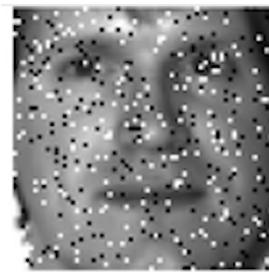


Image from https://medium.com/@amelie_yeh/

Data is smooth, Noise is jumpy



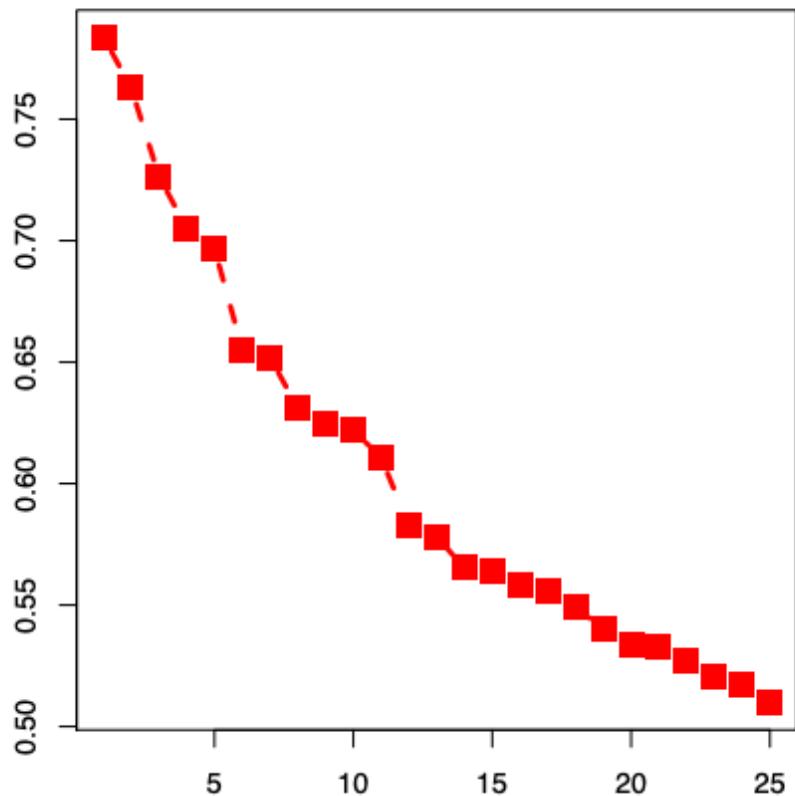
Can take off noise by taking off low-eigenvalued eigenvectors

When poll is active, respond at **PollEv.com/yaleml**

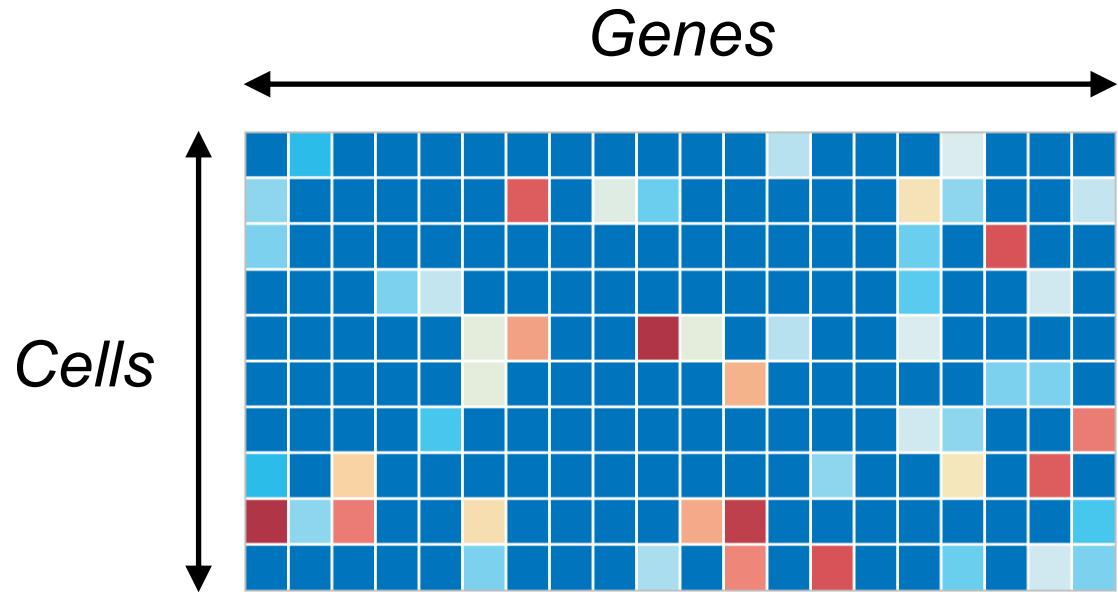
Text **YALEML** to **22333** once to join

How many eigenvectors should we use to reconstruct data? How many eigenvectors are noise? How do we know?

Eigengap



Jumps in eigenvalues
Might give you a clue



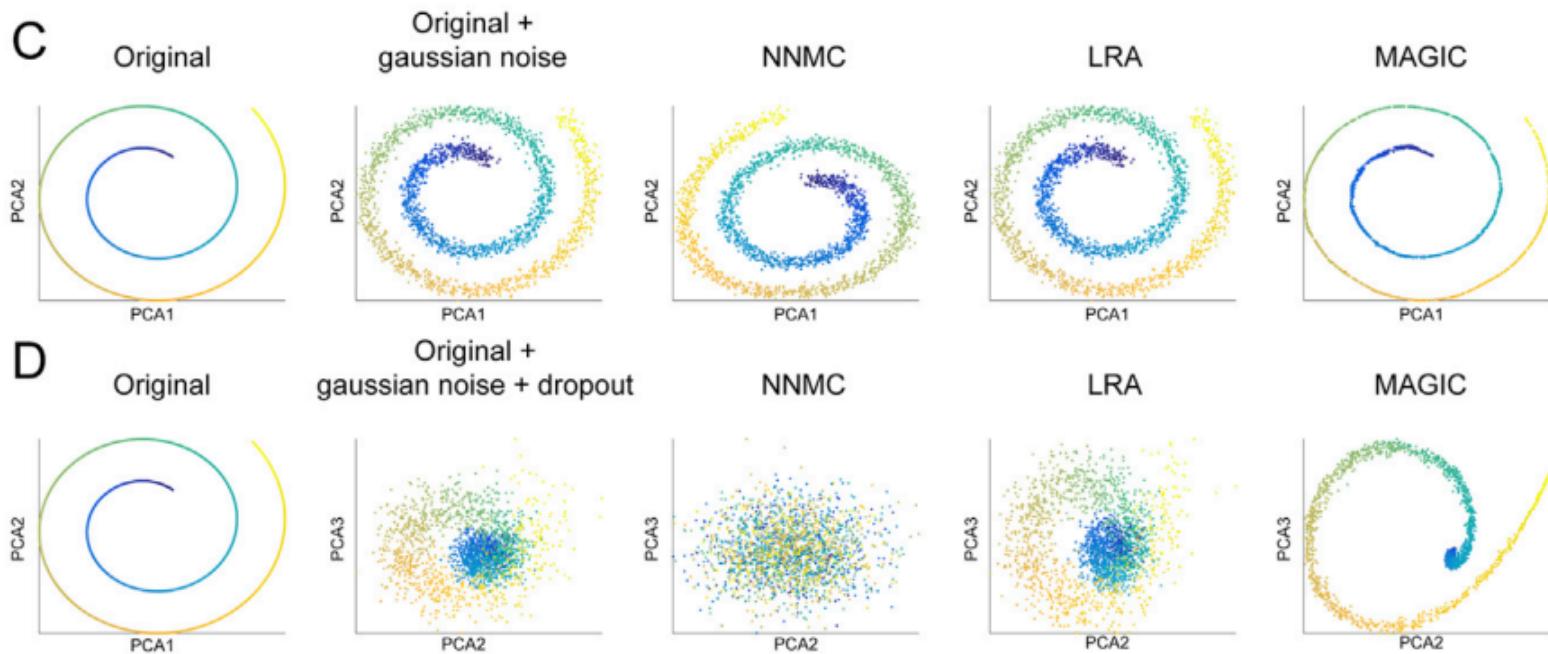
Data is noisy and sparse (scRNA-seq)

When poll is active, respond at **PollEv.com/yaleml**

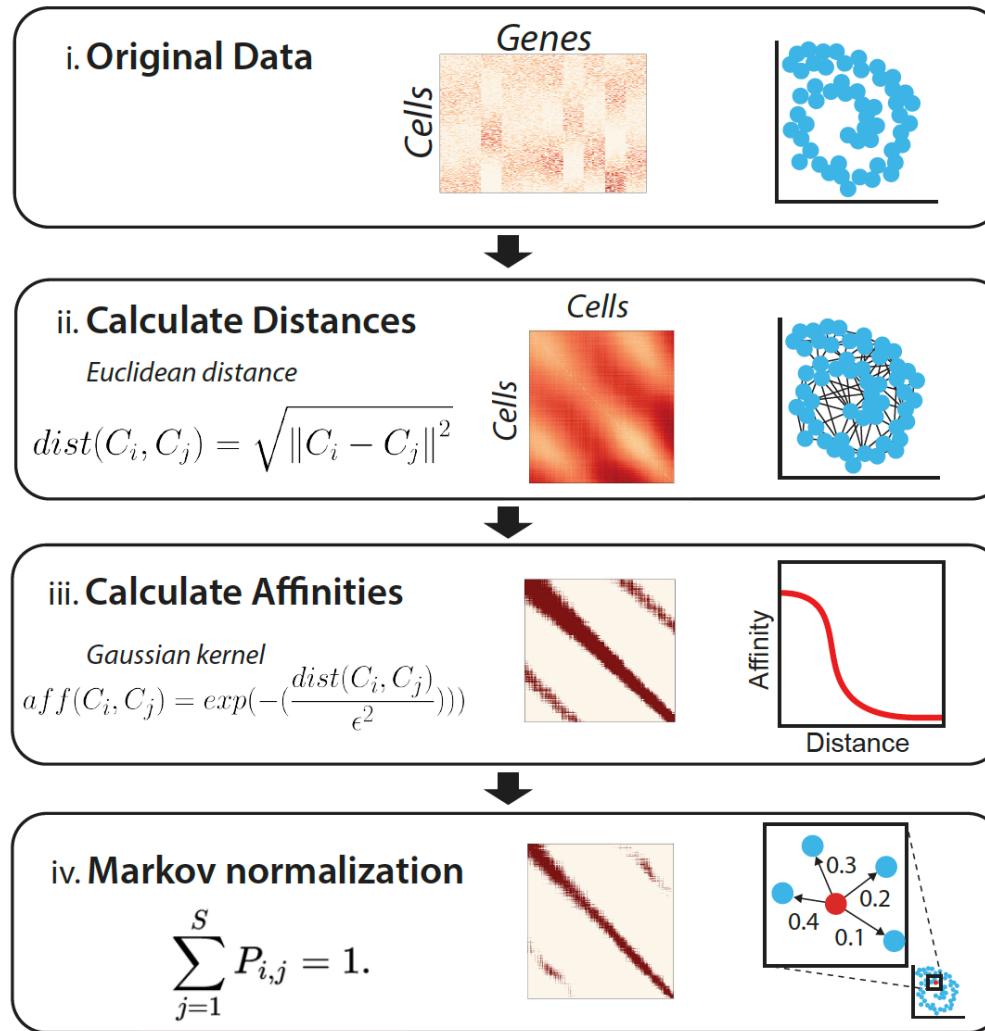
Text **YALEML** to **22333** once to join

Can we use ideas from image denoising to denoise scRNA-seq data?

Low rank approximation for non-linear data

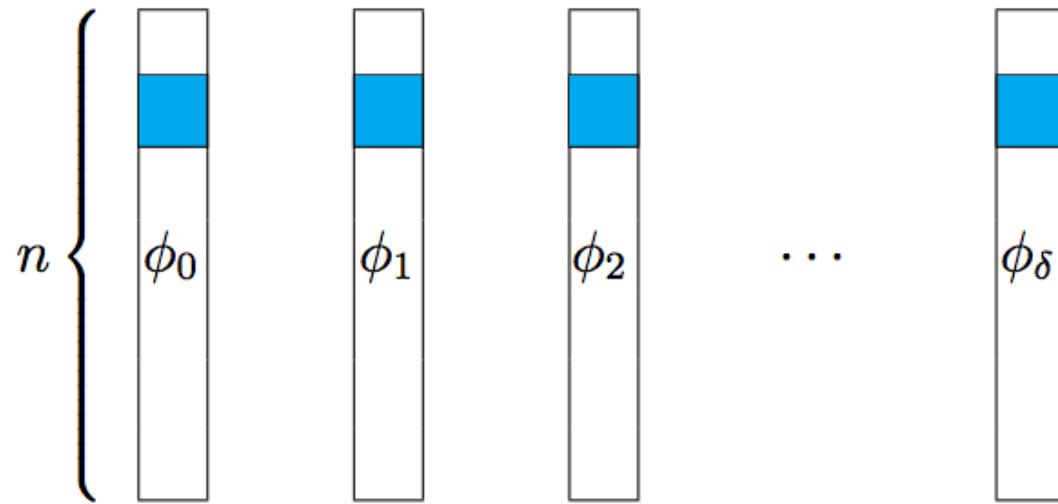


Problem with PCA/Single Value Decomposition (SVD): it takes off linear dimensions in data, noise can be along the non-linear data manifold



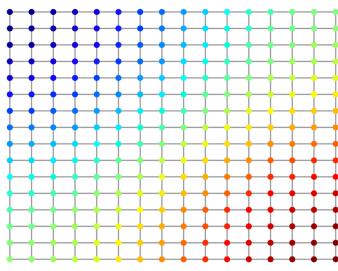
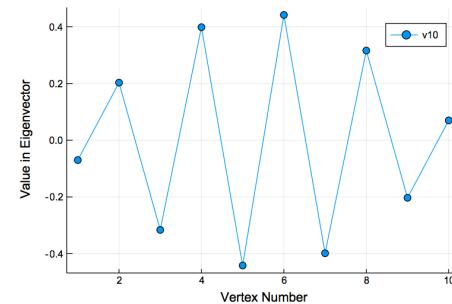
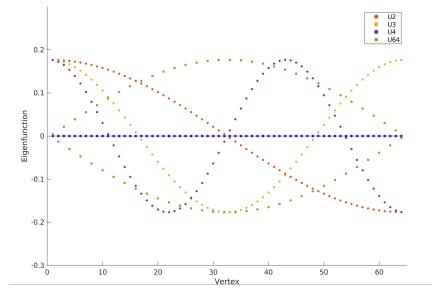
Eigenvectors of Affinity Matrix

$$1 = \boxed{\lambda_0} \geq \boxed{\lambda_1} \geq \boxed{\lambda_2} \geq \dots \geq \boxed{\lambda_\delta} > 0$$

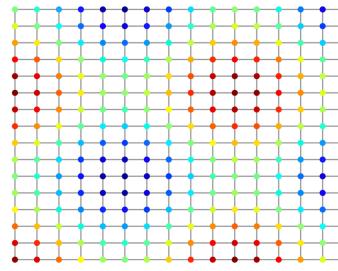


$$x \mapsto \Phi(x) \triangleq [\lambda_0\phi_0(x), \lambda_1\phi_1(x), \lambda_2\phi_2(x), \dots, \lambda_\delta\phi_\delta(x)]^T$$

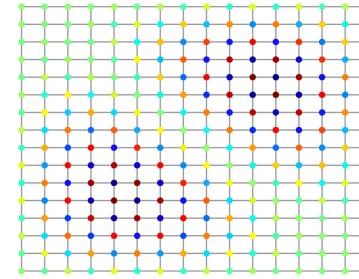
Eigenvectors are frequency harmonics



2nd Eigenvector

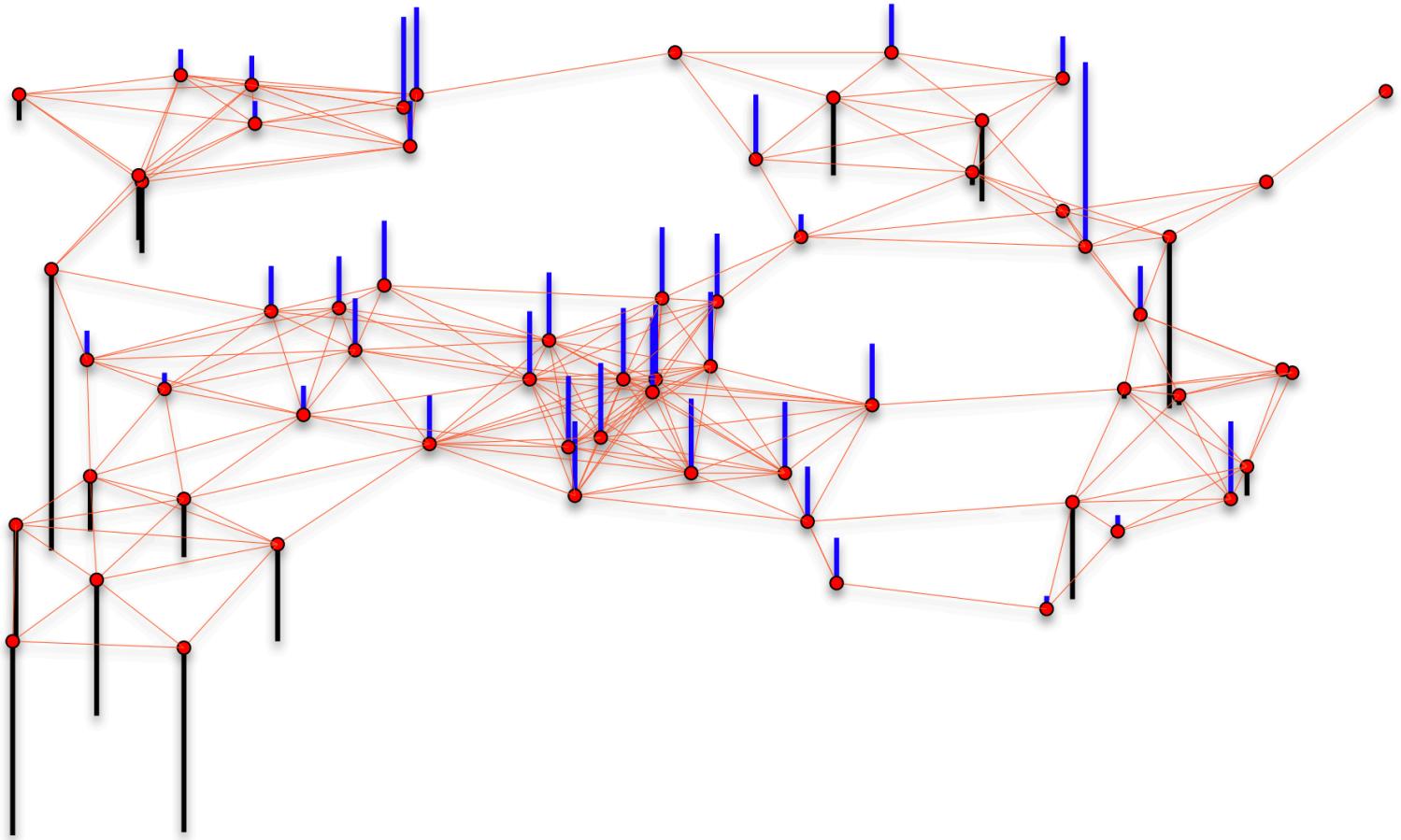


10th Eigenvector



2nd to last eigenvector

Cells are nodes, mRNA gene- counts are signals on a graph



Graph Fourier Transform

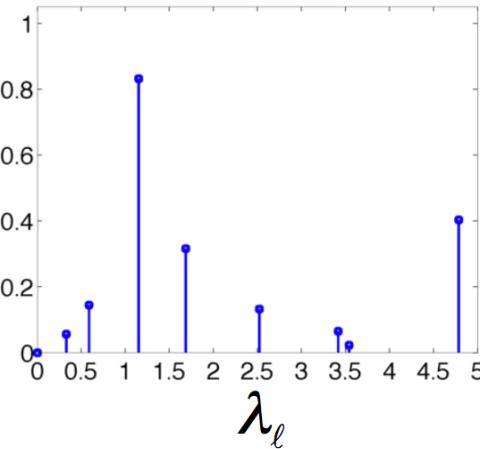
Vertex Domain

Inverse Graph Fourier
Transform = Synthesis

$$\begin{bmatrix} f \end{bmatrix} = \begin{bmatrix} \text{U} \end{bmatrix}^T \times \begin{bmatrix} \hat{f} \end{bmatrix}$$

Graph
Spectral
Domain

$$\hat{f}(\lambda_\ell)$$

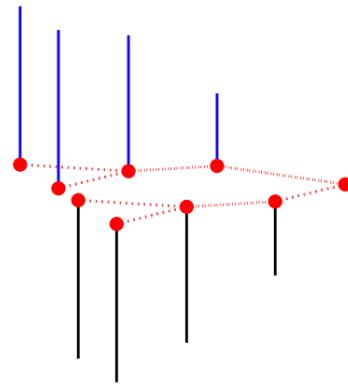


Graph Fourier Transform = Analysis

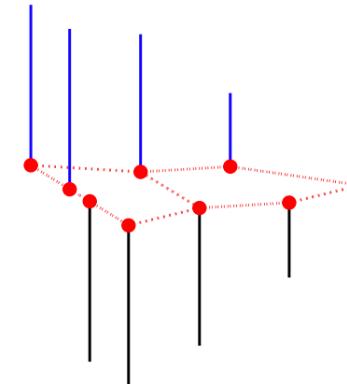
$$\begin{bmatrix} \hat{f} \end{bmatrix} = \begin{bmatrix} \text{U} \end{bmatrix} \times \begin{bmatrix} f \end{bmatrix}$$

Vertex
Domain

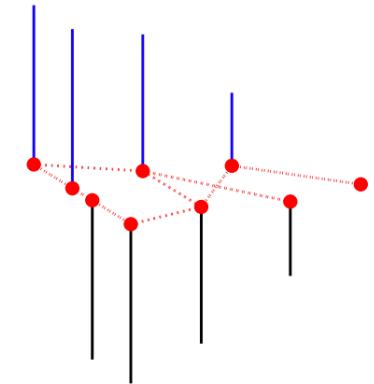
\mathcal{G}_1



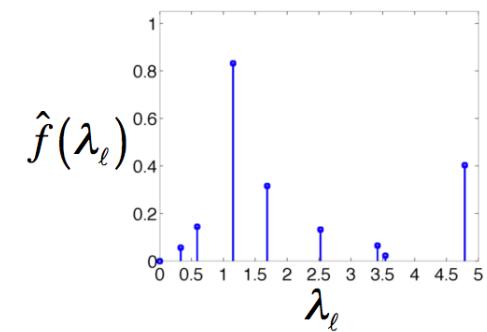
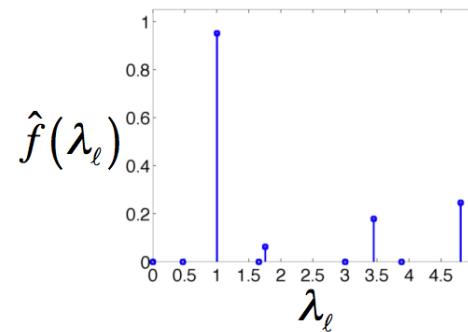
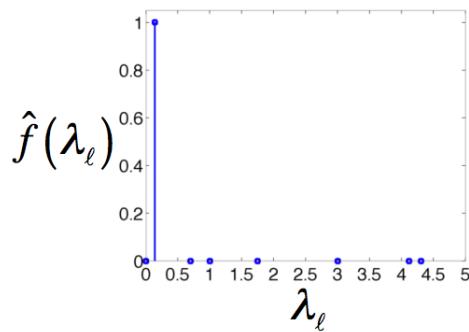
\mathcal{G}_2



\mathcal{G}_3



Graph
Spectral
Domain



When poll is active, respond at **PollEv.com/yaleml**

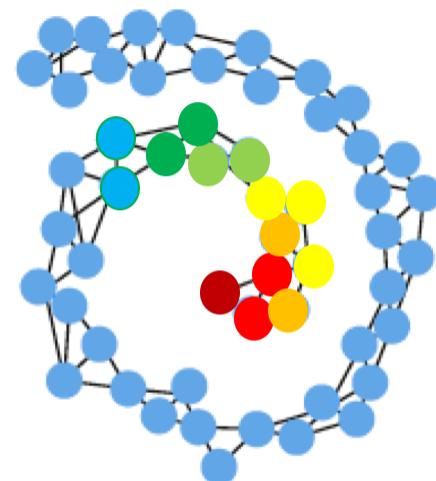
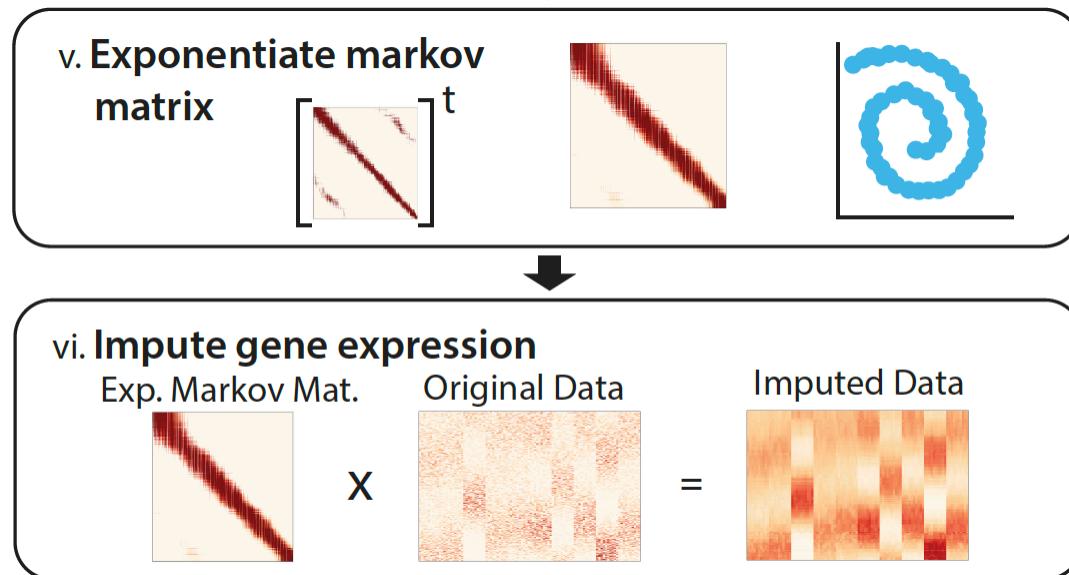
Text **YALEML** to **22333** once to join

How many eigenvectors should we take off?

MAGIC

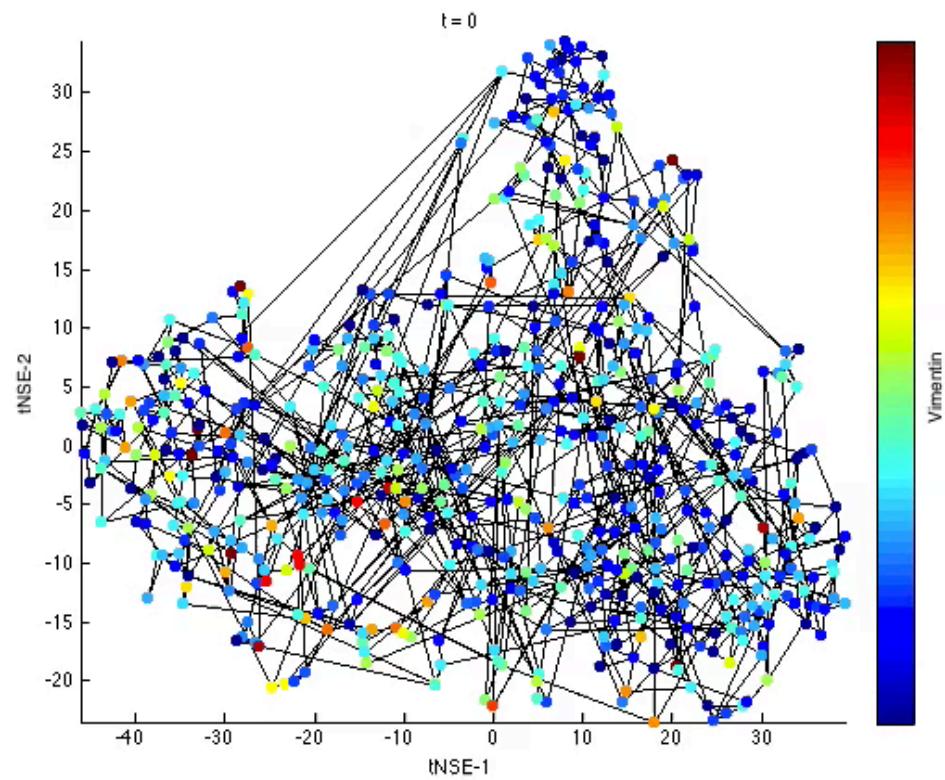
- Softly filters eigenvectors, down-weights them in a regular scheme rather than totally taking them off
- This is called low-pass filtering on the spectrum of the graph

Imputation Step = Smoothing on Graph



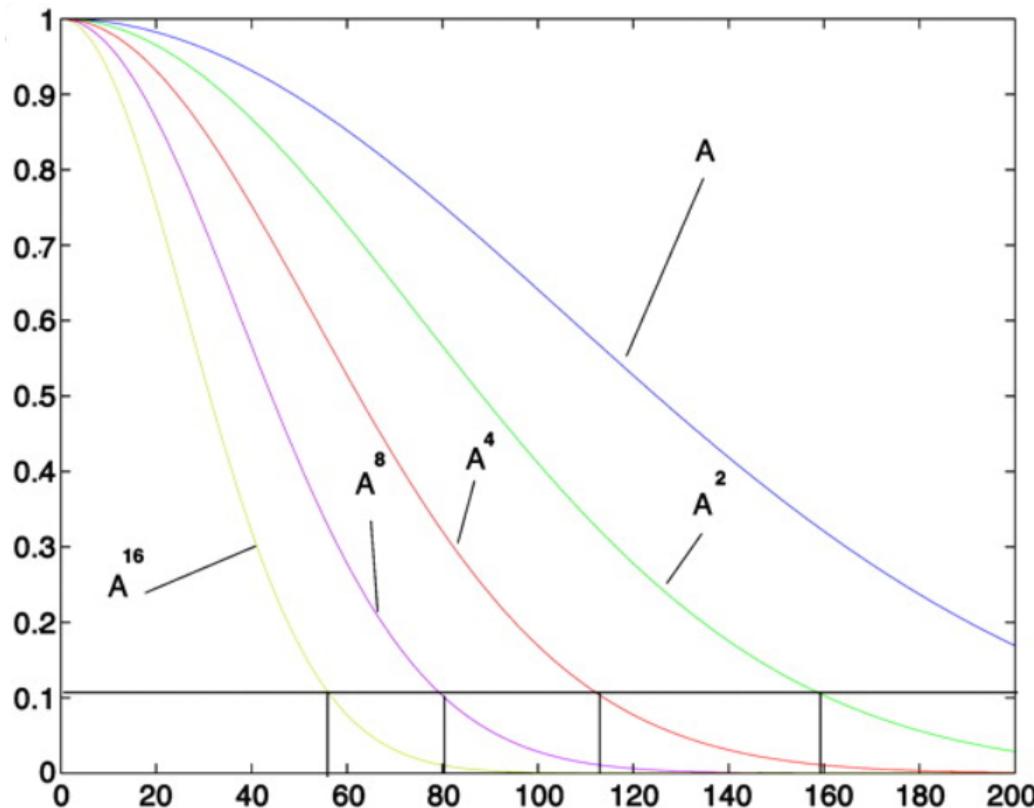
Vertex Domain

- Smooths signal on graph
- Takes weighted average of neighbor

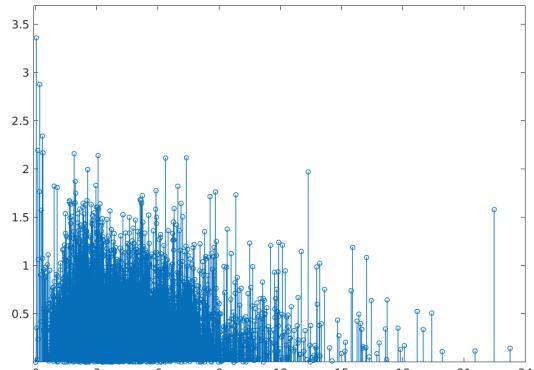


Low pass filter of Eigenvectors

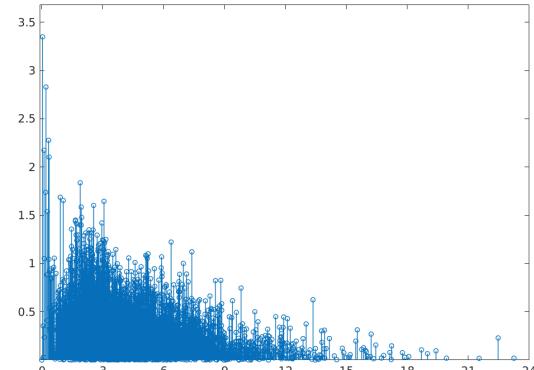
$$\Phi_t(x_i) : x_i \longmapsto [\lambda_1^t \phi_1(i), \lambda_2^t \phi_2(i), \lambda_3^t \phi_3(i), \dots, \lambda_{M-1}^t \phi_{M-1}(i)] \in \mathbb{R}^{M-1}$$



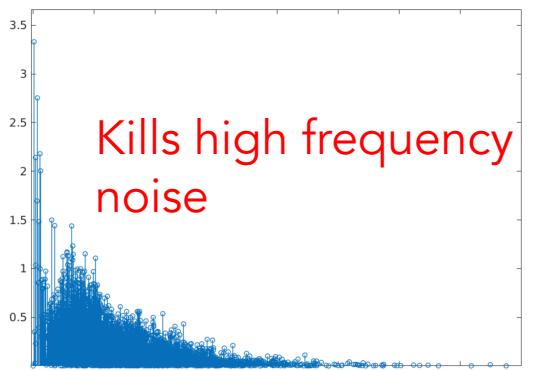
Frequency domain



No Smoothing

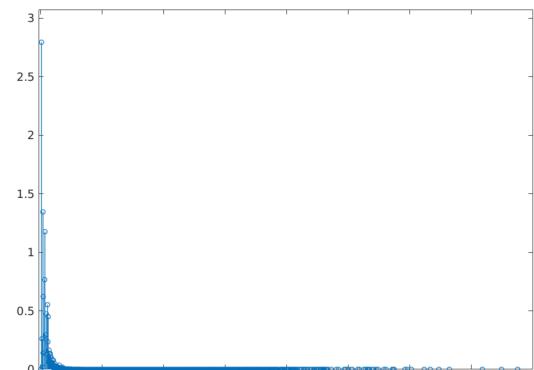


T=2

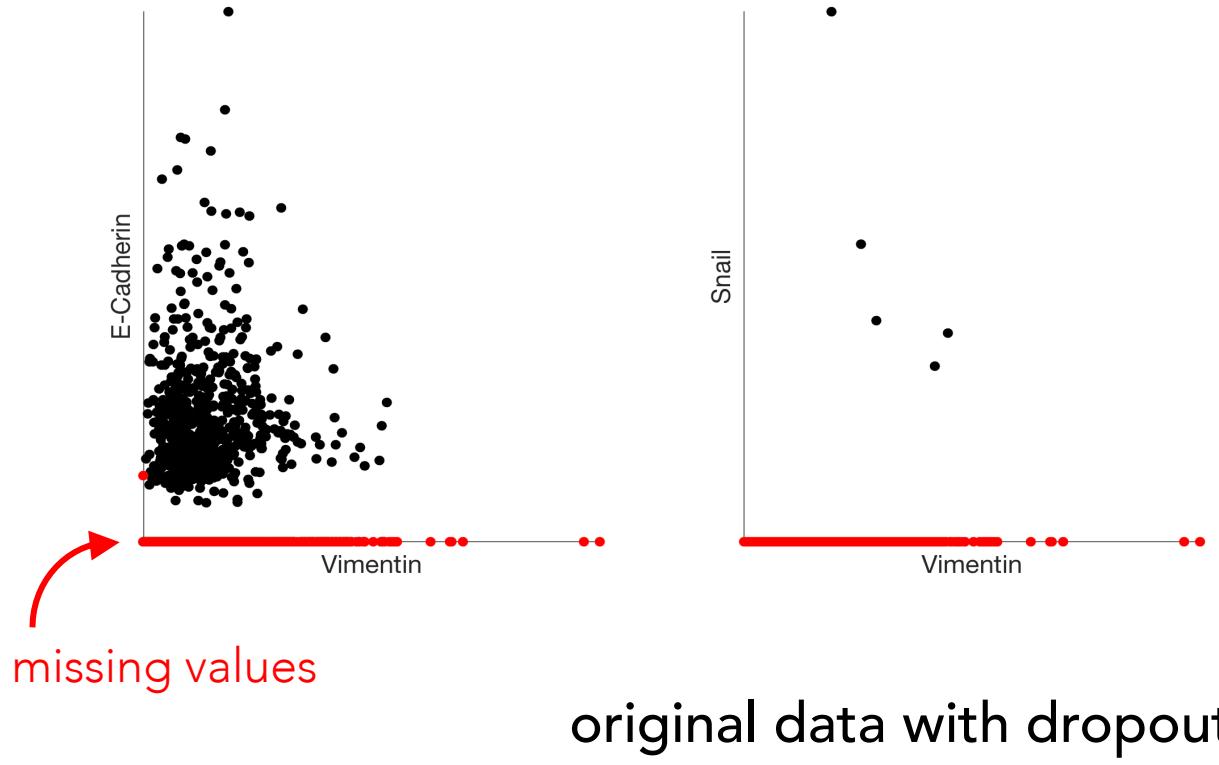


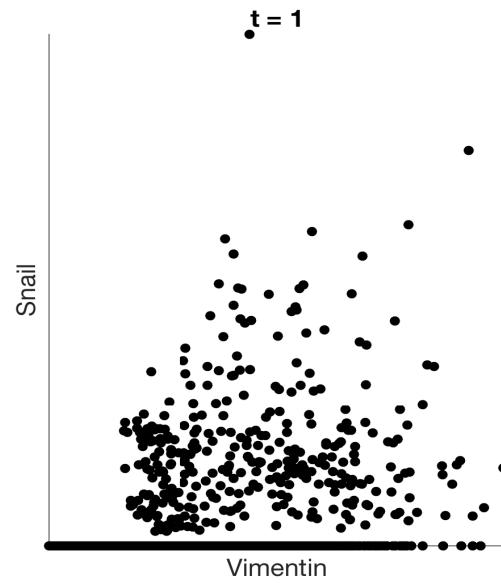
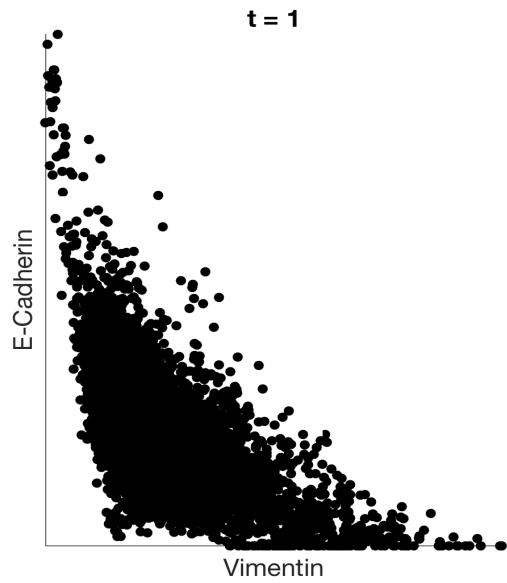
T=5

Kills high frequency
noise

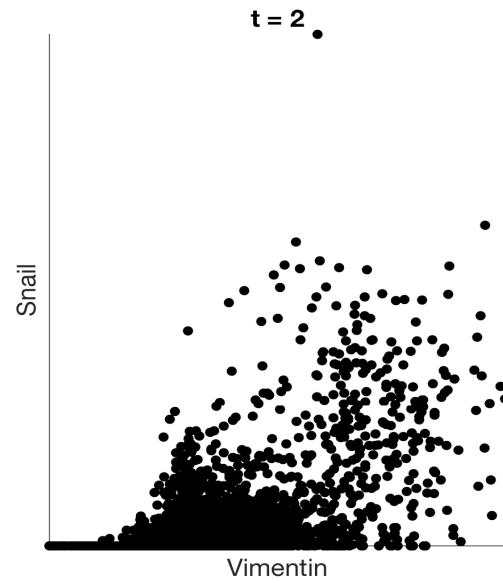
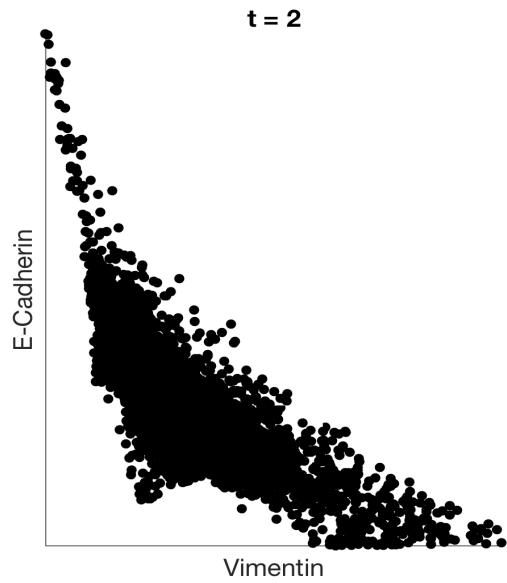


T=100

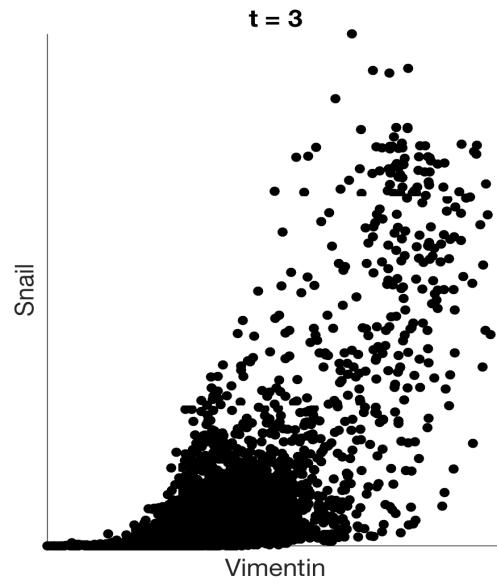
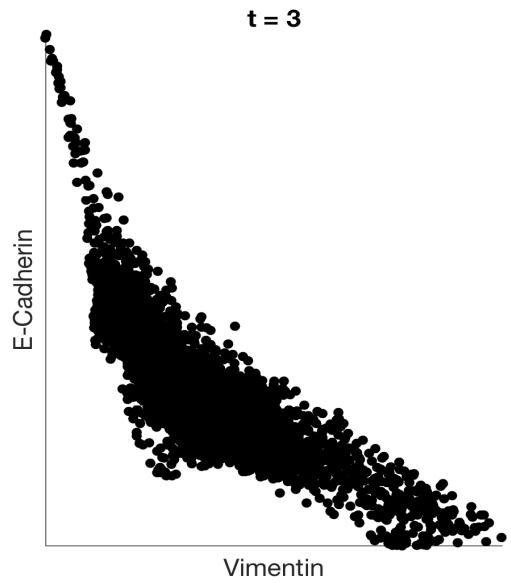




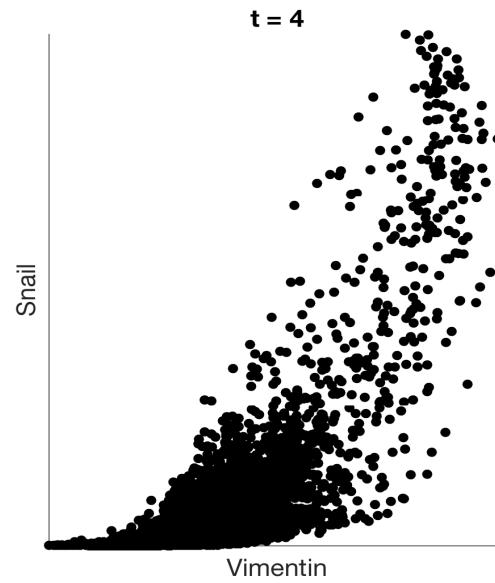
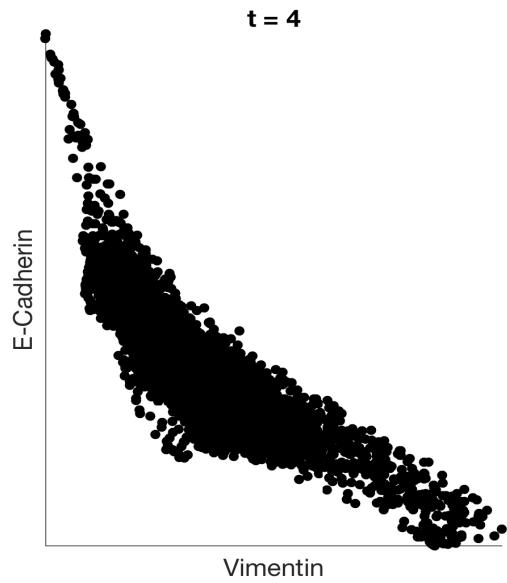
imputation with MAGIC



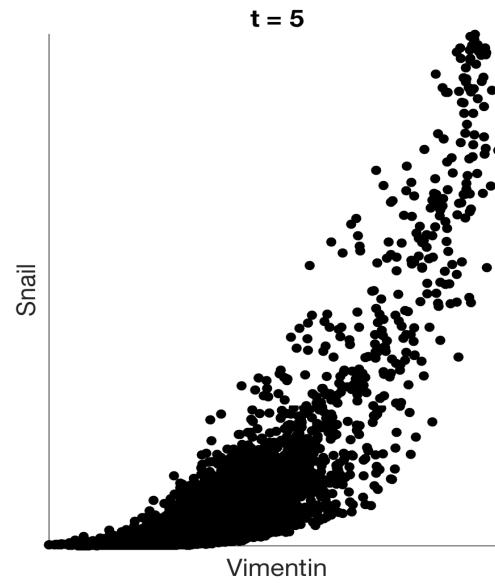
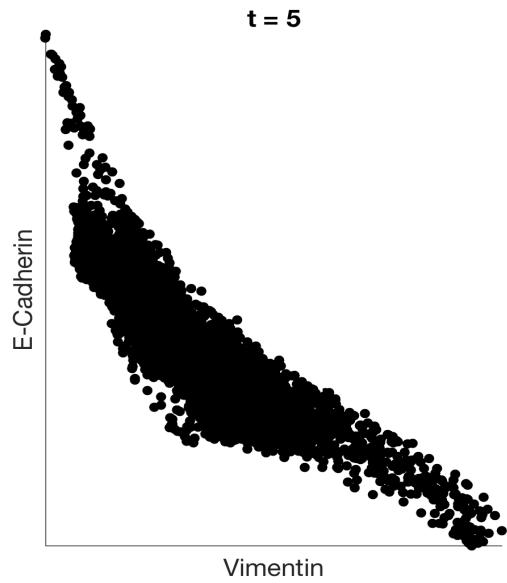
imputation with MAGIC



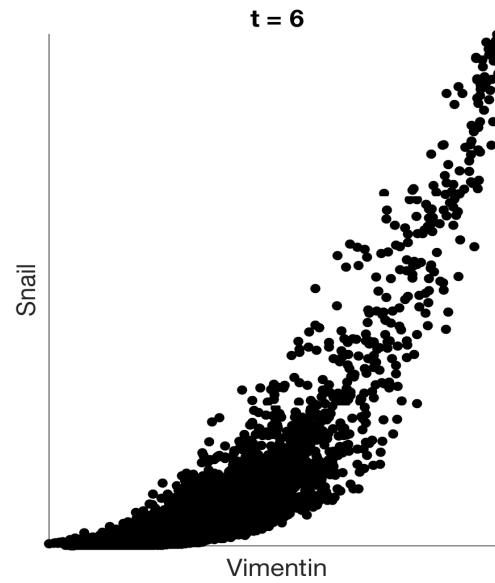
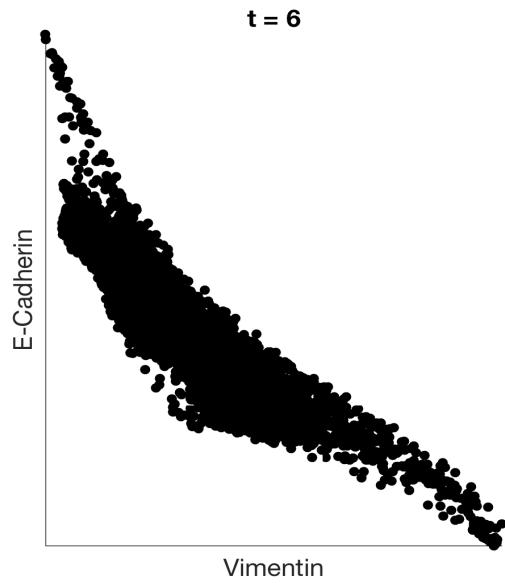
imputation with MAGIC



imputation with MAGIC



imputation with MAGIC

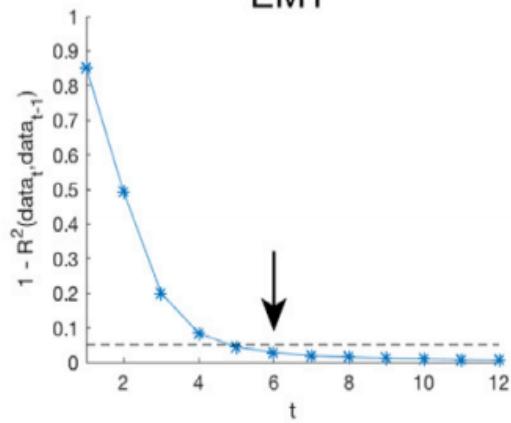


imputation with MAGIC

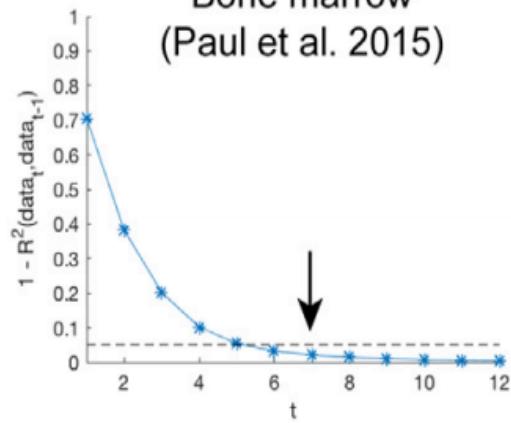
What t should we choose?

C

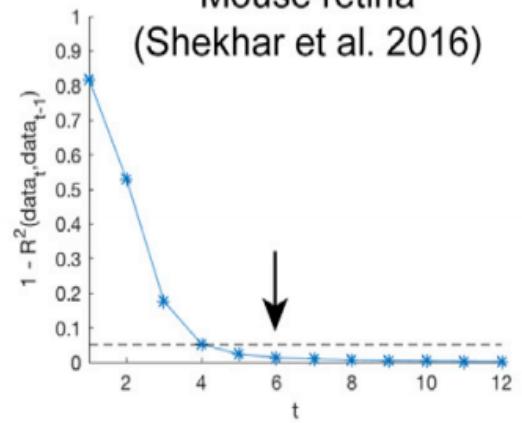
EMT



Bone marrow
(Paul et al. 2015)

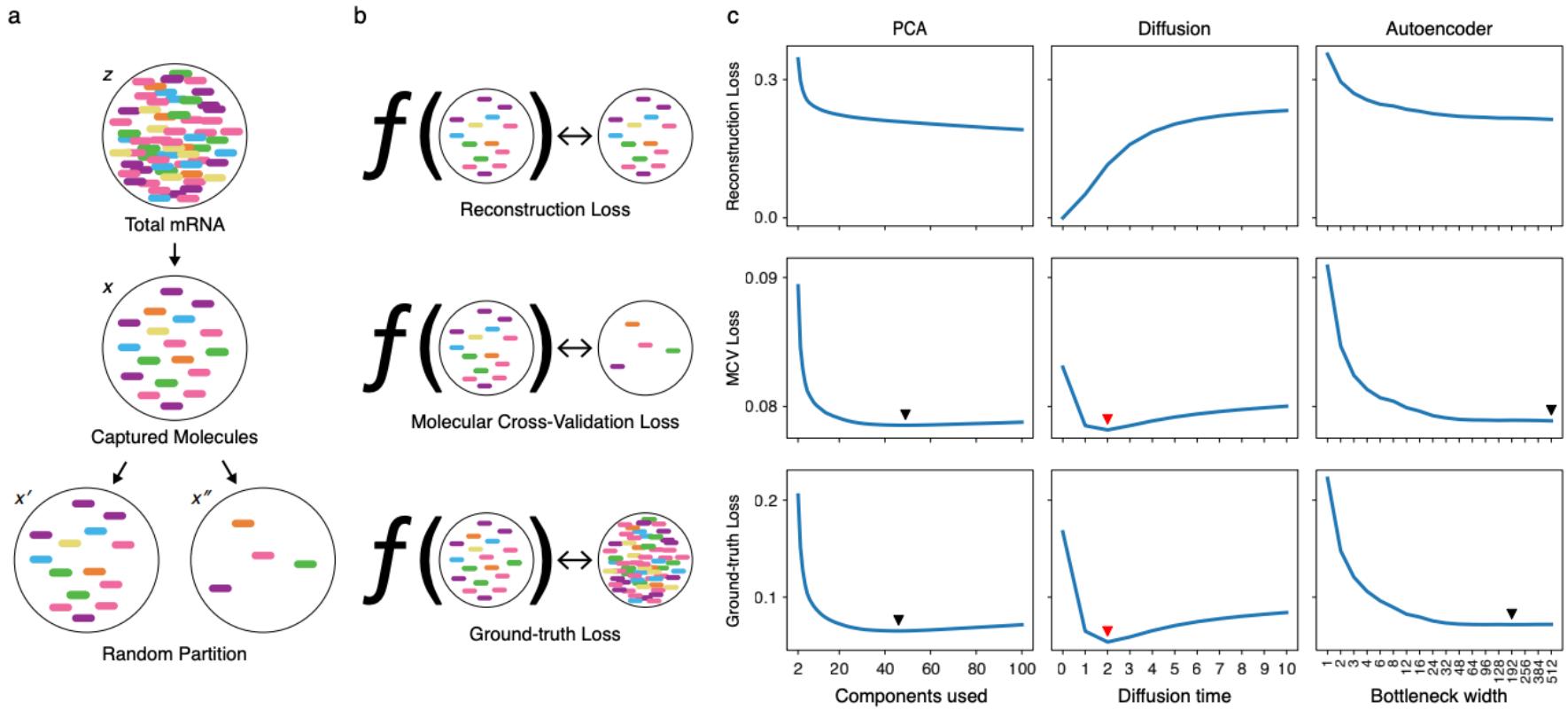


Mouse retina
(Shekhar et al. 2016)



Separate “learning phase” from “stable phase”, quit when data stabilizes

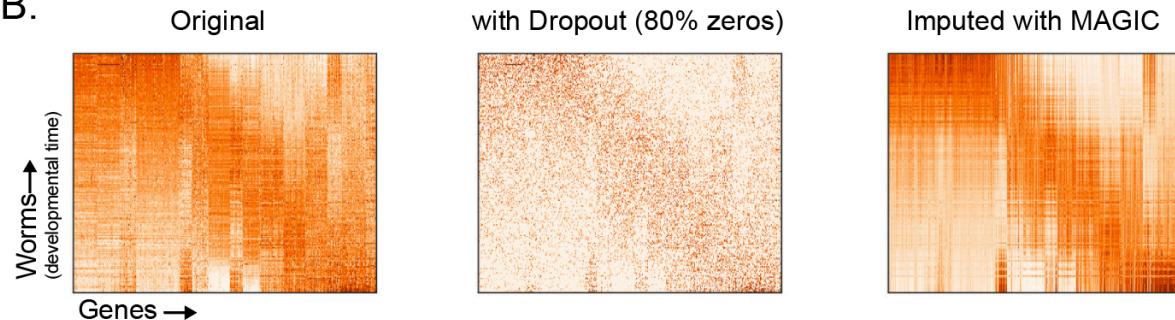
Molecular Validation



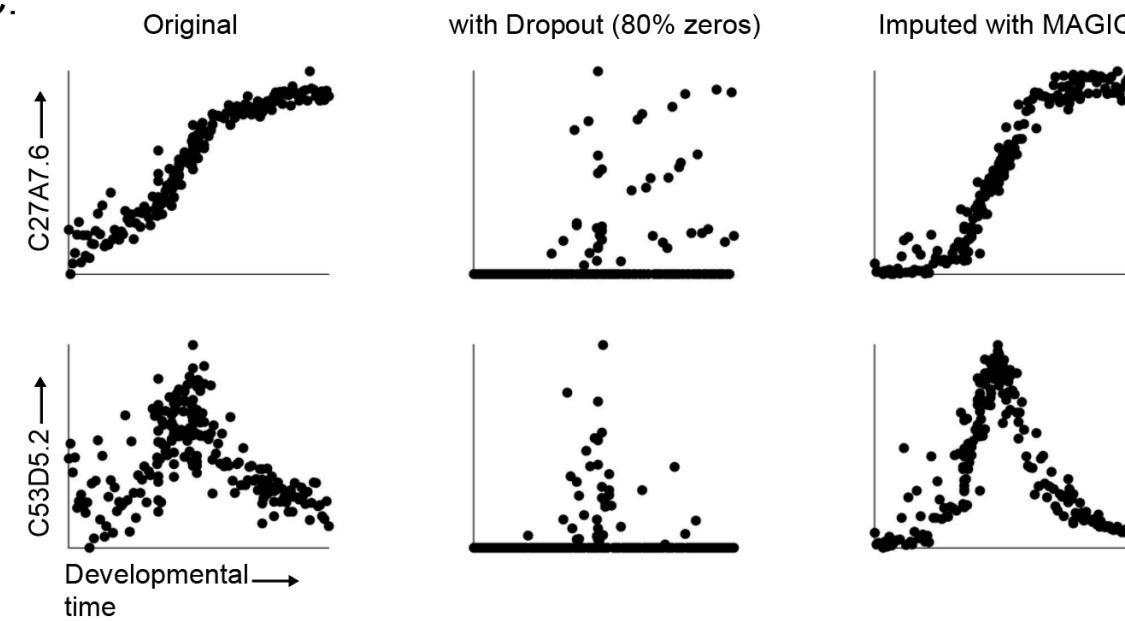
<https://www.biorxiv.org/content/10.1101/786269v1.abstract>

MAGIC recovers gene-gene relationships in an artificially dropped-out dataset

B.



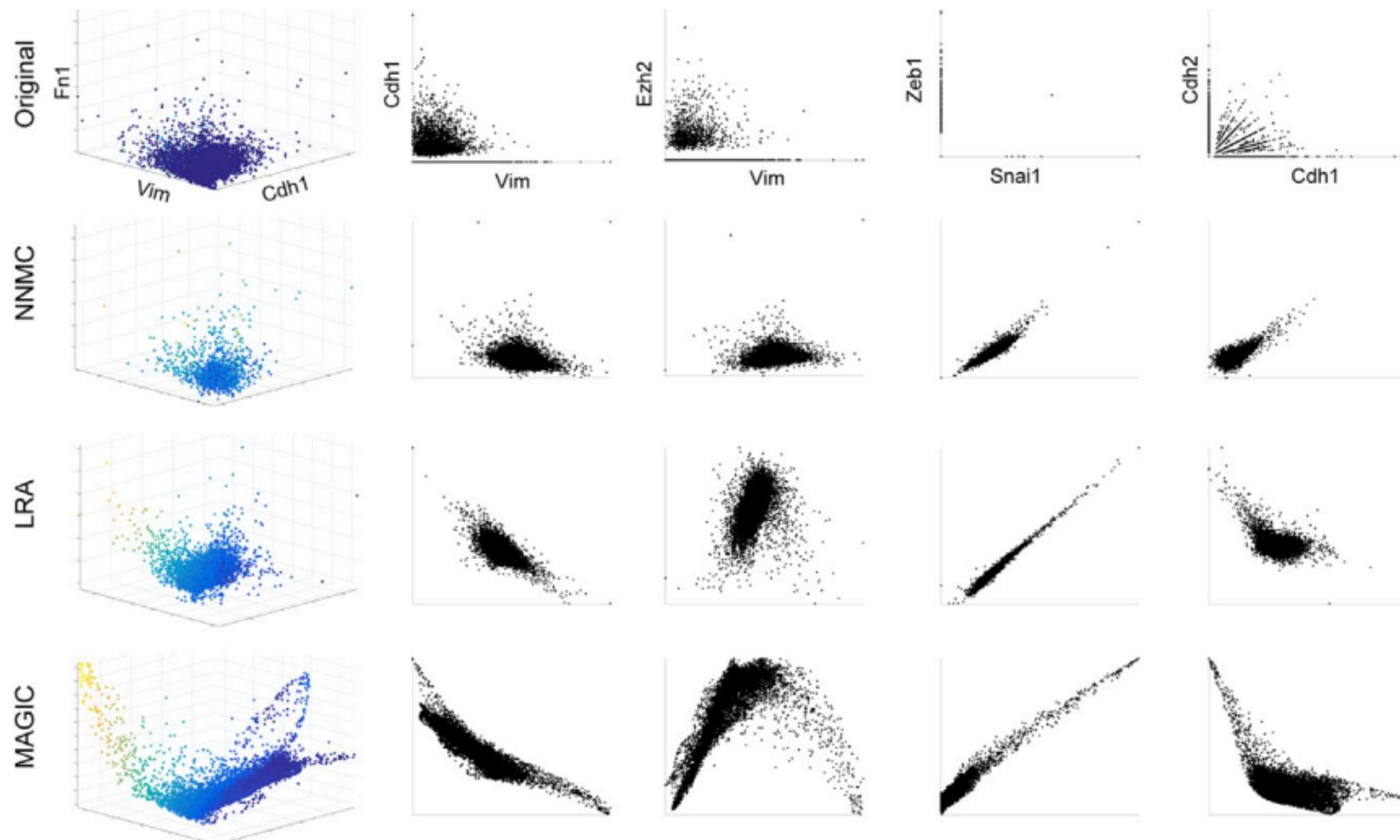
C.



Denoising vs Missing value Imputation

- Denoising assumes that data has some noise, all entries have noise added to them
- Missing value completion methods assume that the existing values are correct and that some values are missing (not measured)
- Matrix completion is a way of filling in missing values
- There is some literature in the single cell world trying to differentiate between true zeros and false zeros
 - People disagree on how this should be done
 - With a denoising viewpoint it is not necessary to distinguish

Comparing MAGIC to LRA and matrix completion

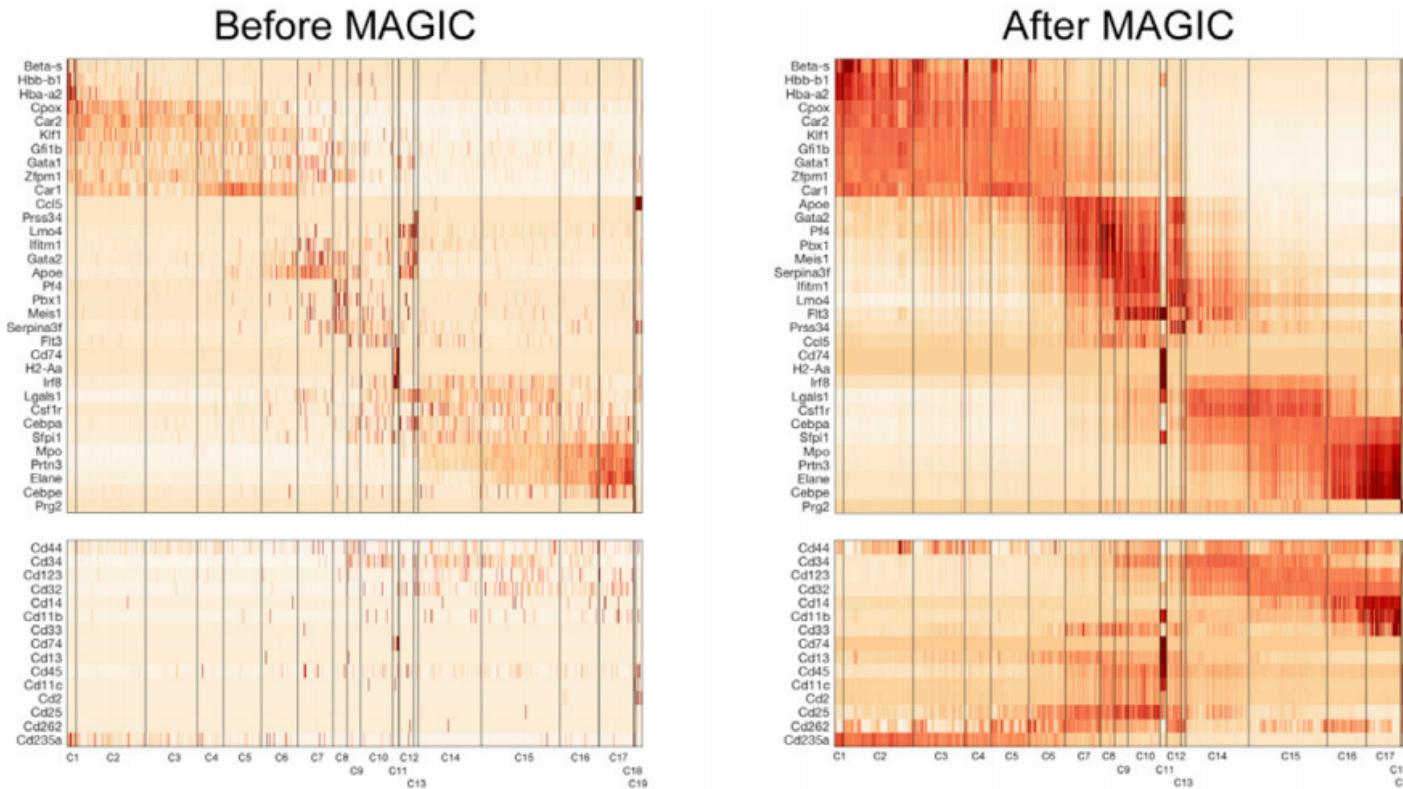


When poll is active, respond at **PollEv.com/yaleml**

Text **YALEML** to **22333** once to join

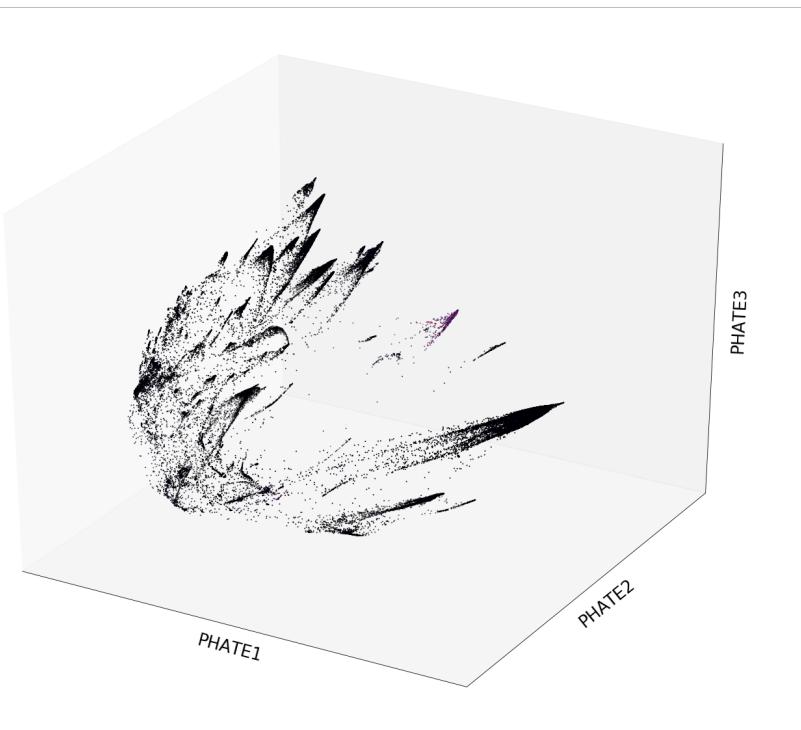
For what tasks is denoising necessary?

Restoring expressions of characteristic markers

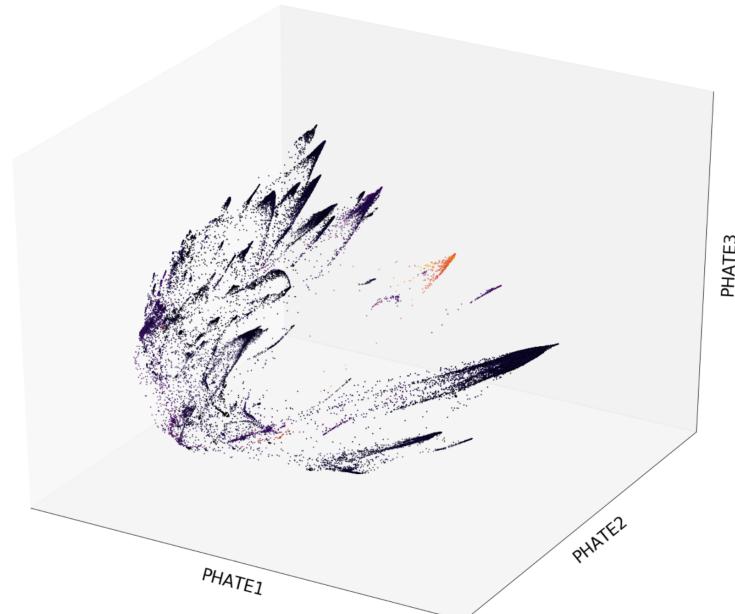


GFAP - marker for astrocytes

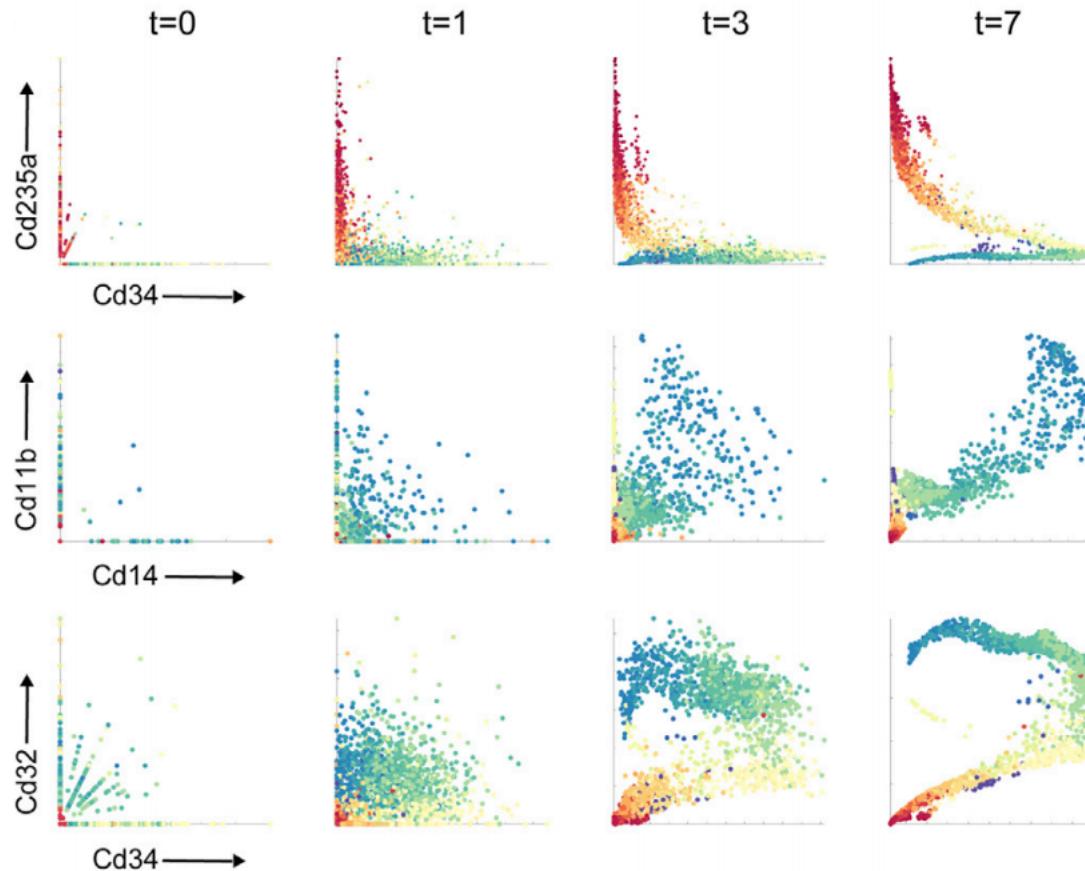
With MAGIC



Without MAGIC



Understanding Gene Gene Relationships



Summary of data denoising

- Diffusing or smoothing values over a graph can denoise data
- This kind of denoising is similar to averaging expression values across neighbors
- More diffusion = more denoising

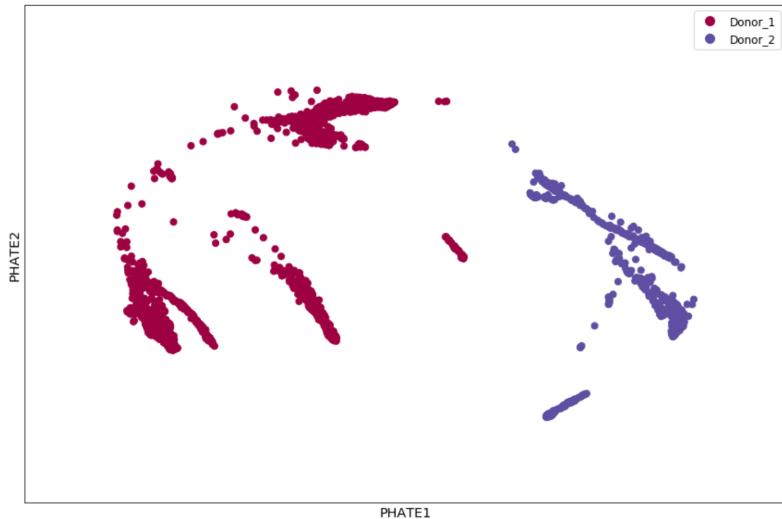
What questions do you have?
Please submit on Slack

Batch correction

Batch Effects in Single Cell

- **Systematic, non-biological differences between samples due to measurement conditions**
- Differences could be due to ambient conditions (temperature, humidity), machine calibration, differences in titration, bead/antibody batch, etc
- Sometimes refers to actually actual biological differences but those that are “uninteresting” (background demographics rather than immediate drug effect).

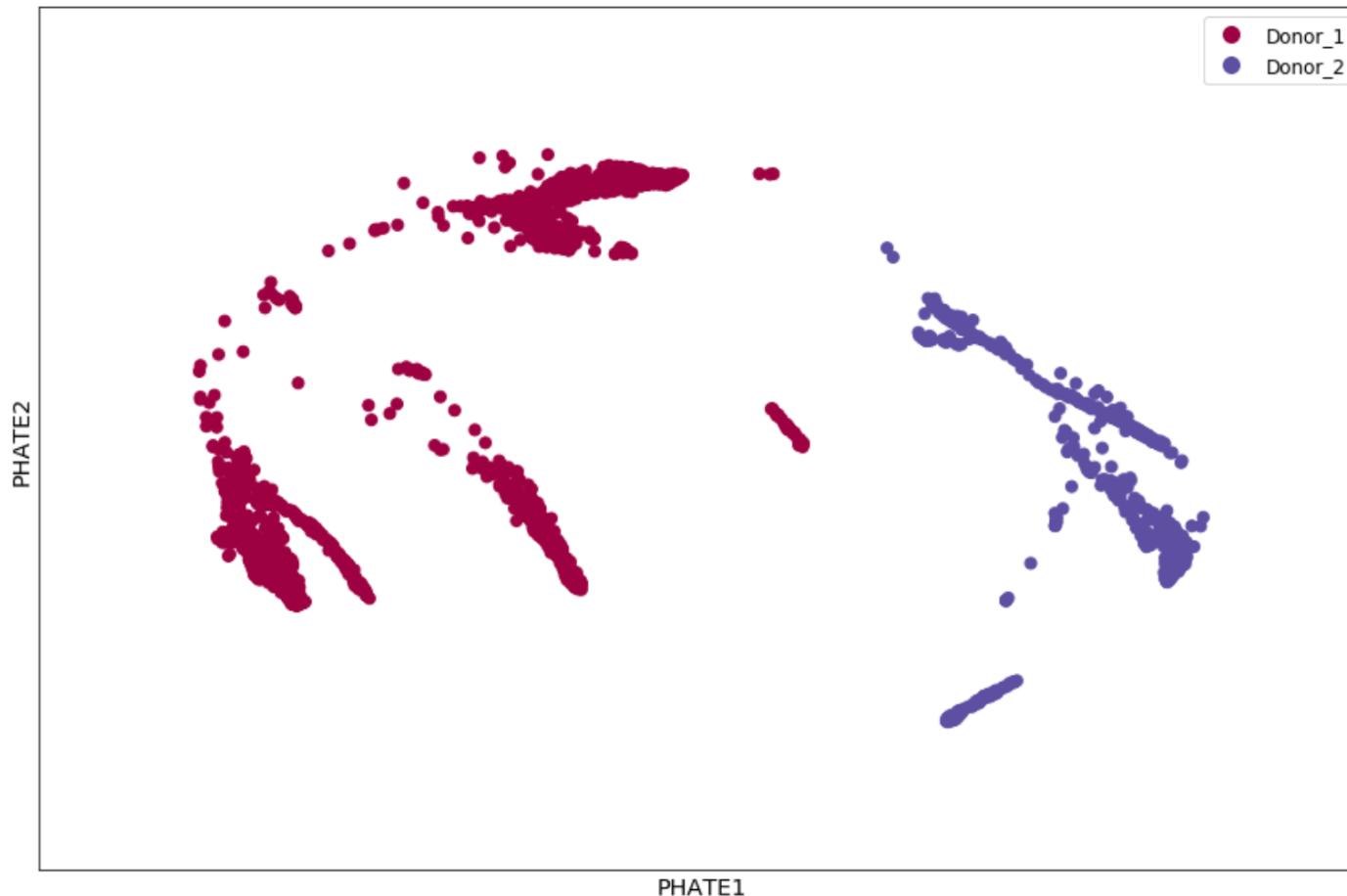
Problem with Batch Effect



Samples become hard to compare

All genes, cell types seem different!

How do we detect batch effect?

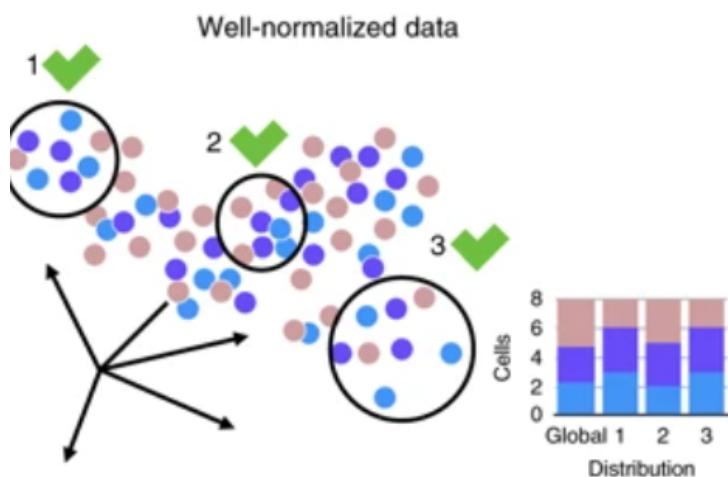


Visualization!

Diversity Measures

$$H' = - \sum_{i=1}^R p_i \ln p_i$$

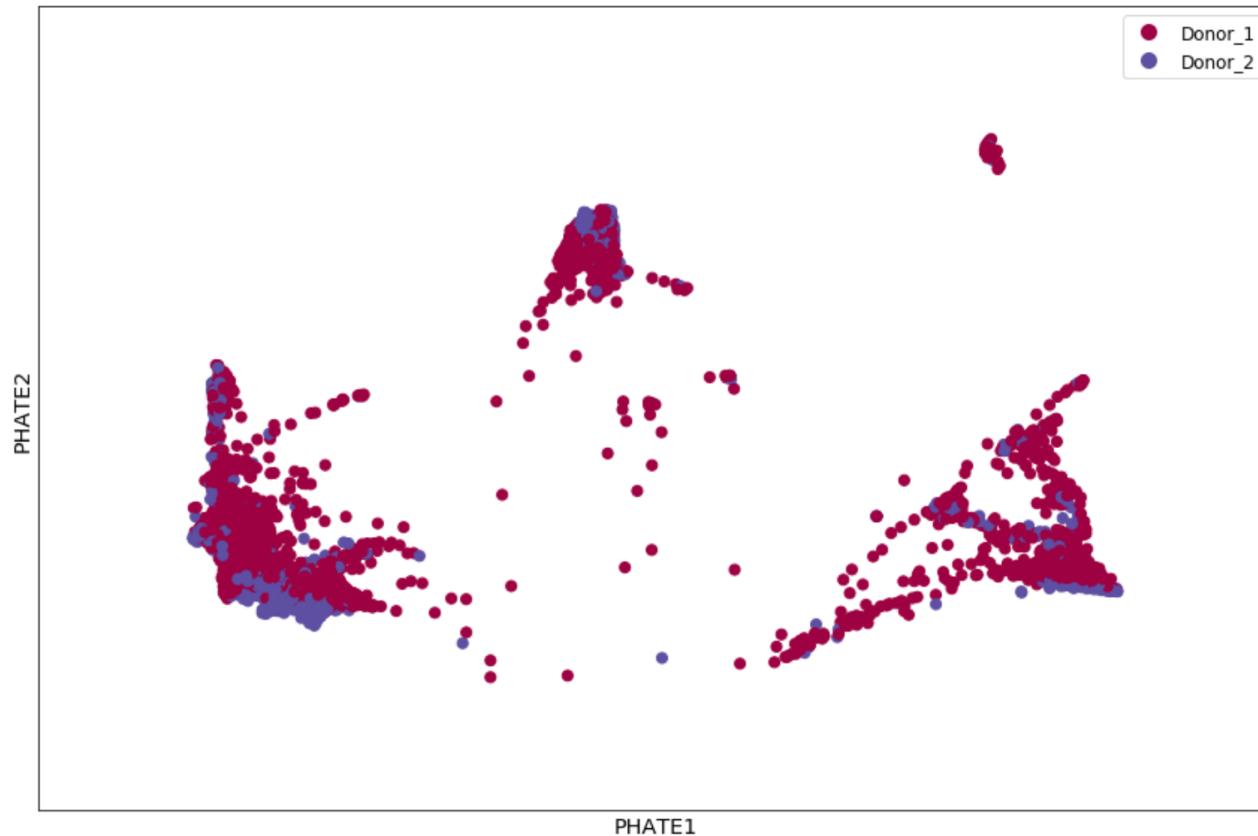
- **Diversity measures including Shannon entropy have been used to quantify the degree of “mixture” between samples**



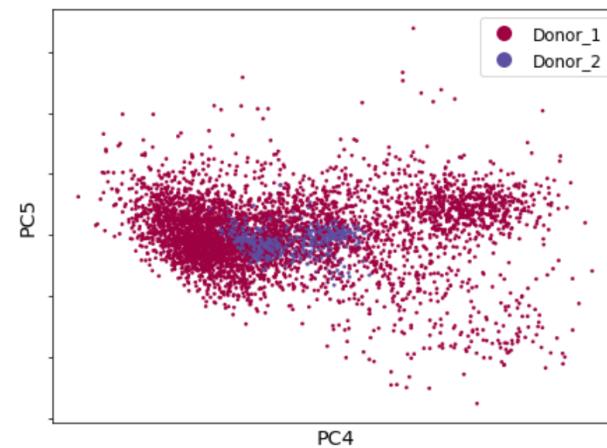
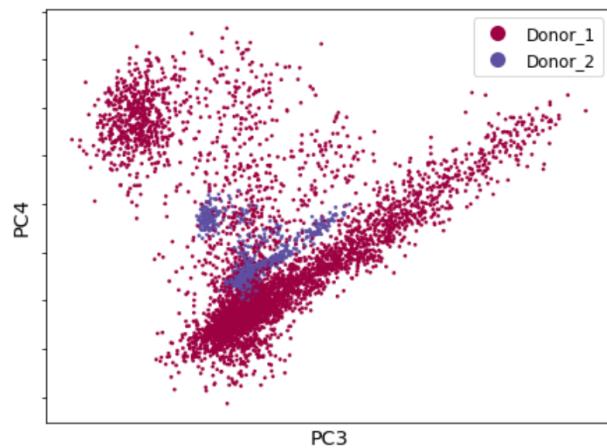
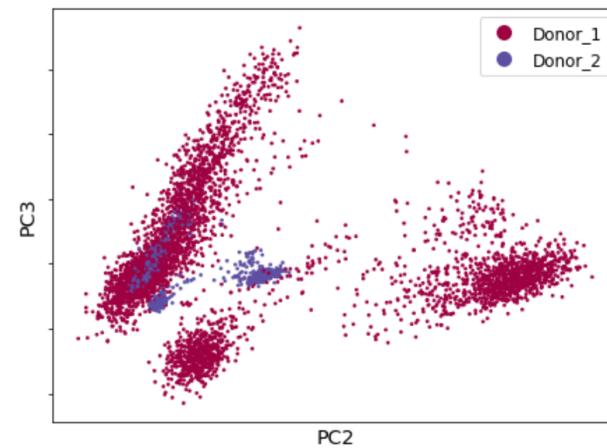
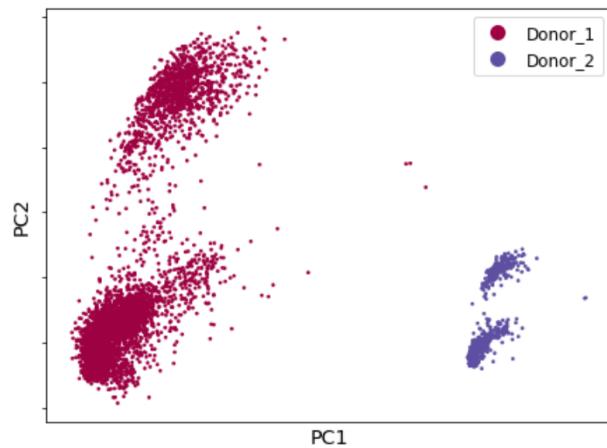
- kBet [Buttner et al] compare global population proportions to local
- A priori, we don't know what the degree of mixture should be, but if batch correction works, it should increase

Batch Correction

Take off artifactual variation and keep biological variation!



Can we use PCA to correct batch effect?



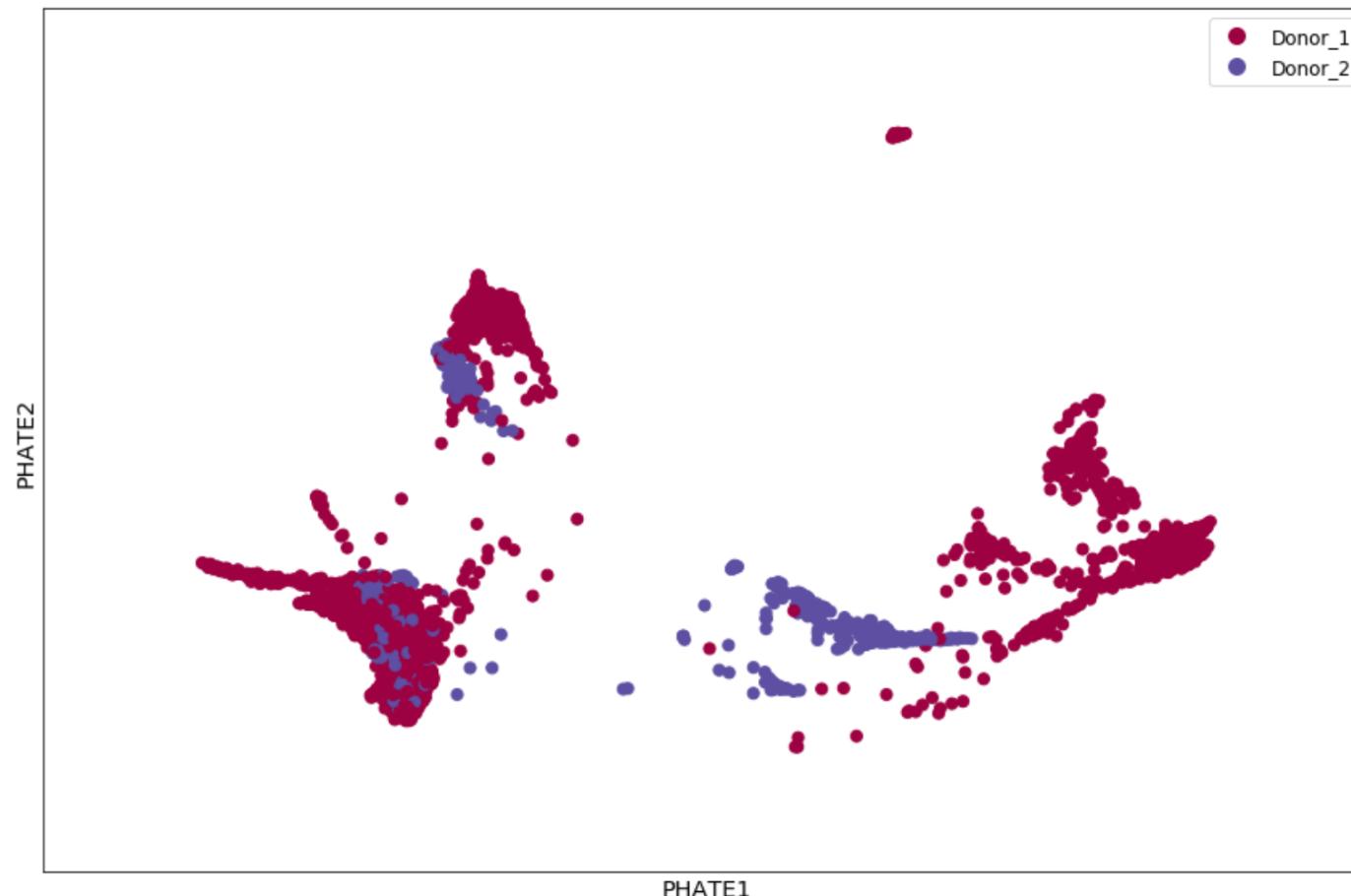
When poll is active, respond at **PollEv.com/yaleml**

Text **YALEML** to **22333** once to join

How could we use PCA or SVD to remove batch effect?

This noise is large scale

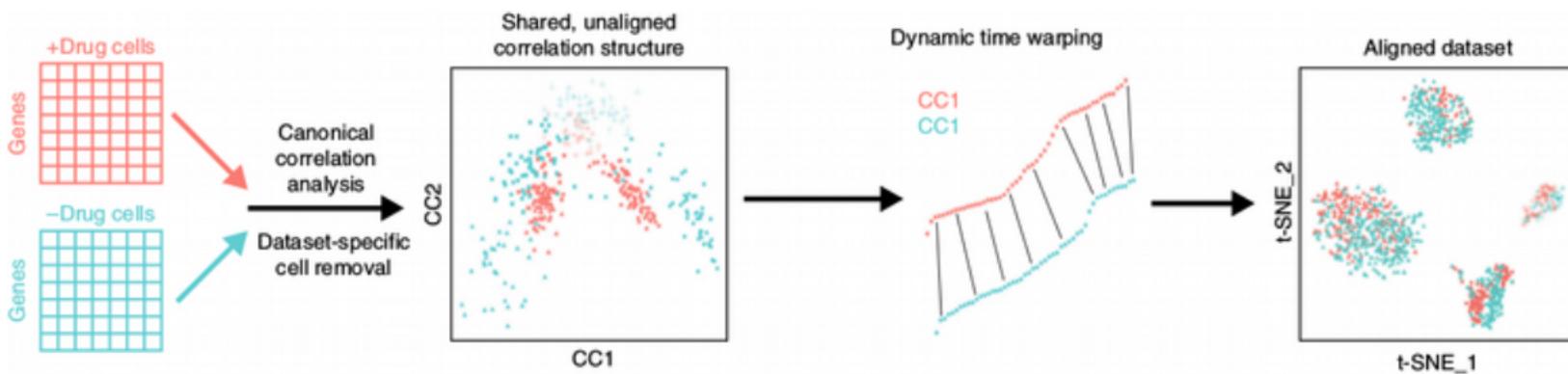
- Could be potentially addressed by removing first PC



Problem with PCA

- It can address a linear shift in the data
- But cannot handle more complex non-linear effects

Canonical Correlation Analysis



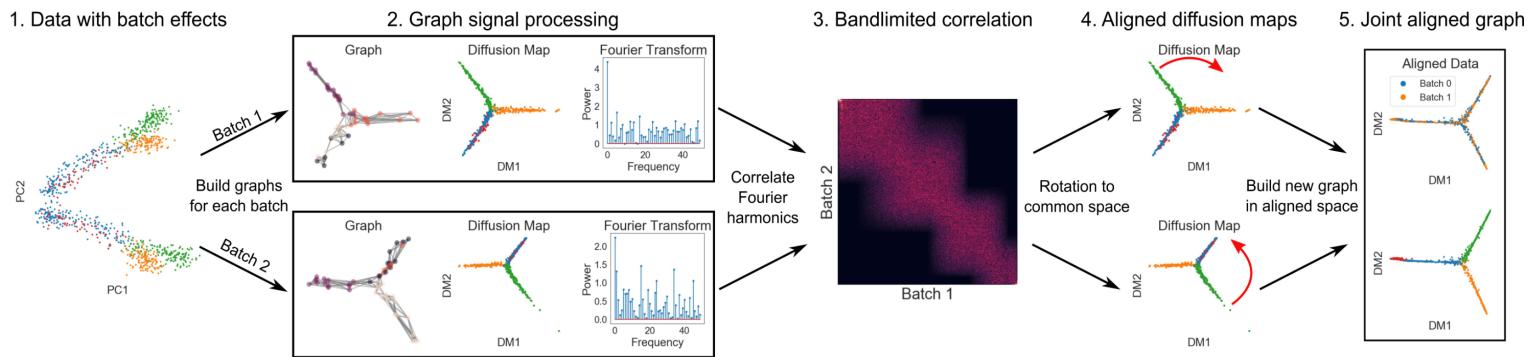
CCA : uses a variant of PCA that brings two sets of samples to a common space

Can cluster or perform other analysis in common space
but cannot go back to original space

Finds linear axes of common variation

Thrown off by non-matching populations

Harmonic Alignment

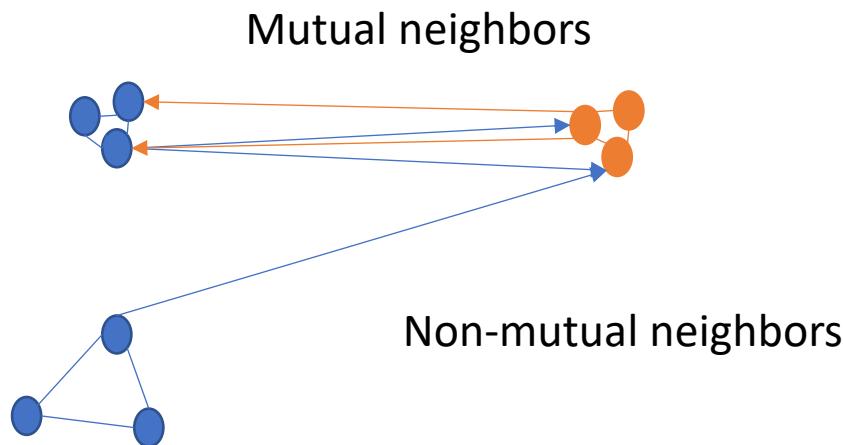


Aligns the diffusion components of two datasets using a rigid rotation that maximizes correlation of gene loadings onto eigenvectors

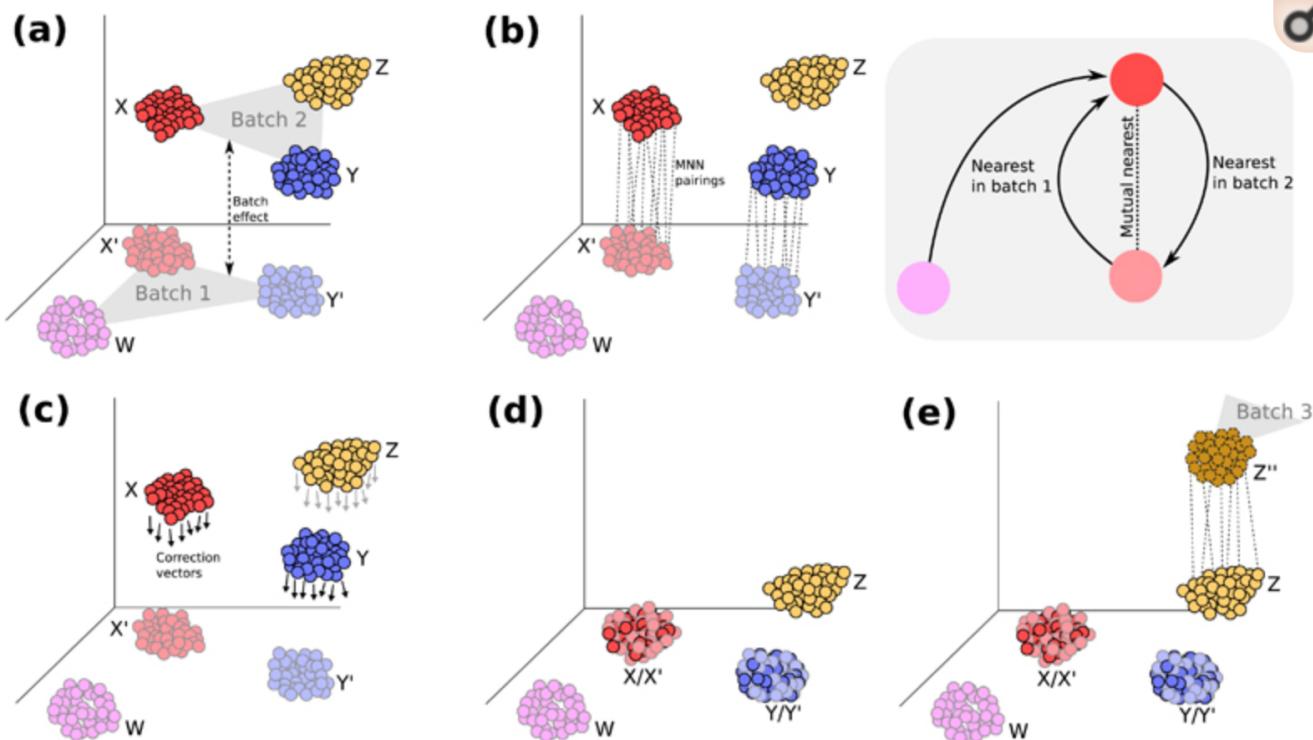
Simple rotation does not seem to handle all cases

Mutual Nearest Neighbors

- Creates a graph between two datasets
- Nearest neighbors in the other dataset could be “matching cells”
- But they have to be mutual!

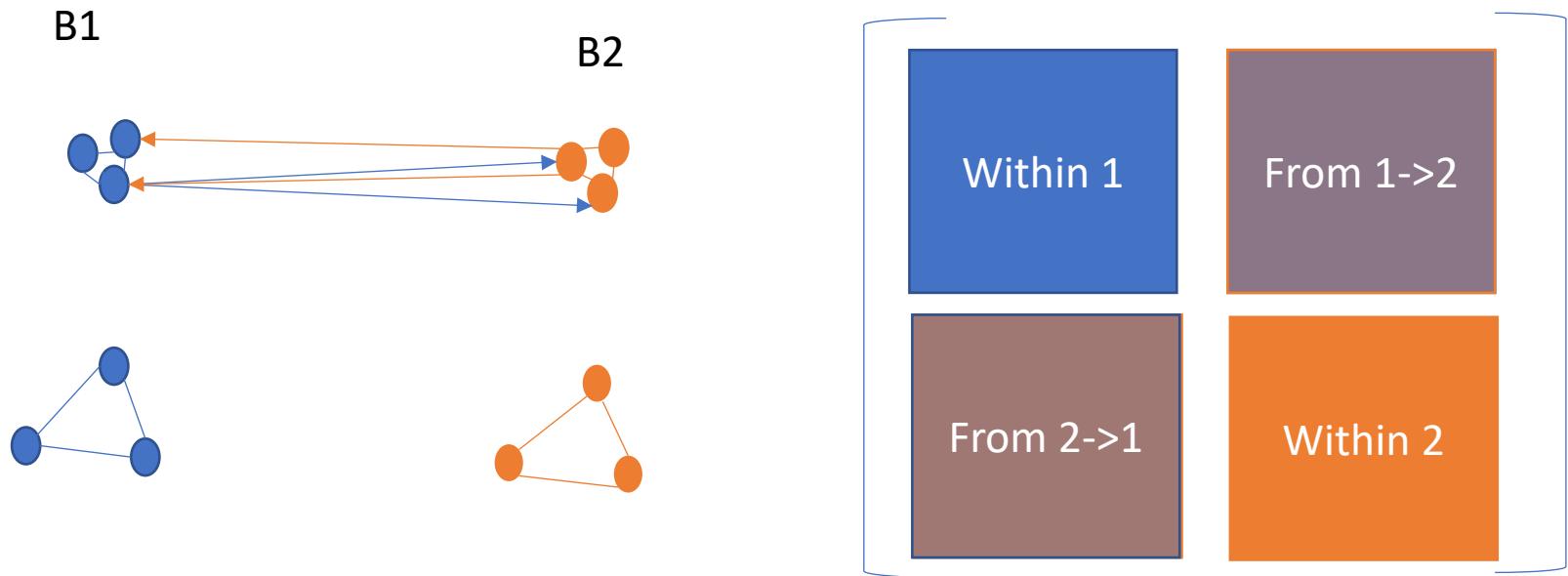


Mutual Nearest Neighbors



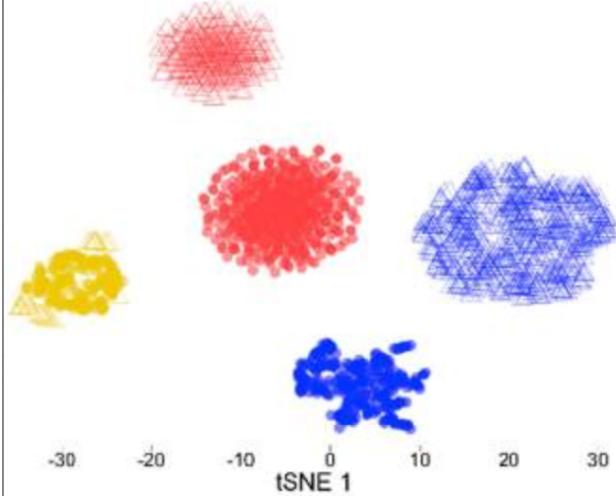
Use MNNs to find out the direction of the batch effect and correct it.

MNN with MAGIC

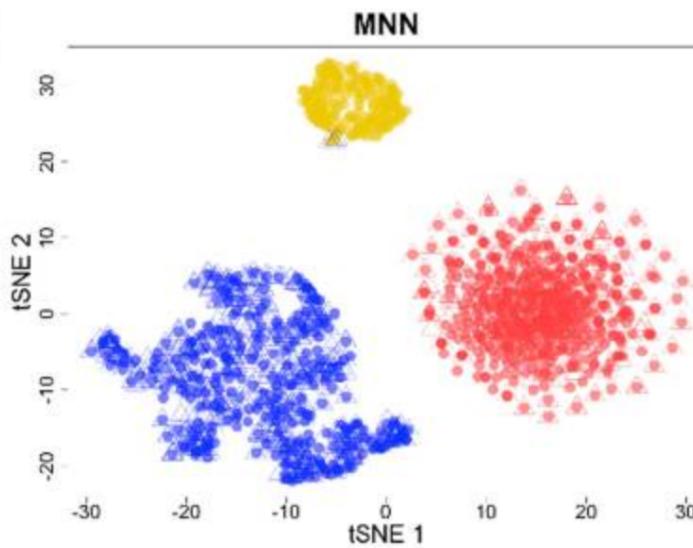


Uses softer affinity matrix and pulls the data in to correct the result

Uncorrected

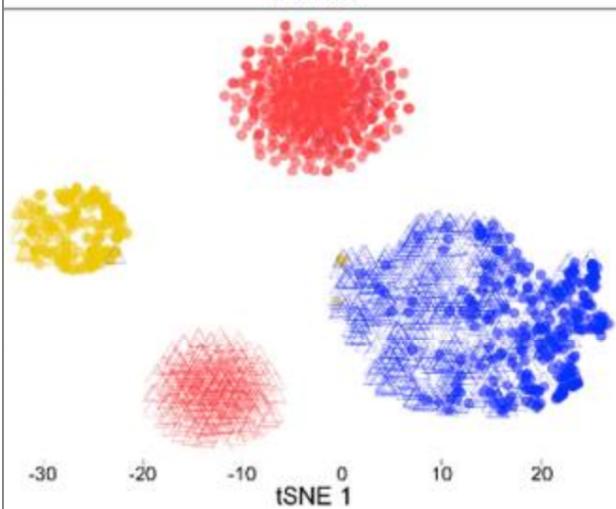


(b)

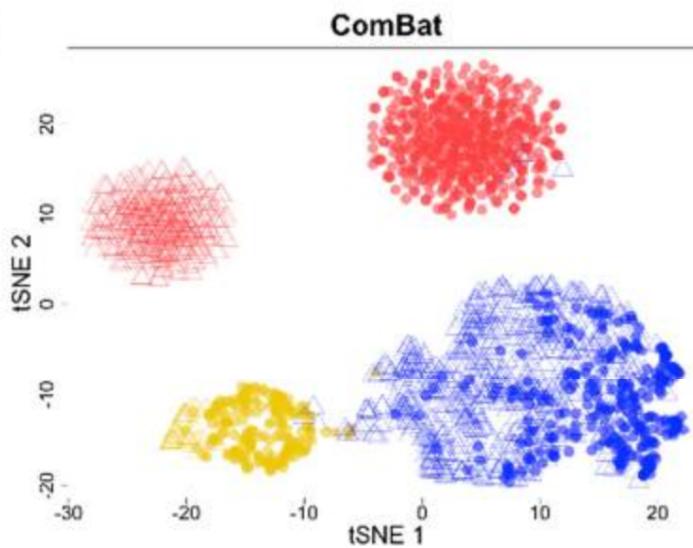


- Cell type 1
- Cell type 2
- Cell type 3
- Batch 1
- △ Batch 2

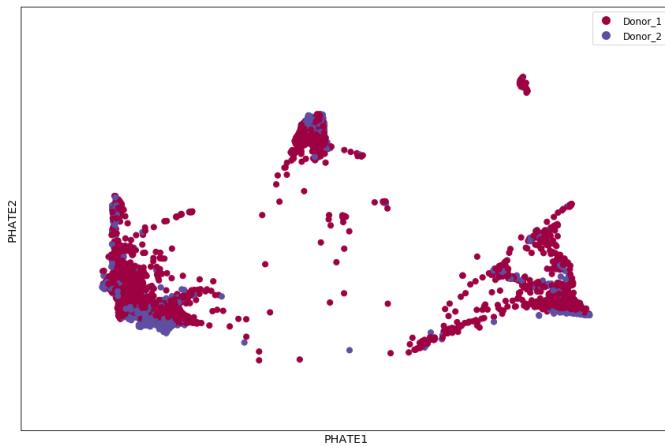
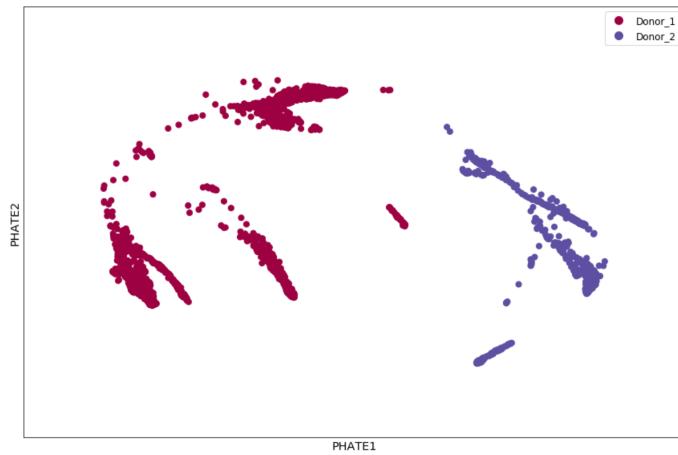
limma



(d)



MNN with MAGIC Correction



Is mutual nearest neighbor (MNN) normalization linear or non-linear?

Linear

Non-linear

Summary of data denoising

- Batch effects are sample-specific changes in measurements
- The goal of batch-normalization is to align cells of the same “type” across samples
- Mutual nearest neighbors (MNN) normalized batches by matching cells that are both close to each other

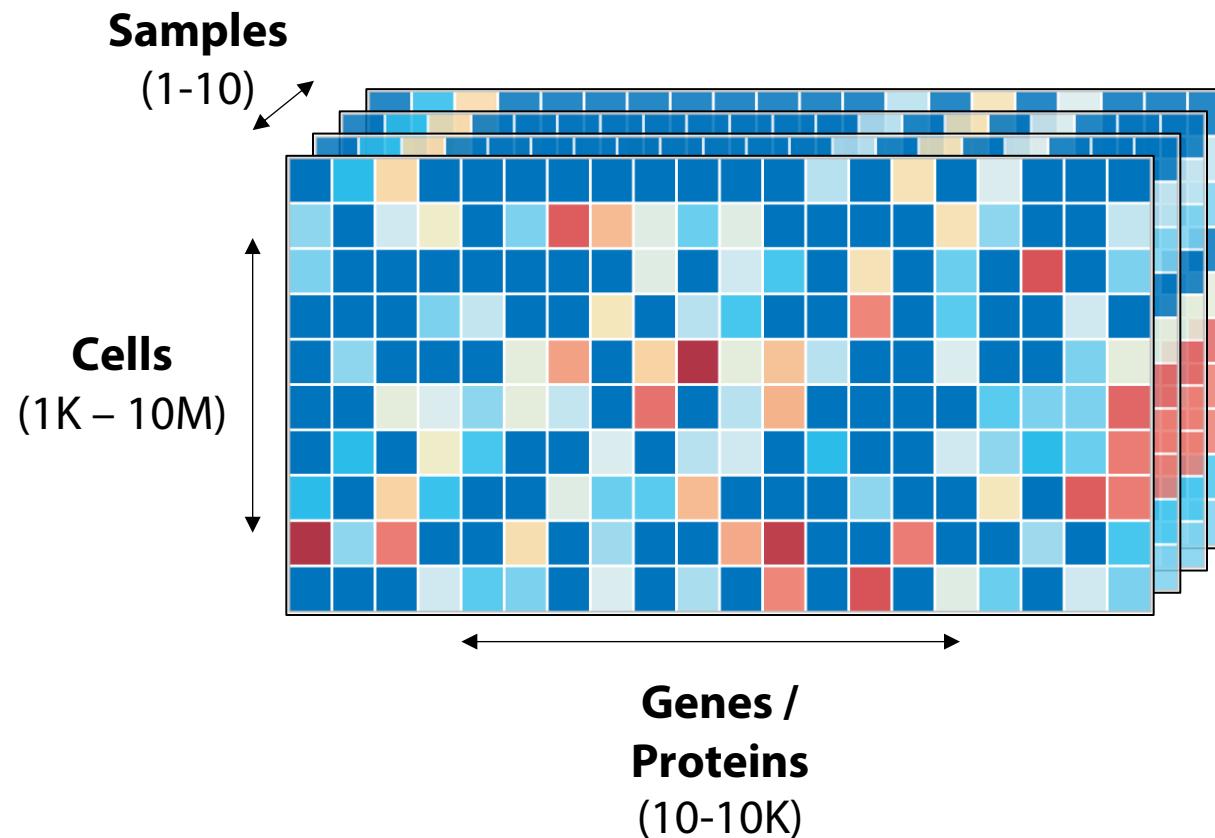
What questions do you have?

Please submit on Slack

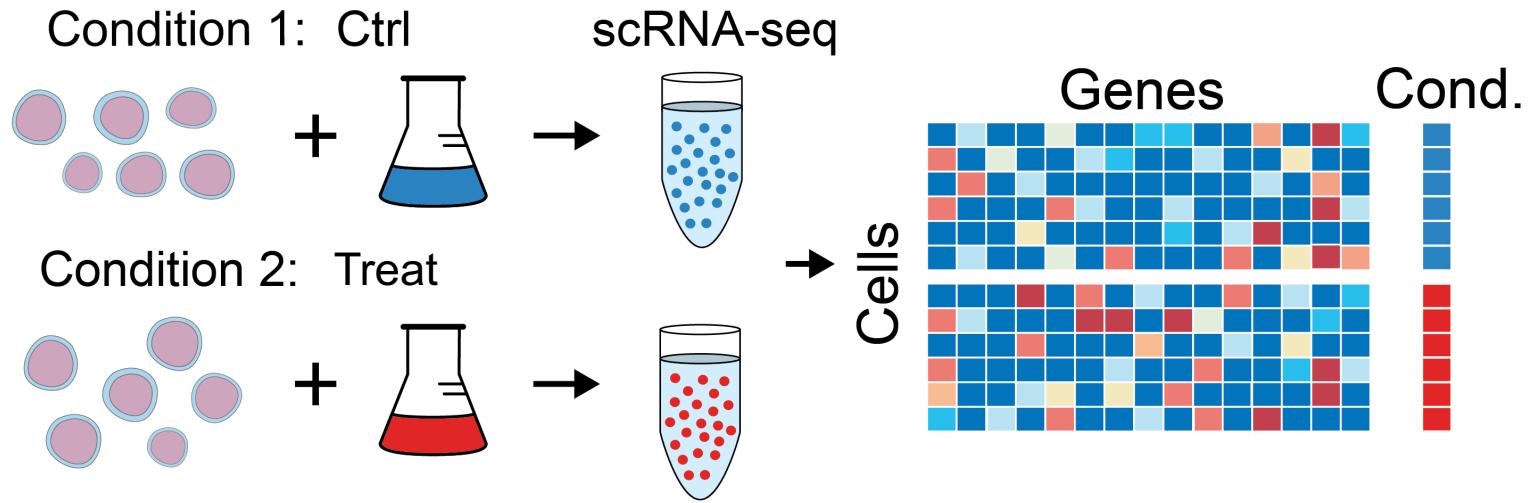
Compositional Analysis

http://bit.ly/MELD_lecture

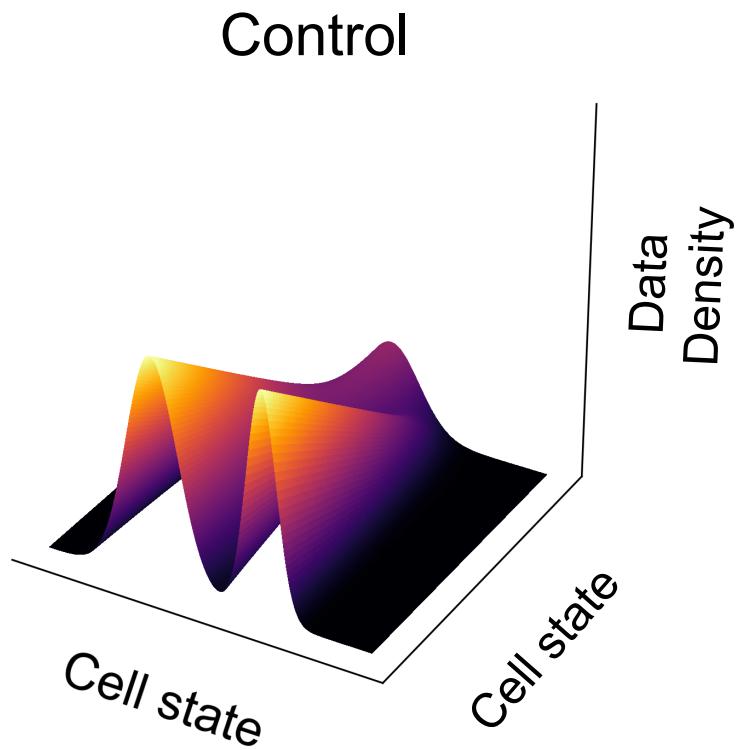
Single Cell Data



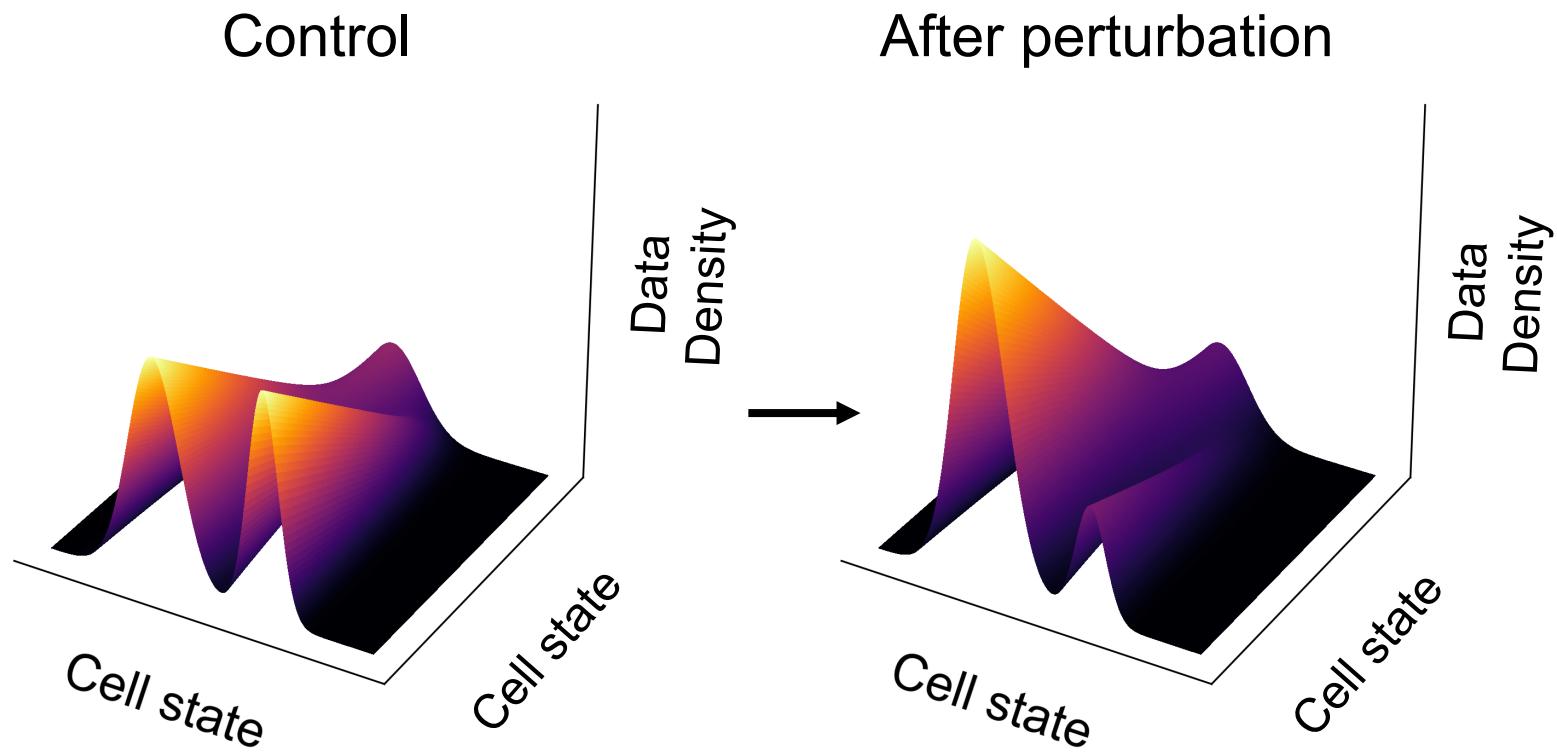
Common experimental setup



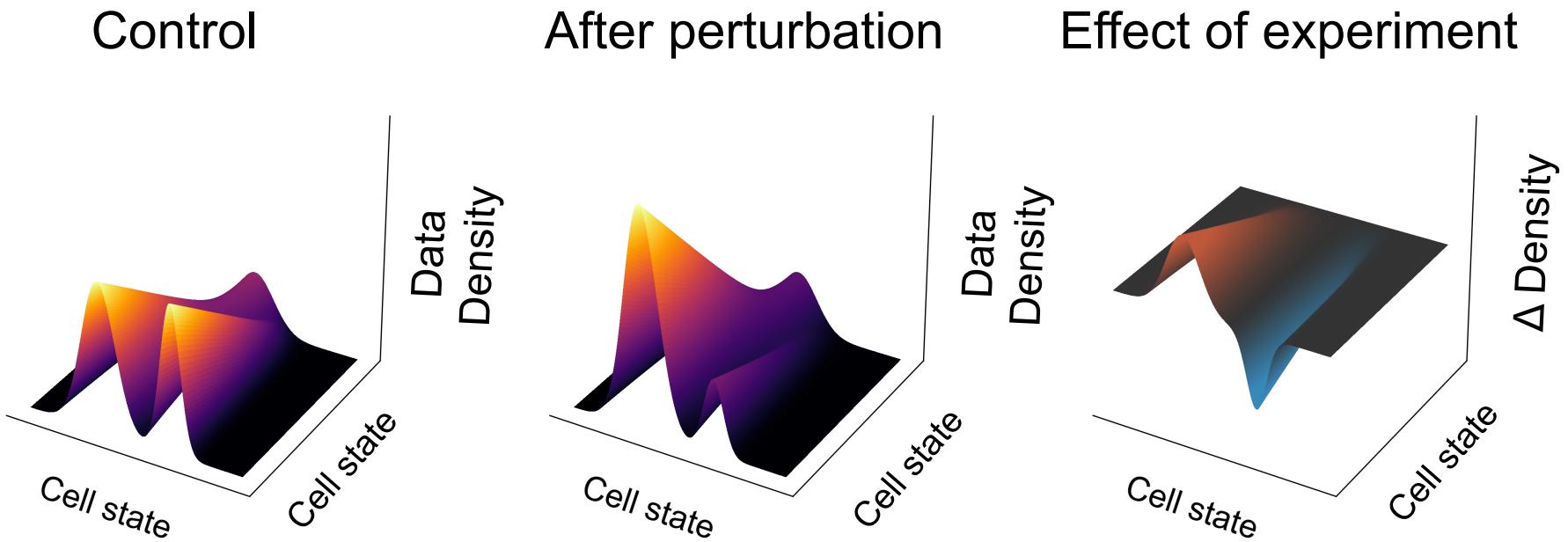
Goal: Quantify difference in data density



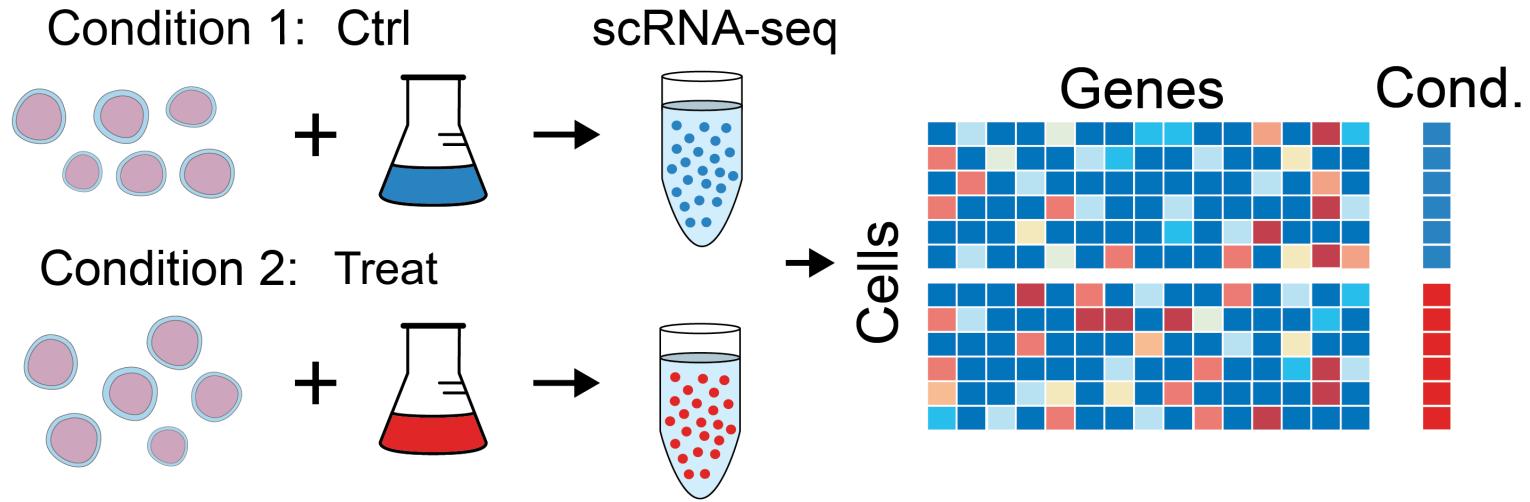
Goal: Quantify difference in data density



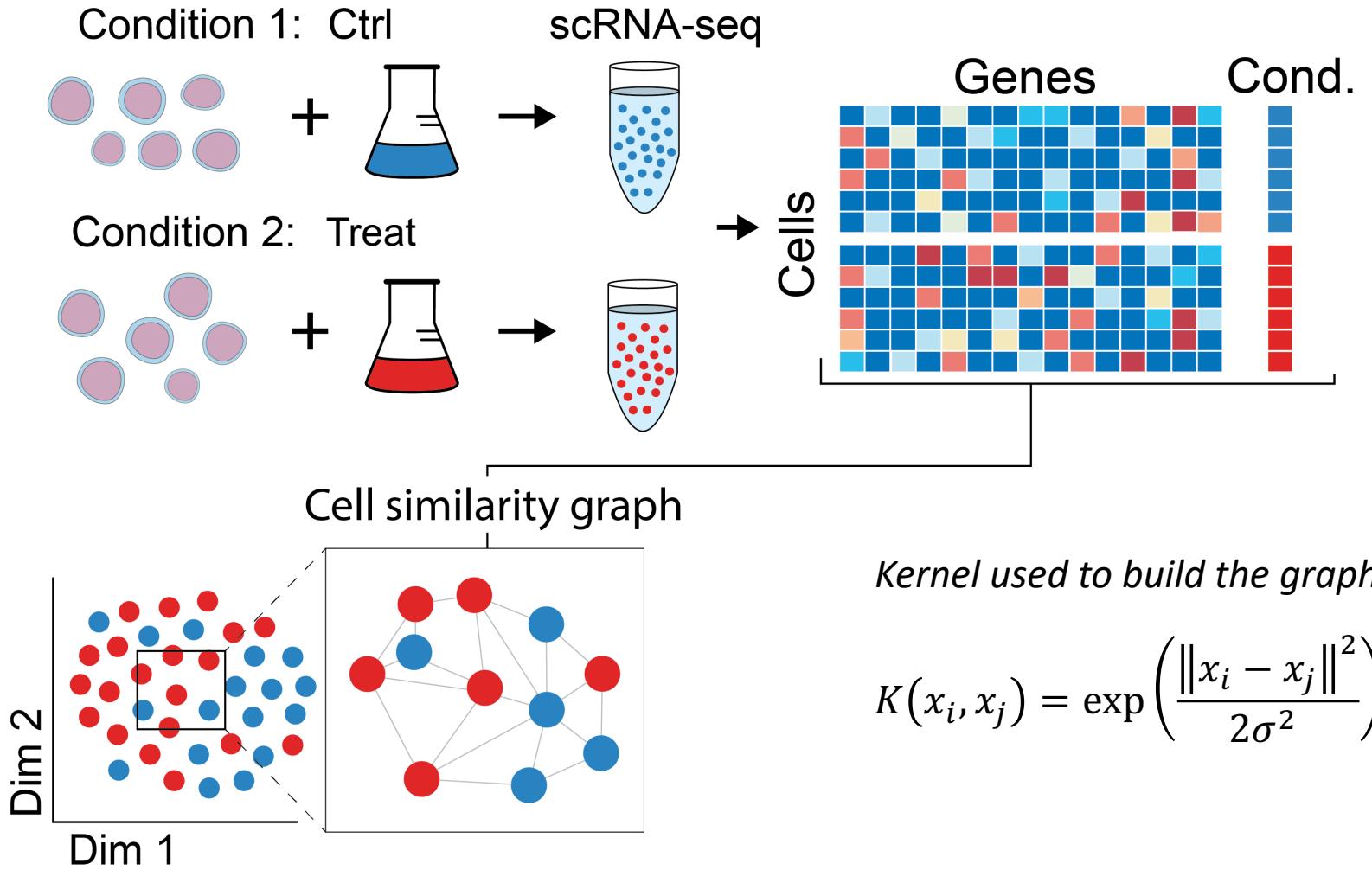
Goal: Quantify difference in data density



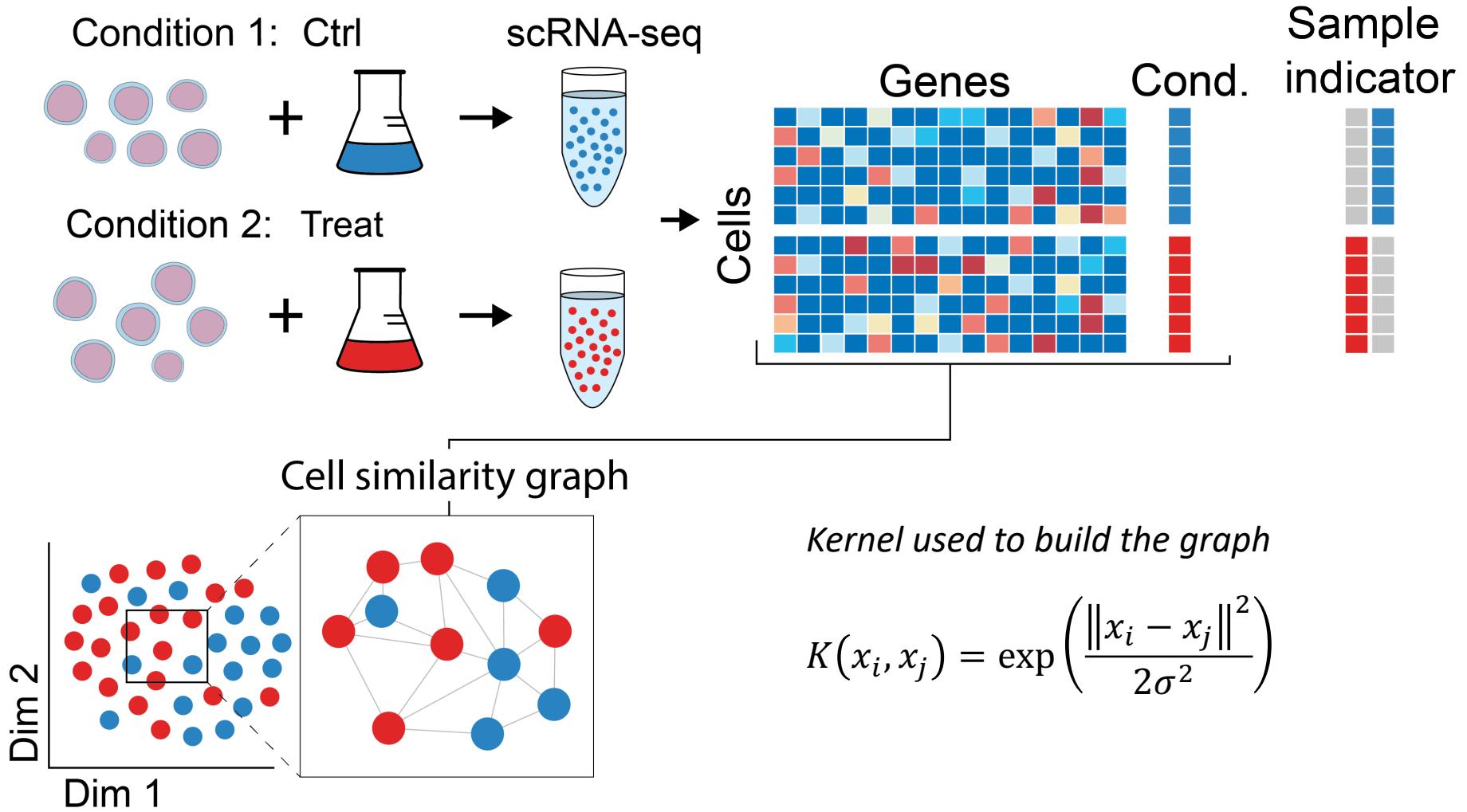
Estimating sample density over a graph



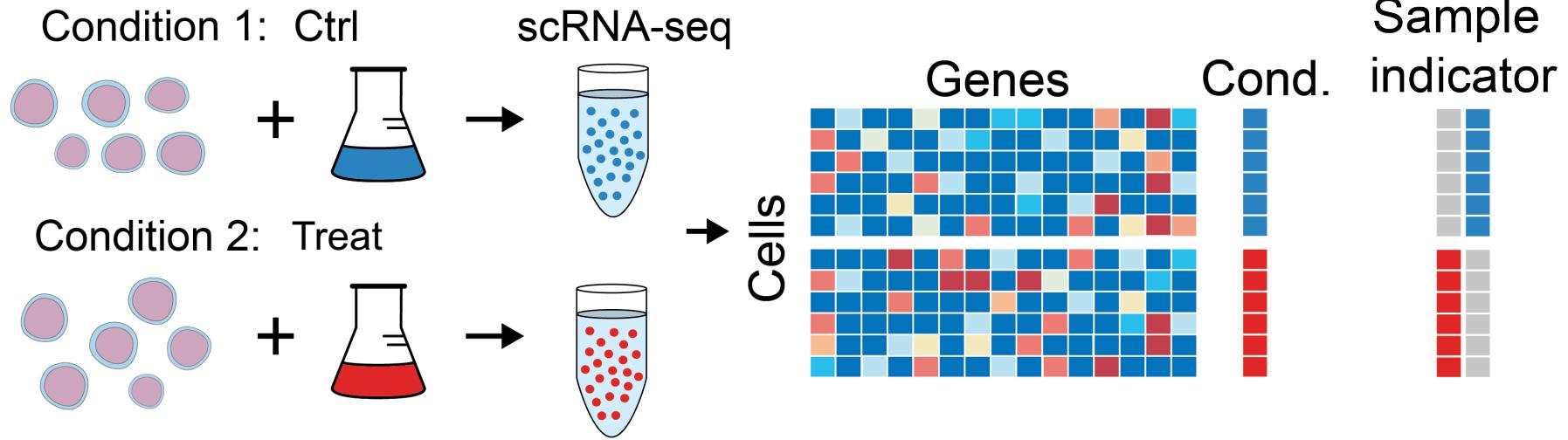
Estimating sample density over a graph



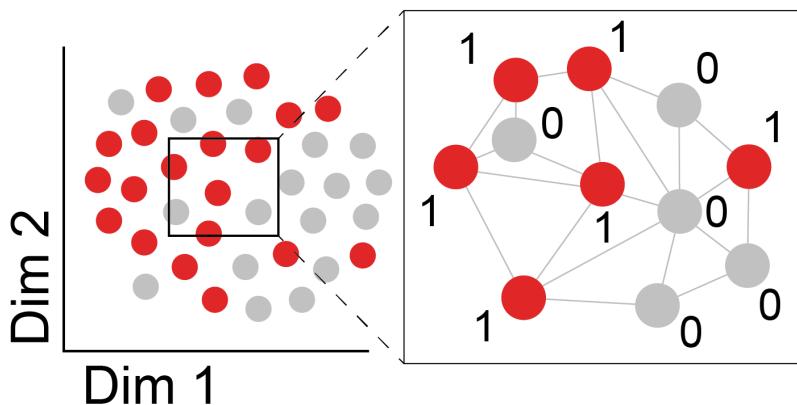
Estimating sample density over a graph



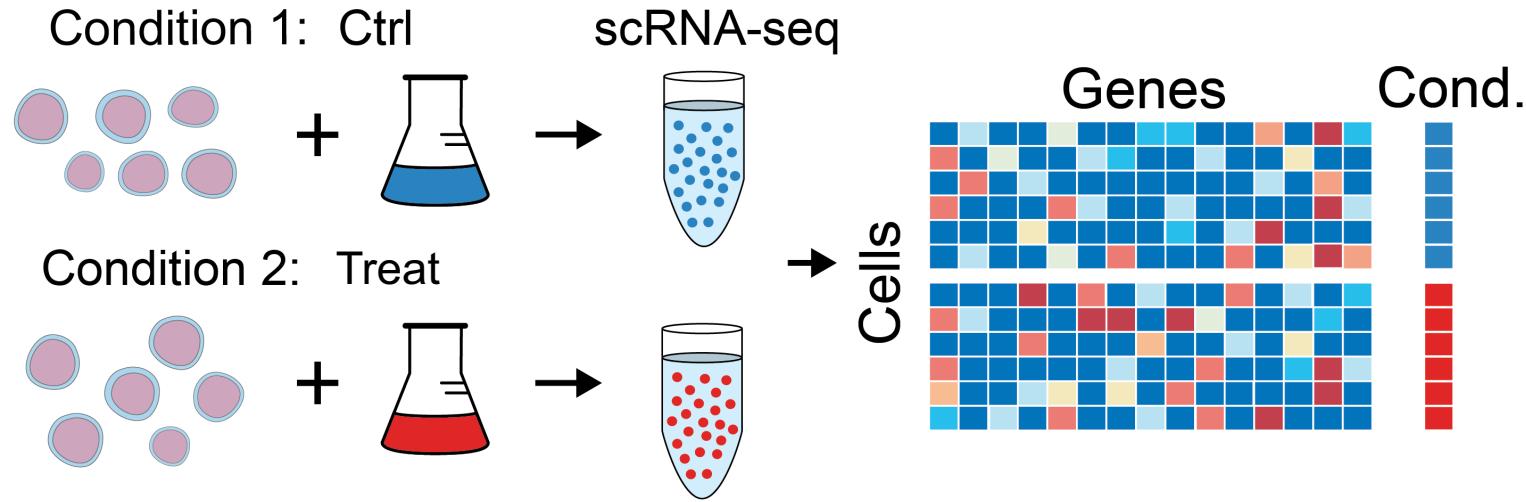
Estimating sample density over a graph



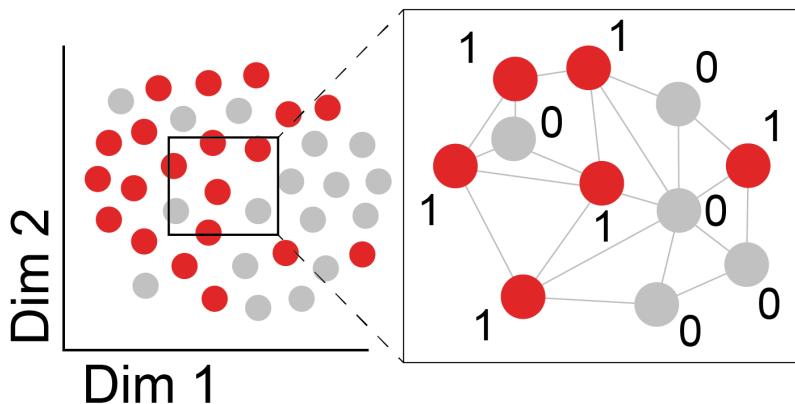
Treatment-associated indicator signal



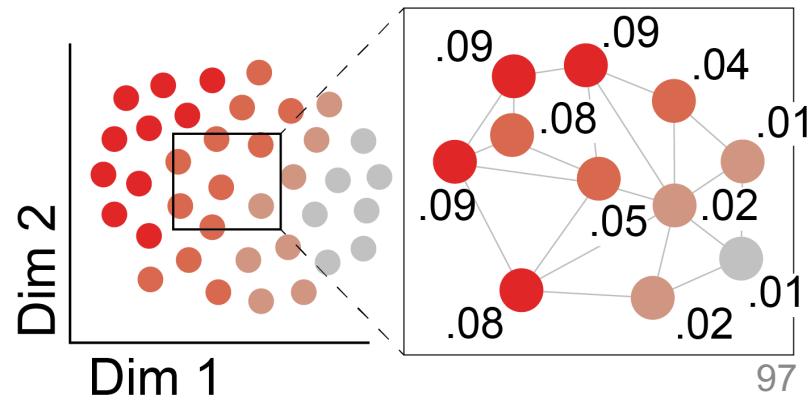
Estimating sample density over a graph

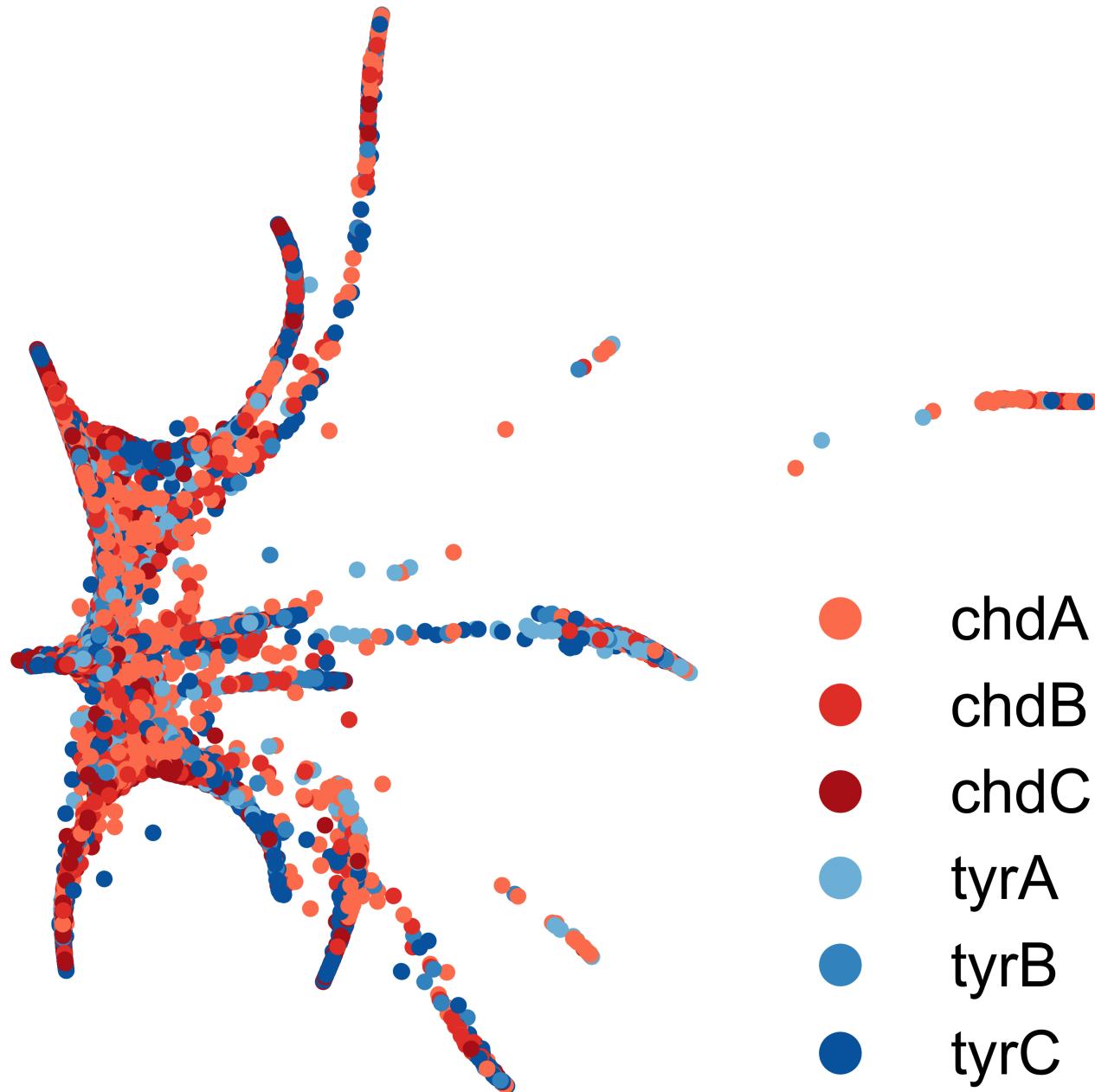


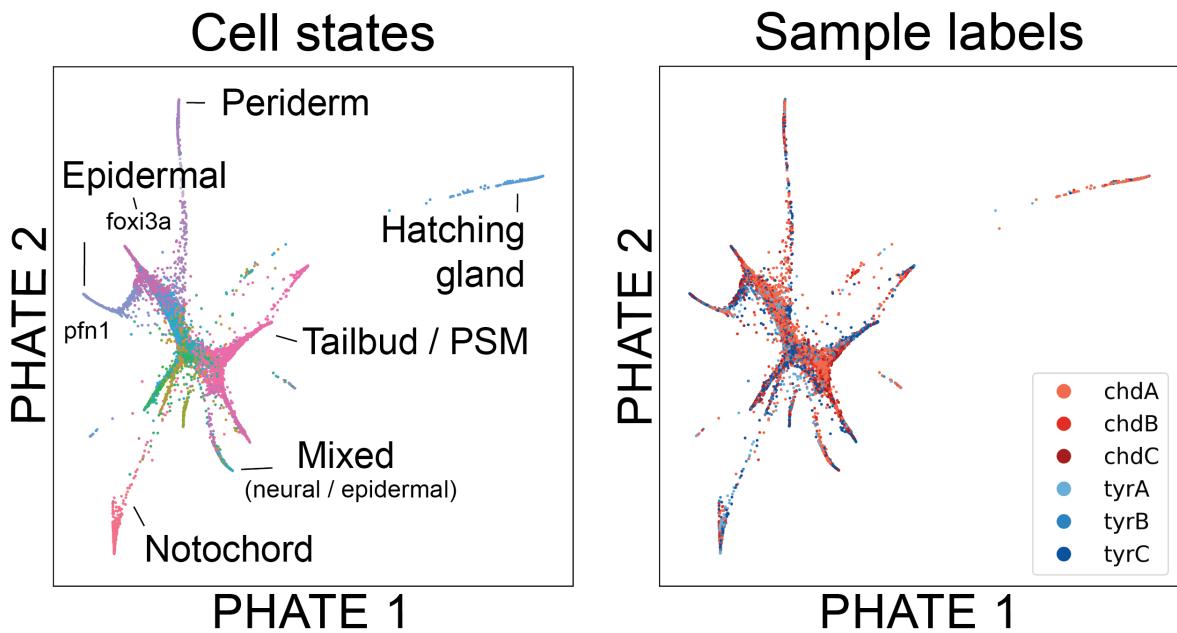
Treatment-associated indicator signal

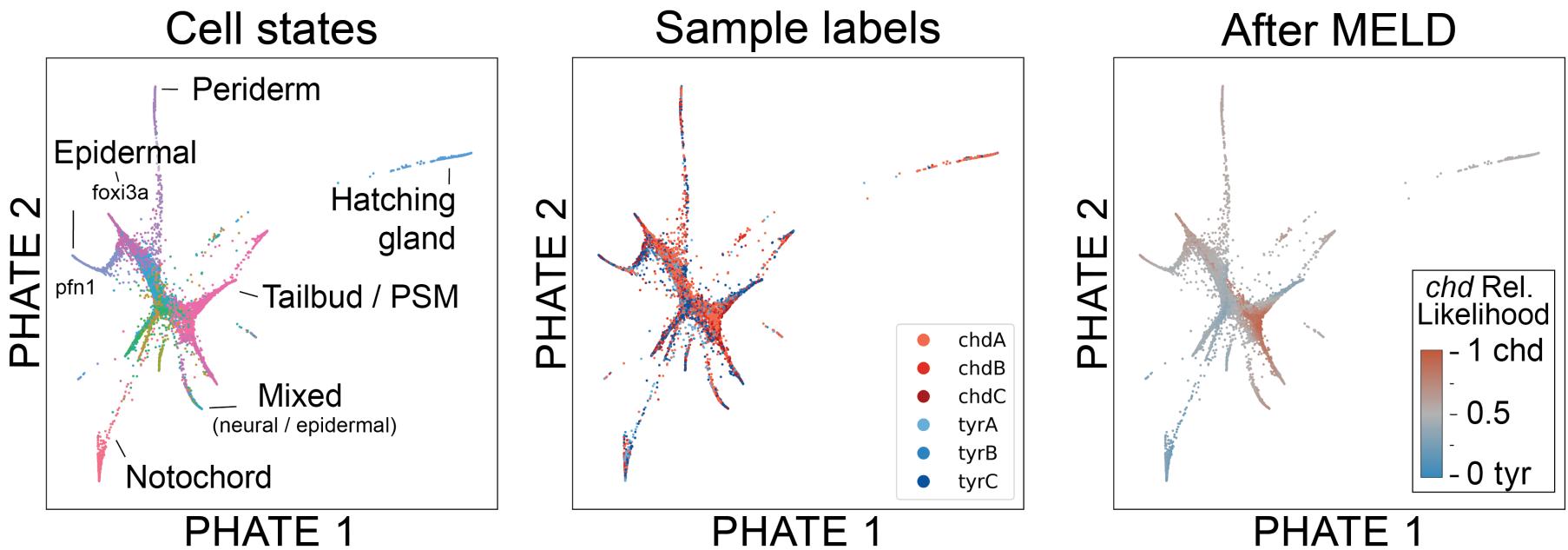


Treatment-associated sample density

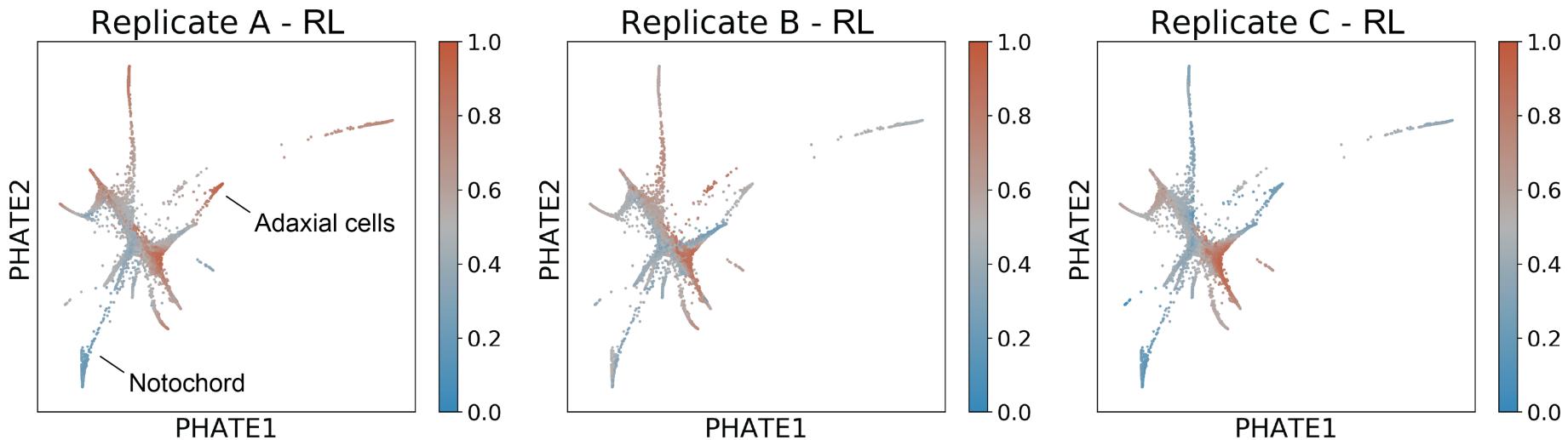




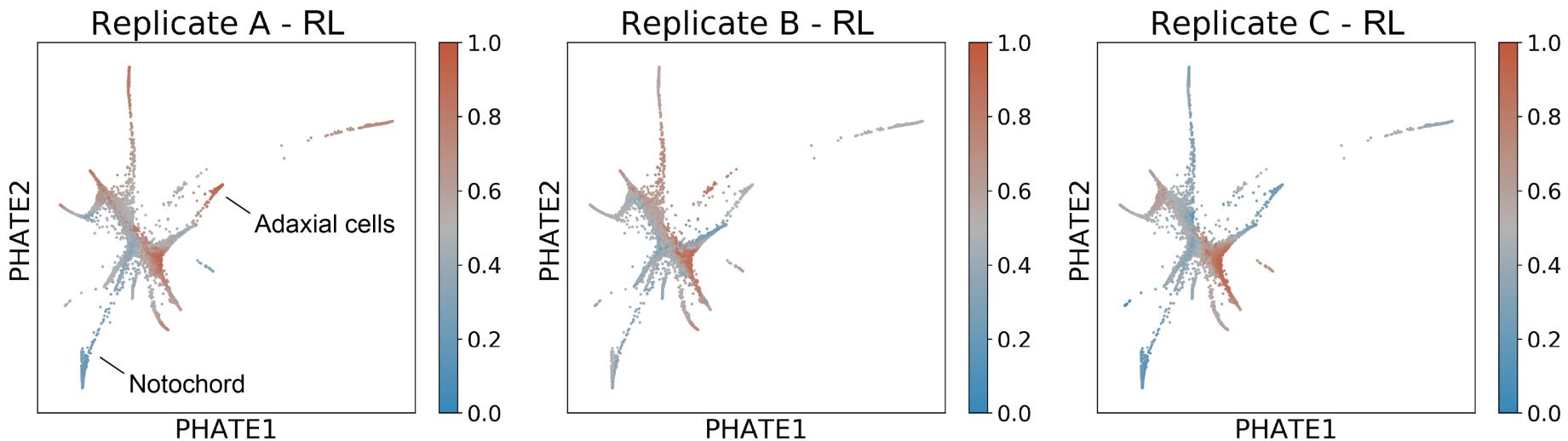




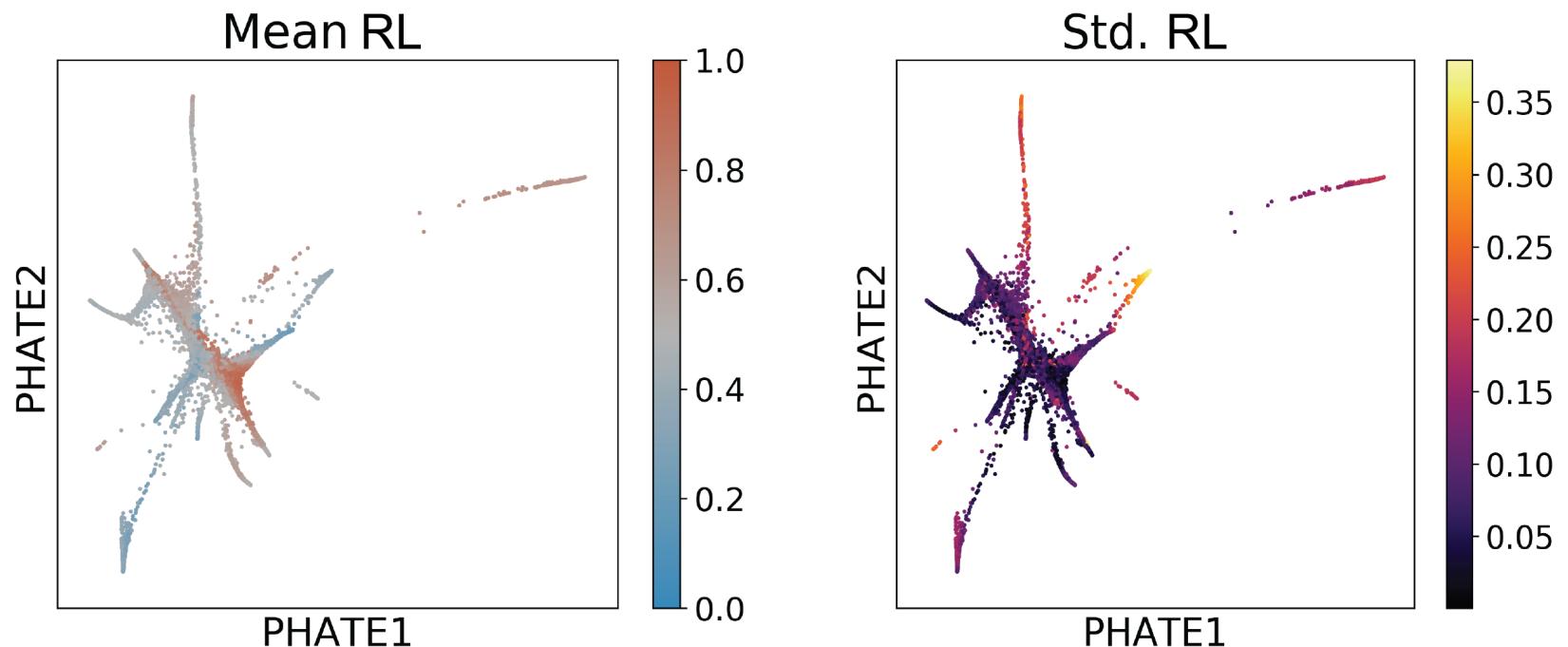
Analysis of replicates reveals heterogenous response

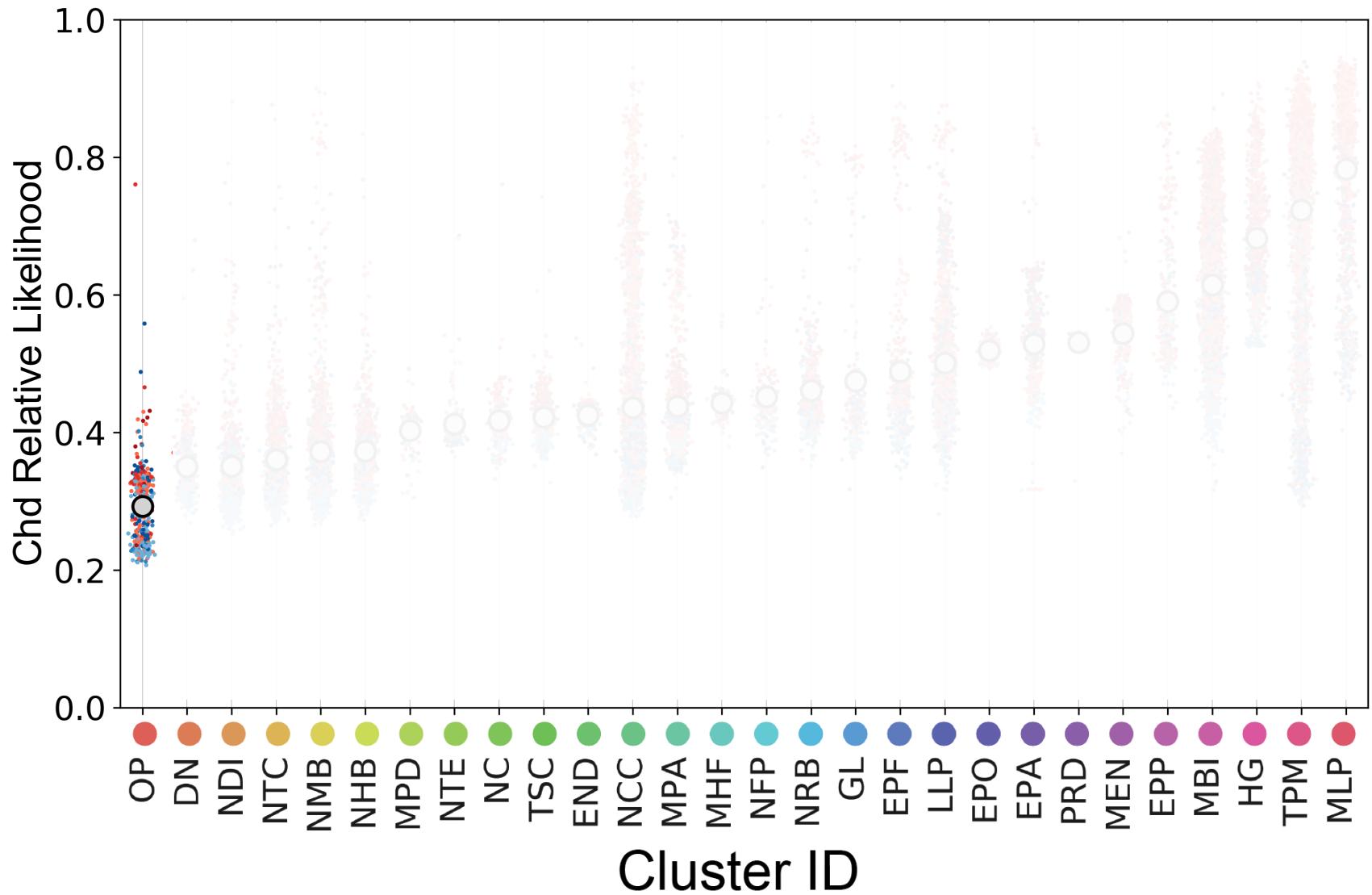


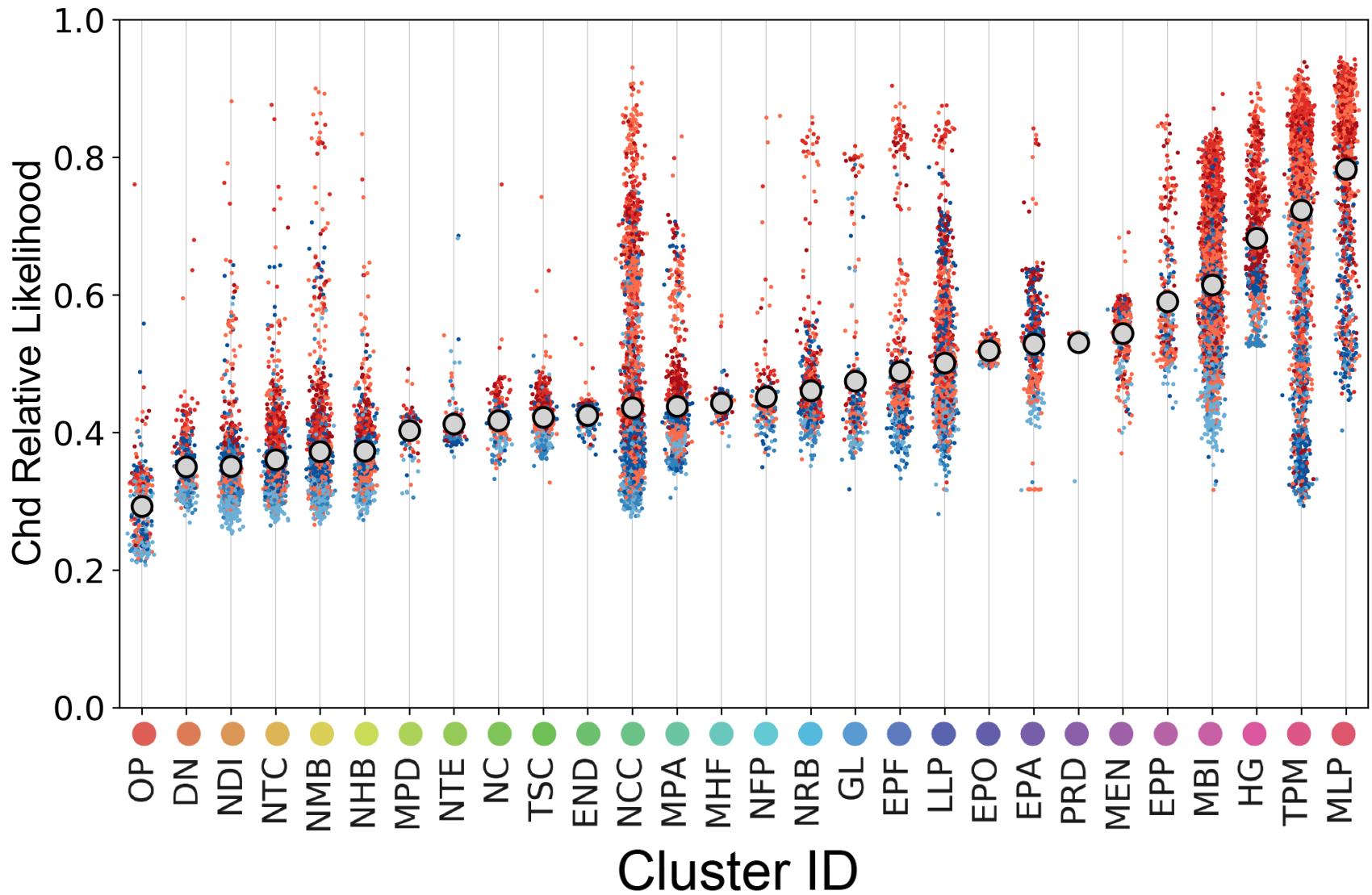
Analysis of replicates reveals heterogenous response

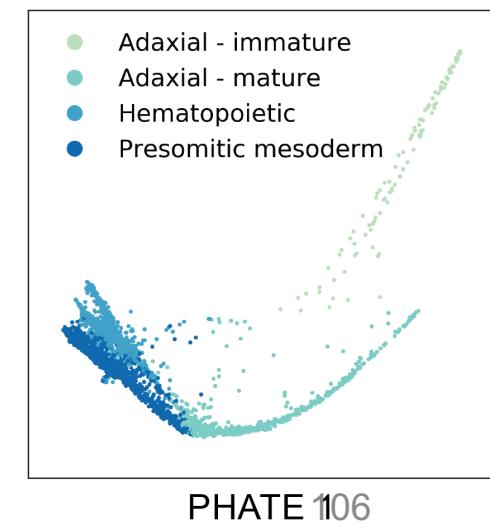
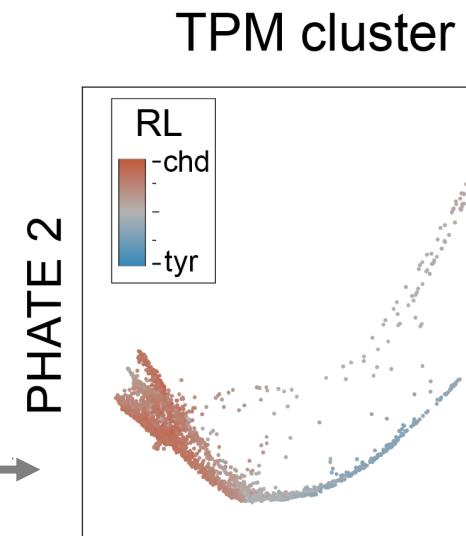
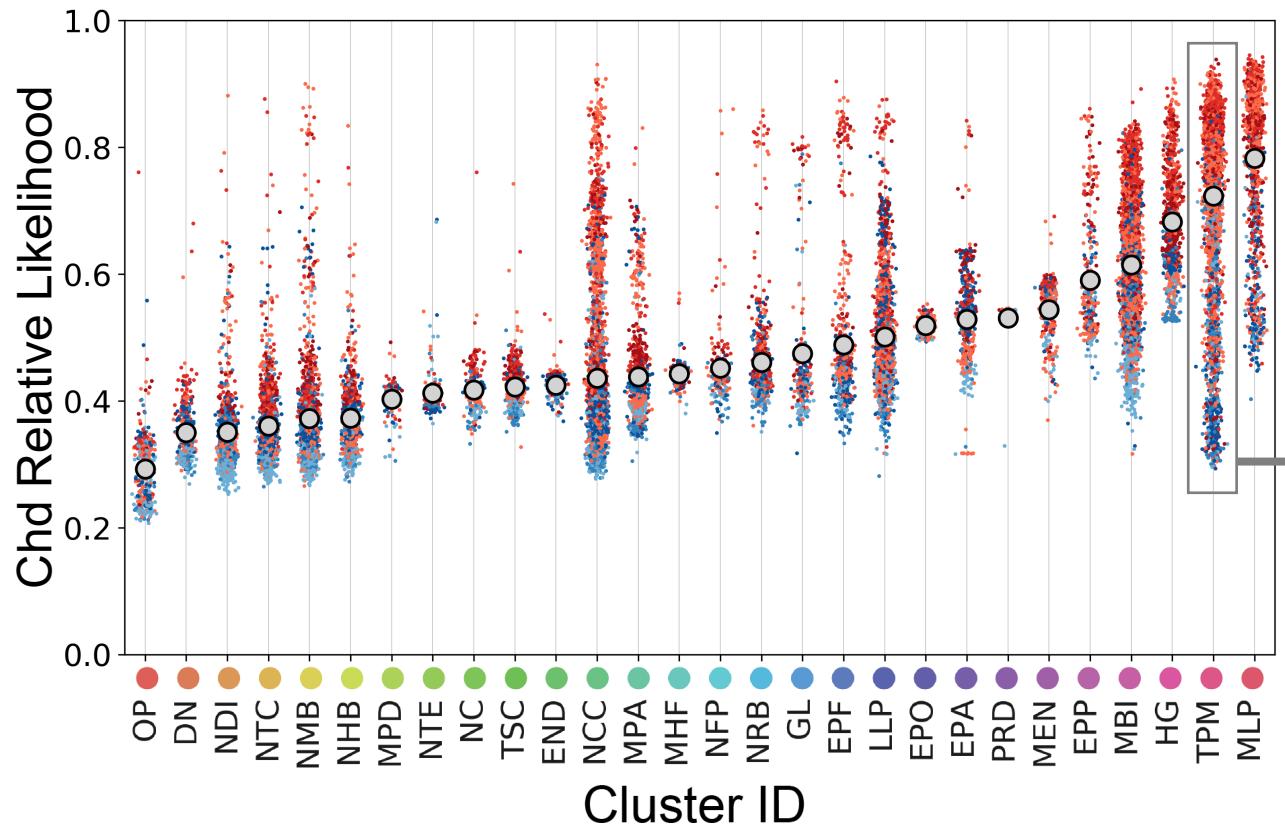


Analysis of replicates reveals heterogenous response









The screenshot shows a GitHub repository page for 'KrishnaswamyLab/MELD: Quantifying the effect of experimental perturbations at single-cell resolution'. The page includes sections for 'Quick Start' (with a link to a 'Guided tutorial in Python'), 'Introduction' (describing MELD as a Python package for quantifying experimental perturbations), and a manuscript link. A 'Languages' sidebar indicates 100% Python usage.

MELD

Quantifying the effect of experimental perturbations at single-cell resolution

pypi v1.0.0 Unit Tests passing coverage 100% docs passing DOI 10.1101/532846

Follow 2.9k Stars 30

Quick Start

- Guided tutorial in Python.

Introduction

MELD is a Python package for quantifying the effects of experimental perturbations. For an in depth explanation of the algorithm, read our manuscript on BioRxiv.

Quantifying the effect of experimental perturbations at single-cell resolution. Daniel B Burkhardt*, Jay S Stanley*, Alexander Tong, Ana Luisa Perdigoto, Scott A Gigante, Kevan C Herold, Guy Wolf, Antonio J Giraldez, David van Dijk, Smita Krishnaswamy. BioRxiv. doi:10.1101/532846.

The goal of MELD is to identify populations of cells that are most affected by an experimental perturbation. Rather than clustering the data first and calculating differential abundance of samples within clusters, MELD provides a density estimate for each scRNA-seq sample for every cell in each dataset.

bioRxiv bit.ly/quantsinglecell



github.com/KrishnaswamyLab/MELD



Coming soon!