

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

What is your favorite model organism?

Machine Learning for Single Cell Analysis

Course introduction

Search Krishnaswamy Lab Help

#2021-workshop-announcements ☆
Announcements from the organizers of the 2021 ML Workshop

Threads All DMs Mentions & reactions Saved items More Channels # 2019-workshop # 2020-workshop-main # 2021-workshop-ann... # 2021-workshop-byo... # 2021-workshop-codi... # 2021-workshop-gro... # 2021-workshop-mat... # 2021-workshop-tas # general # magic # meld # phate # random # scprep # workshop-main-wint

2021-workshop-announcements

@Scott Gigante created this channel yesterday. This is the very beginning of the #2021-workshop-announcements channel. Description: Announcements from the organizers of the 2021 ML Workshop (edit)

Add an app Add people Share channel Send emails to channel

Yesterday

Scott Gigante 2:29 PM joined #2021-workshop-announcements along with Daniel Burkhardt.

Scott Gigante 2:32 PM set the channel topic: Announcements from the organizers of the 2021 ML Workshop

Scott Gigante 2:35 PM set the channel description: Announcements from the organizers of the 2021 ML Workshop

clarice 3:05 PM was added to #2021-workshop-announcements by Scott Gigante, along with 36 others.

Message #2021-workshop-announcements

<https://krishnaswamylab.org/get-help>



Smita Krishnaswamy



Daniel Burkhardt



Scott Gigante



Kofi Ansong



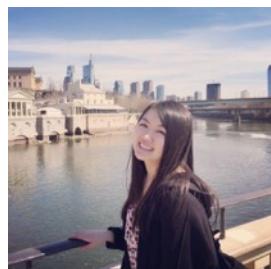
Andrew Benz



Egbert Castro



Pranik Chainani



Joanna Chen



Annie Gao



Michał Gerasimiuk



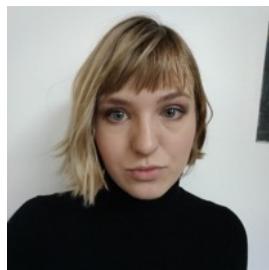
Wes Lewis



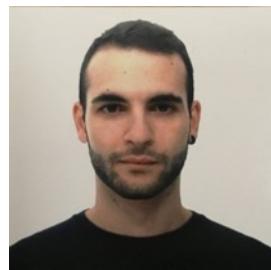
Francesc Lopez



Sameet Mehta



Sasha Safanova



Giacomo Scanavini



Alexander Tong



Aarthi Venkat

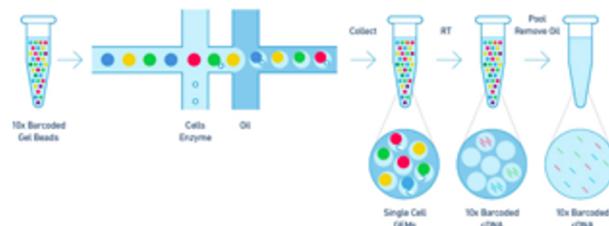


Max Yuan

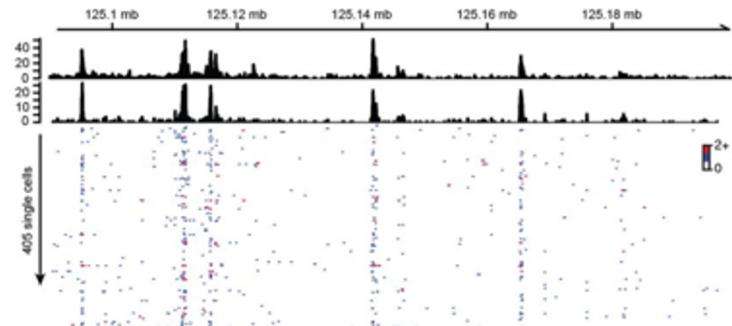


Jeffrey Zhou

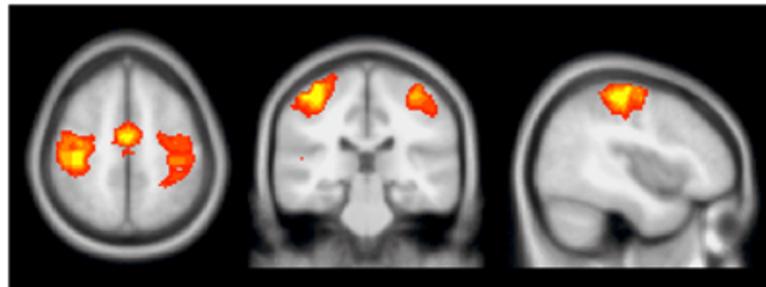
Big biomedical data



ScRNA-seq



ScATAC-seq



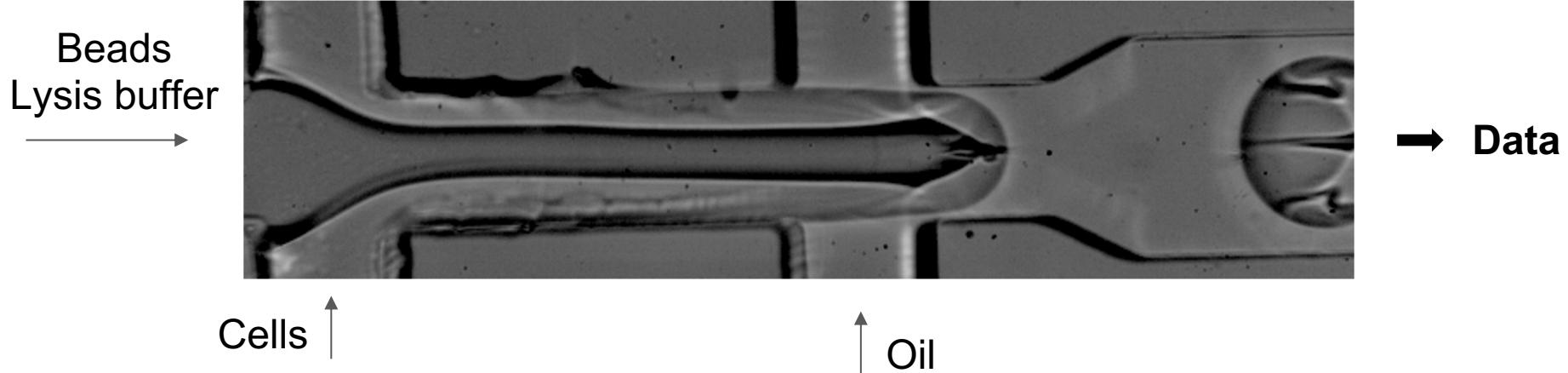
FMRI



Patient Data

Big = Any dataset with many many observations

The single cell revolution



The single cell revolution

Interesting Biological Experiments



A black arrow pointing to the right, indicating a continuation or next step.

Computation



A black arrow pointing to the right, indicating a continuation or next step.

High impact paper

LETTER

RNA velocity of single cells

Kin velocity of single cells
Giorgia La Manno¹, Basilia Soldatow¹, Annet Zeijl^{1,2}, Emelie Braun^{1,3}, Hannah Hochgerber^{1,3}, Viktor Petukhov^{1,4}, Karina Lidschweiler¹, Michaela Kühn¹, Peter Löschner¹, Alessandro Farfán¹, Jean Fan¹, Lars E. Børns¹, Zebha Liu¹, David B. Suter¹, Michael H. Hahn¹, Erik Stensdotter¹, Gonzalo Casteleiro-Brancio¹, Patrick Cramer^{1,5}, Ingrid Arshamyan¹, Steen Lindegaard^{1,6}, B. Dyrig¹, K. Kahrer^{1,7*}

RNA abundance is a powerful indicator of the state of individual cells. Single-cell RNA sequencing can reveal RNA abundance with high precision (Fig. 1A). However, the temporal dimension of this approach captures only a point in time. We therefore sought to extend this approach to reveal RNA dynamics as embryogenesis or tissue regeneration. Here we show that RNA-seq can be used to monitor RNA dynamics over time by simultaneously directly encoding by distinguishing between uncapped and capped mRNAseq to estimate single-cell RNA sequencing protocols. RNA-seq can thus be used to monitor RNA dynamics in individual cells and individual cells on a timescale of hours. We validate its accuracy and utility by monitoring RNA dynamics in mouse embryonic stem cells and technical platforms, reveal the branching lineage of RNA-seq, and demonstrate that RNA dynamics can be monitored in living cells. By monitoring RNA dynamics in mouse embryonic stem cells and in mouse embryonic brain, we expect RNA-seq to reveal the dynamics of gene expression, gene regulation, and cellular dynamics, particularly in humans.

During development, differentiation occurs on a timescale of hours to days, which is comparable to the typical half-life of mRNA. The relative abundance of nascent (unspliced) and mature (spliced) mRNA can be exploited to estimate the rates of gene splicing and degradation, without the need for metabolic labelling, as previously shown in bulk.¹ We reasoned that similar signals may be detectable in single-cell RNA

sequencing (RNA-seq) data, and could reveal the rate and duration of change of the entire transcriptome during dynamic processes. All common sense cell RNA-seq data analysis should include an analysis of unspliced mRNA levels. Numerous previous works have shown that the mRNA levels in the NM-RT-1 and 3S/5S rRNA fractions are highly correlated (Fig. 1d). The analysis of the state of mRNA abundance in the total cell RNA fraction can therefore be used to estimate the mRNA levels in the rRNA fractions.

using single-cell RNA-seq datasets obtained on the SMART-seq², STRI, C1, cl-Deep and 16S Genomics Chromium protocols^{1,2,3,4}, we found that 15–25% of the total RNA was pseudotranscribed across all platforms, in agreement with previous observations by Belli¹ (4.6%) and Engle⁵ (~20%) RNA-seq. Most such reads originated from secondary priming positions within the intronic regions (Extended Data Fig. 1). In 16S Genomics Chromium libraries, we also found abundant divergent rRNA transcripts, and thus the future state of the cell.

Genomics ChIP-seq libraries, we also found abundant discordant priming events between the two sequencing platforms (Dataset S1, Fig. 11), which may have been generated during PCR amplification by priming on the first-strand cDNA. The substantial number of intrinsic molecules and their correlation with the excess and many circadian-associated genes showed the expected excess of circadian mRNA relative to the day/night cycle amplitude, and a corresponding diurnal decrease in detection rate (Fig. 11g). Solving the proposed differential equations for each gene allowed us to extrapolate each measurement throughout the circadian cycle, accurately capturing

of interest and their corresponding mRNA levels. Our results confirm that these mRNAs encode unlabelled precursor mRNAs. This was confirmed by metabolic labelling of newly transcribed cRNA¹ followed by RNA sequencing using oligo-dT primed single-cell-tagged reverse transcription (STRT) (Extended Data Fig. 1c). The sequencing data showed that the cDNA library contained many transcripts that were longer than the expected length of the precursor mRNAs, indicating that the cDNA library contained many full-length transcripts. This demonstrates that the STRT method can be used to predict the expected direction of progression of the circadian cycle (Fig. 1b).

Fig. 2b; 83% of all genes displayed expression time courses consistent with simple first-order kinetics, as expected if unplicated mRNAs represent nascent mRNA.

dance of precursor and mature mRNA, we assumed a simple model tracing. Phase portraits of many genes showed the expected deviations

¹Department of Biochemistry and Nutrition, Karolinska Institutet, Stockholm, Sweden. ²Department of Physiology and Pharmacology, Karolinska Institutet, Stockholm, Sweden. ³Zivs-van Gendt Centre for Brain Repair, Department of Clinical Neuroscience, University of Cambridge, Cambridge, UK. ⁴Division of Neurodegeneration, Cerebral Sciences and Society, Karolinska Institutet, Stockholm, Sweden. ⁵Max-Planck Institute for Biophysical Chemistry, Department of Molecular Biology, Göttingen, Germany. ⁶Ronald Stern Cell Institute, Cambridge, MA, USA. ⁷Macmillan Cancer Support, Peter Macmillan Research Unit, London, United Kingdom.

694 | NATURE | VOL 340 | 23 AUGUST 2006

- Machine learning
 - Linear algebra
 - Probability theory
 - Statistical analysis
 - Algorithm design

It's all Greek to me...

Definition 1. The t -step potential distance is defined as $\mathfrak{V}^t(x, y) \triangleq \|U_x^t - U_y^t\|_2$, $x, y \in \mathcal{X}$.

The following proposition shows a relation between the two metrics by expressing the potential distance in embedded diffusion map coordinates¹ for fixed-bandwidth Gaussian-based diffusion (i.e., generated by P_ε from Eq. 2):

Proposition 1. Given a diffusion process defined by a fixed-bandwidth Gaussian kernel, the potential distance from Def 1 can be written as $\mathfrak{V}^t(x, y) = \left(\sum_{z \in \mathcal{X}} \log^2 \left(\frac{1 + \langle \Phi_\varepsilon^{t/2}(x), \Phi_\varepsilon^{t/2}(z) \rangle}{1 + \langle \Phi_\varepsilon^{t/2}(y), \Phi_\varepsilon^{t/2}(z) \rangle} \right) \right)^{1/2}$

Proof. According to the spectral theorem, the entries of P_ε^t can be written as

$$[P_\varepsilon^t]_{(x,y)} = \psi_0(y) + \sum_{i=1}^{n-1} \lambda_i^t \phi_i(x) \psi_i(y)$$

since powers of the operator P_ε only affect the eigenvalues, which are taken to the same power, and since the trivial eigenvalue λ_0 is one and the corresponding right eigenvector ϕ_0 only consists of ones. Furthermore, it can be verified that the left and right eigenvectors of P_ε are related by $\psi_i(y) = \phi_i(y)\psi_0(y)$, thus, combined with Eqs. 4 and 6, we get

$$p_{\varepsilon,x}^t(y) = \psi_0(y) \left(1 + \sum_{i=1}^{n-1} \lambda_i^t \phi_i(x) \phi_i(y) \right) = \psi_0(y) (1 + \langle \Phi_\varepsilon^{t/2}(x), \Phi_\varepsilon^{t/2}(y) \rangle) .$$

By applying the logarithm to both ends of this equation we express the entries of the potential representation $U_{\varepsilon,x}^t$ as

$$U_{\varepsilon,x}^t(y) = -\log(1 + \langle \Phi_\varepsilon^{t/2}(x), \Phi_\varepsilon^{t/2}(y) \rangle) - \log(\psi_0(y)) ,$$

and thus for any $j = 1, \dots, N$,

$$\begin{aligned} (U_{\varepsilon,x}^t(x_j) - U_{\varepsilon,y}^t(x_j))^2 &= [\log(1 + \langle \Phi_\varepsilon^{t/2}(x), \Phi_\varepsilon^{t/2}(x_j) \rangle)]^2 \\ &\quad - [\log(1 + \langle \Phi_\varepsilon^{t/2}(y), \Phi_\varepsilon^{t/2}(x_j) \rangle)]^2 \\ &= \log^2 \left(\frac{1 + \langle \Phi_\varepsilon^{t/2}(x), \Phi_\varepsilon^{t/2}(x_j) \rangle}{1 + \langle \Phi_\varepsilon^{t/2}(y), \Phi_\varepsilon^{t/2}(x_j) \rangle} \right) , \end{aligned}$$

which yields the result in the proposition. \square

What reading single cell methods can feel like



What is machine learning?

What is machine learning?

Machine learning is the process of identifying patterns in data.

Two kinds of machine learning

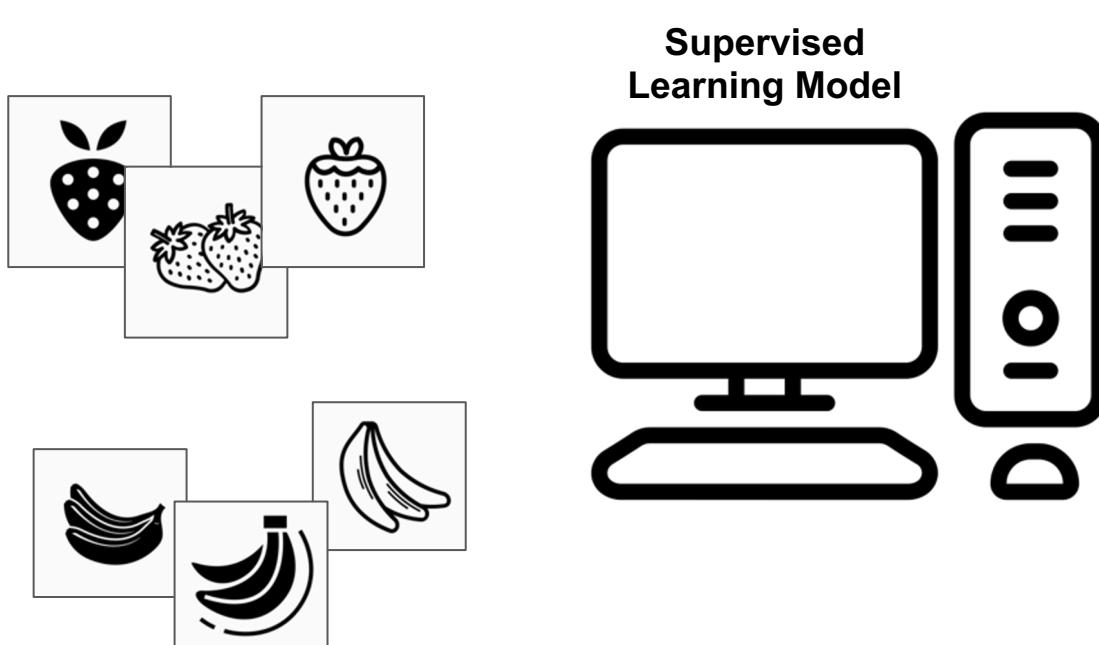
Supervised learning

- Have a bunch of labelled data, want to label new data

Two kinds of machine learning

Supervised learning

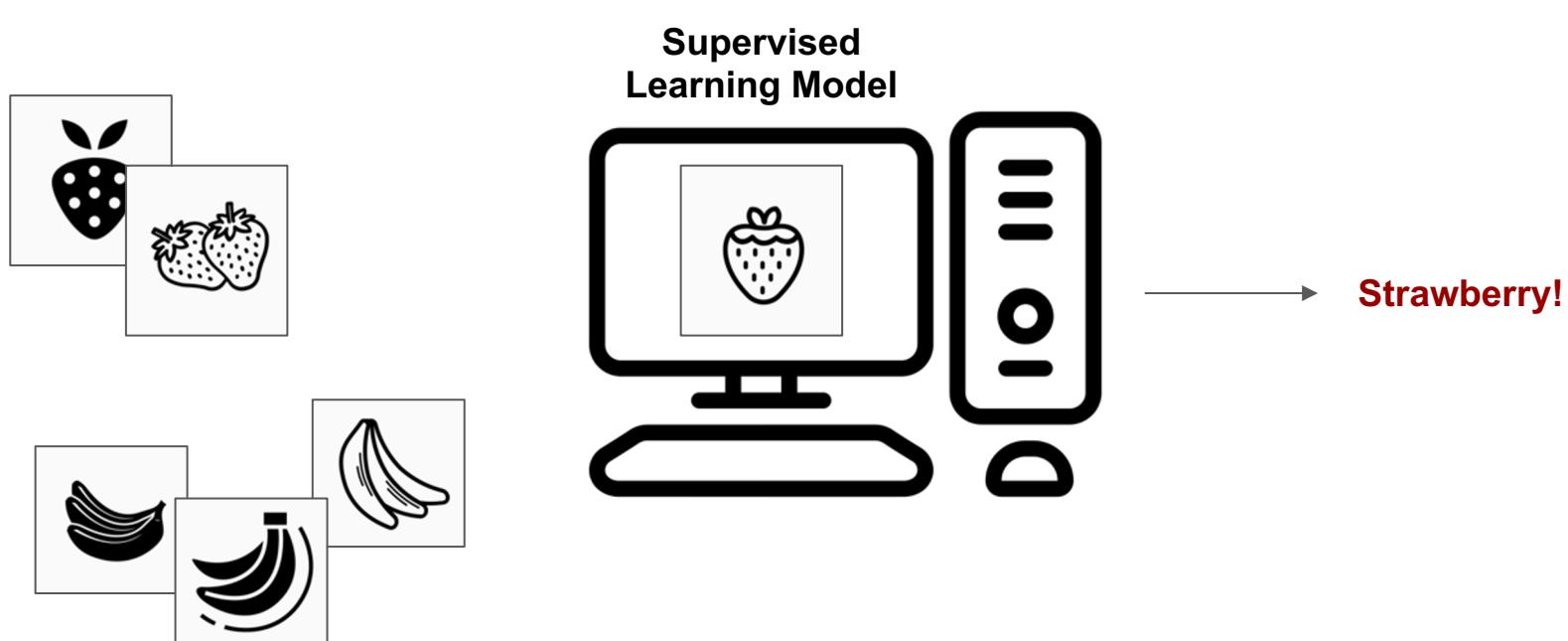
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

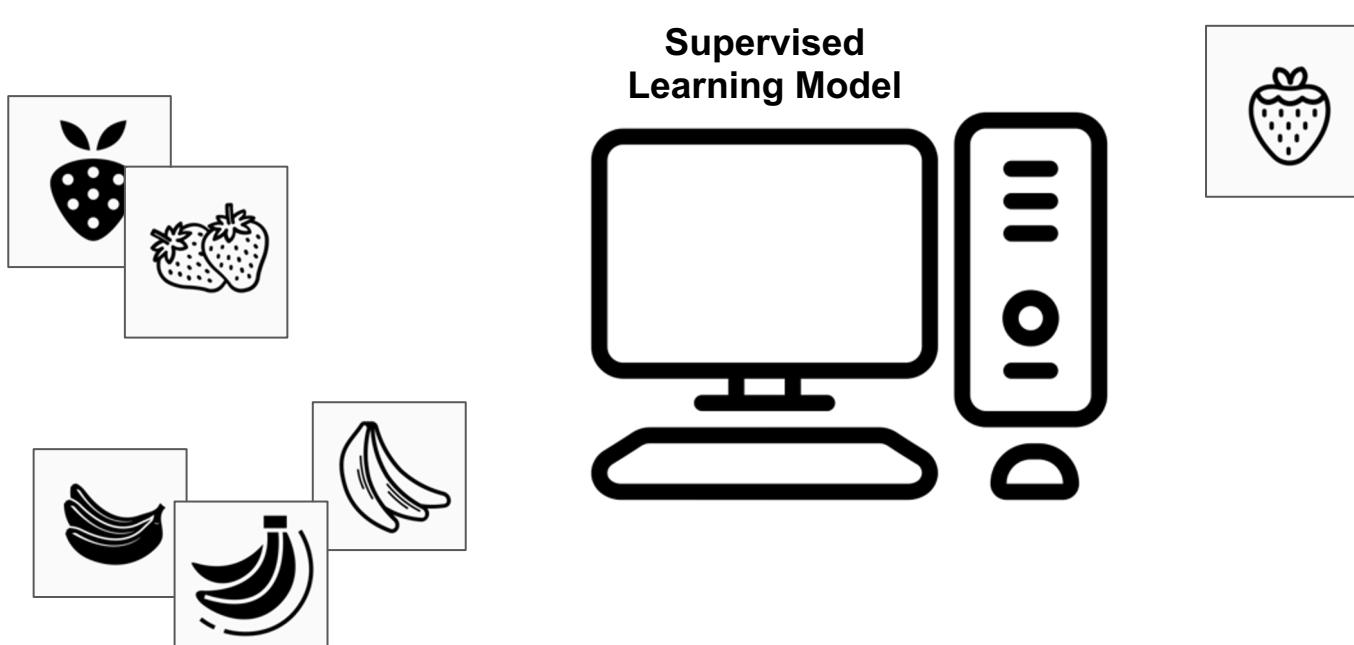
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

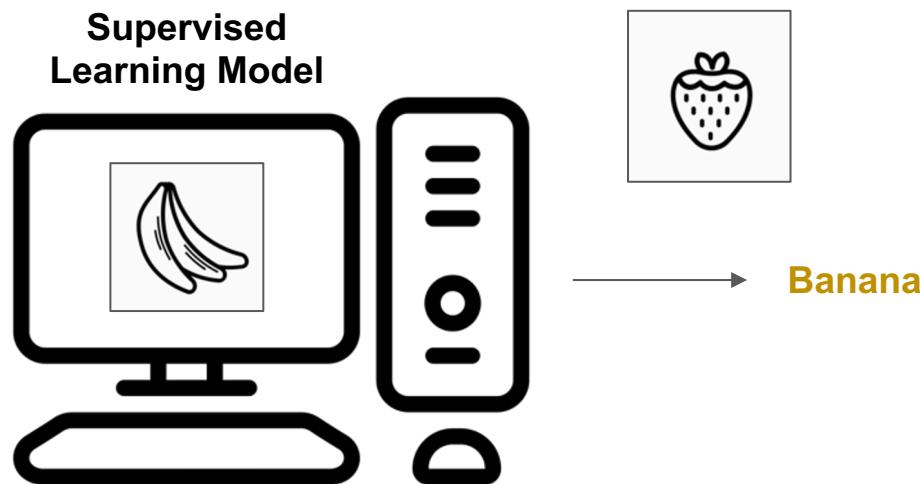
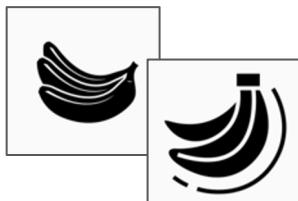
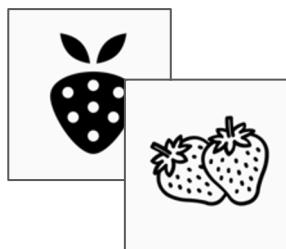
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

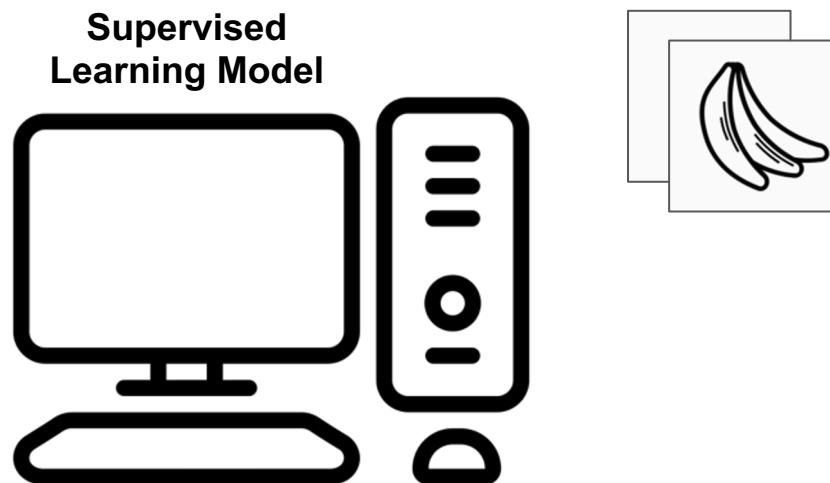
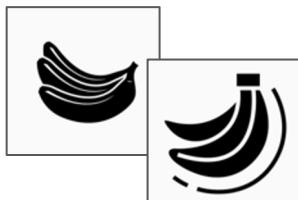
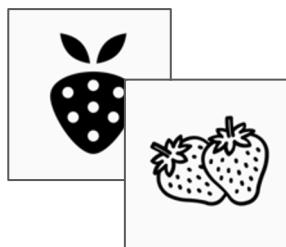
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

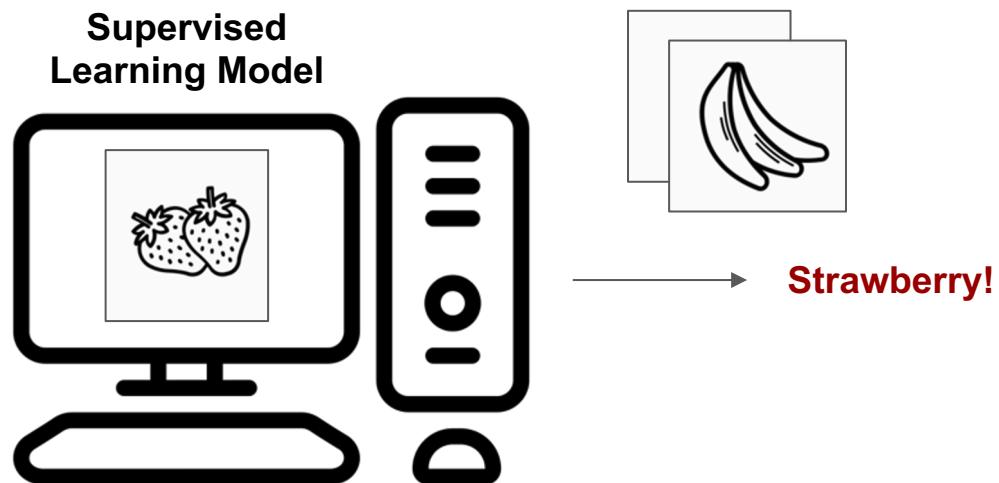
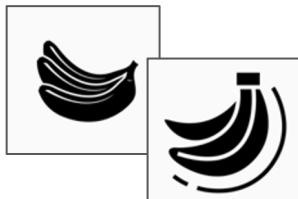
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

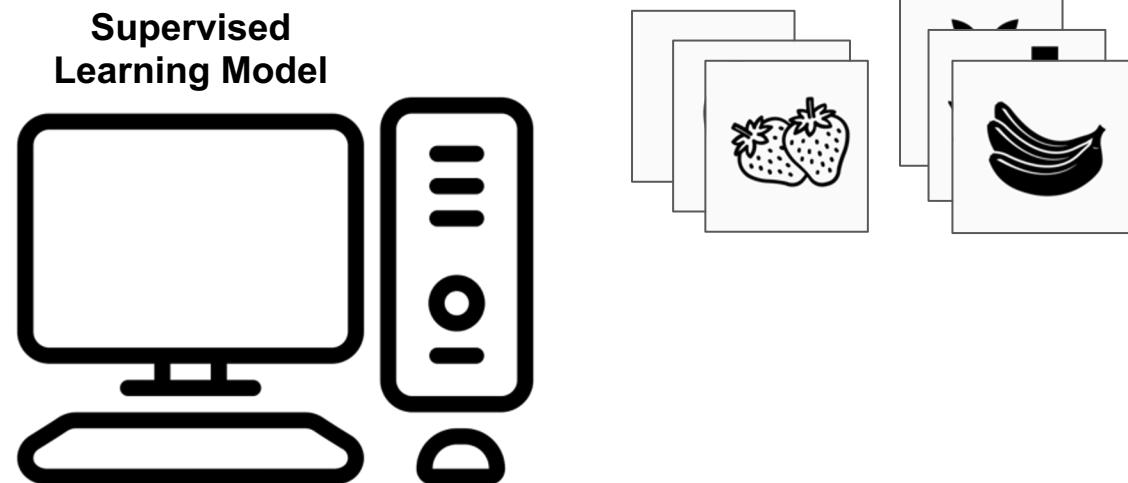
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

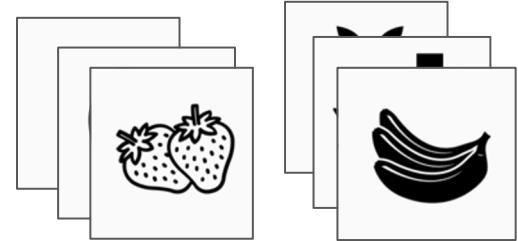
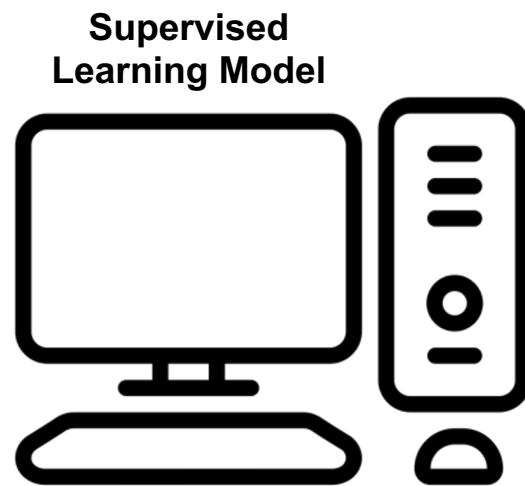
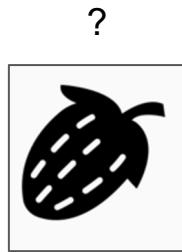
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

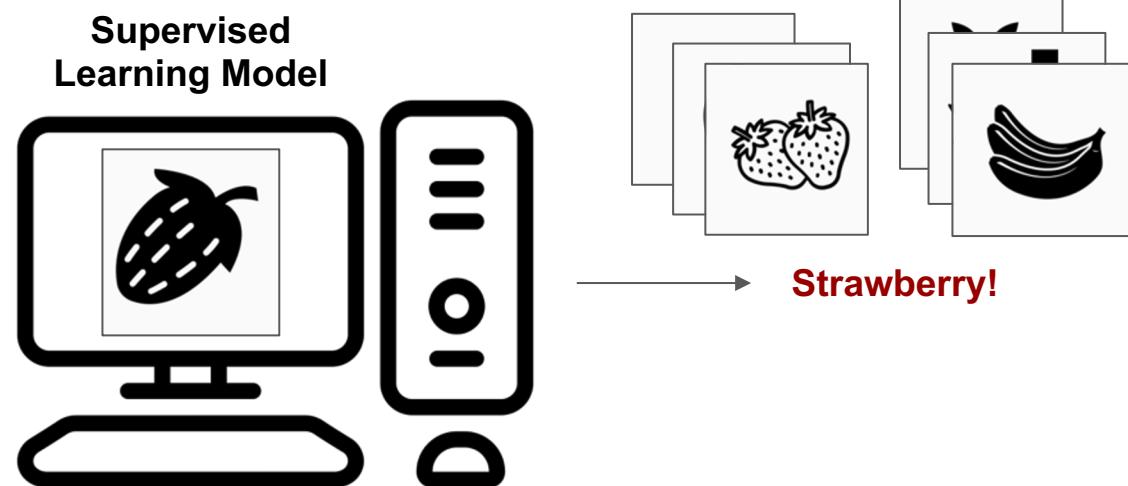
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data

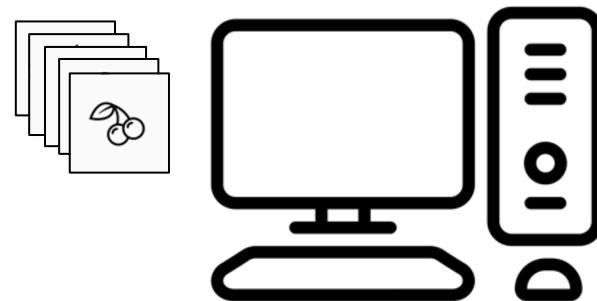
Supervised Learning Model



Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

Unsupervised Learning Model



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data

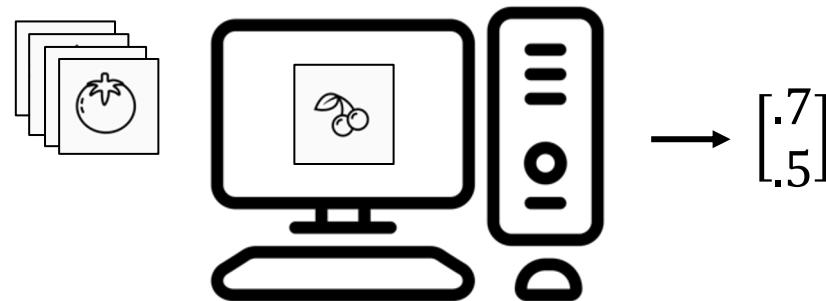
Supervised Learning Model



Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

Unsupervised Learning Model



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data

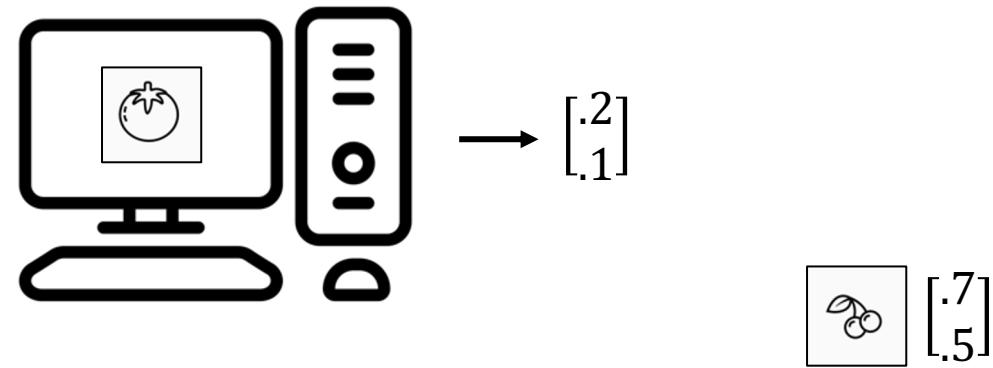
Supervised Learning Model



Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

Unsupervised Learning Model



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data

Supervised Learning Model

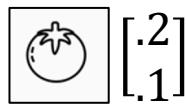
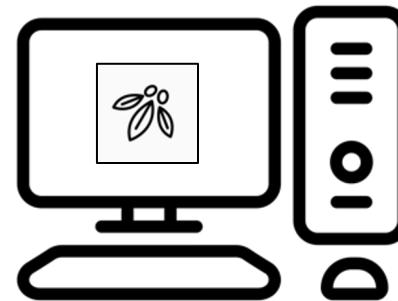


→ Strawberry

Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

Unsupervised Learning Model



Two kinds of machine learning

Supervised learning

- Have a bunch of labelled data, want to label new data

Supervised Learning Model

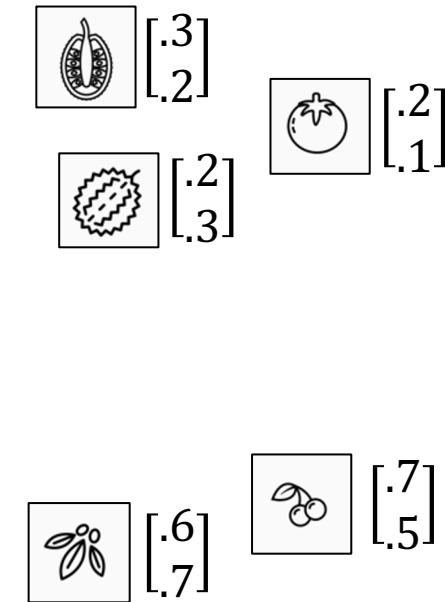


→ Strawberry

Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

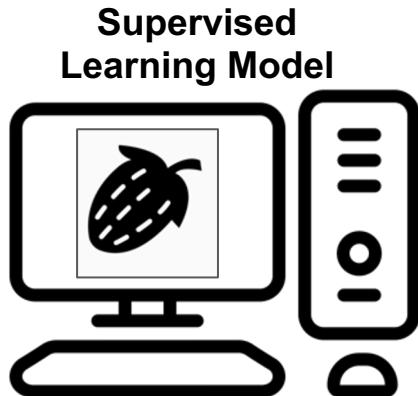
Unsupervised Learning Model



Two kinds of machine learning

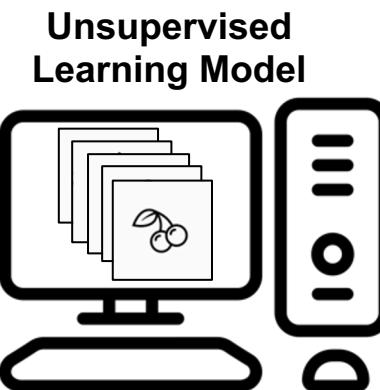
Supervised learning

- Have a bunch of labelled data, want to label new data
- Learn a function $f(X) \rightarrow Y$ where all values of Y are known for some samples of X



Unsupervised learning

- Have a bunch of unlabeled data, want to organize it
- Learn an embedding $f(X) \rightarrow Y, X \in \mathbb{R}^n, Y \in \mathbb{R}^m, n \gg m$
- Lower dimensional, easier to interpret (e.g. as clusters)



Is linear regression an example of supervised or unsupervised machine learning?

Supervised
machine
learning

Unsupervised
machine
learning

Is clustering an example of supervised or unsupervised machine learning?



Supervised
machine learning **A**

Unsupervised
machine learning **B**

Course Schedule

The screenshot shows a web browser window titled "Workshop — Krishnaswamy Lab". The URL is <https://www.krishnaswamylab.org/workshop>. The page content is organized into sections for each day of the workshop.

Course Schedule

Day 1 – Wednesday, May 20th

Lecture	View on Google Drive	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	1.1. Loading and pre-processing your own data (optional)

Day 2 – Thursday, May 21st

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

Day 3 – Friday, May 22nd

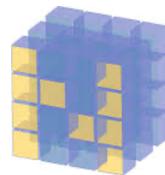
Lecture	View on Google Drive	Clustering and Data Denoising
Exercise	Run in Google Colab	3.0 Clustering Toy Data (Beginner)
	Run in Google Colab	3.0 Clustering Toy Data (Advanced)
	Run in Google Colab	3.1 Clustering & Denoising Embryoid Body Data (Advanced)
	Run in Google Colab	3.2 Batch correction in PBMCs

Day 4 – Wednesday, May 27th

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

Why Python?

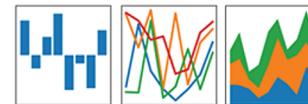




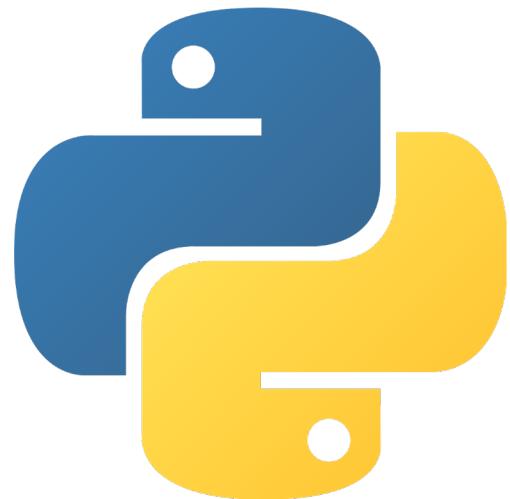
NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Why Python?



Tensorflow



Pytorch



The screenshot shows the Google Colab interface. At the top, there's a browser-like header with tabs, a search bar, and various icons. Below it is the Colab navigation bar with links for File, Edit, View, Insert, Runtime, Tools, Help, and a user profile icon. A sidebar on the left contains a 'Table of contents' section with links to Getting started, Data science, Machine learning, More Resources, and Machine Learning Examples. There's also a '+ SECTION' button. The main content area features a large yellow 'CO' logo and the title 'What is Colab?'. It explains that Colab allows writing and executing Python in a browser with zero configuration, free access to GPUs, and easy sharing. It encourages users to watch the 'Introduction to Colab' video. A section titled 'Getting started' is expanded, showing a code cell with the Python script:

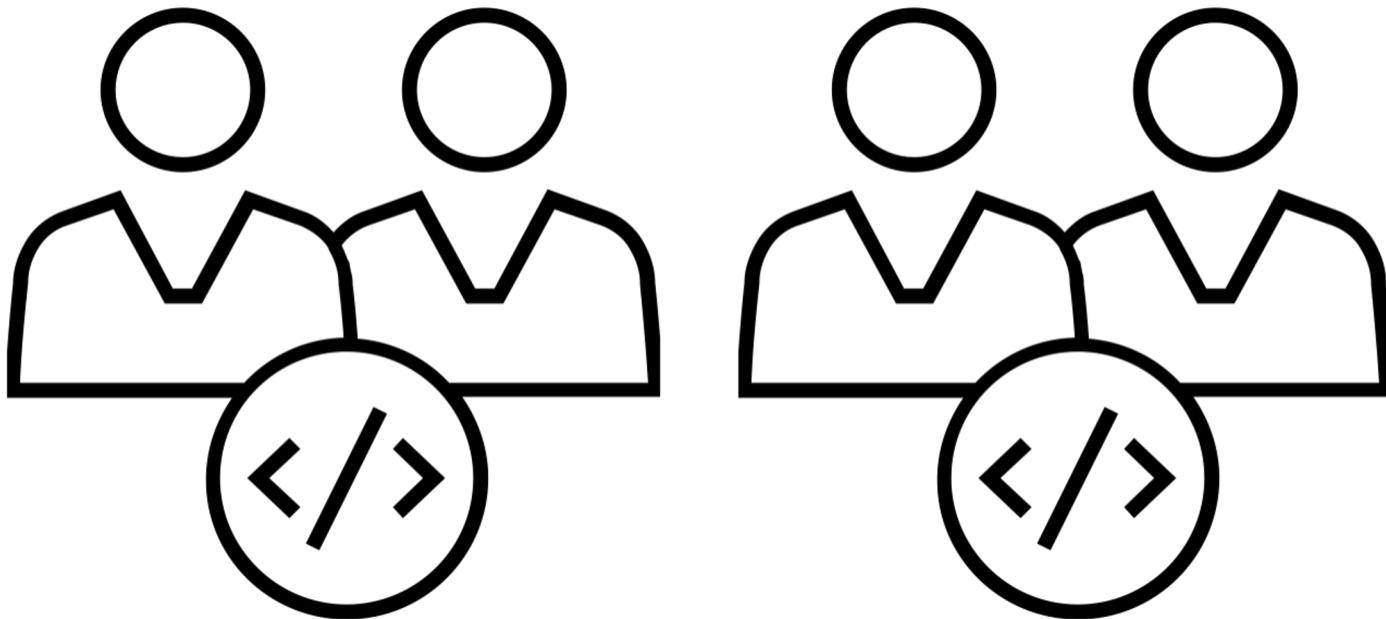
```
[ ] seconds_in_a_day = 24 * 60 * 60
```

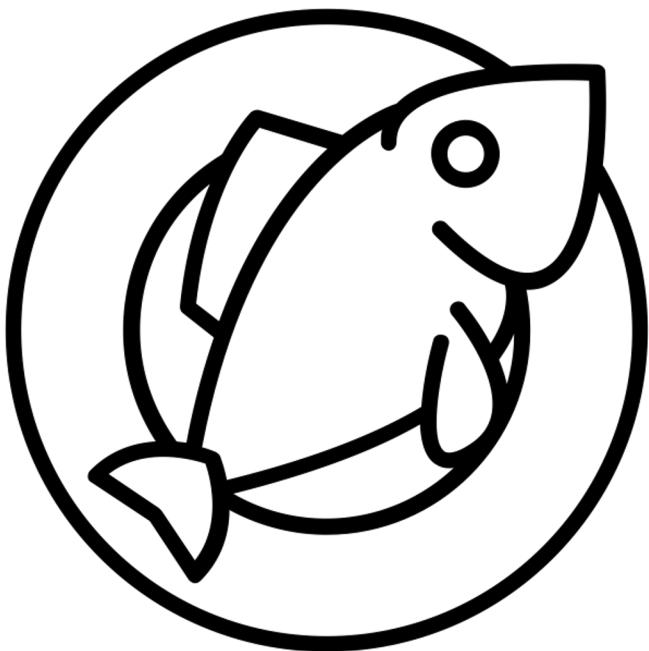
seconds_in_a_day

72000

 and instructions on how to execute it. A footer at the bottom of the content area says 'Variables that you define in one cell can later be used in other cells.'

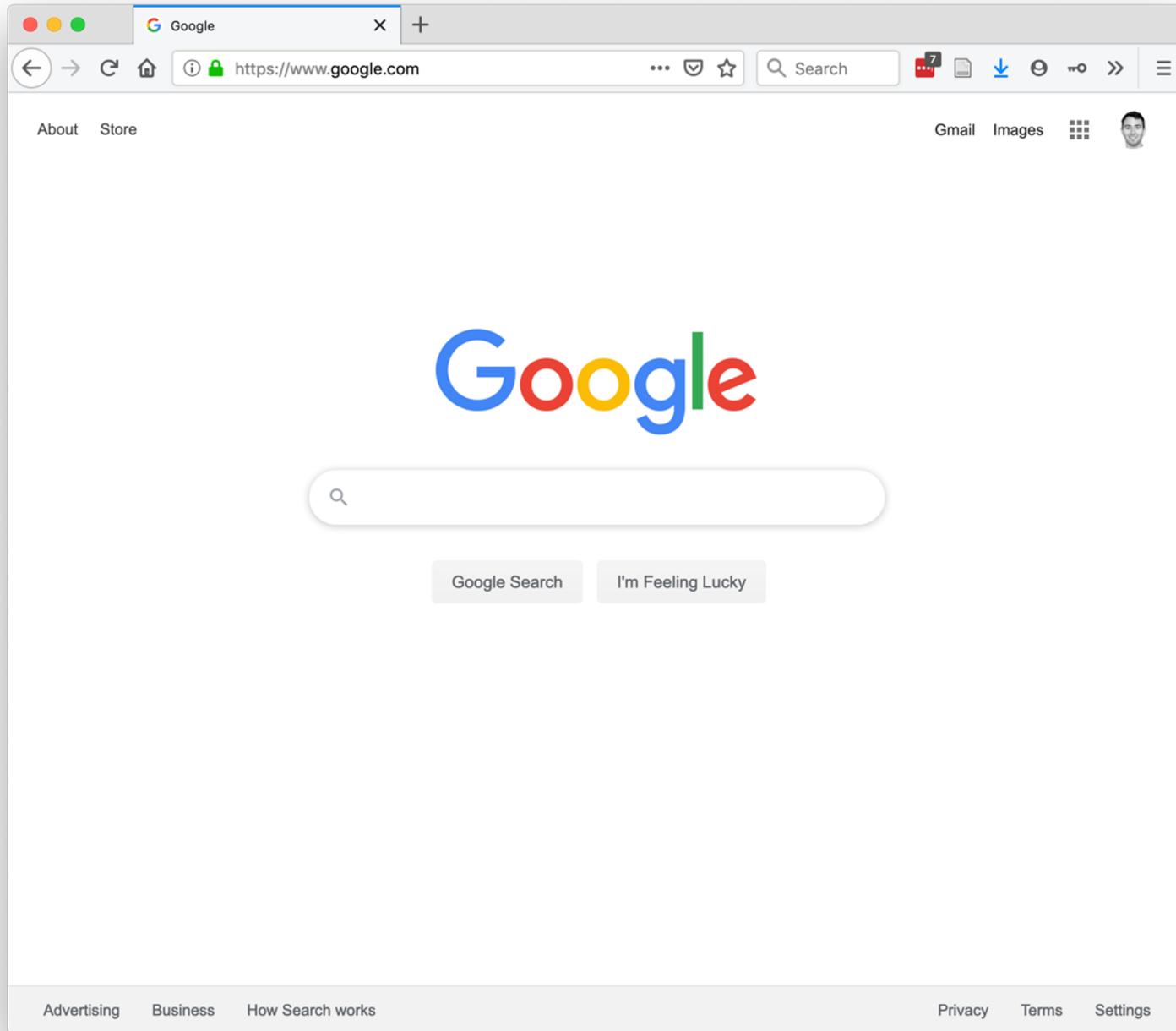
Team programming





vs.





Reference — scprep 1.0.1 documentation

scprep.io.load_10X(`data_dir`, `sparse=True`, `gene_labels='symbol'`, `allow_duplicates=None`) [source]

Basic IO for 10X data produced from the 10X Cellranger pipeline.

A default run of the `cellranger count` command will generate gene-barcode matrices for secondary analysis. For both “raw” and “filtered” output, directories are created containing three files: ‘matrix.mtx’, ‘barcodes.tsv’, ‘genes.tsv’. Running `scprep.io.load_10X(data_dir)` will return a Pandas DataFrame with genes as columns and cells as rows.

Parameters:

- `data_dir (string)` – path to input data directory expects ‘matrix.mtx’, ‘genes.tsv’, ‘barcodes.tsv’ to be present and will raise an error otherwise
- `sparse (boolean)` – If True, a sparse Pandas DataFrame is returned.
- `gene_labels (string, {'id', 'symbol', 'both'}) optional, default: 'symbol'` – Whether the columns of the dataframe should contain gene ids or gene symbols. If ‘both’, returns symbols followed by ids in parentheses.
- `allow_duplicates (bool, optional (default: None))` – Whether or not to allow duplicate gene names. If None, duplicates are allowed for dense input but not for sparse input.

Returns:

`data` – If sparse, data will be a pd.DataFrame[pd.SparseArray]. Otherwise, data will be a pd.DataFrame.

Return type:

array-like, shape=[n_samples, n_features]

scprep.io.load_10X_HDF5(`filename`, `genome=None`, `sparse=True`, `gene_labels='symbol'`, `allow_duplicates=None`, `backend=None`) [source]

Basic IO for HDF5 10X data produced from the 10X Cellranger pipeline.

Installation Examples Reference Data Input/Output Filtering Normalization Transformation Measurements Statistics Plotting Dimensionality Reduction Row/Column Selection Utilities External Tools

The POWERFUL PYTHON PLAYBOOK for intermediate+ Python. Download free here

Sponsored · Ads served ethically

Read the Docs v: stable ▾

Bring-your-own-data workshop



#2021-workshop-byod-help
<https://krishnaswamylab.org/get-help>

Workshop Materials — Krishnas X +

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

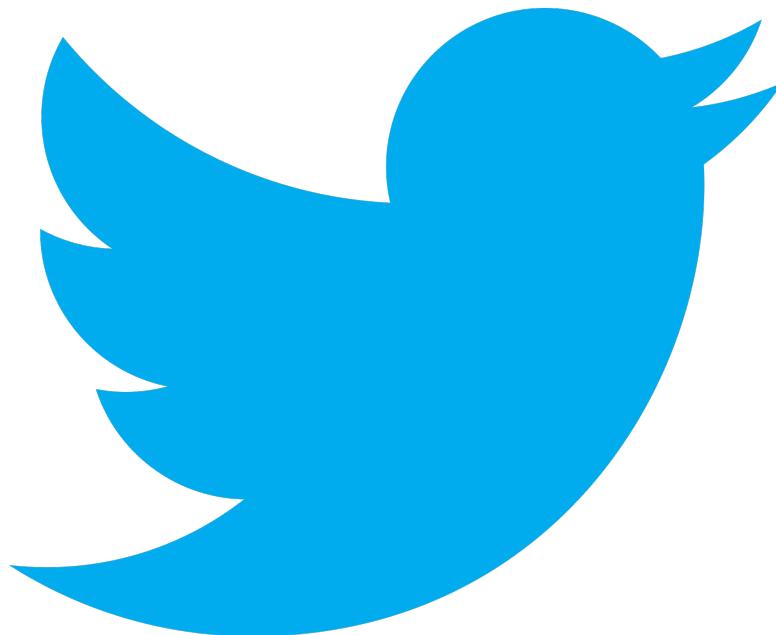
Useful links to outside resources

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



@KrishnaswamyLab
#YaleML

Data Matrices and Representations

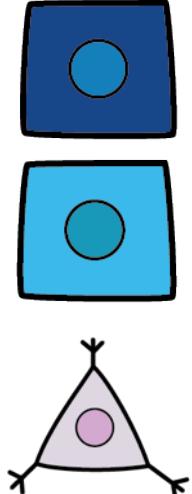
Single Cell Data

- Each cell is a vector of measurements
 - e.g. Cell A = [40 0 20 18 5 0 ...]
- The whole data is a matrix with many observations (cells) and features (proteins, genes)

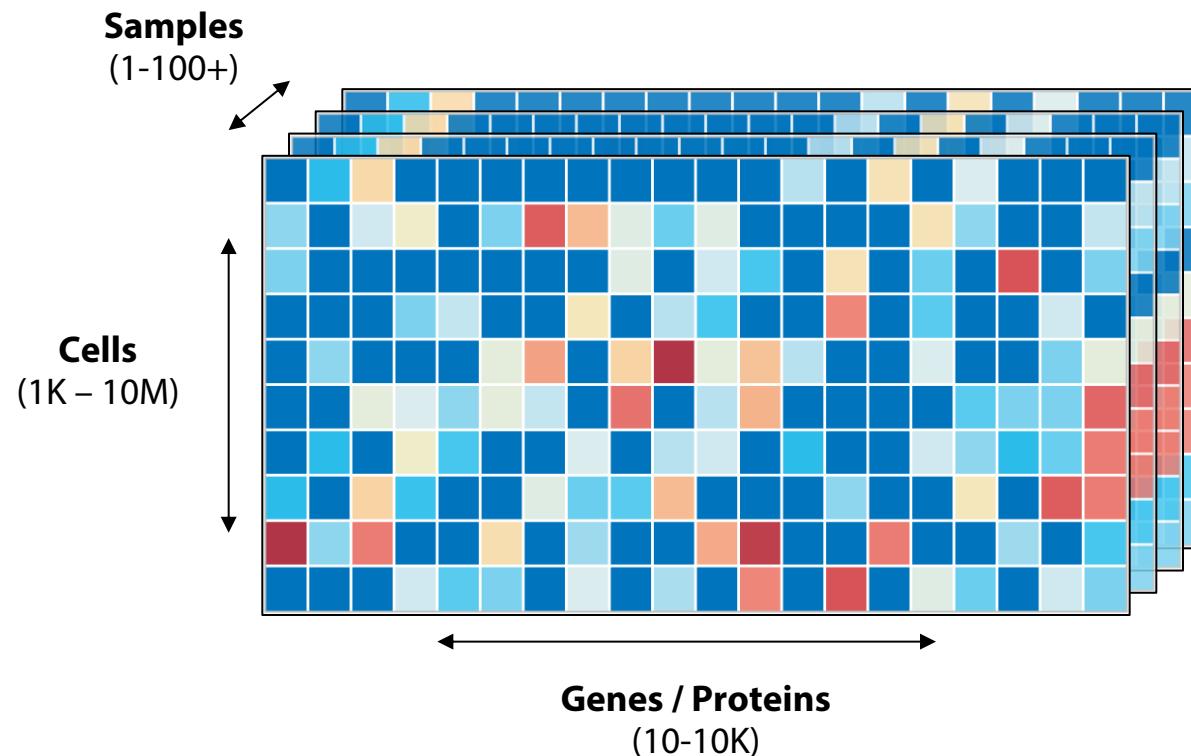
Features
(e.g. genes)

	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

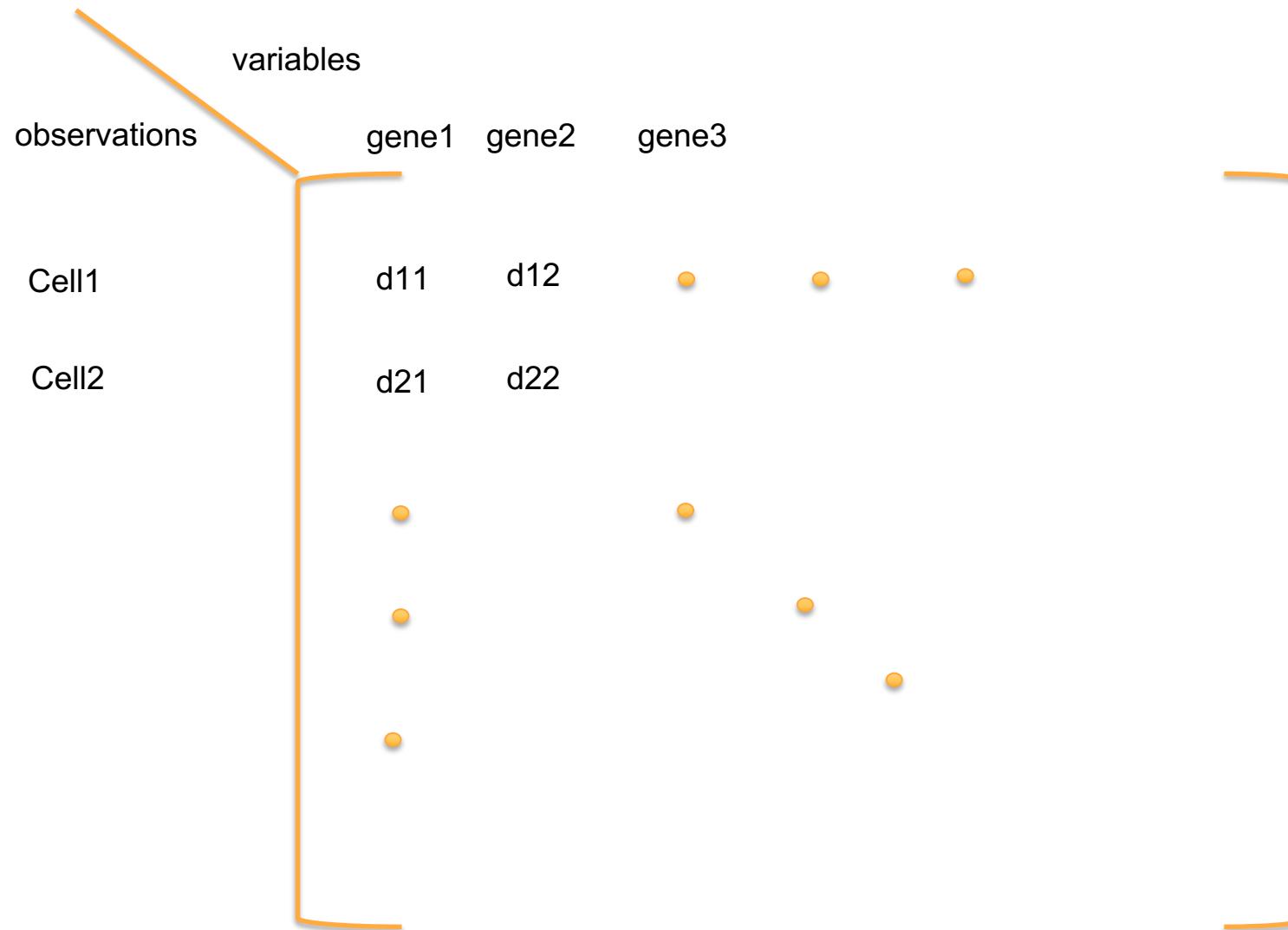
Observations
(e.g. cells)



Single Cell Data

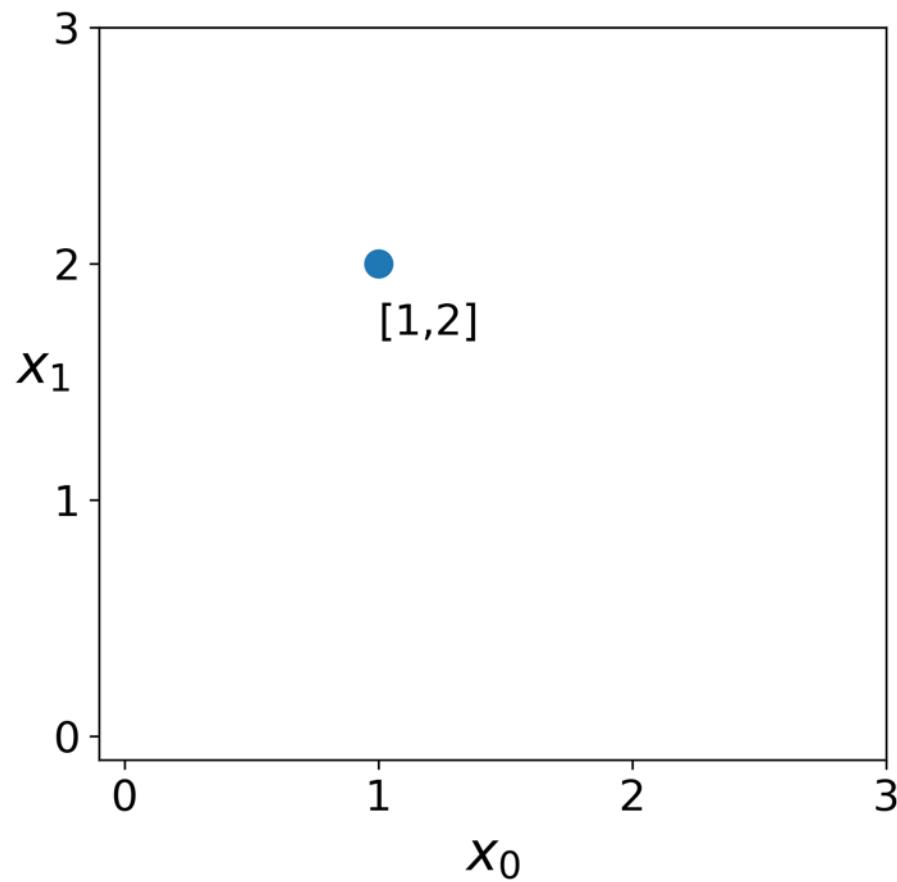


Our Data is a High-dimensional Matrix



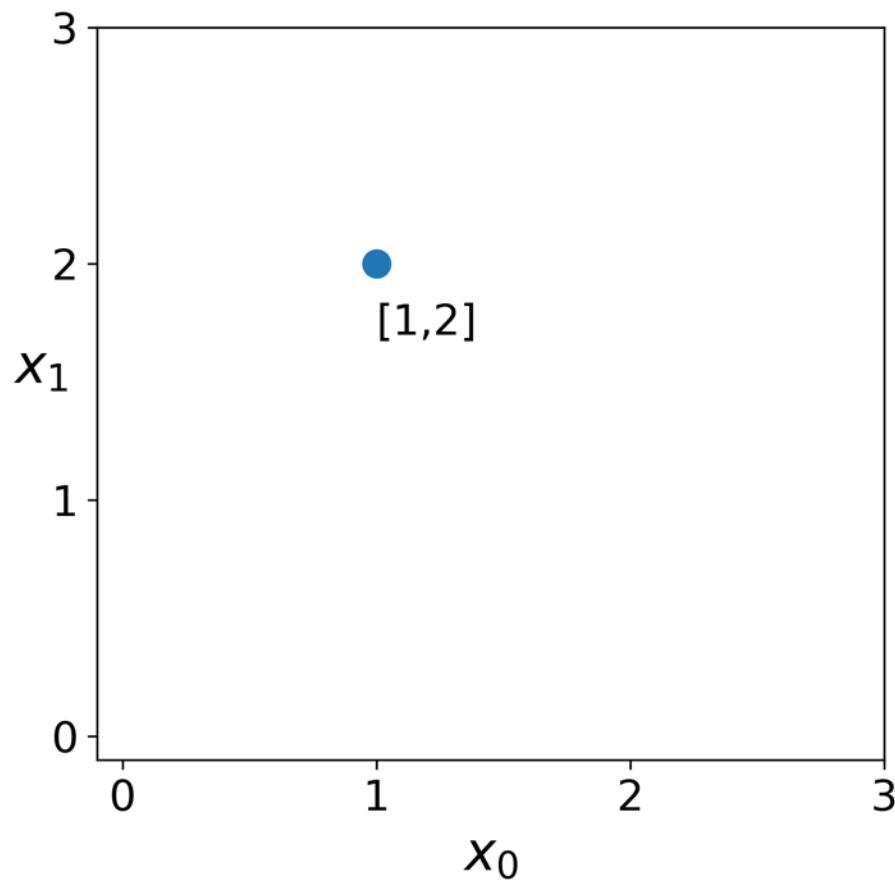
Spatial Representation of Data

\mathbb{R}^2 two-dimensional space



Spatial Representation of Data

\mathbb{R}^2 two-dimensional space



Spatial Representation of Data

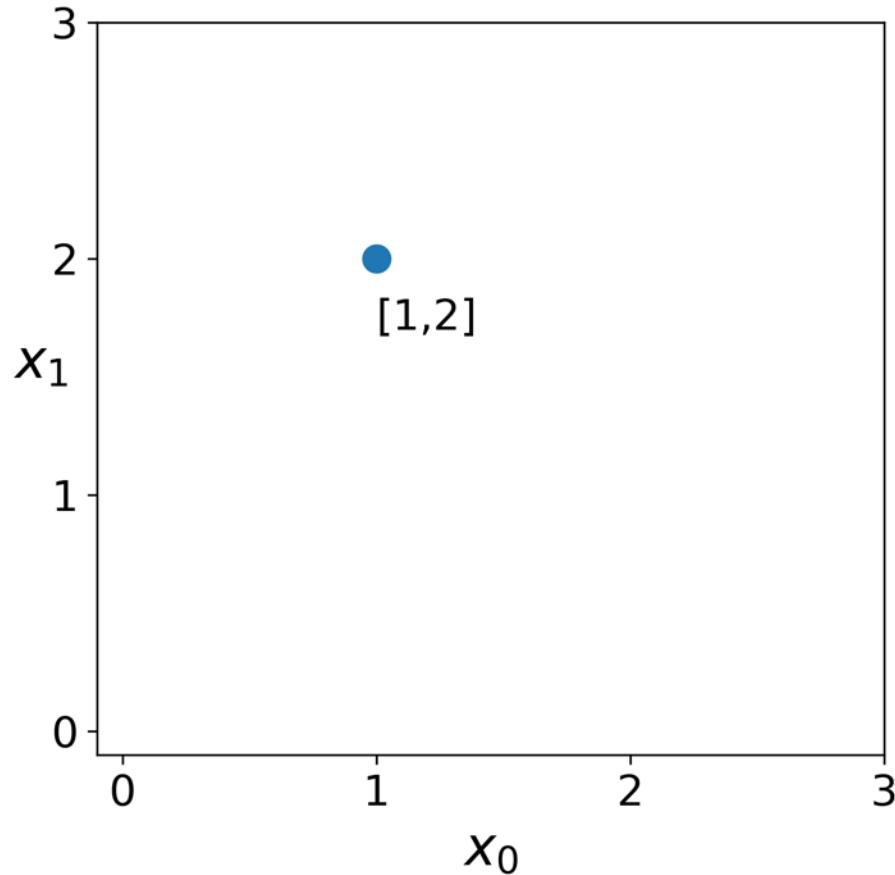
$$a \in \mathbb{R}^2$$

$$a = [1, 2]$$

x_0

x_1

\mathbb{R}^2 two-dimensional space



Features are dimensions

$$\mathbf{a} \in \mathbb{R}^2$$

$$\mathbf{a} = [1, 2]$$

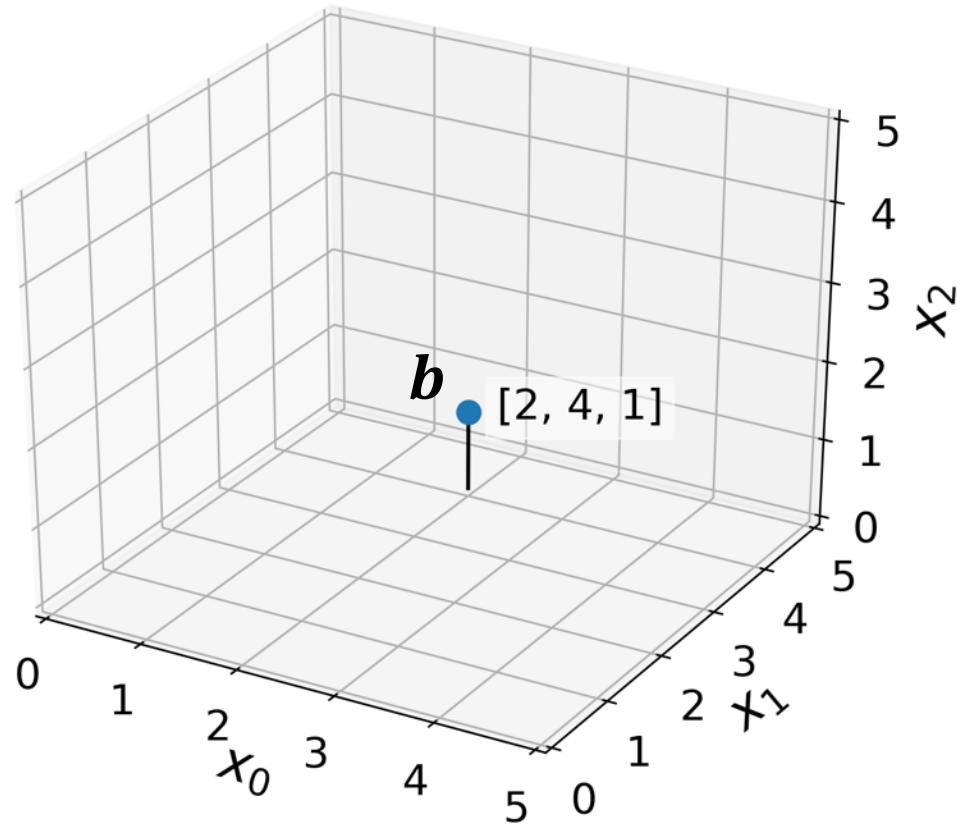
x_0 x_1

$$\mathbf{b} \in \mathbb{R}^3$$

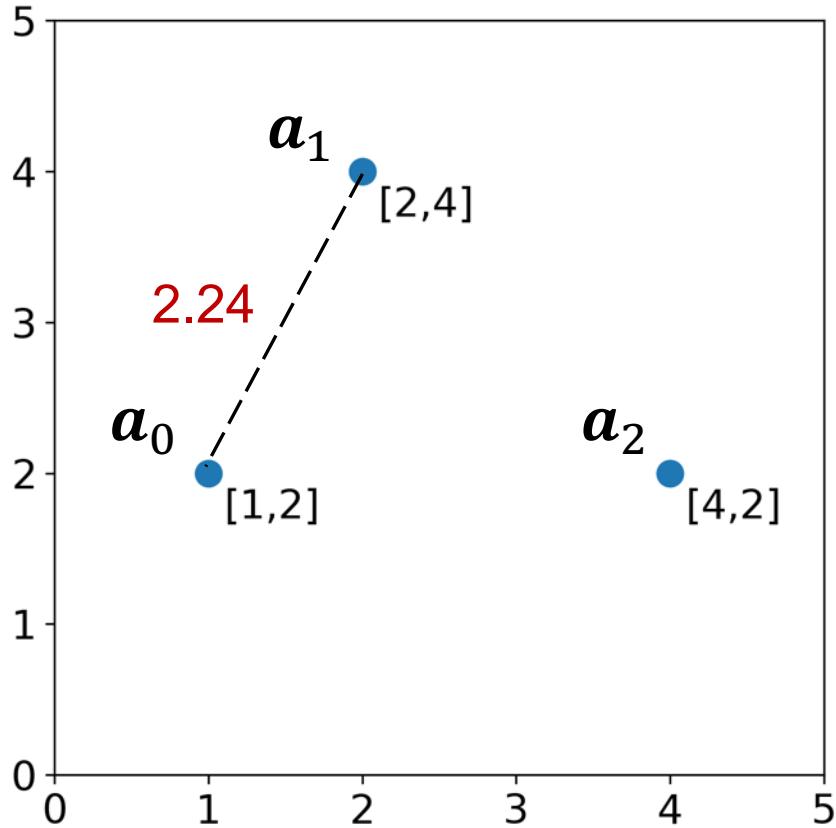
$$\mathbf{b} = [2, 4, 1]$$

x_0 x_1 x_2

\mathbb{R}^3 three-dimensional space

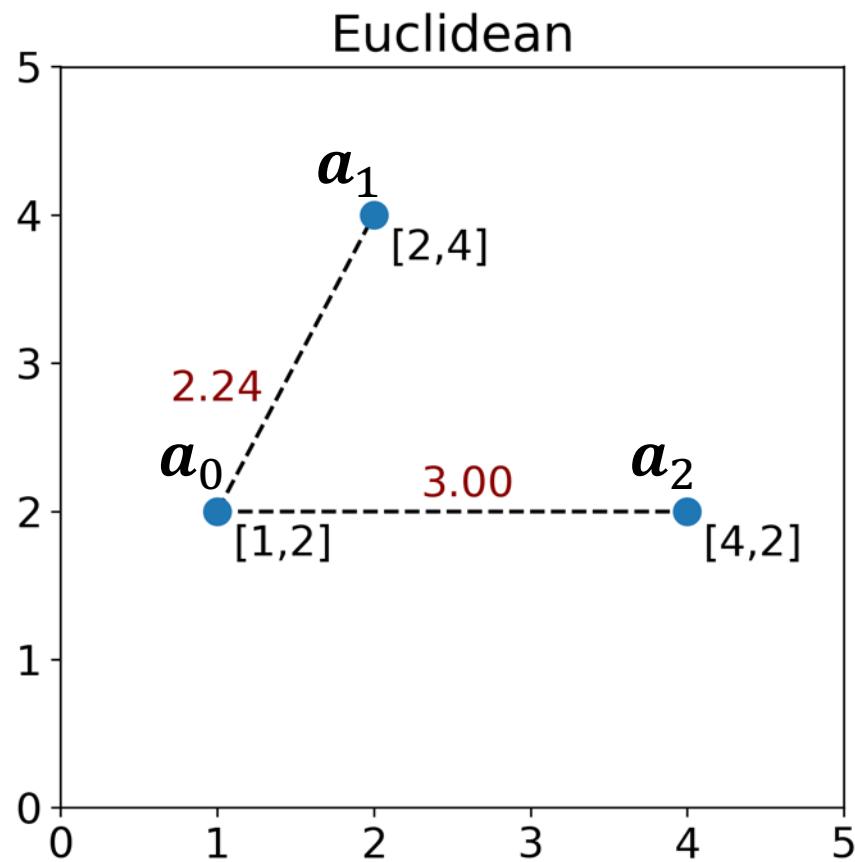


Euclidean Distance between observations

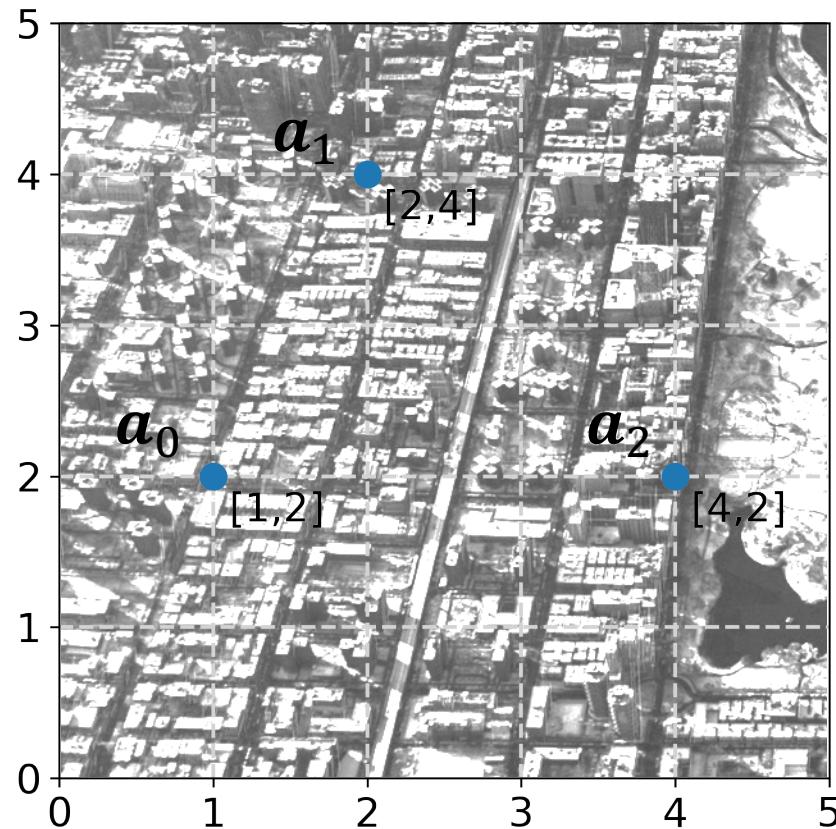


$$\begin{aligned}d_{euclidean}(a_0, a_1) &= \|a_0 - a_1\|_2 \\&= \sqrt{(a_{0,0} - a_{1,0})^2 + (a_{0,1} - a_{1,1})^2} \\&= \sqrt{(1 - 2)^2 + (2 - 4)^2} \\&\approx 2.24\end{aligned}$$

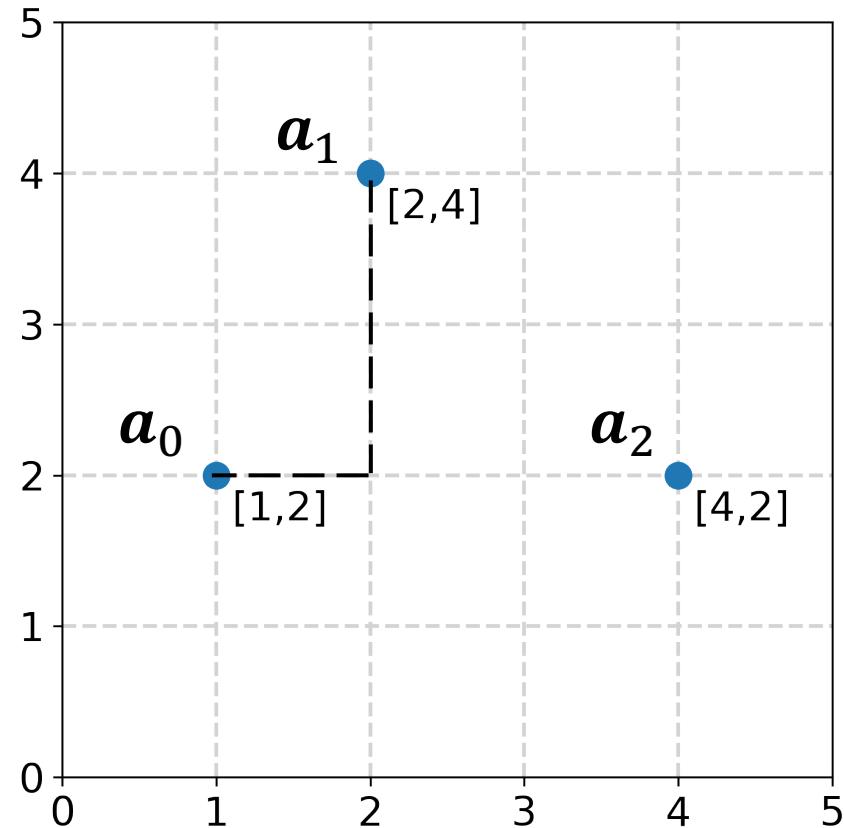
How far away is each point from a_0 ?



How far away is each point from a_0 ?



Manhattan Distance



Manhattan distance

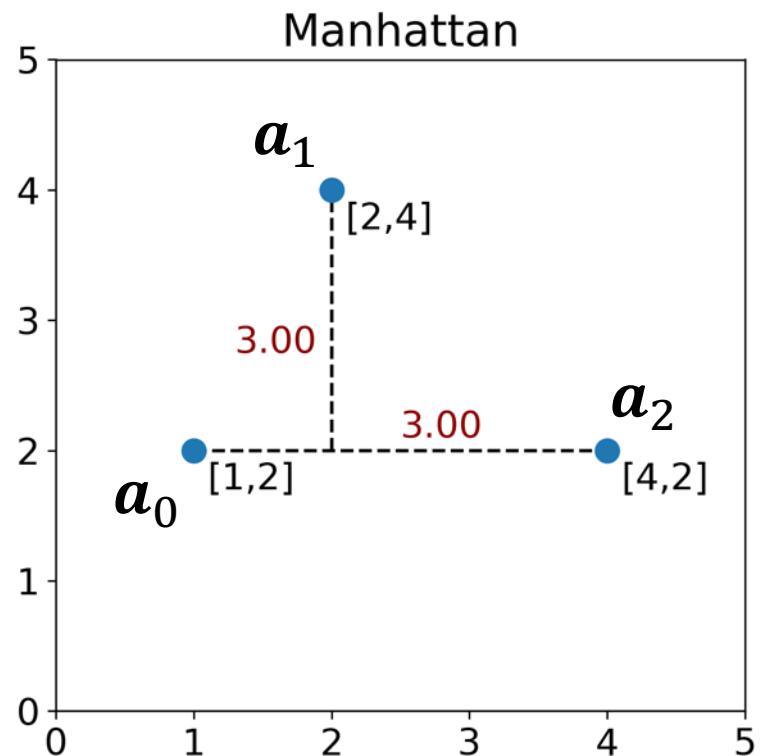
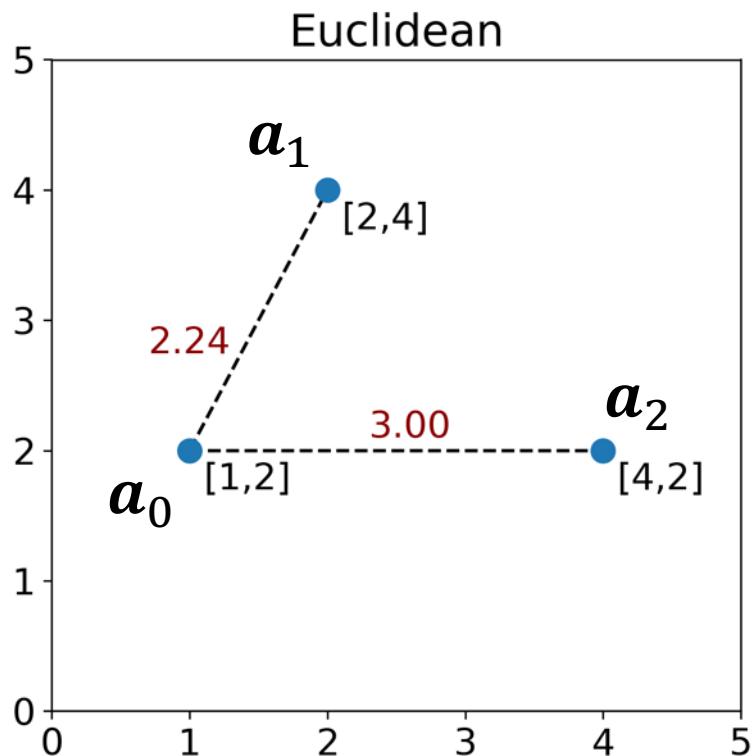
$$d_{manhattan}(a_0, a_1) = |a_{0,1} - a_{1,1}|$$

$$= |a_{0,0} - a_{1,0}| + |a_{0,1} - a_{1,1}|$$

$$= |1 - 2| + |2 - 4|$$

$$= 3$$

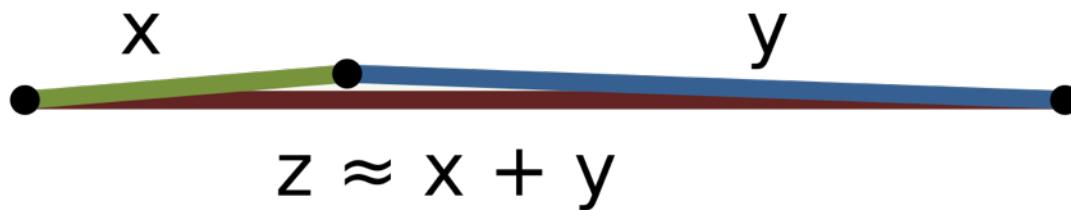
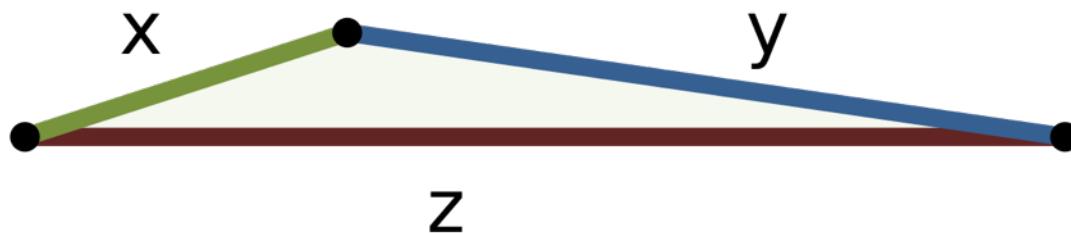
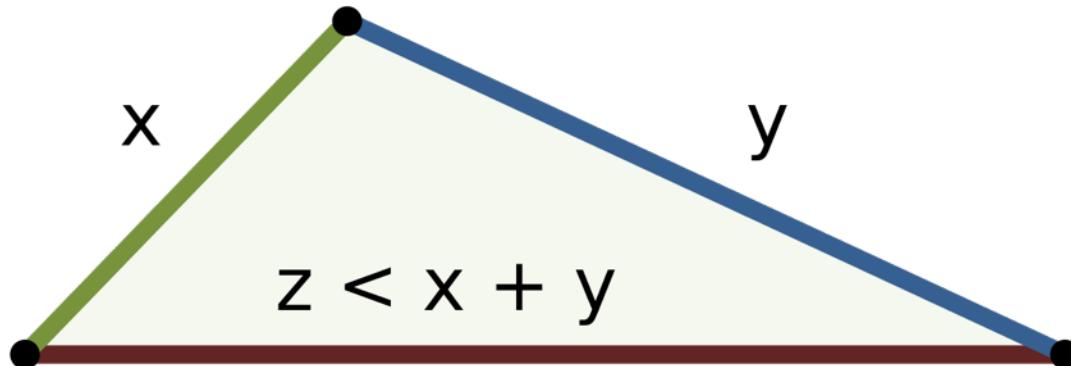
How far away is each point from a_0 ?



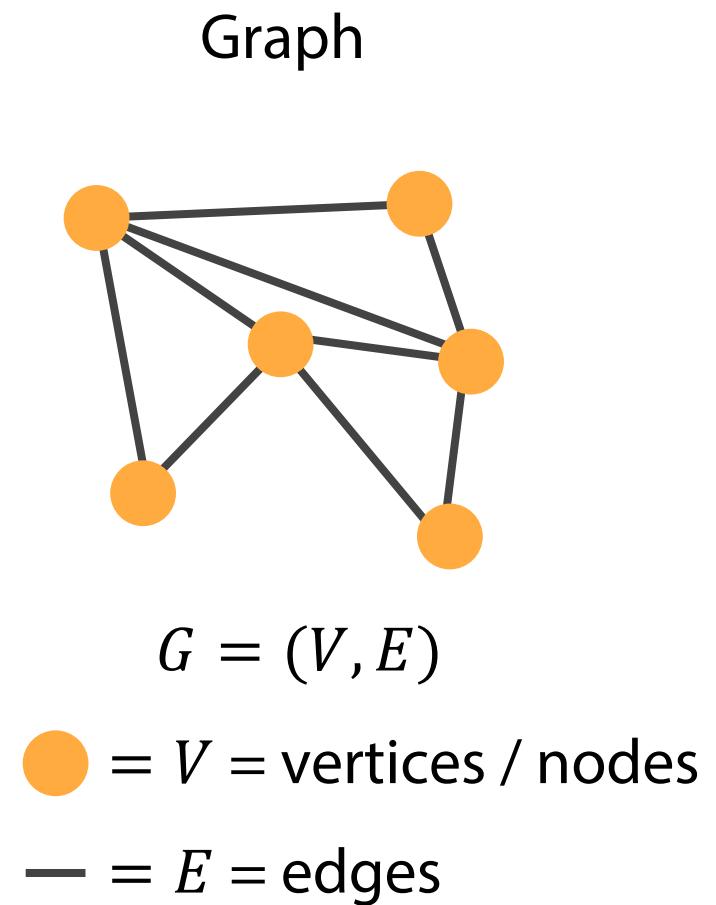
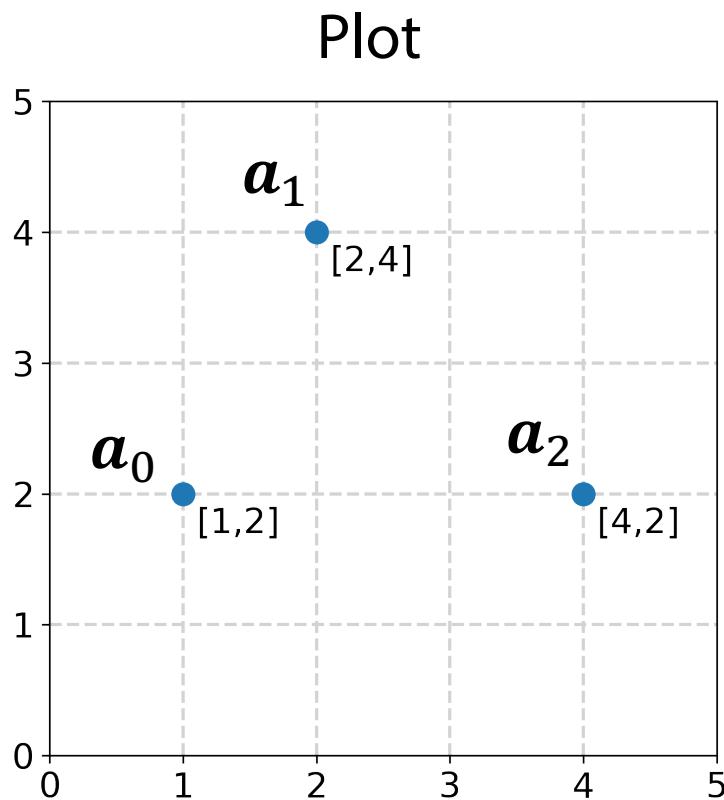
Distances

- There are many ways to measure distance
 - Hamming, Euclidean, Cosine
- Distances are functions that take two points and return a real number that is **positive or 0**
- Distances can be any function that is:
 - **Symmetric:** $\text{dist}(a \rightarrow b) = \text{dist}(b \rightarrow a)$
 - **Non-negative:** $\text{dist}(a \rightarrow b) \geq 0$
 - **Follow triangle inequality:** $\text{dist}(a \rightarrow c) \leq \text{dist}(a \rightarrow b) + \text{dist}(b \rightarrow c)$

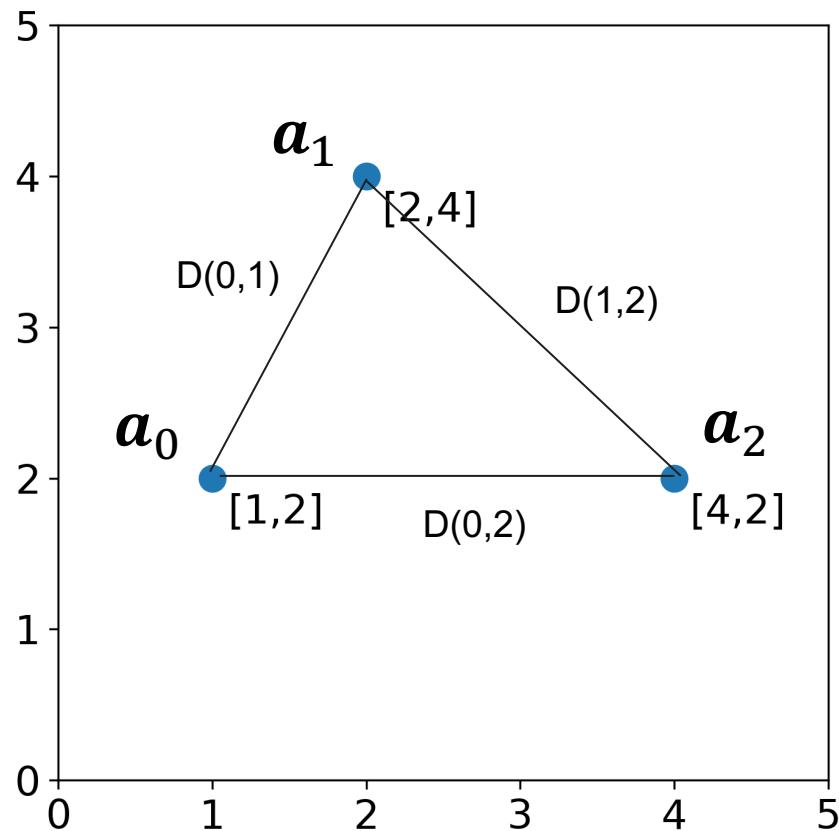
Triangle Inequality



Representing Data as a Graph



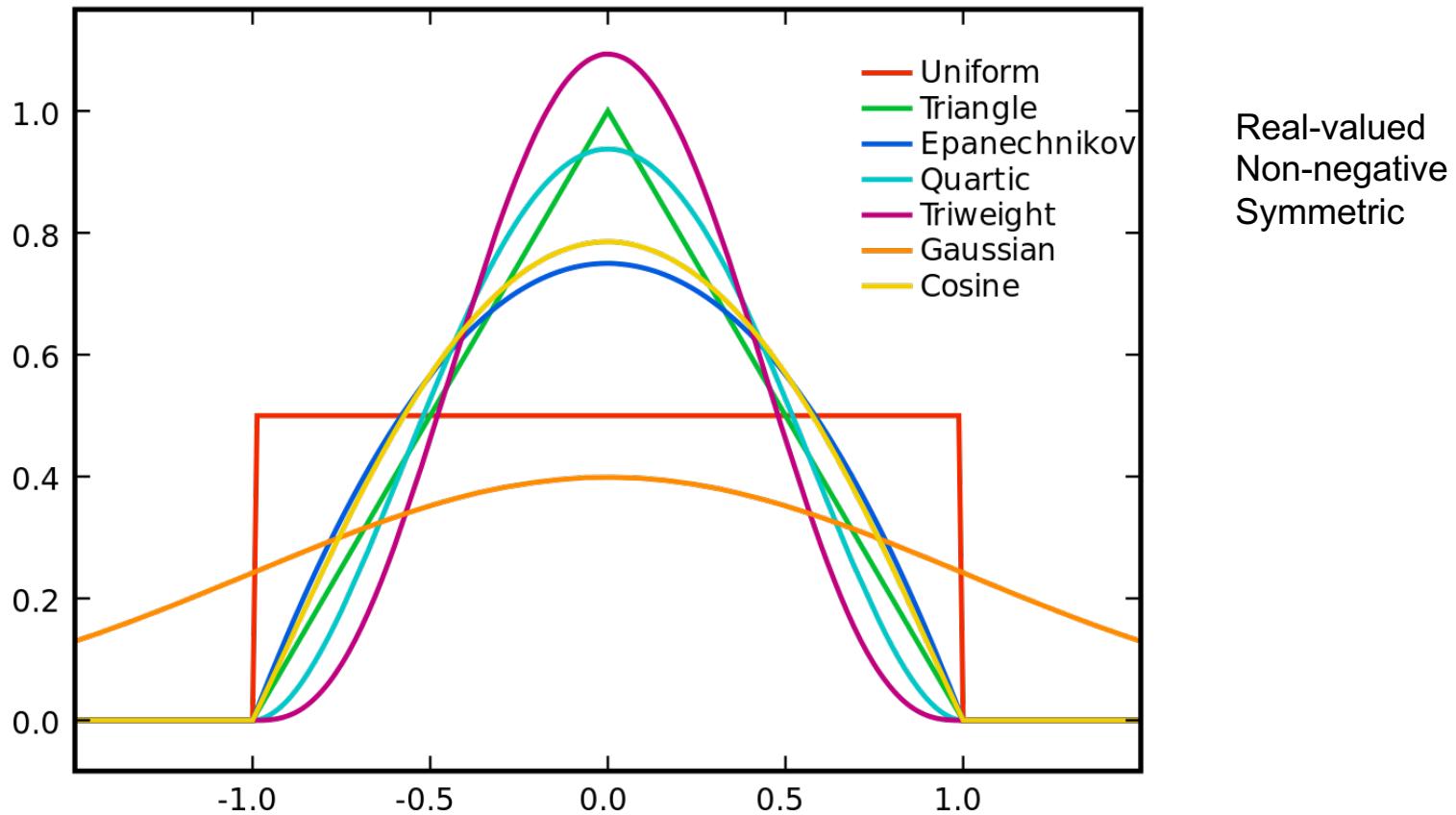
Fully connected Graph



a_0	a_1	a_2	
a_0	0	$\sqrt{5}$	3
a_1	$\sqrt{5}$	0	$\sqrt{8}$
a_2	3	$\sqrt{8}$	0

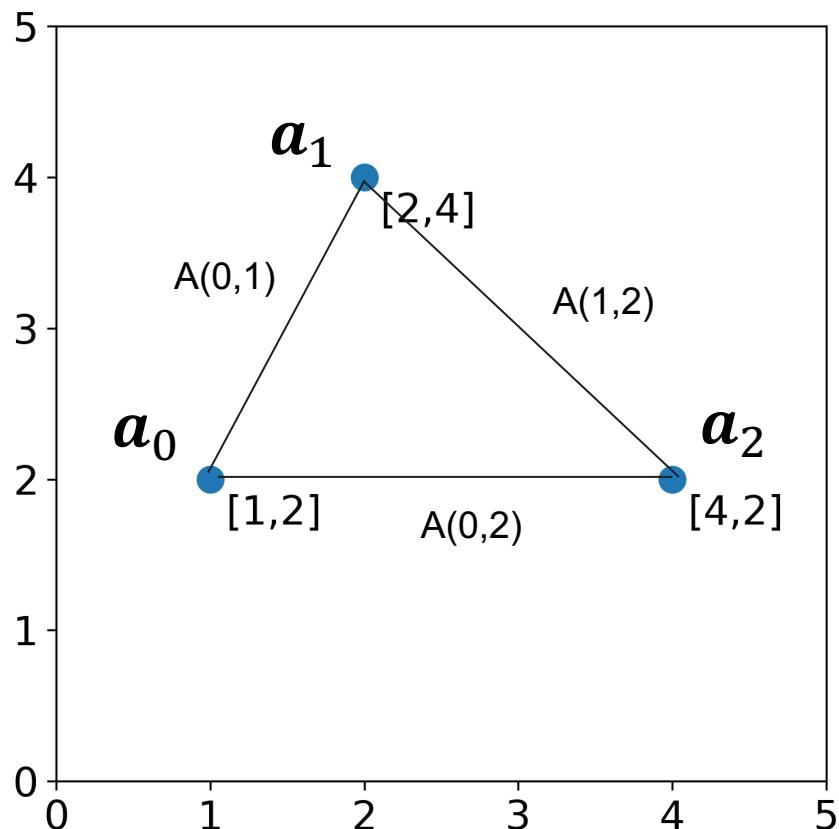
Distance matrix

Distance to Affinity via Kernels



Affinities correlations in this hidden hypothetical space

Affinities

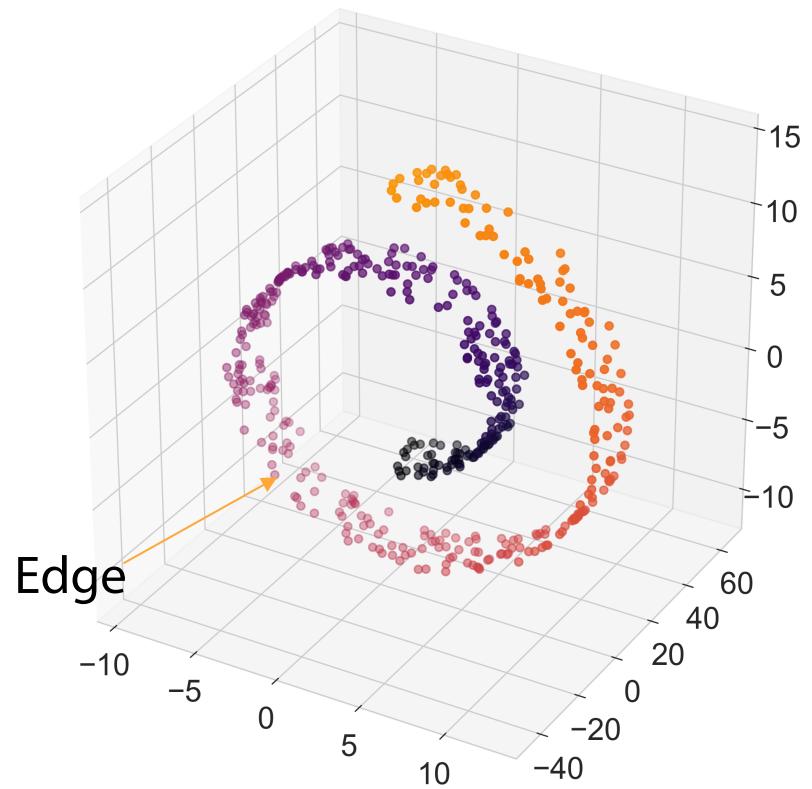


	a_0	a_1	a_2
a_0	1	0.032	0.0044
a_1	0.032	1	0.0075
a_2	0.0044	0.0075	1

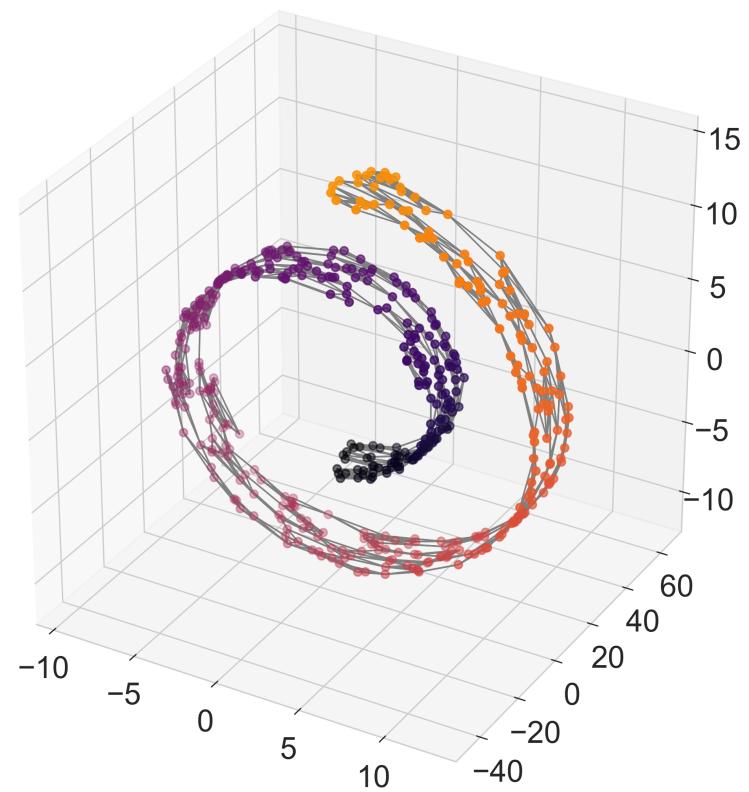
Affinity or Adjacency Matrix

Nearest Neighbors Graph

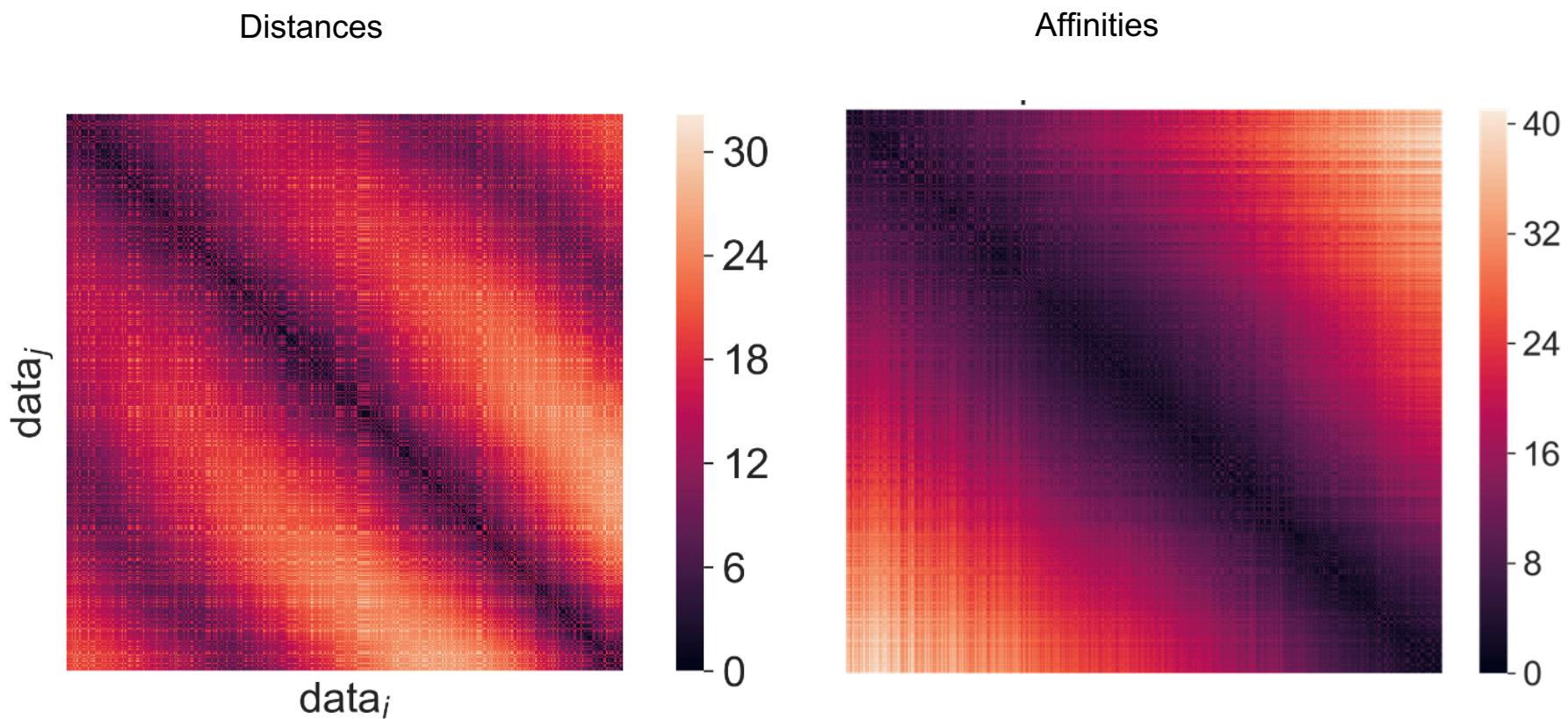
Data



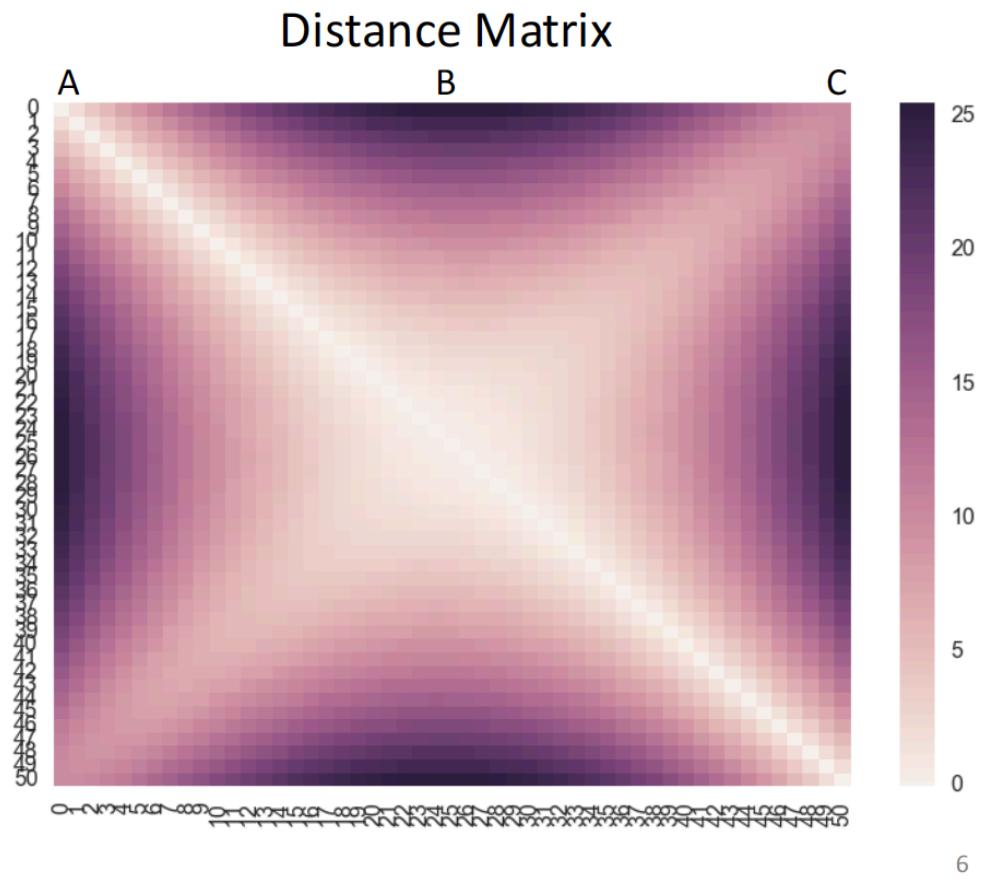
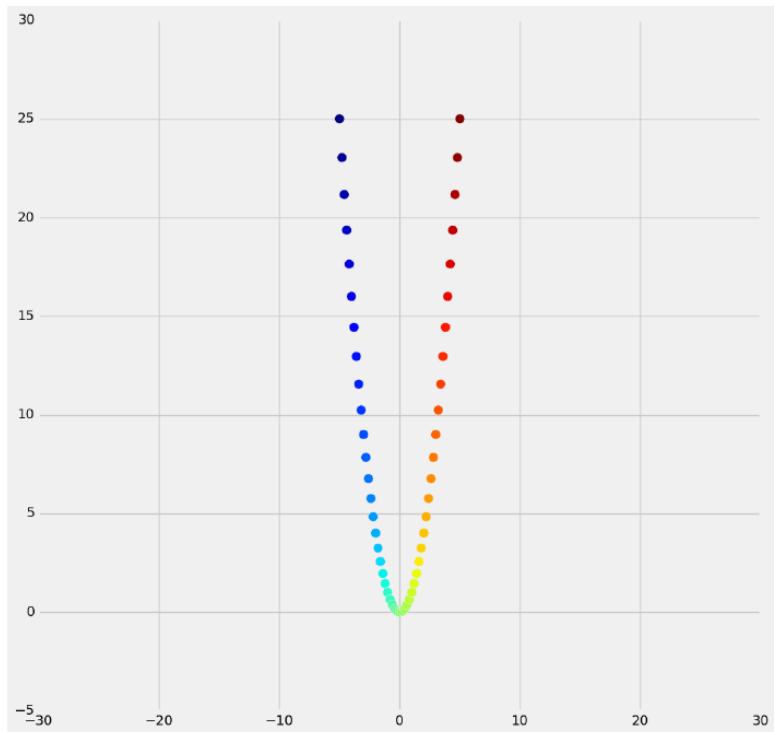
Nearest Neighbor Graph



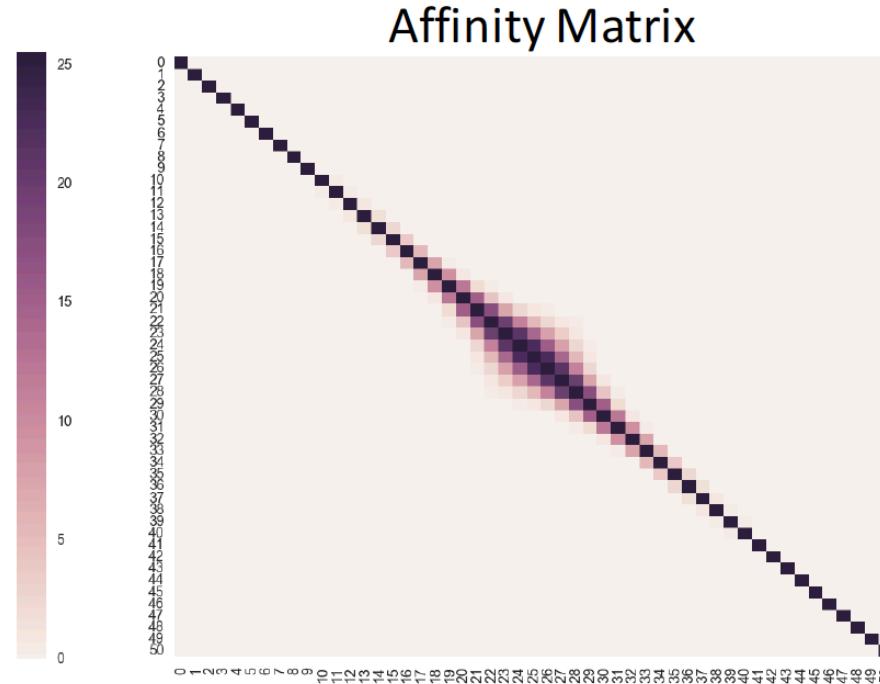
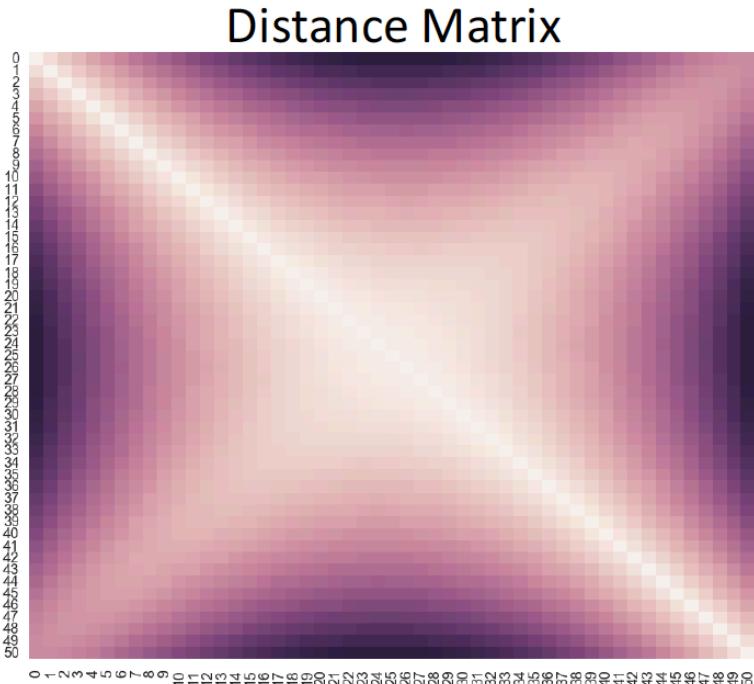
Swiss Roll



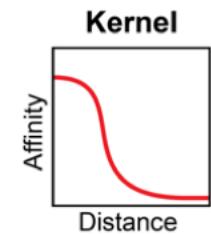
Example



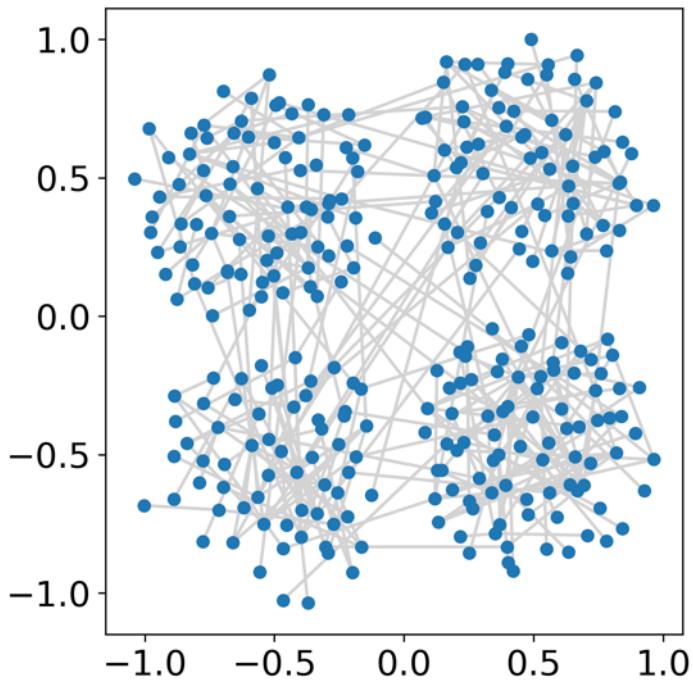
Affinity is the inverse proportional to distance



$$Affinity_{i,j} = s_{i,j} = \exp\left(-\frac{dist(x_i, x_j)^2}{2\sigma^2}\right)$$



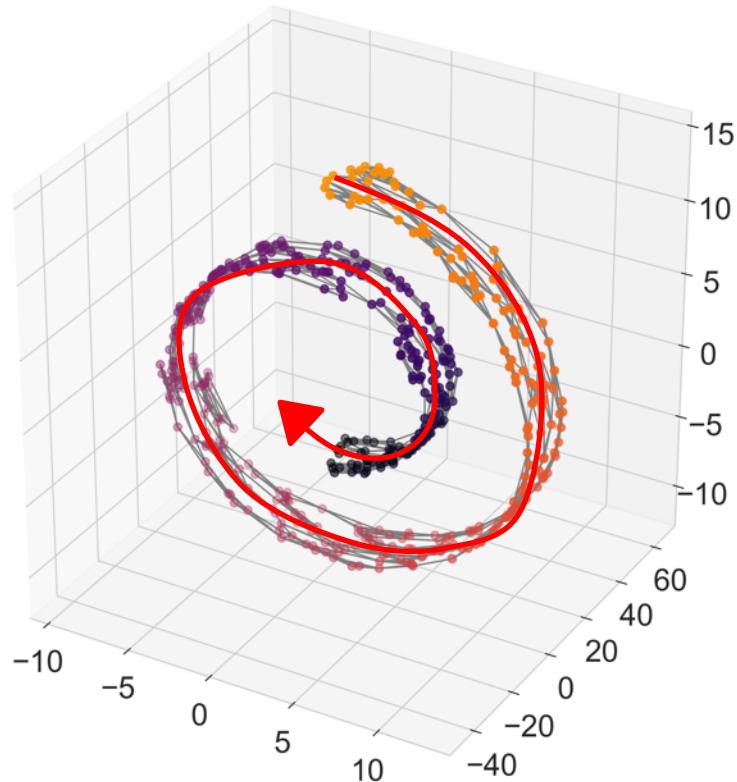
Why Represent Data as a Graph?



Graphs can be easy to cluster---look for a minimal way of “cutting edges” to form groups

Graphs avoid needing to operate in high dimensions
they can just be represented as list of edges

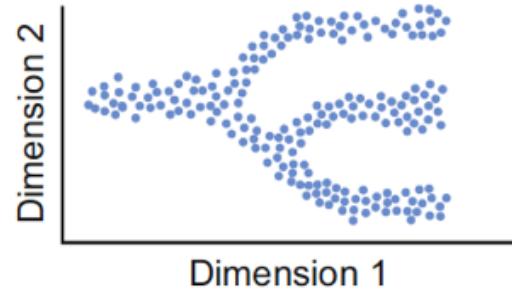
Why Represent Data as a Graph?



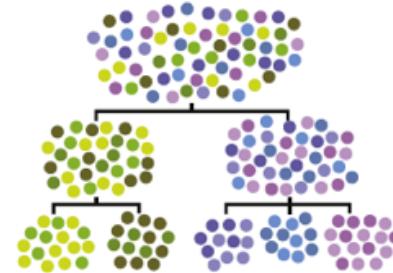
Paths through data graphs can represent progression trajectories

And much more to come!!

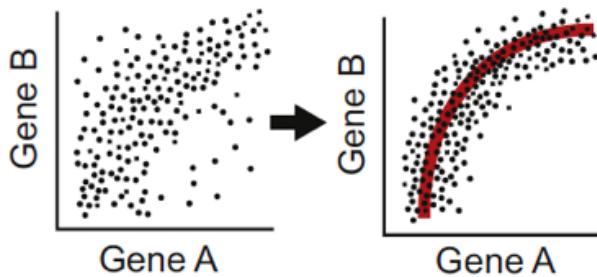
Vizualization



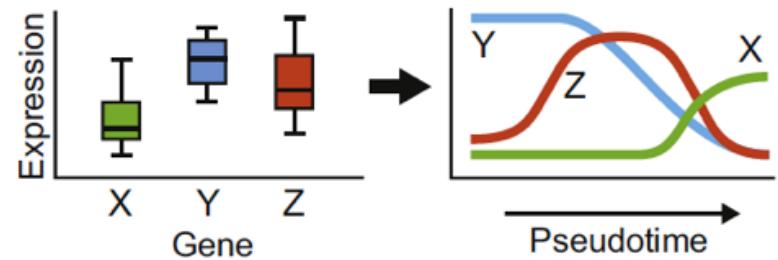
Clustering



Denoising



Pseudotime analysis



Preprocessing single-cell data

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

Malte D Luecken¹  & Fabian J Theis^{1,2,*} 

Abstract

Single-cell RNA-seq has enabled gene expression to be studied at an unprecedented resolution. The promise of this technology is attracting a growing user base for single-cell analysis methods. As more analysis tools are becoming available, it is becoming increasingly difficult to navigate this landscape and produce an up-to-date workflow to analyse one's data. Here, we detail the steps of a typical single-cell RNA-seq analysis, including pre-processing (quality control, normalization, data correction, feature selection, and dimensionality reduction) and cell- and gene-level downstream analysis. We formulate current best-practice recommendations for these steps based on independent comparison studies. We have integrated these best-practice recommendations into a workflow, which we apply to a public dataset to further illustrate how these steps work in practice. Our documented case study can be found at <https://www.github.com/theislab/single-cell-tutorial>. This review will serve as a workflow tutorial for new entrants into the field, and help established users update their analysis pipelines.

Keywords analysis pipeline development; computational biology; data analysis tutorial; single-cell RNA-seq

DOI 10.15252/msb.20188746 | Received 16 November 2018 | Revised 15 March 2019 | Accepted 3 April 2019

Mol Syst Biol. (2019) 15: e8746

Introduction

In recent years, single-cell RNA sequencing (scRNA-seq) has significantly advanced our knowledge of biological systems. We have been able to both study the cellular heterogeneity of zebrafish, frogs

outline current best practices to lay a foundation for future analysis standardization.

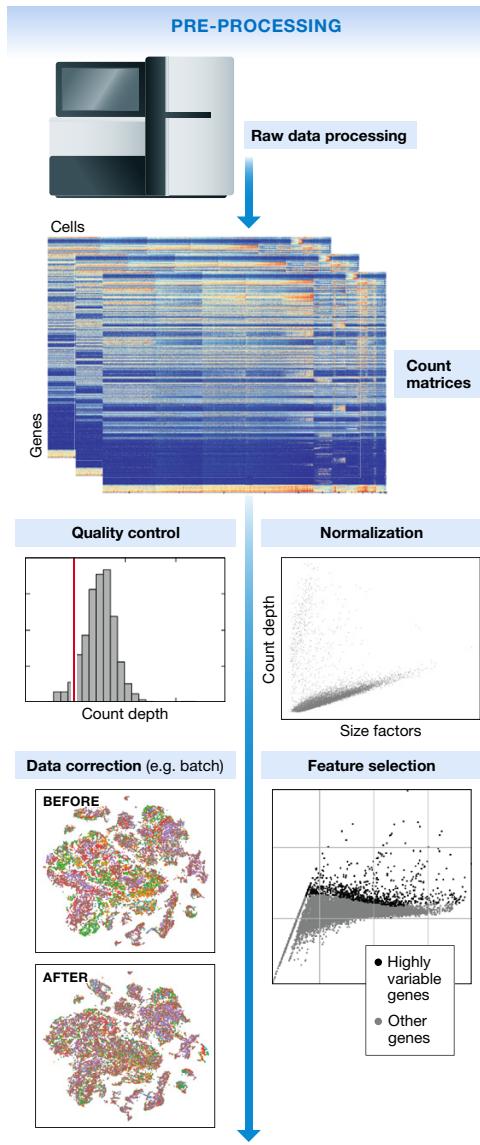
The challenges to standardization include the growing number of analysis methods (385 tools as of 7 March 2019) and exploding dataset sizes (Angerer *et al.*, 2017; Zappia *et al.*, 2018). We are continuously finding new ways to use the data at our disposal. For example, it has recently become possible to predict cell fates in differentiation (La Manno *et al.*, 2018). While the continuous improvement of analysis tools is beneficial for generating new scientific insight, it complicates standardization.

Further challenges for standardization lie in technical aspects. Analysis tools for scRNA-seq data are written in a variety of programming languages—most prominently R and Python (Zappia *et al.*, 2018). Although cross-environment support is growing (preprint: Scholz *et al.*, 2018), the choice of programming language is often also a choice between analysis tools. Popular platforms such as Seurat (Butler *et al.*, 2018), Scater (McCarthy *et al.*, 2017), or Scanpy (Wolf *et al.*, 2018) provide integrated environments to develop pipelines and contain large analysis toolboxes. However, out of necessity these platforms limit themselves to tools developed in their respective programming languages. By extension, language restrictions also hold true for currently available scRNA-seq analysis tutorials, many of which revolve around the above platforms (R and bioconductor tools: <https://github.com/drisso/bioc2016singlecell> and <https://hemberg-lab.github.io/scRNA.seq.course/>; Lun *et al.*, 2016b; Seurat: https://satijalab.org/seurat/get_started.html; Scanpy: <https://scanpy.readthedocs.io/en/stable/tutorials.html>).

Considering the above-mentioned challenges, instead of targeting a standardized analysis pipeline, we outline current best practices and common tools independent of programming language. We guide the reader through the various steps of a scRNA-seq analysis pipeline (Fig 1), present current best practices, and discuss analysis

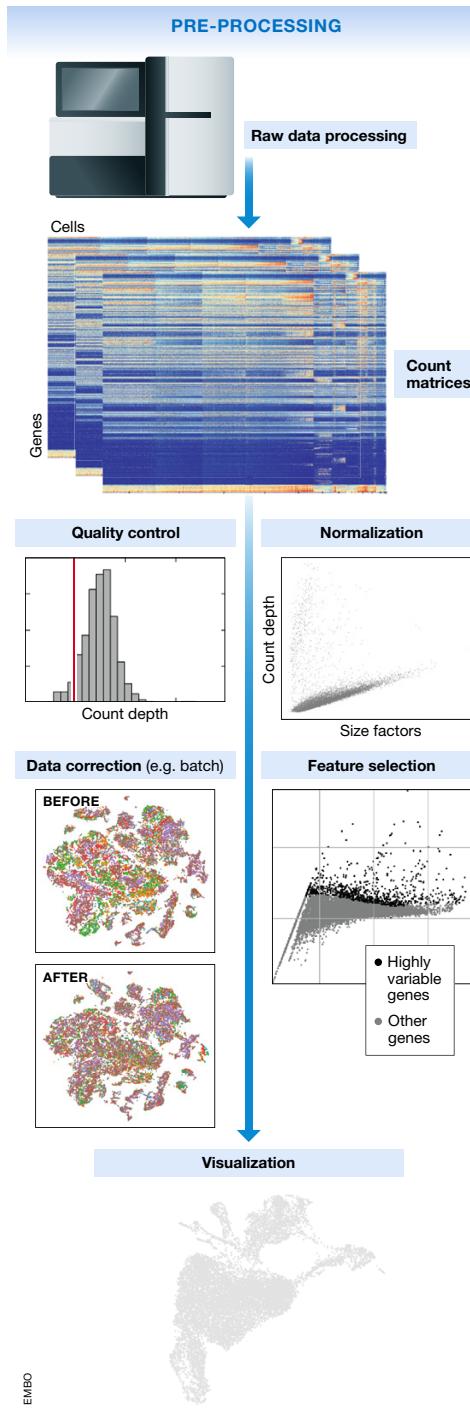
Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction



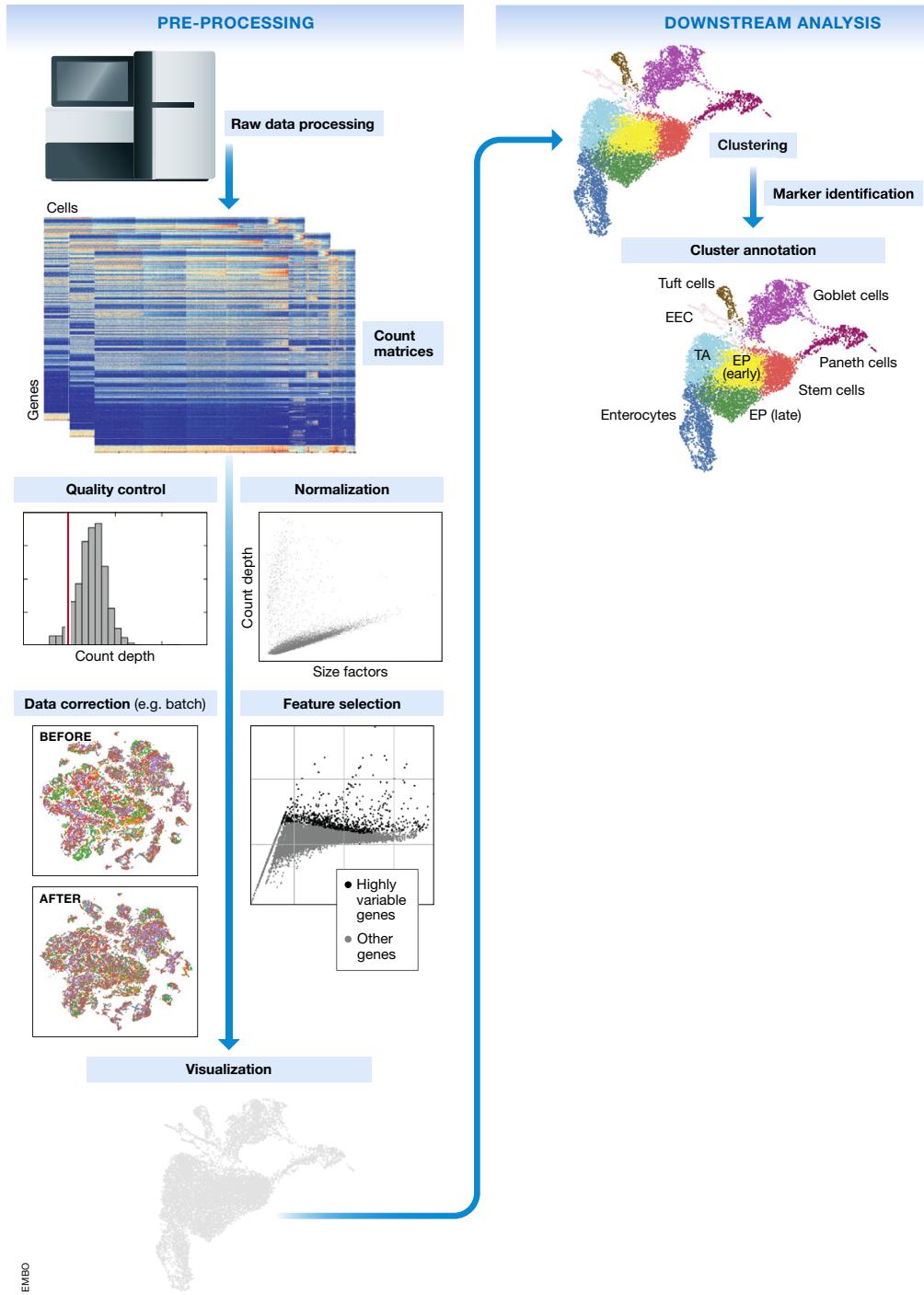
Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction
5. Dimensionality reduction and visualization



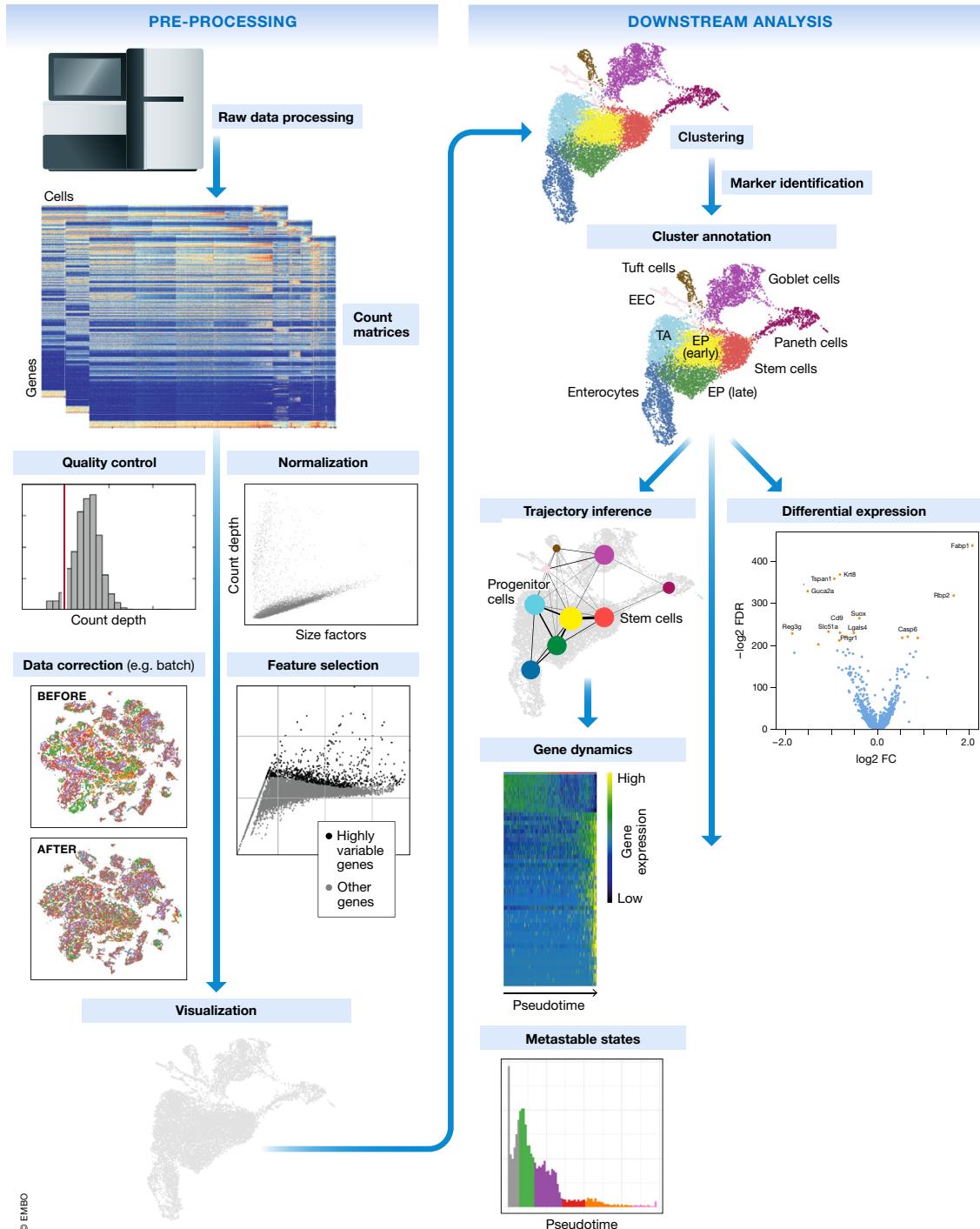
Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction
5. Dimensionality reduction and visualization
6. Downstream analysis
 1. Clustering
 2. Trajectory inference



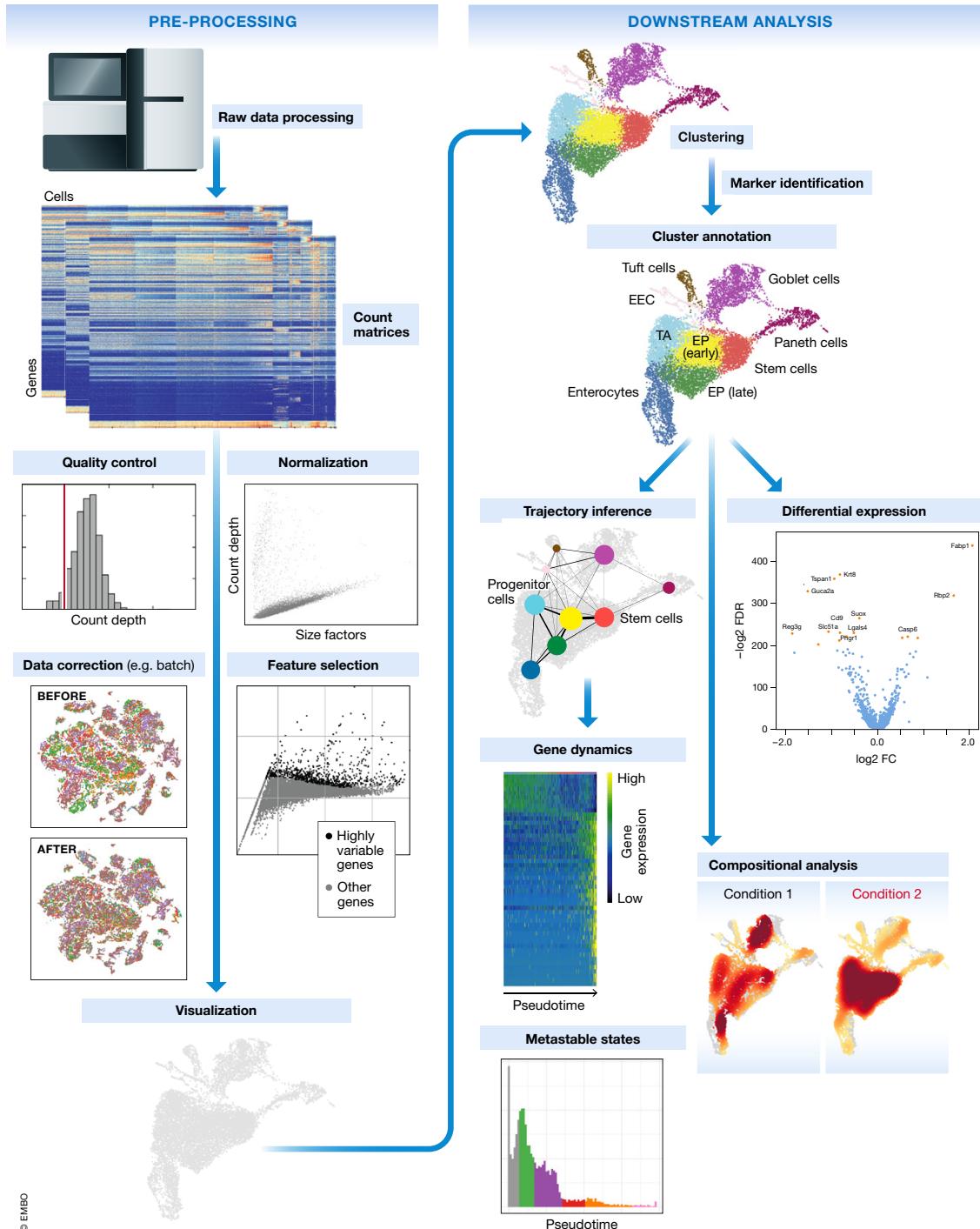
Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction
5. Dimensionality reduction and visualization
6. Downstream analysis
 1. Clustering
 2. Trajectory inference
 3. Differential expression

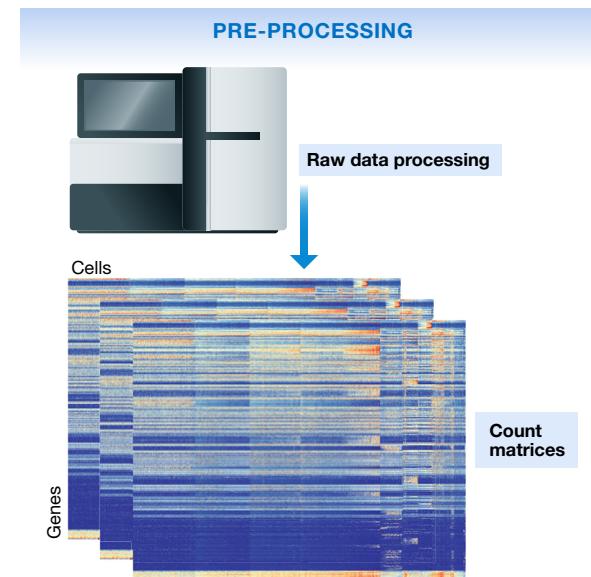
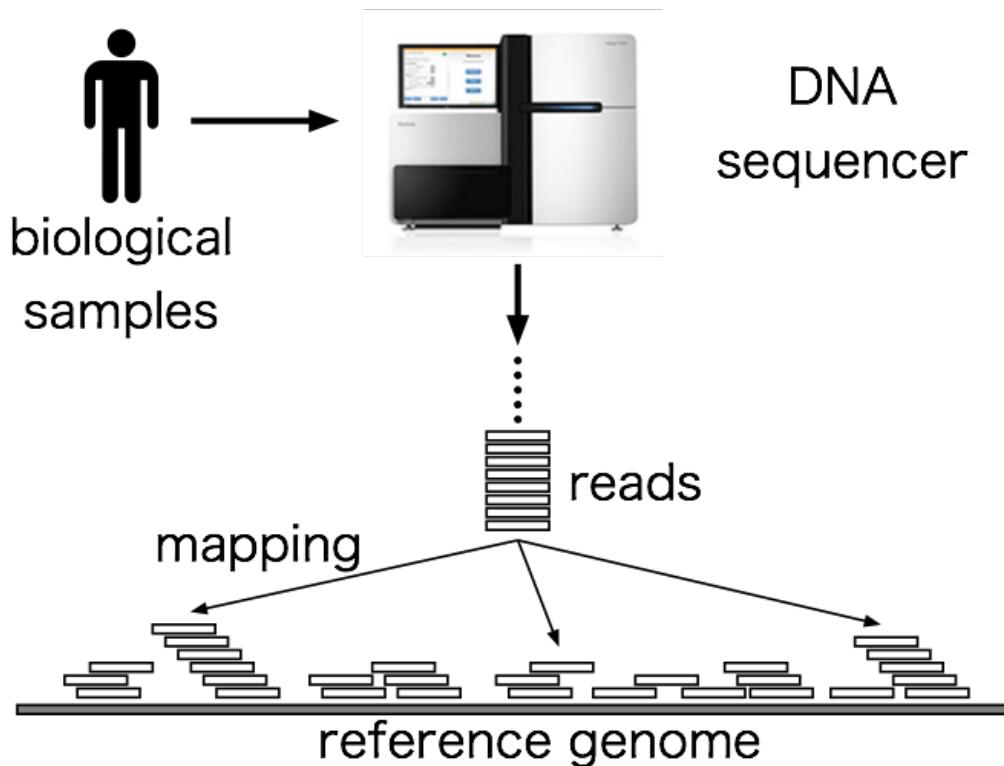


Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction
5. Dimensionality reduction and visualization
6. Downstream analysis
 1. Clustering
 2. Trajectory inference
 3. Differential expression
7. Comparison of multiple conditions



Step 1 - Sequencing & read mapping



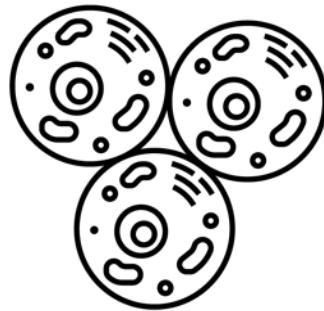
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering

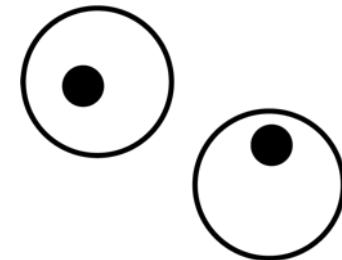
Dying cells



Multiplets



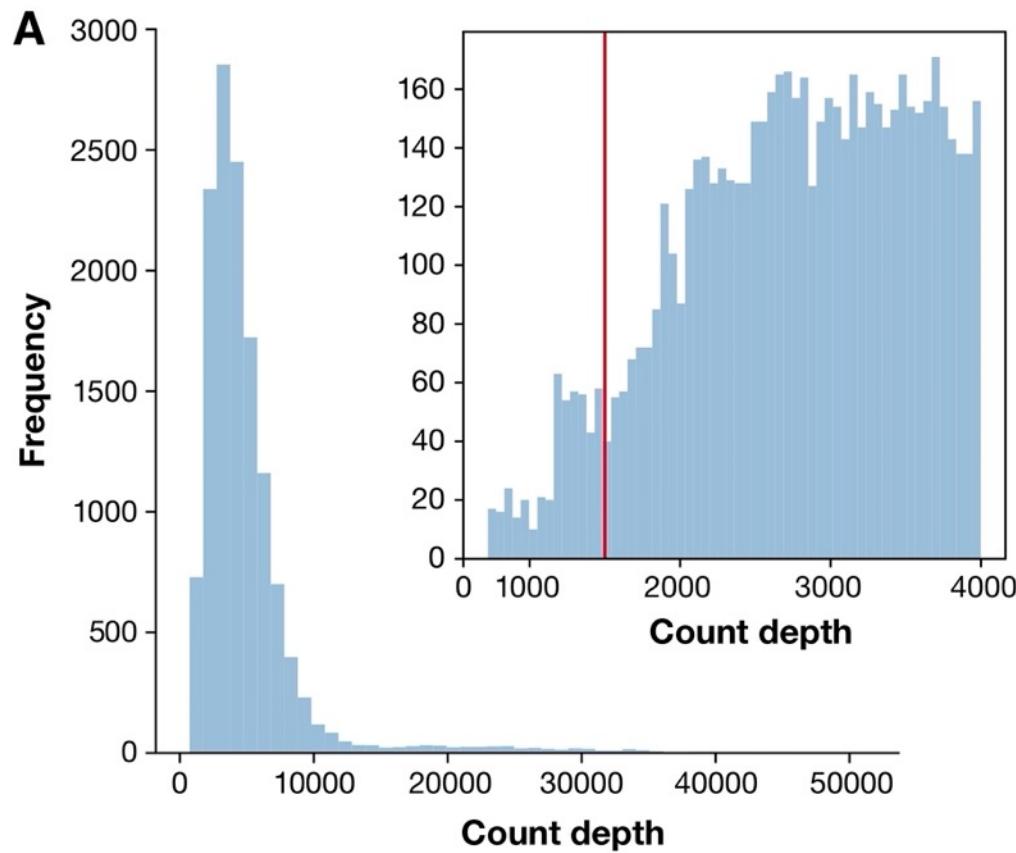
Empty Droplets



What could we look at to discriminate between dying cells, multiplets, or empty droplets and healthy single cells?

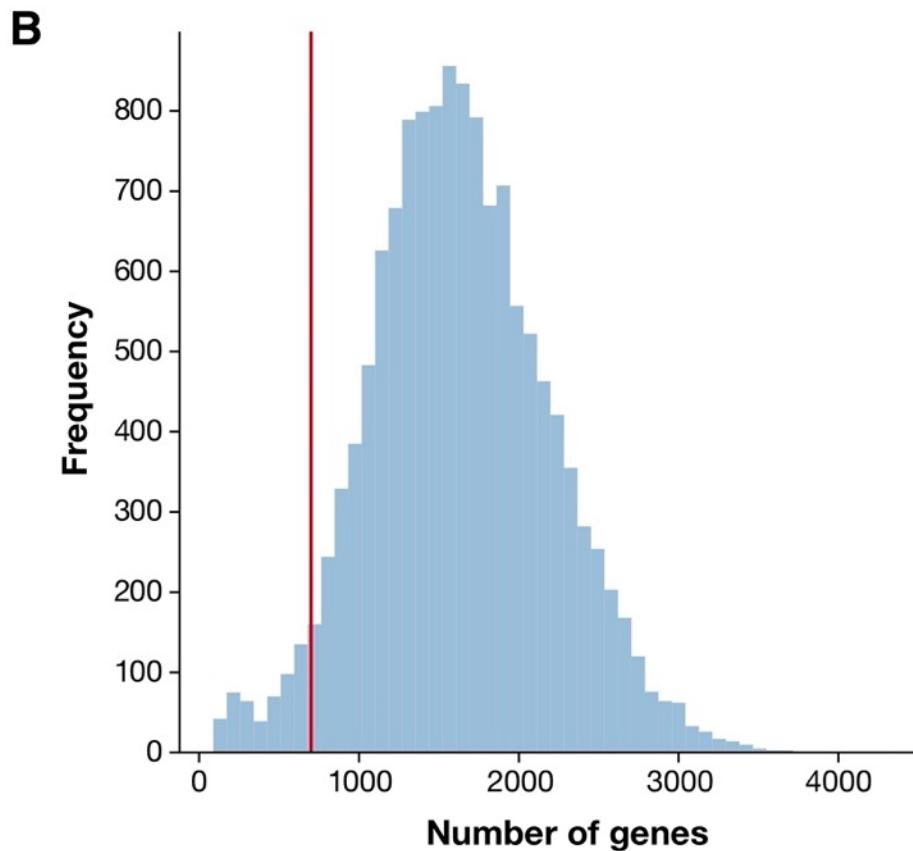
Top

Step 2 – Quality control and filtering



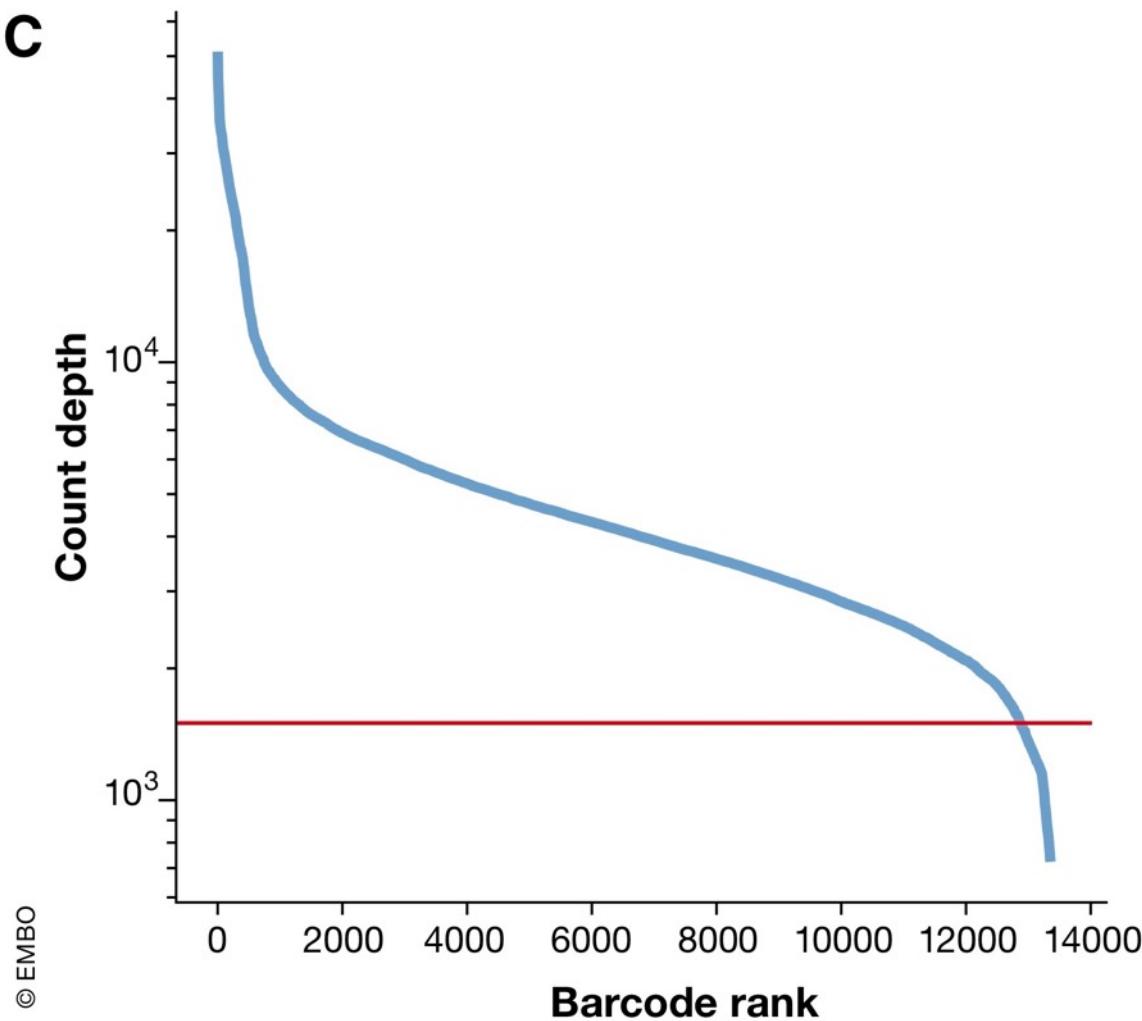
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



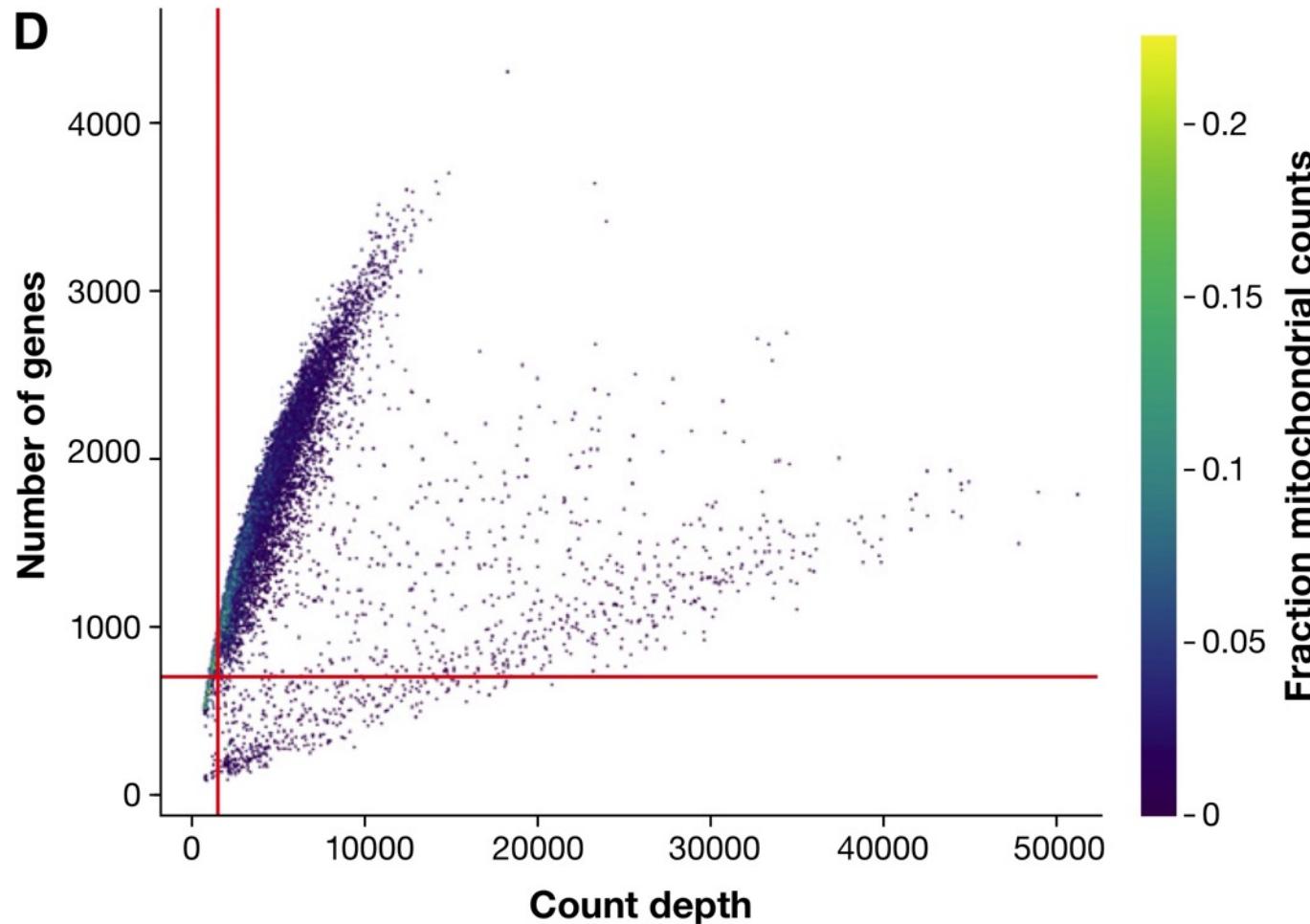
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



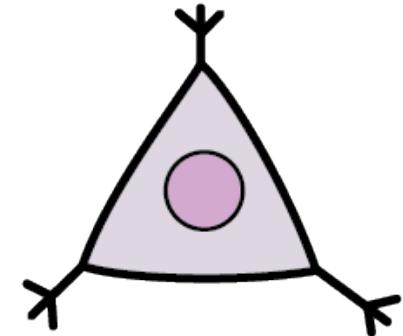
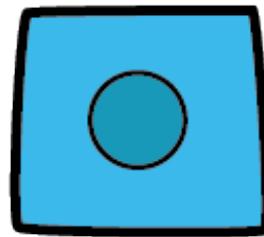
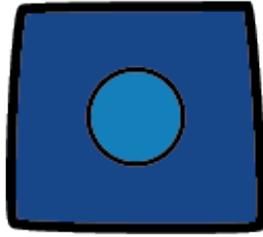
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



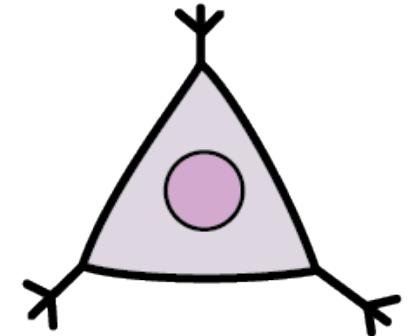
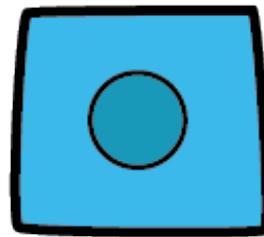
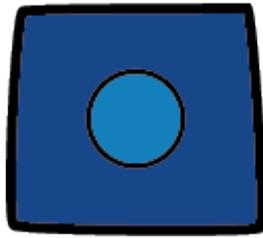
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 3 - Normalization



If we only have gene expression, how can we determine which cells are similar?

Step 3 - Normalization

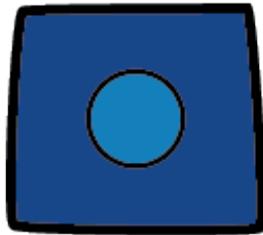


10% Capture Efficiency

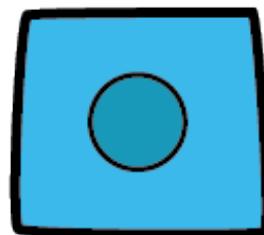
Gene	Cell A
X	10
Y	20
Z	70

Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

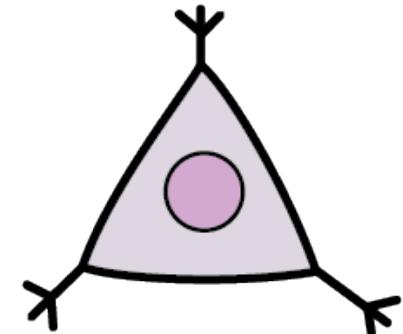
Step 3 - Normalization



10% Capture Efficiency



20% CE

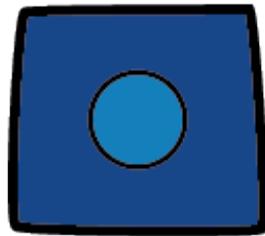


Gene	Cell A
X	10
Y	20
Z	70

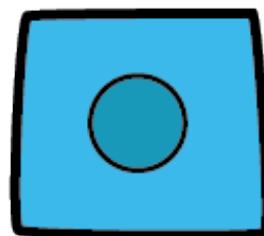
Gene	Cell B
X	20
Y	40
Z	140

Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

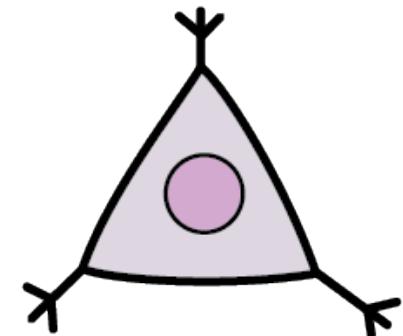
Step 3 - Normalization



10% Capture Efficiency



20% CE



20% CE

Gene	Cell A
X	10
Y	20
Z	70

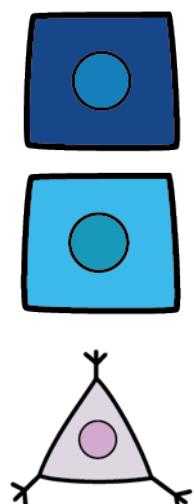
Gene	Cell B
X	20
Y	40
Z	140

Gene	Cell C
X	20
Y	0
Z	80

Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 3 - Normalization

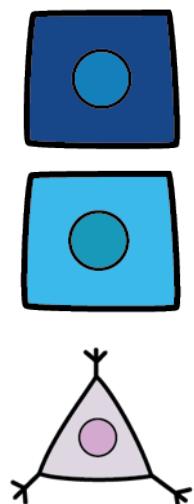
Raw counts



	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

Step 3 - Normalization

Raw counts

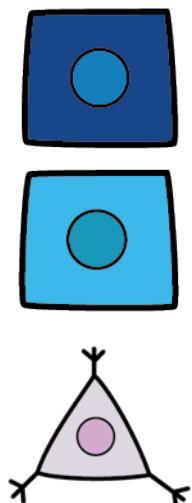


	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

Step 3 - Normalization

Raw counts



	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

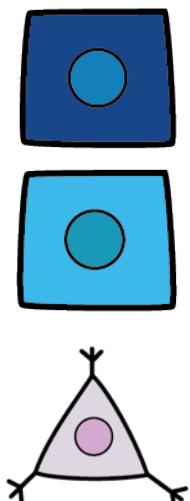
Pairwise distances

$$\text{dist}(A,B) = 71.4$$

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

Step 3 - Normalization

Raw counts



	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

Pairwise distances

$$\text{dist}(A,B) = 71.4$$

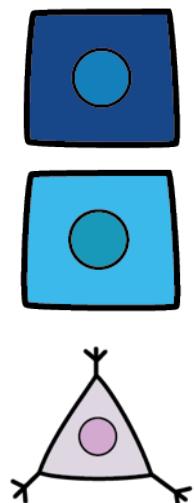
$$\text{dist}(A,C) = 24.5$$

$$\text{dist}(B,C) = 67.1$$

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

Step 3 - Normalization

Raw counts

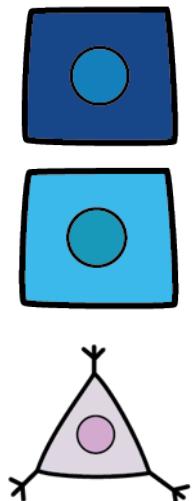


	X	Y	Z	Library Size	Pairwise distances
A	10	20	70	100	$\text{dist}(A,B) = 71.4$
B	20	40	140	200	$\text{dist}(A,C) = 24.5$
C	20	0	80	100	$\text{dist}(B,C) = 67.1$

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

Step 3 - Normalization

Normalized counts

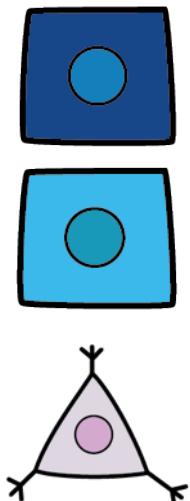


	X	Y	Z	Library Size	Pairwise distances
A	0.1	0.2	0.7	100	$\text{dist}(A,B) = 71.4$
B	0.1	0.2	0.7	200	$\text{dist}(A,C) = 24.5$
C	0.2	0	0.8	100	$\text{dist}(B,C) = 67.1$

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

Step 3 - Normalization

Normalized counts

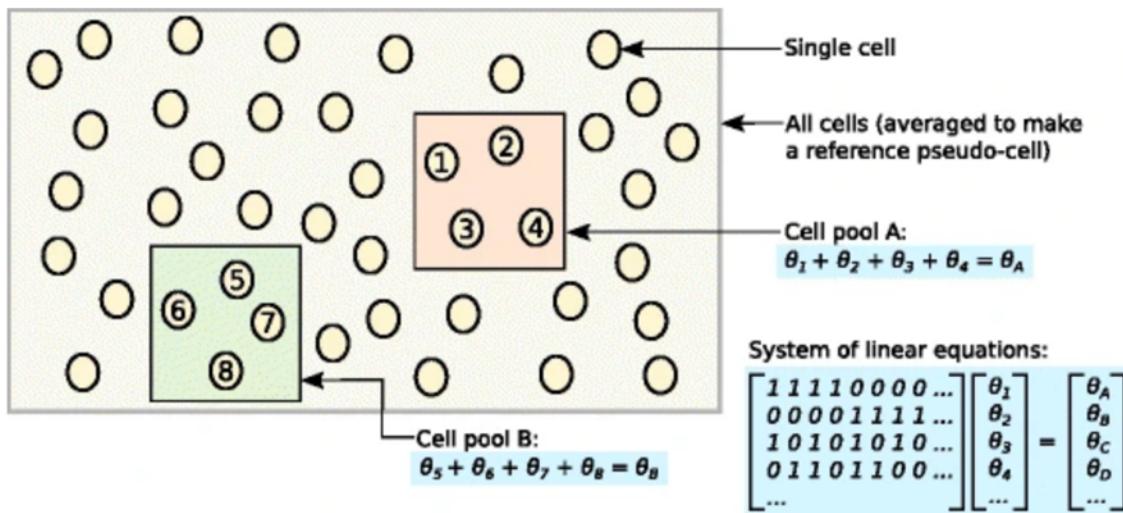


	X	Y	Z	Library Size	Pairwise distances
A	0.1	0.2	0.7	100	$\text{dist}(A,B) = 0$
B	0.1	0.2	0.7	200	$\text{dist}(A,C) = 0.25$
C	0.2	0	0.8	100	$\text{dist}(B,C) = 0.25$

$$\text{dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

More complex normalization approaches exist

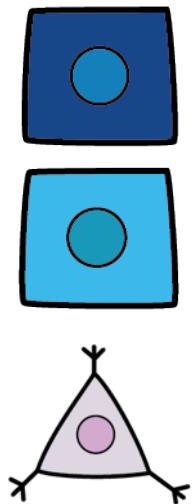
Fig. 3



Schematic of the deconvolution method. All cells in the data set are averaged to make a reference pseudo-cell. Expression values for cells in pool A are summed together and normalized against the reference to yield a pool-based size factor θ_A . This is equal to the sum of the cell-based factors θ_j for cells $j=1-4$ and can be used to formulate a linear equation. (For simplicity, the t_j term is assumed to be unity here.) Repeating this for multiple pools (e.g., pool B) leads to the construction of a linear system that can be solved to estimate θ_j for each cell j .

Step 3.5 – Transformation / Scaling

Normalized counts

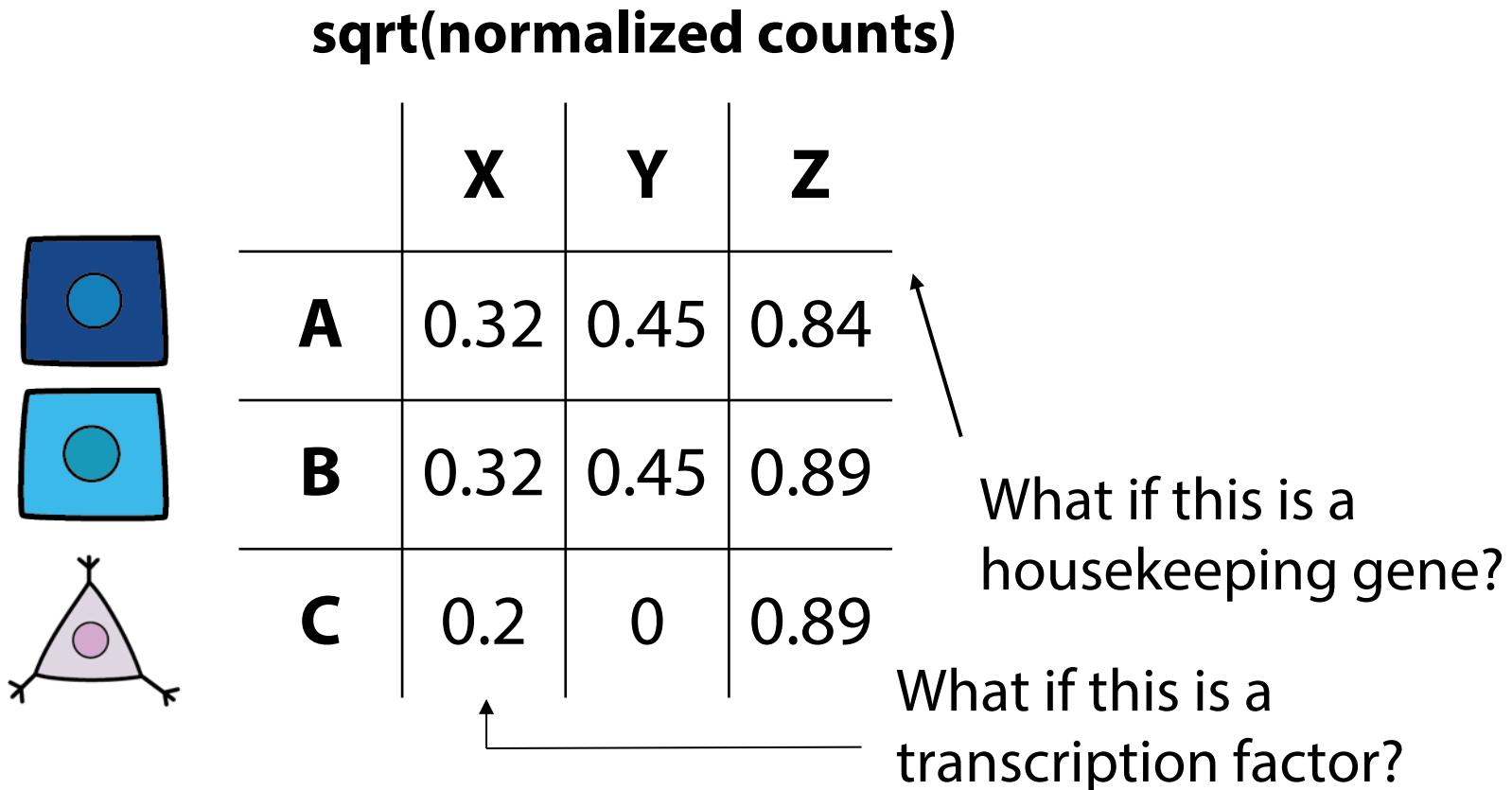


	X	Y	Z
A	0.1	0.2	0.7
B	0.1	0.2	0.7
C	0.2	0	0.8

What if this is a housekeeping gene?

What if this is a transcription factor?

Step 3.5 – Transformation / Scaling



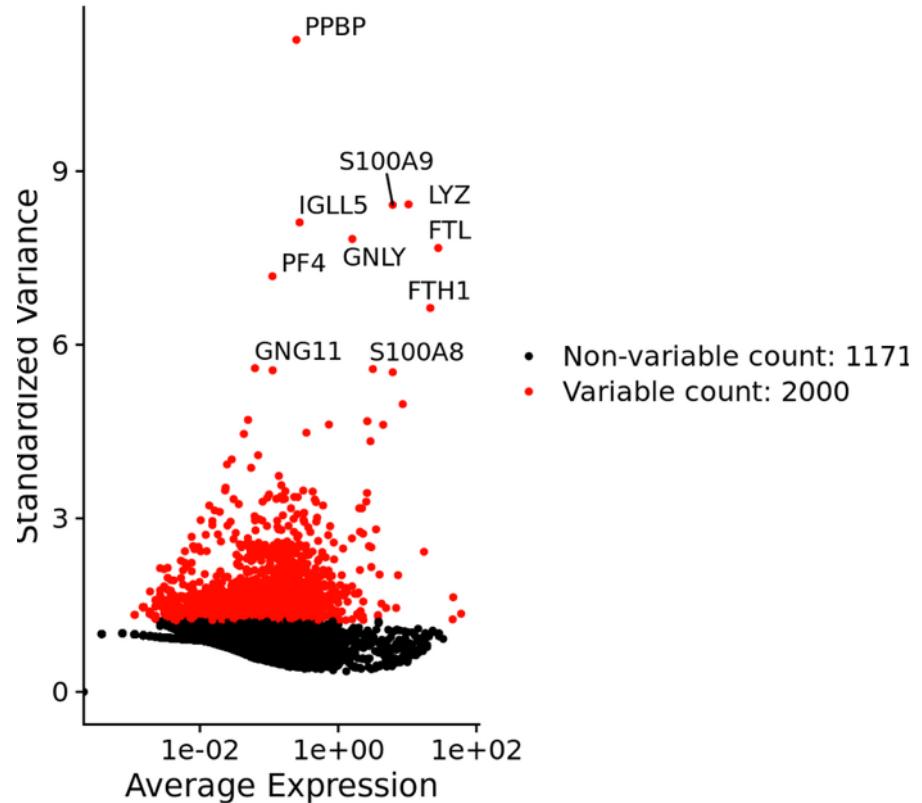
What kind of transformations, other than square-root, could we apply to single cell data?

Top

Step 5 – Dimensionality reduction and visualization

Selecting highly variable genes (HVGs):

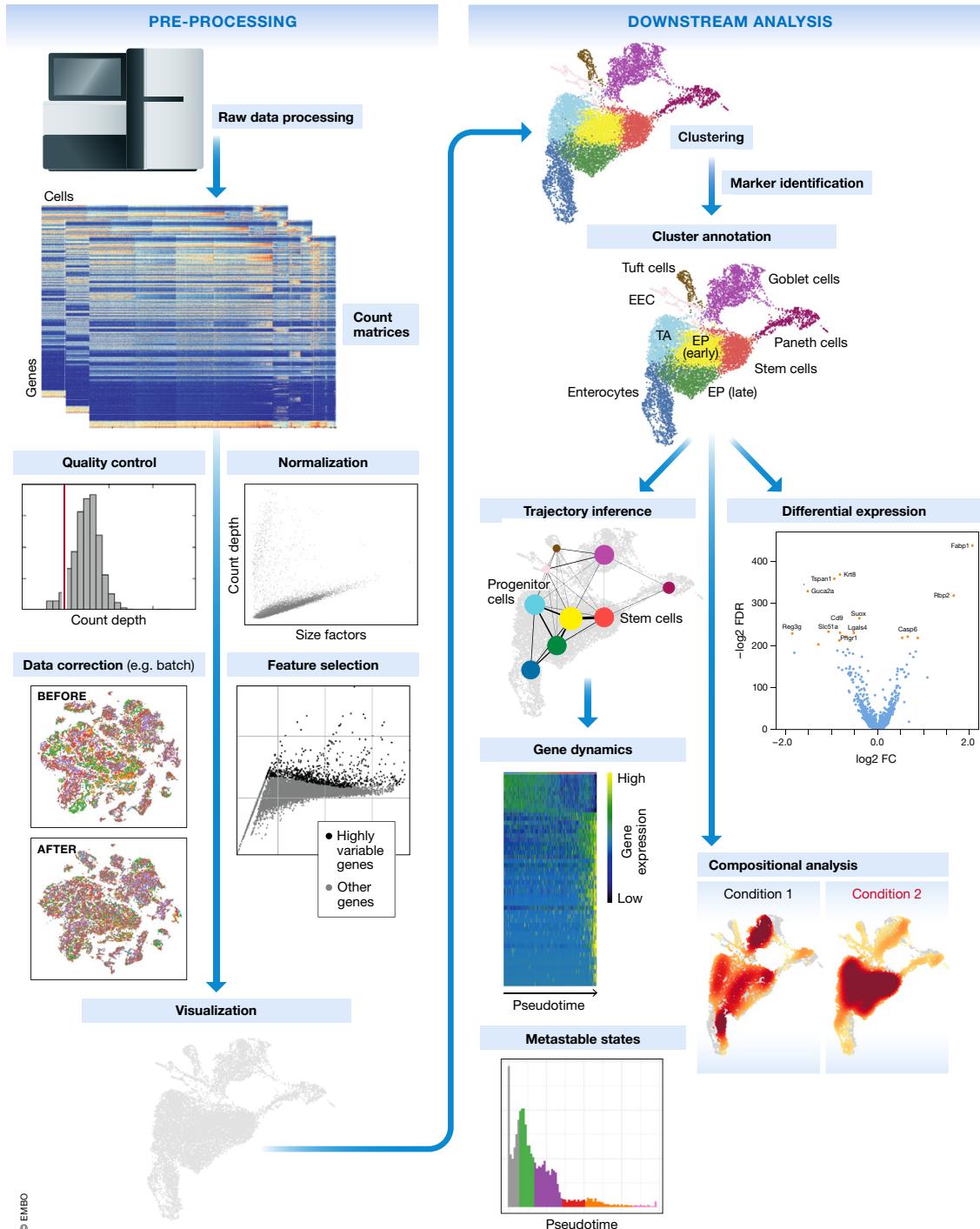
- Calculate log10 mean expression and variance
- Fit a loess curve
- Standardize variance to mean 0 std 1
- Take the top 2000 HVGs



Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Standard Single-Cell RNA-seq Workflow

1. Sequencing and read mapping
2. Quality control and filtering
3. Normalization
4. Data Correction
5. Dimensionality reduction and visualization
6. Downstream analysis
 1. Clustering
 2. Trajectory inference
 3. Differential expression
7. Comparison of multiple conditions



What questions do you have about today's material?

Top



Exercise!

Load, preprocess, and visualize a scRNAseq dataset generated from a time course of embryoid bodies

