

- ➡ 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



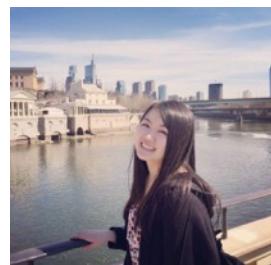
Andrew Benz



Egbert Castro



Pranik Chainani



Joanna Chen



Annie Gao



Michał Gerasimiuk



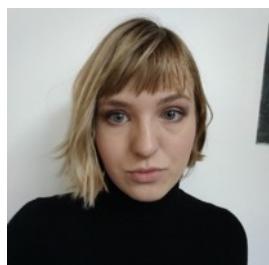
Wes Lewis



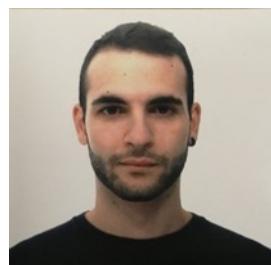
Francesc Lopez



Sameet Mehta



Sasha Safanova



Giacomo Scanavini



Alexander Tong



Aarthi Venkat

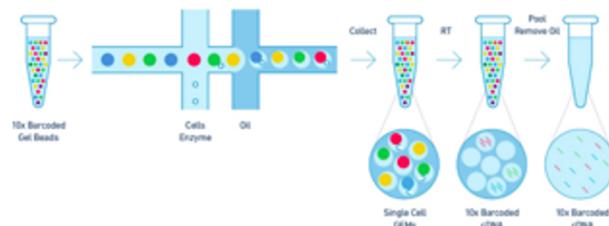


Max Yuan

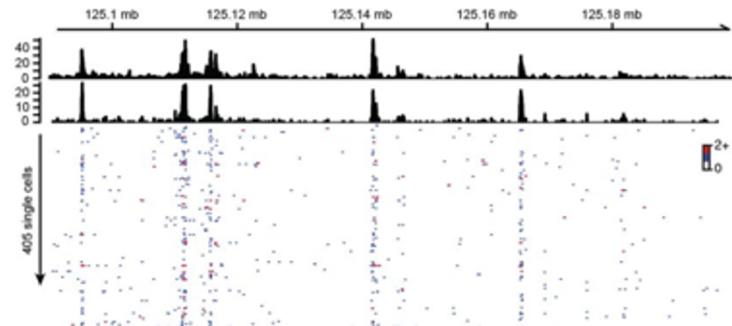


Jeffrey Zhou

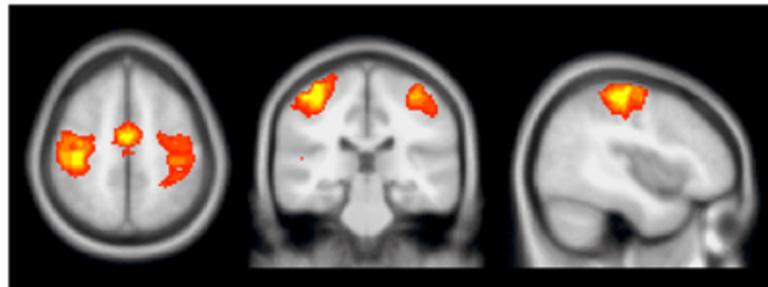
Big biomedical data



ScRNA-seq



ScATAC-seq



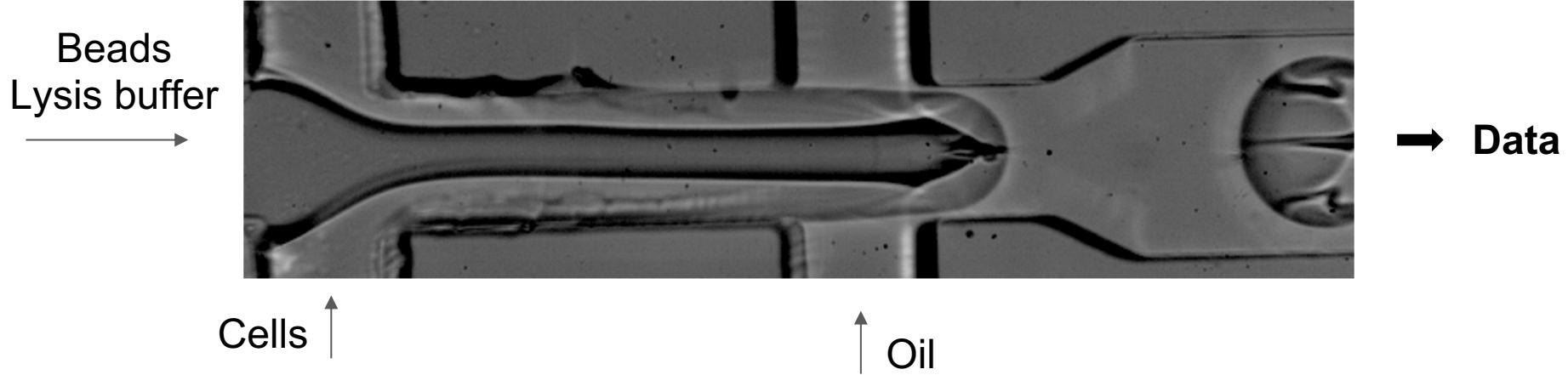
FMRI



Patient Data

Big = Any dataset with many many observations

The single cell revolution



The single cell revolution

Interesting Biological Experiments



Computation



High impact paper

LETTER

<https://doi.org/10.1038/nature08518>

RNA velocity of single cells

Giovanni La Mantia^{1,2}, Rüdiger Soldatow³, Anja Zelzer⁴, Emanuele Bozzani⁵, Hannah Hochegger^{6,7}, Viktor Překlýš^{8,9}, Kjetil Liebschuetz¹⁰, Maria E. Kavallaris¹¹, Peter Lønneberg¹², Alessandro Furlan¹³, Jean Fan¹⁴, Lars E. Børresen¹⁵, Zehua Liu¹⁶, Daniel C. Westover¹⁷, Michael A. Hickey¹⁸, Michael S. Hodge¹⁹, Michael S. Strickland²⁰, Gonzalo Gómez-Bravo²¹, Patrick Cramer²², Igor Adelman²³, Sten Linnemann^{24,25} & Peter V. Kharchenko^{1,2,26}

RNA abundance is a powerful indicator of the state of individual cells. Single-cell RNA sequencing can reveal RNA abundance with high quantitative accuracy, sensitivity and throughput. However, posing a challenge for the analysis of time resolved phenomena such as gene expression dynamics is the fact that the rate of RNA velocity—the time derivative of the gene expression state—can be inferred only by comparing the abundance of mRNA molecules at two time points. In common single-cell RNA sequencing protocols, RNA velocity is a high-dimensional vector that predicts the future state of the cell lineage. To validate this approach, we used RNA velocity to predict the neural crest lineage, demonstrate its use on multiple published datasets and show that it can be used to predict the differentiation of the developing mouse hippocampus, and examine the kinetics of gene expression dynamics in the mouse brain. Our results should greatly aid the analysis of developmental lineage and cellular heterogeneity, 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 timescale of RNA velocity is therefore much longer than that which can be exploited to estimate the rates of gene splicing and degradation, and thus to predict the future state of the cell lineage with confidence. We measured that similar signals may be detectable in single-cell RNA sequencing data, and used them to predict the future structure of the entire transcriptome during dynamic processes.

All common single-cell RNA sequencing protocols rely on cDNA-IT priming to generate complementary DNA (cDNA) for sequencing. Using single-cell RNA sequencing based on the SMART-seq², STRT³ and Dropbead⁴ methods, we found that ~1–15% of reads contained unique simple intronic sequences (Fig. 1a), in approximately 1–20% of genes. Most such reads originated from secondary priming sites (Fig. 1b). By contrast, when we used the NEBNext⁵ and Genomics Chromium libraries, we also found abundant discordant priming from the 5' termini of transcripts, poly-A⁺ sequences (Fig. 1c), which are usually removed by exonuclease treatment before amplification by priming on the first strand cDNA. The substantial fraction of discordant priming in these samples, and the corresponding counts suggest that these molecules represent unspliced precursor RNA followed by RNA sequencing using single-dT primed cDNA (Fig. 1d). During development, a substantial proportion of genes have simple first-order kinetics, as expected if unspliced mRNAs represent most of the transcriptome.

To quantify the time-dependent relationship between the abundance of precursor and mature mRNA, we assumed a simple model

for transcriptional dynamics⁶, in which the first time derivative of the spliced mRNA abundance (RNA velocity) is determined by the balance between the production of new mRNA from pre-mRNA, and the degradation of existing mRNA (Supplementary Fig. 1). Under such a model, steady states are approached asymptotically when the rate of gene expression (α) is constant, and the rate of mRNA degradation (β) is constant. If degraded mRNA is rapidly recycled, the rate of degraded mRNA (γ) is equal to the rate of degradation of spliced mRNA.

The equilibrium steps γ (conversion degradation and splicing rates, capturing gene specific regulatory properties, the rate of intrinsic gene expression and the rate of mRNA degradation) are unknown. As recently published computations of mouse tissues⁷, we found that the rate of gene expression (α) was constant across all genes, and the rate of mRNA degradation (β) was consistent with a single fixed step (Extended Data Fig. 1c). The rate of conversion of unspliced mRNA to spliced mRNA (γ) was also found to be constant (Extended Data Fig. 1c), suggesting tissue specific alternative splicing rates.

During a dynamic process, an increase in the transcription rate α results in a rapid increase in unspliced mRNA, followed by a subsequent decrease in spliced mRNA (Fig. 2a, b, Extended Data Fig. 2a, b). Until a new steady state is reached, Conversely, a drop in the transcription rate α results in a rapid decrease in unspliced mRNA, followed by a reduction in spliced mRNA. During induction of gene expression, an increase in the transcription rate α results in an increase in the abundance of unspliced mRNA, and a decrease in the abundance of spliced mRNA, until the new steady state is reached.

Given that the timescale of RNA velocity is much longer than that of mRNA abundance, and thus the future state of the cell can be extrapolated from the current state, we used the model to extrapolate the mature mRNA abundance into the future, we examined a time course of gene expression dynamics in the developing mouse hippocampus⁸. Consistent with mRNA levels at each time point were consistently observed in the hippocampus (Fig. 2c, d), and the RNA velocity was consistent with the expected direction of progression of the circuit cycle (Fig. 2e). The RNA velocity of the hippocampus was also consistent with the expected direction of progression of the circuit cycle (Fig. 2f). To validate the RNA velocity of the hippocampus, we performed RNA sequencing of single-cell measurements, we analyzed recently published single-cell RNA sequencing data from the hippocampus of the mouse⁹ (Extended Data Fig. 2g). During development, a substantial proportion of cells in the hippocampus had simple first-order kinetics, as expected if unspliced mRNAs represent most of the transcriptome. This provides a way to validate the direction of differentiation in which the direction of differentiation can be validated by lineage tracing. Phase portraits of many genes showed the expected deviation

© 2014 Macmillan Publishers Limited. All rights reserved. Printed in the UK.

1. Department of Molecular Neurobiology, Institute of Medical Biotechnology and Bioinformatics, Karlsruhe Institute of Technology, Germany. 2. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 3. Department of Biochemistry and Nutrition, Karlsruhe Institute of Technology, Germany. 4. Department of Pharmacology and Pharmacogenomics, Karlsruhe Institute of Technology, Germany. 5. Life Sciences Division, The Jackson Laboratory, Bar Harbor, ME, USA. 6. Department of Cell Biology, University of Cambridge, Cambridge, UK. 7. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 8. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 9. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 10. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 11. Department of Cell Biology, University of Cambridge, Cambridge, UK. 12. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 13. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 14. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 15. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 16. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 17. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 18. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 19. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 20. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 21. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 22. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 23. Department of Cell Biology, University of Cambridge, Cambridge, UK. 24. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 25. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK. 26. Institute for Cell and Molecular Medicine, University of Edinburgh, Edinburgh, UK.

494 | NATURE | VOL 508 | 22 AUGUST 2014

The single cell revolution

Interesting Biological Experiments



Computation



High impact paper

LETTER

<https://doi.org/10.1038/nature208-018-00184>

RNA velocity of single cells

Giovanni La Manno^{1,2}, Rüdiger Soldatov³, Anja Zelena⁴, Emanuele Bracci^{5,6}, Hannah Hochegger^{7,8}, Viktor Pevsner^{9,10}, Katja Lischwege¹⁰, Maria E. Kavallaris¹⁰, Peter Lennertzberg¹⁰, Alessandro Fornai¹⁰, Jean Fan¹⁰, Lars E. Boerig¹⁰, Zehan Liu¹⁰, Michaela Schäfer¹⁰, Barbara Städler¹⁰, Daniel Städler¹⁰, Stephan Schneider¹⁰, Gonzalo Gómez-Bravo¹⁰, Patrick Cramer¹⁰, Igor Adelman¹⁰, Sten Linnemann¹⁰ & Peter V. Kharchenko^{1,2,10}

RNA abundance is a powerful indicator of the state of individual cells. Single-cell RNA sequencing can reveal RNA abundance with high quantitative accuracy, sensitivity and throughput. However, posing a challenge for the analysis of time resolved phenomena such as gene expression dynamics is the fact that the rate of RNA velocity—the time derivative of the gene expression state—can be measured only by comparing the abundance of mRNA molecules at two time points. In common single-cell RNA sequencing protocols, RNA velocity is a high-dimensional vector that predicts the future state of a cell based on its current transcriptome. To validate this model, we first used it to analyze the gene expression dynamics of the neural crest lineage, demonstrating its use on multiple published datasets. We then used RNA velocity to study the gene expression dynamics of the developing mouse hippocampus, and examine the kinetics of gene expression changes during differentiation. Our results will greatly aid the analysis of developmental lineage and cellular dynamics, particularly in humans.

During development, gene expression occurs on a timescale of hours to days, which is comparable to the typical half-life of mRNA. The timescale of RNA velocity is therefore much longer, and can be exploited to estimate the rates of gene splicing and degradation, as well as the rates of gene expression changes during lineage progression. We measured that similar signals may be detectable in single-cell RNA sequencing data, and used them to predict the time course of change of the entire transcriptome during dynamic processes.

All common single-cell RNA sequencing protocols rely on cDNA-IT priming to reduce the bias introduced by the use of different sequencing platforms. We found that cDNA-IT priming is also effective for sequencing single-cell RNA seq data sets based on the SMART-seq², STRET³, and Dropbead⁴ platforms. Using cDNA-IT priming, we found that ~15–22% of reads contained simple intrinsic sequences (Fig. 1a), in agreement with previous reports^{5,6}. The remaining ~78–85% of RNA seq. Most such reads originated from secondary priming sites, and were found to contain complex intrinsic sequences. Using the Genomics Chromatin Library, we also found abundant discordant priming from the commonly occurring intrinsic poly-A⁺ sequences (Fig. 1b). We used these reads to validate the quality of cDNA-IT amplification by priming on the first strand cDNA. The substantial fraction of reads containing simple intrinsic sequences in our data sets suggest that these molecules represent unspliced precursor RNA followed by RNA sequencing using single-dT primed cDNA (Fig. 1c). During development, a substantial proportion of genes displayed expression dynamics, as expected if uncoupled reads represent different genes.

To quantify the time-dependent relationship between the abundance of precursor and mature mRNA, we assumed a simple model

for transcriptional dynamics⁷, in which the first time derivative of the spliced mRNA abundance (RNA velocity) is determined by the balance between production of new mRNA from transcription (rate α), and degradation of existing mRNA (rate β) (Extended Data Fig. 1a,b). Under this model, steady states are approached asymptotically when the rate of production of new mRNA equals the rate of degradation of existing mRNA ($\alpha = \beta$). The time constant of this approach is $\tau = \ln(2)/(\alpha - \beta)$.

The equilibrium state \bar{m} contains degradation and splicing rates, capturing gene specific regulatory properties, the rate of intrinsic transcription and the rate of degradation of mRNA (Fig. 1d). Using a recently published compilation of mouse tissues⁸, we found that the rate of production of new mRNA was approximately constant across all tissues, while the rate of degradation of mRNA was tissue specific (Fig. 1e). The equilibrium state \bar{m} was consistent with a single fixed state (Extended Data Fig. 1c–f). The time constant of this approach was $\tau = 1.5$ days, and the steady state of mRNA abundance was $\bar{m} = 0.001$ (Extended Data Fig. 1g). This suggests that the time constant of mRNA abundance is ~ 1 day.

During a dynamic process, an increase in the transcription rate results in a rapid increase in single-cell mRNA, followed by a subsequent decrease (Fig. 1g). The time course of mRNA abundance is shown in Extended Data Fig. 1h. We found that the time constant of mRNA abundance is ~ 1 day (Fig. 1g), until a new steady state is reached. Conversely, a drop in the transcription rate leads to a rapid decrease in mRNA abundance, followed by a reduction in spliced mRNA. During induction of gene expression, an increase in the transcription rate results in an increase in the equilibrium state \bar{m} , whereas the opposite is true during repression (Fig. 1g). The balance of uncoupled reads also advances during differentiation, as shown by the time course of mRNA abundance, and thus the future state of the cell can be extrapolated.

To validate this model, we used it to extrapolate the future state of the cell based on the current state of mRNA abundance, and then the future state of the cell can be extrapolated

using the time course of mRNA abundance (Fig. 1h). We found that the time course of mRNA abundance is consistent with the time course of gene expression dynamics (Fig. 1g). Uncoupled mRNA levels at each time point were consistently higher than spliced mRNA levels (Fig. 1g), and many circadian associated genes showed the expected excess of uncoupled mRNA relative to the \bar{m} during upregulation, and a decrease in uncoupled mRNA relative to the \bar{m} during downregulation. Using the proposed differential equations for each gene allowed us to extrapolate the time course of mRNA abundance, and thus the time course of gene expression dynamics. By extrapolating the time course of mRNA abundance, and then the future state of the cell can be extrapolated using the time course of mRNA abundance (Fig. 1h).

Using the time course of mRNA abundance to predict gene expression dynamics in single-cell measurements, we analyzed recently published single-cell RNA-seq data sets from the mouse forebrain⁹ and the mouse cortex¹⁰ (Fig. 1g). During development, a substantial proportion of cells in the forebrain and cortex displayed expression dynamics, as expected if uncoupled reads represent different genes.

To validate the time course of gene expression dynamics, we used the time course of mRNA abundance to predict the time course in which the direction of differentiation can be validated by lineage tracing. Phase portraits of many genes showed the expected deviation

- 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

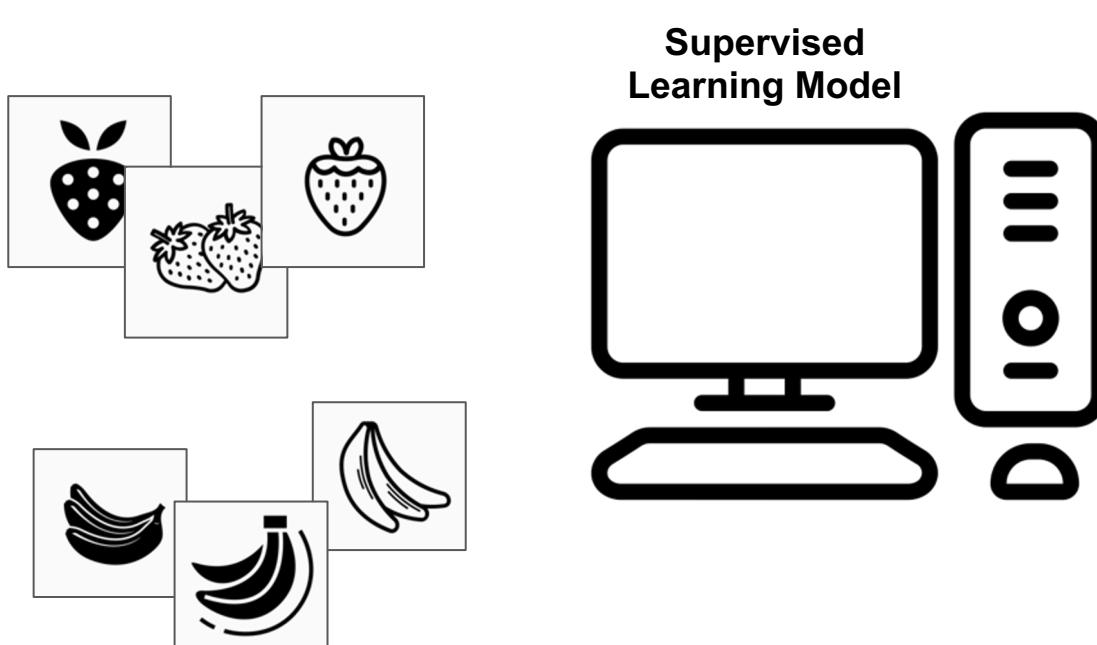
Supervised learning

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

Two kinds of machine learning

Supervised learning

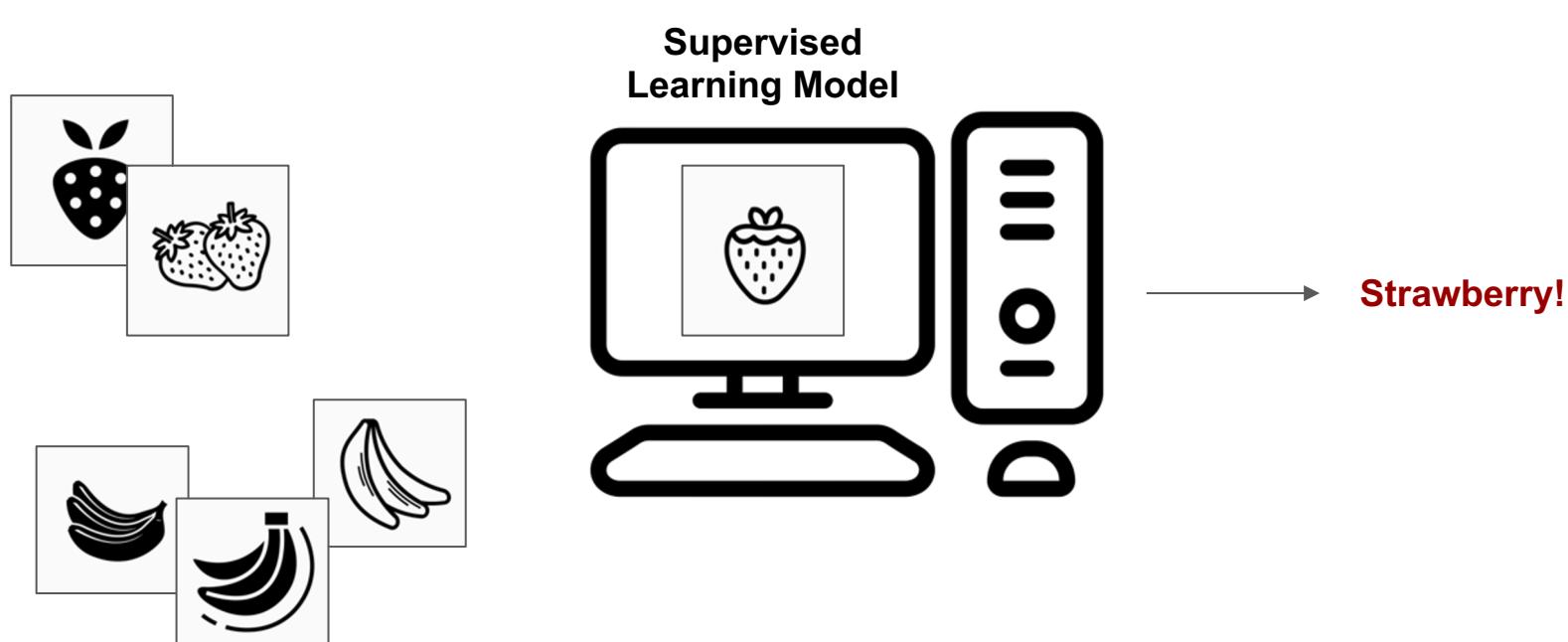
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

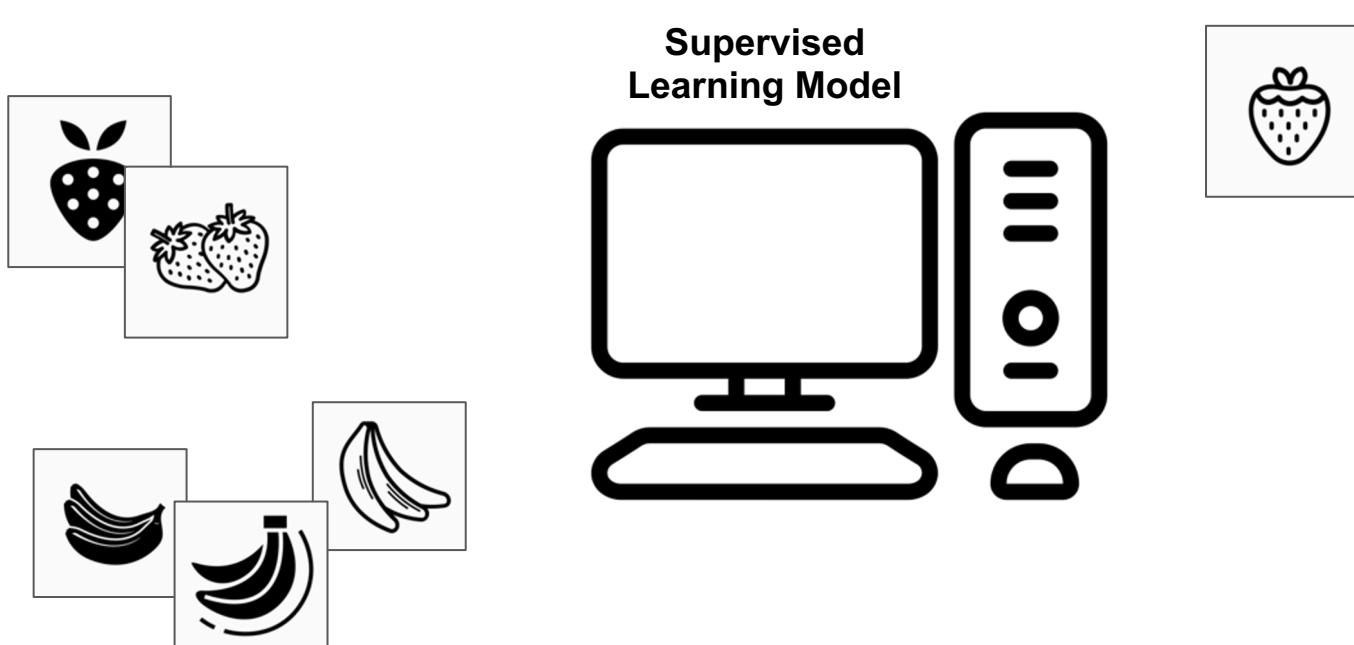
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

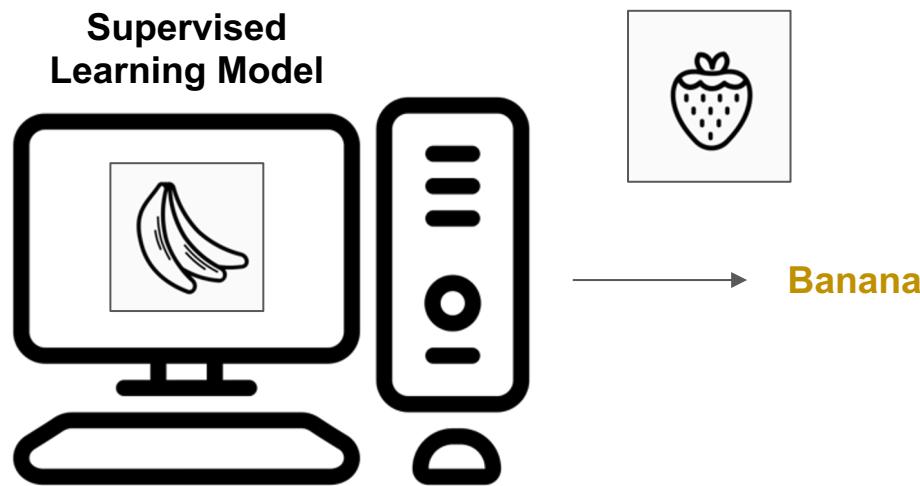
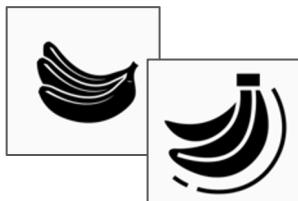
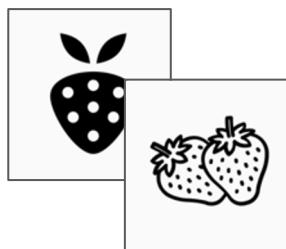
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

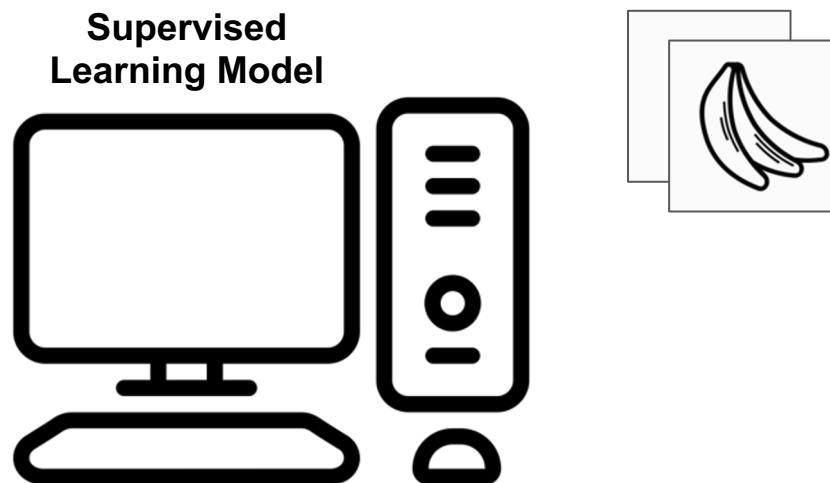
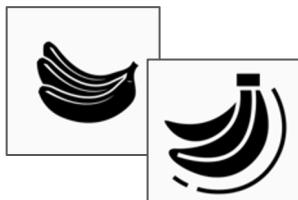
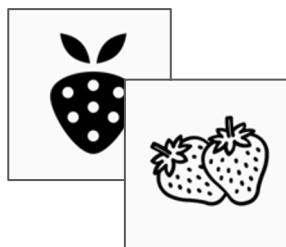
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

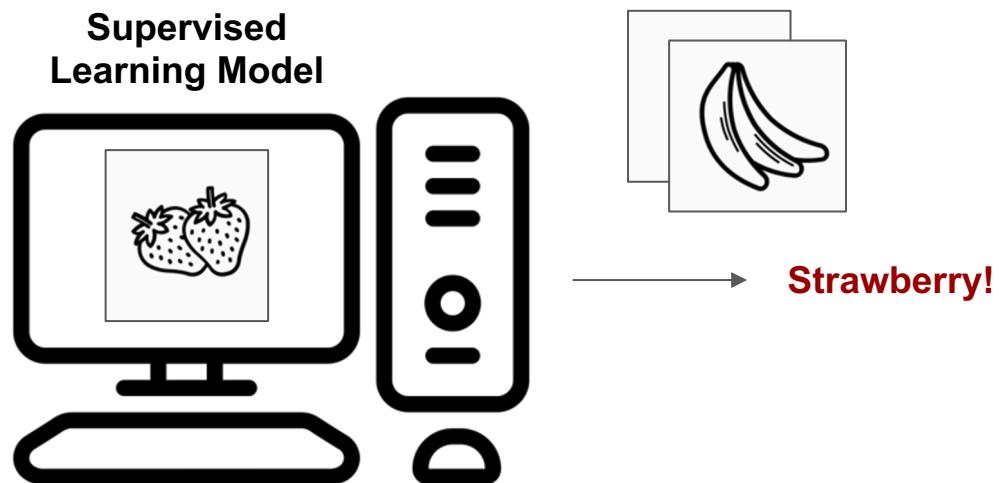
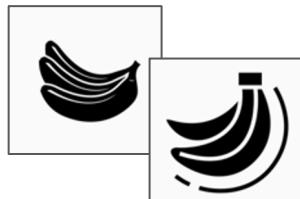
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

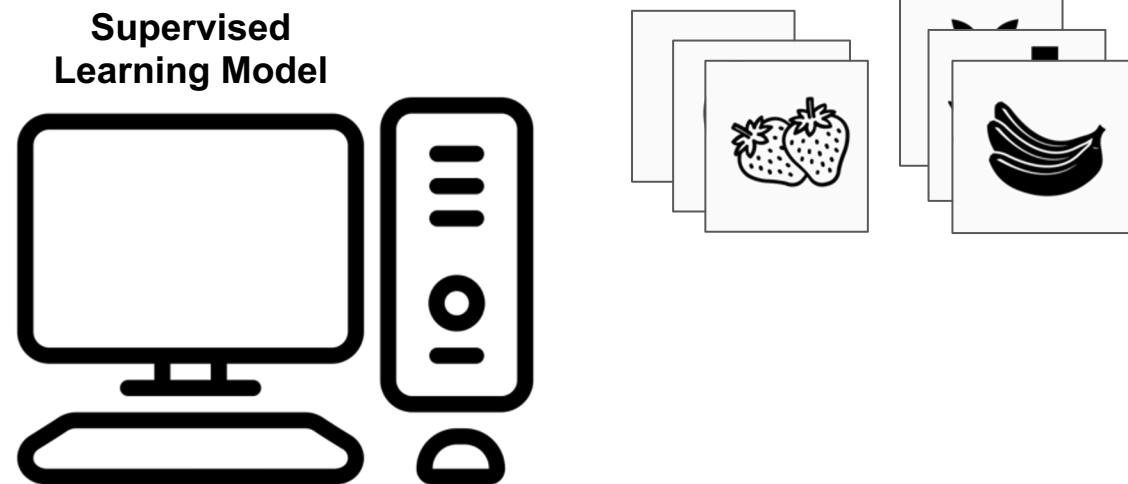
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

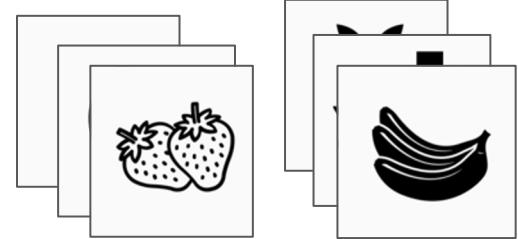
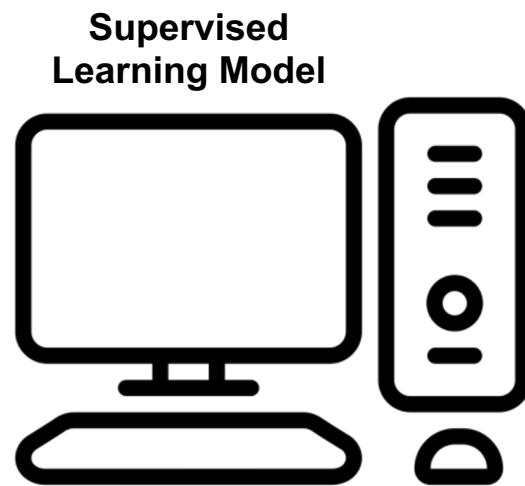
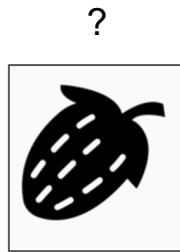
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

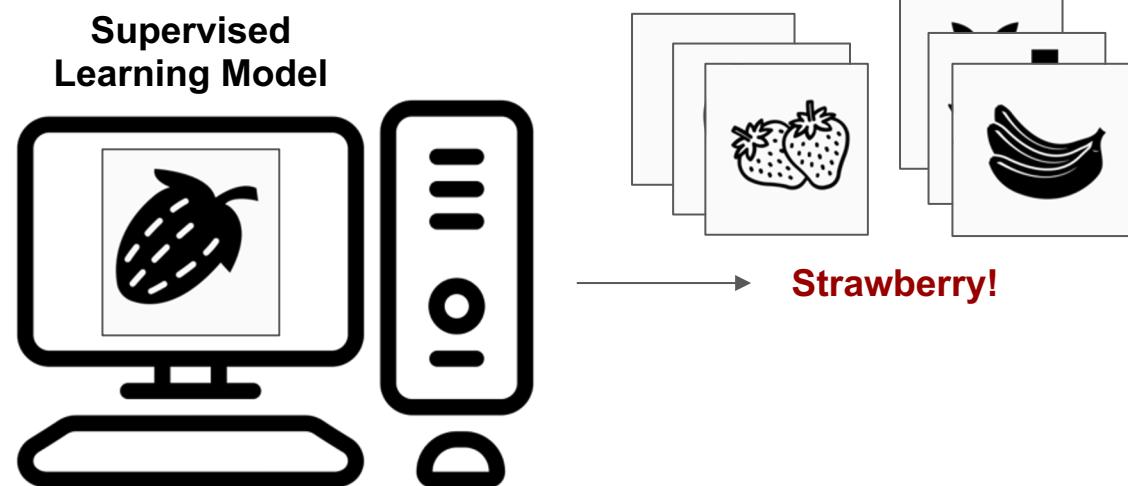
- Have a bunch of labelled data, want to label new data



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

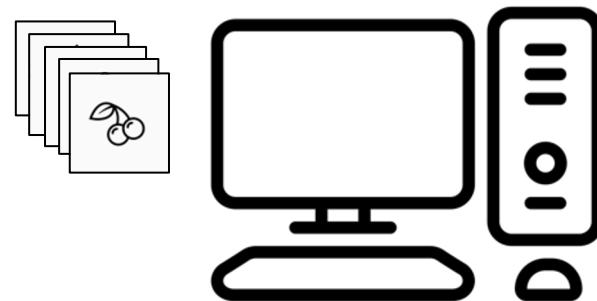
Supervised Learning Model



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

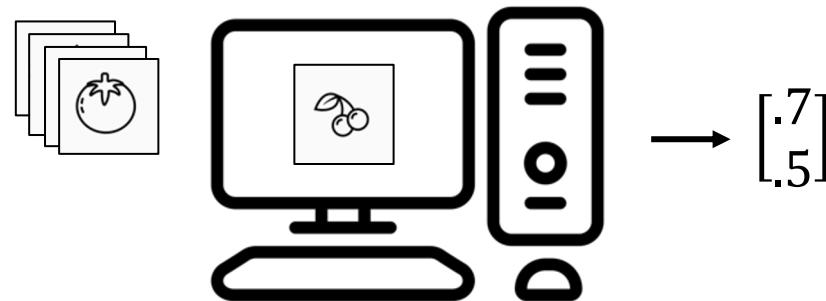
Supervised Learning Model



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

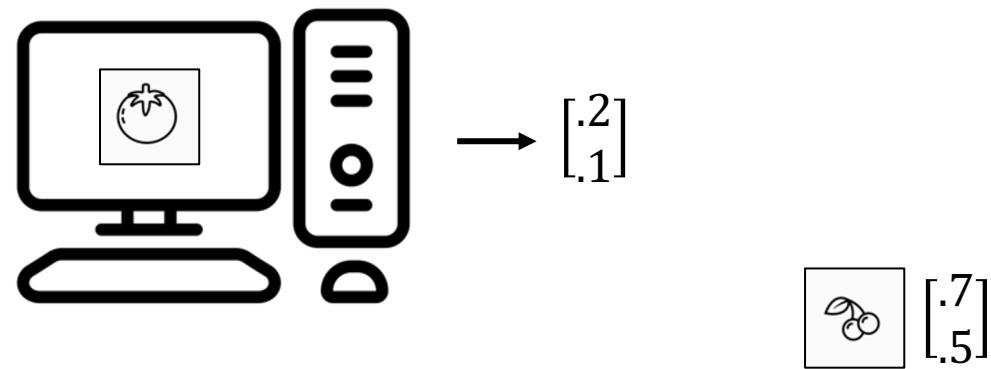
Supervised Learning Model



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



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

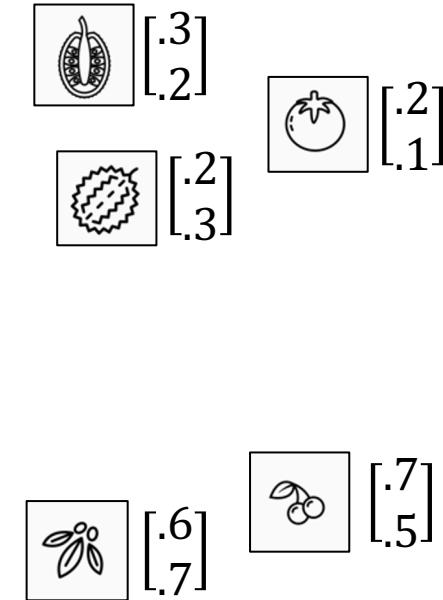
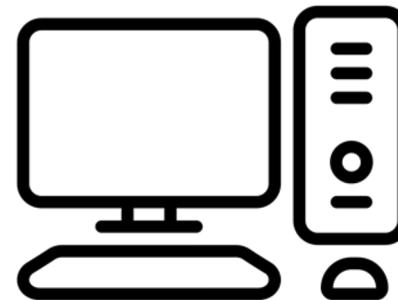


→ Strawberry

Unsupervised learning

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

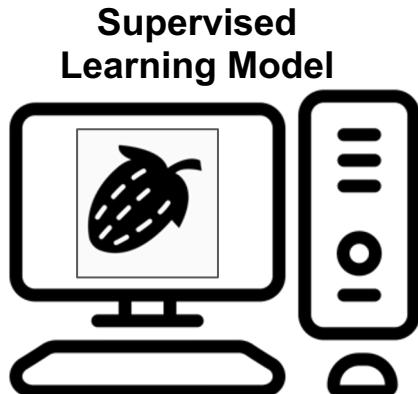
Unsupervised Learning Model



Two kinds of machine learning

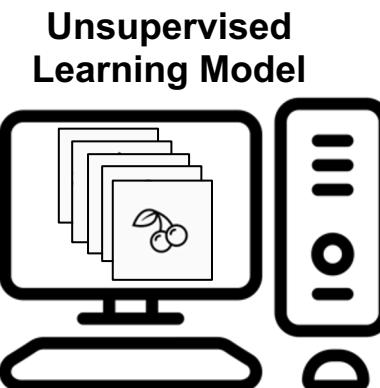
Supervised learning

- 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

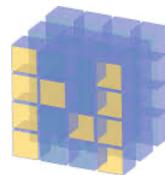
Lecture	View on Google Drive	Clustering and Data Denoising
Exercise	Run in Google Colab	3.0 Clustering Toy Data (Beginner)
	Run in Google Colab	3.0 Clustering Toy Data (Advanced)
	Run in Google Colab	3.1 Clustering & Denoising Embryoid Body Data (Advanced)
	Run in Google Colab	3.2 Batch correction in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

Day 4 – Wednesday, May 27th

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

Why Python?

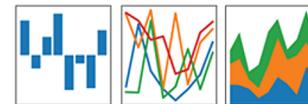




NumPy

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:

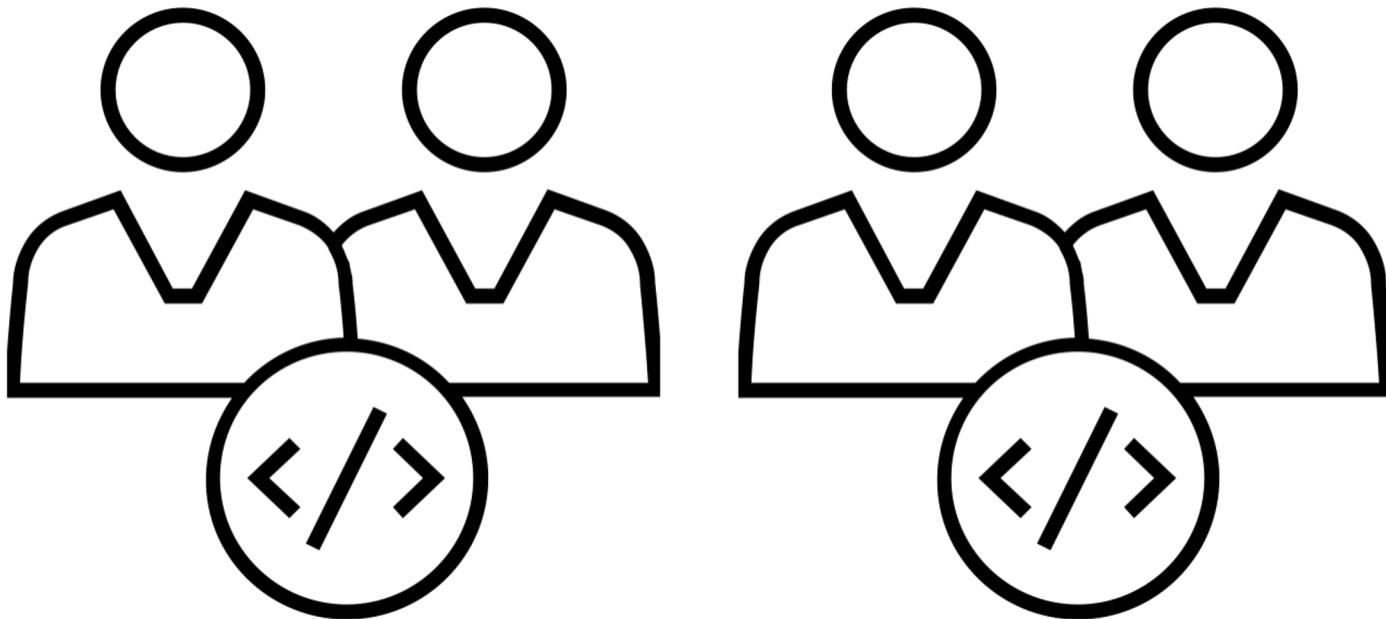
```
[ ] seconds_in_a_day = 24 * 60 * 60
```

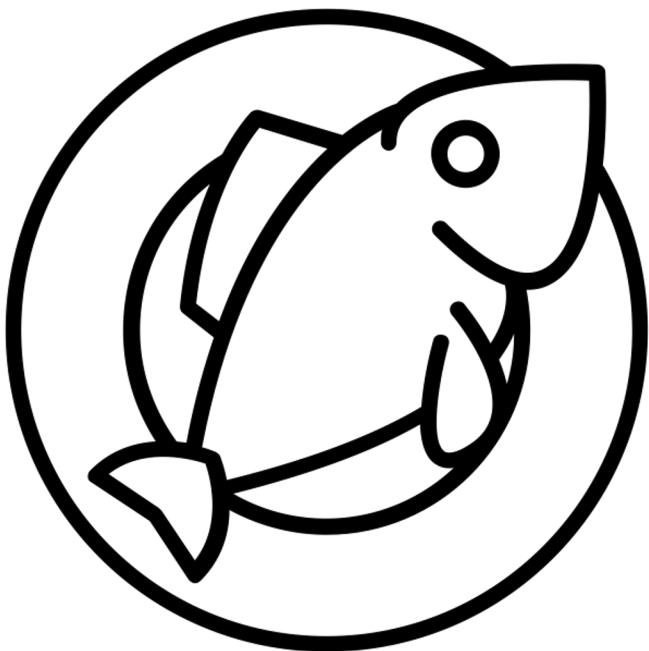
seconds_in_a_day

72000

 and instructions on how to execute it.

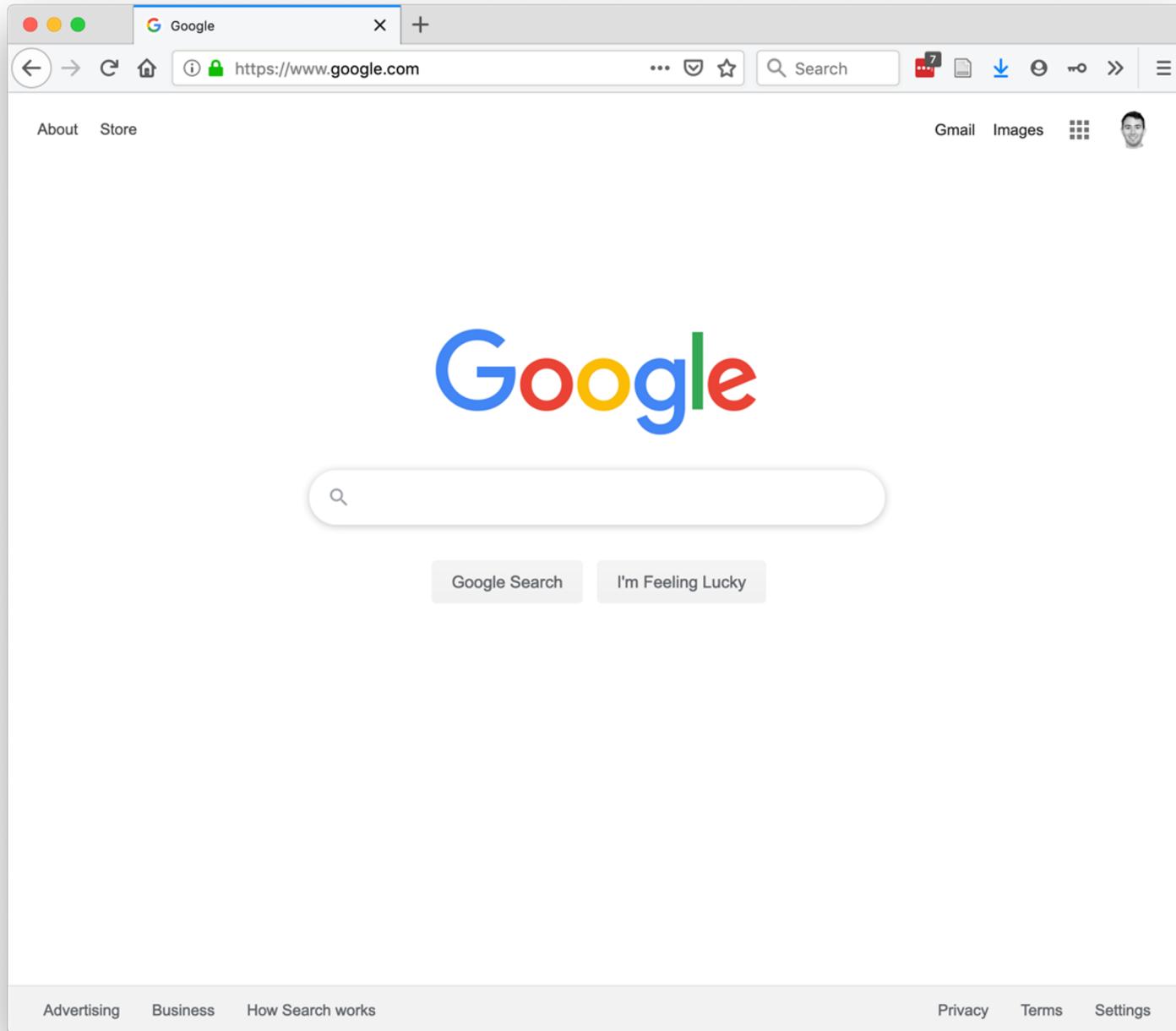
Team programming





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



#2020-workshop-byod-help
<https://krishnaswamylab.org/get-help>

Data Matrices and Representations

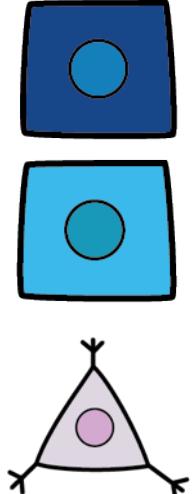
Single Cell Data

- 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)

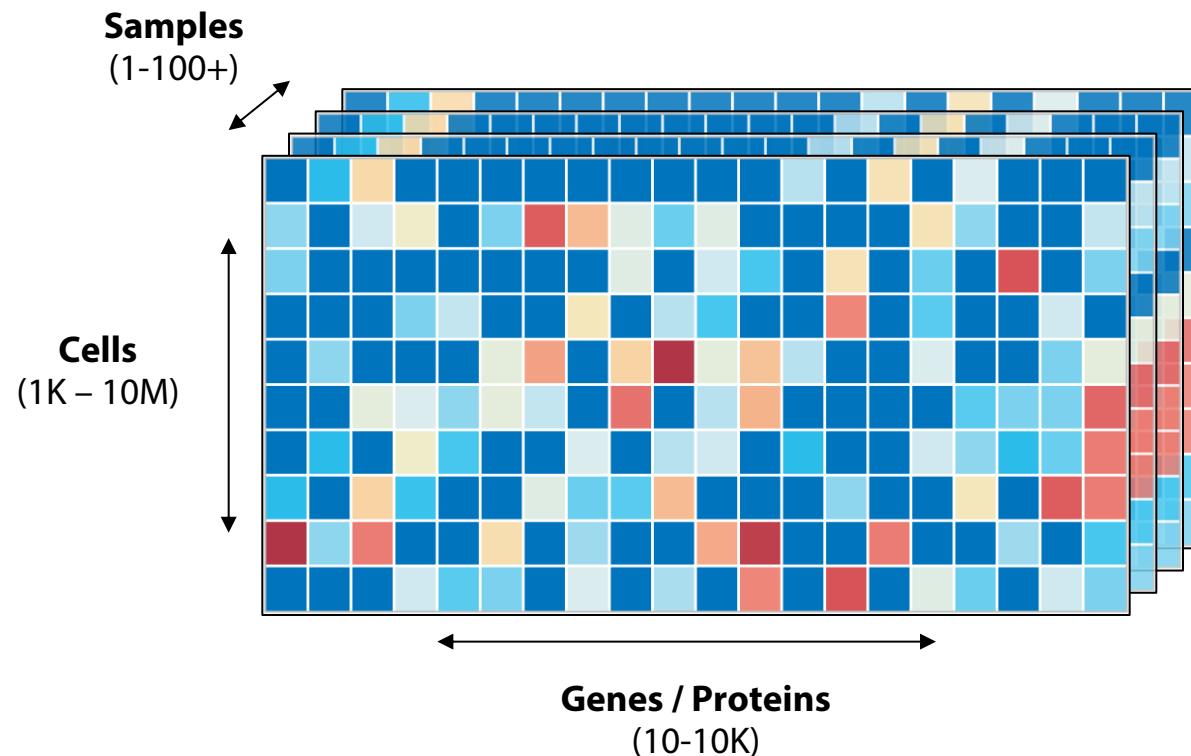
Features
(e.g. genes)

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

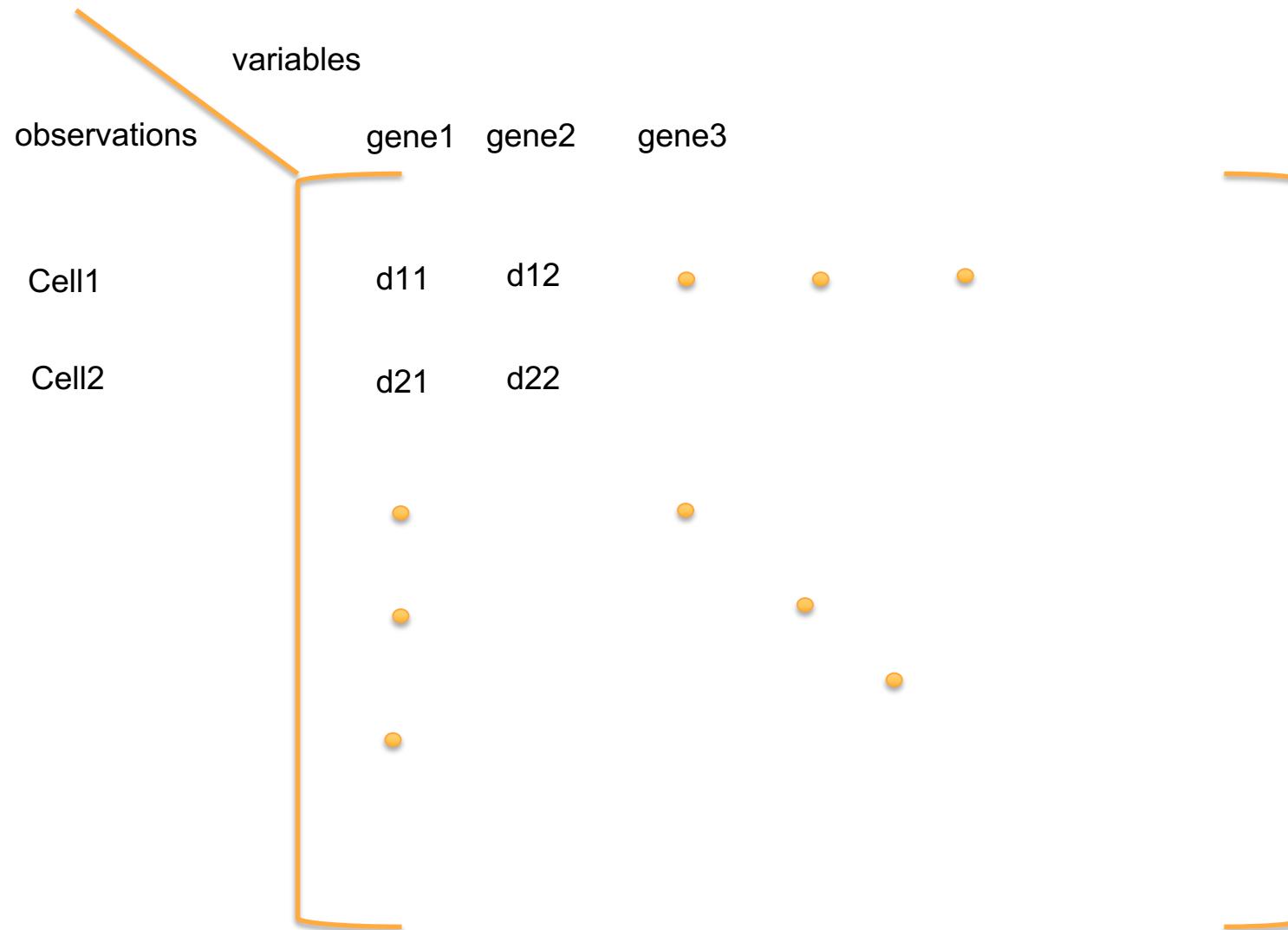
Observations
(e.g. cells)



Single Cell Data

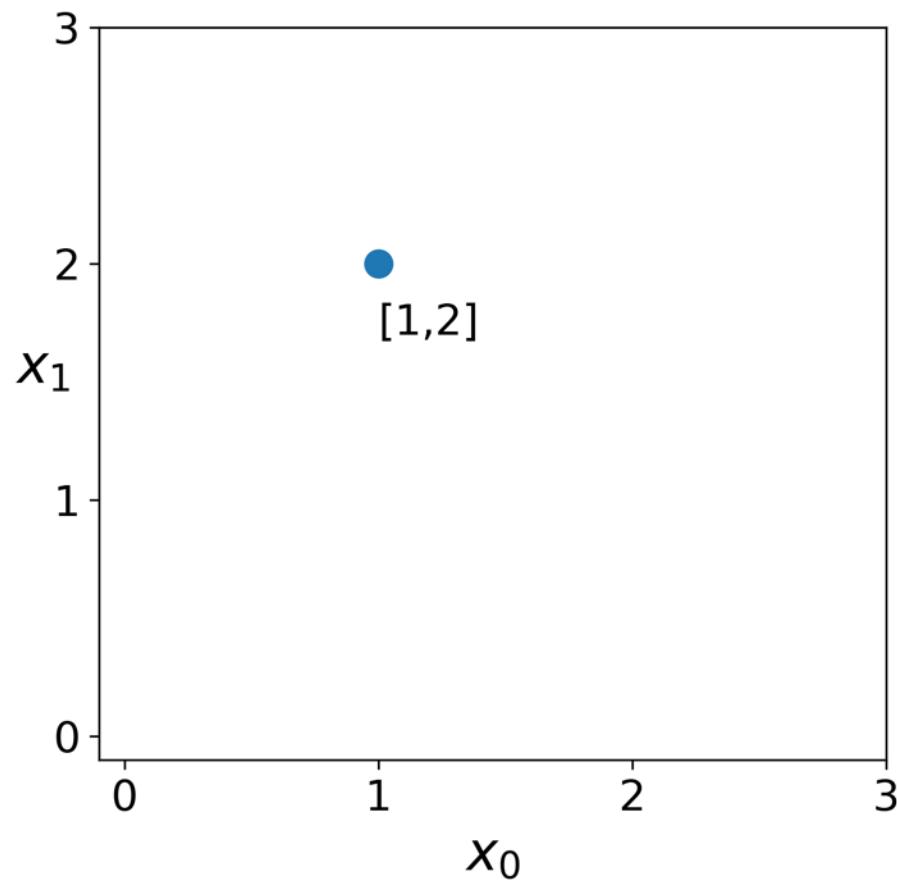


Our Data is a High-dimensional Matrix



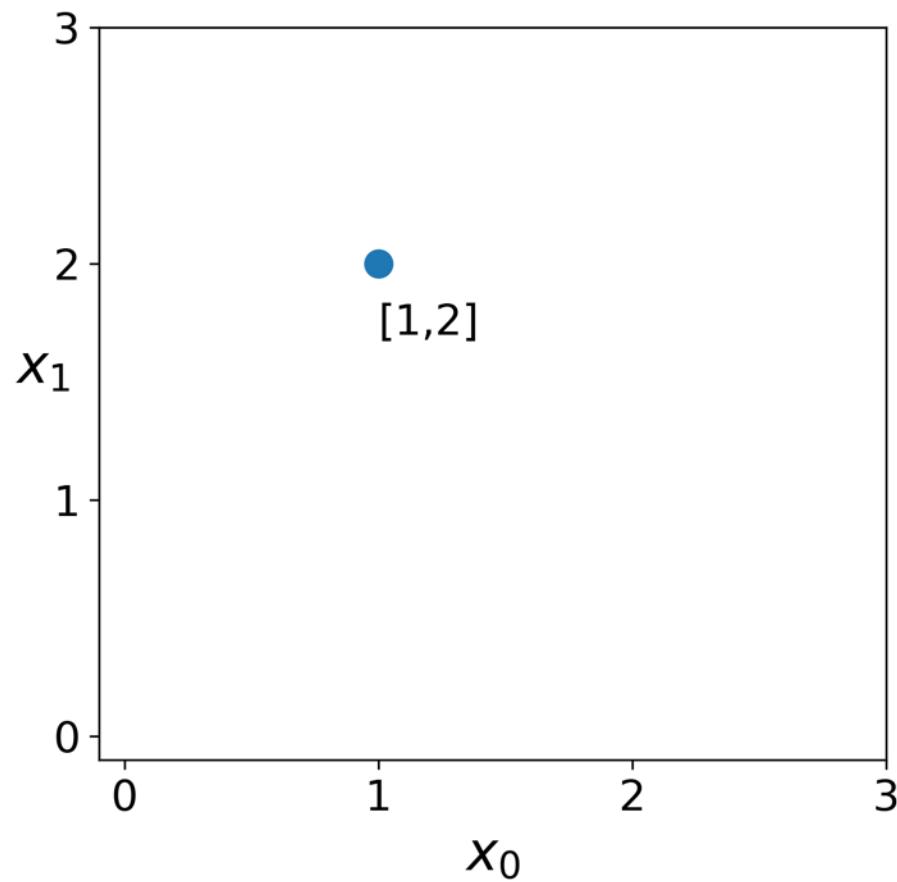
Spatial Representation of Data

\mathbb{R}^2 two-dimensional space



Spatial Representation of Data

\mathbb{R}^2 two-dimensional space



Spatial Representation of Data

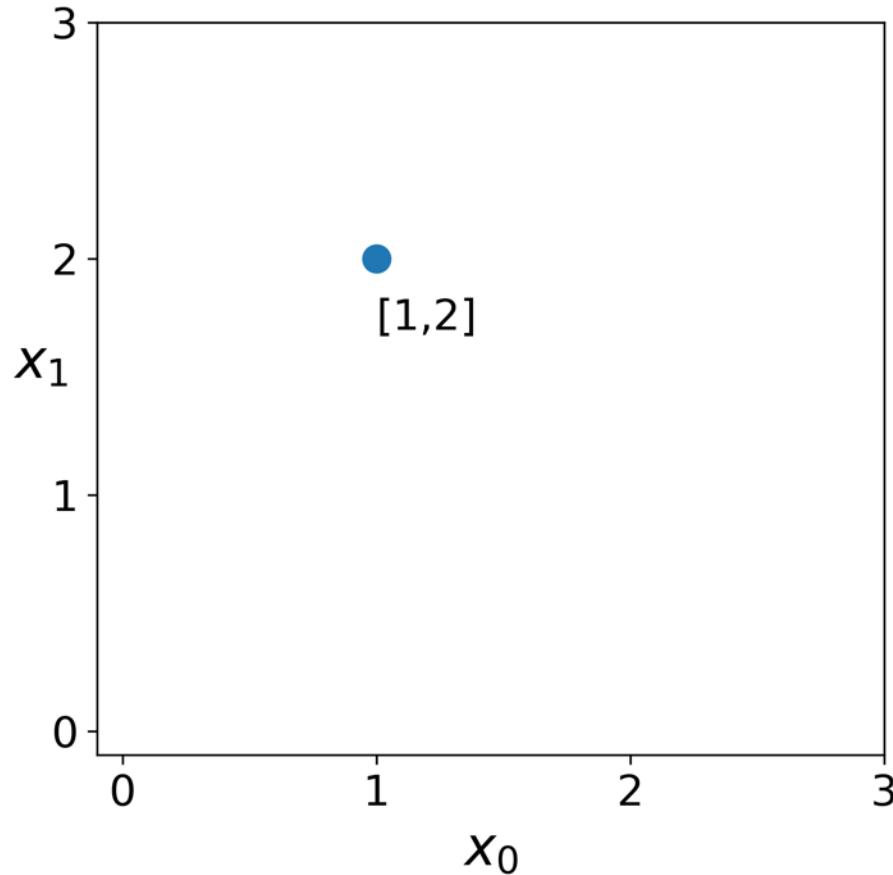
$$a \in \mathbb{R}^2$$

$$a = [1, 2]$$

x_0

x_1

\mathbb{R}^2 two-dimensional space



Features are dimensions

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

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

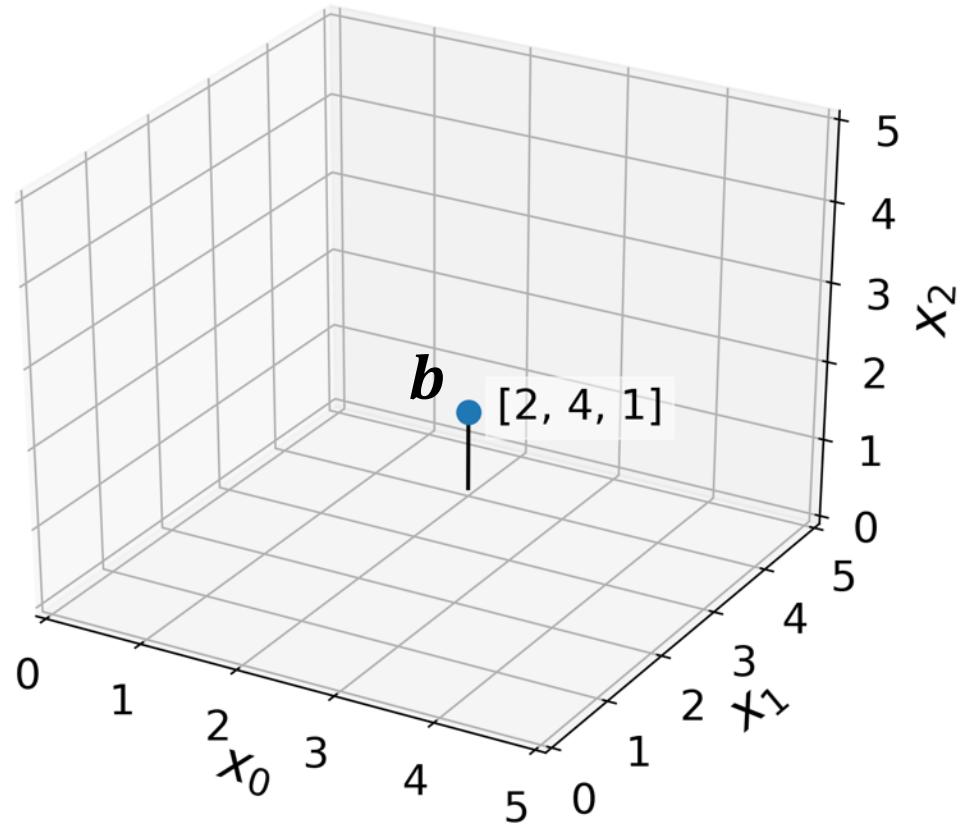
x_0 x_1

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

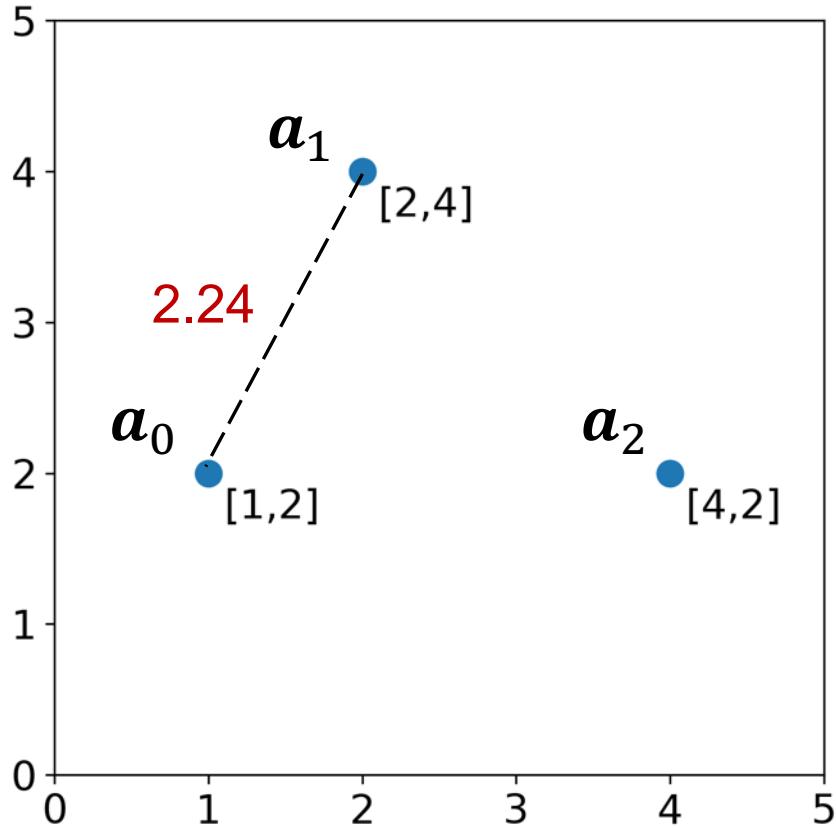
$$\mathbf{b} = [2, 4, 1]$$

x_0 x_1 x_2

\mathbb{R}^3 three-dimensional space

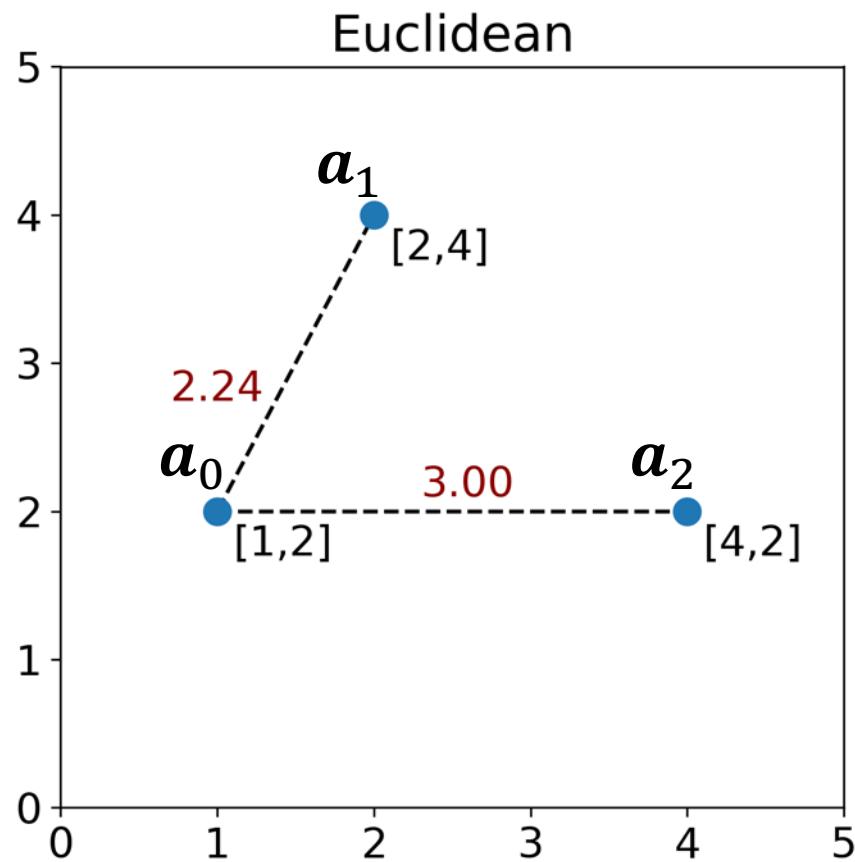


Euclidean Distance between observations

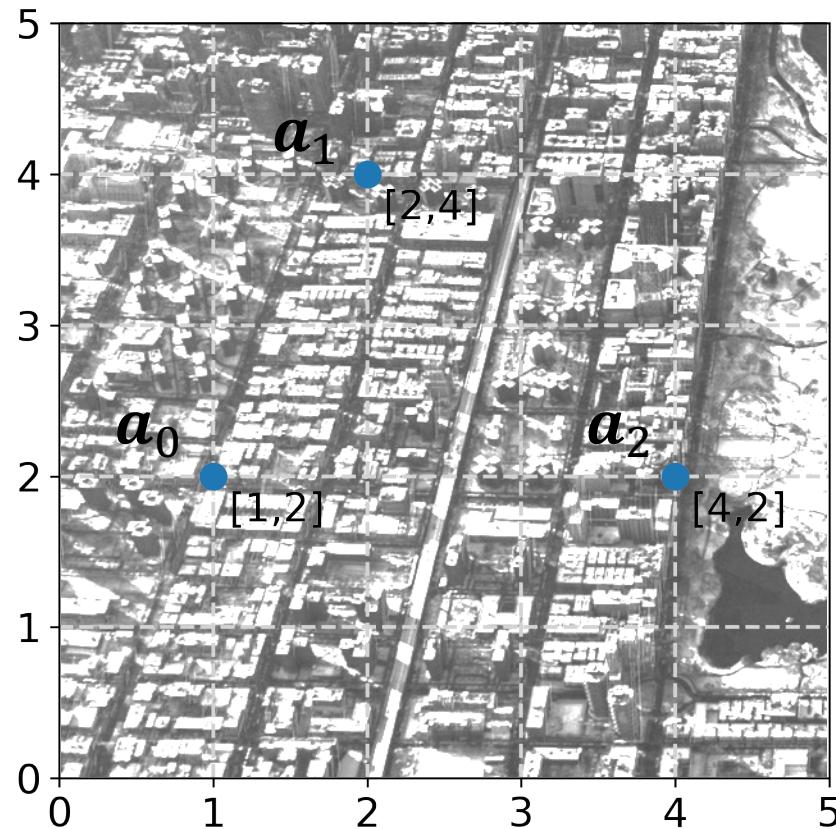


$$\begin{aligned}d_{euclidean}(a_0, a_1) &= \|a_0 - a_1\|_2^2 \\&= \sqrt{(a_{0,0} - a_{1,0})^2 + (a_{0,1} - a_{1,1})^2} \\&= \sqrt{(1 - 2)^2 + (2 - 4)^2} \\&\approx 2.24\end{aligned}$$

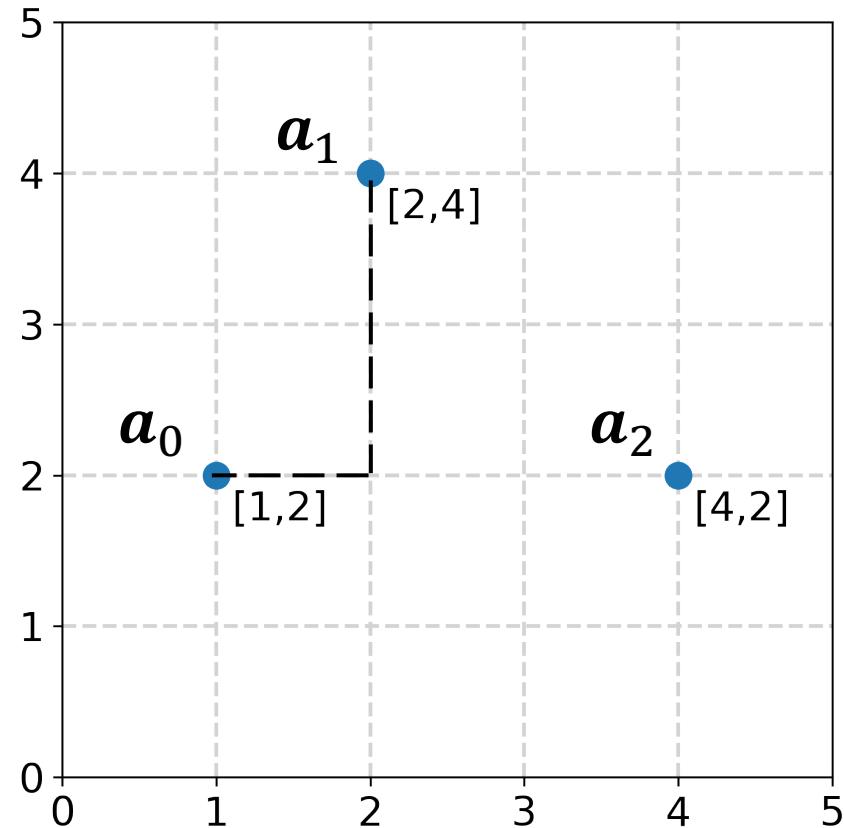
How far away is each point from a_0 ?



How far away is each point from a_0 ?



Manhattan Distance



Manhattan distance

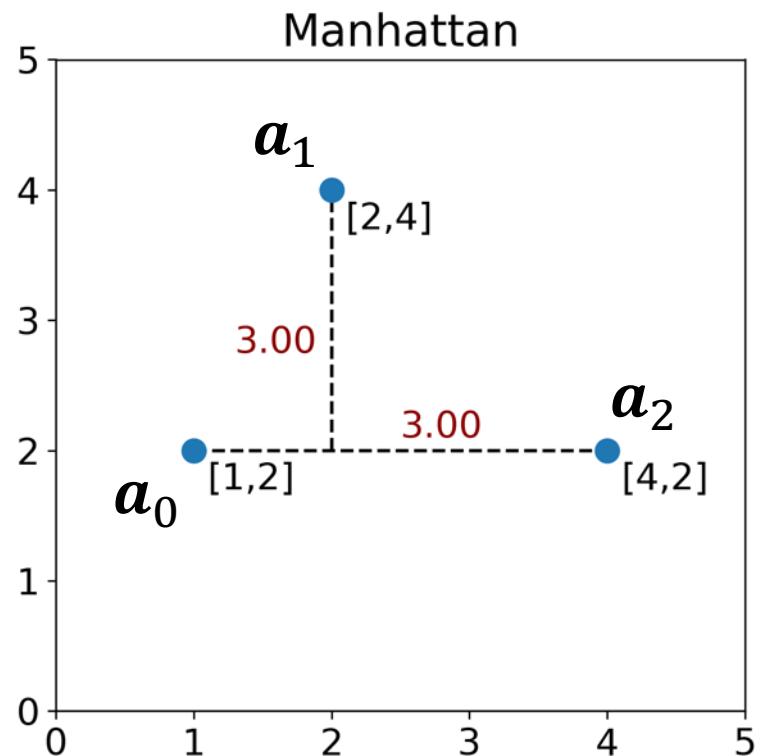
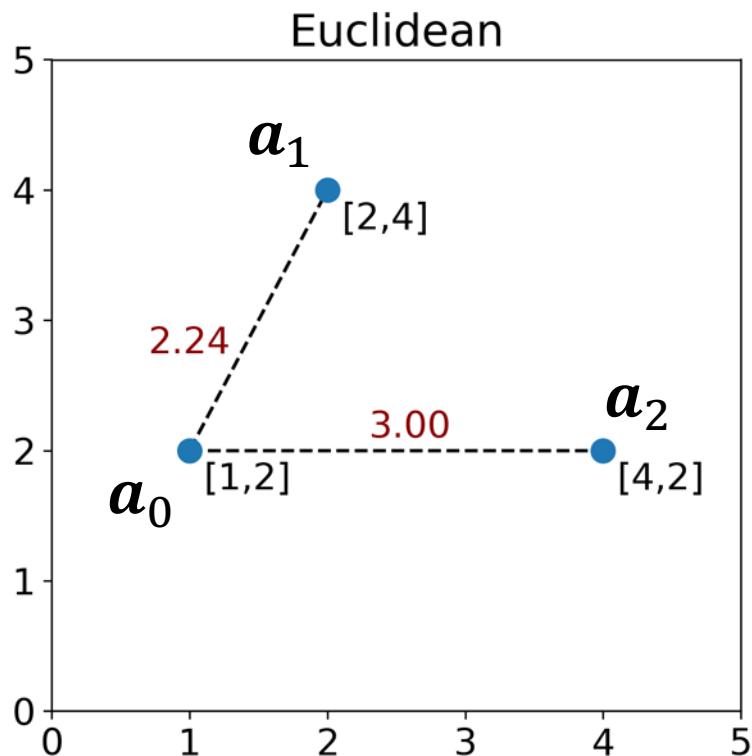
$$d_{manhattan}(a_0, a_1) = |a_{0,1} - a_{1,1}|$$

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

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

$$= 3$$

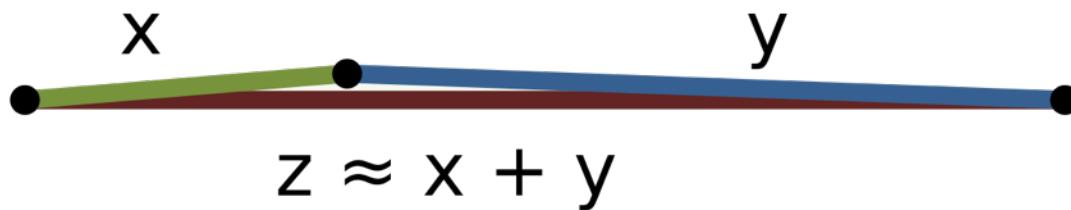
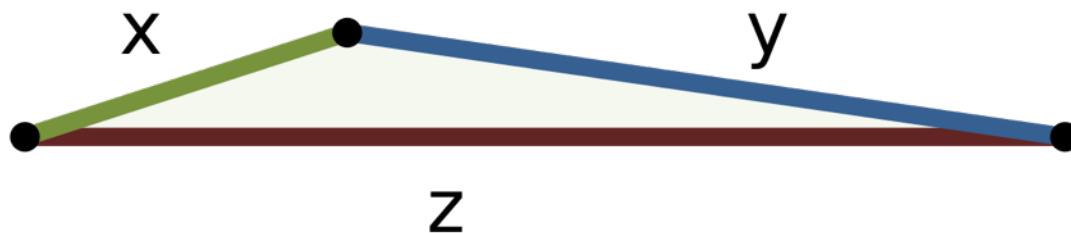
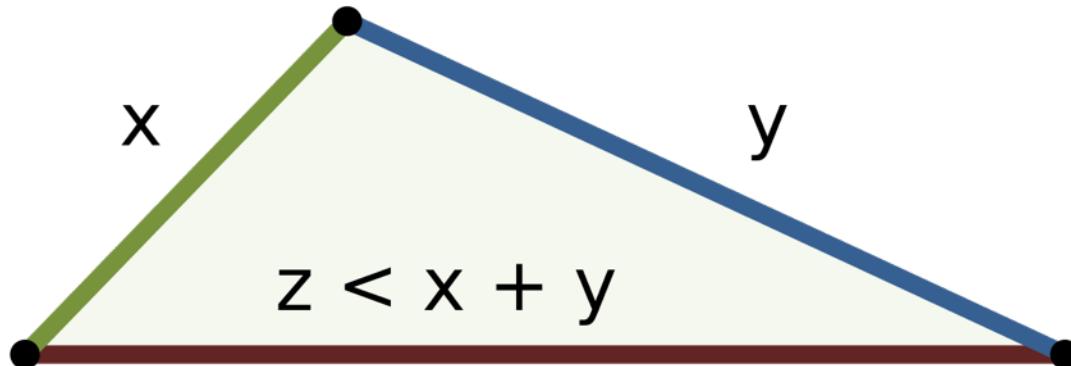
How far away is each point from a_0 ?



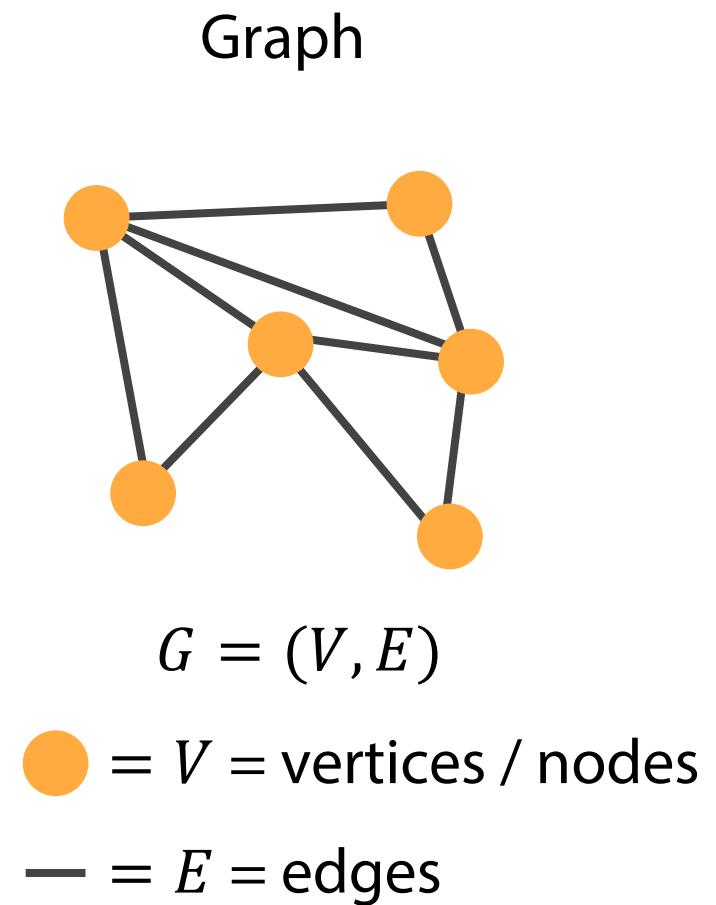
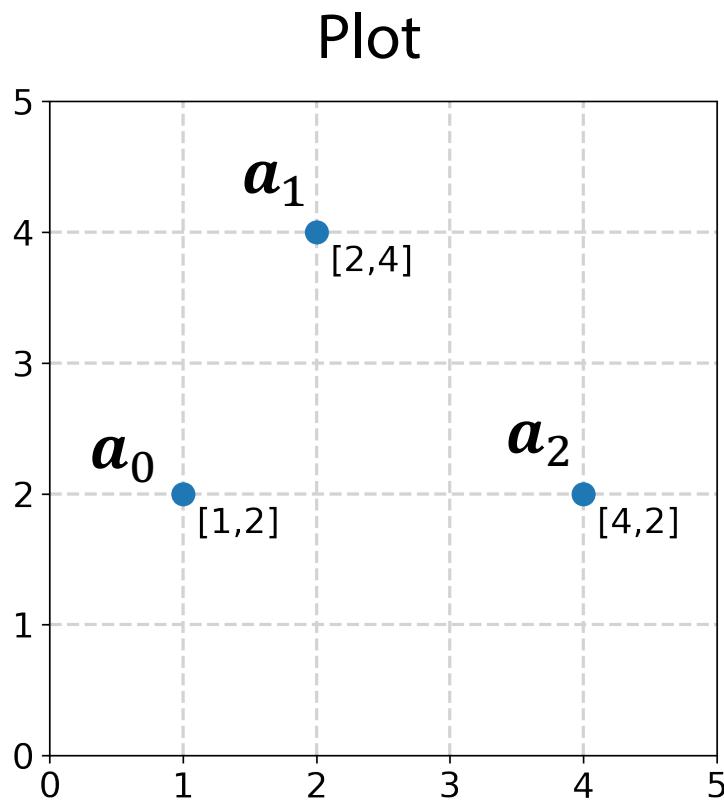
Distances

- 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:** $\text{dist}(a \rightarrow c) \leq \text{dist}(a \rightarrow b) + \text{dist}(b \rightarrow c)$

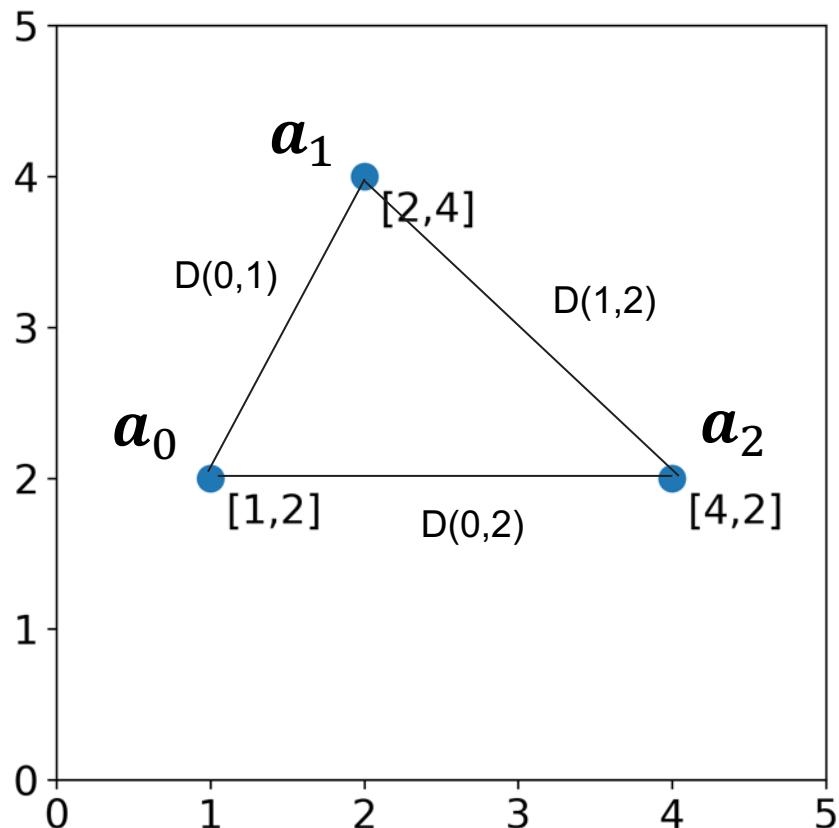
Triangle Inequality



Representing Data as a Graph



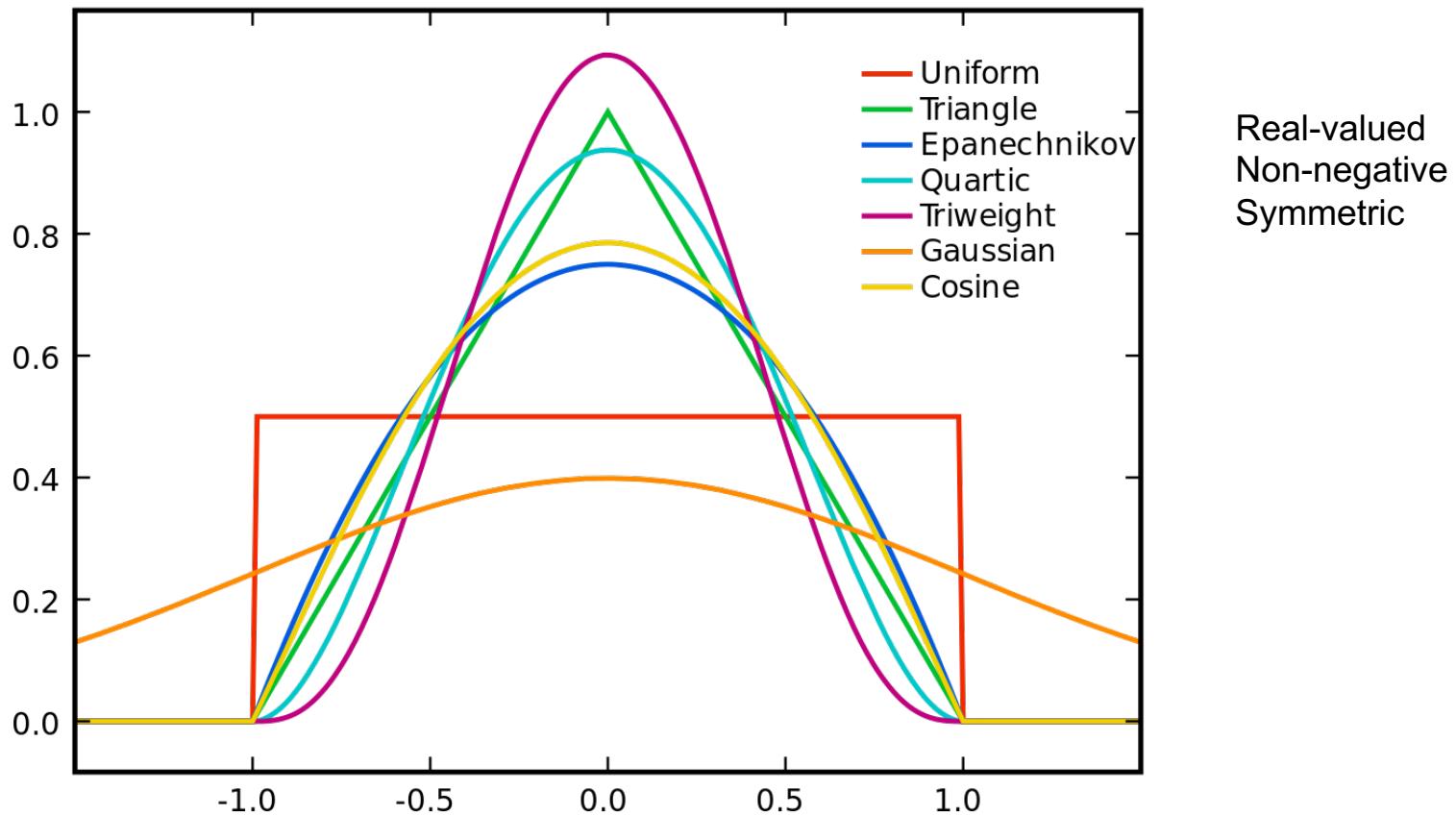
Fully connected Graph



	a_0	a_1	a_2
a_0	0	$D(0,1)$	$D(0,2)$
a_1	$D(0,1)$	0	$D(1,0)$
a_2	$D(2,0)$	$D(2,1)$	0

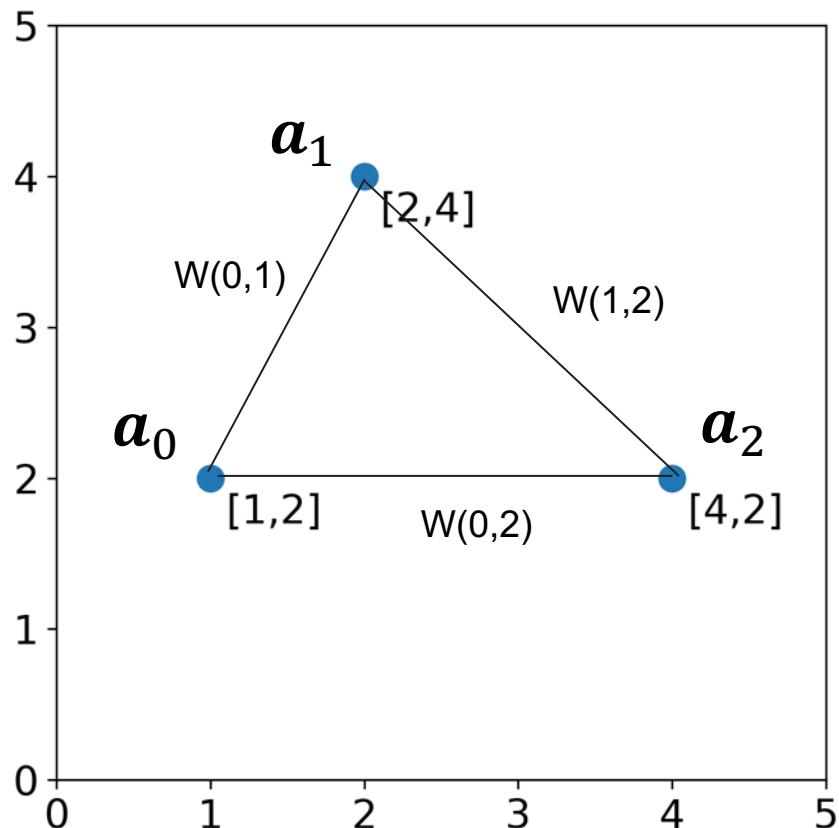
Distance matrix

Distance to Affinity via Kernels



Affinities correlations in this hidden hypothetical space

Affinities

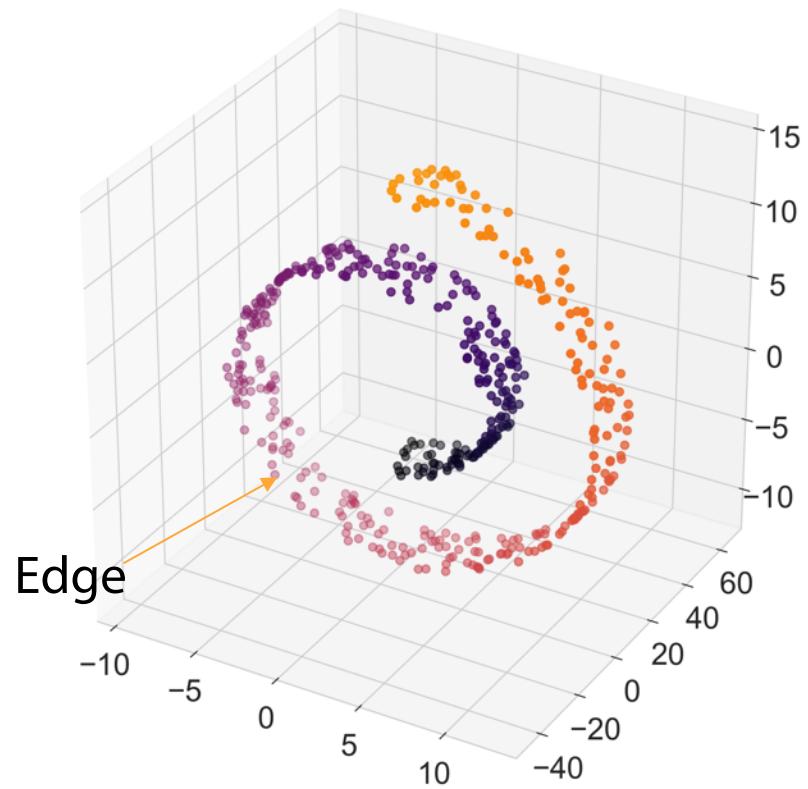


	a_0	a_1	a_2
a_0	0	$W(0,1)$	$W(0,2)$
a_1	$W(0,1)$	0	$W(1,0)$
a_2	$W(2,0)$	$W(2,1)$	0

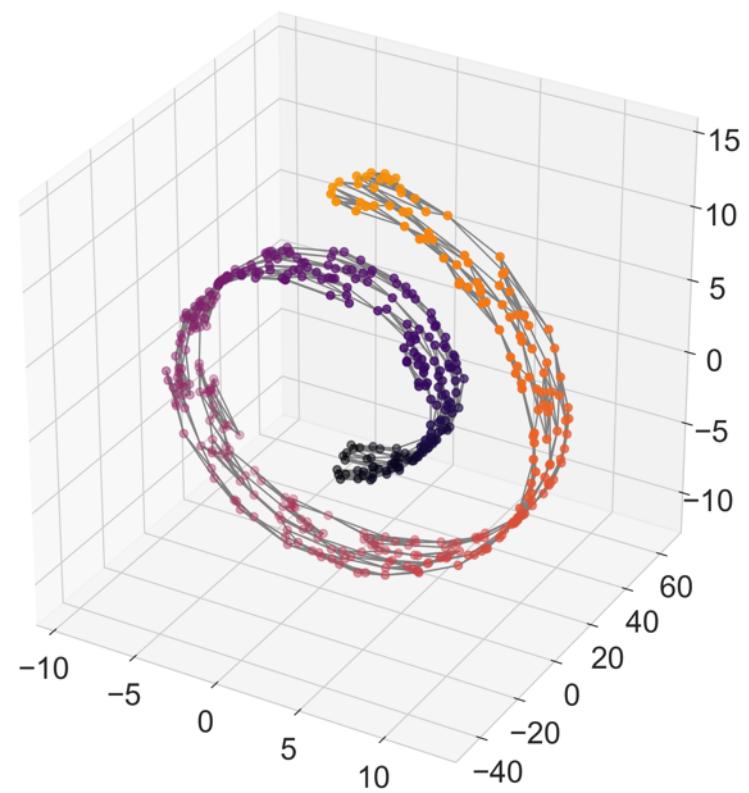
Affinity or Adjacency Matrix

Nearest Neighbors Graph

Data

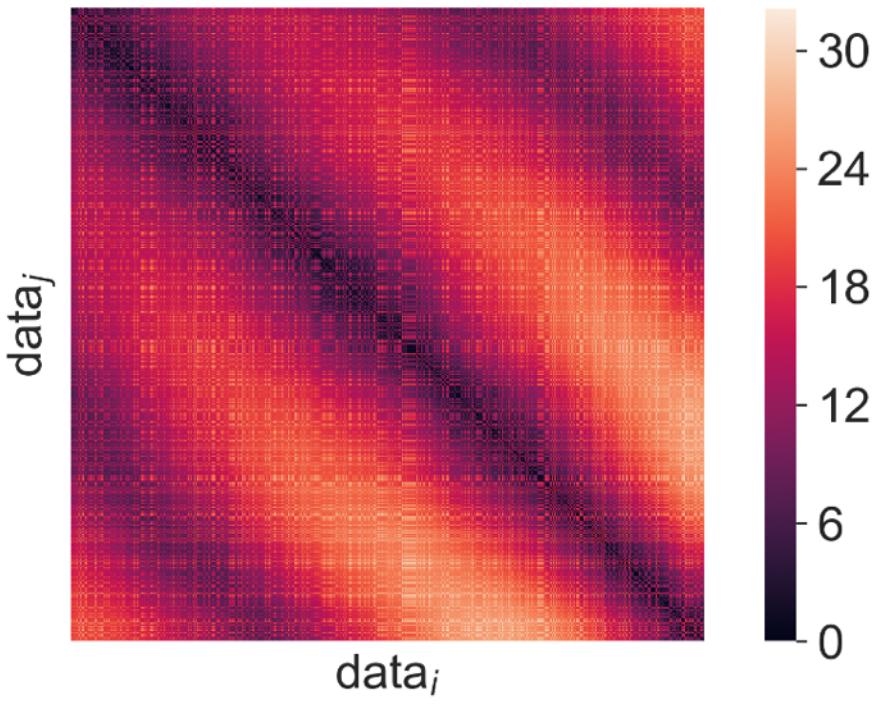


Nearest Neighbor Graph

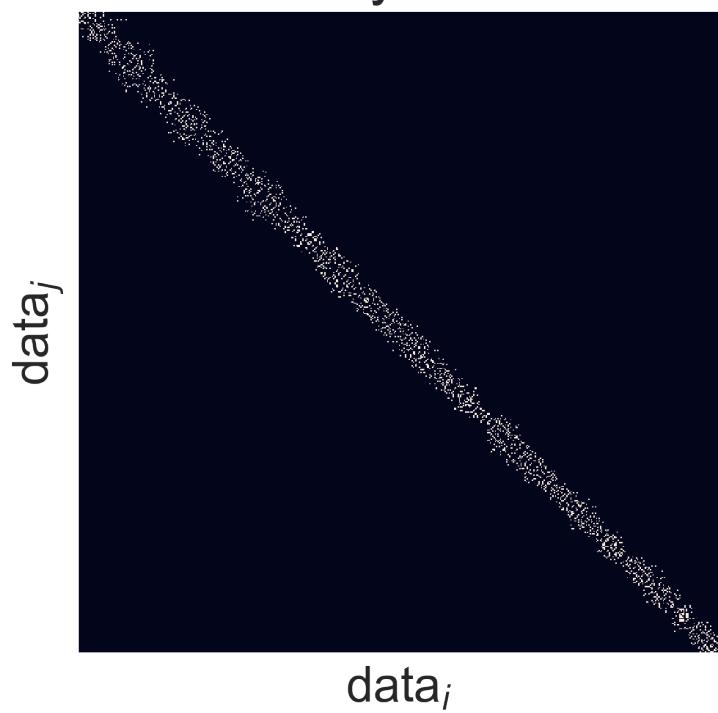


Swiss Roll

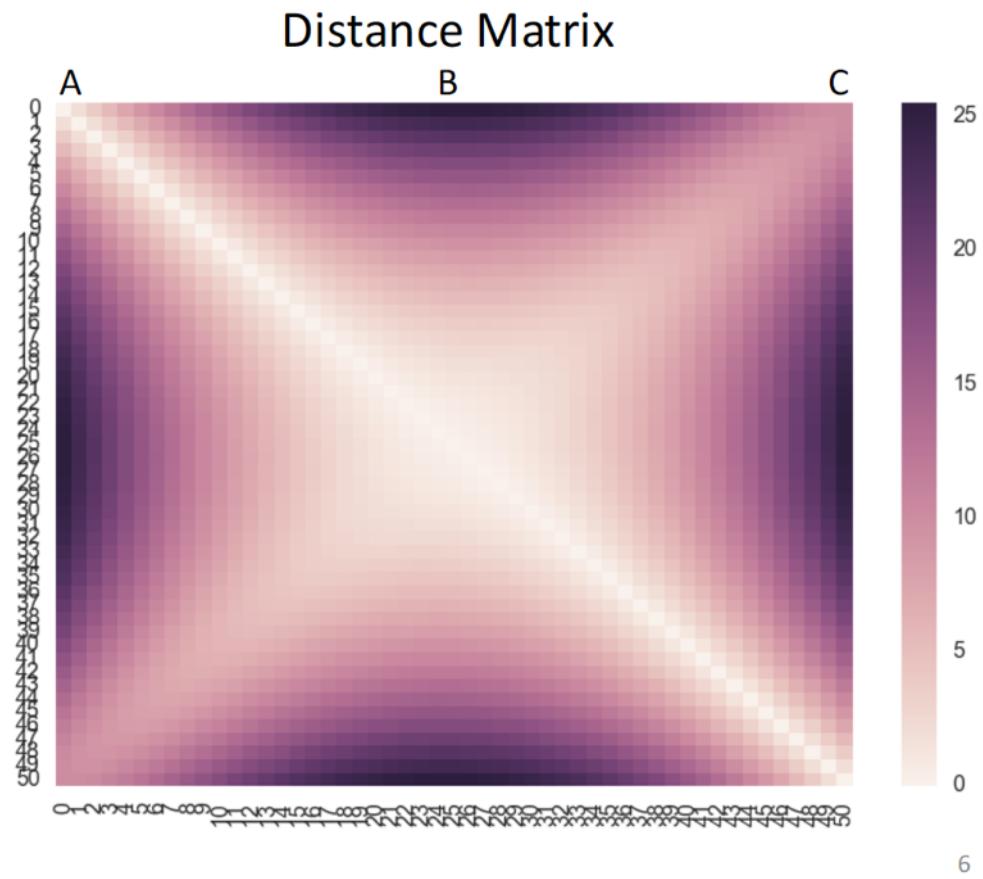
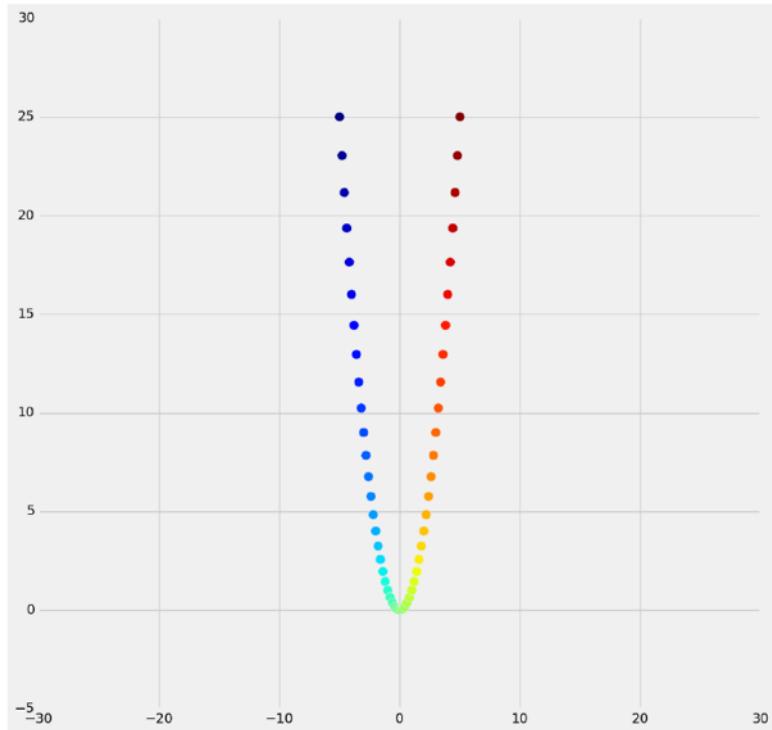
Distance Matrix

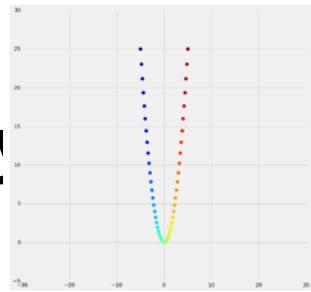


Affinity Matrix

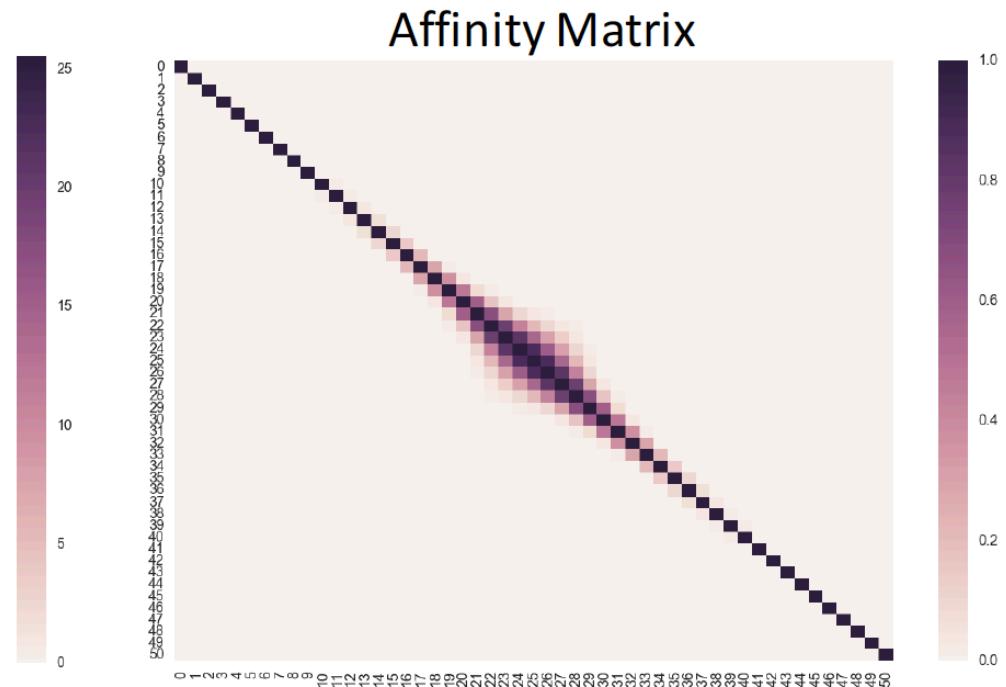
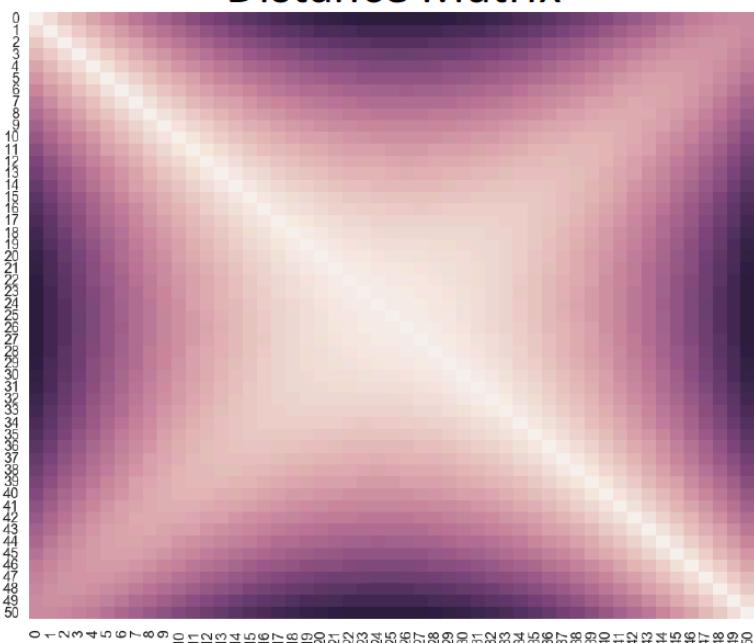


Kernel functions measure “similarity” or “affinity” on a graph

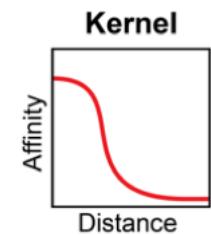




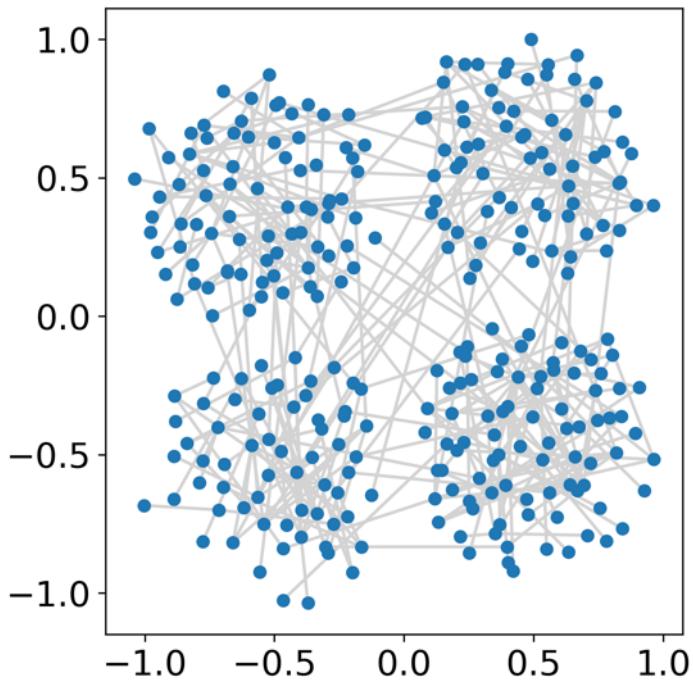
Affinity is the inverse of distance + locality



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



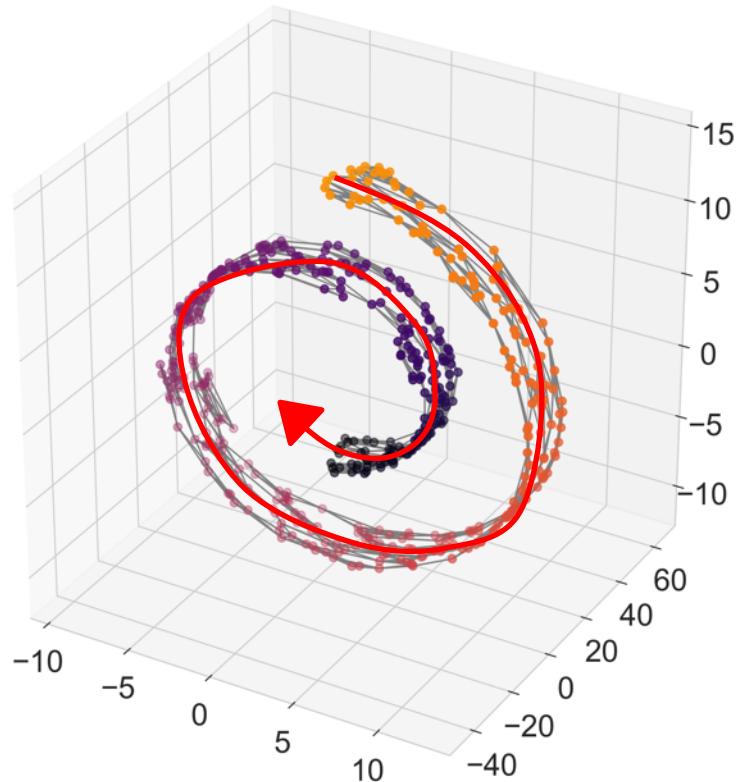
Why Represent Data as a Graph?



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

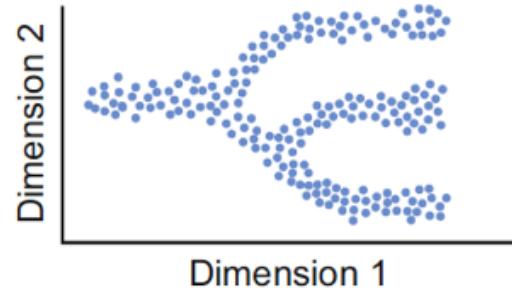
Why Represent Data as a Graph?



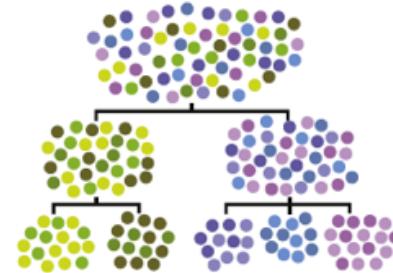
Paths through data graphs can represent progression trajectories

And much more to come!!

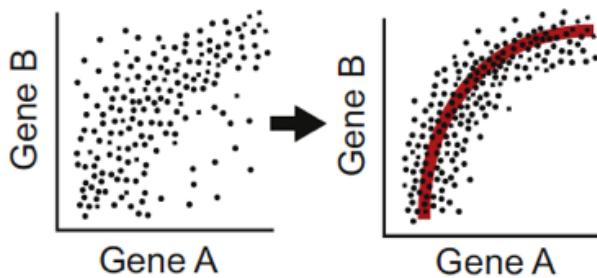
Vizualization



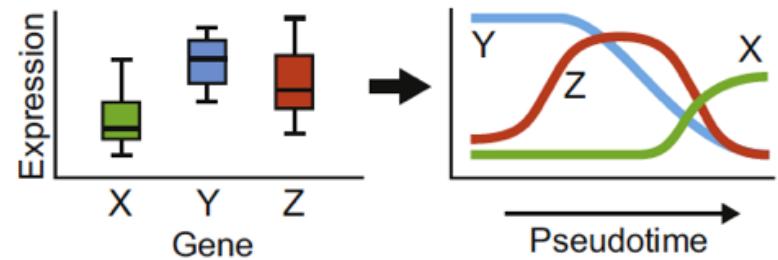
Clustering



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.

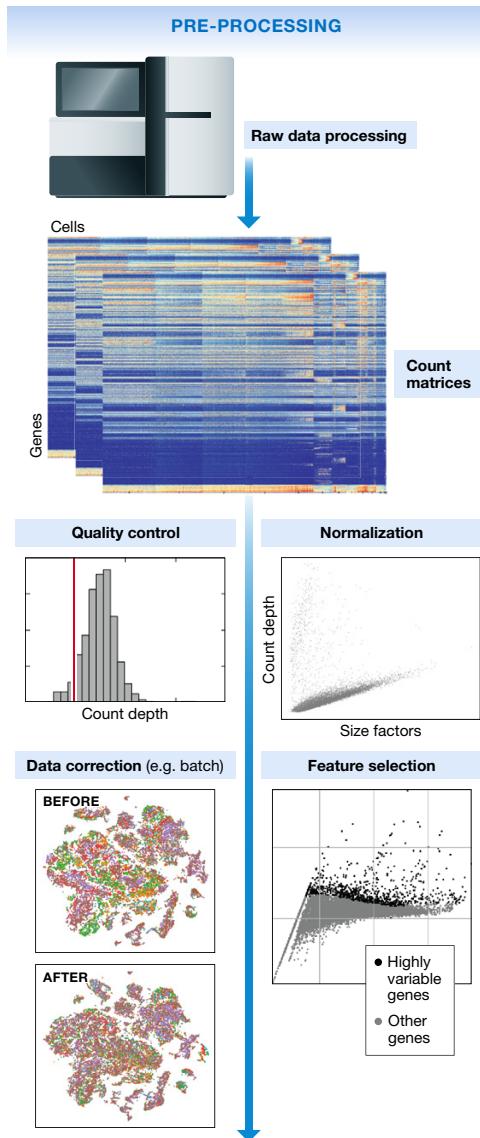
The challenges to standardization include the growing number of analysis methods (385 tools as of 7 March 2019) and exploding dataset sizes (Angerer *et al.*, 2017; Zappia *et al.*, 2018). We are continuously finding new ways to use the data at our disposal. For example, it has recently become possible 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

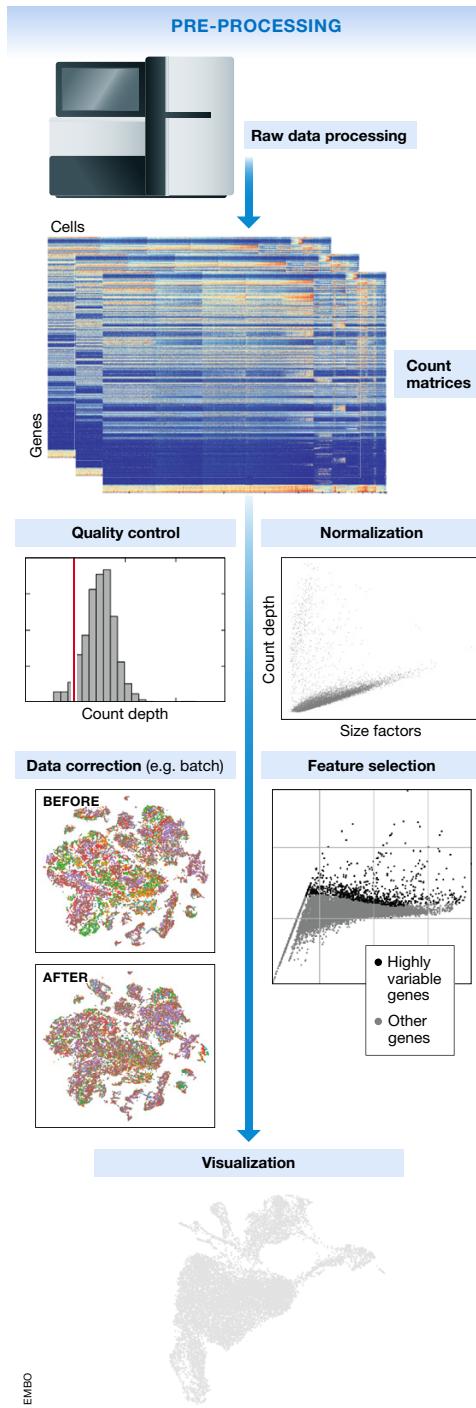
Standard Single-Cell RNA-seq Workflow

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



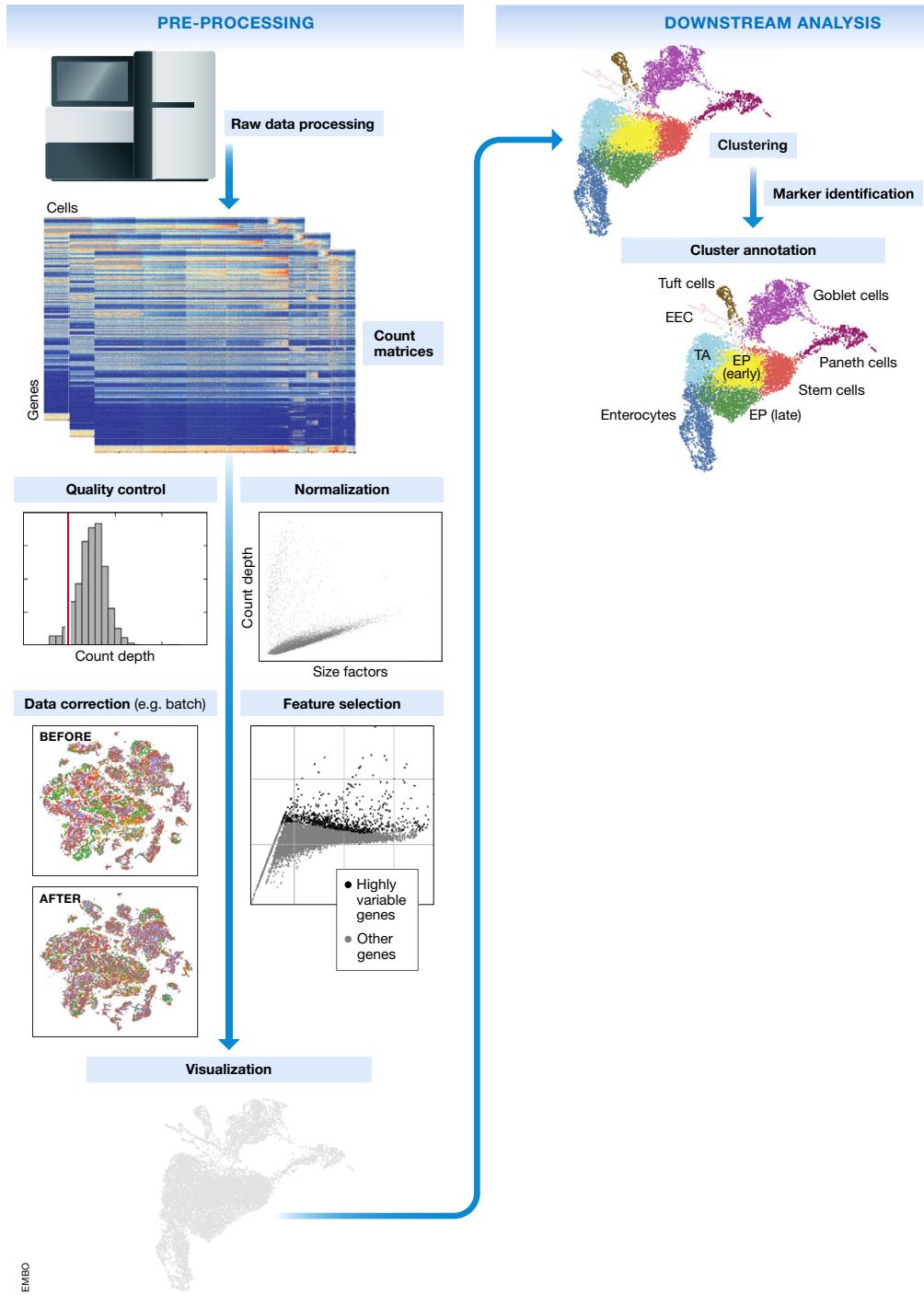
Standard Single-Cell RNA-seq Workflow

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



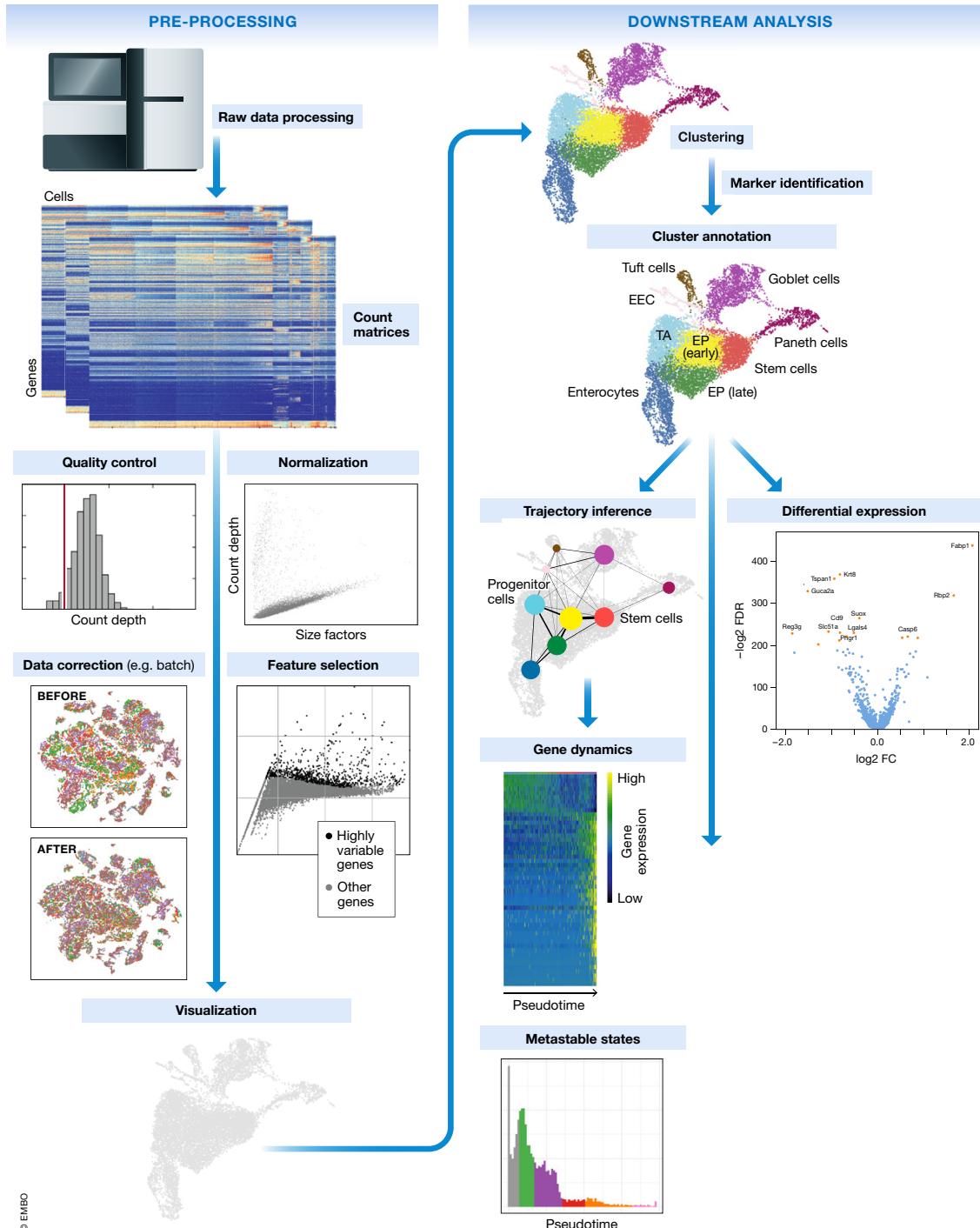
Standard Single-Cell RNA-seq Workflow

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



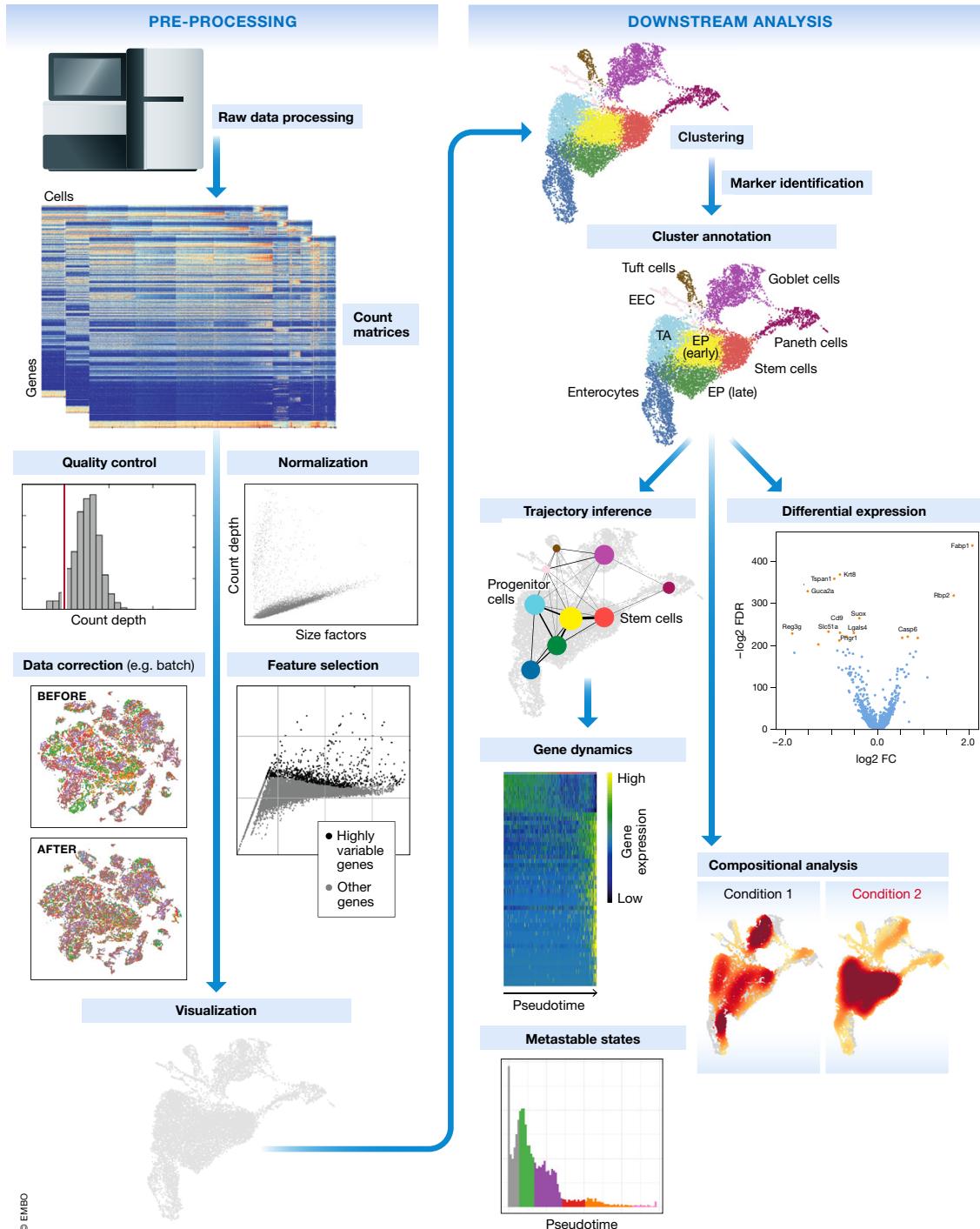
Standard Single-Cell RNA-seq Workflow

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

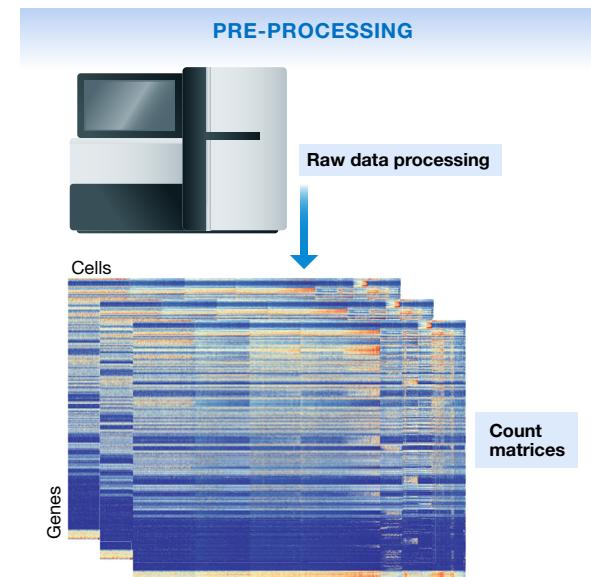
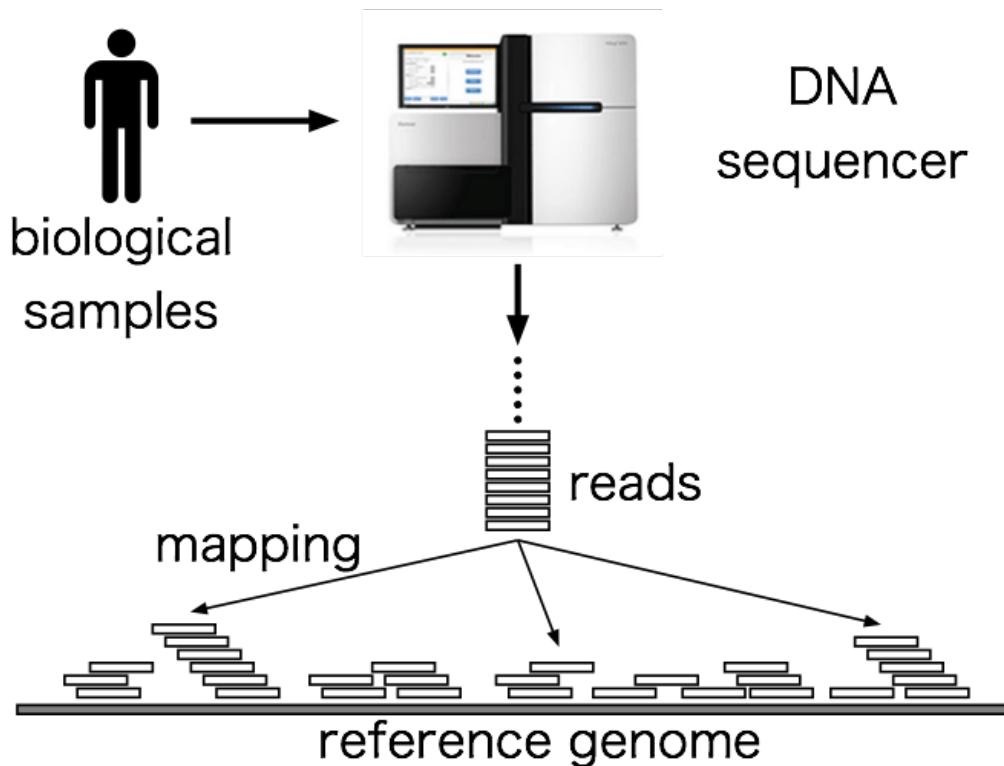


Standard Single-Cell RNA-seq Workflow

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



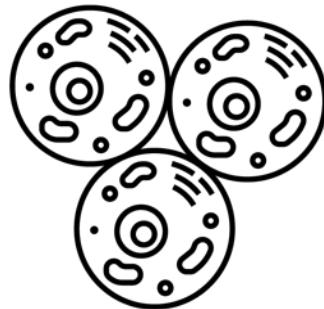
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering

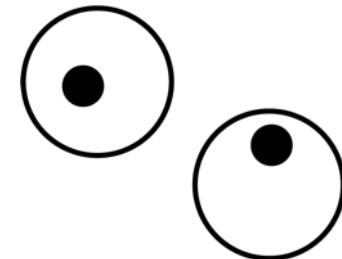
Dying cells



Multiplets



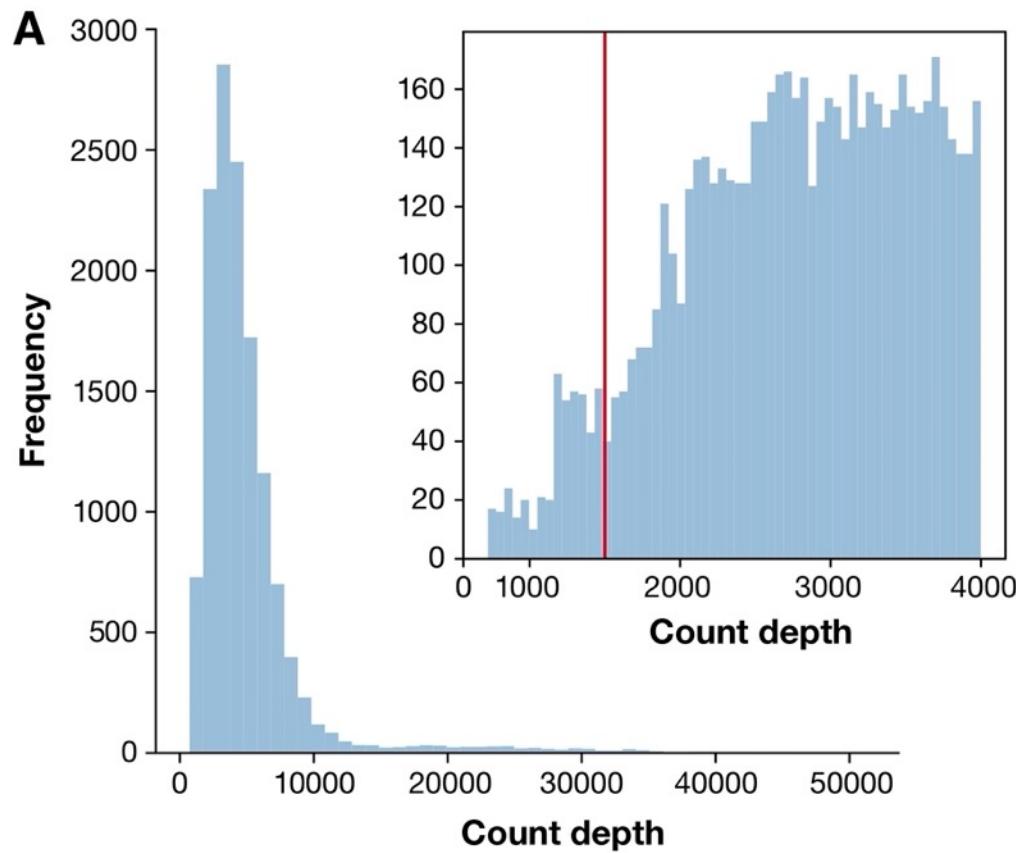
Empty Droplets



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

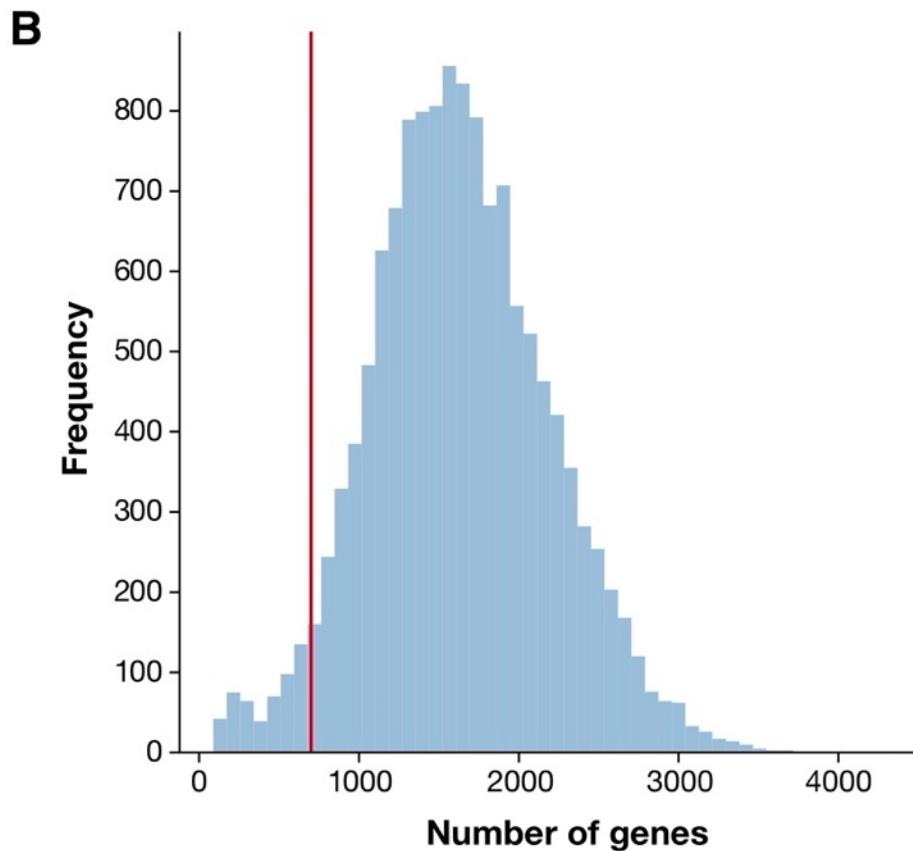
Top

Step 2 – Quality control and filtering



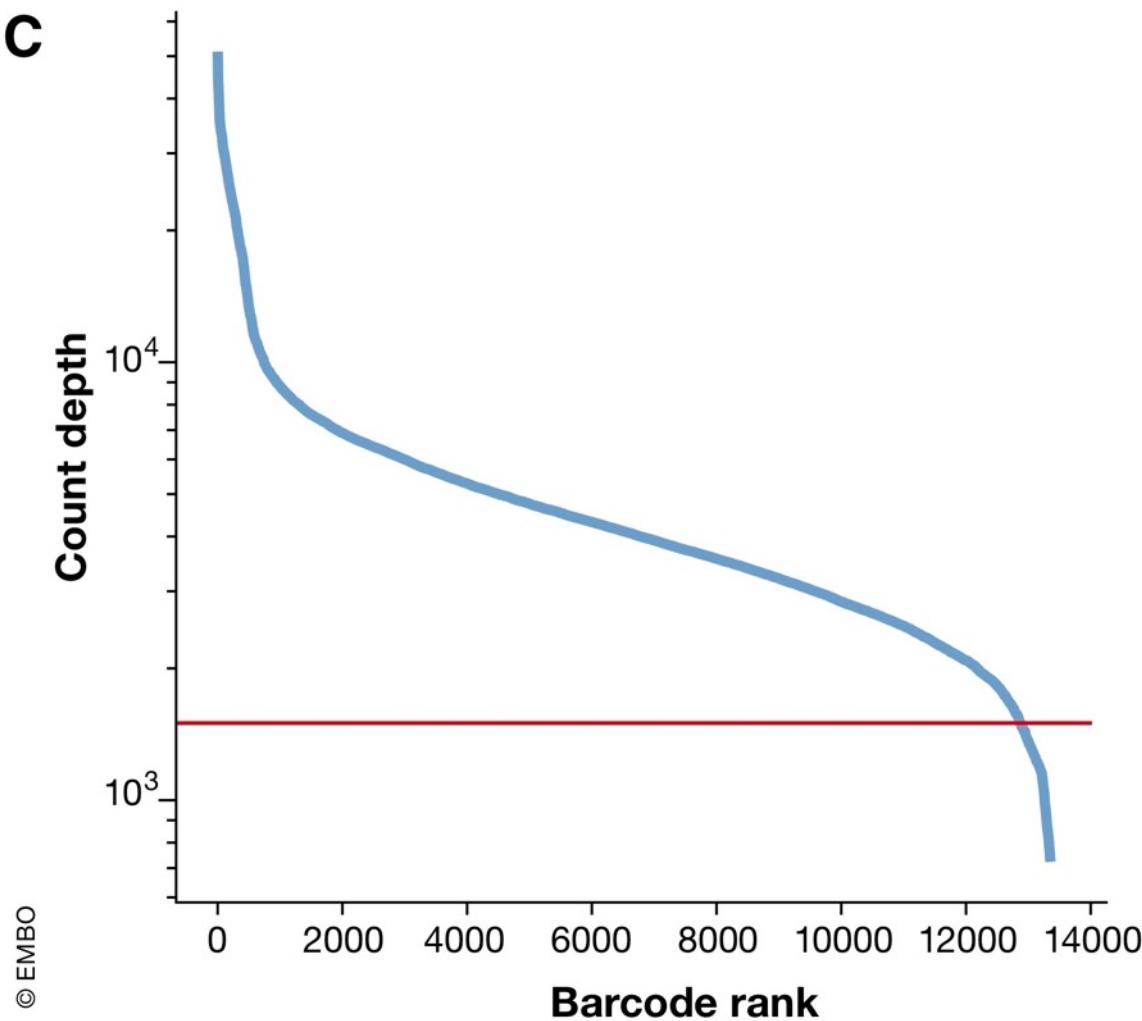
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



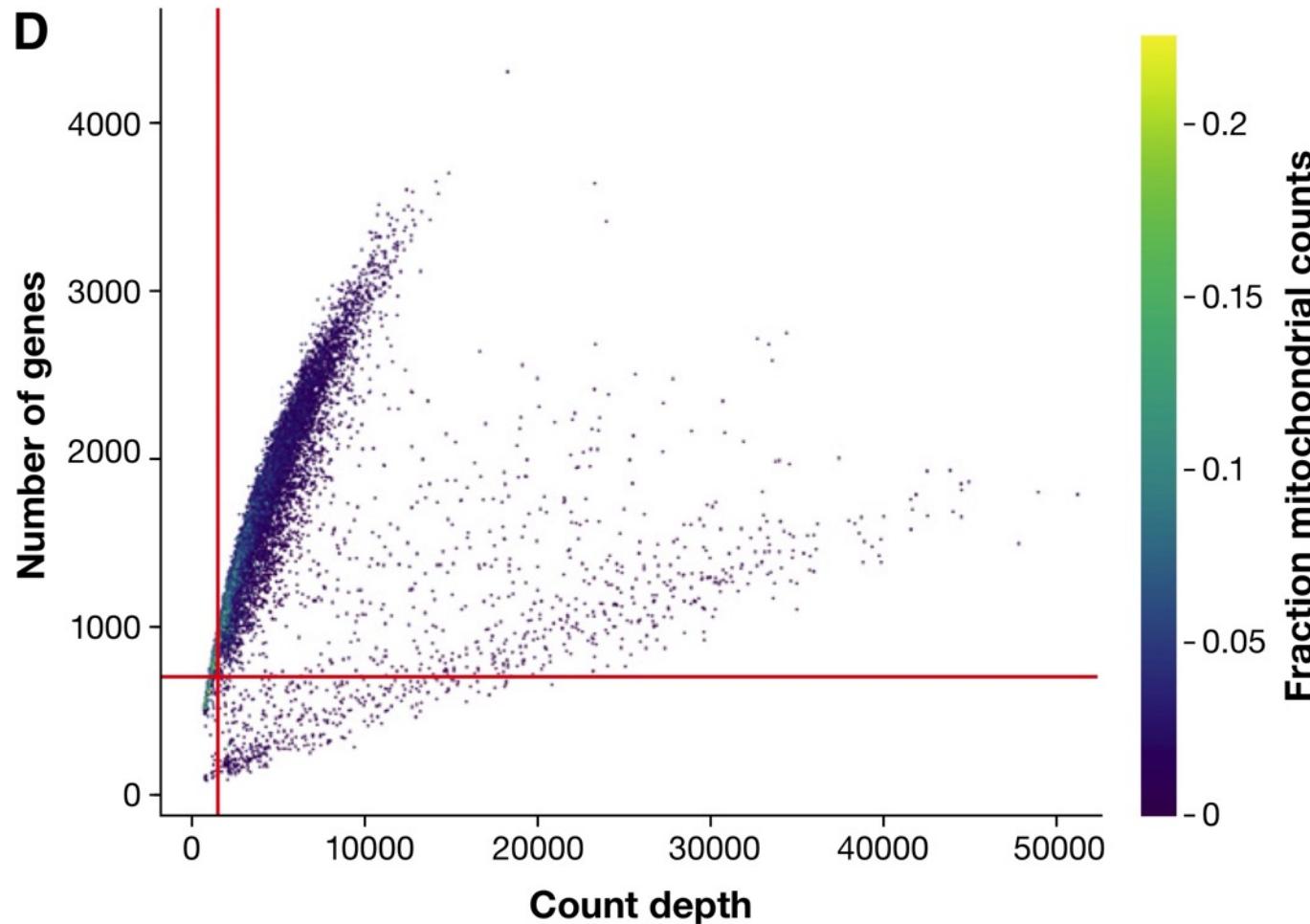
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



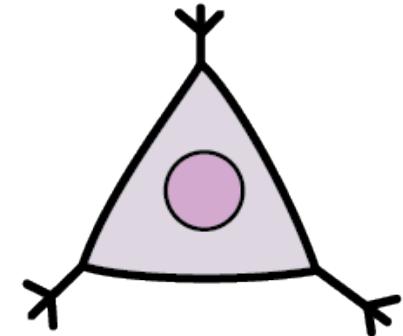
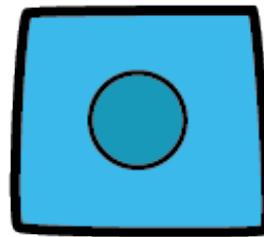
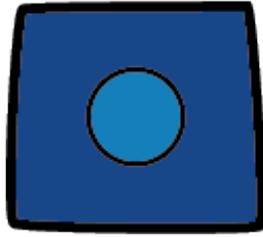
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 2 – Quality control and filtering



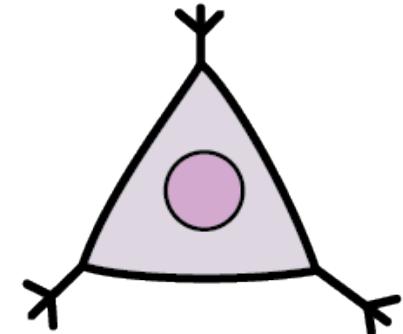
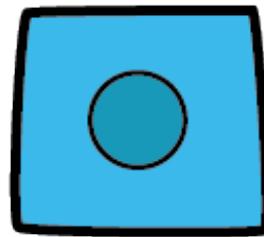
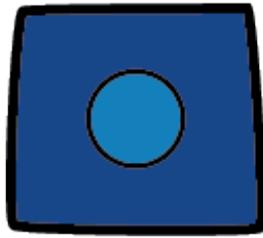
Building Counts Matrix → QC & Filtering → Normalization → Visualization → Analysis

Step 3 - Normalization



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

Step 3 - Normalization

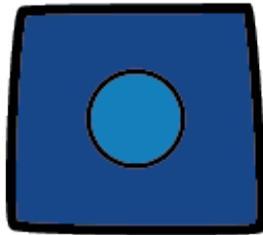


10% Capture Efficiency

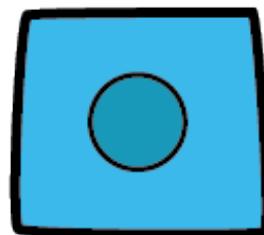
Gene	Cell A
X	10
Y	20
Z	70

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

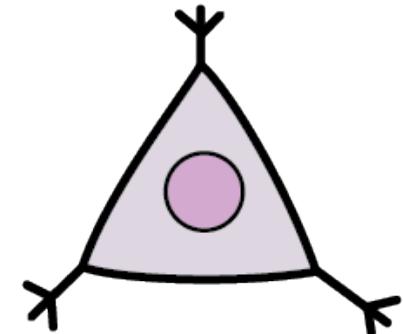
Step 3 - Normalization



10% Capture Efficiency



20% CE

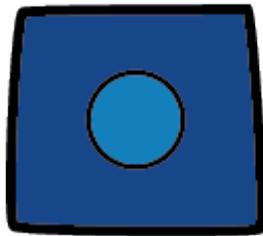


Gene	Cell A
X	10
Y	20
Z	70

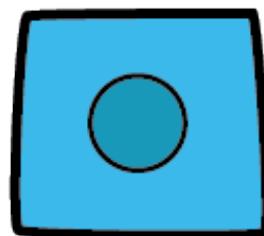
Gene	Cell B
X	20
Y	40
Z	140

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

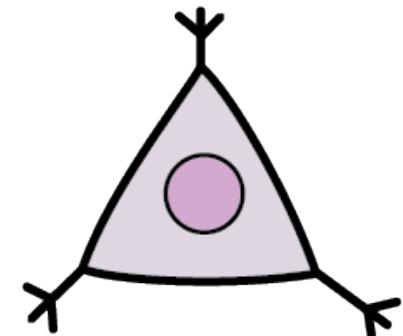
Step 3 - Normalization



10% Capture Efficiency



20% CE



20% CE

Gene	Cell A
X	10
Y	20
Z	70

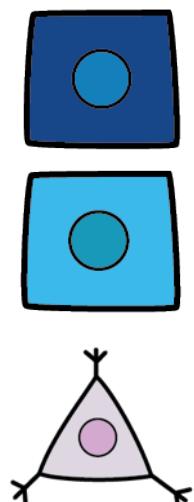
Gene	Cell B
X	20
Y	40
Z	140

Gene	Cell C
X	20
Y	0
Z	80

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

Step 3 - Normalization

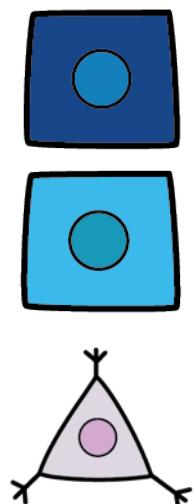
Raw counts



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

Step 3 - Normalization

Raw counts

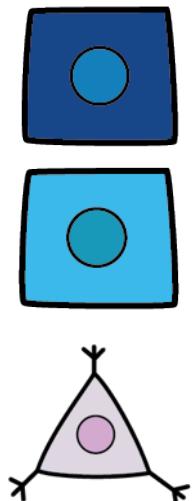


	X	Y	Z
A	10	20	70
B	20	40	140
C	20	0	80

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

Step 3 - Normalization

Raw counts



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

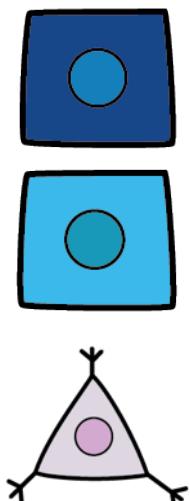
Pairwise distances

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

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

Step 3 - Normalization

Raw counts



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

Pairwise distances

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

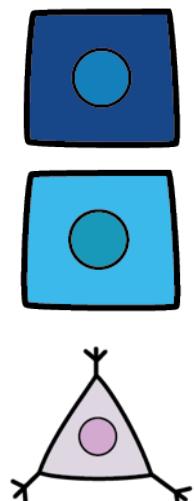
$$\text{dist}(A,C) = 24.5$$

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

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

Step 3 - Normalization

Raw counts

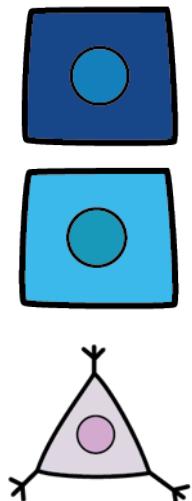


	X	Y	Z	Library Size	Pairwise distances
A	10	20	70	100	$\text{dist}(A,B) = 71.4$
B	20	40	140	200	$\text{dist}(A,C) = 24.5$
C	20	0	80	100	$\text{dist}(B,C) = 67.1$

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

Step 3 - Normalization

Normalized counts

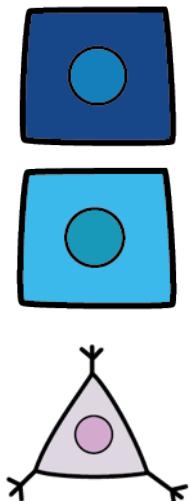


	X	Y	Z	Library Size	Pairwise distances
A	0.1	0.2	0.7	100	$\text{dist}(A,B) = 71.4$
B	0.1	0.2	0.7	200	$\text{dist}(A,C) = 24.5$
C	0.2	0	0.8	100	$\text{dist}(B,C) = 67.1$

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

Step 3 - Normalization

Normalized counts

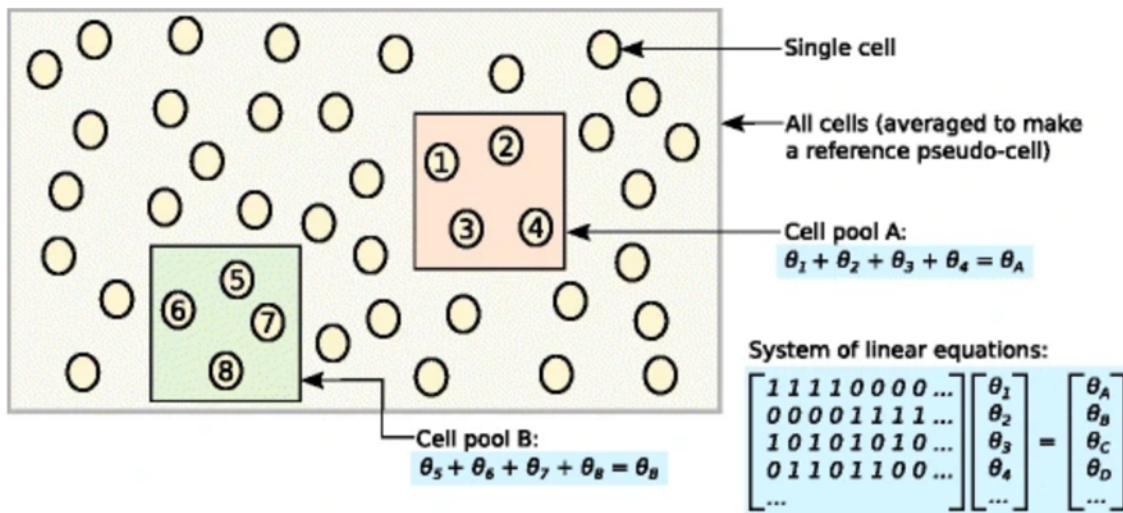


	X	Y	Z	Library Size	Pairwise distances
A	0.1	0.2	0.7	100	$\text{dist}(A,B) = 0$
B	0.1	0.2	0.7	200	$\text{dist}(A,C) = 0.25$
C	0.2	0	0.8	100	$\text{dist}(B,C) = 0.25$

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

More complex normalization approaches exist

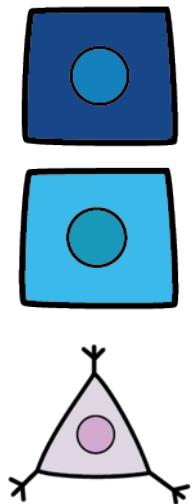
Fig. 3



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

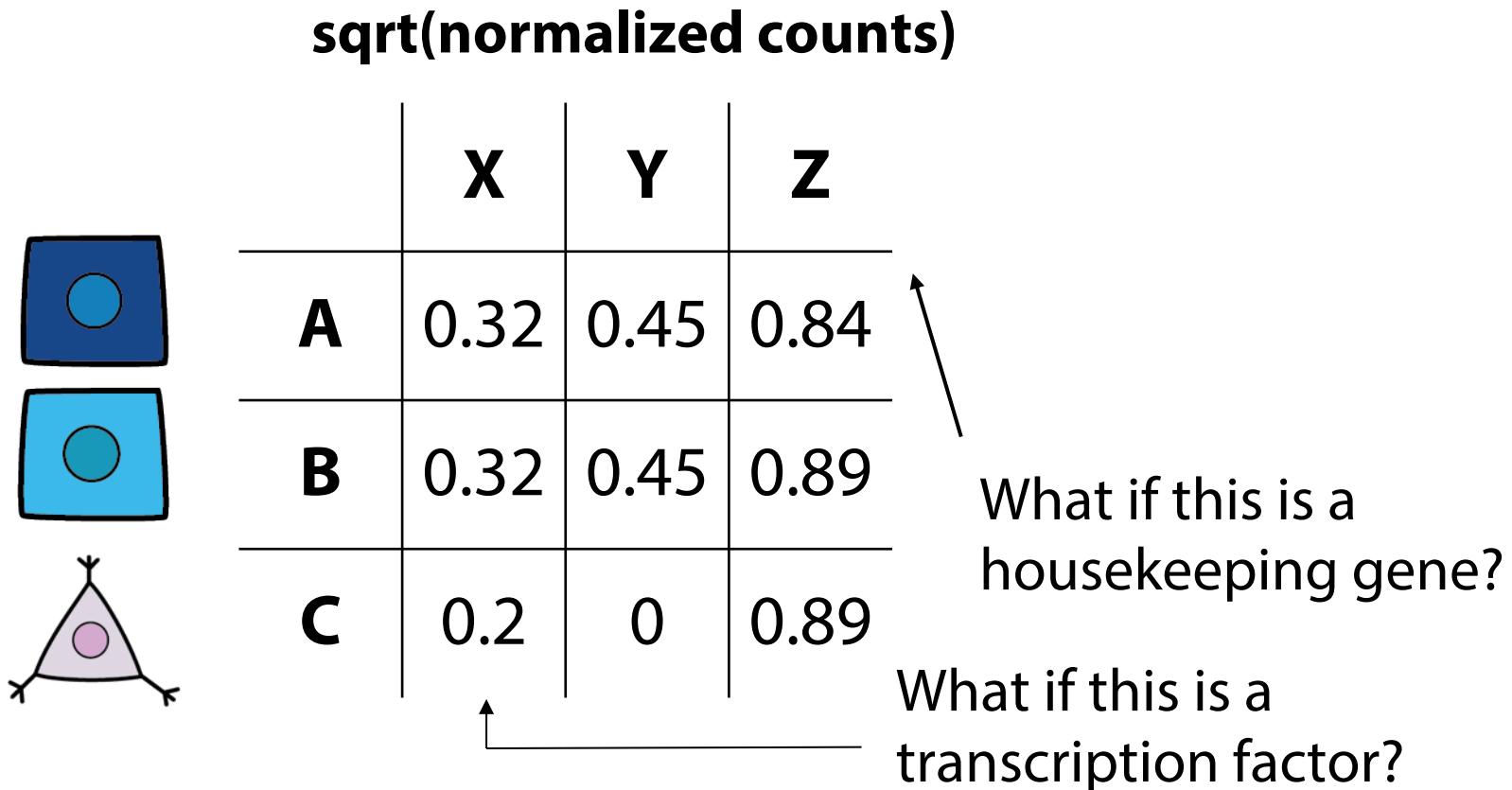


	X	Y	Z
A	0.1	0.2	0.7
B	0.1	0.2	0.7
C	0.2	0	0.8

What if this is a housekeeping gene?

What if this is a transcription factor?

Step 3.5 – Transformation / Scaling



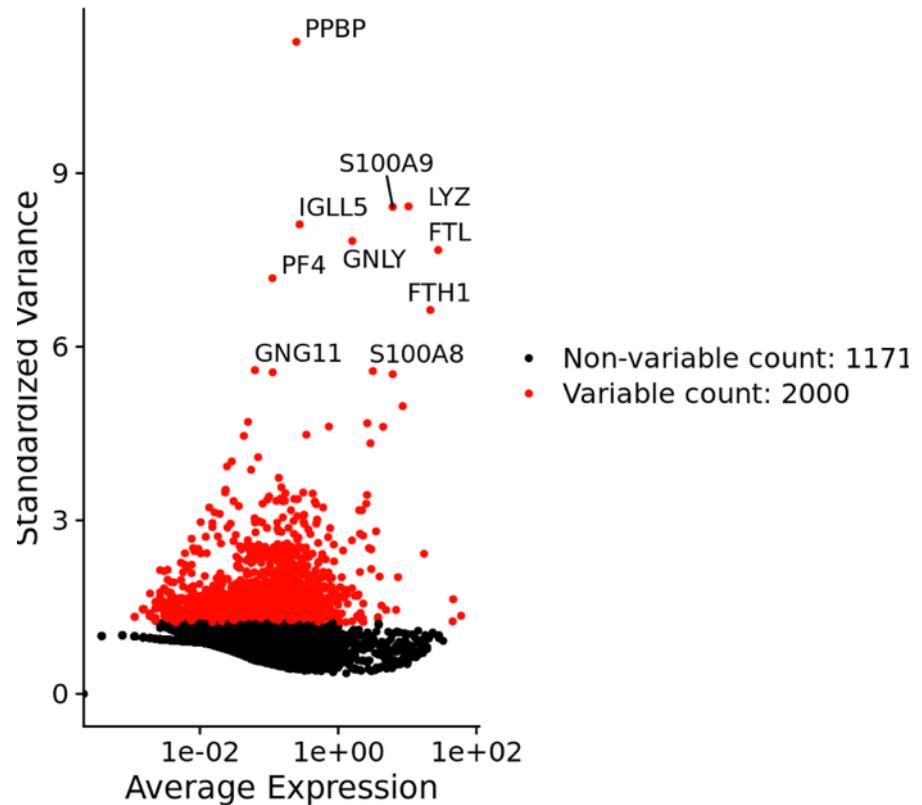
What kind of transformations, other than square-root, could we apply to single cell data?

Top

Step 5 – Dimensionality reduction and visualization

Selecting highly variable genes (HVGs):

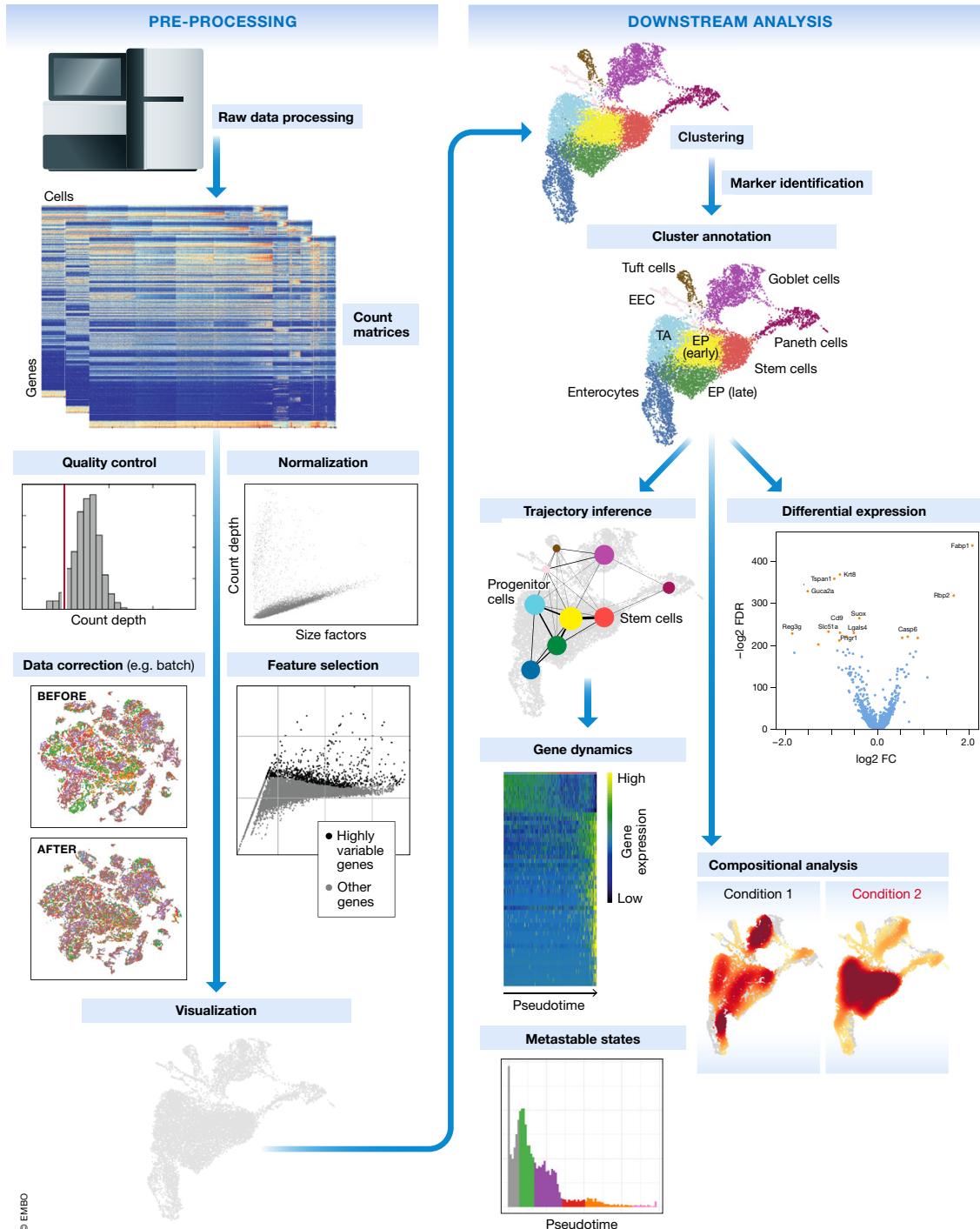
- Calculate log10 mean expression and variance
- Fit a loess curve
- Standardize variance to mean 0 std 1
- Take the top 2000 HVGs



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

