

Bioinformatics Software Lab

Introduction to Analysis of Single Cell Sequencing

Ivan Gesteira Costa, Mingbo Cheng, Martin Manolov, James Nagai, Mina Shaigon
Institute for Computational Genomics

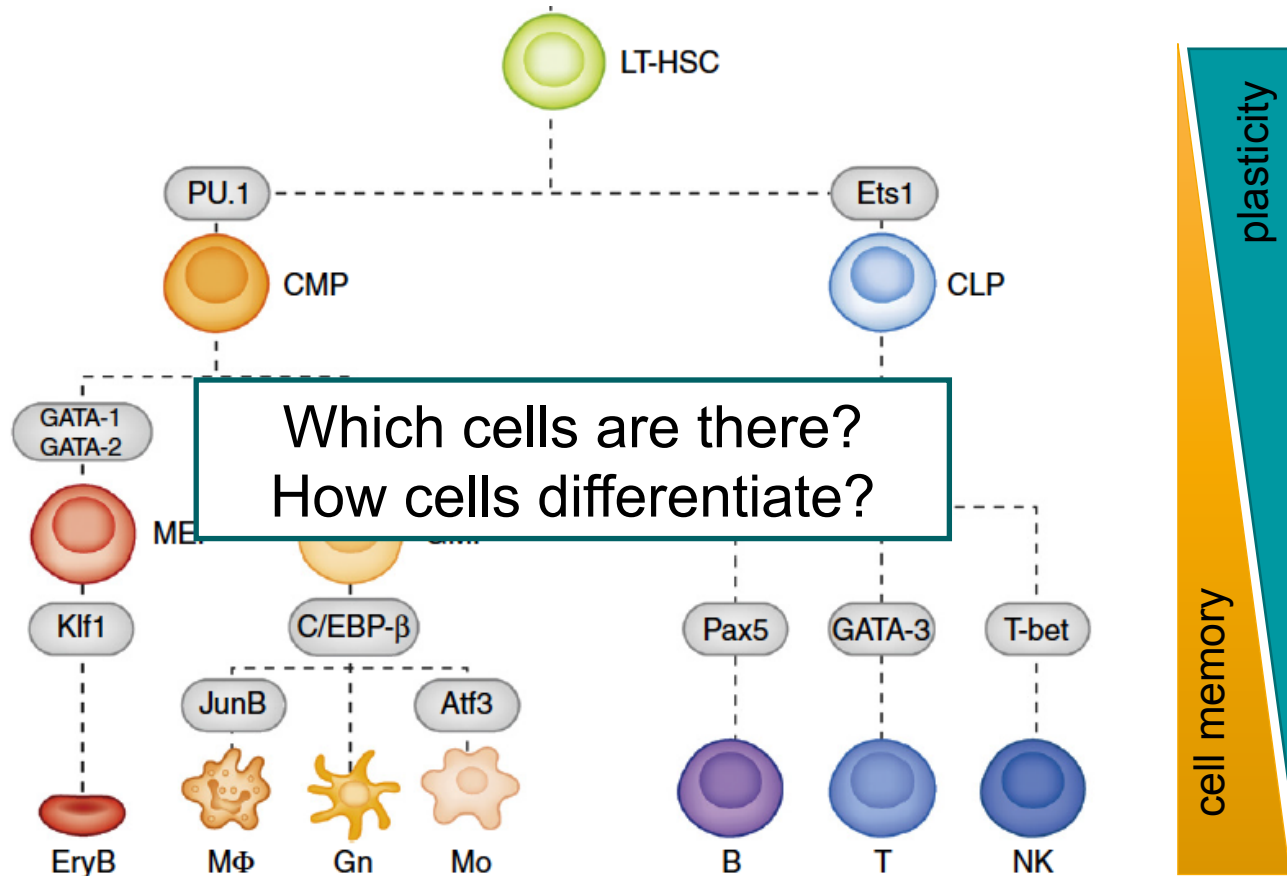
Objectives

- 1. basics of single cell sequencing**
- 2. basic bioinformatics/computational problems**
 - dimension reduction**
 - clustering**
 - data integration**

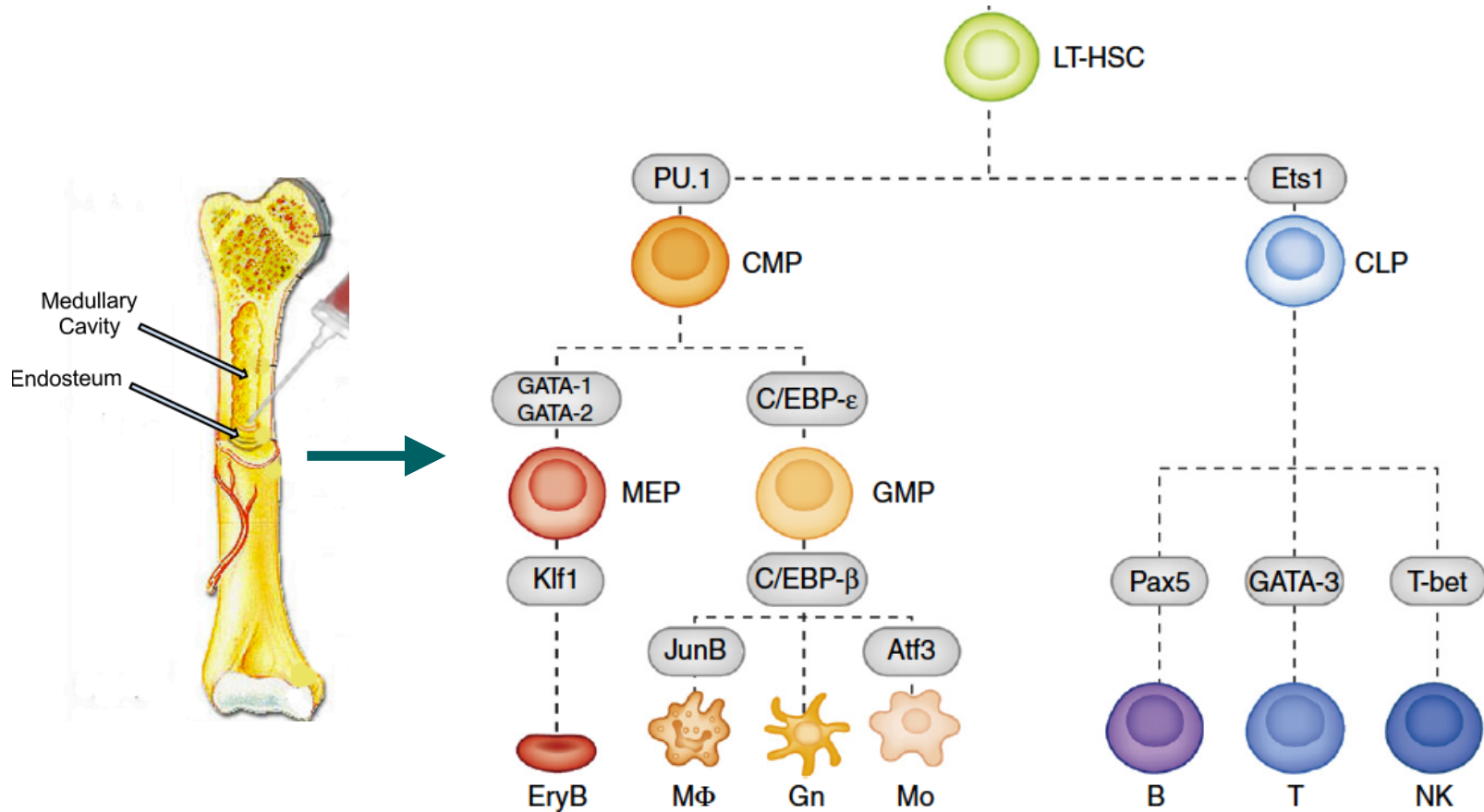
Expression at Single Cell Level

Cell Differentiation

Hematopoiesis

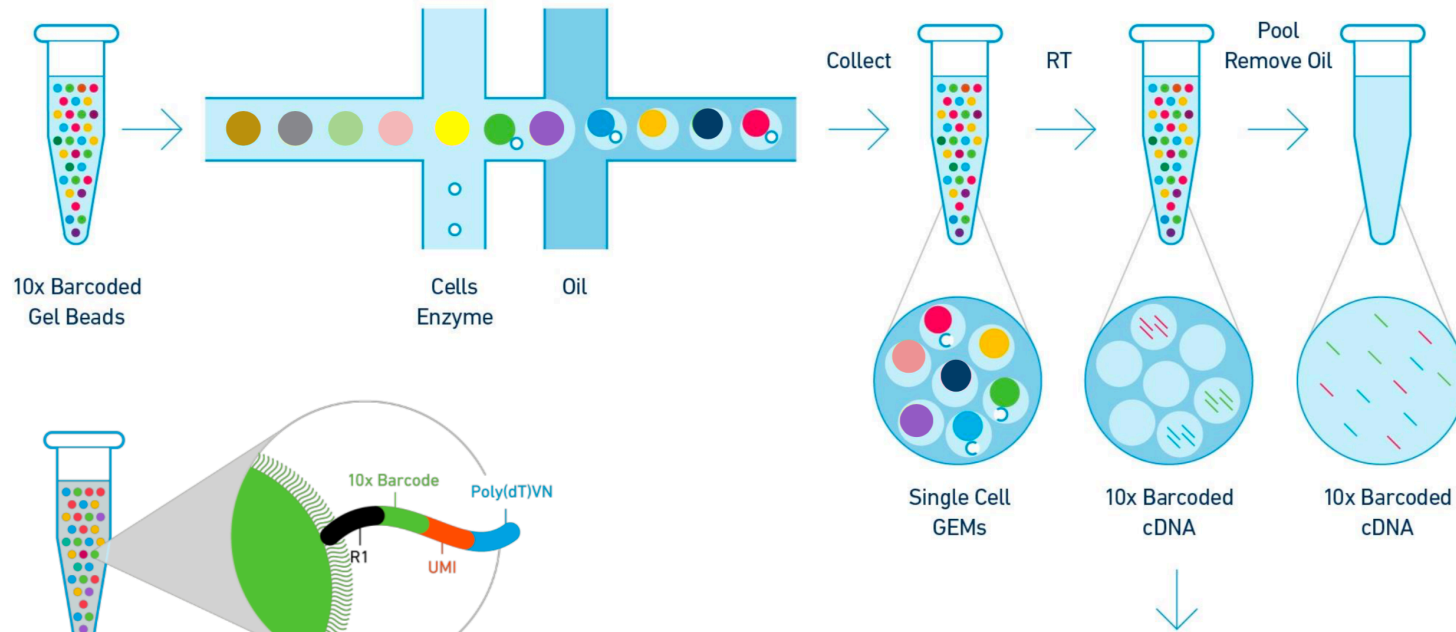


Cell Differentiation



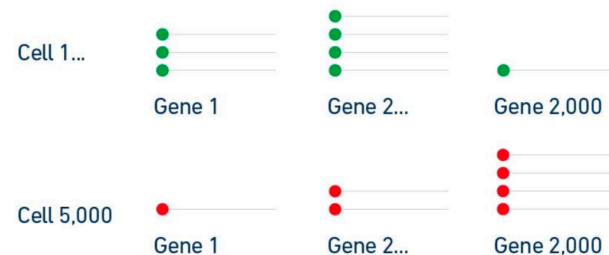
Source: Amit (2016), *Nature Immunology*.

Droplet based RNA single cell sequencing



- Input: Single cells in suspension + 10x Gel Beads and Reagents
- Output: Digital gene expression profiles from every partitioned cell

Transcriptional profiling of individual cells



Droplet based RNA single cell sequencing

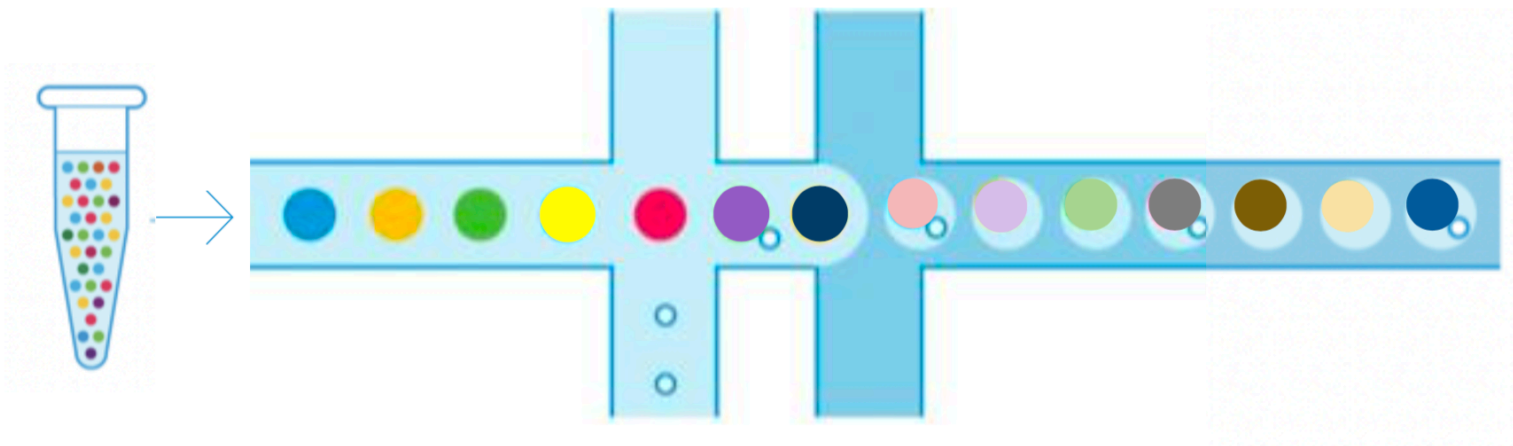


↑
Gel Beads

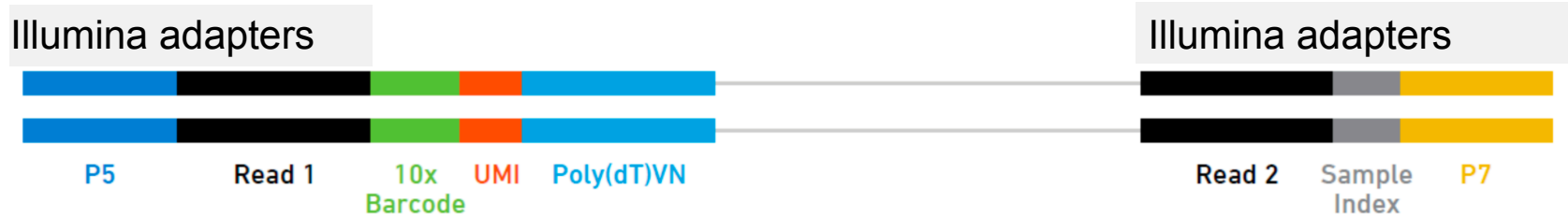
↑
Sample

↑
Oil

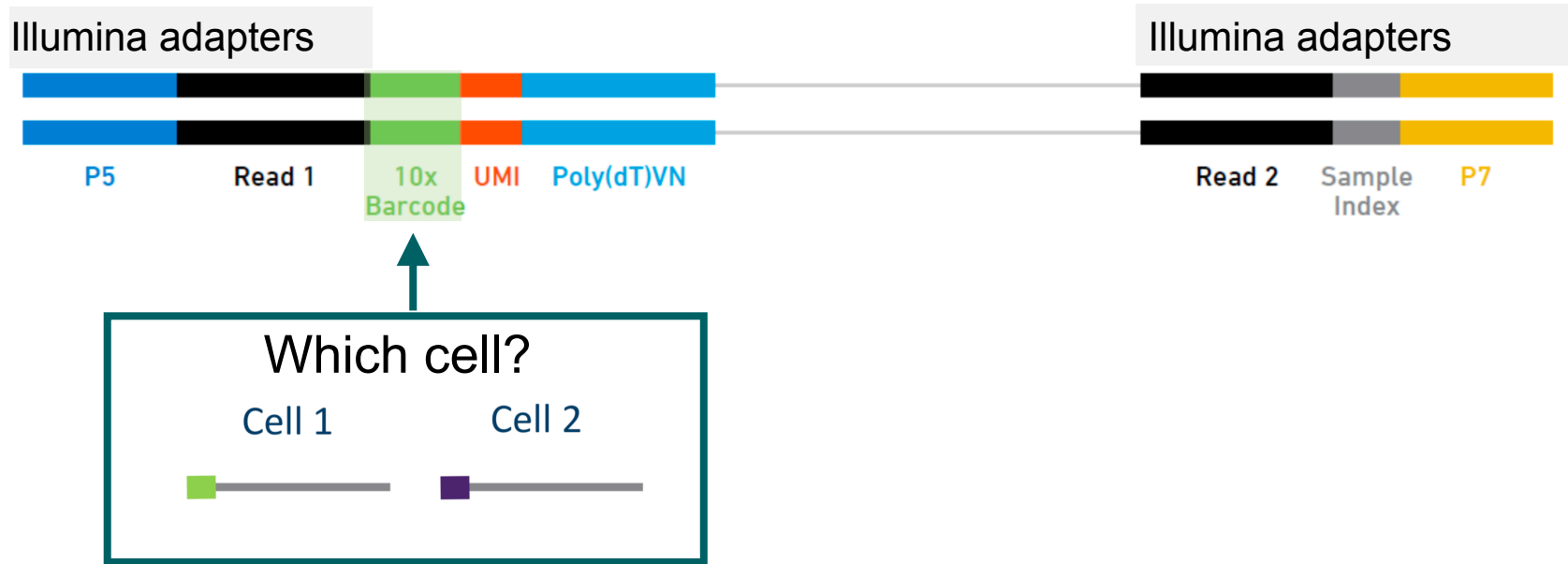
↑
Droplets with Gel Beads



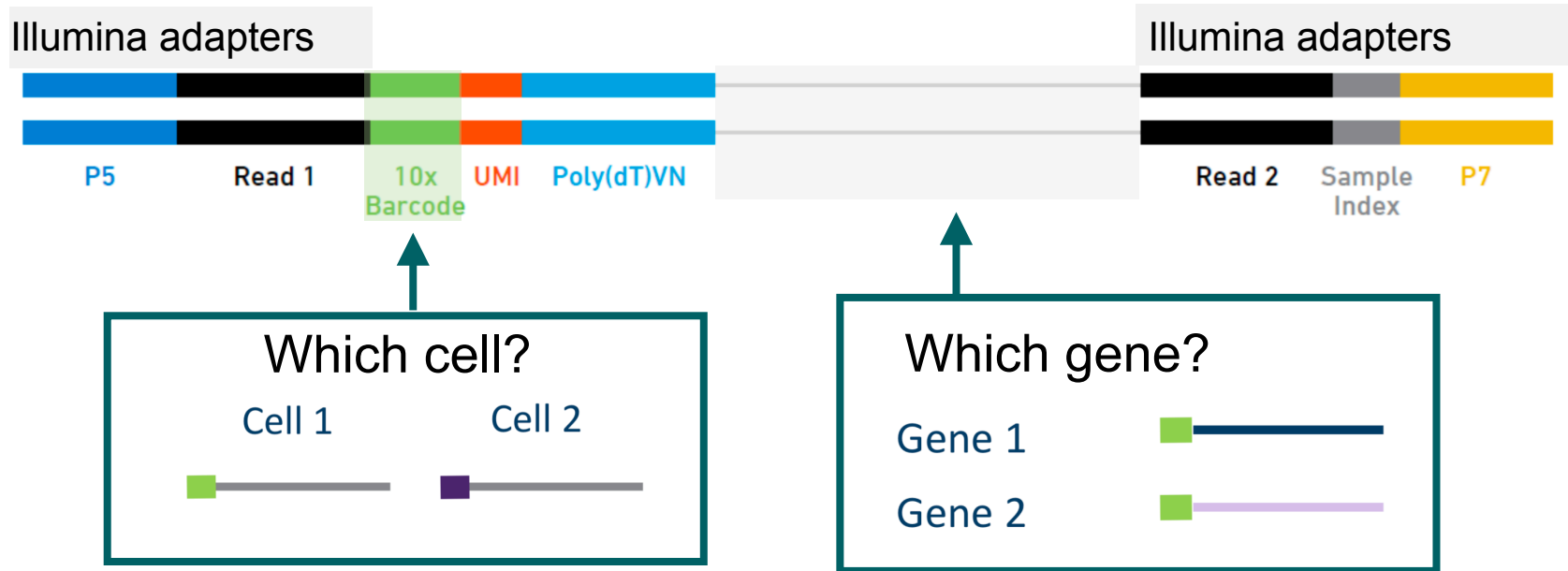
Basics Bioinformatics - Transcript Counts



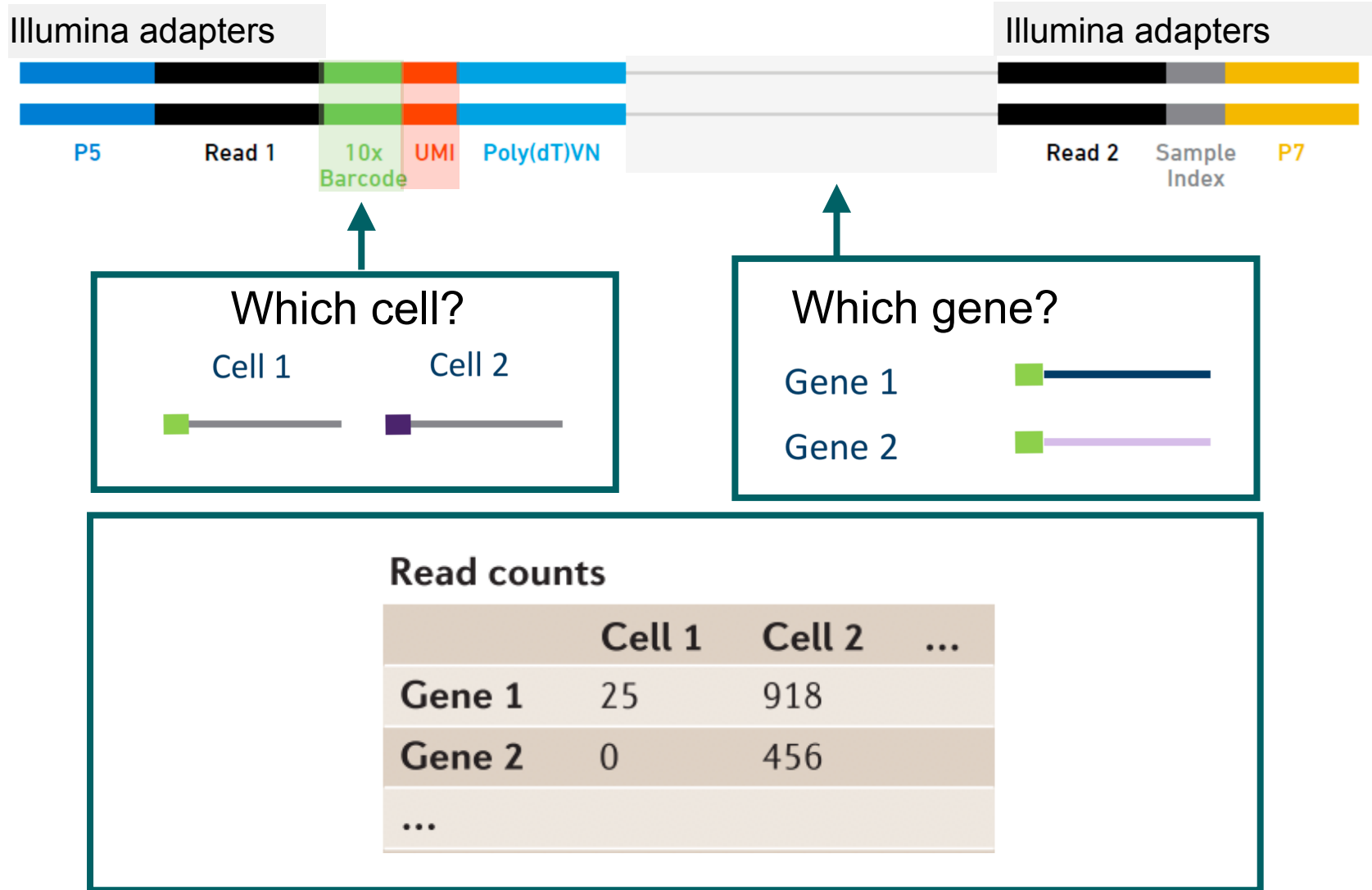
Basics Bioinformatics - Transcript Counts



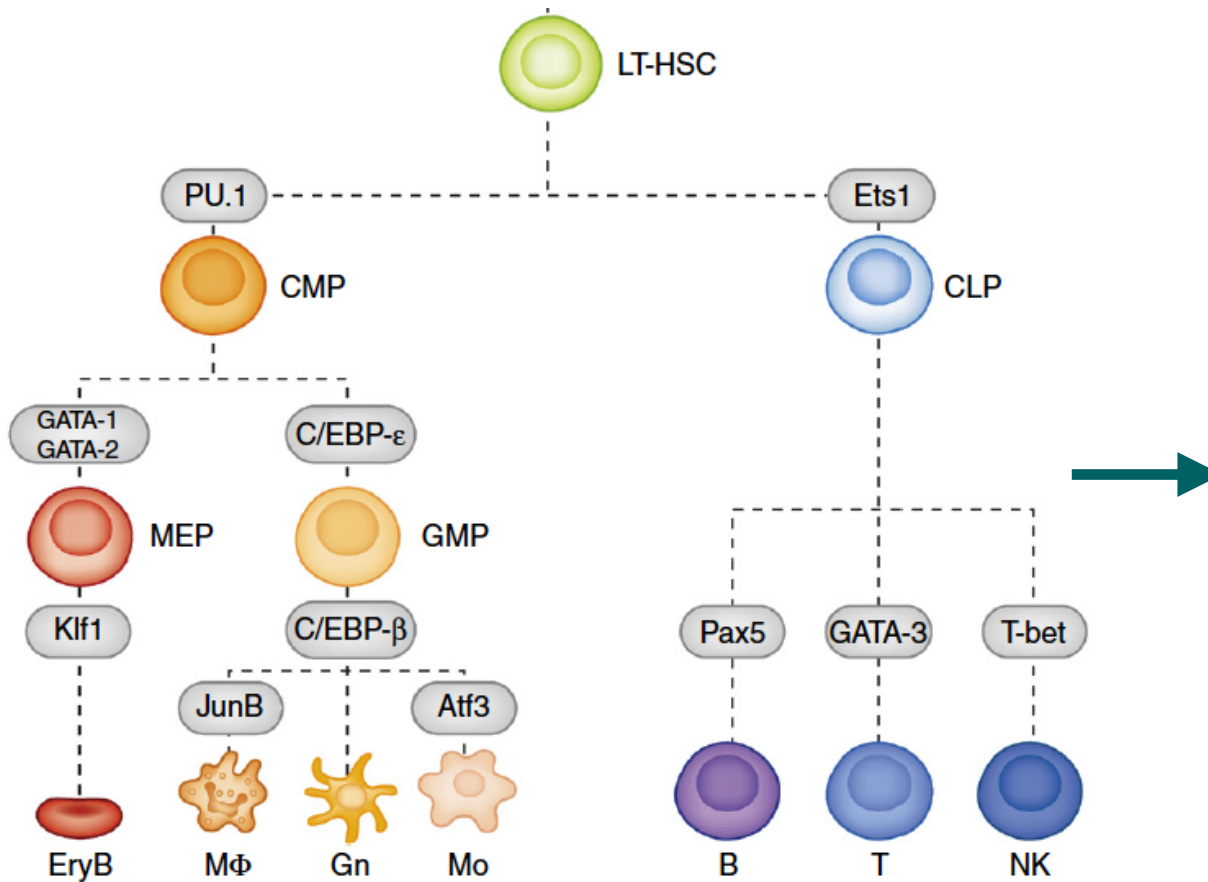
Basics Bioinformatics - Transcript Counts



Basics Bioinformatics - Transcript Counts



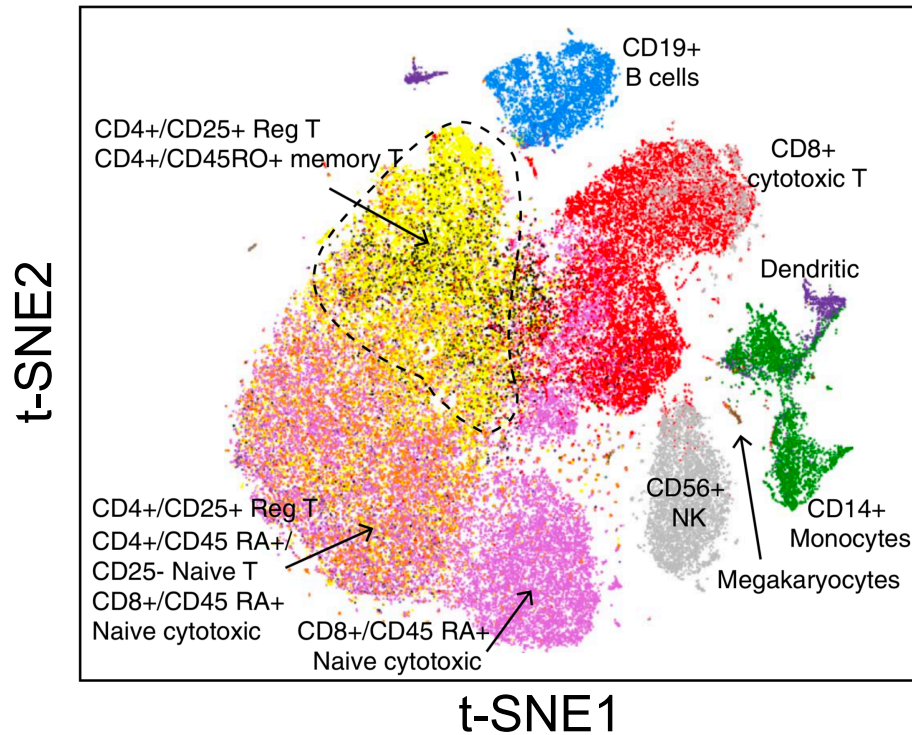
Cell Differentiation & Gene Expression



	Cell 1	Cell 2	...
Gene 1	25	918	
Gene 2	0	456	
Gene 3	20	342	
Gene 4	0	214	
...			

Gene Expression of Lymphoid Cells

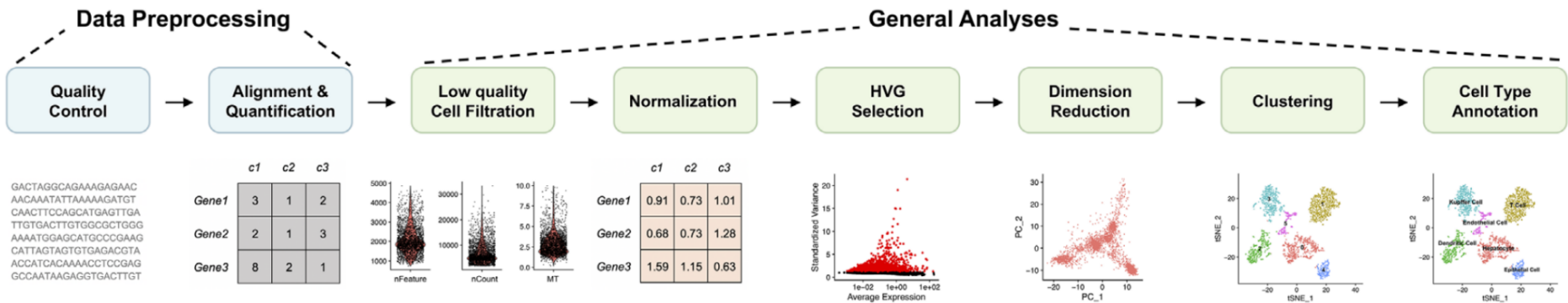
PBMCs from Humans



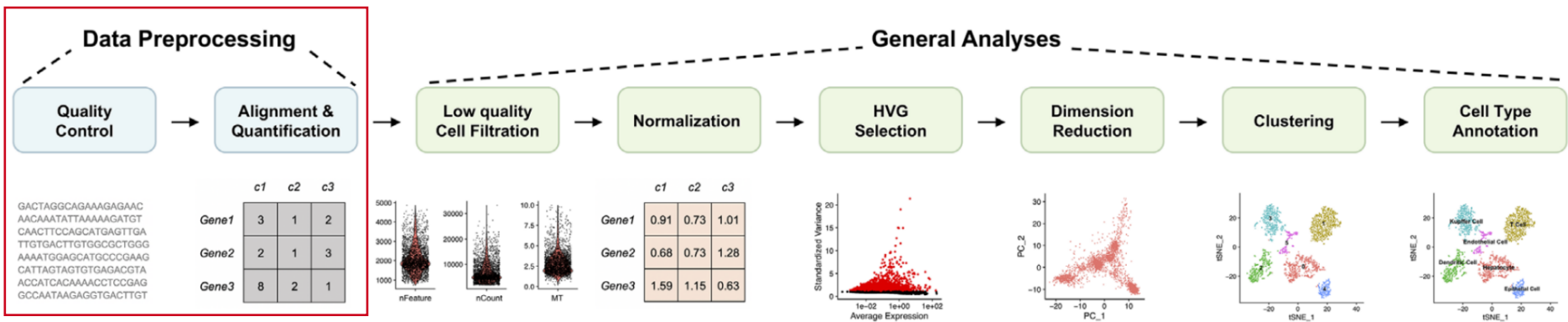
Single cell RNA-seq from 68k cells

Source: Zheng et al. 2017 & Buenrostro et al. 2018

Basics Bioinformatics - single cell RNA-seq

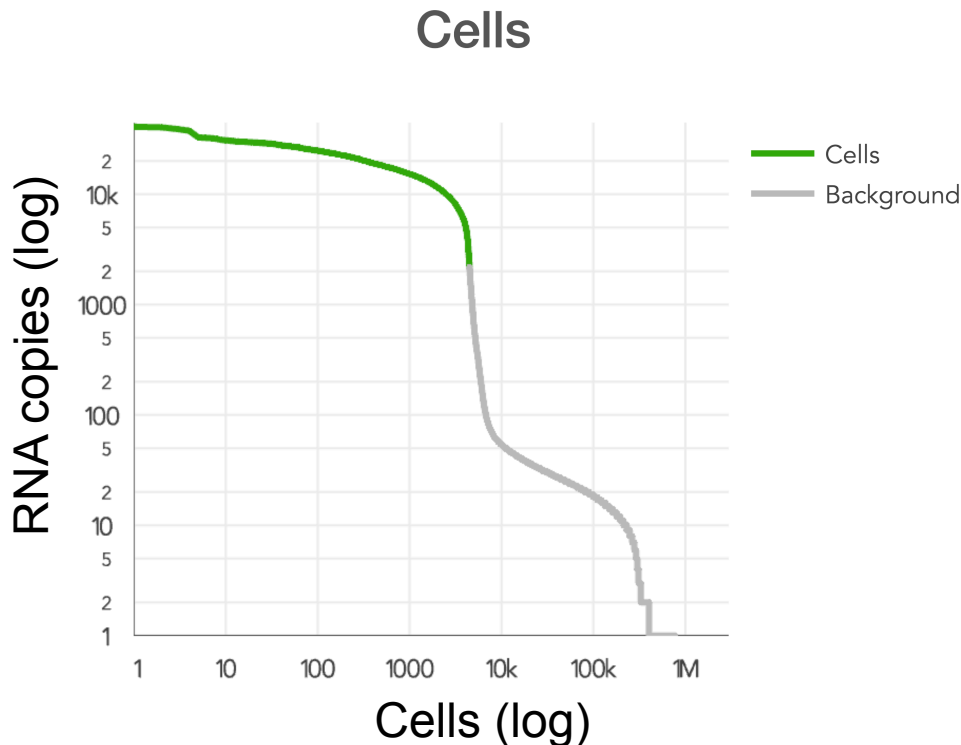


Basics Bioinformatics - single cell RNA-seq



Basics Bioinformatics - Cell Filtering

1. sum UMIs (copy of transcripts) per cell
2. consider cells with total UMI count > 99th of expected recovered cells



Estimated Number of Cells

4,495

Post-Normalization Mean
Reads per Cell

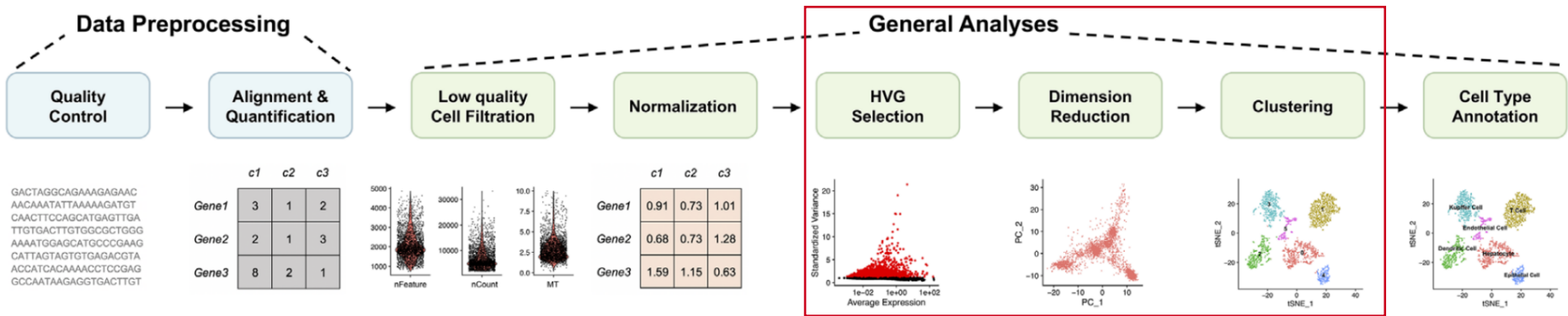
89,289

Median Genes per Cell

2,504

cell ranger - 10x genomics

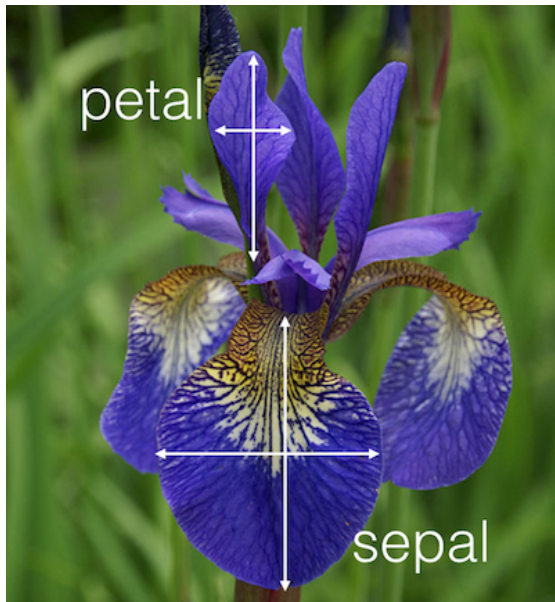
Basics Bioinformatics - single cell RNA-seq



Clustering & Dimension reduction

Clustering

- **Given a data description**
 - i.e. measurement of size of iris flowers
- **Find groups of similar observations**
 - i.e. iris flower sub-types

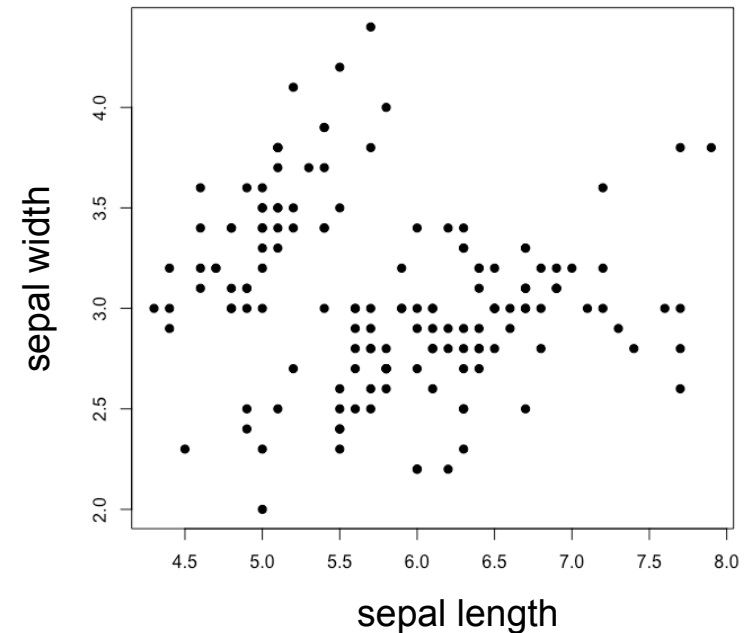


	<i>Sepal Length</i>	<i>Sepal Width</i>	<i>Petal Length</i>	<i>Petal Width</i>
<i>Flower 1</i>	5.1	3.5	1.4	0.2
<i>Flower 2</i>	4.9	3.0	1.4	0.2
<i>Flower 3</i>	4.7	3.2	1.3	0.2
<i>Flower 4</i>	4.6	3.1	1.5	0.2
...	...	...	...	...

Clustering

- **Given a data description**
 - i.e. measurement of size of iris flowers
- **Find groups of similar observations**
 - i.e. iris flower sub-types

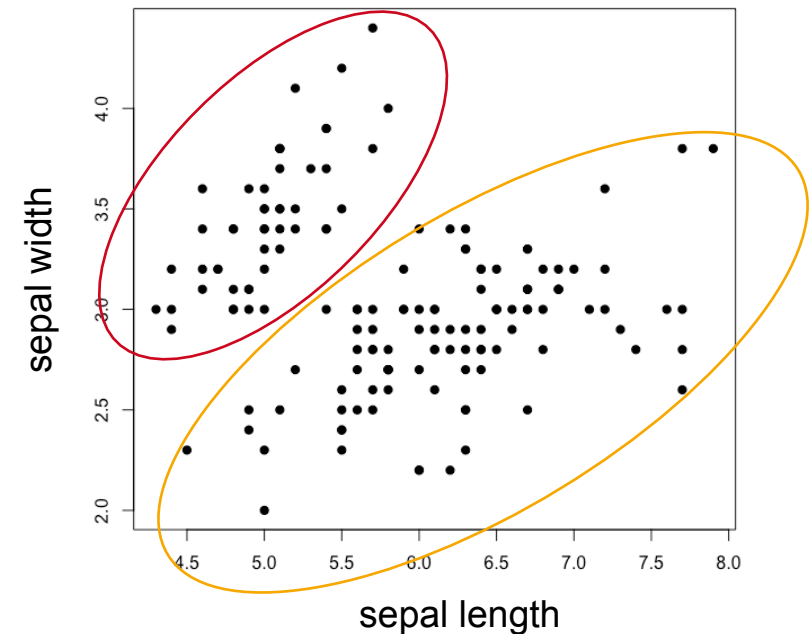
	<i>Sepal Length</i>	<i>Sepal Width</i>	<i>Petal Length</i>	<i>Petal Width</i>
<i>Flower 1</i>	<i>5.1</i>	<i>3.5</i>	<i>1.4</i>	<i>0.2</i>
<i>Flower 2</i>	<i>4.9</i>	<i>3.0</i>	<i>1.4</i>	<i>0.2</i>
<i>Flower 3</i>	<i>4.7</i>	<i>3.2</i>	<i>1.3</i>	<i>0.2</i>
<i>Flower 4</i>	<i>4.6</i>	<i>3.1</i>	<i>1.5</i>	<i>0.2</i>
...	...	...	...	...



Clustering

- **Given a data description**
 - i.e. measurement of size of iris flowers
- **Find groups of similar observations**
 - i.e. iris flower sub-types

	<i>Sepal Length</i>	<i>Sepal Width</i>	<i>Petal Length</i>	<i>Petal Width</i>
<i>Flower 1</i>	5.1	3.5	1.4	0.2
<i>Flower 2</i>	4.9	3.0	1.4	0.2
<i>Flower 3</i>	4.7	3.2	1.3	0.2
<i>Flower 4</i>	4.6	3.1	1.5	0.2
...	...	...	...	...

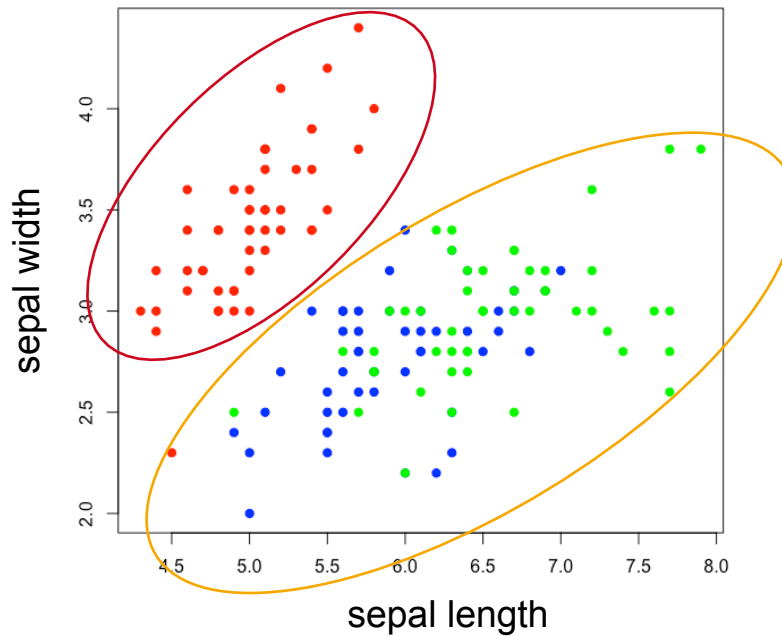


Clustering

- **Given a data description**
 - i.e. measurement of size of iris flowers
- **Find groups of similar observations**
 - i.e. iris flower sub-types



Iris Setosa



Iris Virginia



Iris Versicolor

Clustering Formalism

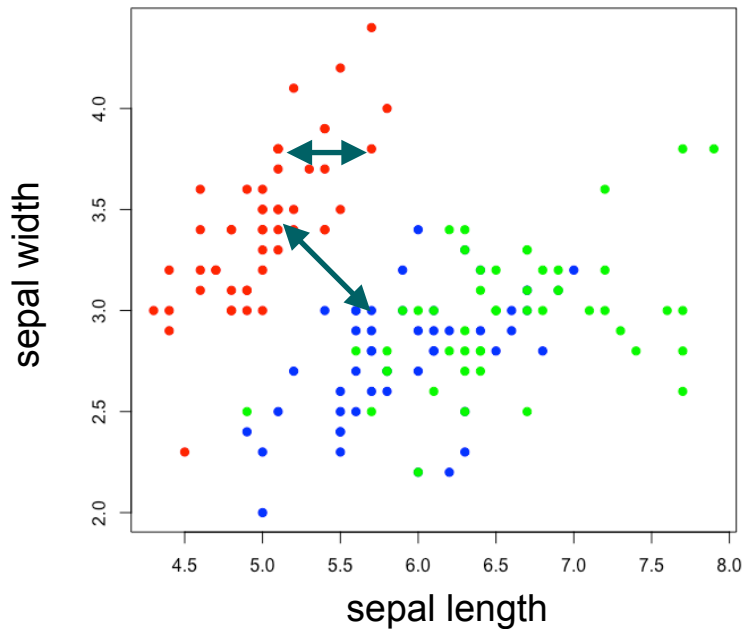
- **For a given data:**
 - Matrix X with N observations and L dimensions
where x_i is a vector representing observation i

X_{11}	X_{12}	$\dots$	X_{1L}
X_{21}	X_{22}	$\dots$	X_{2L}
X_{31}	X_{32}	$\dots$	X_{3L}
$\dots$	$\dots$	$\dots$	$\dots$
X_{N1}	X_{N2}	$\dots$	X_{NL}

- **find groups of similar observations**
 - vector $Y = (y_1, \dots, y_N)$
where $y_i \in \{1, \dots, K\}$ indicates the cluster of observation i

Distance

- A important concept in clustering is a distance (similarity) between a pair of objects x_i and x_j
 - Observations of a same group should be close in space

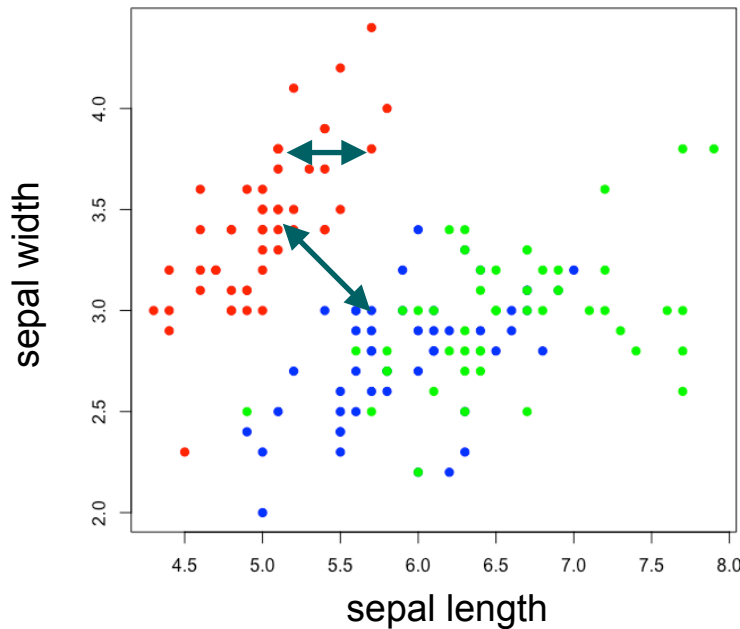


Euclidean distance
(sensitive to scale)

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^L (x_{il} - x_{jl})^2}$$

Distance

- A important concept in clustering is a distance (similarity) between a pair of objects x_i and x_j
 - Observations of a same group should be close in space



Euclidean distance
(sensitive to scale)

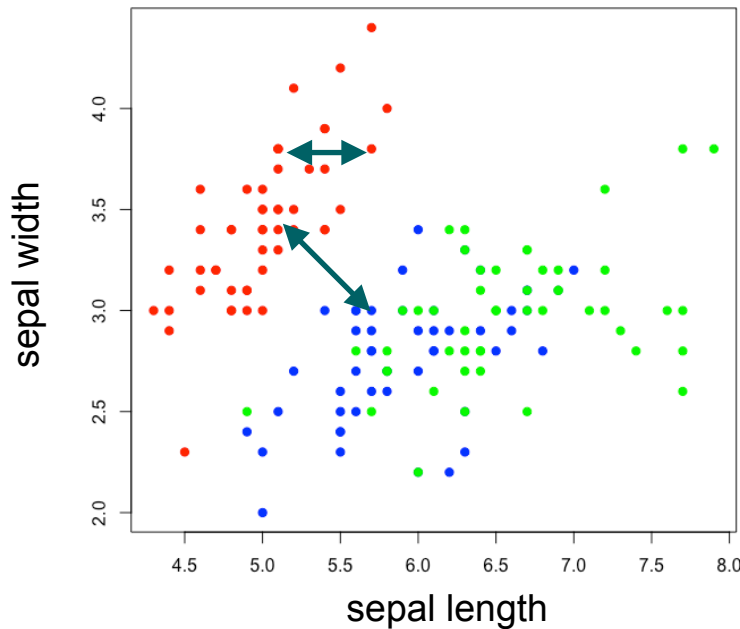
$$d(x_i, x_j) = \sqrt{\sum_{l=1}^L (x_{il} - x_{jl})^2}$$

Pearson Correlation
(scale insensitive/ similarity)

$$d(x_i, x_j) = \frac{\sum_{l=1}^L (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sigma_i^2 \sigma_j^2}$$

Distance

- A important concept in clustering is a distance (similarity) between a pair of objects x_i and x_j
 - Observations of a same group should be close in space



Euclidean distance
(sensitive to scale)

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^L (x_{il} - x_{jl})^2}$$

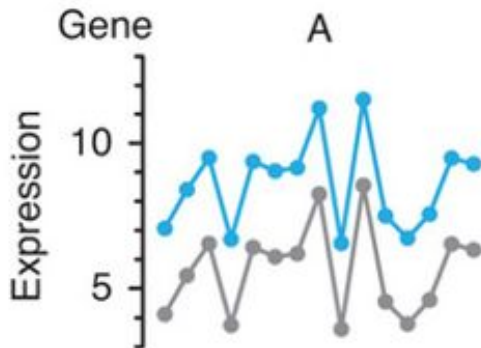
Pearson Correlation
(scale insensitive/ similarity)

$$d(x_i, x_j) = \frac{\sum_{l=1}^L (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sigma_i^2 \sigma_j^2}$$

Distance and Scale

- **In some problems scale can be important!**
 - Similarly in changes are more important / not absolute values.

unscaled data



Euclidean - not similar
Correlation - similar

z-score normalised data



Euclidean - similar
Correlation - similar

Clustering Methods

- **Hierarchical methods**
 - Mostly bottom up
 - based on distance / simple to interpret
- **Partitional methods (k-means or mixture models)**
 - Mostly top down
 - Use models of groups, centroids
- **Graph based methods**
 - Use graph formalisms to represent data:
 - nodes are objects
 - edges weights represent similarities
 - find well connected graphs

K-means

Iterative algorithm using **centroids** as cluster representations

Requires specification of number of clusters (**K**)

Algorithm:

Start cluster (Y) randomly

Repeat for a number of iterations

- estimate centroid (m_k) for each cluster

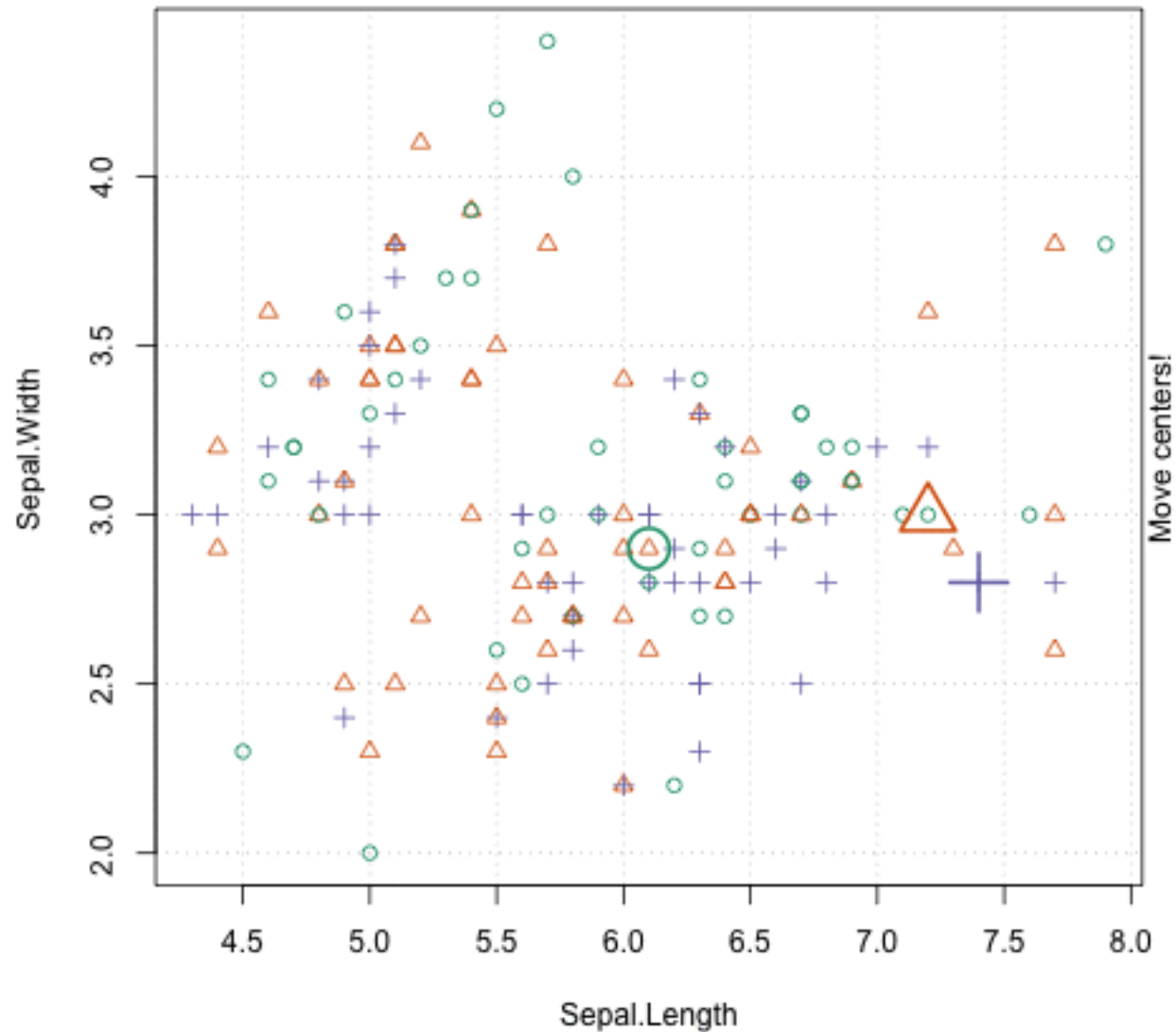
$$m_k = \frac{\sum_{i=1}^N 1(y_i = k) x_i}{\sum_{i=1}^N 1(y_i = k)}$$

- Assign objects to closest centroid:

$$y_i = \operatorname{argmin}_k d(x_i, m_k)$$

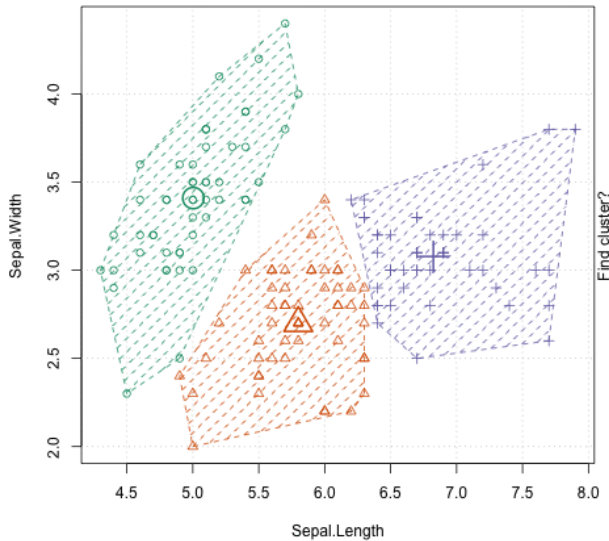
* convergence is only guaranteed for Euclidean distance

K-means on Iris

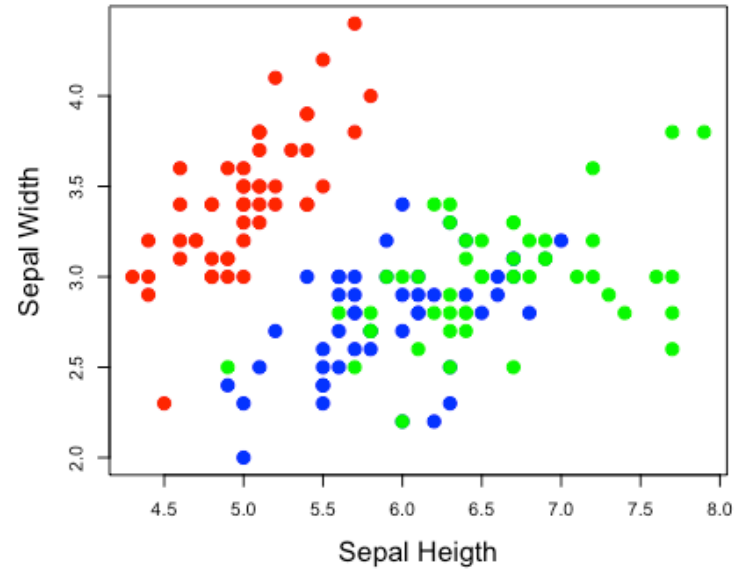


K-means on Iris

K-means solutions

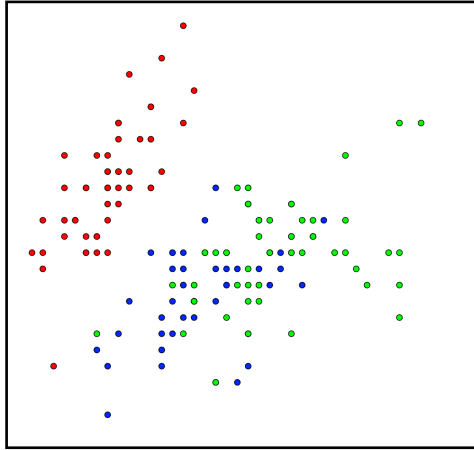


True labels



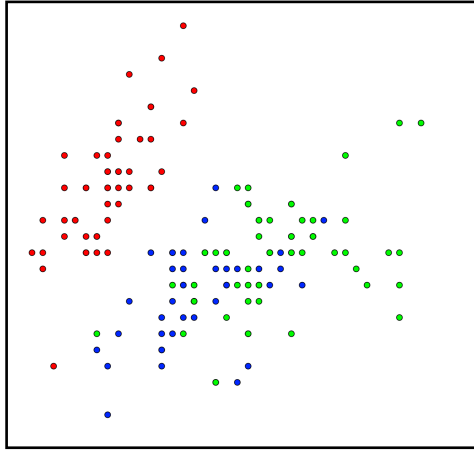
- K-means tends to find spherical clusters
- Sensitive to initialisation

Graph based clustering

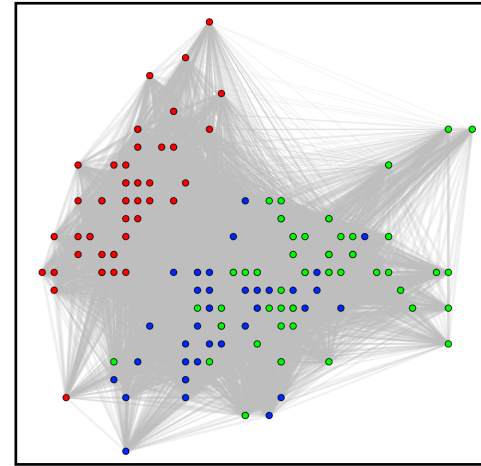


- data points are nodes

Graph based clustering

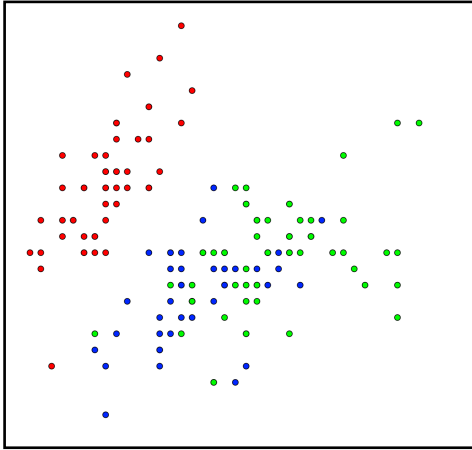


- data points are nodes

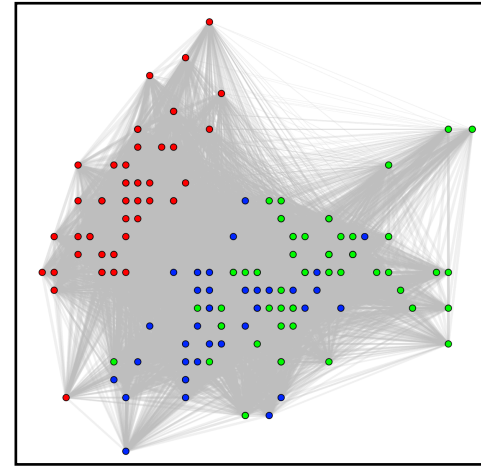


- edges represent similarities

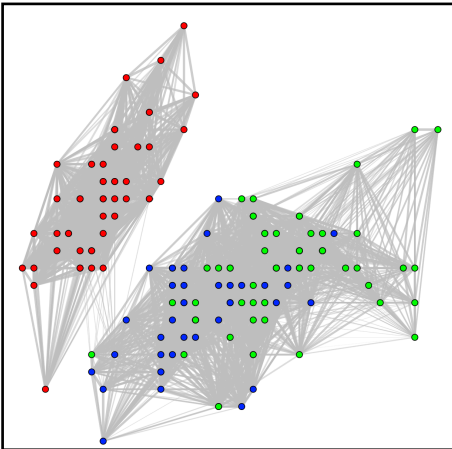
Graph based clustering



- data points are nodes

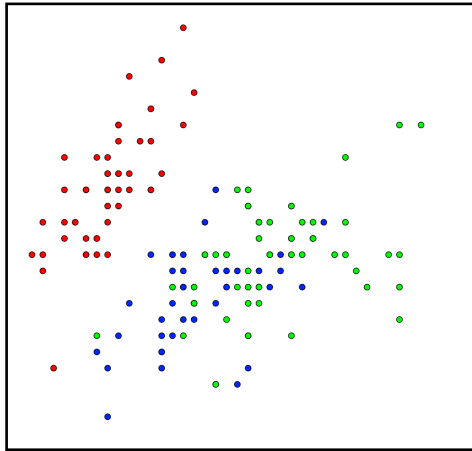


- edges represent similarities

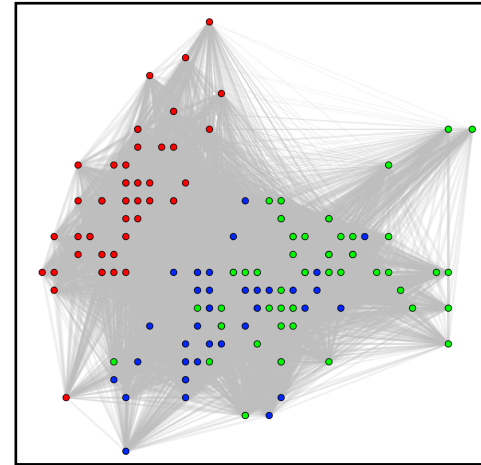


- k-nearest neighbours (KNN) -> sparse graphs

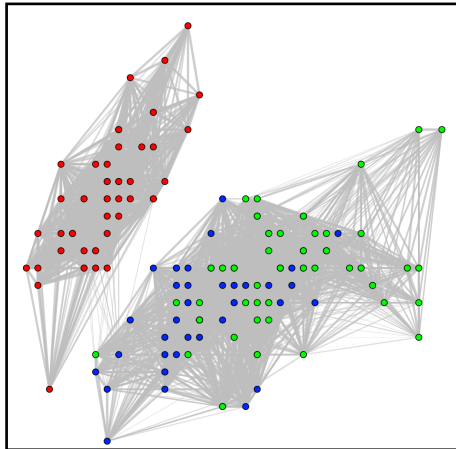
Graph based clustering



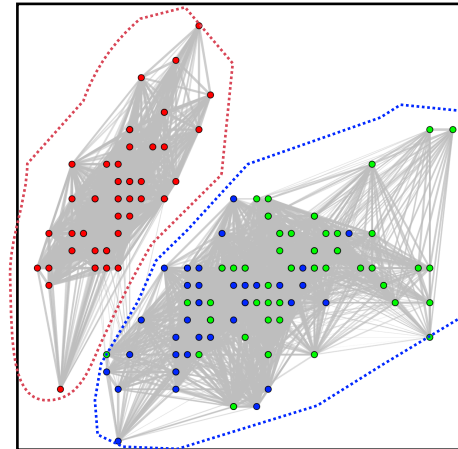
- data points are nodes



- edges represent similarities

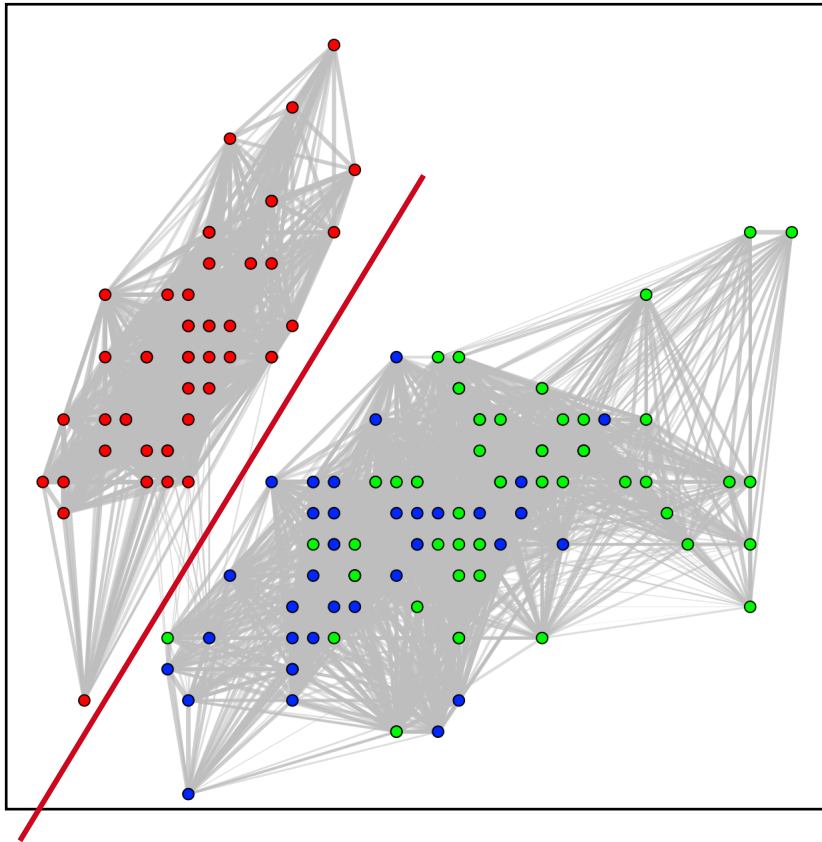


- k-nearest neighbours (KNN) -> sparse graphs



- find well connected sub-graphs

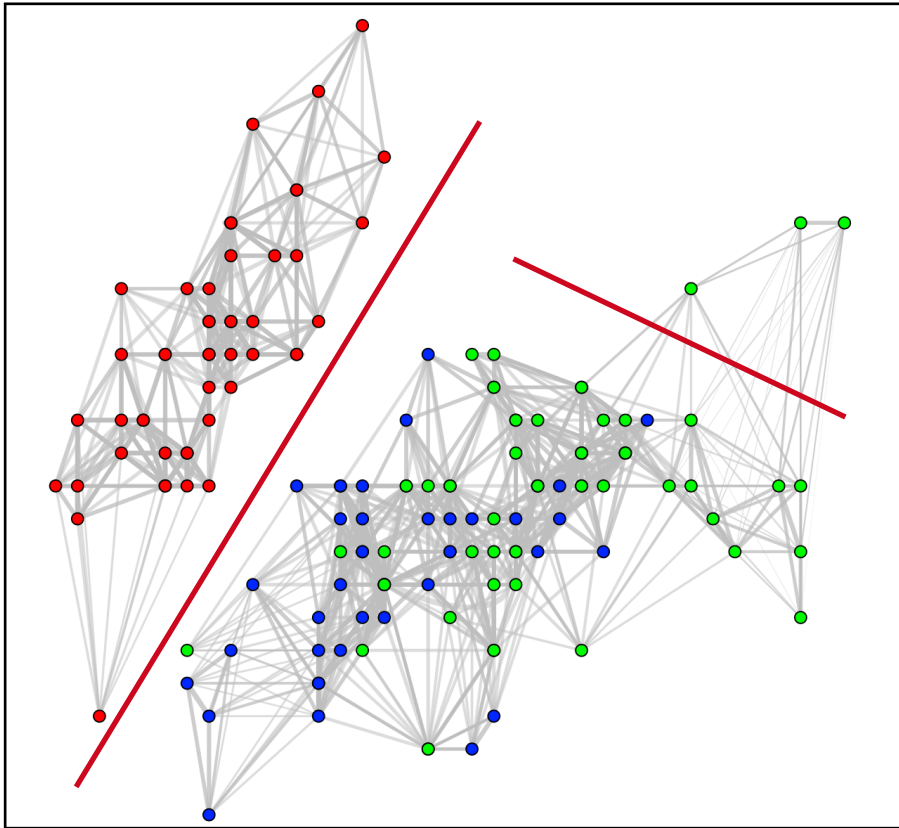
Graph cut



KNN = 50

- Cluster by finding cuts in the graph
- Cut cost $\mathbf{C(A,B)}$ = sum of edge weights in cut

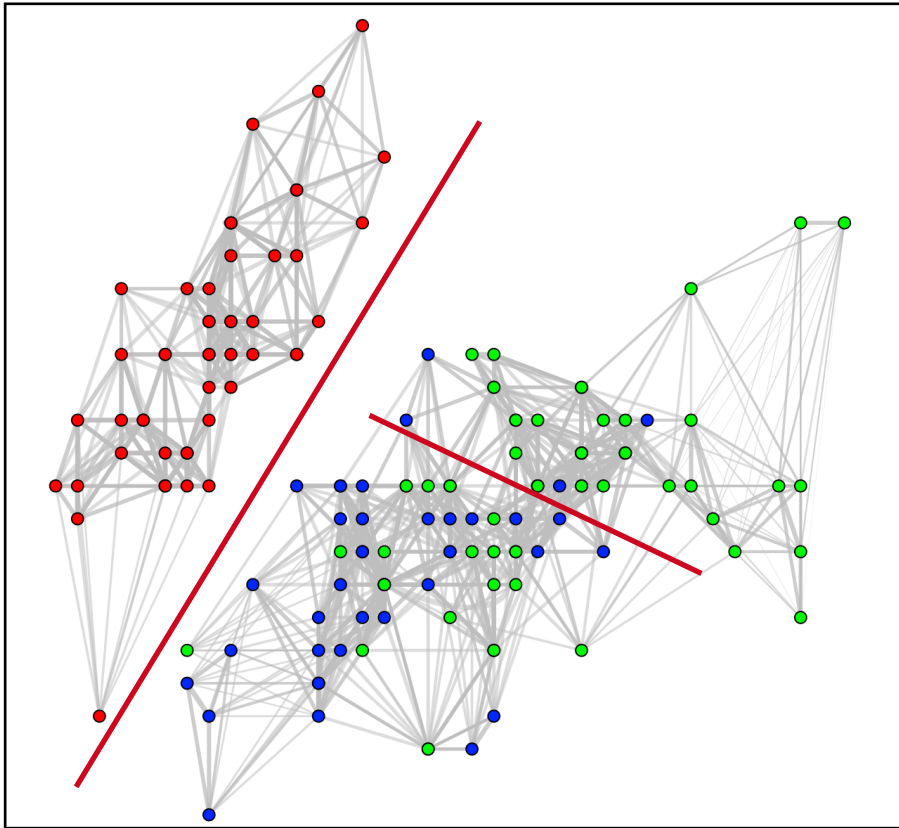
Graph cut



KNN = 10

- Cluster by finding cuts in the graph
- Cut cost $\mathbf{C(A,B)}$ = sum of edge weights in cut
 - smallest cuts might not be the best

Normalized graph cut



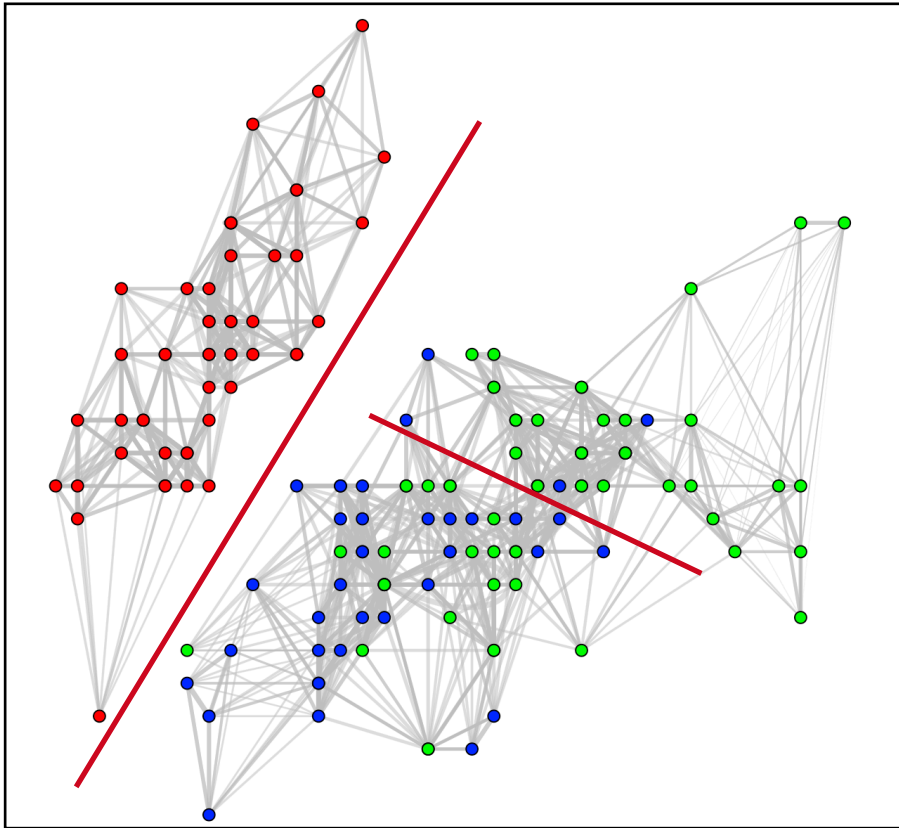
KNN = 10

- Normalized graph cut avoids small graphs

$$normCUT(A, B) = \frac{CUT(A, B)}{VOL(A)} + \frac{CUT(A, B)}{VOL(B)}$$

where $VOL(A)$ is the weight sums of cluster A.

Spectral Clustering

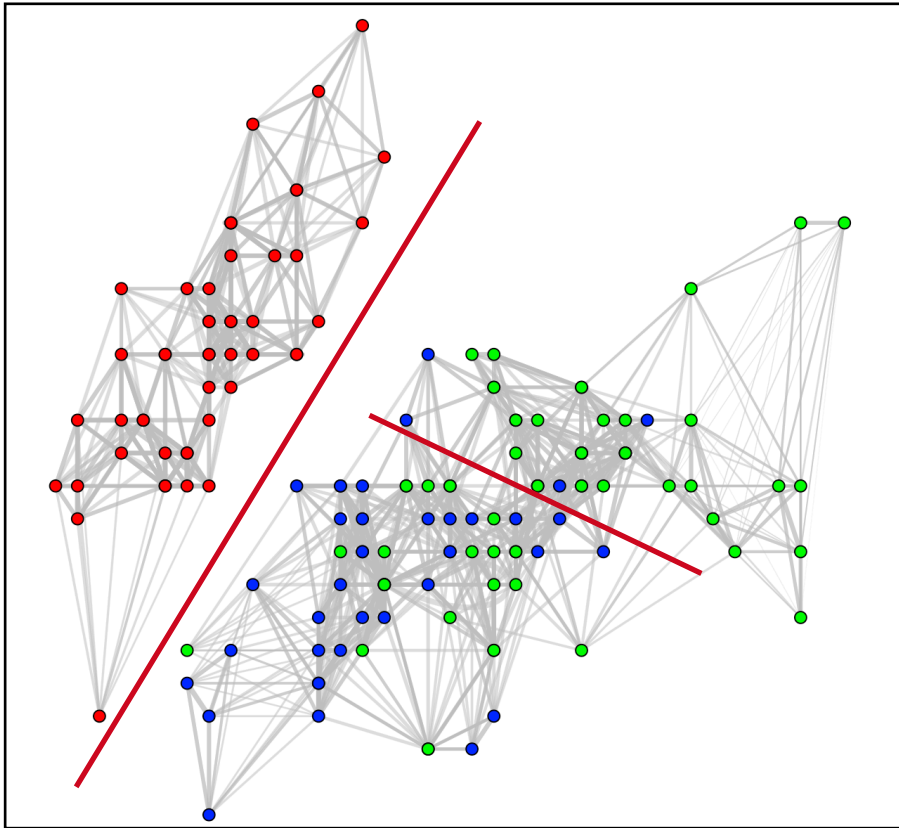


KNN = 10

- Let A be an adjacent matrix of the graph:
 - $a_{ij}=1$ if nodes i and j are connected
- A laplacian matrix is defined as:
$$L = D - A$$
 - where D is a diagonal matrix with the number of neighbours of a node
- If we perform a spectral analysis of L^*
$$L\lambda = u\lambda$$
 - eigenvectors (λ) provides CUTs in the graph
 - eigenvalues (u) provides the cost of the CUT.
- Perform k -means on lowest K eigenvalues

* see for more details: http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

Spectral Clustering



KNN = 10

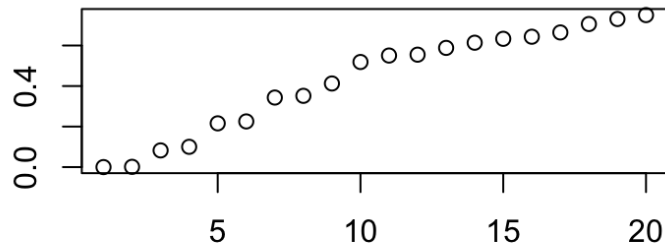
- Let A be an adjacent matrix of the graph:
 - $a_{ij}=1$ if nodes i and j are connected
- A laplacian matrix is defined as:
$$L = D - A$$
 - where D is a diagonal matrix with the number of neighbours of a node
- If we perform a spectral analysis of L^*
$$L\lambda = u\lambda$$
 - eigenvectors (λ) provides CUTs in the graph
 - eigenvalues (u) provides the cost of the CUT.
- Perform k -means on lowest K eigenvalues

* see for more details: http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

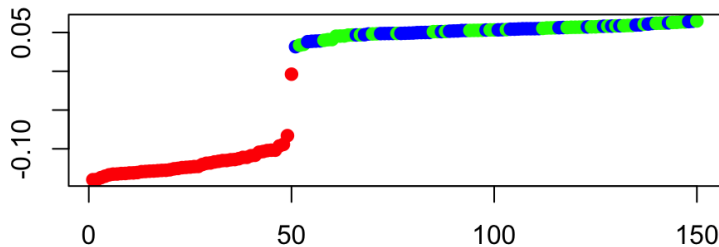
Graph cut and spectral analysis of laplacian matrices

Spectral analysis: $L\lambda = u\lambda$

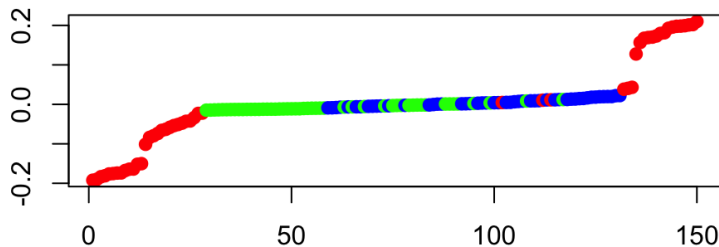
eigenvalues (u)



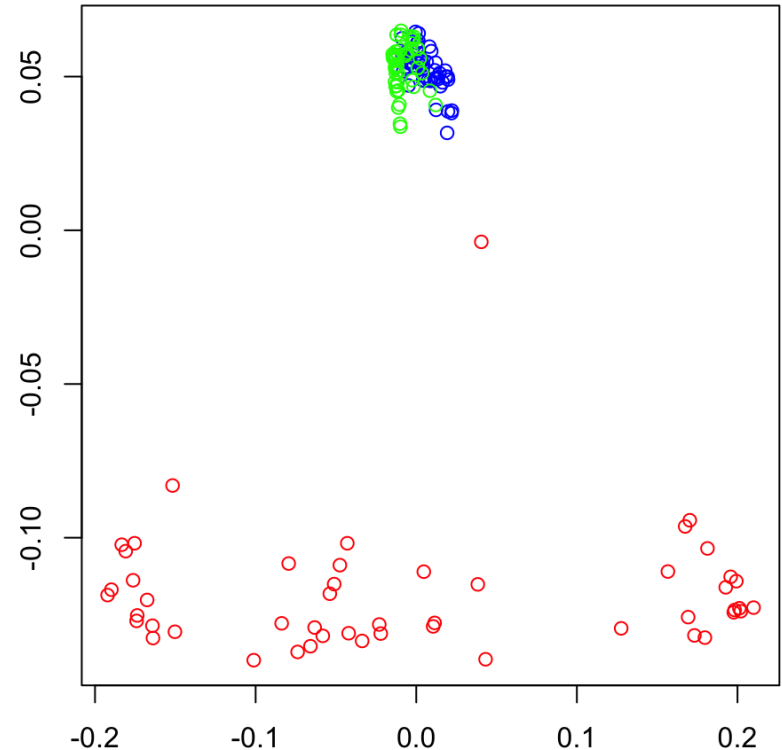
λ_1



λ_2



λ_1

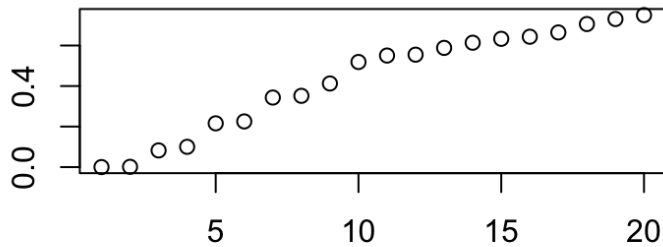


λ_2

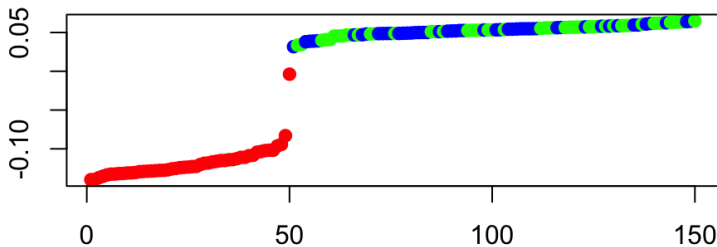
Graph cut and spectral analysis of laplacian matrices

Spectral analysis: $L\lambda = u\lambda$

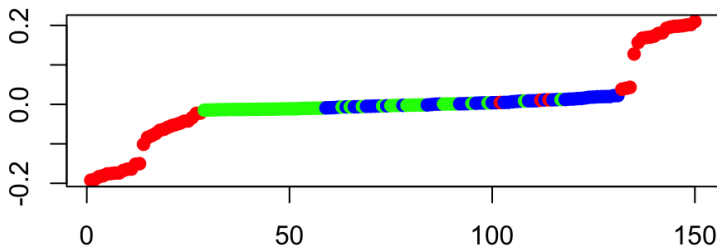
eigenvalues (u)



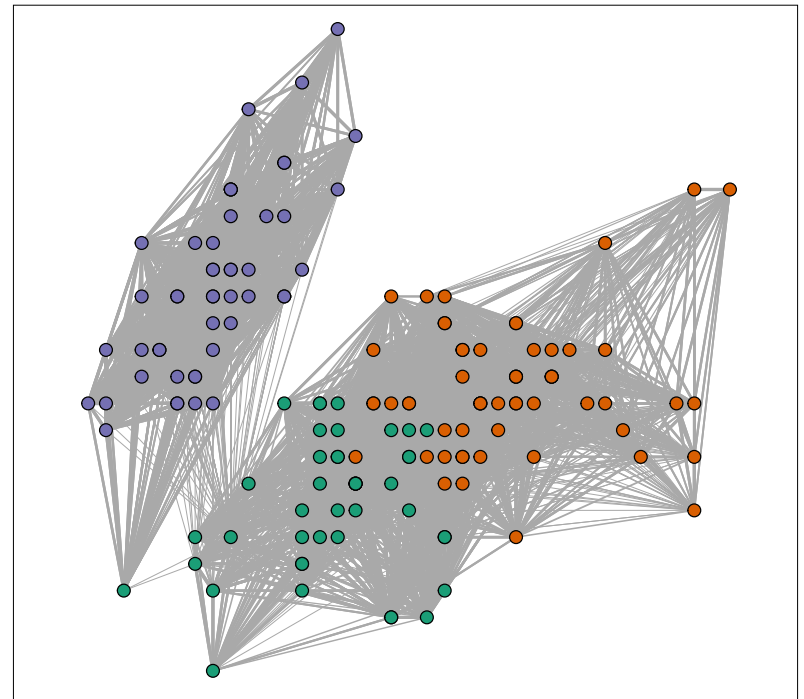
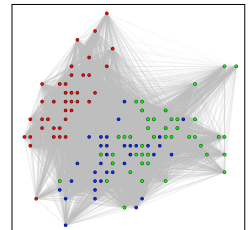
λ_1



λ_2



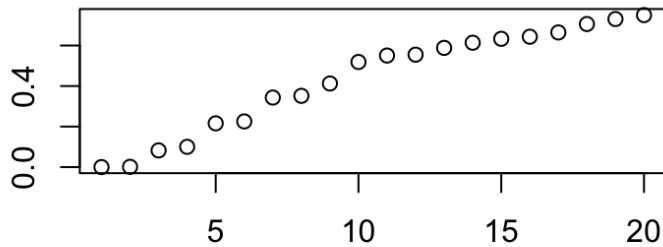
original labels



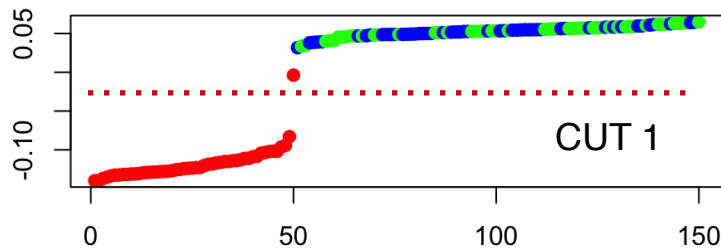
Graph cut and spectral analysis of laplacian matrices

Spectral analysis: $L\lambda = u\lambda$

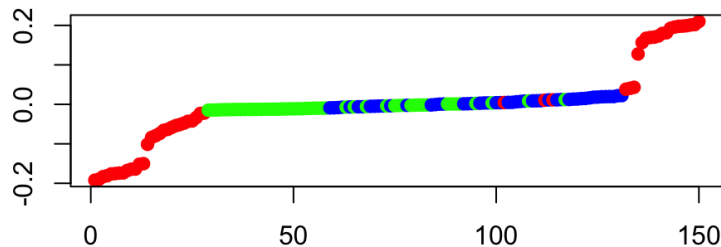
eigenvalues (u)



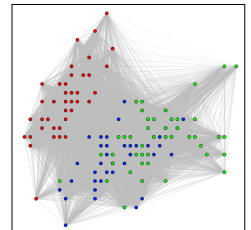
λ_1



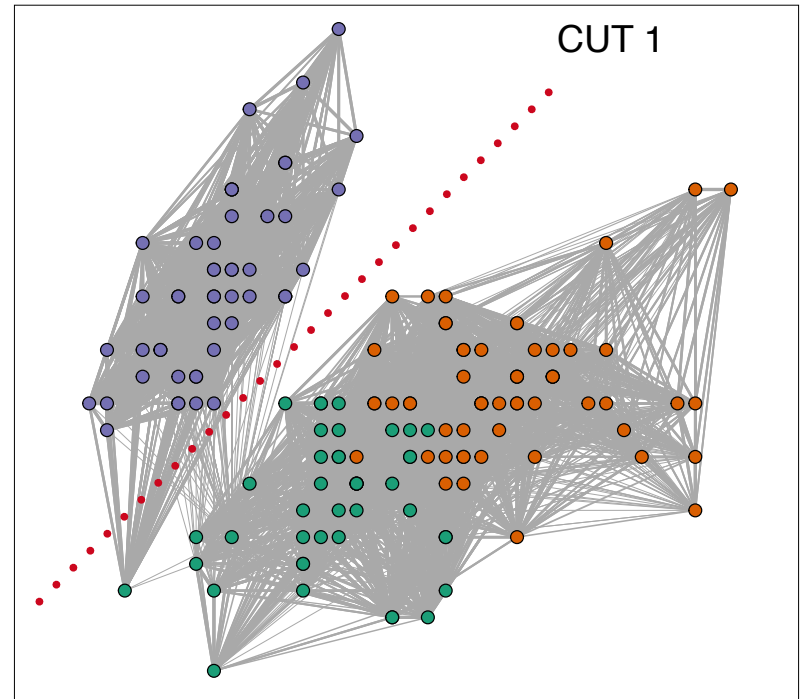
λ_2



original labels



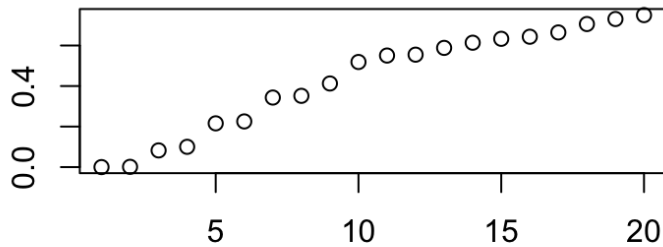
CUT 1



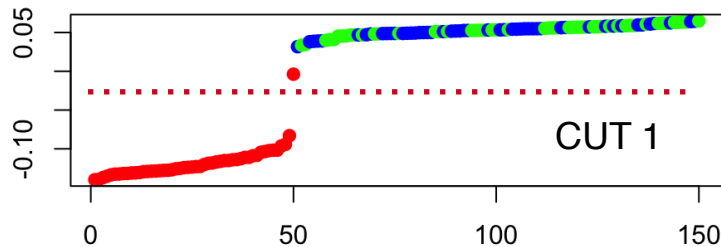
Graph cut and spectral analysis of laplacian matrices

Spectral analysis: $L\lambda = u\lambda$

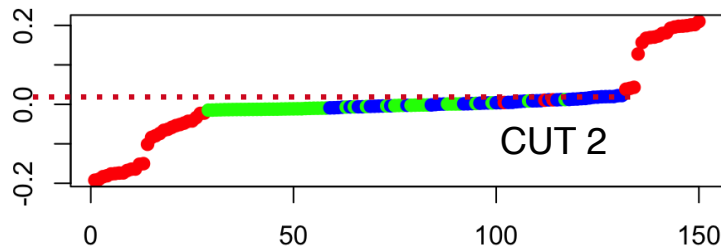
eigenvalues (u)



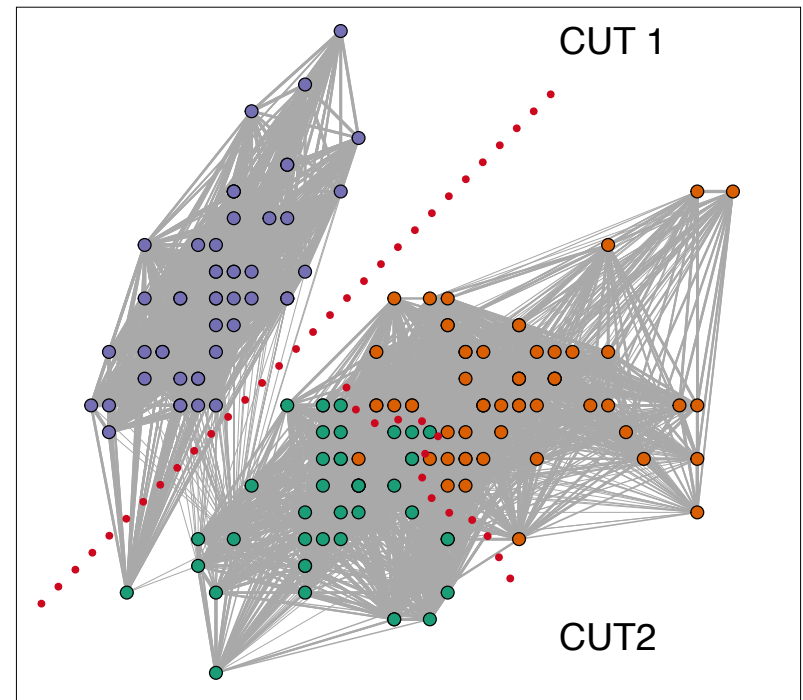
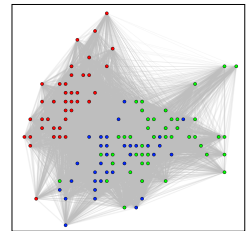
λ_1



λ_2



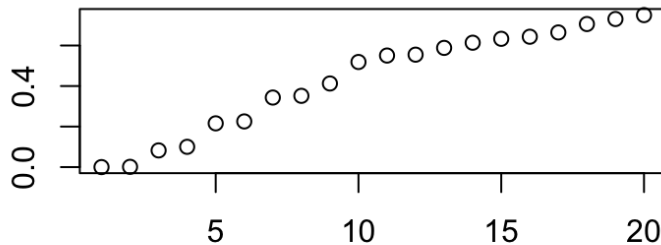
original labels



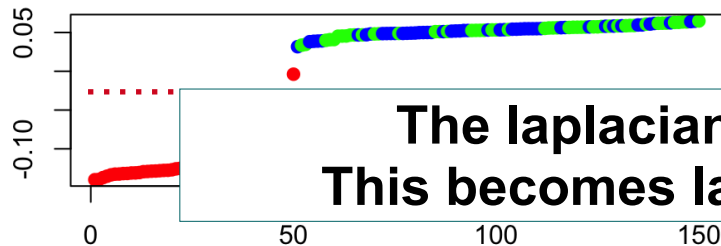
Graph cut and spectral analysis of laplacian matrices

Spectral analysis: $L\lambda = u\lambda$

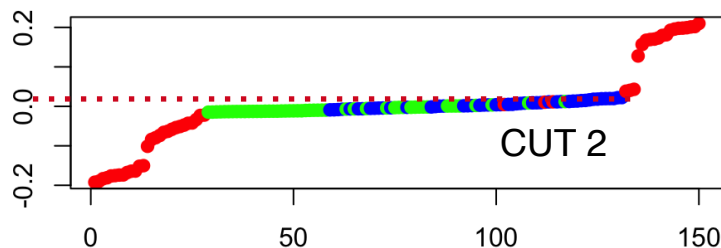
eigenvalues (u)



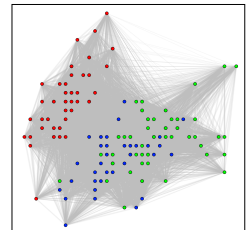
λ_1



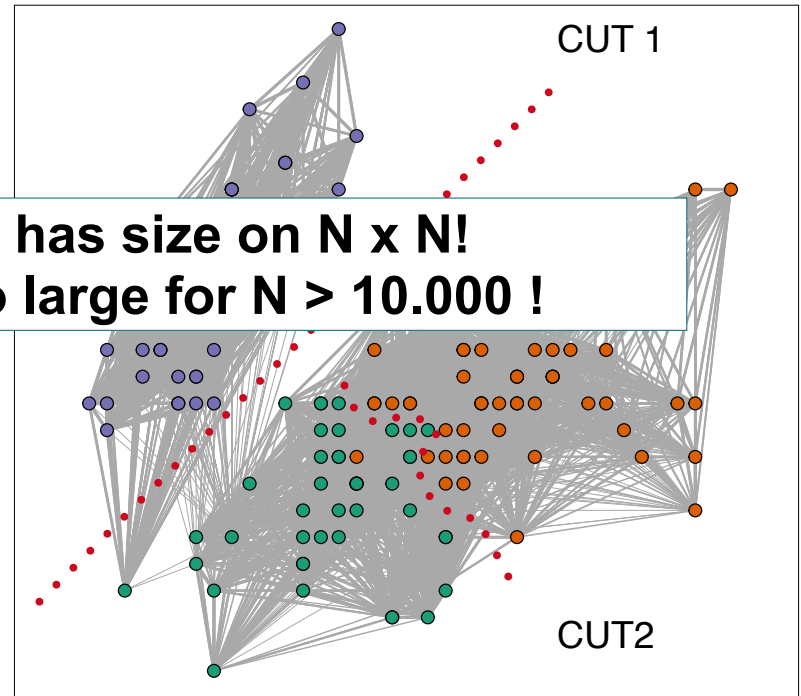
λ_2



original labels



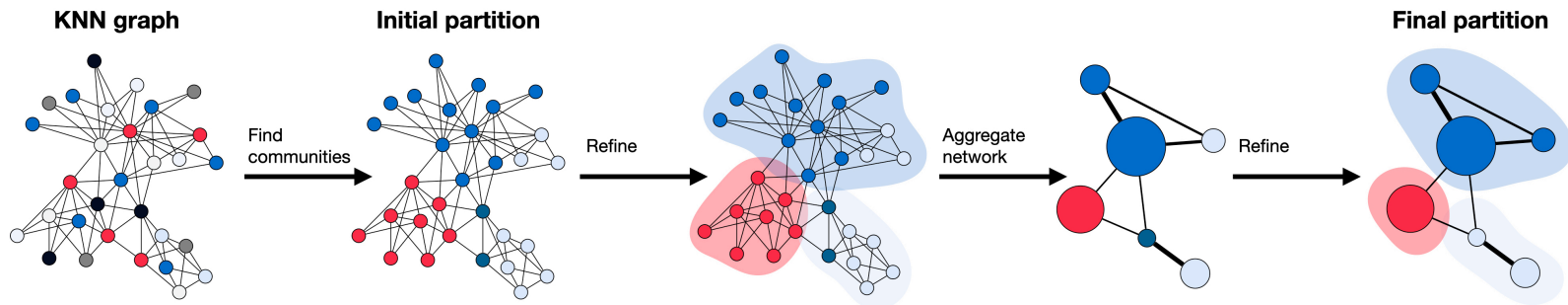
CUT 1



CUT2

The laplacian matrix has size on $N \times N$!
This becomes large too large for $N > 10.000$!

Single cell Clustering / Louvain & Leiden algorithm



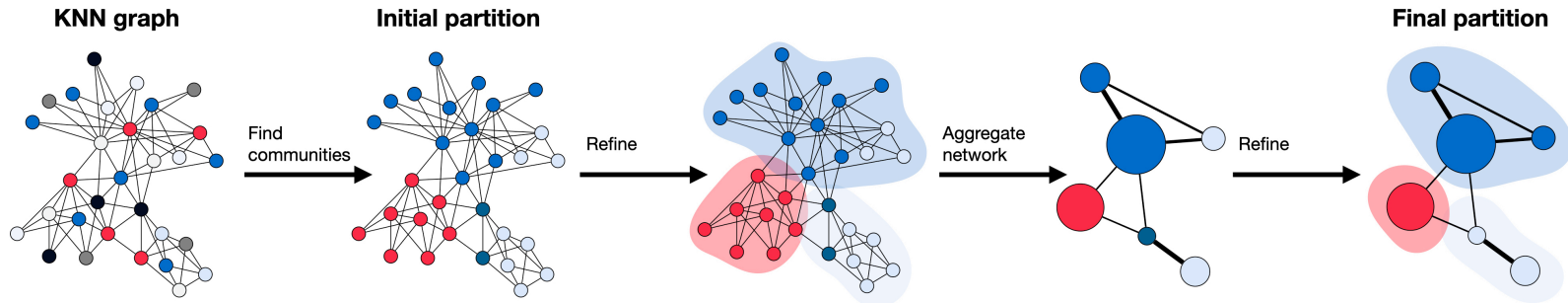
Optimize cluster modularity

$$\mathcal{H} = \sum_c [e_c - \gamma \binom{n_c}{2}],$$

where n_c is the size of cluster and e_c is the number of expected edges

- A) Start with a random partition
- B) Cluster objects improving H
- C) Create a meta-graph level:
 - one meta-node for each cluster
- D) Move objects improving H

Single cell Clustering / Louvain & Leiden algorithm



Source: <https://www.sc-best-practices.org/preamble.html>

Optimize

**Meta-nodes and sparse graphs (knn) allows
Leiden/Louvain to cope with millions of objects !**

$$\mathcal{H} = \sum_c [e_c - \gamma \binom{n_c}{2}],$$

where n_c is the size of cluster and
 e_c is the number of expected edges

A) Start with a random partition

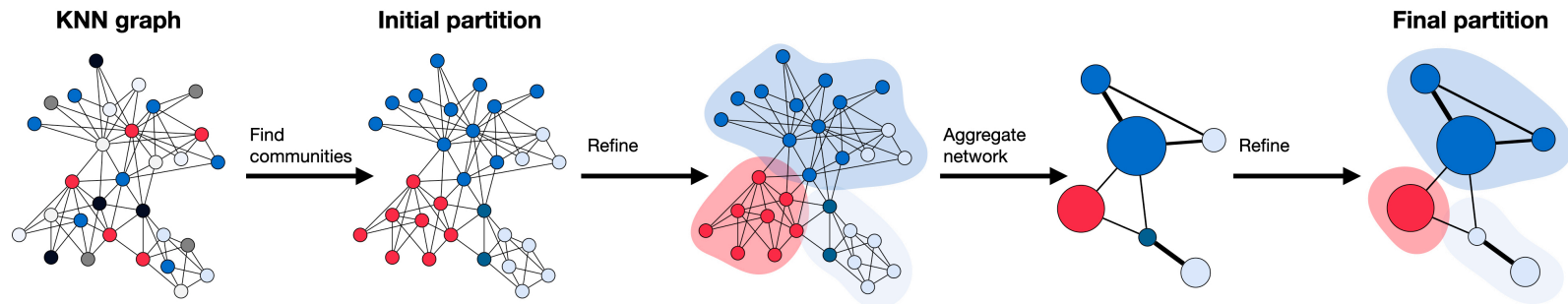
B) Cluster objects improving H

C) Create a meta-graph level:

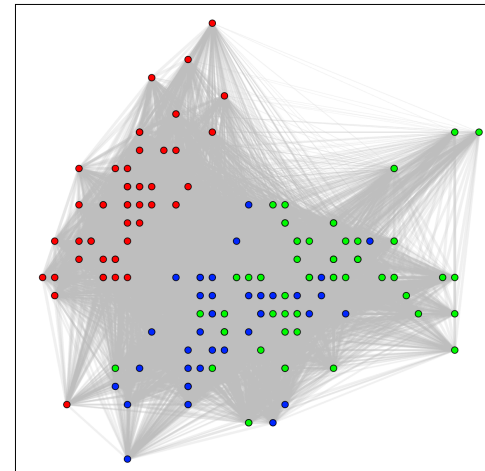
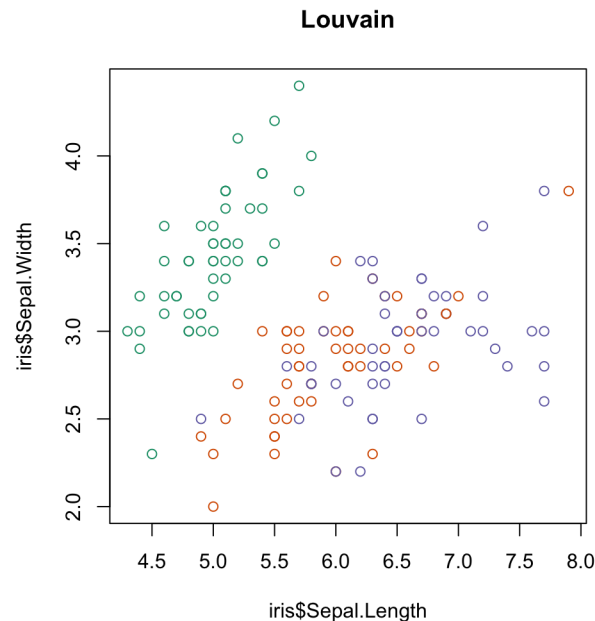
- one meta-node for each cluster

D) Move objects improving H

Single cell Clustering / Louvain & Leiden algorithm



Source: <https://www.sc-best-practices.org/preamble.html>



Resume / Clustering Methods

- K-means, hierarchical clustering, spectral clustering
 - standard algorithms with standard performance on simple clustering problems
- Clustering of single cell algorithms
 - Leiden and louvain clustering
 - Robust and scale well to large data sets on sparse graphs (knn)
- Further issues:
 - Data dimensionality:
 - distances do not work well on high dimension
 - visualisation is easier in low level space
 - Validation:
 - How many clusters is present in the data?
 - Which is the best method?

More details on clustering

- Hastie, Tibshirani and Friedman, The Elements of Statistical Learning, Chapter 14
- Video lecture: <https://www.youtube.com/watch?v=Qa6k7RIwltg>

Resume / Single cell clustering

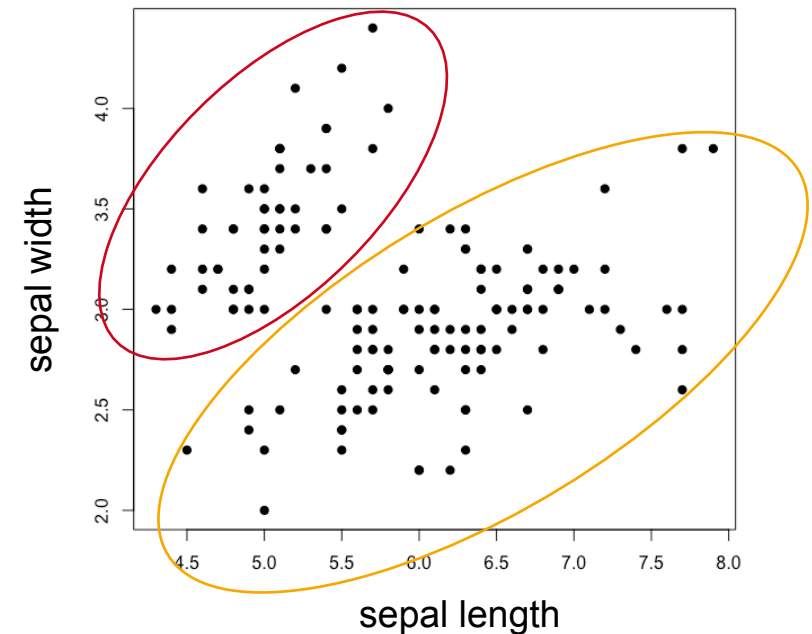
- Finding groups of single cells require complex pipeline:
 - Cell filtering
 - Normalisation
 - Artefact removal
 - **Dimension reduction**
 - **Integration**
 - **Clustering**
 - **Cell annotation / visualisation**

Clustering & Dimension reduction

Clustering

- **Given a data description**
 - i.e. measurement of size of iris flowers
- **Find groups of similar observations**
 - i.e. iris flower sub-types

	<i>Sepal Length</i>	<i>Sepal Width</i>	<i>Petal Length</i>	<i>Petal Width</i>
<i>Flower 1</i>	5.1	3.5	1.4	0.2
<i>Flower 2</i>	4.9	3.0	1.4	0.2
<i>Flower 3</i>	4.7	3.2	1.3	0.2
<i>Flower 4</i>	4.6	3.1	1.5	0.2
...	...	...	...	...

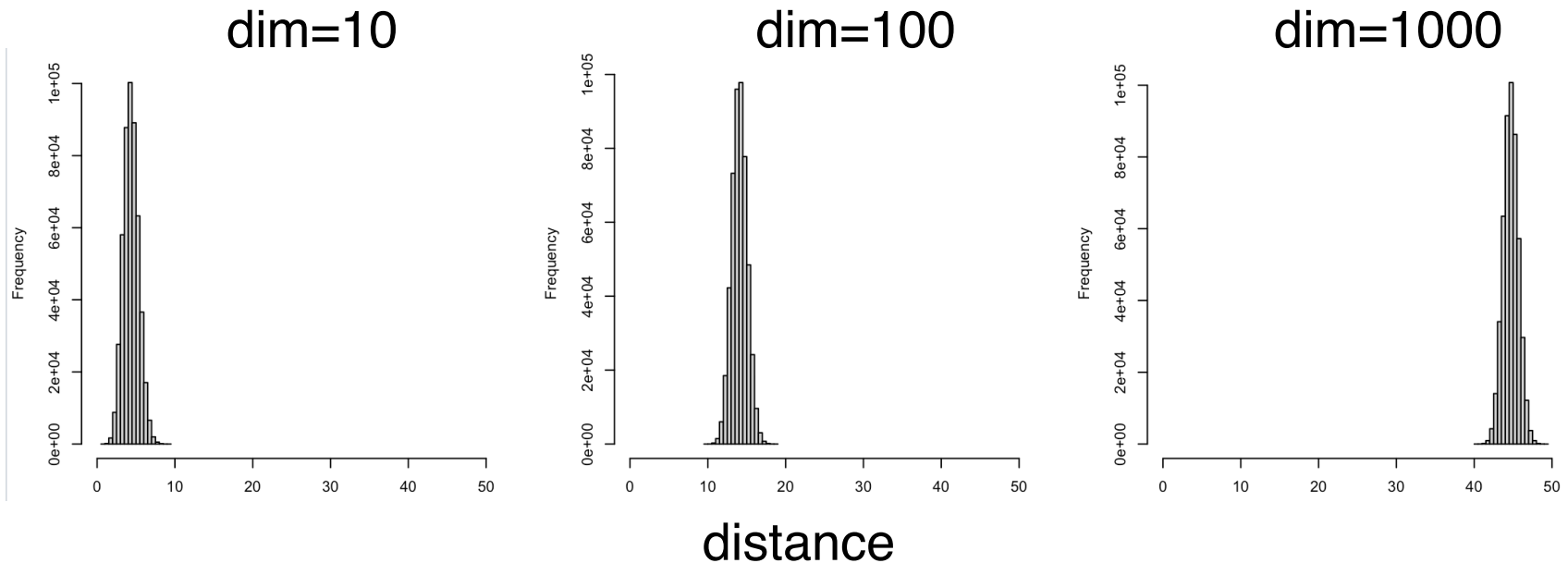


Dimension Reduction

- Distances lose meaning at high dimensional space (curse of dimensionality)

$$\frac{D_{\max} - D_{\min}}{D_{\min}} \rightarrow 0.$$

- Example: distance between points sampled from a normal distribution



Dimension Reduction

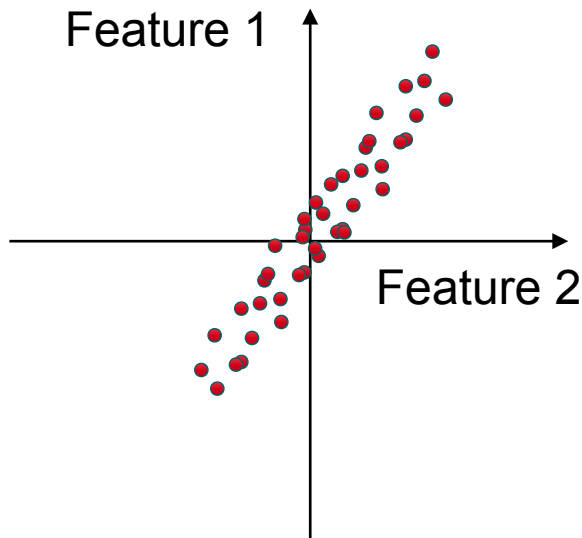
- Distances lose meaning at high dimensional space (curse of dimensionality)
- Unspecific Filtering (without class labels):
 - Keep variables with highest variance (high variable genes)
 - ***Rationale:*** important features change values across groups
- Dimensionality Reduction by Transformation:
 - linear: principal component analysis (PCA)
 - Non-linear / manifold learning: t-SNE & UMAP (for visualisation)

Principal Component Analysis

- For a data X , find linear combination of features (w) capturing most of data variance

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{ \|\mathbf{X}\mathbf{w}\|^2 \}$$

- Can be solved by linear algebra / eigenvector decomposition



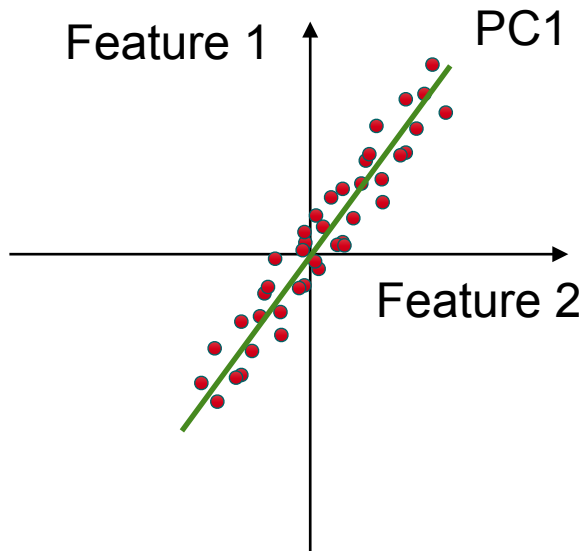
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

- For a data X , find linear combination of features (w) capturing most of data variance

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{ \|\mathbf{X}\mathbf{w}\|^2 \}$$

- Can be solved by linear algebra / eigenvector decomposition



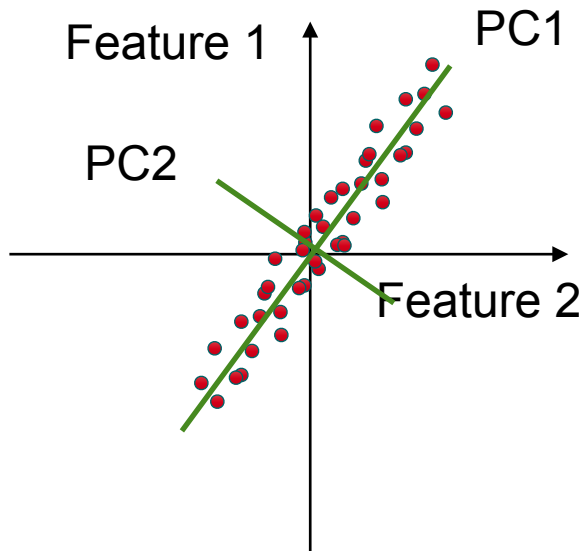
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

- For a data X , find linear combination of features (w) capturing most of data variance

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{ \|\mathbf{X}\mathbf{w}\|^2 \}$$

- Can be solved by linear algebra / eigenvector decomposition



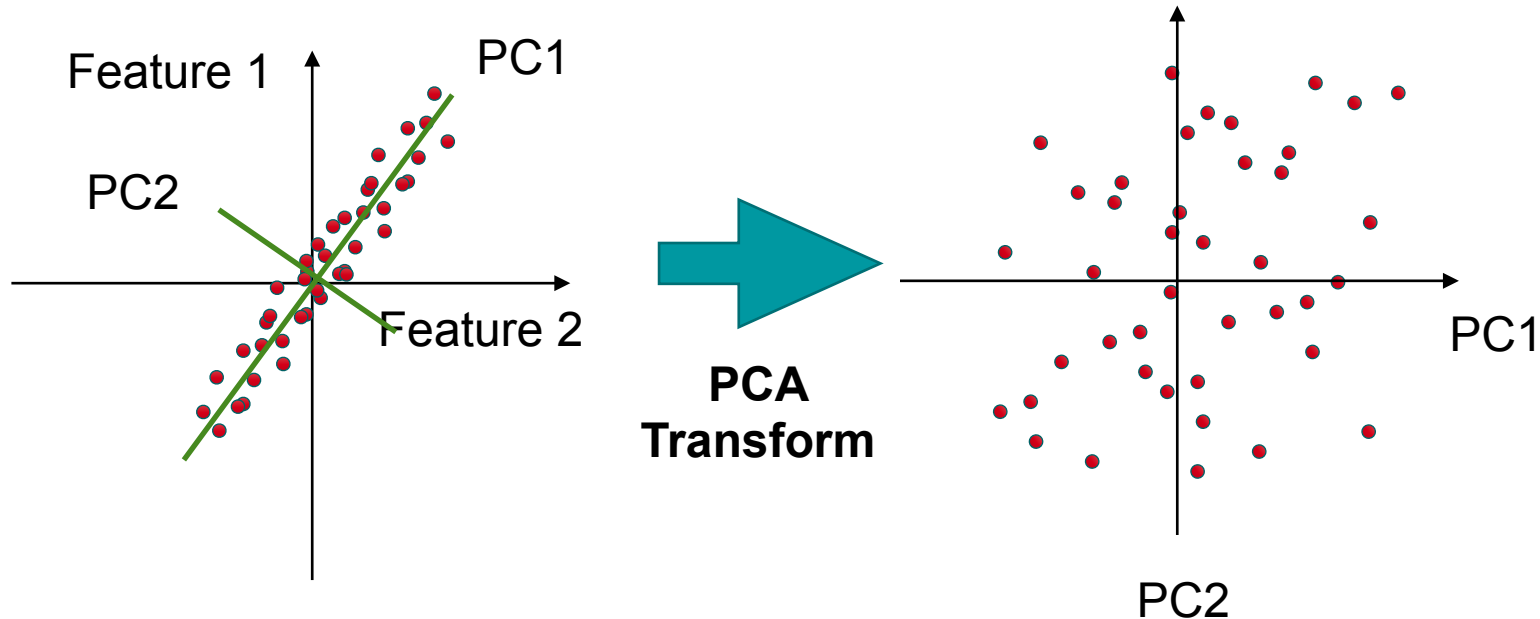
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

- For a data X , find linear combination of features (w) capturing most of data variance

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{ \|\mathbf{X}\mathbf{w}\|^2 \}$$

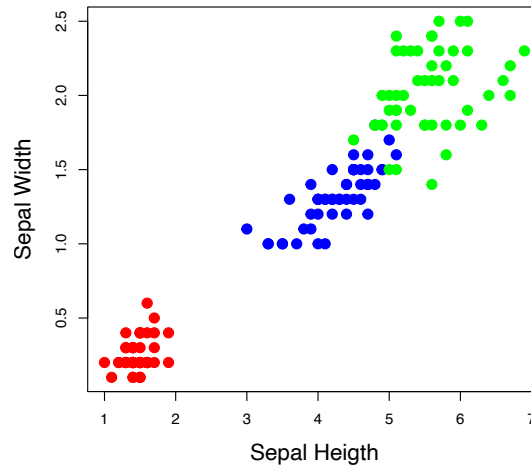
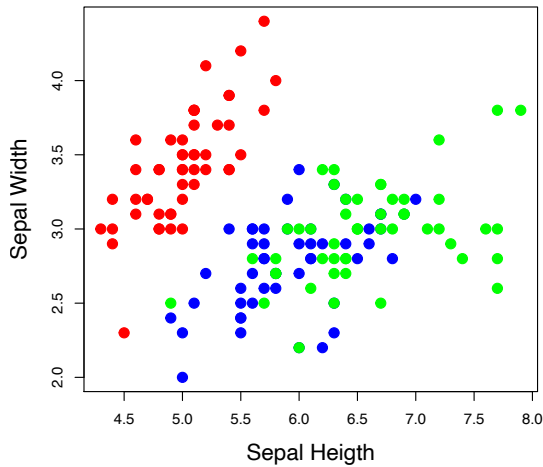
- Can be solved by linear algebra / eigenvector decomposition



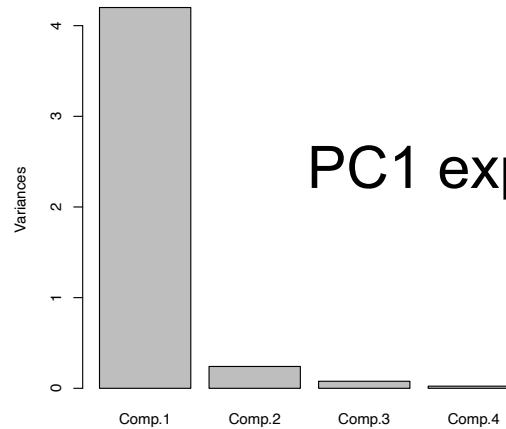
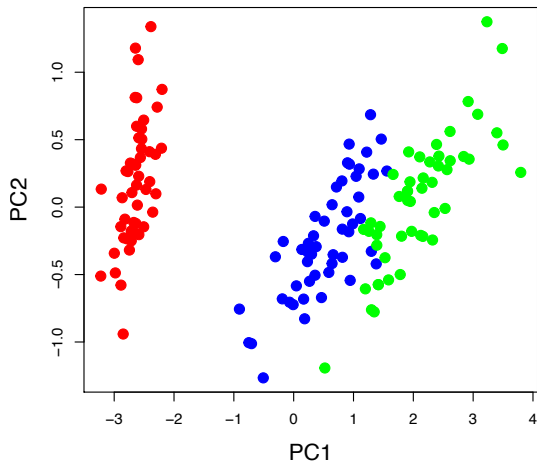
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

PCA - Iris

- Original iris data had 4 variables



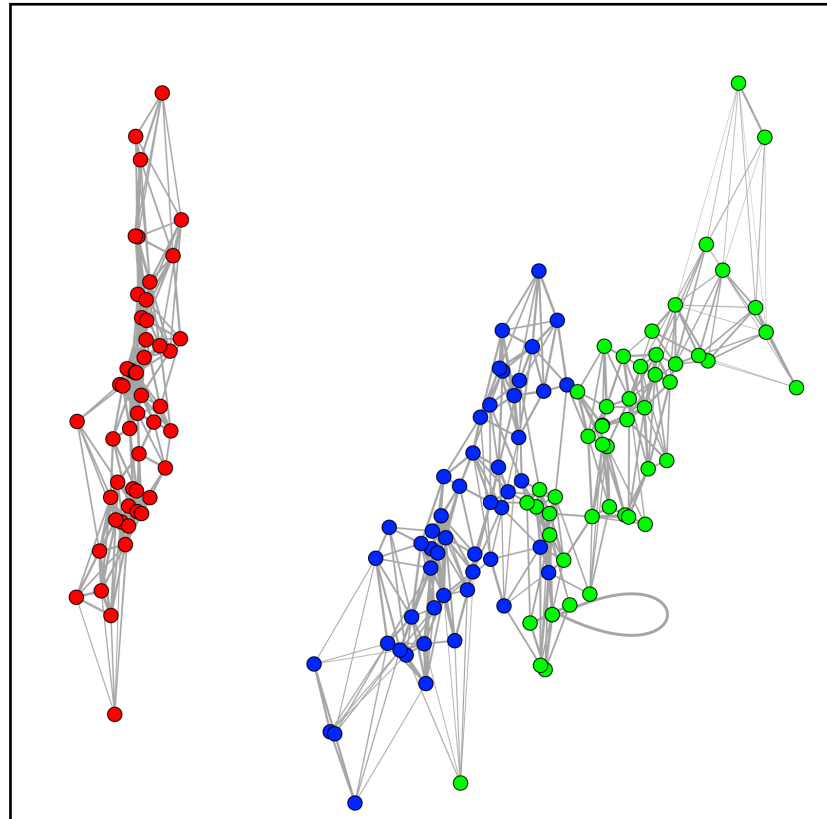
res_pca



Clustering on PCA space

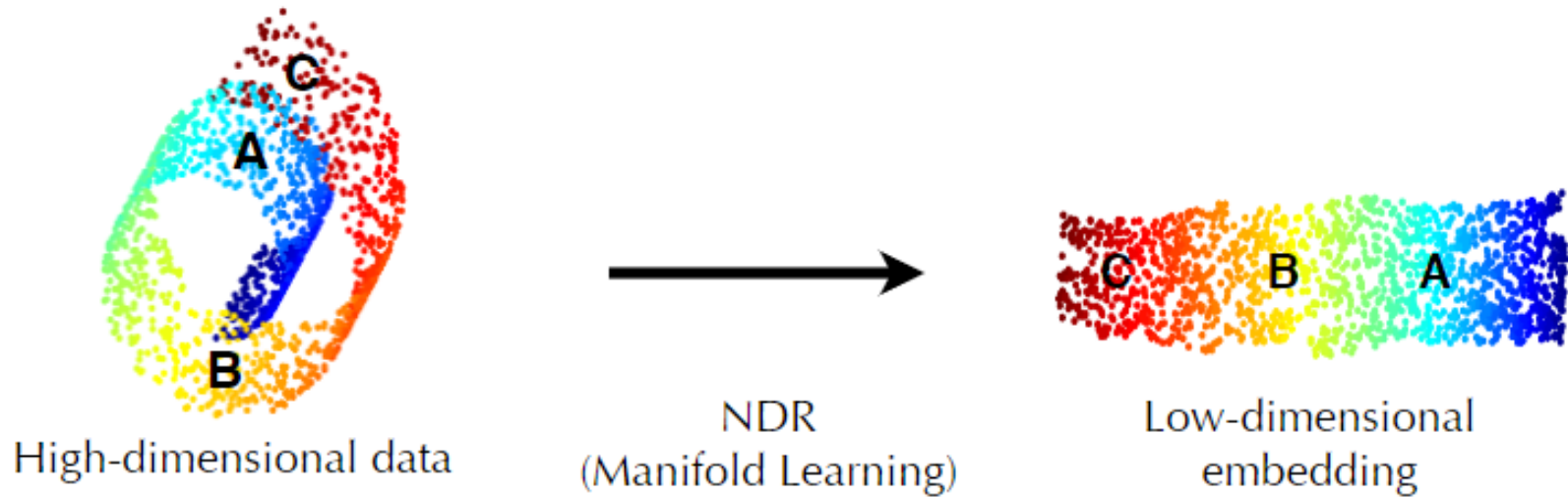
- For single cell data it is usually cluster in PCA space
 - This is crucial for high-dimensional data !

KNN graph of IRIS
in PCA space



Non-linear / Manifold methods

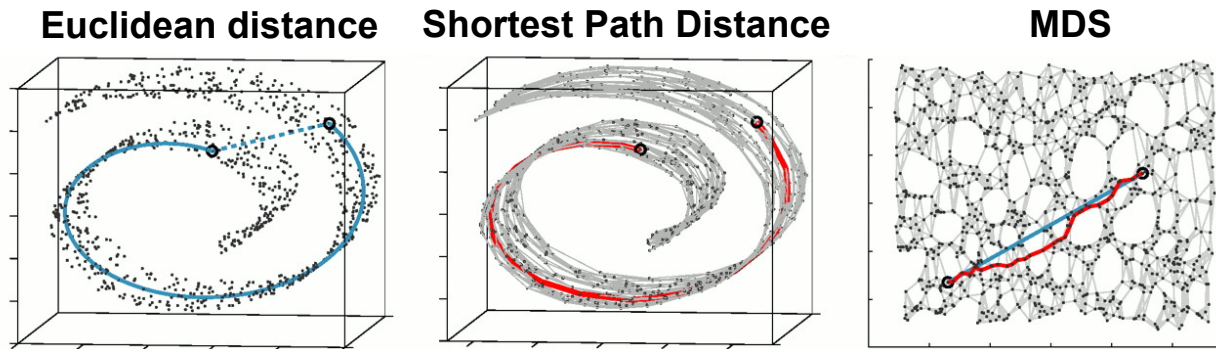
- Data might be distributed at particular regions of a high dimensional space



Adapted from Tenenbaum, et al. 2000

Non-linear / Isomap

- Explore topological distance on nearest neighbour graph



Isomap algorithm:

- (1) create a *knn* graph
- (2) estimate shortest path between nodes (Dijkstra's algorithm)
- (3) use multidimensional scaling (MDS) on shortest paths

MDS algorithm:

find vectors $y_1, \dots, y_n \in Y^N$ such that
$$\sum_{i,j} (|y_i - y_j| - d_{ij})^2$$

where d_{ij} is the similarity between nodes and $N = 2$

Non-linear methods

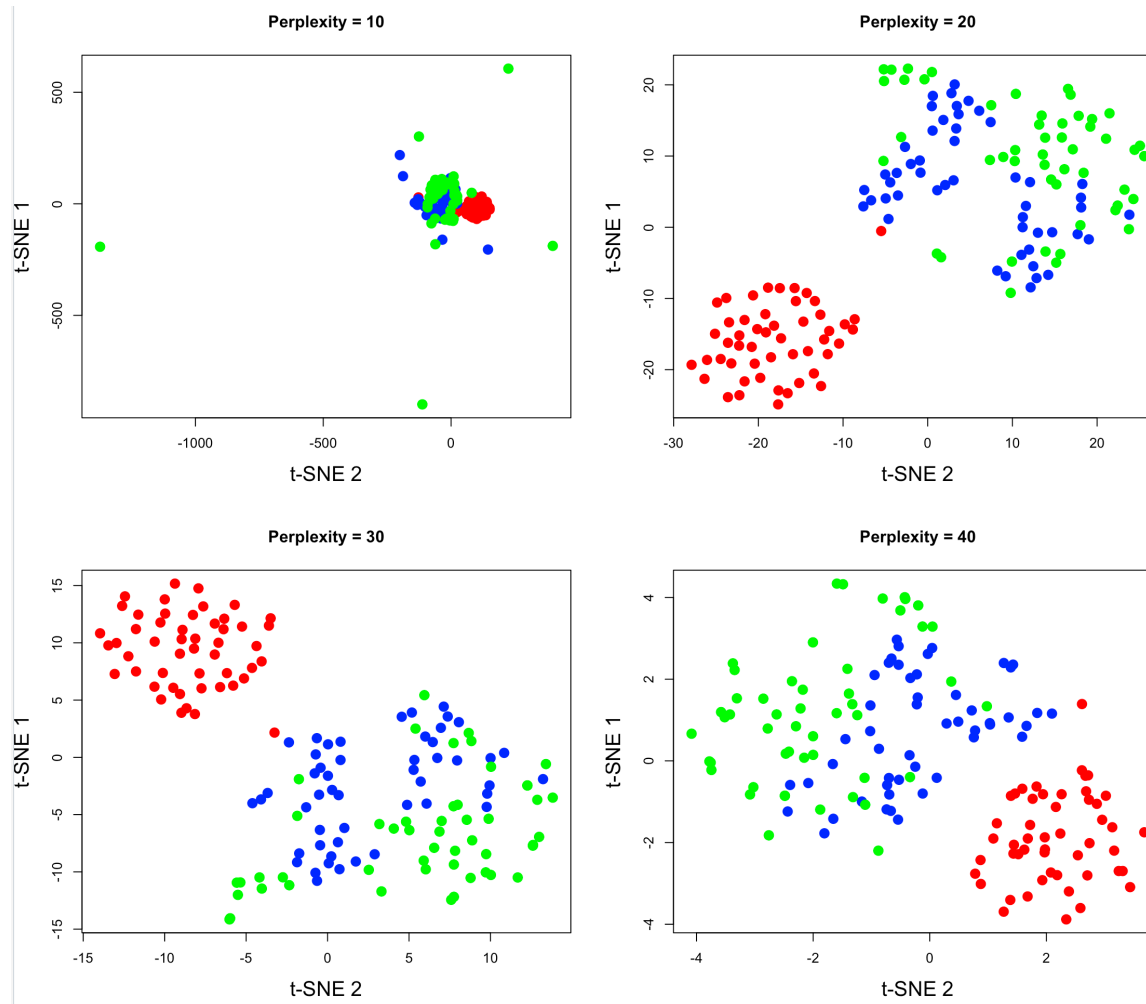
- Current variants of Isomap (t-SNE or UMAP) are often used
- t-SNE - for a given kernel (similarity) D learn a N dimensional map Y

$$KL(D \parallel Q) = \sum d_{ij} \log\left(\frac{d_{ij}}{q_{ij}}\right) \quad \text{where} \quad q_{ij} = \frac{|y_i - y_j|^2}{\sum_k \sum_l |y_k - y_l|^2}$$

KL - Kullback–Leibler divergence

See for more details: <https://www.youtube.com/watch?v=CsUqmug7ZMc>

t-distributed stochastic neighbour



- Sensitive to distinct starts and parametrisation
 - Perplexity $\sim$ neighbourhood (k) size
- **t-SNE focus on preserving close neighbourhood**

See for more details: <https://www.youtube.com/watch?v=9iol3Lk6kyU&t=350s>

Non-linear methods

- Variants of Isomap (t-SNE or UMAP) are currently used
- t-SNE - for a given kernel (similarity) D learn a N dimensional map Y

$$KL(D \parallel Q) = \sum d_{ij} \log\left(\frac{d_{ij}}{q_{ij}}\right) \quad \text{where} \quad q_{ij} = \frac{|y_i - y_j|^2}{\sum_k \sum_l |y_k - y_l|^2}$$

KL - Kullback–Leibler divergence

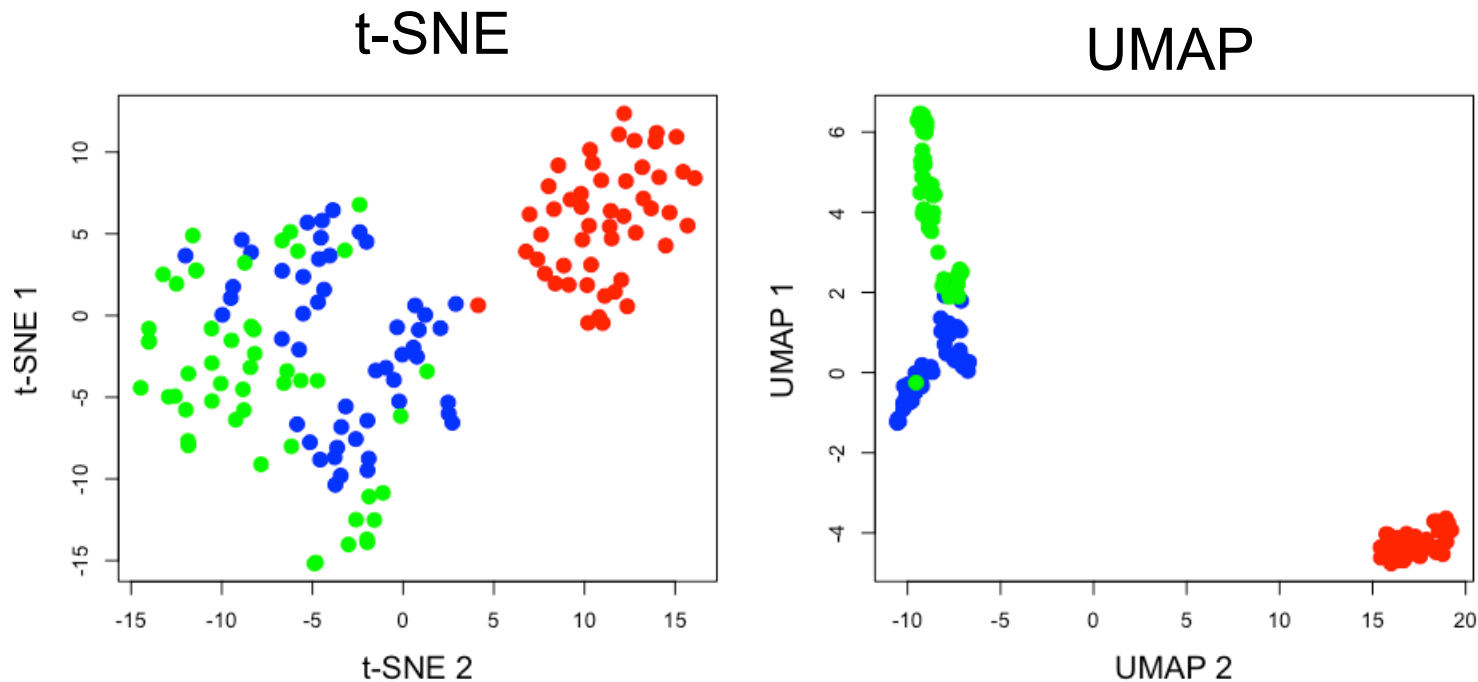
- UMAP - dimension reduction based on Fuzzy Simplicial Sets

$$C((A, \mu), (A, \nu)) = \sum_{a \in A} \mu(a) \log\left(\frac{\mu(a)}{\nu(a)}\right) + (1 - \mu(a)) \log\left(\frac{1 - \mu(a)}{1 - \nu(a)}\right)$$

uses negative samples (non-neighbors)
increasing repulsion between non-neighbors!

See for more details: <https://www.youtube.com/watch?v=CsUqmug7ZMc>

Manifold learning and IRIS



- Low dimensional visualisation of the data
- Caution: These methods fail capturing global structures (distance between clusters!)

See for more details: <https://www.youtube.com/watch?v=9iol3Lk6kyU&t=350s>

Resume / Dimension Reduction

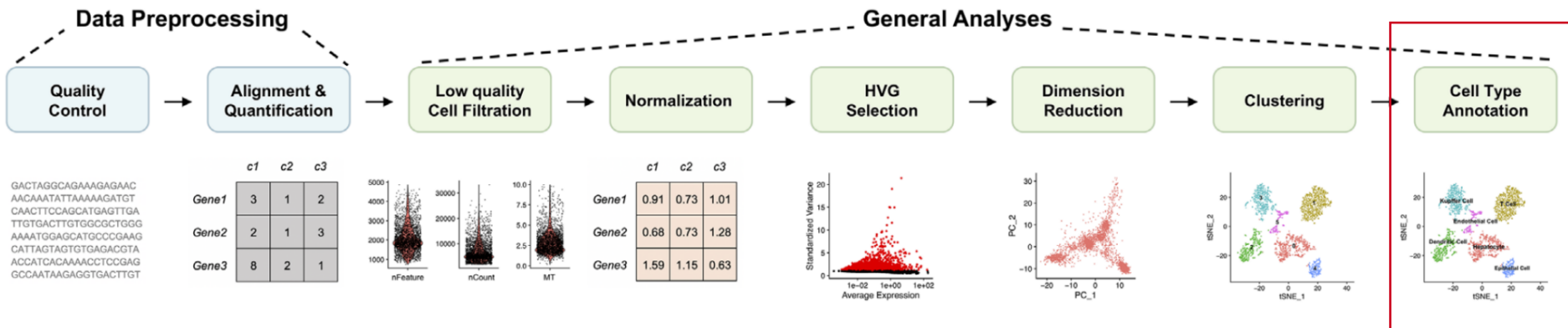
- PCA analysis is a wide spread technique to reduce dimension!
 - Can only capture linear relationships
- Manifold methods
 - Nice low dimensional representation of data
 - Require parametrisation and lose global distance information

Complete course on manifolds/dimension reduction:

<https://www.youtube.com/watch?v=evGm6IJKrDI>

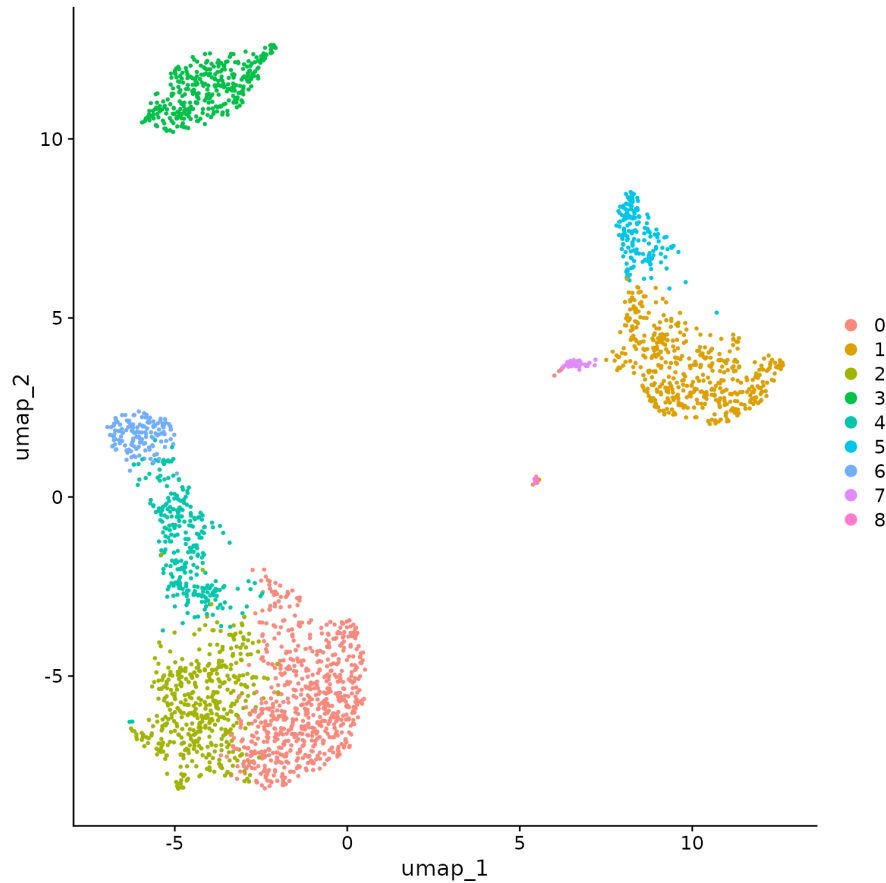
<https://www.youtube.com/watch?v=CsUqmug7ZMc>

Basics Bioinformatics - single cell RNA-seq



Basics Bioinformatics - Clustering

Blood cells (PMBC)

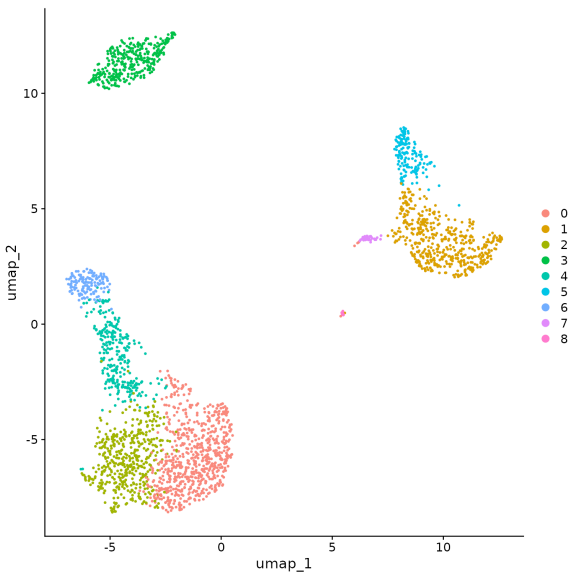


Clustering - identify cells with similar expression patterns

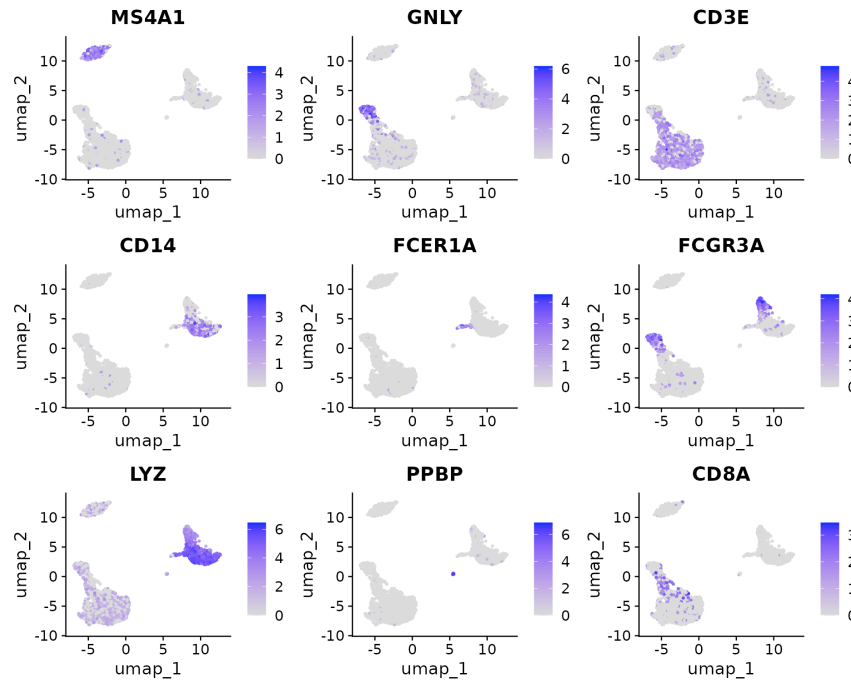
How to identify cell types?

Basics Bioinformatics - Clustering

Blood cells (PMBC)



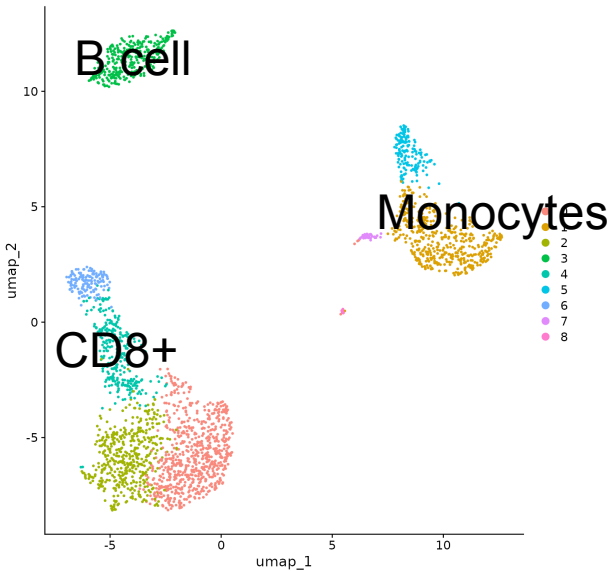
Check expression of known genes:



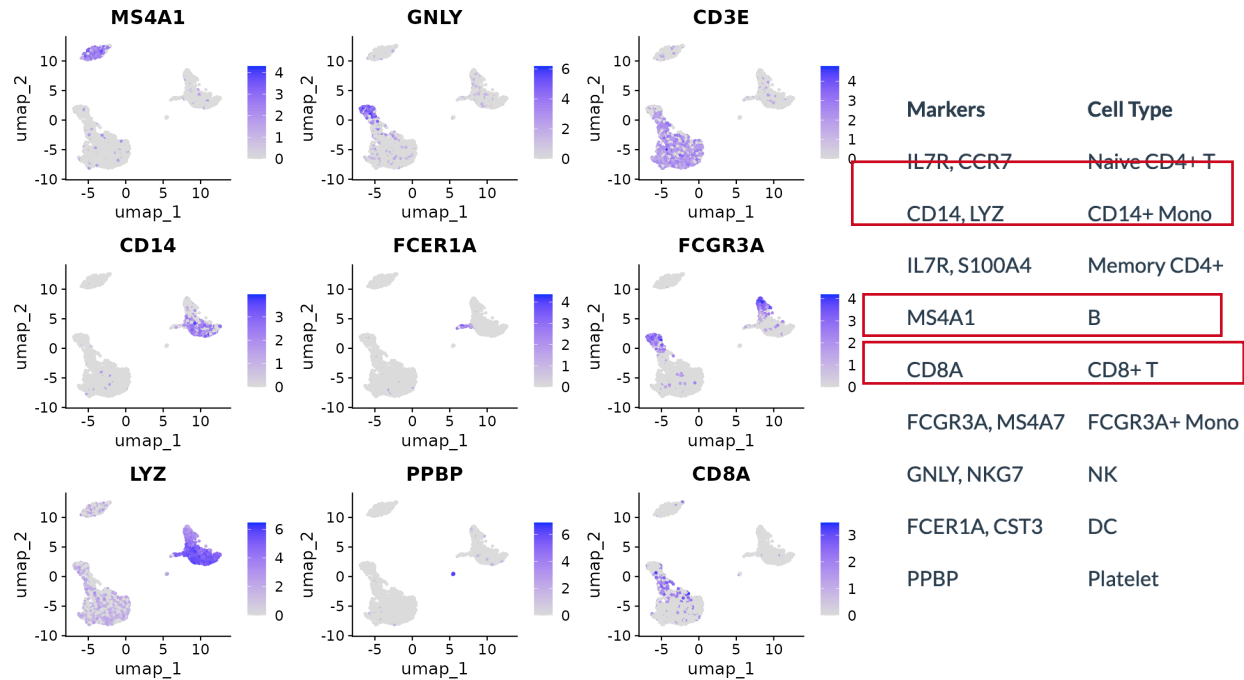
Markers	Cell Type
IL7R, CCR7	Naive CD4+ T
CD14, LYZ	CD14+ Mono
IL7R, S100A4	Memory CD4+
MS4A1	B
CD8A	CD8+ T
FCGR3A, MS4A7	FCGR3A+ Mono
GNLV, NKG7	NK
FCER1A, CST3	DC
PPBP	Platelet

Basics Bioinformatics - Clustering

Blood cells (PMBC)



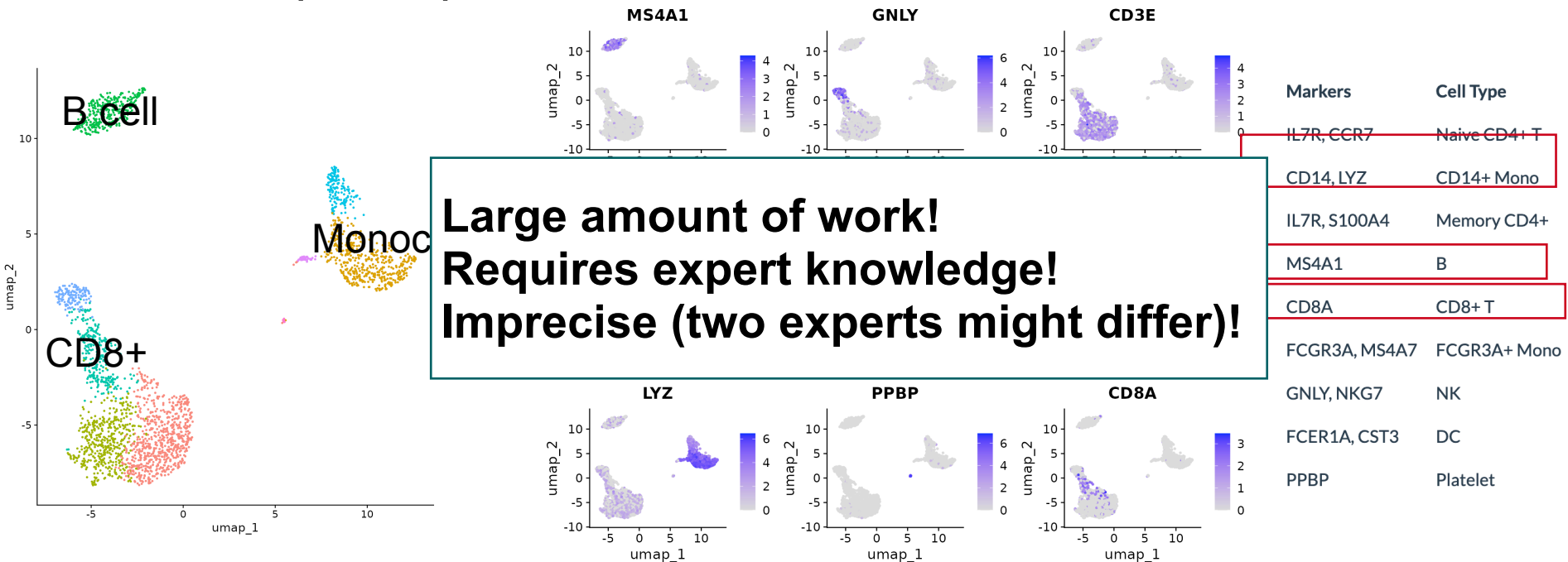
Check expression of known genes:



Basics Bioinformatics - Clustering

Blood cells (PMBC)

Check expression of known genes:



Calendar

8.4 – Introduction to Bioinformatics and Single Cell Sequencing Analysis

15.4 – Single Cell Sequencing Analysis (cont.) & Practice

22.4 – Introduction to HPC clusters and GPU / Project Proposal

29/4 - 1.7 – Project development

8.07 – Project Presentation

Communication/discord channel:

Thank you!