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

Institute for Computational Genomics | RWTH AACHEN UNIVERSITY

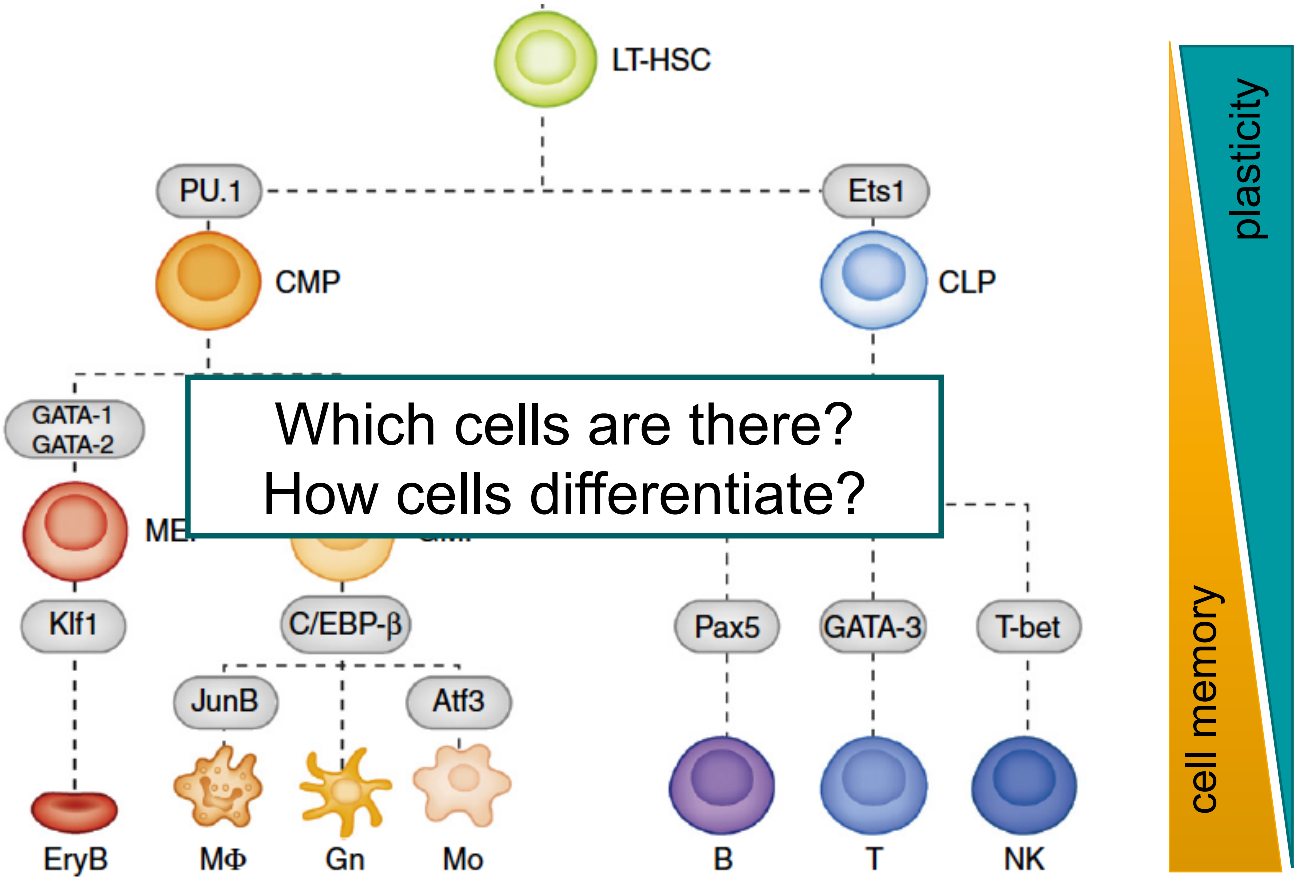
Objectives

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

Expression at Single Cell Level

Cell Differentiation

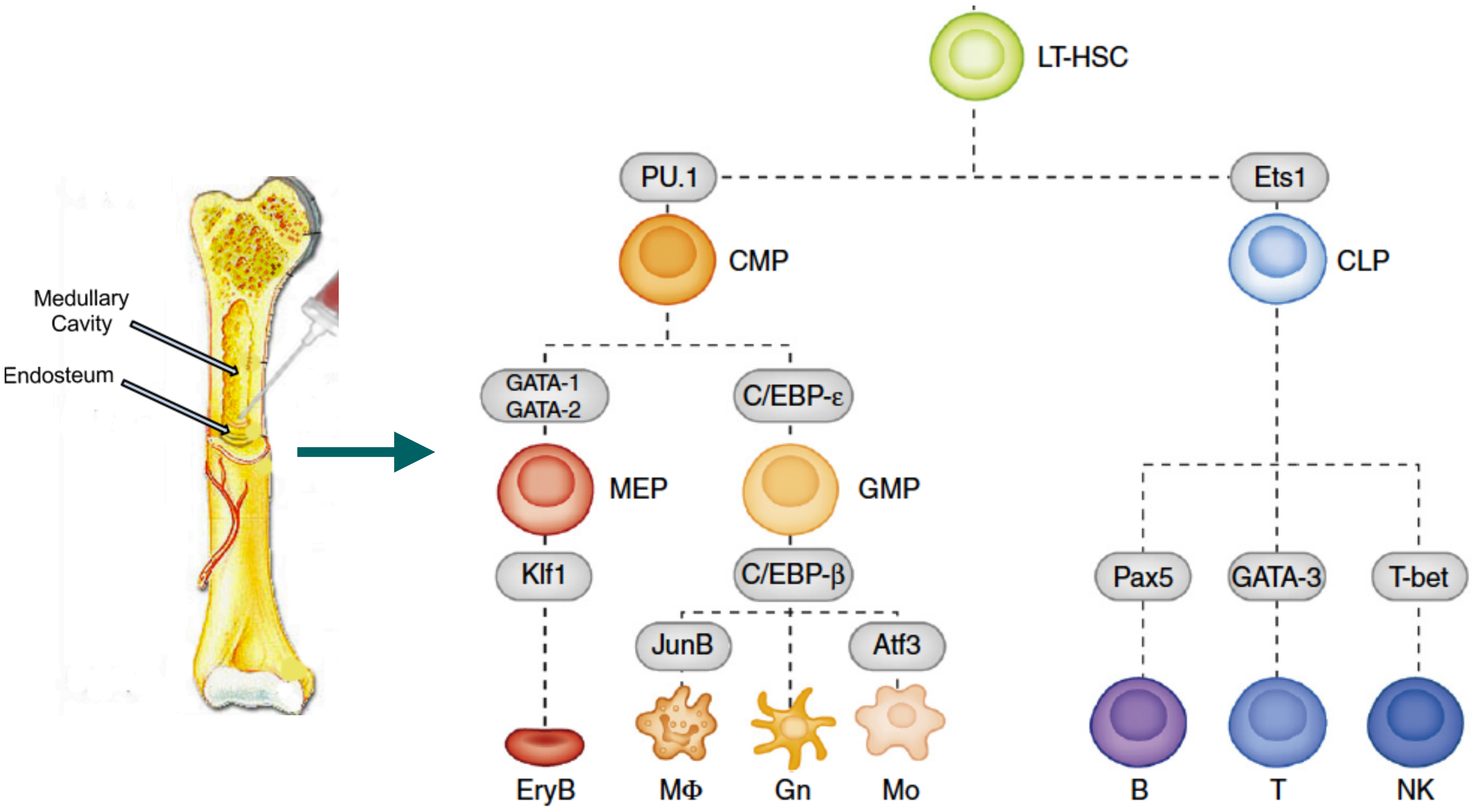
Hematopoiesis



Which cells are there?
How cells differentiate?

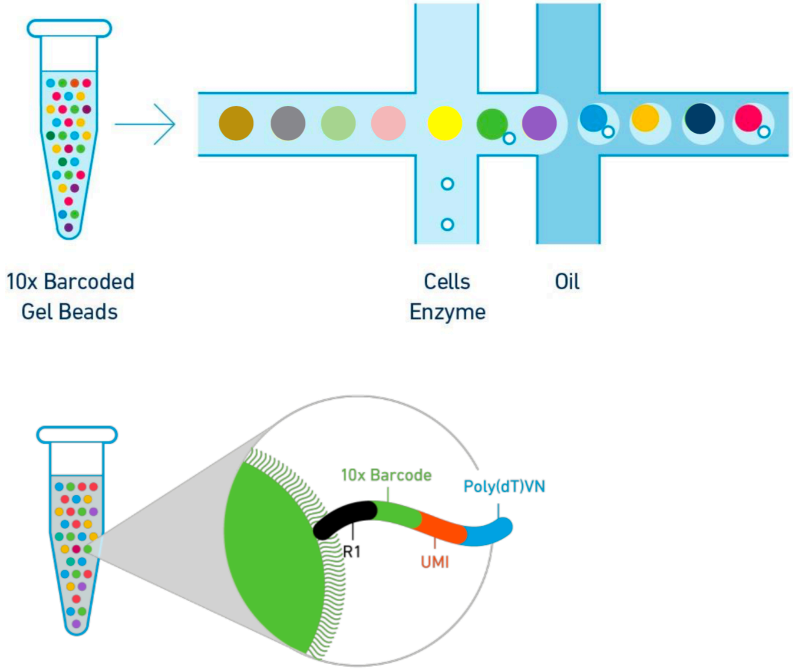
Source: Amit (2016), *Nature Immunoloy.*

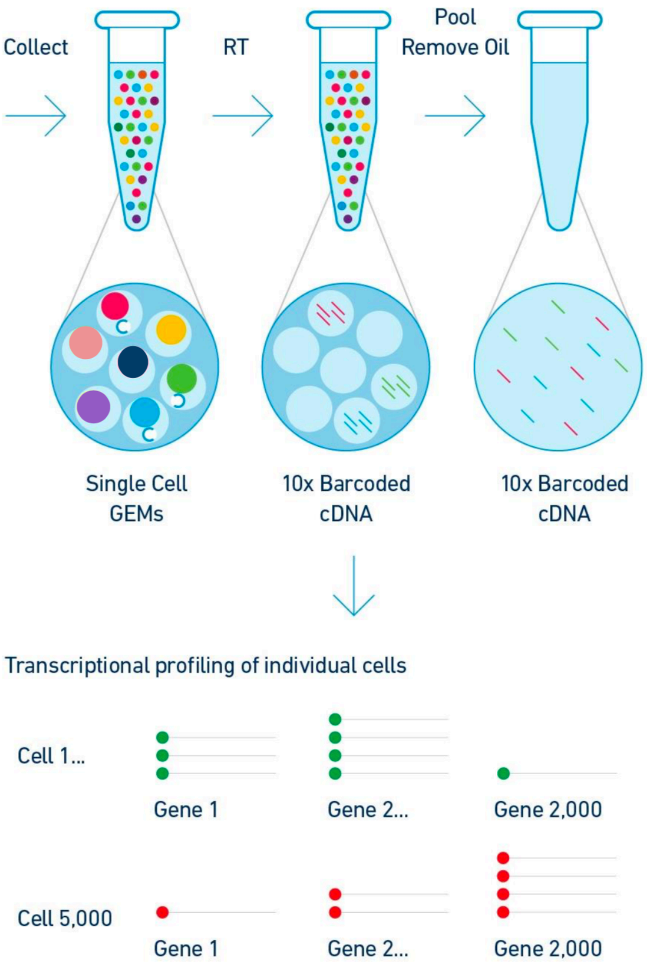
Cell Differentiation



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

Droplet based RNA single cell sequencing

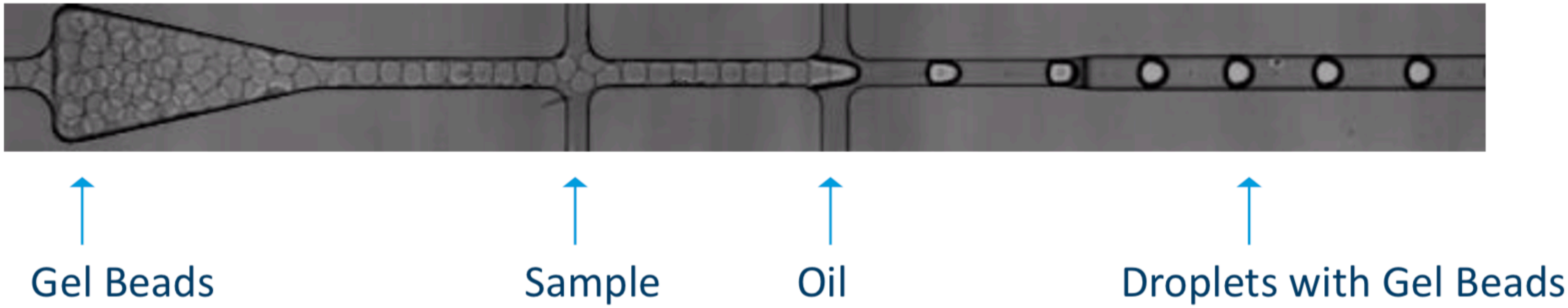




- Input: Single cells in suspension + 10x Gel Beads and Reagents
- Output: Digital gene expression profiles from every partitioned cell

Source: 10x genomics

Droplet based RNA single cell sequencing

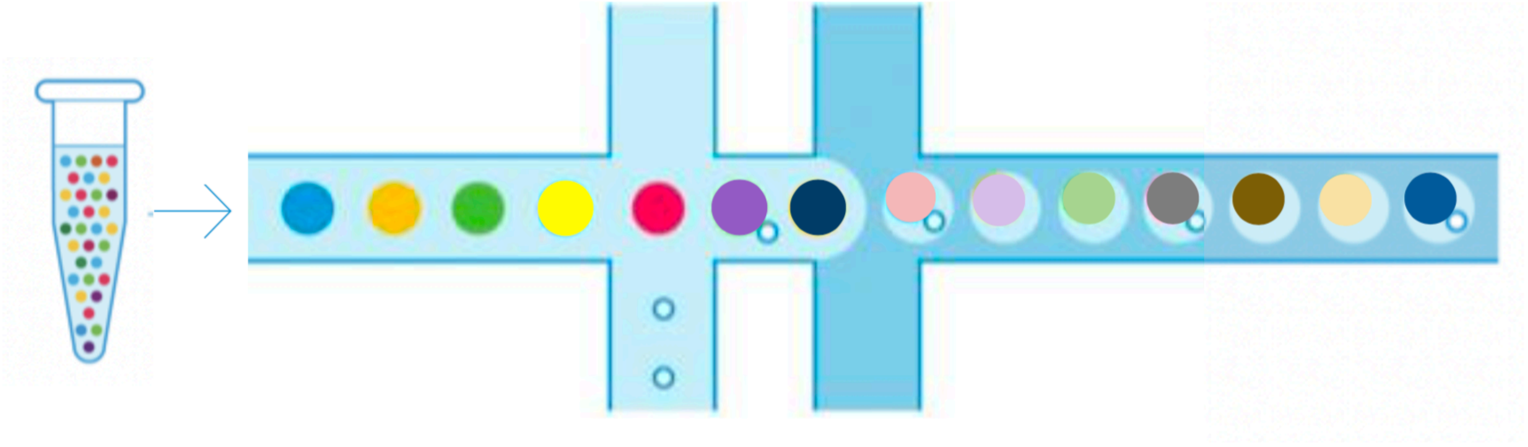


Gel Beads

Sample

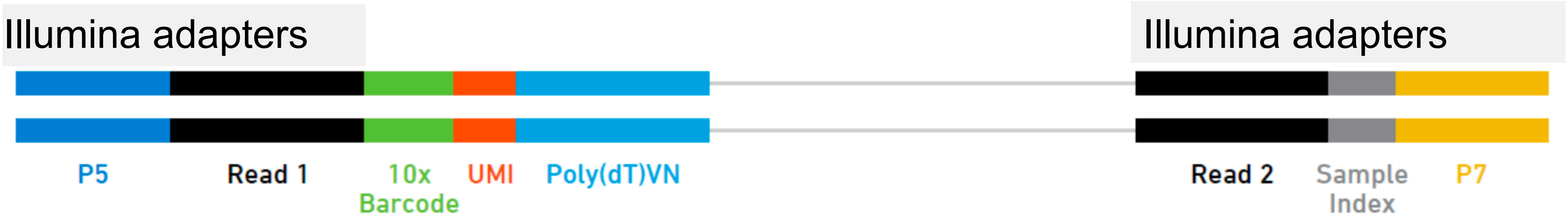
Oil

Droplets with Gel Beads



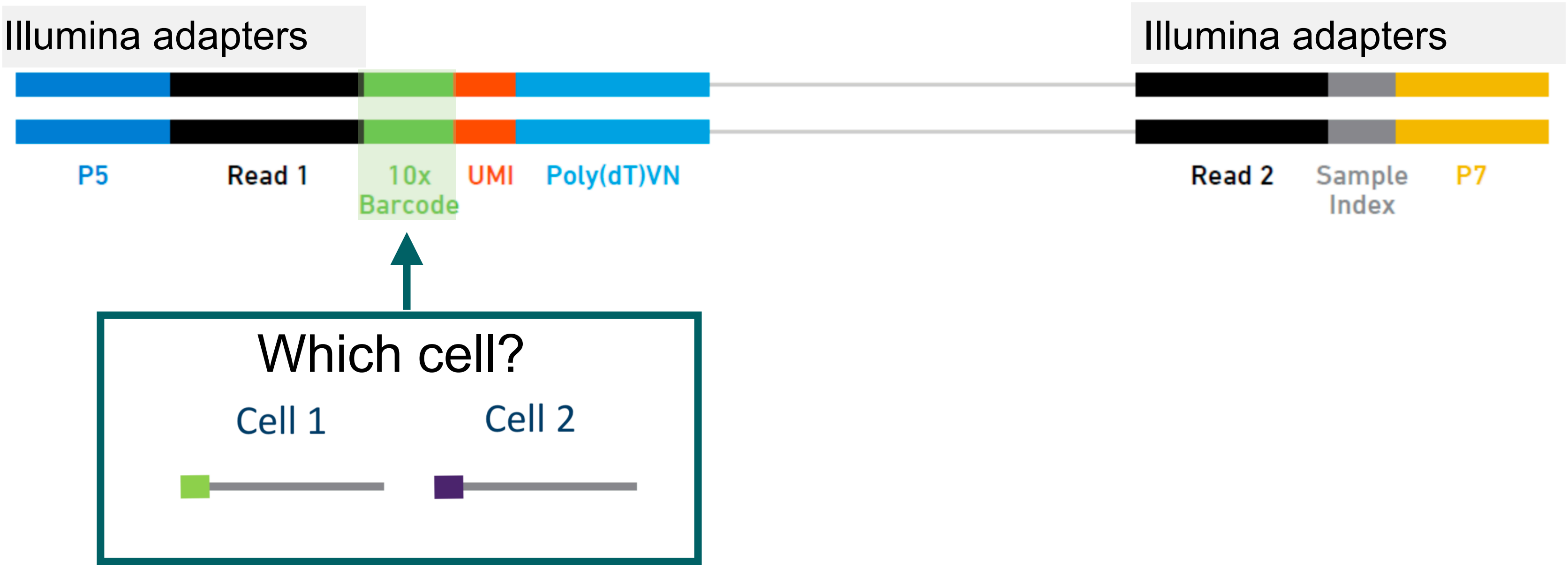
Source: 10x genomics

Basics Bioinformatics - Transcript Counts



Source: 10x genomics

Basics Bioinformatics - Transcript Counts



Illumina adapters

Illumina adapters

P5

Read 1

10x Barcode

UMI

Poly(dT)VN

Read 2

Sample Index

P7

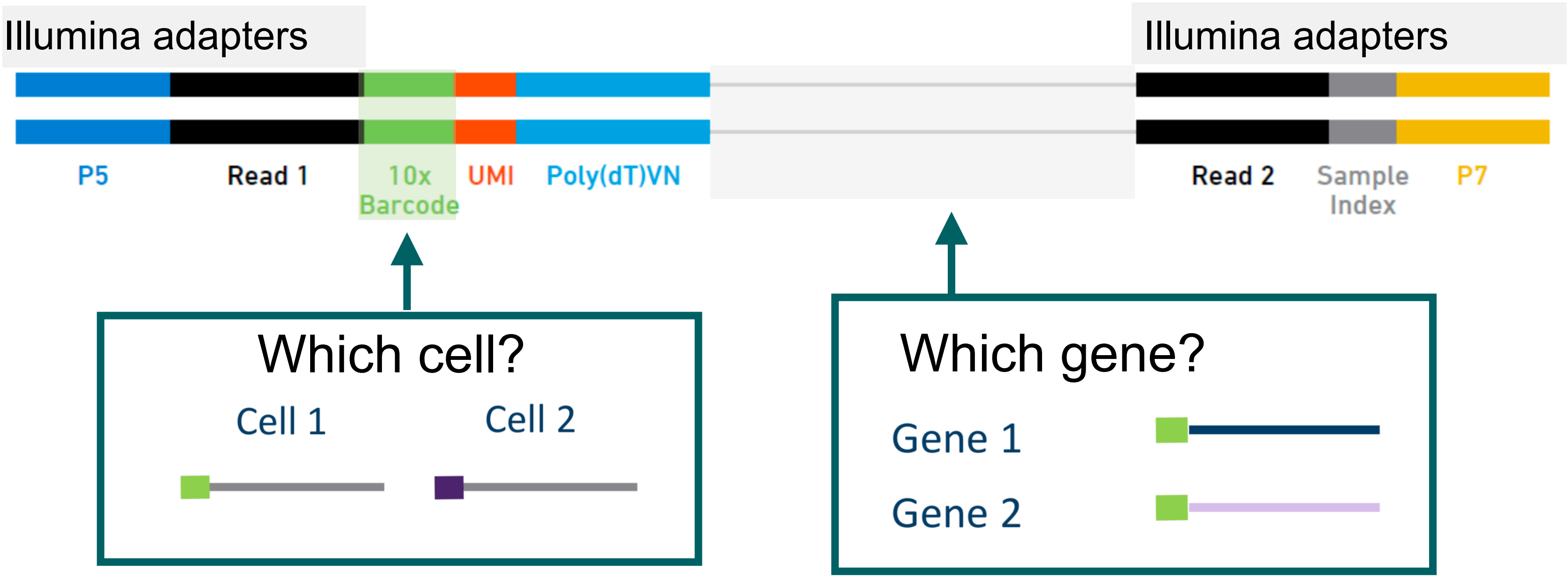
Which cell?

Cell 1

Cell 2

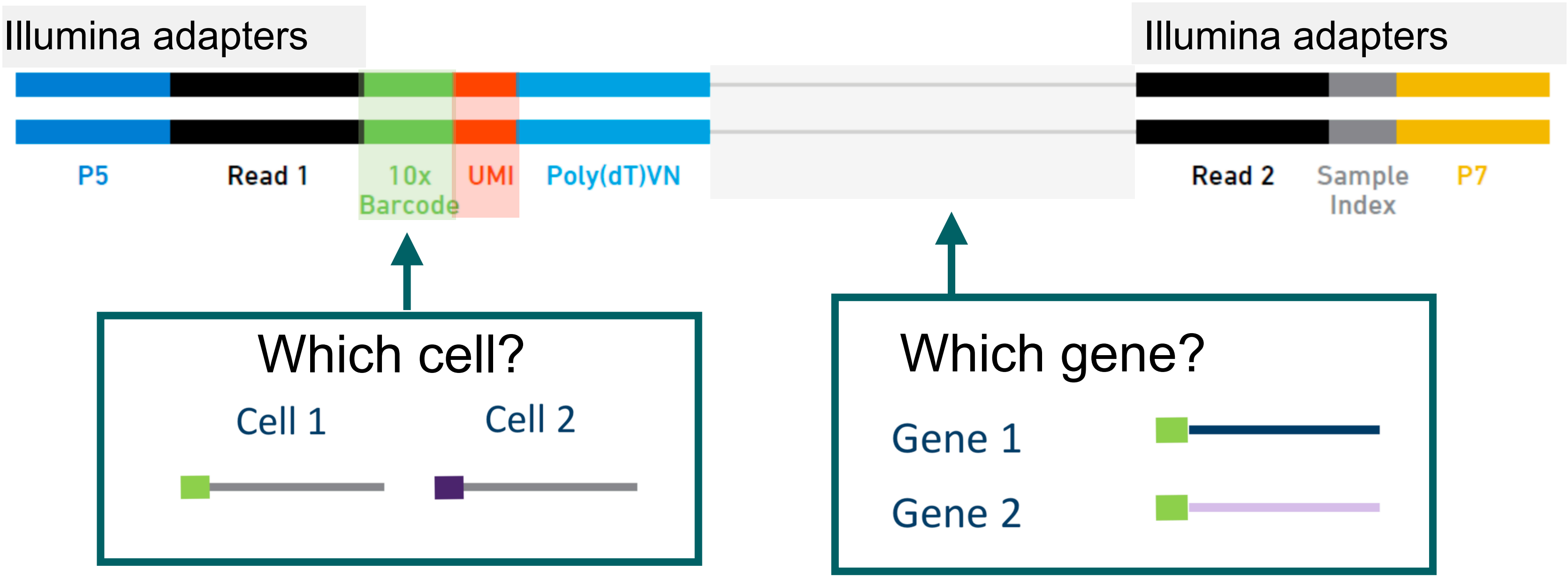
Source: 10x genomics

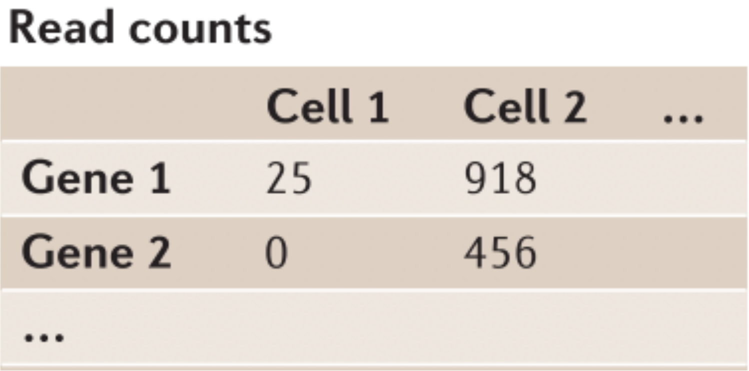
Basics Bioinformatics - Transcript Counts



Source: 10x genomics

Basics Bioinformatics - Transcript Counts



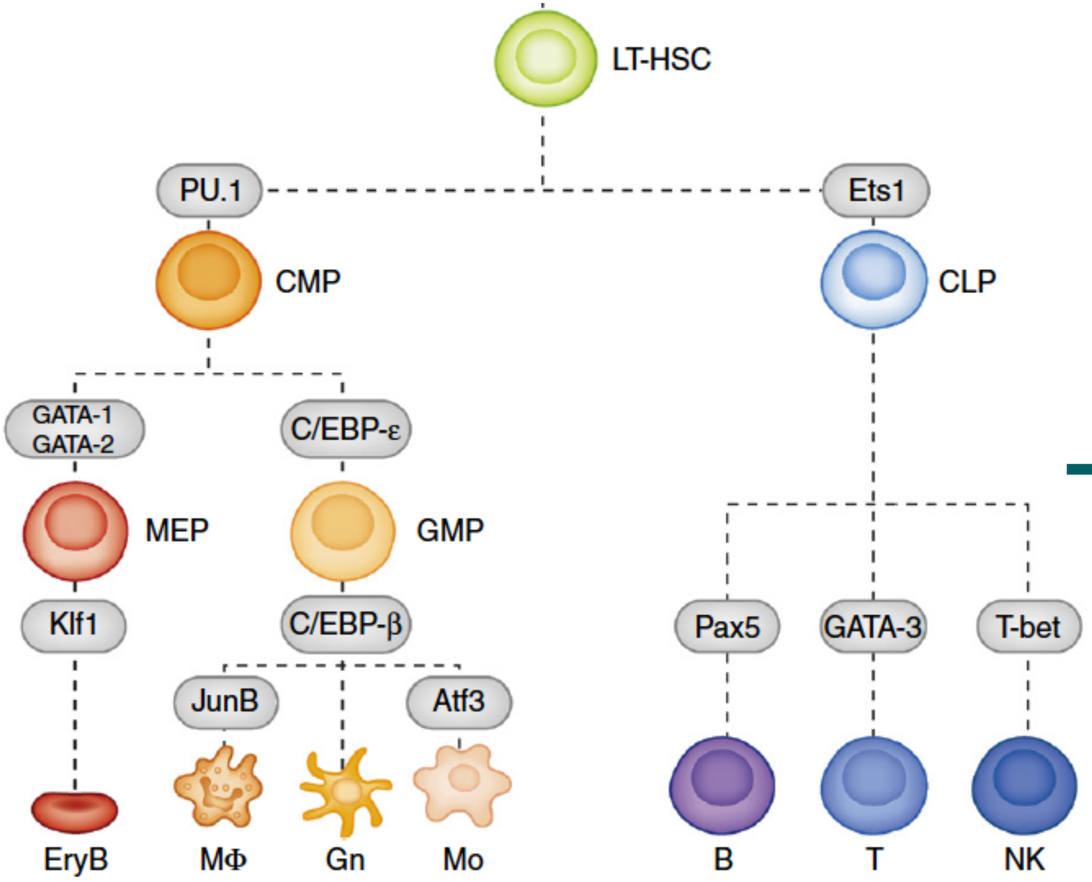


Read counts

	Cell 1	Cell 2	...
Gene 1	25	918	
Gene 2	0	456	
...			

Source: 10x genomics

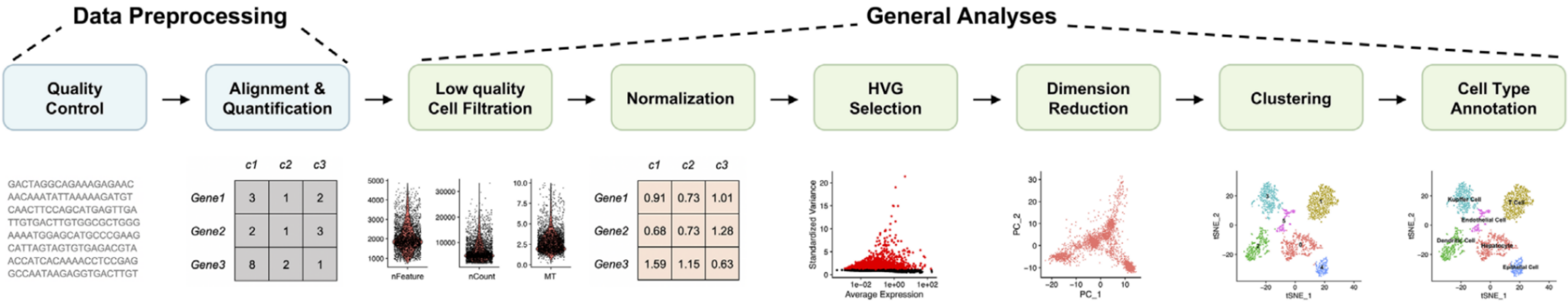
Cell Differentiation & Gene Expression



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

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

Basics Bioinformatics - single cell RNA-seq



Data Preprocessing

General Analyses

Quality Control → Alignment & Quantification → Low quality Cell Filtration → Normalization → HVG Selection → Dimension Reduction → Clustering → Cell Type Annotation

GACTAGGCAGAAAGAGAAC
AACAAATATTAAAAAGATGT
CAACTTCCAGCATGAGTTGA
TTGTGACTTGTGGCGCTGGG
AAAATGGAGCATGCCCGAAG
CATTAGTAGTGTGAGACGTA
ACCATCACAAAACCTCCGAG
GCCAATAAGAGGTGACTTGT

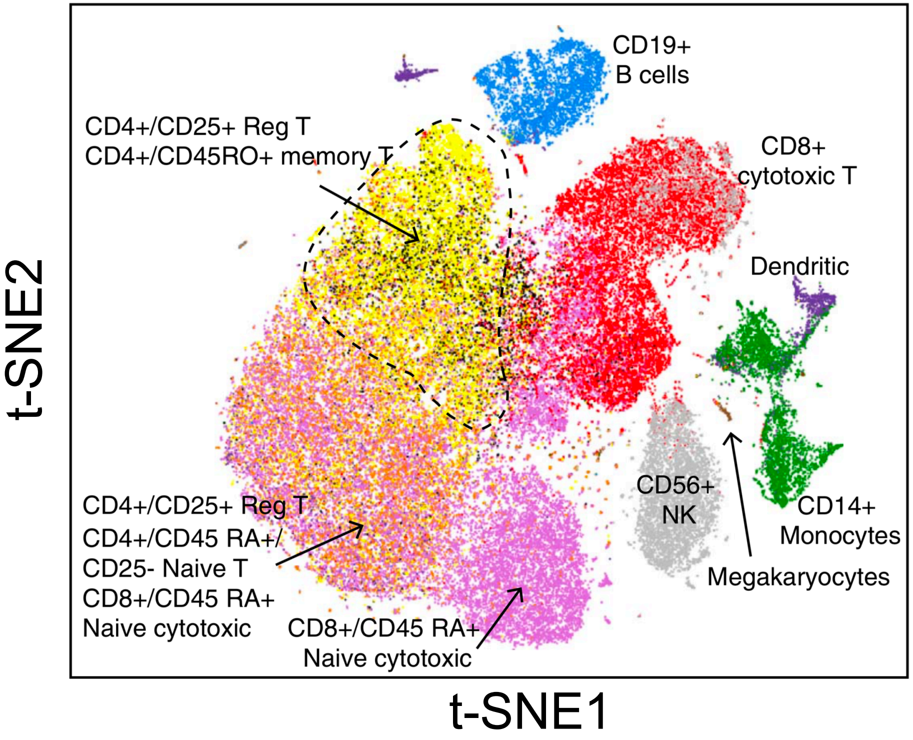
	c1	*c2*	*c3*
Gene1	3	1	2
Gene2	2	1	3
Gene3	8	2	1

	c1	*c2*	*c3*
Gene1	0.91	0.73	1.01
Gene2	0.68	0.73	1.28
Gene3	1.59	1.15	0.63

D Jovic, X Liang, H Zeng, L Lin, F Xu, Y Luo, 2022

Gene Expression of Lymphoid Cells

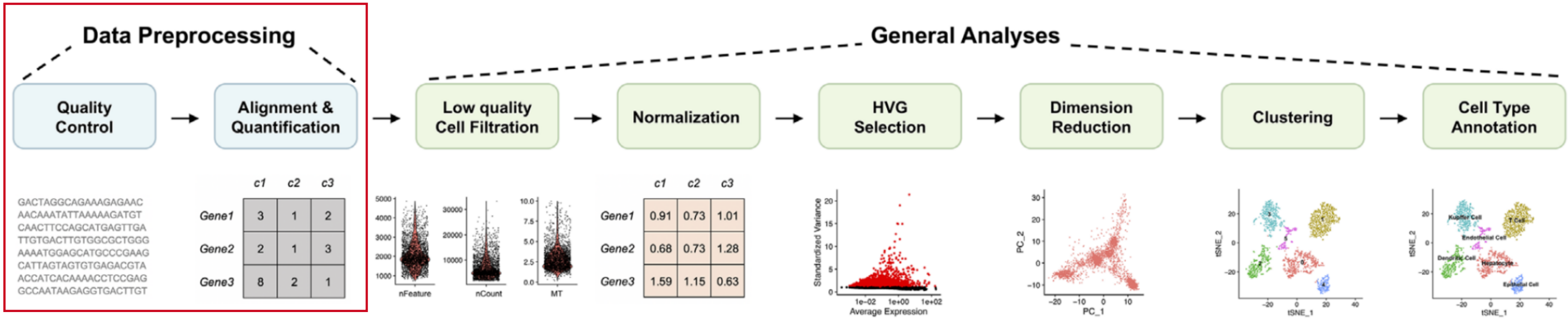
PBMCs from Humans



Single cell RNA-seq from 68k cells

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

Basics Bioinformatics - single cell RNA-seq



Data Preprocessing

Quality Control → Alignment & Quantification

General Analyses

→ Low quality Cell Filtration → Normalization → HVG Selection → Dimension Reduction → Clustering → Cell Type Annotation

GACTAGGCAGAAAGAGAAC
AACAAATATTAAAAAGATGT
CAACTTCCAGCATGAGTTGA
TTGTGACTTGTGGCGCTGGG
AAAATGGAGCATGCCCGAAG
CATTAGTAGTGTGAGACGTA
ACCATCACAAAACCTCCGAG
GCCAATAAGAGGTGACTTGT

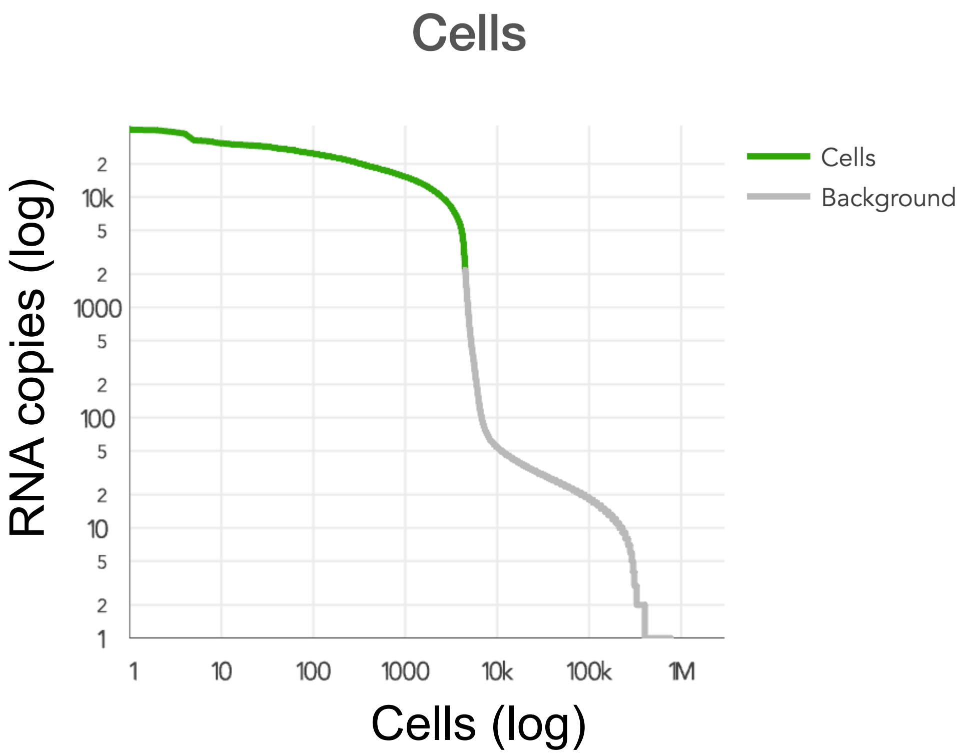
	c1	*c2*	*c3*
Gene1	3	1	2
Gene2	2	1	3
Gene3	8	2	1

	c1	*c2*	*c3*
Gene1	0.91	0.73	1.01
Gene2	0.68	0.73	1.28
Gene3	1.59	1.15	0.63

D Jovic, X Liang, H Zeng, L Lin, F Xu, Y Luo, 2022

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

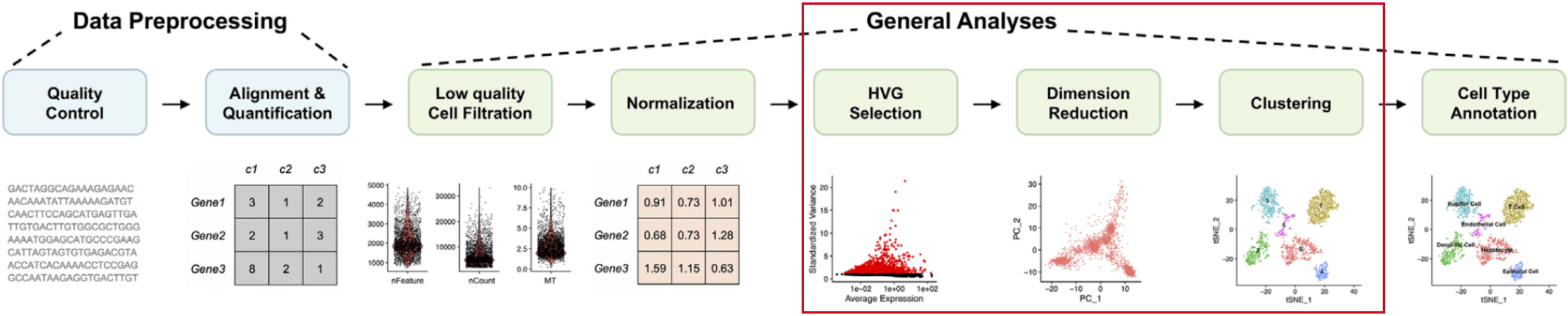
89,289

Median Genes per Cell

2,504

cell ranger - 10x genomics

Basics Bioinformatics - single cell RNA-seq



Data Preprocessing

General Analyses

Quality Control → Alignment & Quantification → Low quality Cell Filtration → Normalization → HVG Selection → Dimension Reduction → Clustering → Cell Type Annotation

GACTAGGCAGAAAGAGAAC
AACAAATATTAAAAAGATGT
CAACTTCCAGCATGAGTTGA
TTGTGACTTGTGGCGCTGGG
AAAATGGAGCATGCCCGAAG
CATTAGTAGTGTGAGACGTA
ACCATCACAAAACCTCCGAG
GCCAATAAGAGGTGACTTGT

	c1	*c2*	*c3*
Gene1	3	1	2
Gene2	2	1	3
Gene3	8	2	1

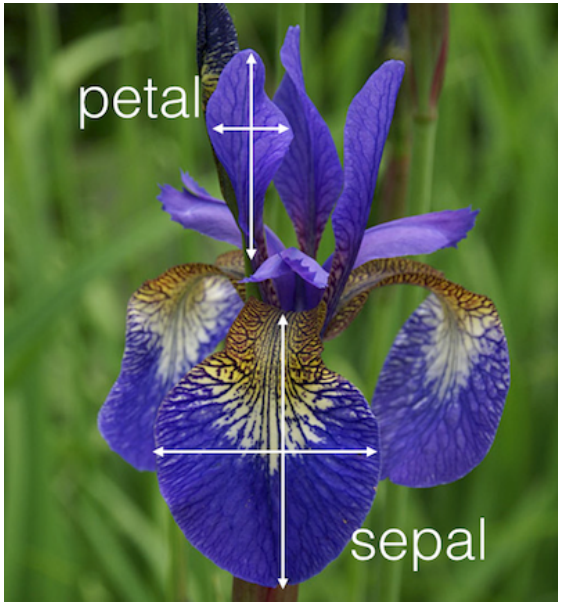
	c1	*c2*	*c3*
Gene1	0.91	0.73	1.01
Gene2	0.68	0.73	1.28
Gene3	1.59	1.15	0.63

D Jovic, X Liang, H Zeng, L Lin, F Xu, Y Luo, 2022

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

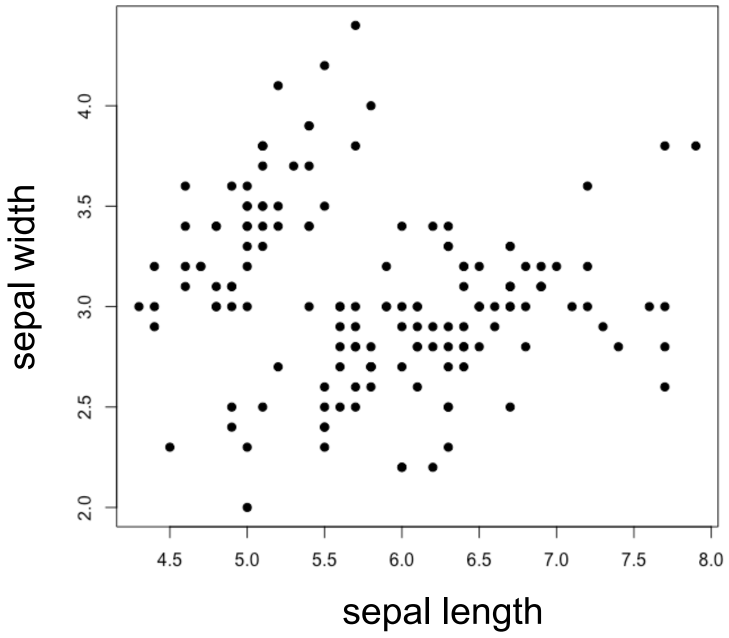


	Sepal Length	*Sepal Width*	*Petal Length*	*Petal Width*
Flower 1	*5.1*	*3.5*	*1.4*	*0.2*
Flower 2	*4.9*	*3.0*	*1.4*	*0.2*
Flower 3	*4.7*	*3.2*	*1.3*	*0.2*
Flower 4	*4.6*	*3.1*	*1.5*	*0.2*
…	*…*	*…*	*…*	*…*

Clustering

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

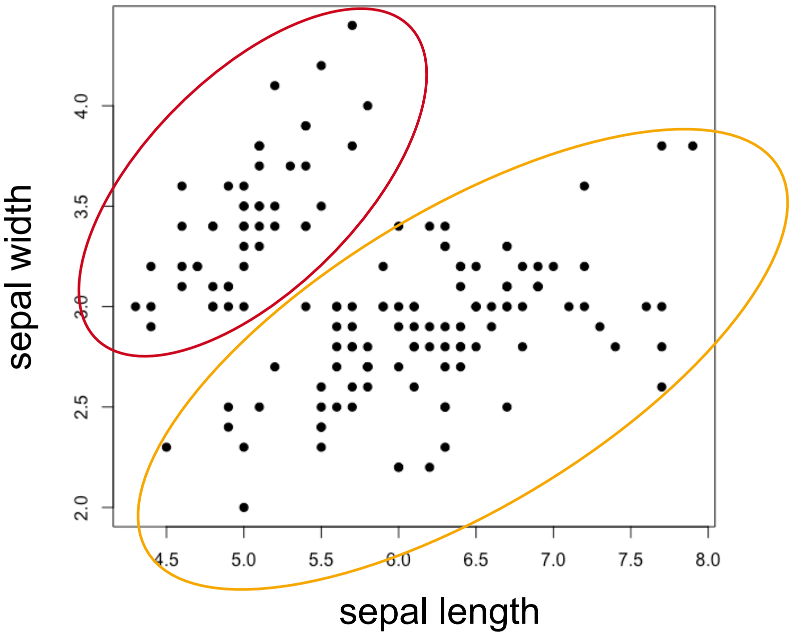
	Sepal Length	*Sepal Width*	*Petal Length*	*Petal Width*
Flower 1	*5.1*	*3.5*	*1.4*	*0.2*
Flower 2	*4.9*	*3.0*	*1.4*	*0.2*
Flower 3	*4.7*	*3.2*	*1.3*	*0.2*
Flower 4	*4.6*	*3.1*	*1.5*	*0.2*
…	*…*	*…*	*…*	*…*



Clustering

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

	Sepal Length	*Sepal Width*	*Petal Length*	*Petal Width*
Flower 1	***5.1***	***3.5***	***1.4***	***0.2***
Flower 2	***4.9***	***3.0***	***1.4***	***0.2***
Flower 3	***4.7***	***3.2***	***1.3***	***0.2***
Flower 4	***4.6***	***3.1***	***1.5***	***0.2***
…	***…***	***…***	***…***	***…***

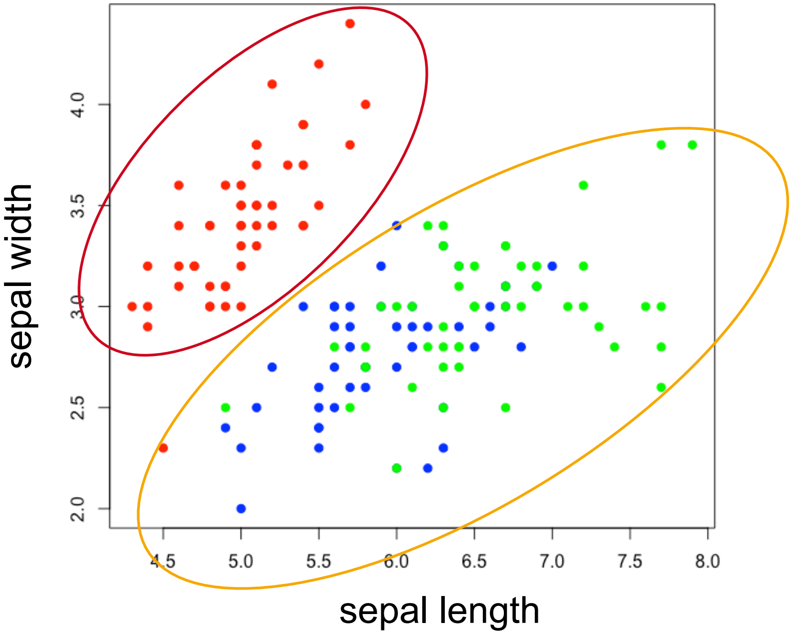


Clustering

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

Iris Virginia

Iris Setosa



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

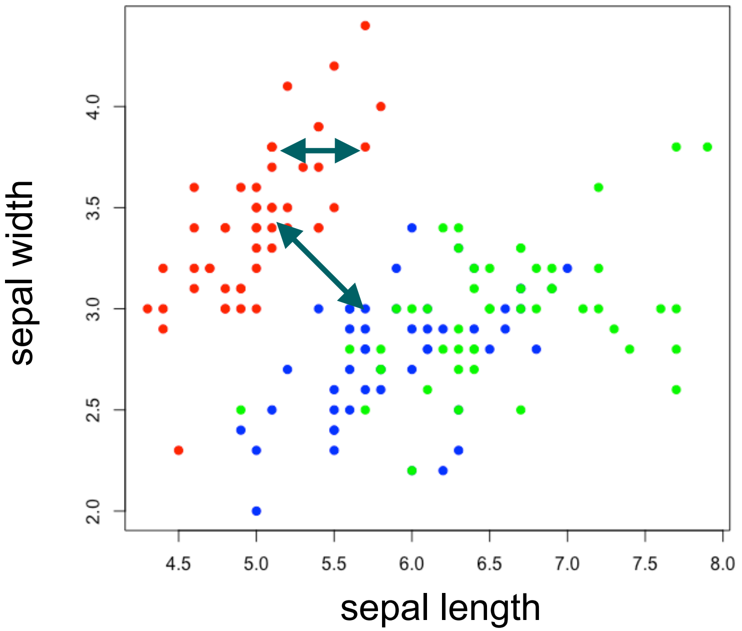
x_{11}	x_{12}	…	x_{1L}
x_{21}	x_{22}	…	x_{2L}
x_{31}	x_{32}	…	x_{3L}
…	…	…	…
x_{N1}	x_{N2}	…	x_{NL}

- **find groups of similar observations**
  - *vector* $Y = (y_1, \ldots, y_N)$

    where $y_i \in \{1, \ldots, K\}$ 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

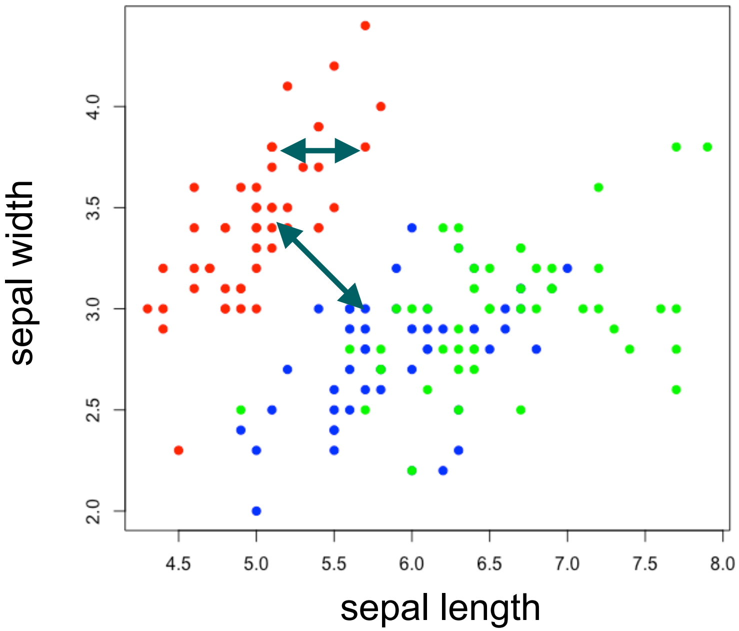


Euclidean distance (sensitive to scale)

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

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)

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

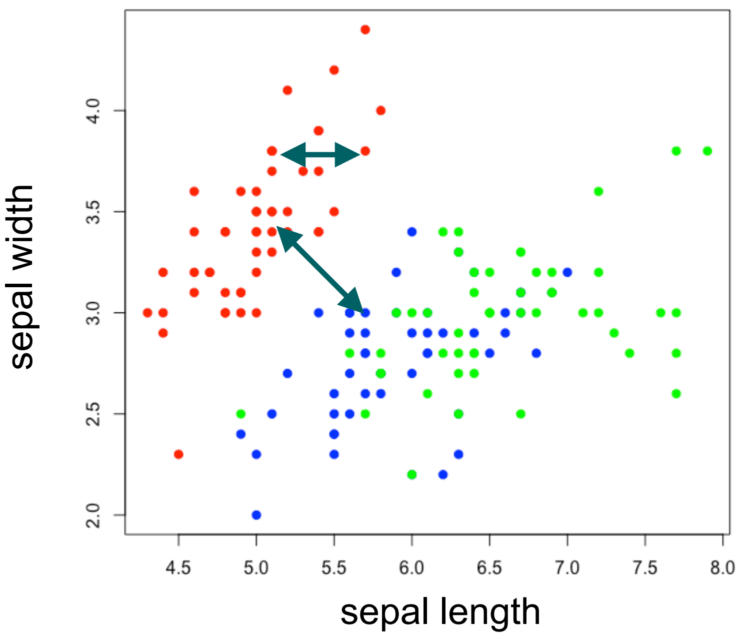
Pearson Correlation

(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

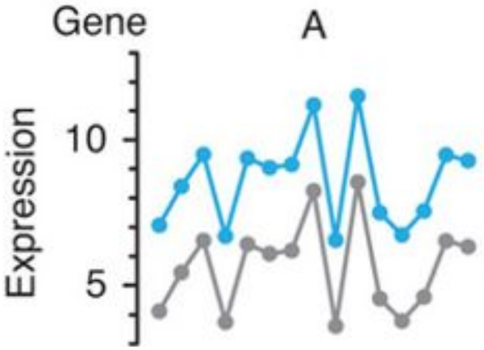
(scale insensitive/ similarity)

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

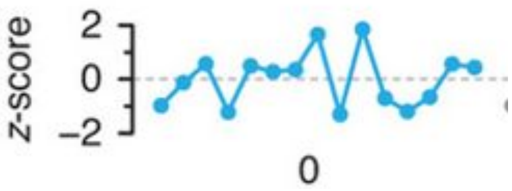
Distance and Scale

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

unscaled data



z-score normalised data



$$z = \frac{x_{ij} - \mu_i}{\sigma_i}$$

Euclidean - not similar

Correlation - similar

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

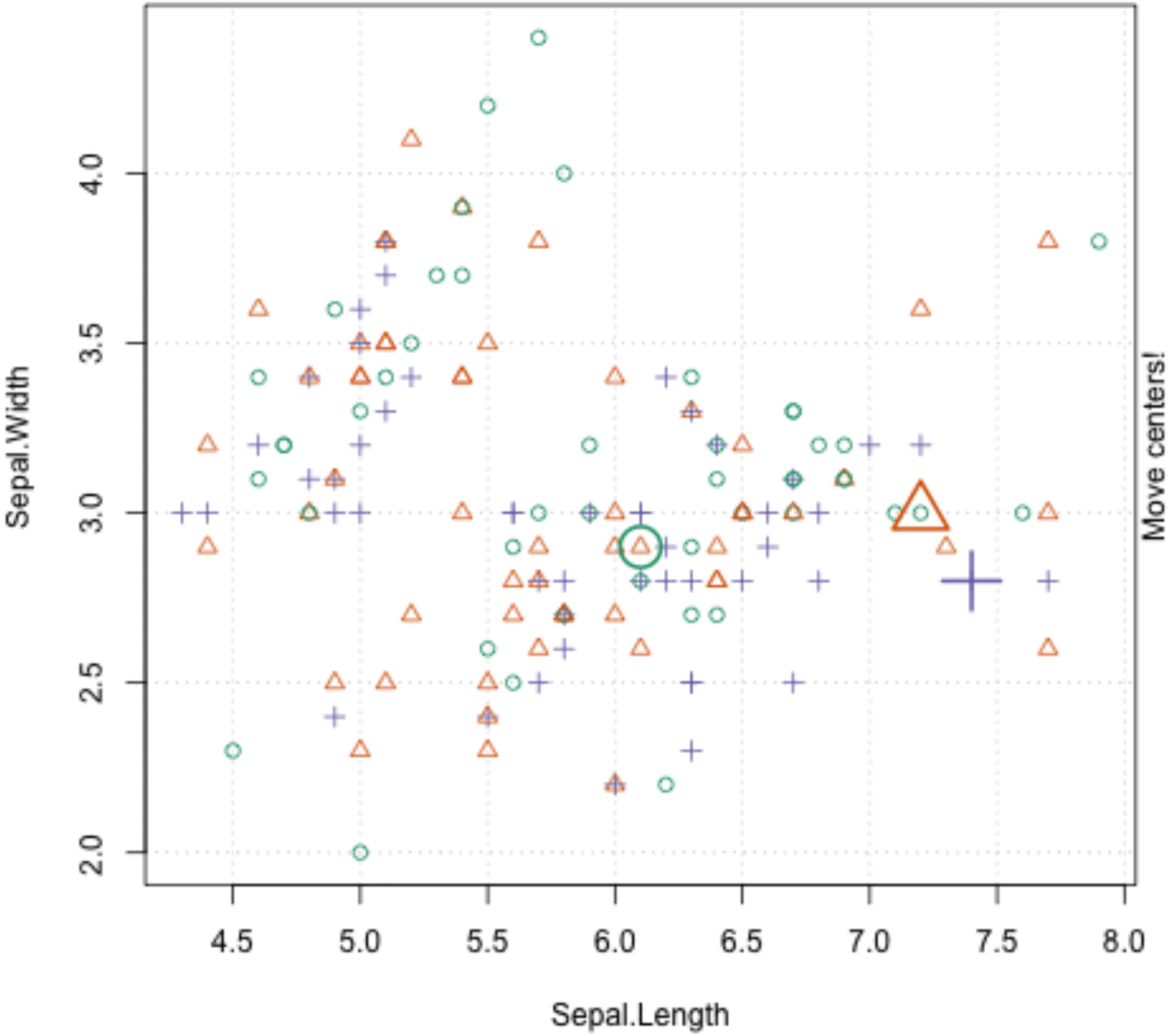
Requires specification of number of clusters (***K***)

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:
        $y_i = \text{argmin}_k \, d(x_i, m_k)$
```

* convergence is only guaranteed for Euclidean distance

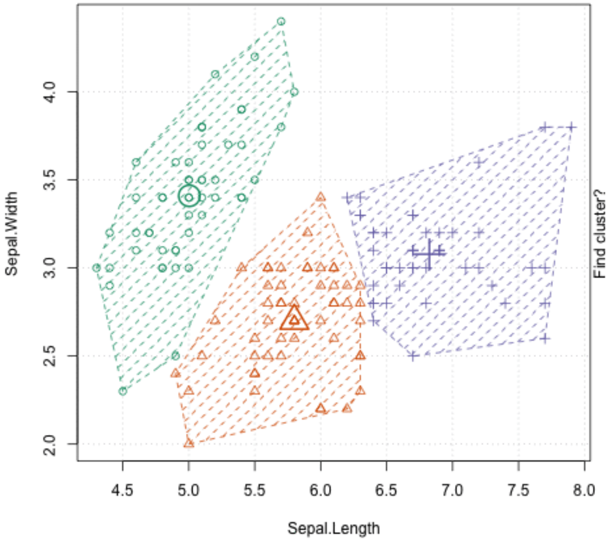
K-means on Iris



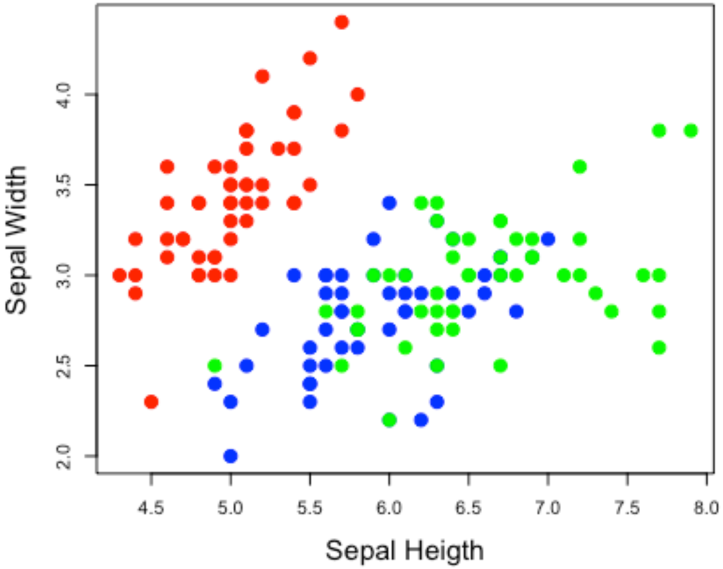
Move centers!

K-means on Iris

K-means solutions

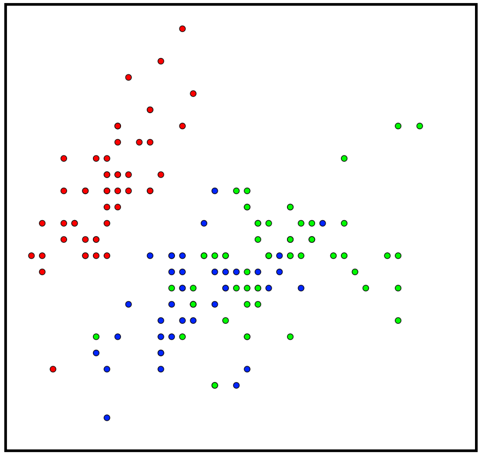


True labels



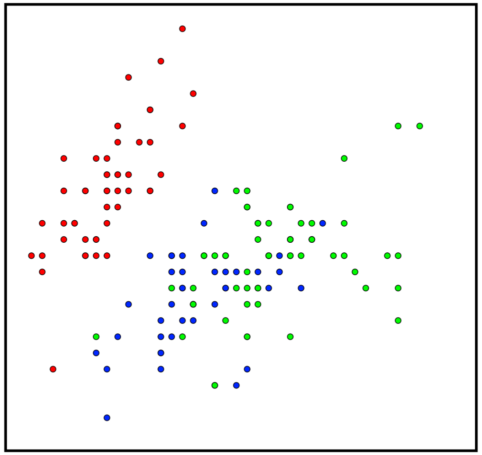
- K-means tends to find spherical clusters
- Sensitive to initialisation

Graph based clustering

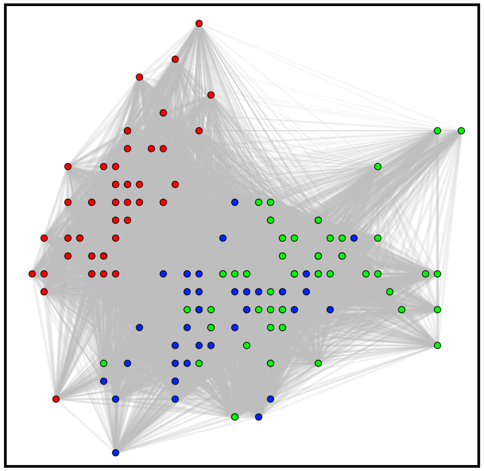


- data points are nodes

Graph based clustering

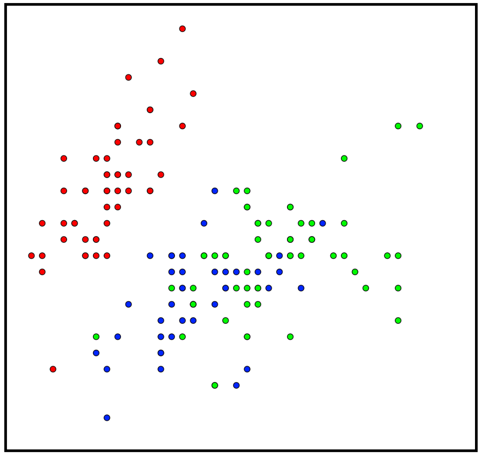


- data points are nodes

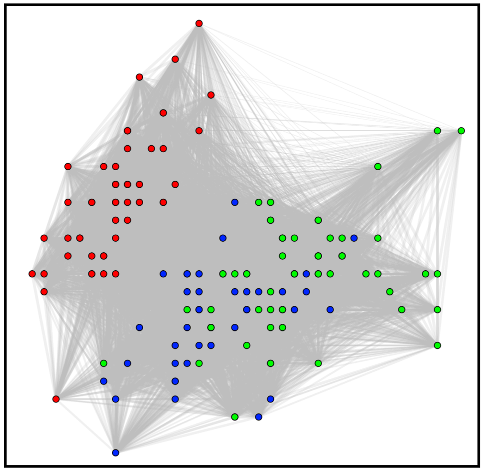


- edges represent similarities

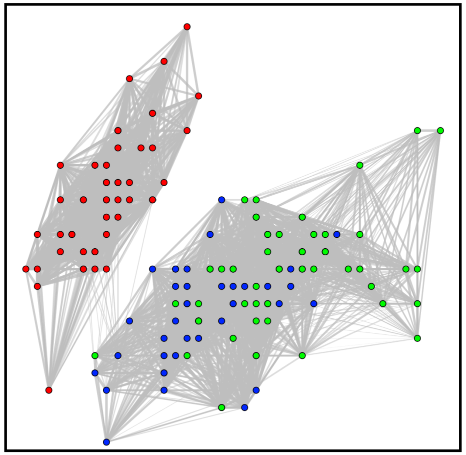
Graph based clustering



- data points are nodes

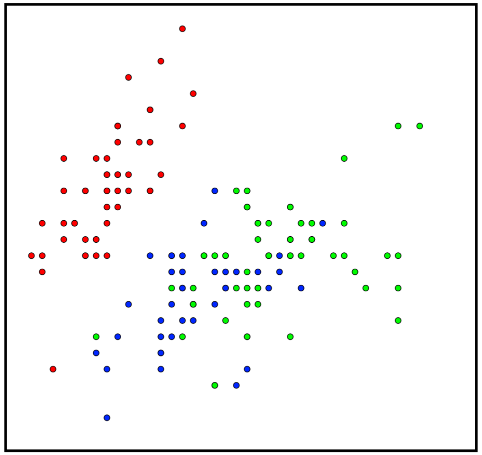


- edges represent similarities

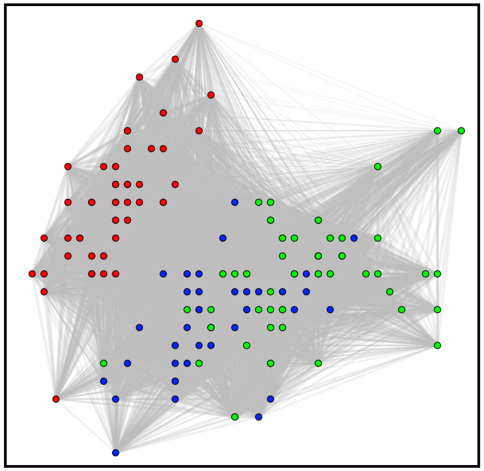


- k-nearest neighbours (KNN) -> sparse graphs

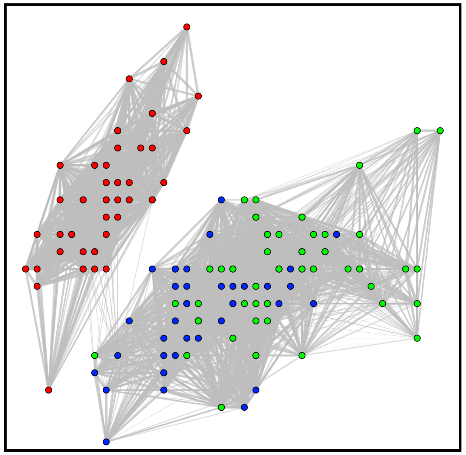
Graph based clustering



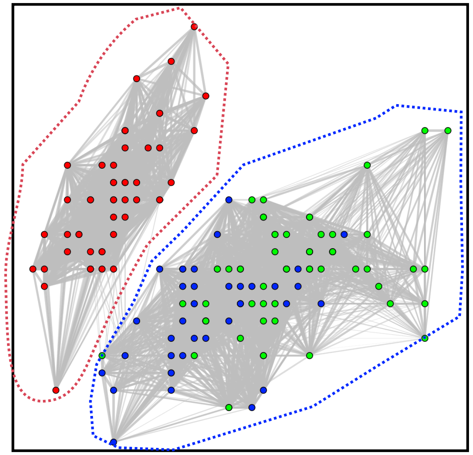
- data points are nodes



- edges represent similarities

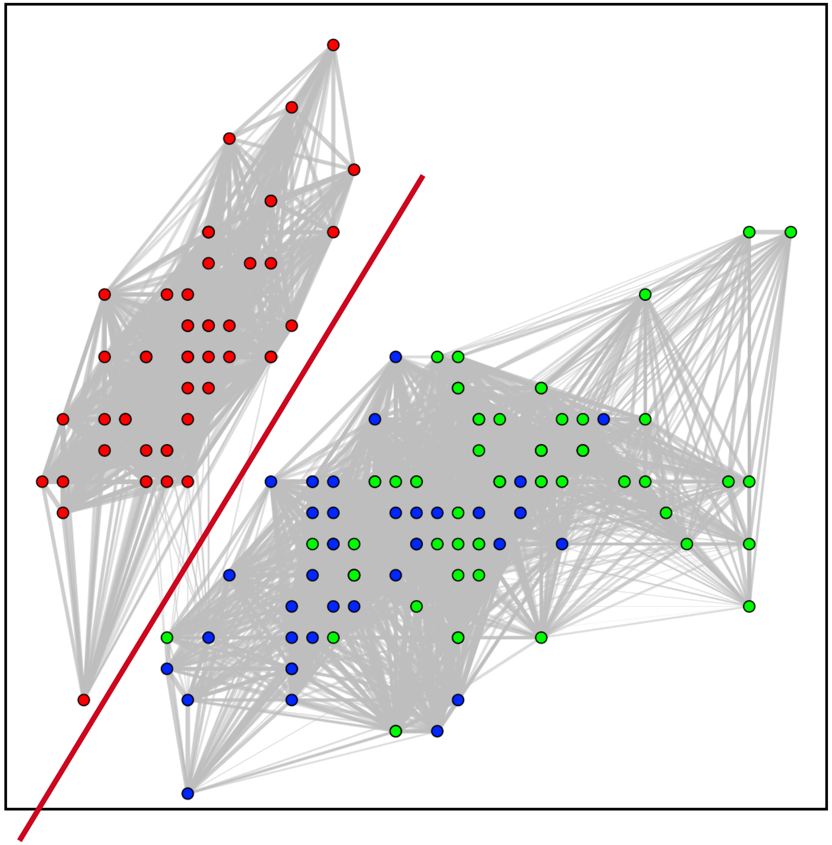


- k-nearest neighbours (KNN) -> sparse graphs



- find well connected sub-graphs

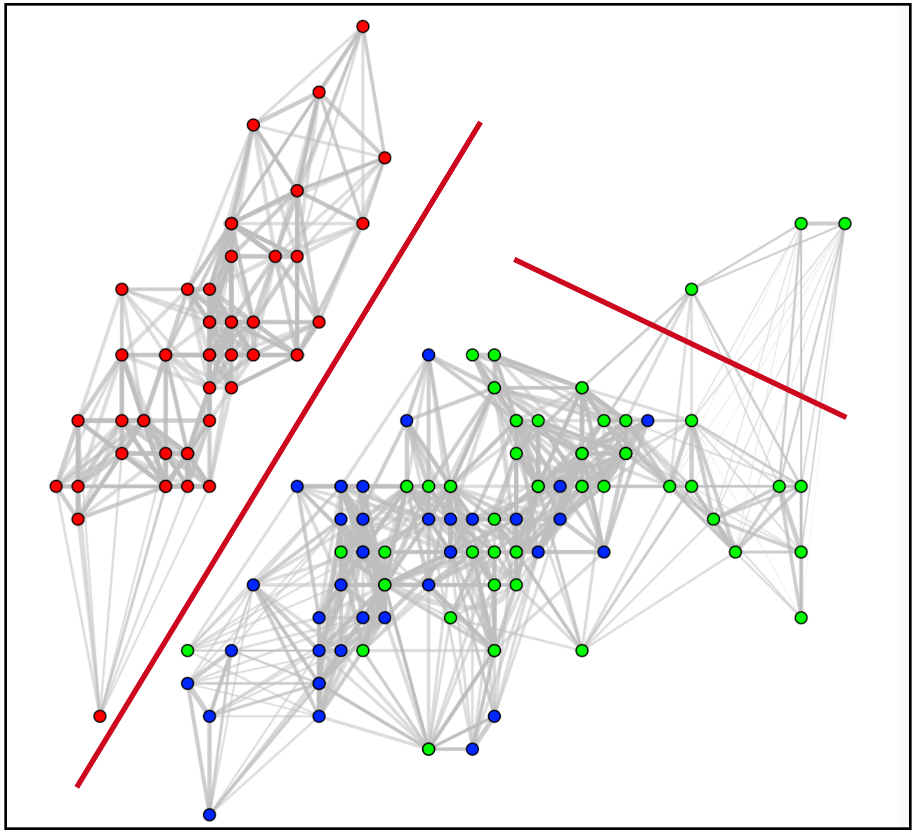
Graph cut



KNN = 50

- Cluster by finding cuts in the graph
- Cut cost **C(*A*,*B*)** = sum of edge weights in cut

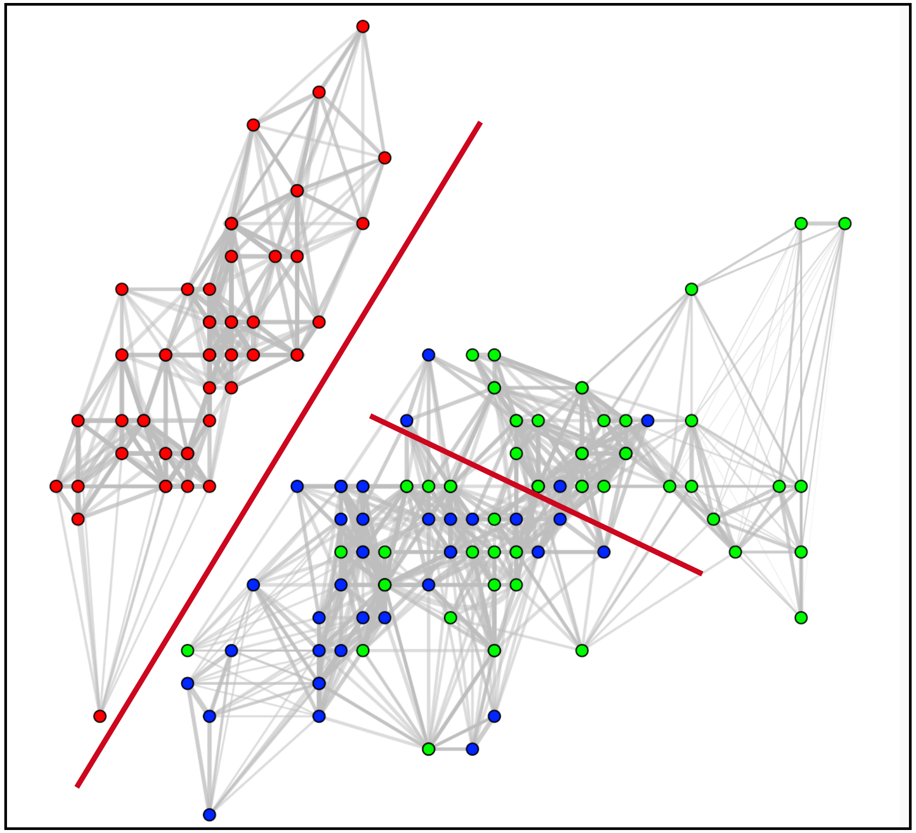
Graph cut



KNN = 10

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

Normalized graph cut



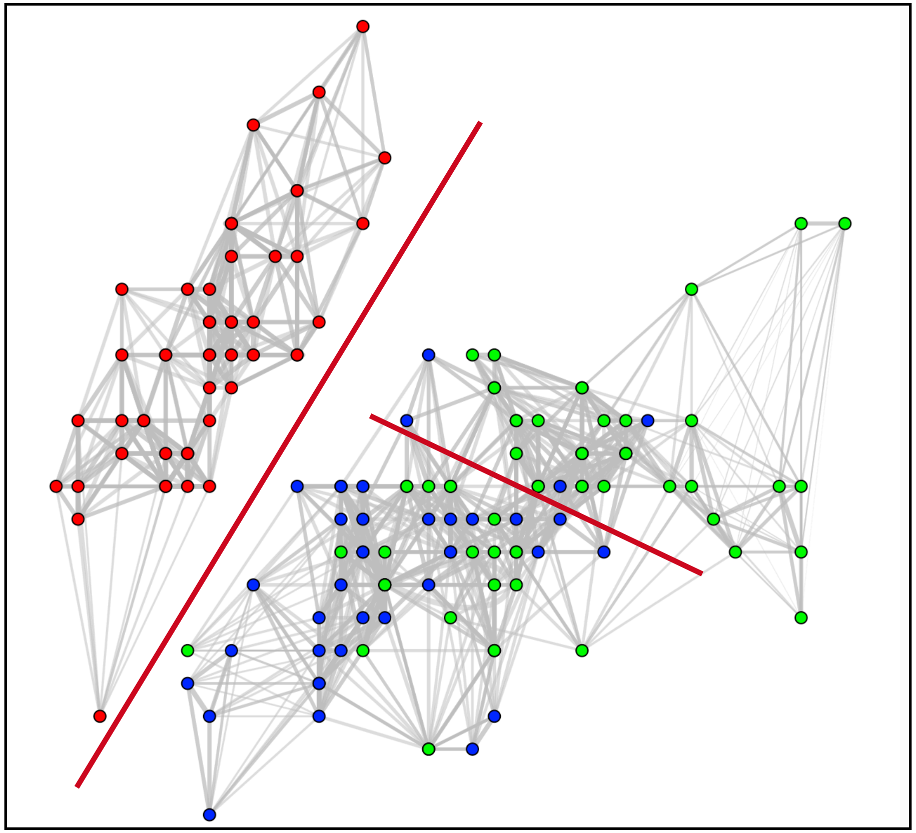
KNN = 10

- 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

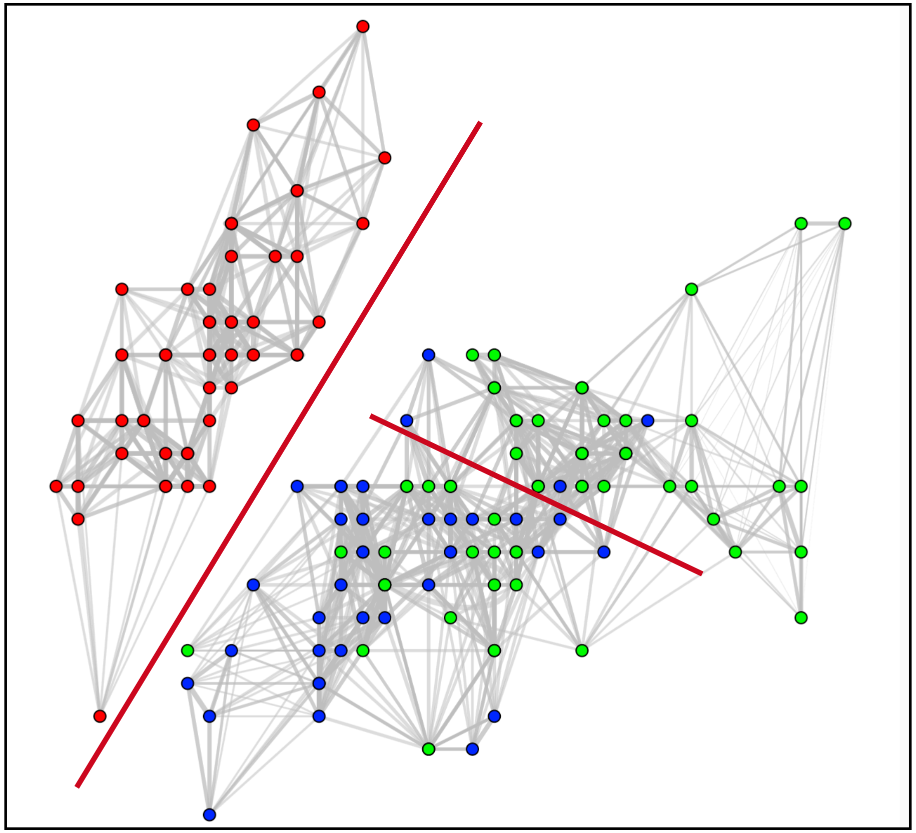
  - where *D* is a diagonal matrix with the number of neighbours of a node
- If we perform a spectral analysis of *L**

  $$L\lambda = u\lambda$$

  - 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**

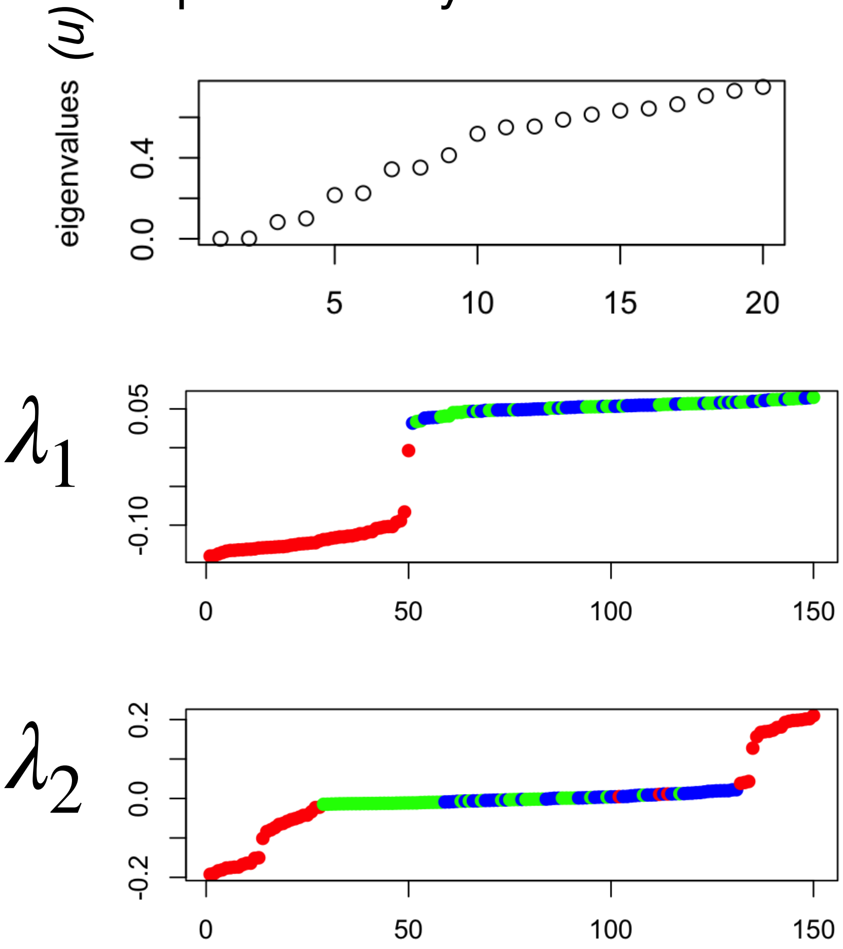
  $$L\lambda = u\lambda$$

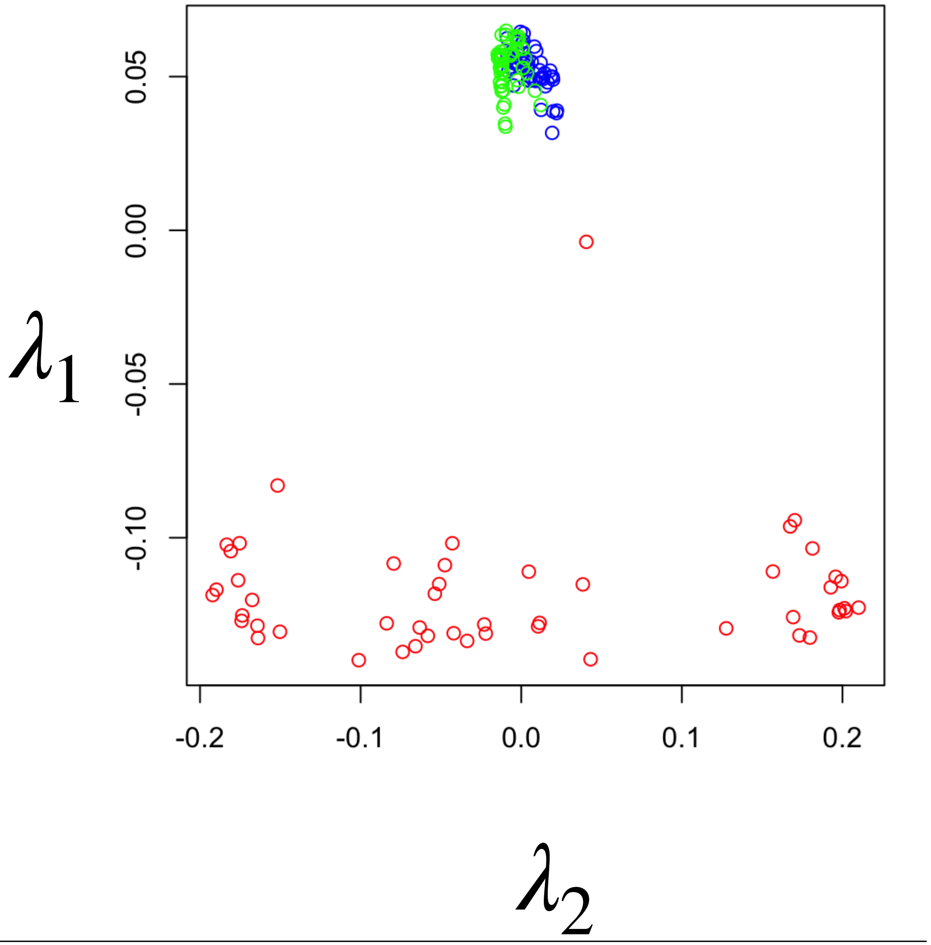
  - eigenvectors (λ) provides CUTs in the graph
  - eigenvalues (*u*) provides the cost of the CUT.
  - Perform *k*-means on lowest K eigenvalues

* 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: $L\lambda = u\lambda$

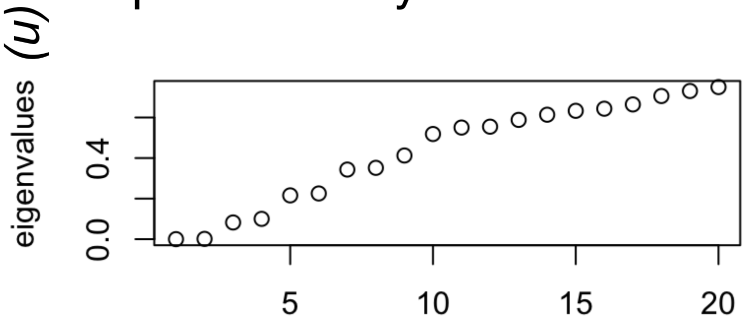


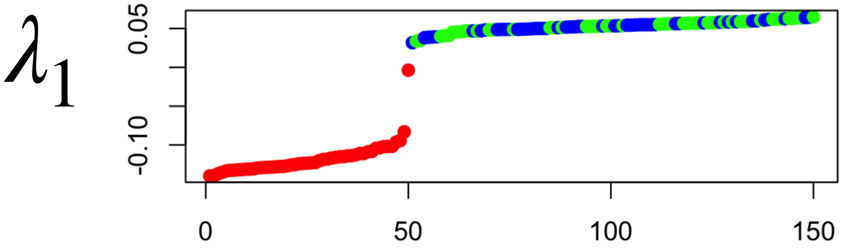


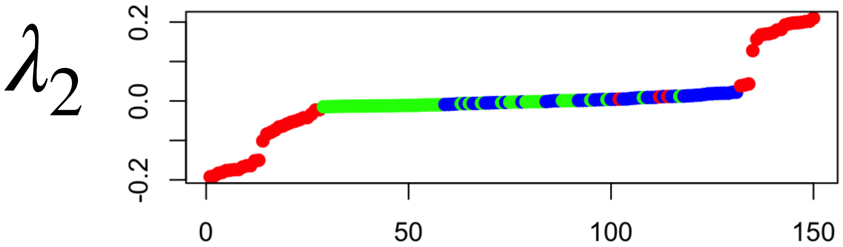
http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

Graph cut and spectral analysis of laplacian matrices

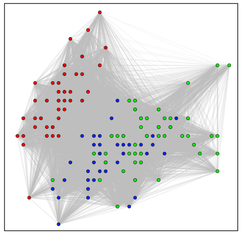
Spectral analysis: $L\lambda = u\lambda$

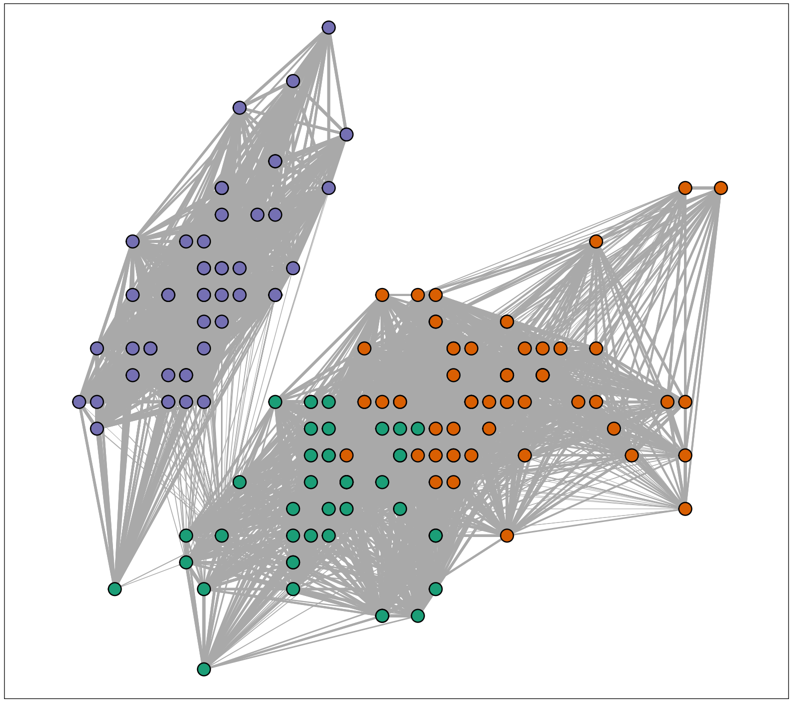






original labels

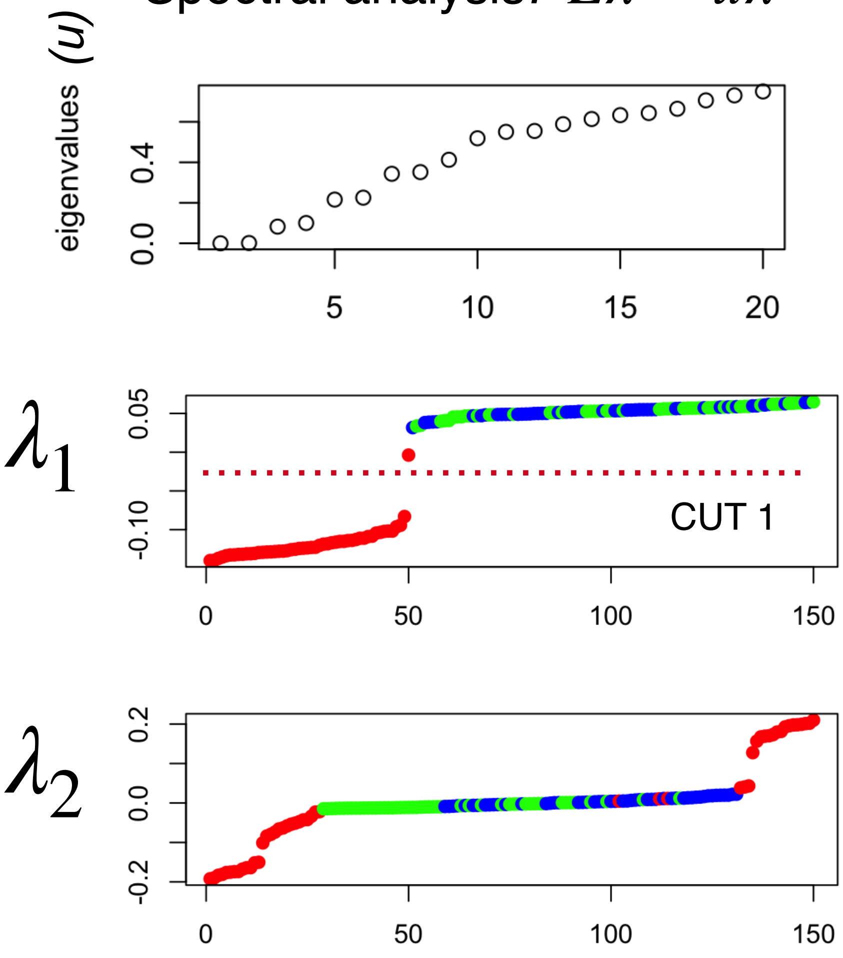




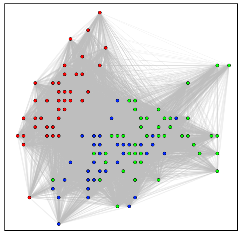
http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

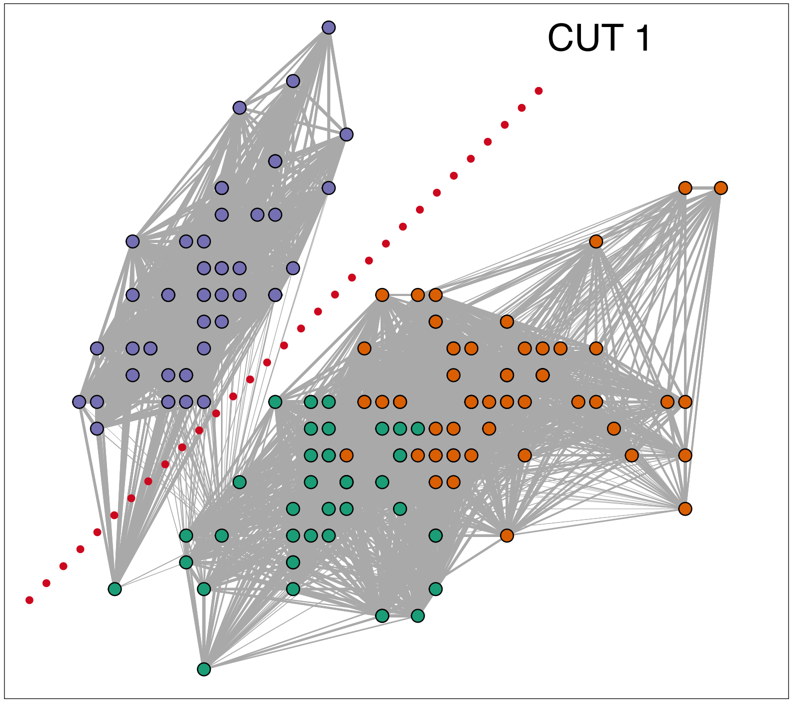
Graph cut and spectral analysis of laplacian matrices

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



original labels

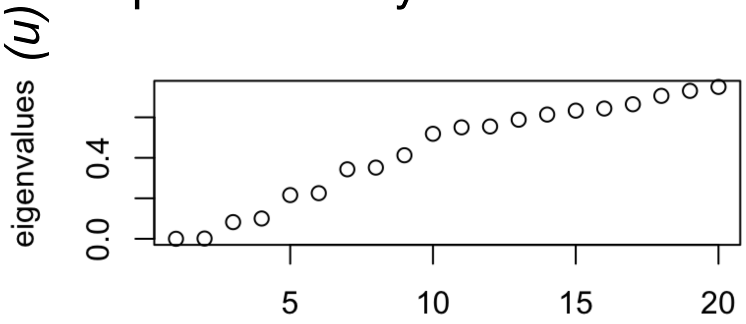




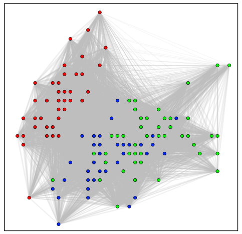
http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

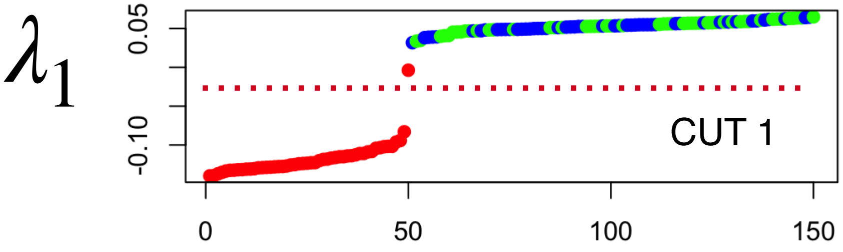
Graph cut and spectral analysis of laplacian matrices

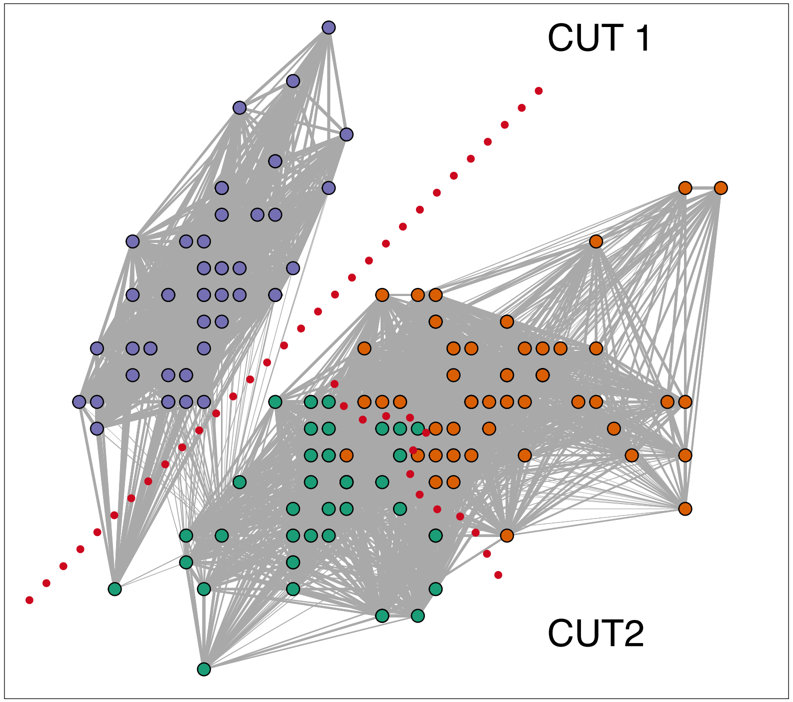
Spectral analysis: $L\lambda = u\lambda$

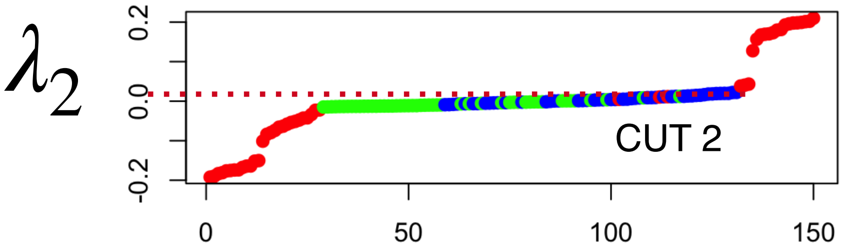


original labels





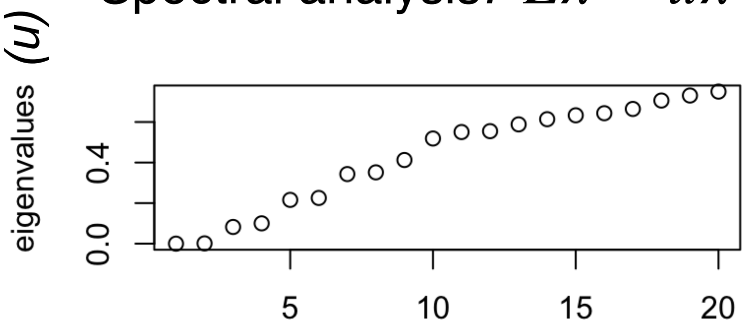


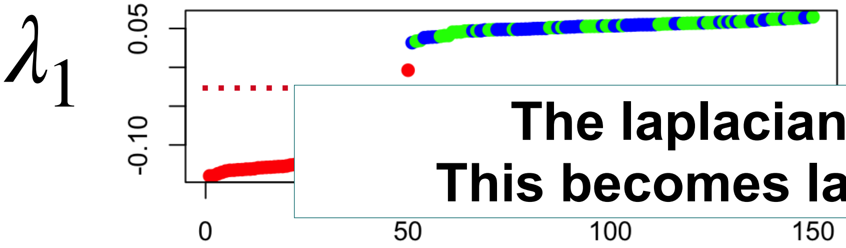


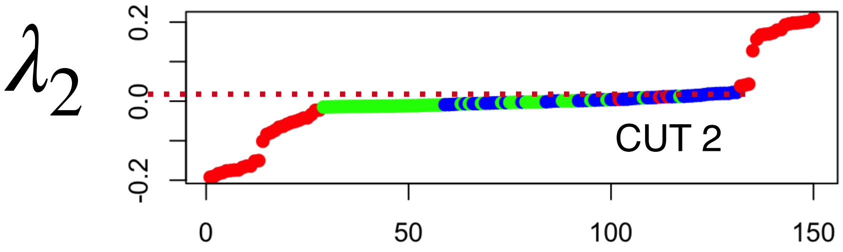
http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

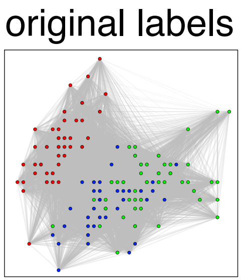
Graph cut and spectral analysis of laplacian matrices

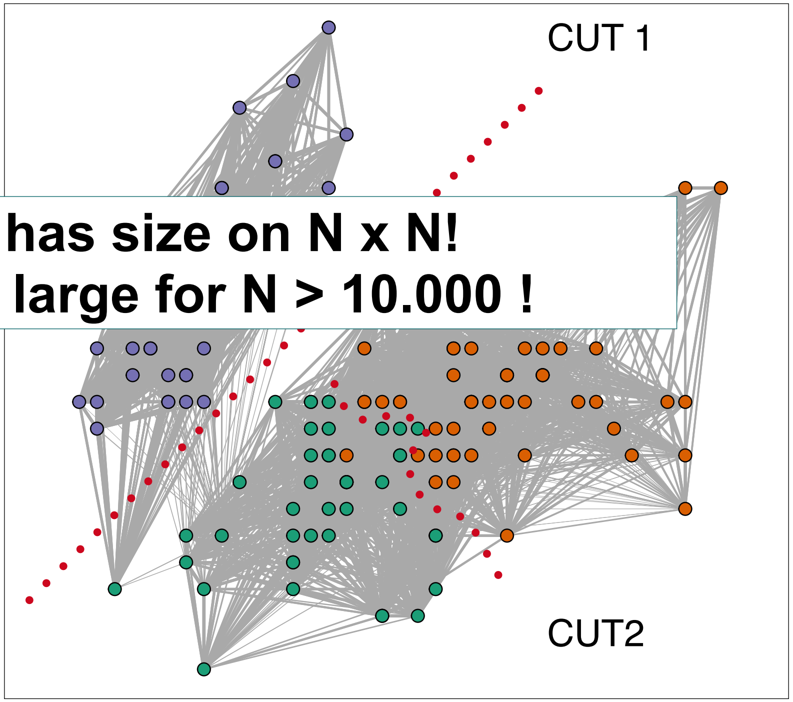
Spectral analysis: $L\lambda = u\lambda$







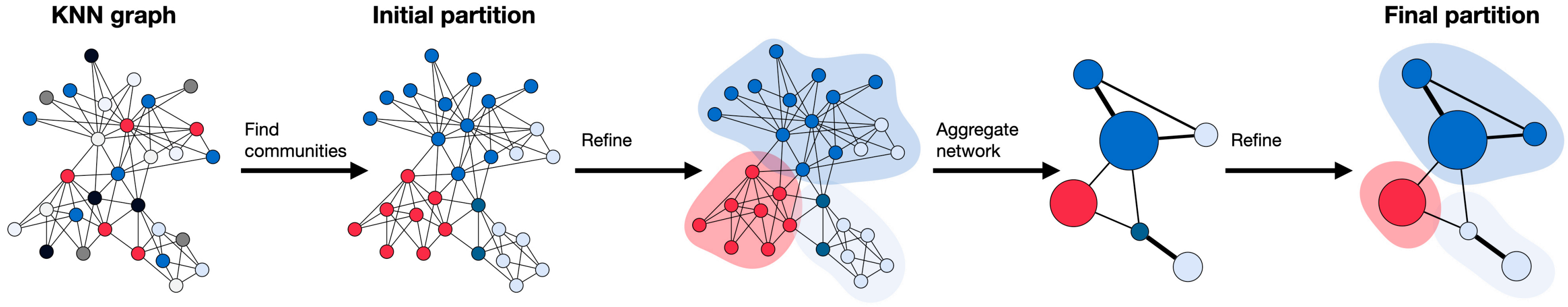




The laplacian matrix has size on N x N!
This becomes large too large for N > 10.000 !

http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

Single cell Clustering / Louvain & Leiden algorithm



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

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*

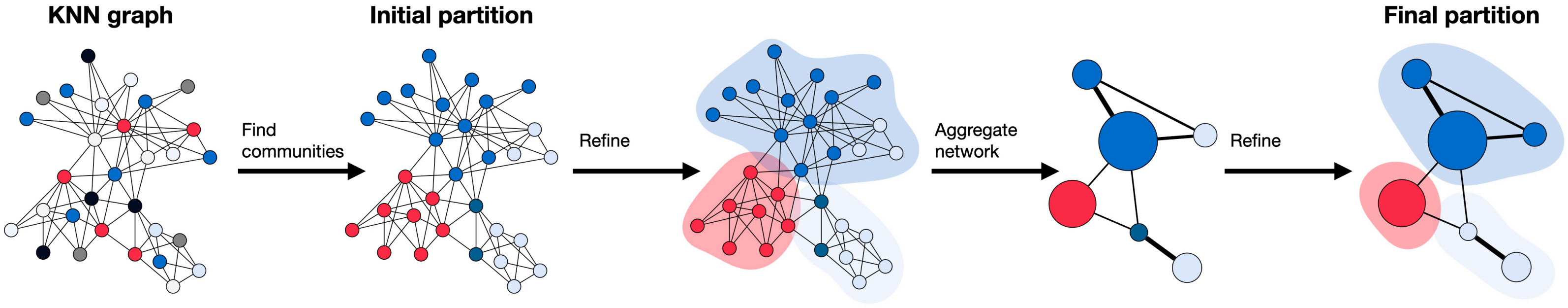
C) Create a meta-graph level:

- one meta-node for each cluster

D) Move objects improving *H*

Van Traag, Scientific Reports, 2019.
Blondel, Journal of Statistical Mechanics, 2008

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*

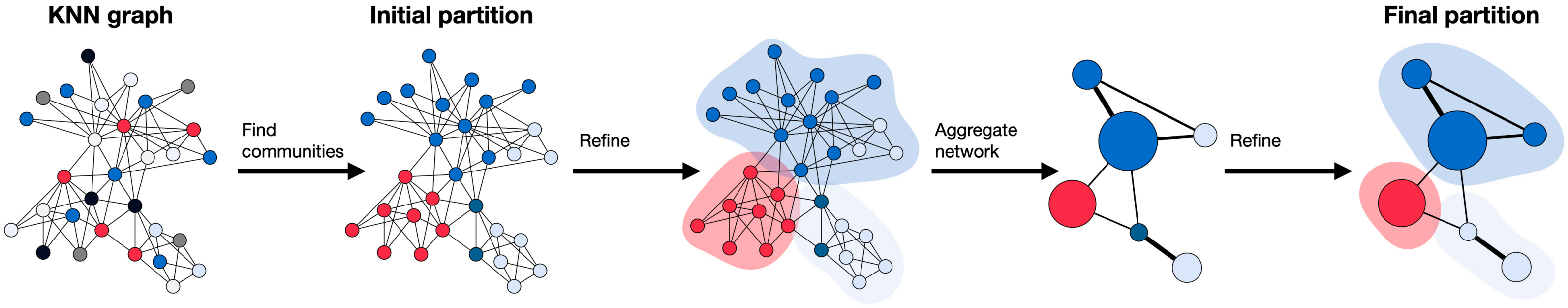
C) Create a meta-graph level:

- one meta-node for each cluster

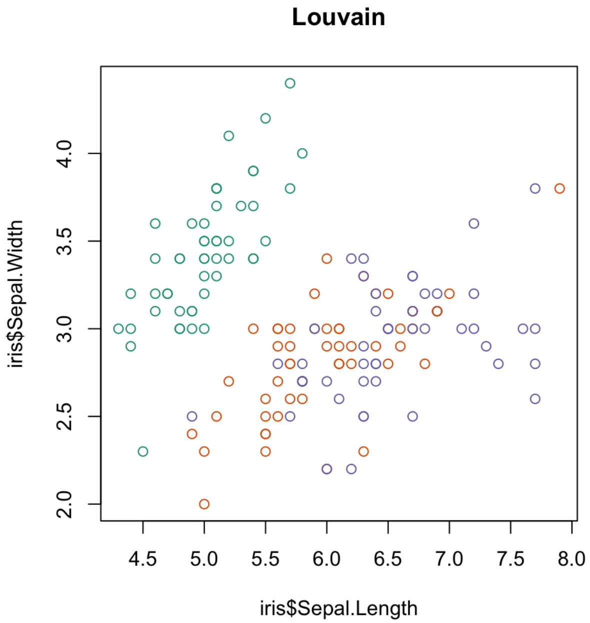
D) Move objects improving *H*

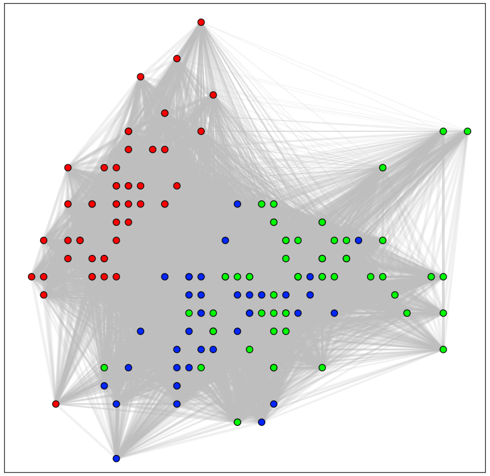
Van Traag, Scientific Reports, 2019.
Blondel, Journal of Statistical Mechanics, 2008

Single cell Clustering / Louvain & Leiden algorithm



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





Van Traag, Scientific Reports, 2019.
Blondel, Journal of Statistical Mechanics, 2008

Institute for Computational Genomics
RWTH AACHEN UNIVERSITY

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

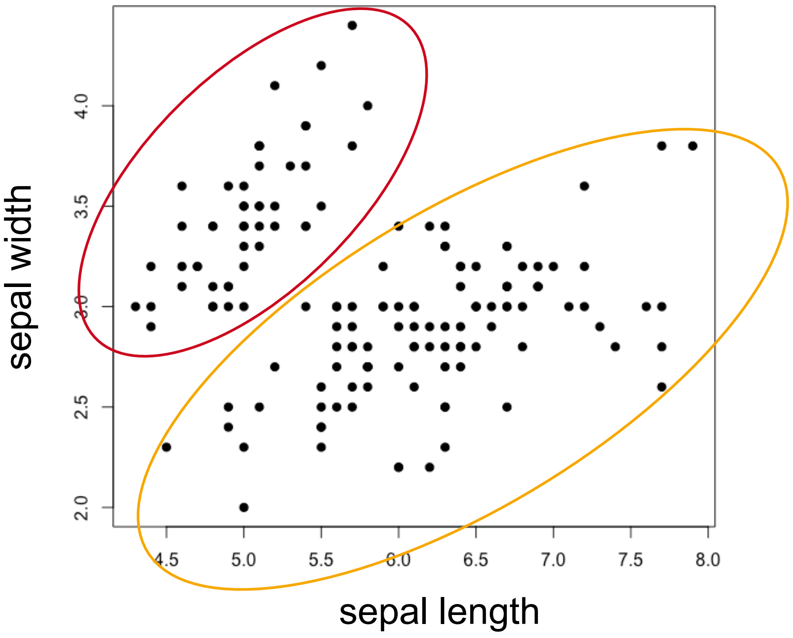
- Hastie, Tibshirani and Friedman, The Elements of Statistical Learning, Chapter 14
- Video lecture: https://www.youtube.com/watch?v=Qa6k7RIwltg

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

	Sepal Length	*Sepal Width*	*Petal Length*	*Petal Width*
Flower 1	*5.1*	*3.5*	*1.4*	*0.2*
Flower 2	*4.9*	*3.0*	*1.4*	*0.2*
Flower 3	*4.7*	*3.2*	*1.3*	*0.2*
Flower 4	*4.6*	*3.1*	*1.5*	*0.2*
…	*…*	*…*	*…*	*…*

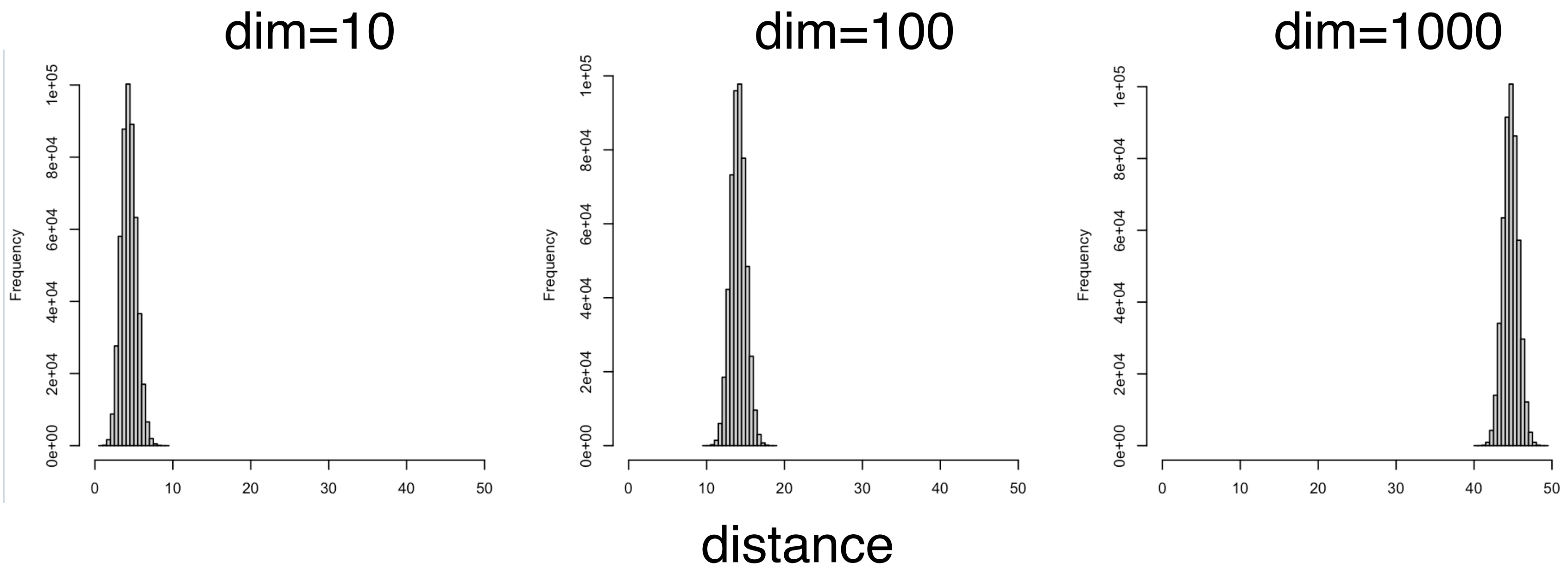


Dimension Reduction

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

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

- Example: distance between points sampled from a normal distribution



dim=10

dim=100

dim=1000

distance

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

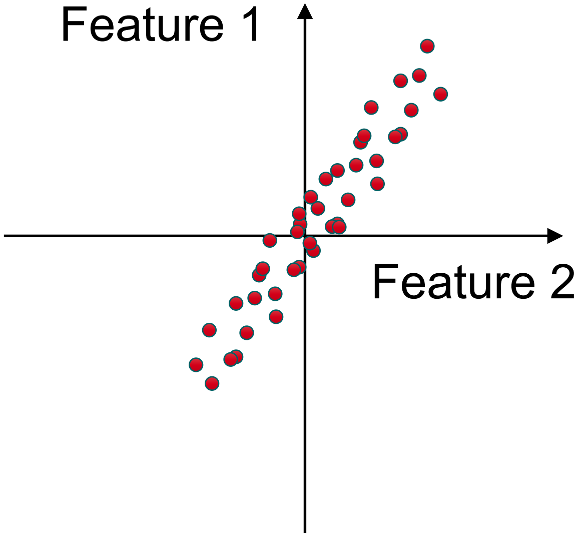
- Dimensionality Reduction by Transformation:
  - linear: principal component analysis (PCA)
  - Non-linear / manifold learning: t-SNE & UMAP (for visualisation)

Principal Component Analysis

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

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

- Can be solved by linear algebra / eigenvector decomposition



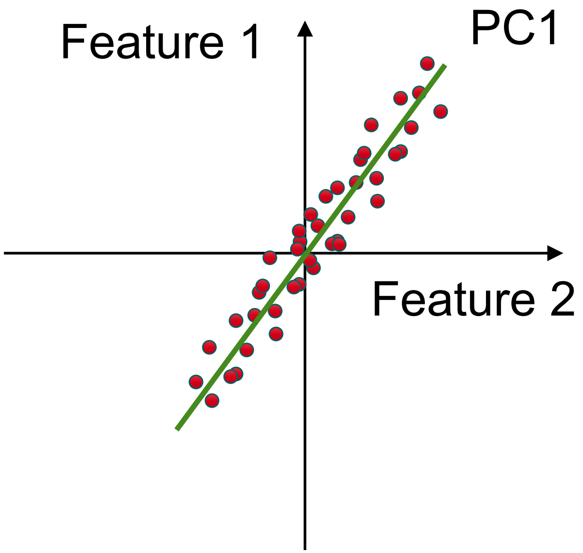
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

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

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max}\,\{\|\mathbf{Xw}\|^2\}$$

- Can be solved by linear algebra / eigenvector decomposition



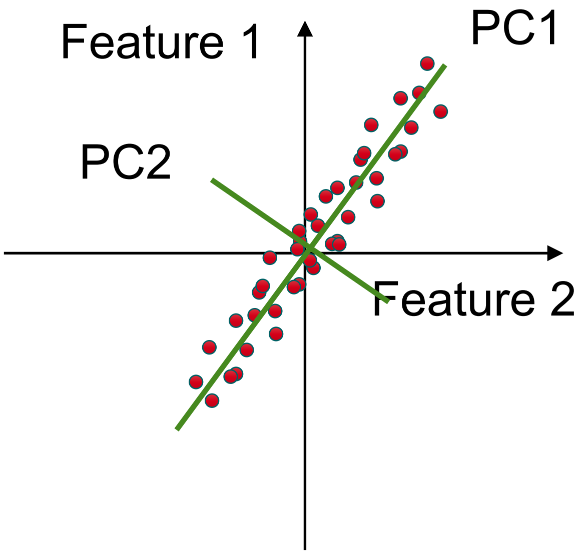
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

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

$$\mathbf{w}_{(1)} = \underset{\Vert \mathbf{w} \Vert = 1}{\arg\,\max} \, \{ \Vert \mathbf{X}\mathbf{w} \Vert^2 \}$$

- Can be solved by linear algebra / eigenvector decomposition



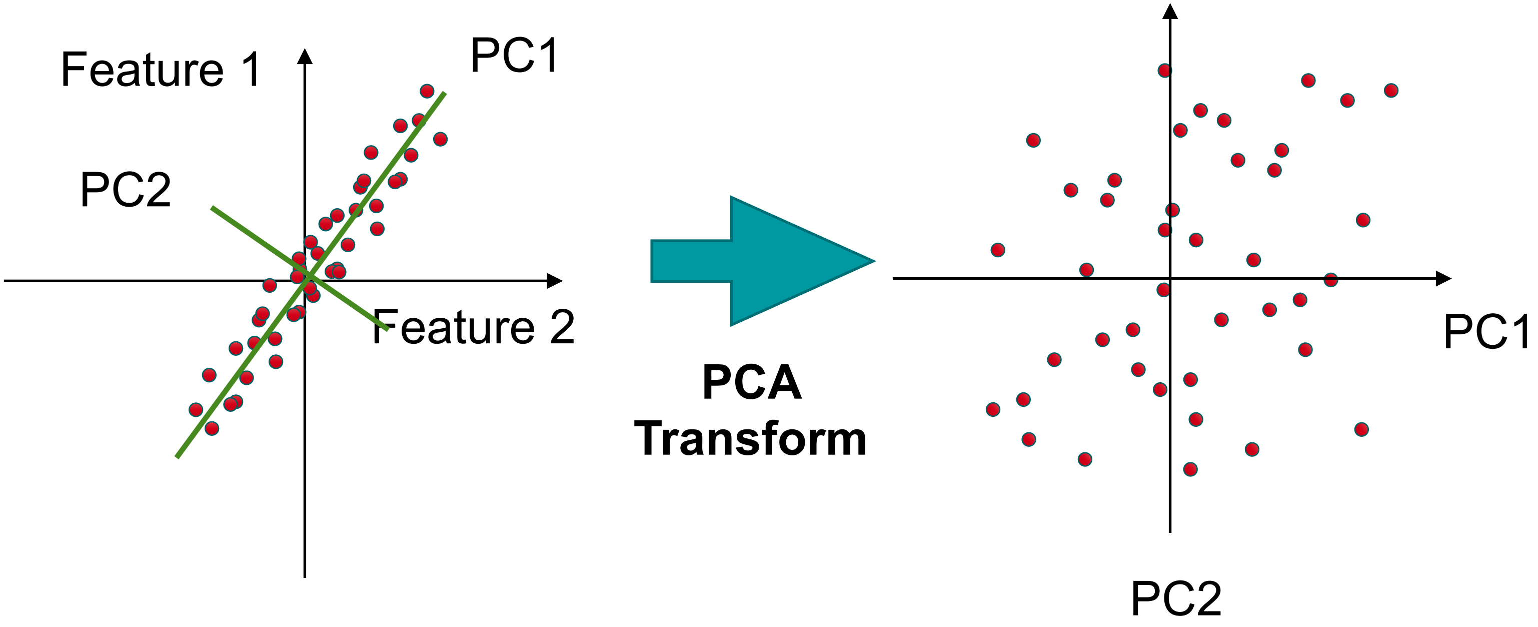
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

Principal Component Analysis

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

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

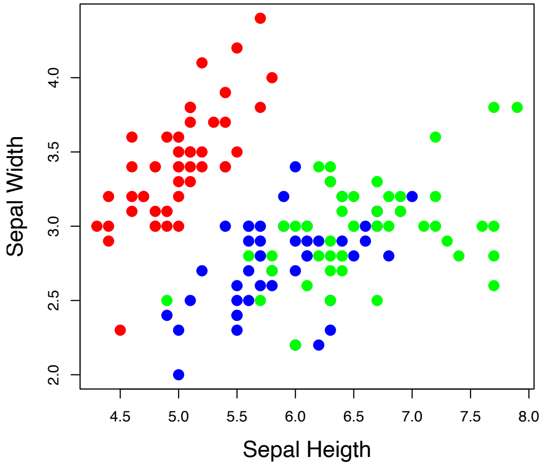
- Can be solved by linear algebra / eigenvector decomposition

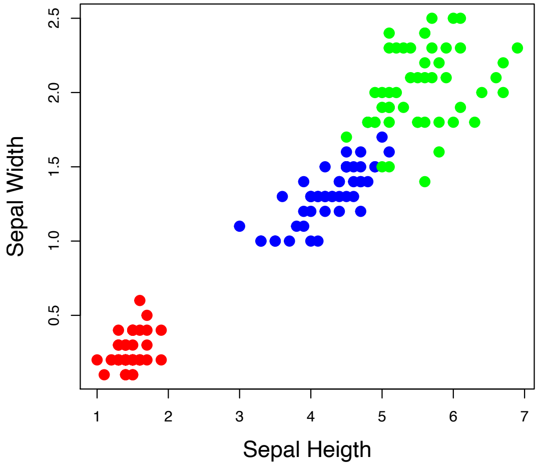


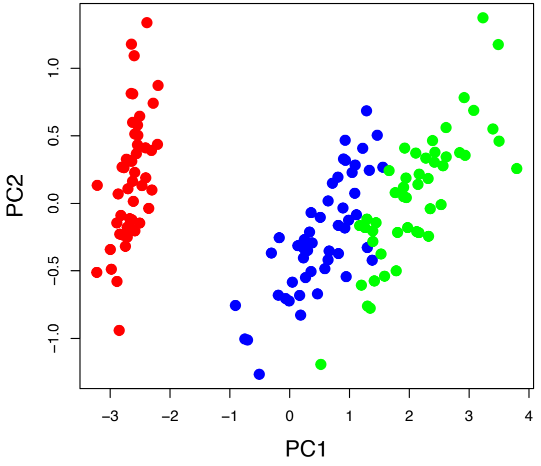
Recommended reading:
Ringner M., *Nature Biotechnology* 26, 303 - 304 (2008)

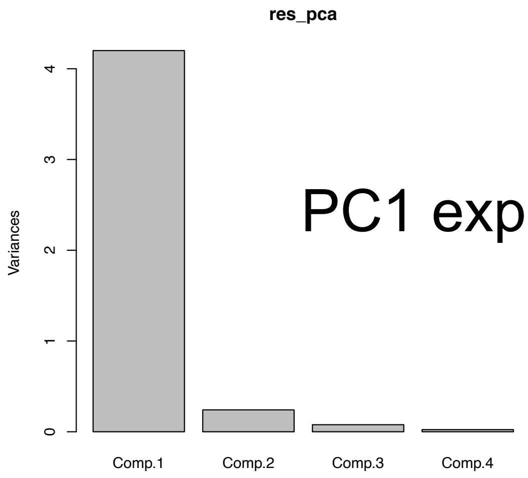
PCA - Iris

- Original iris data had 4 variables







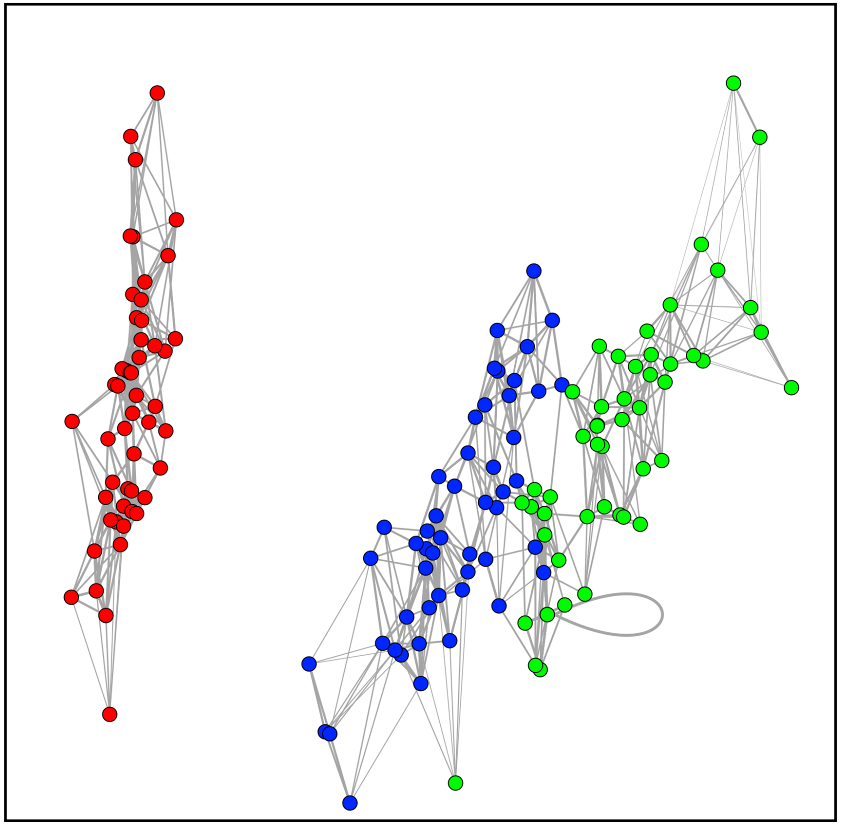


PC1 explains most of variance

Clustering on PCA space

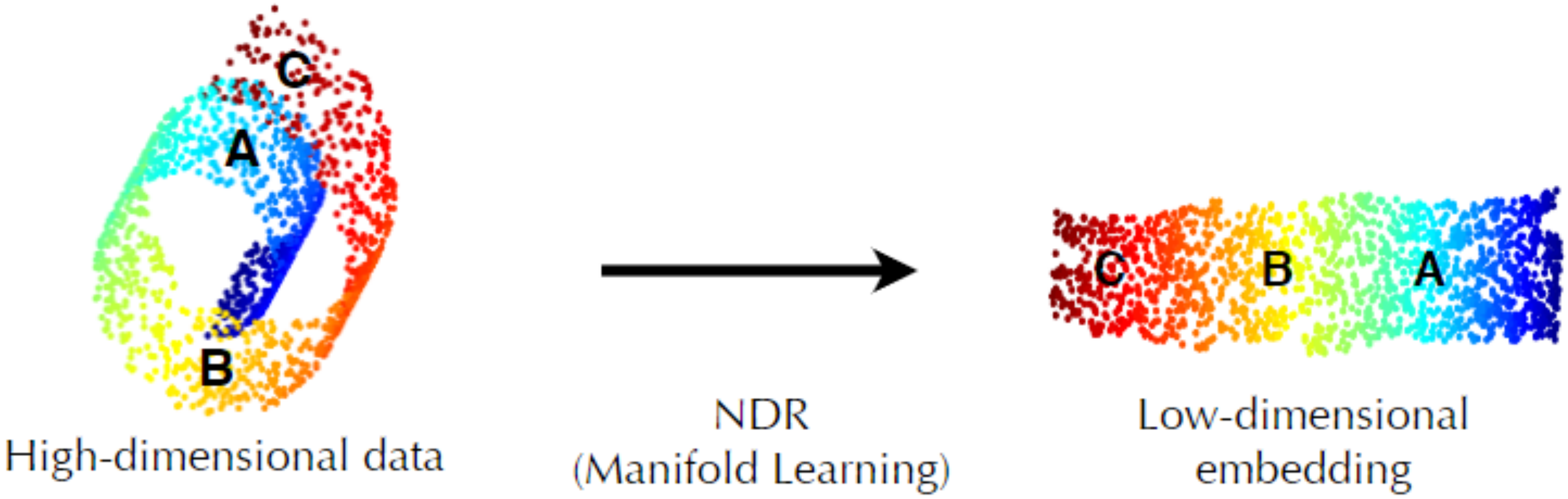
- For single cell data it is usually cluster in PCA space
  - This is crucial for high-dimensional data !

KNN graph of IRIS in PCA space



Non-linear / Manifold methods

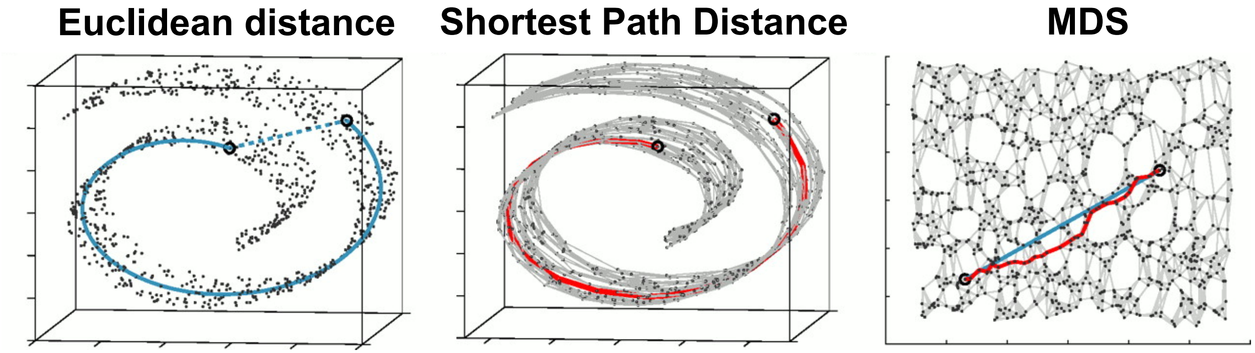
- Data might be distributed at particular regions of a high dimensional space



Adapted from Tenembaum, et al. 2000

Non-linear /Isomap

- Explore topological distance on nearest neighbour graph



Euclidean distance **Shortest Path Distance** **MDS**

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

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

Adapted from Tenembaum, et al. 2000

Non-linear methods

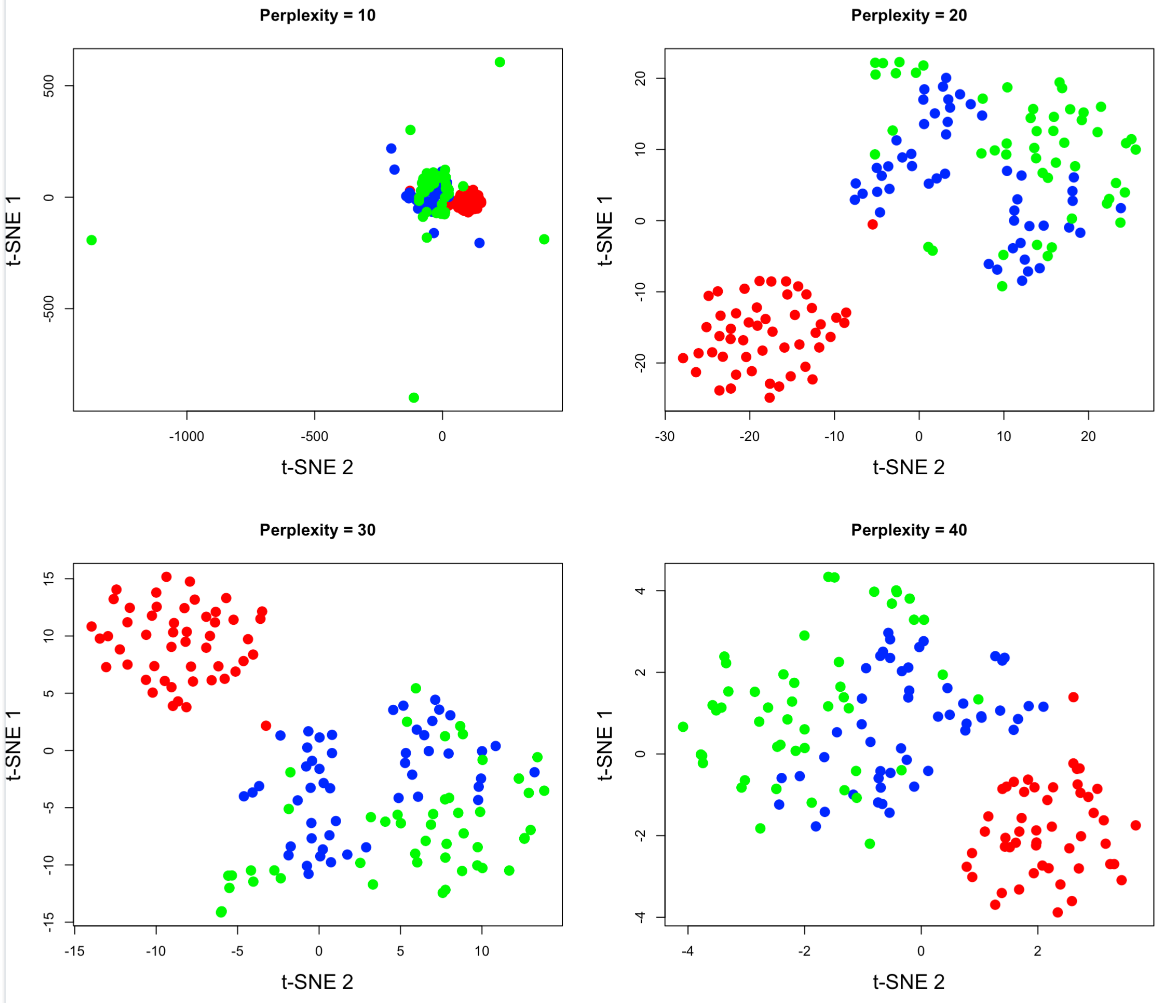
- Variants of Isomap (t-SNE or UMAP) are currently used
- t-SNE - for a given kernel (similarity) ***D*** learn a ***N*** dimensional map ***Y***

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

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 ~ 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***

$$KL(D|Q) = \sum d_{ij} log(\frac{d_{ij}}{q_{ij}})$$

where

$$q_{ij} = \frac{|y_i - y_j|^2}{\sum_k \sum_l |y_k - y_l|^2}$$

KL - Kullback–Leibler divergence

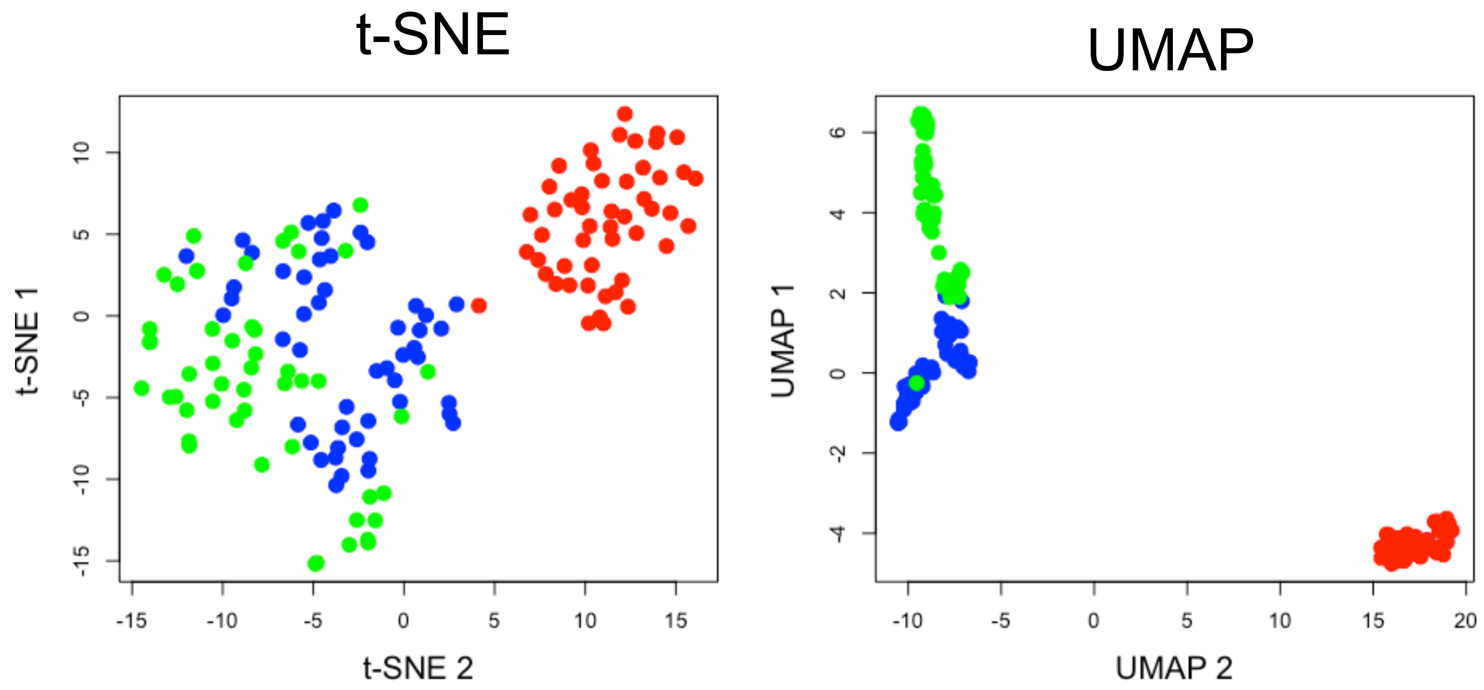
- UMAP - dimension reduction based on Fuzzy Simplicial Sets

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

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

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

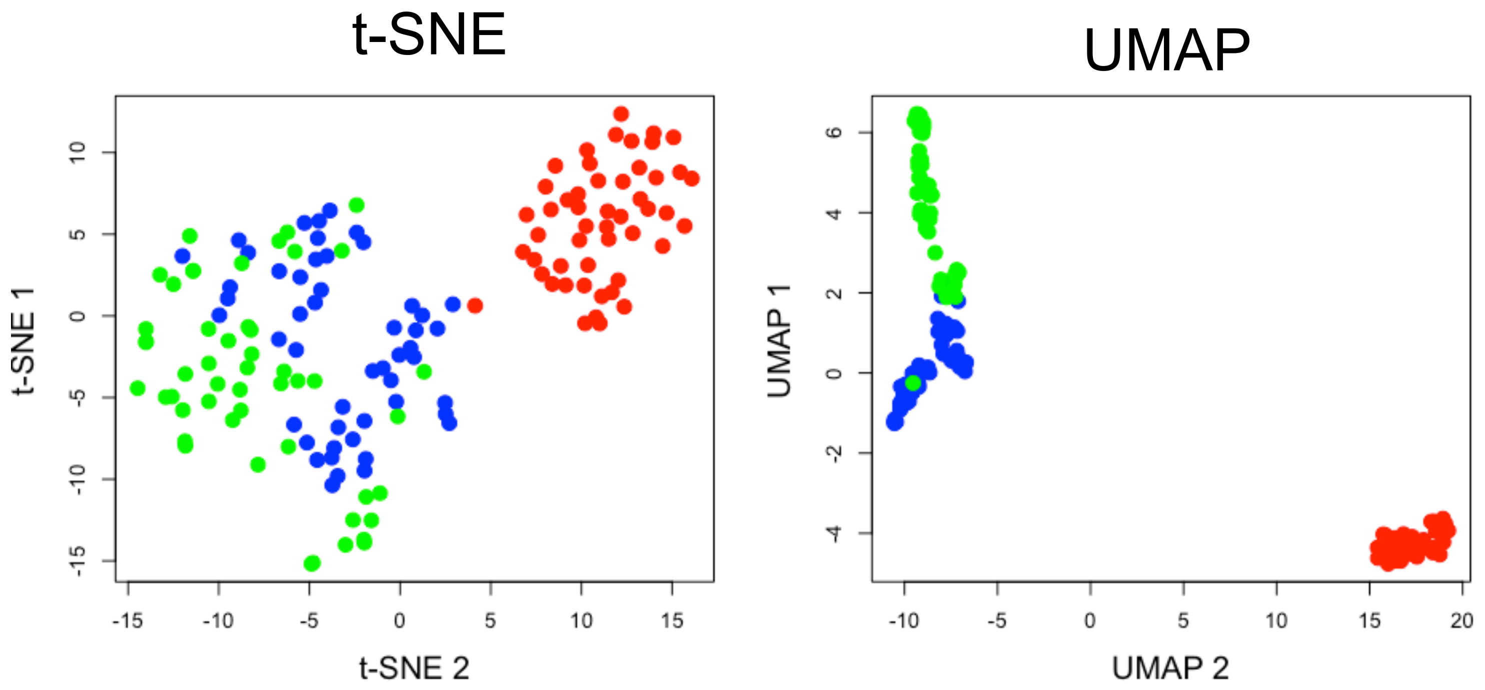
Manifold learning and IRIS



- Nice 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

Manifold learning and IRIS



t-SNE

UMAP

- Nice 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

Calendar

17.04.2023 – Introduction to Bioinformatics and Single Cell Sequencing Analysis

24.05.2023 – Single Cell Sequencing Analysis (cont.) & Practice

8.05.2023 – Introduction to HPC clusters and GPU / Project Proposal

15.05.2023 – 3.7.2023 – Project development

10.07.2023 – Project Presentation

Communication/discord channel: https://discord.gg/hmGxznNpZH .

Thank you!

Cluster Validation

- How to evaluate clustering results? Which is the best method? How many clusters?
- Internal/relative validation:
  - Measure of cluster coherence:
    - Distance within a cluster -> small (compactness)
    - Distance between clusters -> high (separation)
  - Stability measures:
    - Cluster data in part of the data and compare results
- External validation:
  - Compare clusters with class labels (iris data)
    - Not possible in real word problems!

Silhouette - Internal Index

The silhouette for a given object *i* is defined as:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

where

a(i) –mean distance of *i* to objects on same cluster **(compactness)**

d(i,k) – mean distance of *i* to objects of cluster *k* (not own)

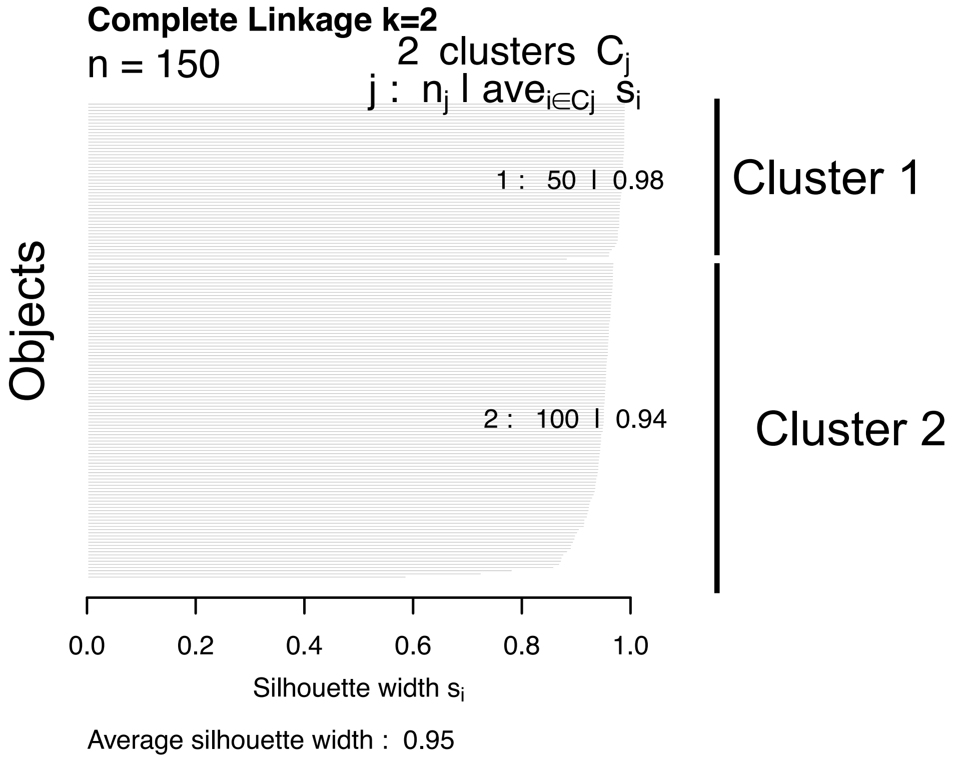
b(i) = *min*$_k$ *(d(i,k))* **(separation)**

Average of *s(i)* -> quality of all results or clusters

Value of 1 indicate perfect solutions!

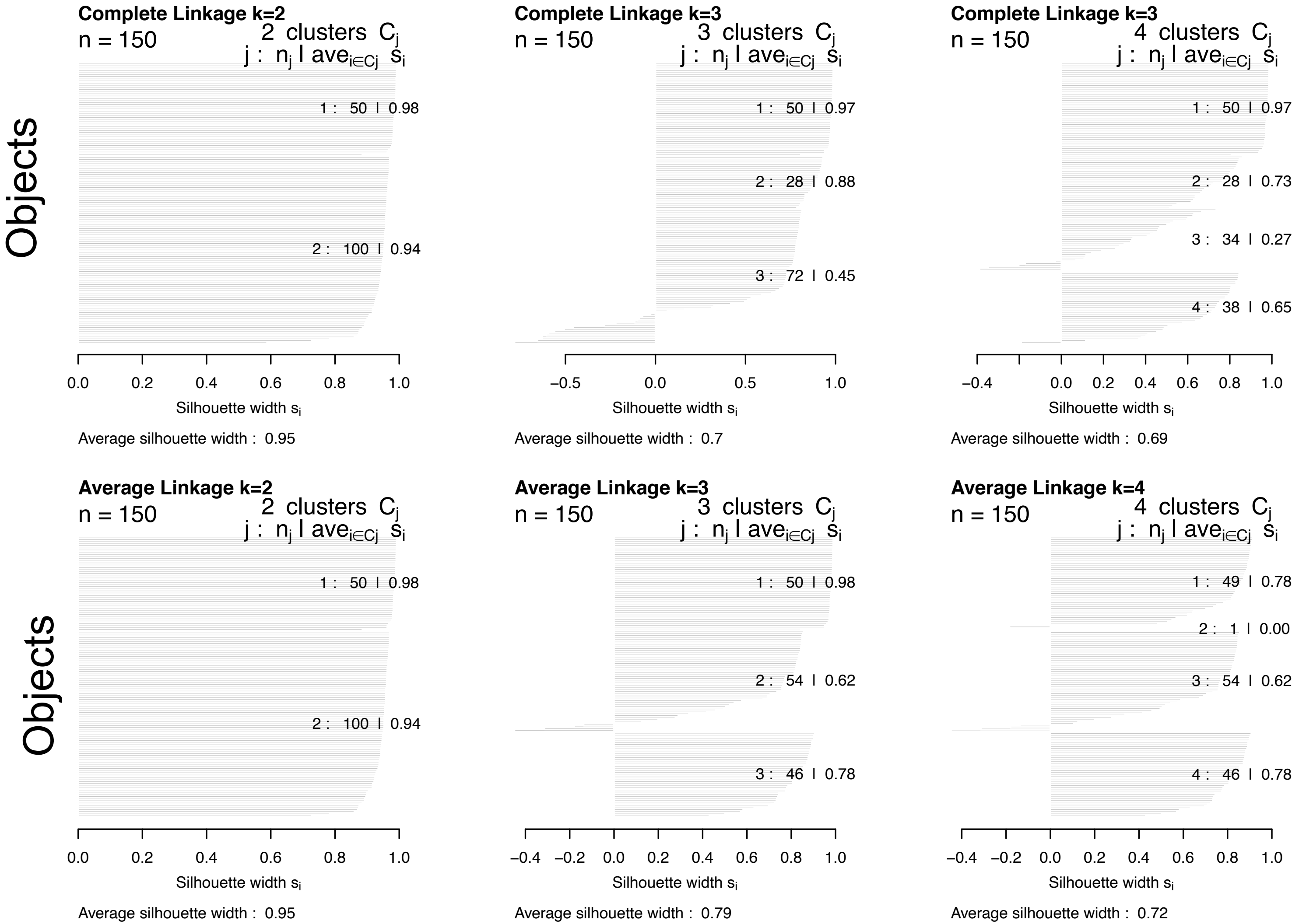
Silhouette - Internal Index / Iris

- silhouette values for hierarchical clustering with Pearson



Silhouette - Internal Index / Iris

- silhouette values for hierarchical clustering with Pearson



Complete Linkage k=2

Complete Linkage k=3

Complete Linkage k=3

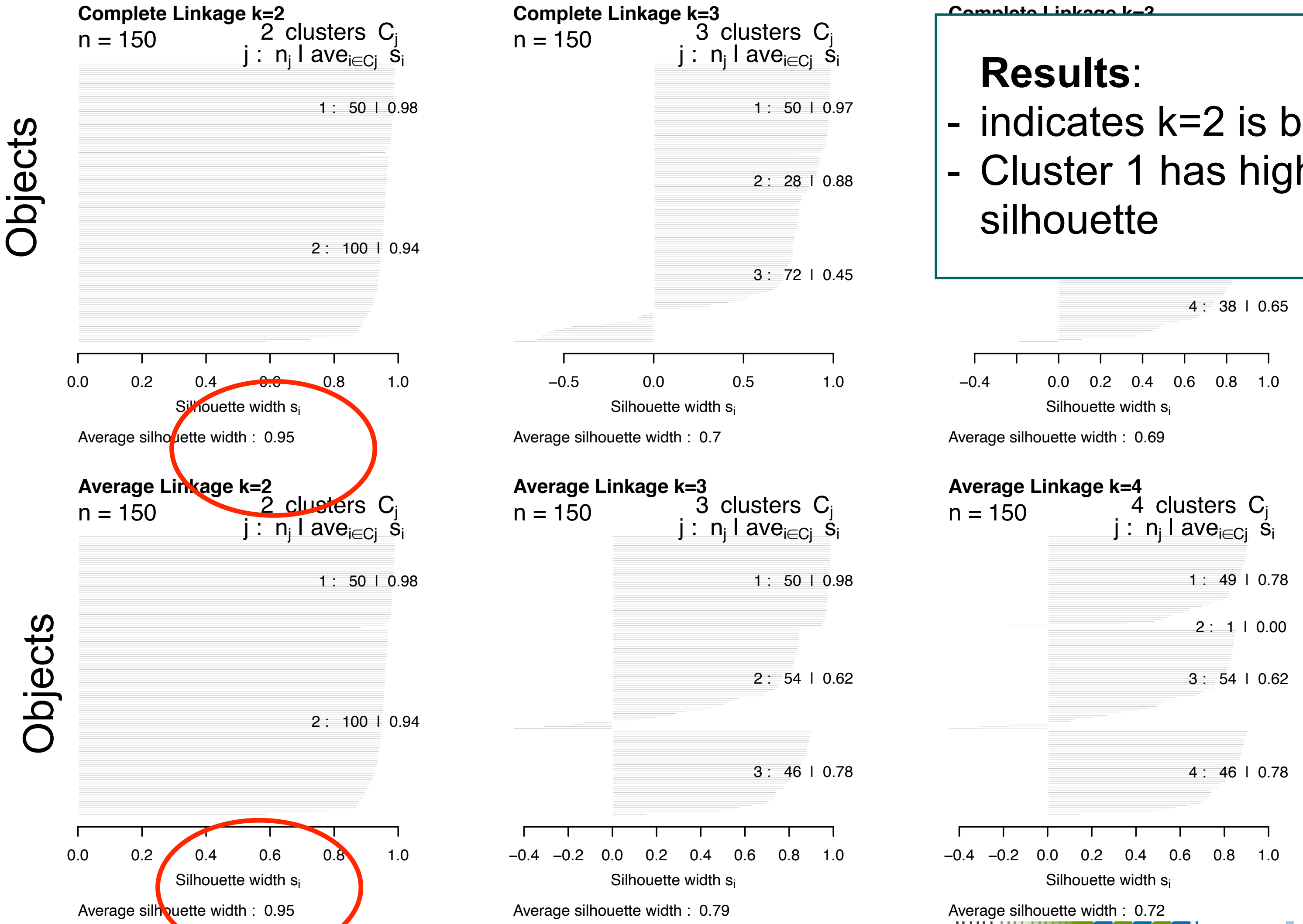
Average Linkage k=2

Average Linkage k=3

Average Linkage k=4

Silhouette - Internal Index / Iris

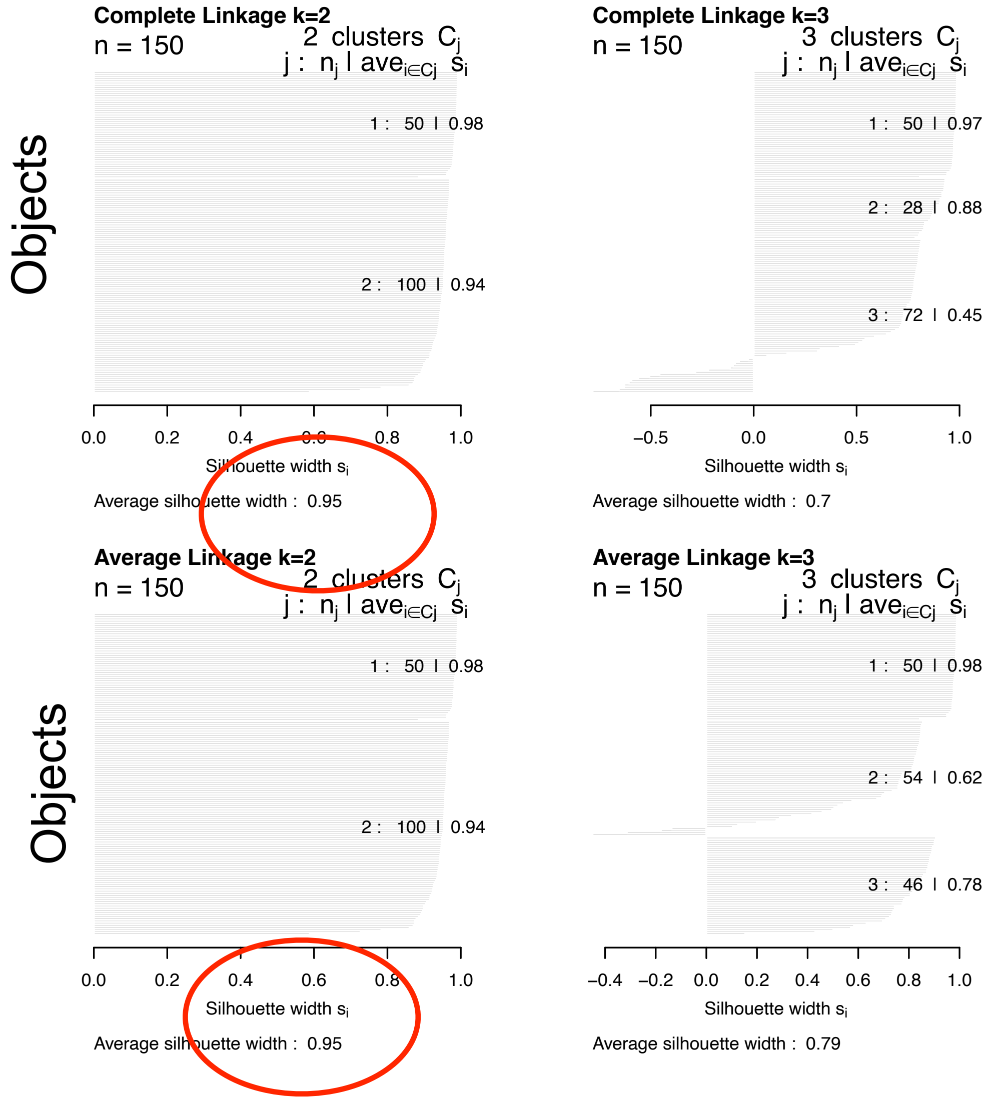
- silhouette values for hierarchical clustering with Pearson



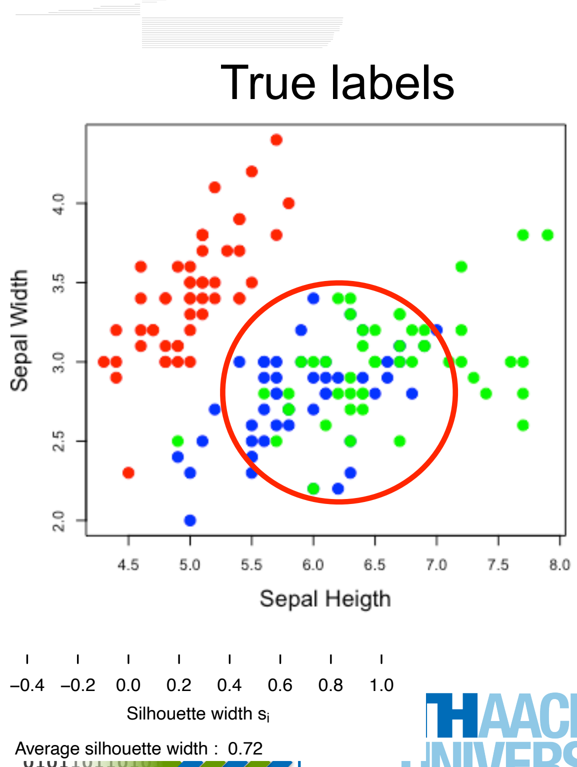
Results:
- indicates k=2 is best
- Cluster 1 has high silhouette

Silhouette - Internal Index / Iris

- silhouette values for hierarchical clustering with Pearson



Results:
- indicates k=2 is best
- Cluster 1 has high silhouette



Gap statistic - Internal Index

For a given solution with *K clusters*

$$W_K = \sum_{k=1}^{K} \sum_{y_i=k} \sum_{y_j=k} ||x_i - x_j||^2$$

W$_K$ - measures cluster compactness
W$_K$ - tends to 0 for increasing K

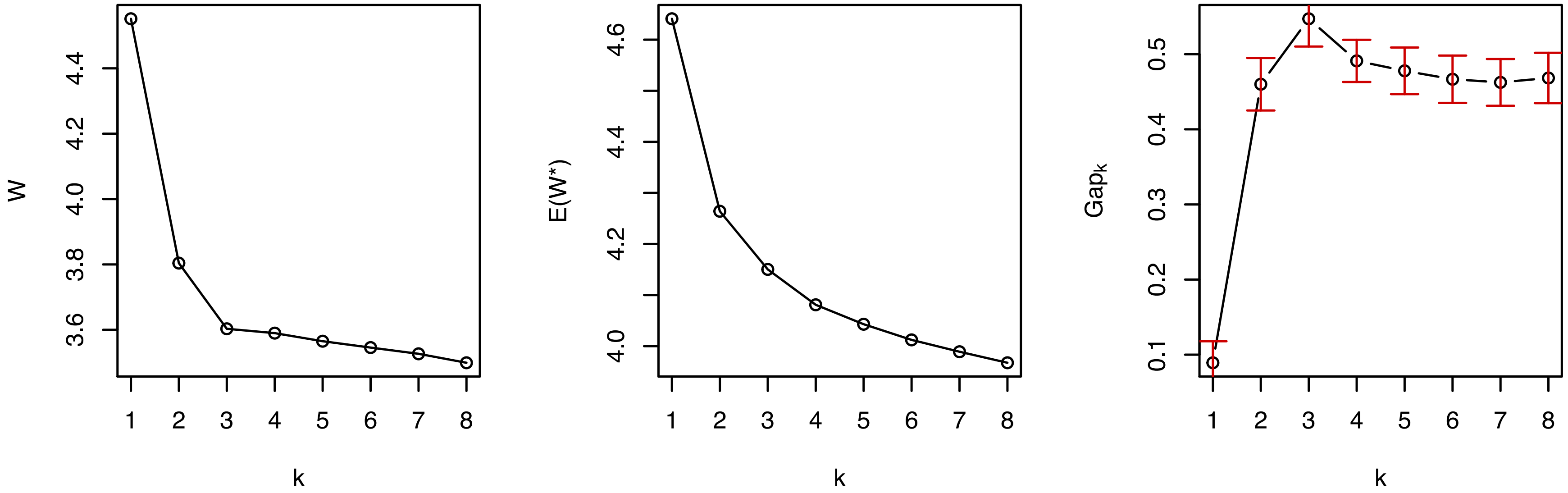
The Gap Statistic consider clustering of random data *W**

$$GAP(k) = E_r[logW_K^*] - logW_K$$

where *W** estimated from clustering random points at the same data space of *X*

Gap statistic - Iris

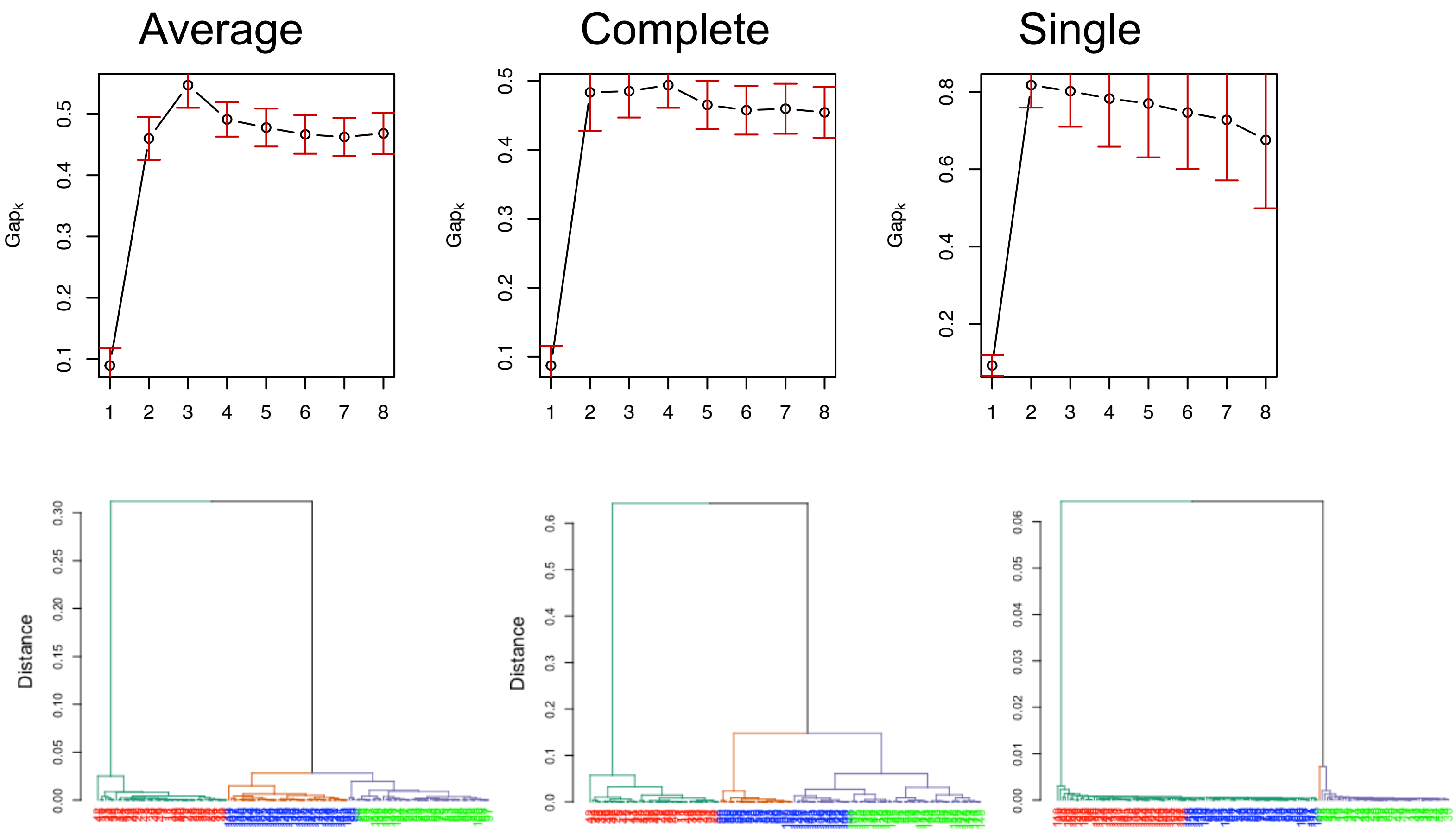
- GAP statistics for Iris / Average Linkage with Pearson



3 clusters has highest Gap !!!

Gap statistic - Iris

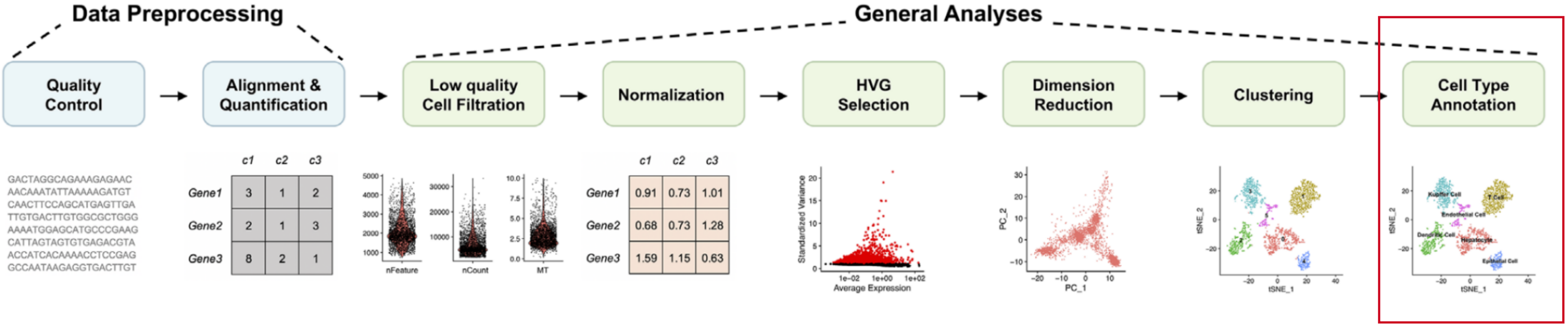
- GAP statistics for distinct linkage methods



Resume / Validation

- Help detection of number of clusters / real clusters
  - Do not work perfectly!
- GAP statistics is widely used
  - Requires *r* data randomisations
    - high computational costs
    - random datasets uniformly distributed (unreal assumption)
- Expert interpretation is important!

Basics Bioinformatics - single cell RNA-seq



Data Preprocessing

General Analyses

Quality Control → Alignment & Quantification → Low quality Cell Filtration → Normalization → HVG Selection → Dimension Reduction → Clustering → Cell Type Annotation

GACTAGGCAGAAAGAGAAC
AACAAATATTAAAAAGATGT
CAACTTCCAGCATGAGTTGA
TTGTGACTTGTGGCGCTGGG
AAAATGGAGCATGCCCGAAG
CATTAGTAGTGTGAGACGTA
ACCATCACAAAACCTCCGAG
GCCAATAAGAGGTGACTTGT

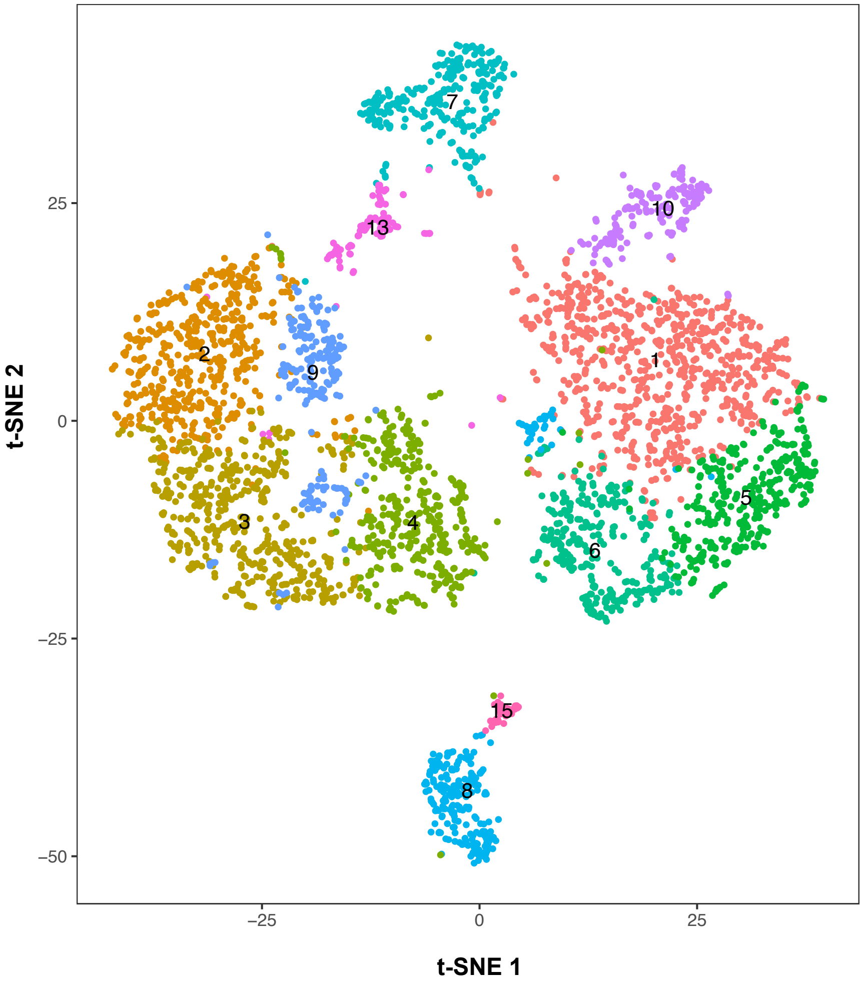
	c1	c2	c3
Gene1	3	1	2
Gene2	2	1	3
Gene3	8	2	1

	c1	c2	c3
Gene1	0.91	0.73	1.01
Gene2	0.68	0.73	1.28
Gene3	1.59	1.15	0.63

D Jovic, X Liang, H Zeng, L Lin, F Xu, Y Luo, 2022

Basics Bioinformatics - Clustering

Gut Immune Cells - 12 groups

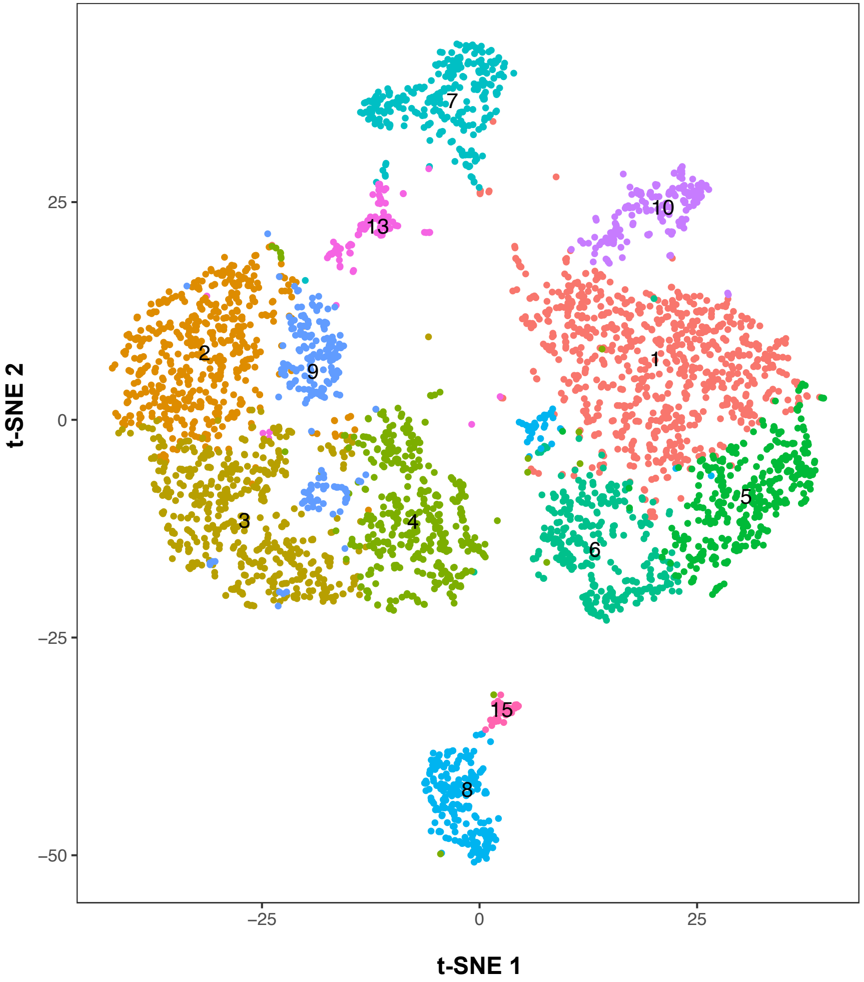


Clustering - identify cells with similar expression patterns
- based on PCA (20 dimension)

How to identify cell types?

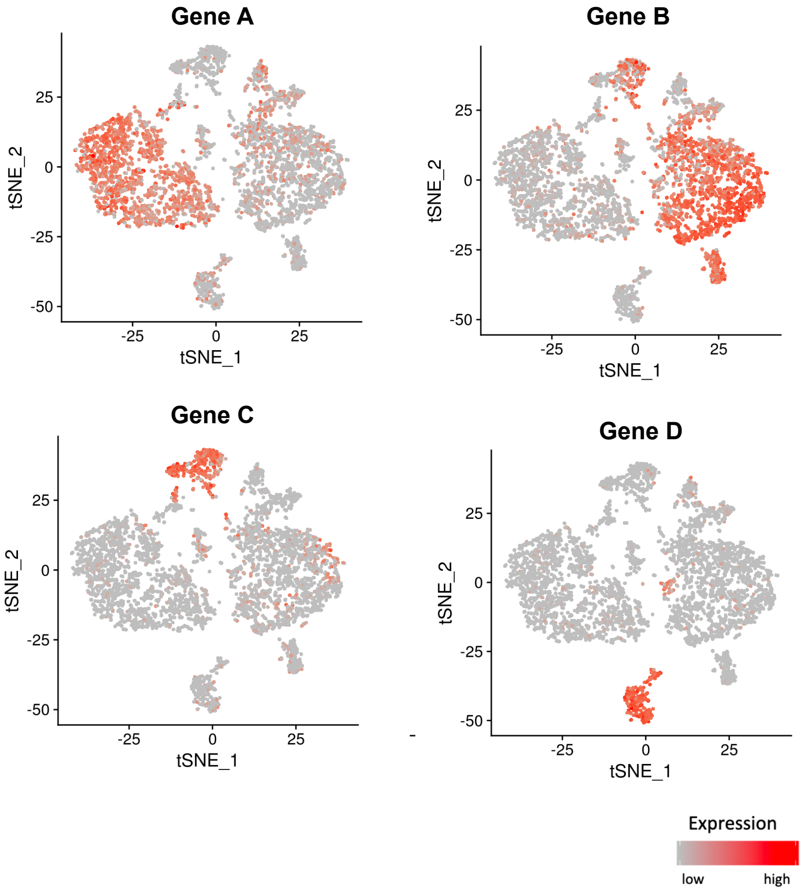
Cell Identity with an Expert

Gut Immune Cells - 12 groups



Check expression of:

1. known genes

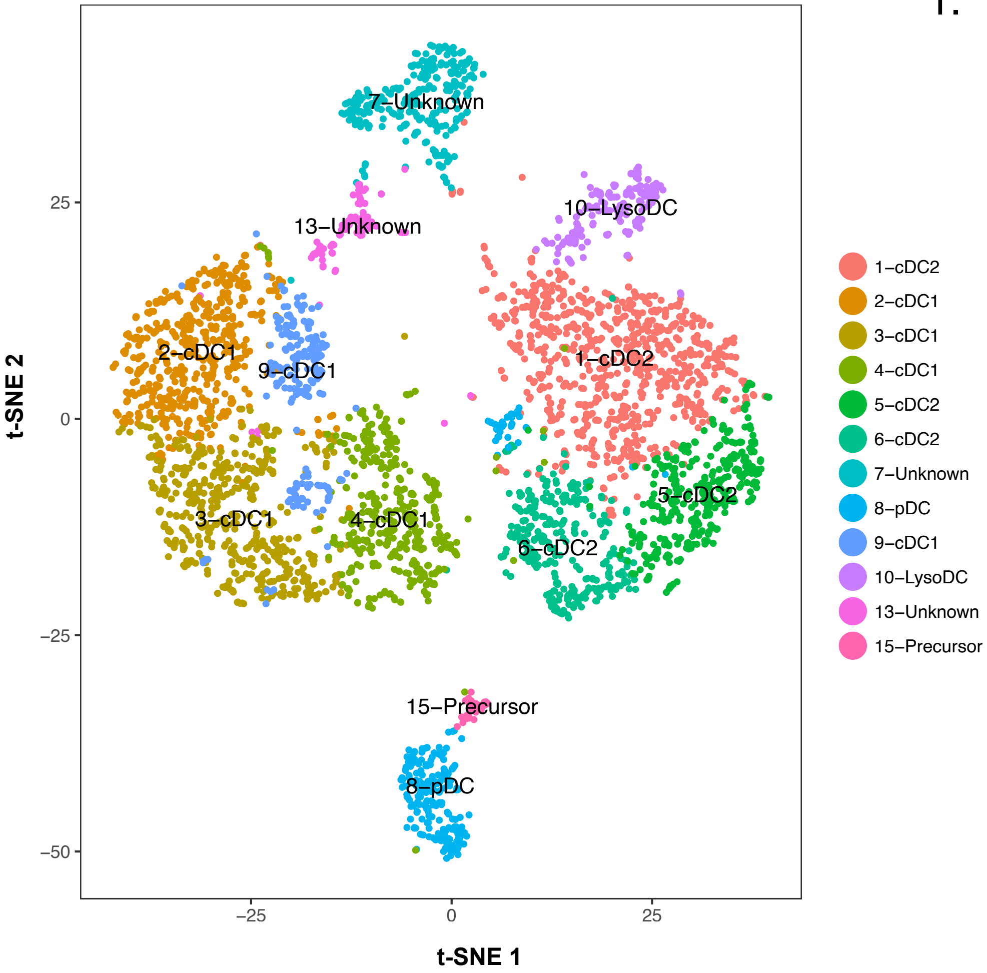


Institute for Computational Genomics

RWTH AACHEN UNIVERSITY

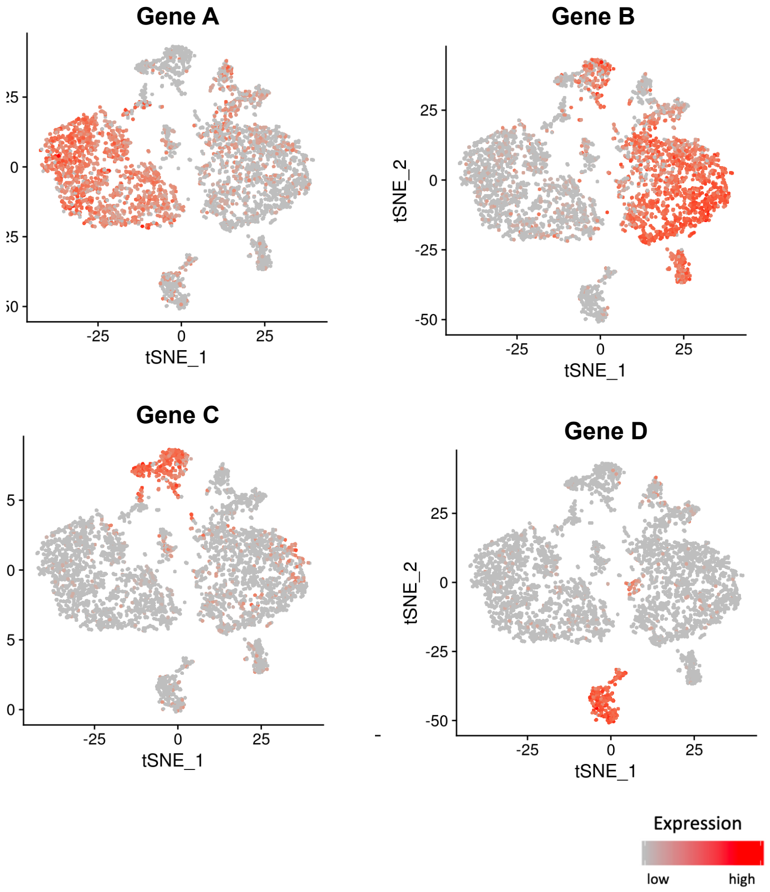
Cell Identity with an Expert

Gut Immune Cells - 12 groups



Check expression of:

1. known genes

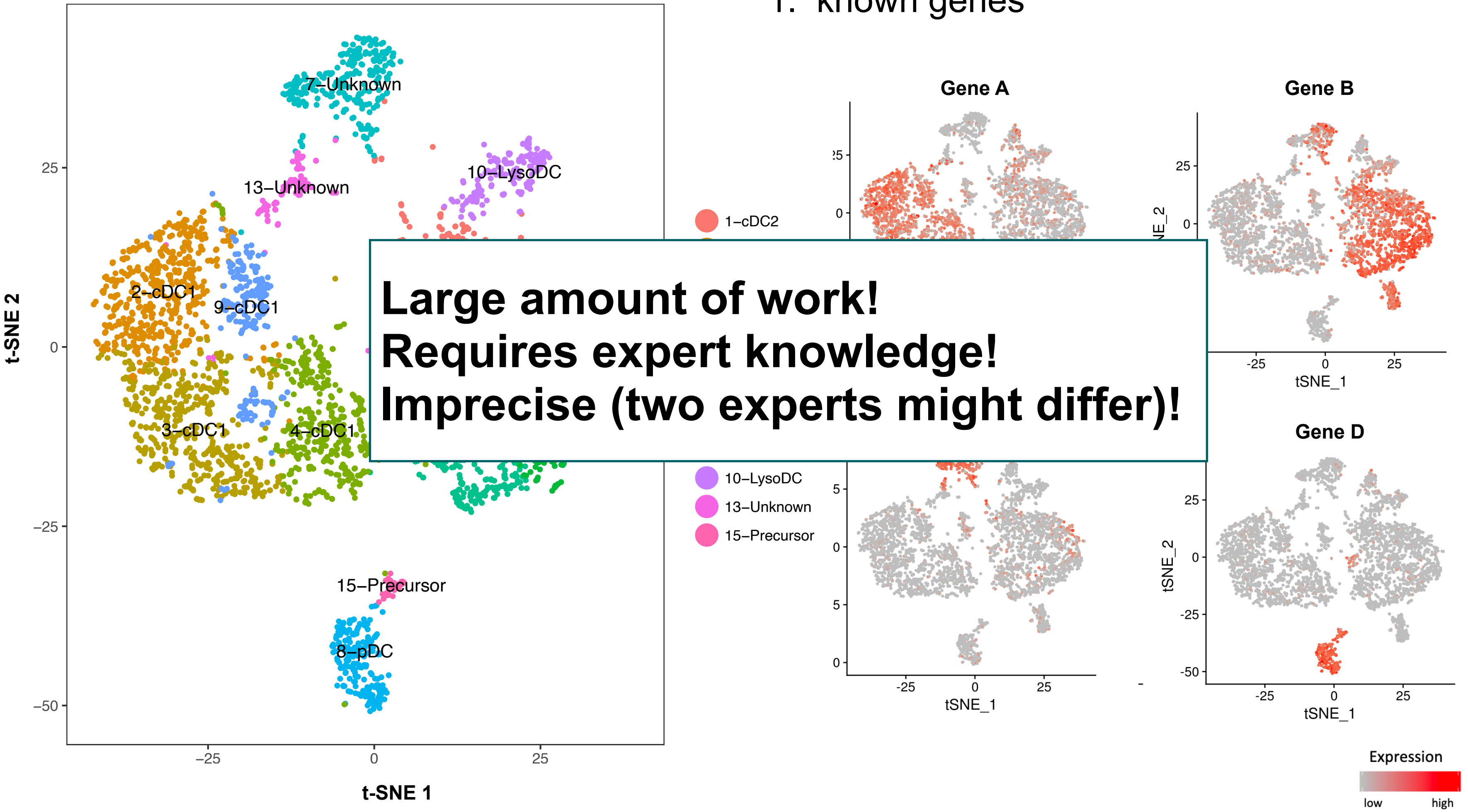


Cell Identity with an Expert

Gut Immune Cells - 12 groups

Check expression of:

1. known genes



Large amount of work!
Requires expert knowledge!
Imprecise (two experts might differ)!

Resume / Single cell clustering

- Finding groups of single cells require complex pipeline:
  - Cell filtering
  - Normalisation
  - Artefact removal
  - **Dimension reduction**
  - **Integration**
  - **Clustering**
  - **Cell annotation / visualisation**
- Open points:
  - How to deal with large data sets (millions of cells)?
  - How to detect cells of rare populations?

Calendar

17.04.2023 – Introduction to Bioinformatics and Single Cell Sequencing Analysis

24.05.2023 – Single Cell Sequencing / Theory & Practice

8.05.2023 – Introduction to HPC clusters and GPU / Project Proposal

15.05.2023 – 3.7.2023 – Project development

10.07.2023 – Project Presentation

Communication/discord channel: https://discord.gg/hmGxznNpZH .

Thank you!