CEITEC

Central European Institute of Technology
BRNO | CZECH REPUBLIC

Introduction to Bioinformatics
(LF:DSIB01)

Week 8 : Clustering and dimension reduction

Clustering (cluster analysis)

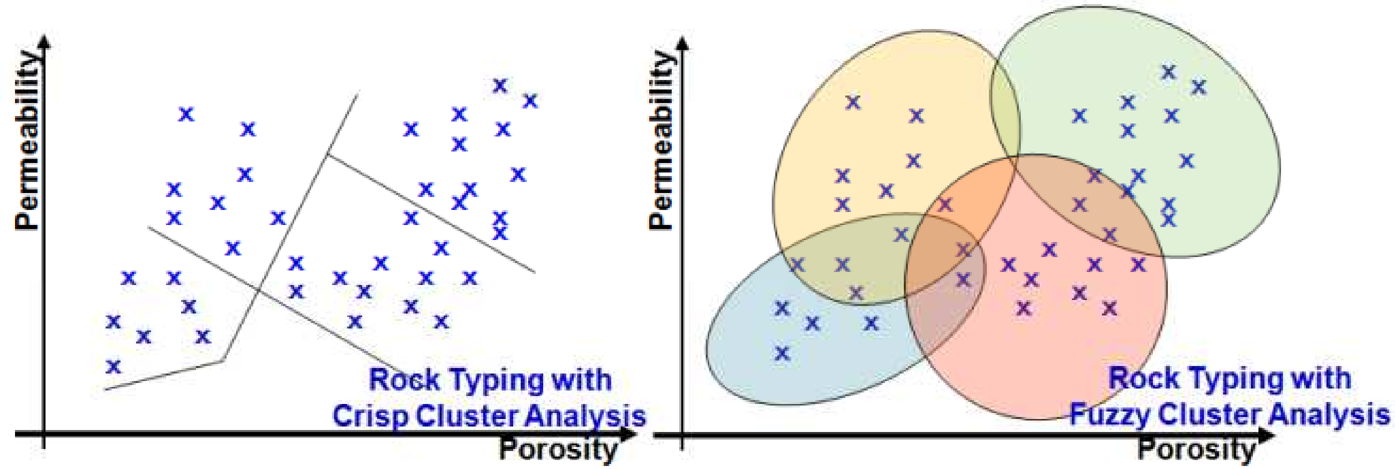
- Very broad definition – group of data objects
- Unsupervised classification
- Data mining
- What data you need for clustering?
  - Data points with features where you can measure distance
    - Some methods need Triangular inequality
  - Distance matrix
  - Important for sequence comparison!

Clustering types

- *Centroid models*
  - *K-mean clustering*
  - Mean-Shift Clustering
- *Distribution models*
  - Expectation–Maximization (EM) Clustering
- *Density models*
  - *DBSPAN*
- *Connectivity models*
  - *Hierarchical clustering*
- *Graph-based models*
  - *Clique, Spanning tree, Markov Cluster Algorithm (MCL)*
- …

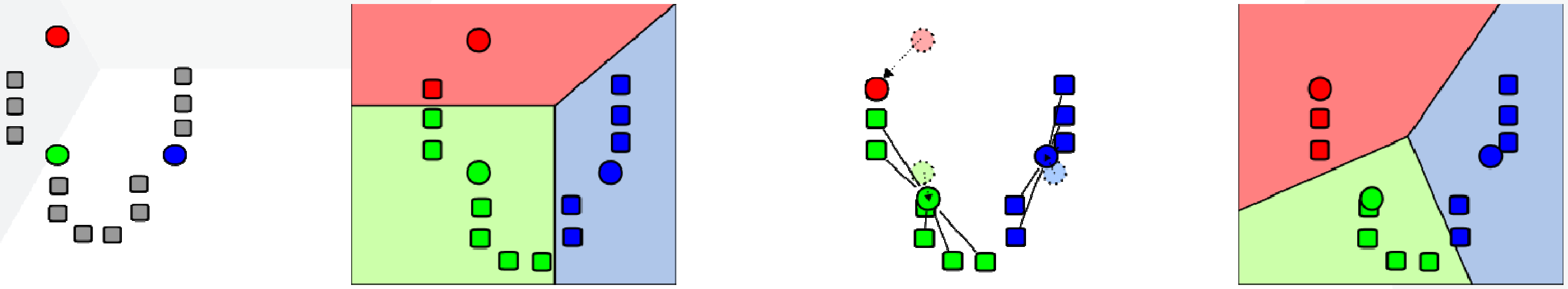
Clustering

- Hard vs. Fuzzy clustering
  - Fuzzy = each object belong to each cluster to some degree (probability)
  - Overlapping clustering =~ discreet Fuzzy clustering
  - More detailed analysis, but hard to interpret – good not as a final step

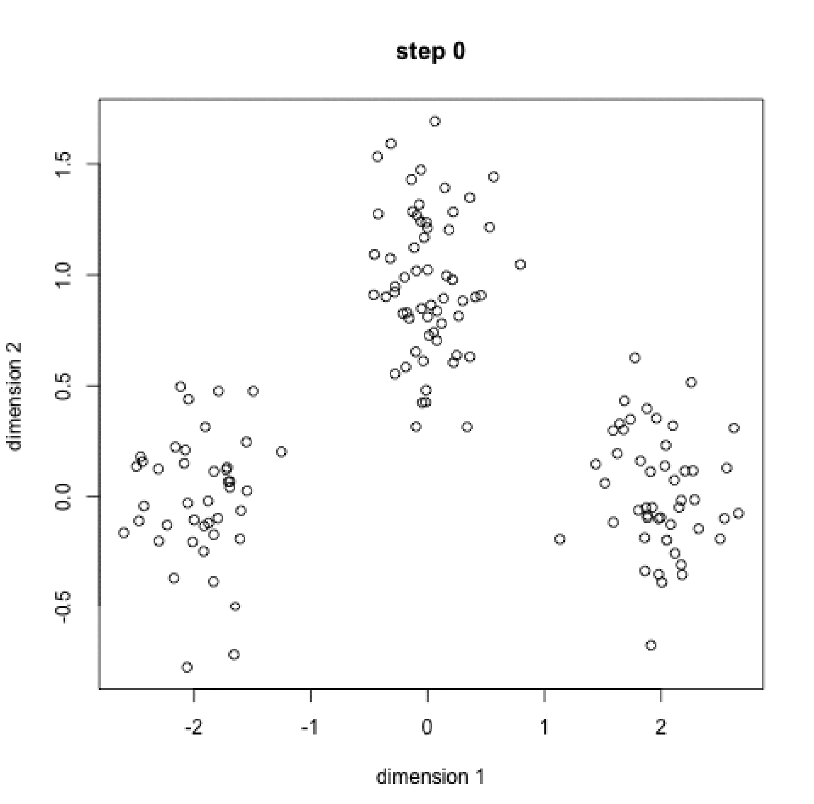


K-mean clustering

- most well-known clustering algorithm
  1. Select a number of clusters to use and randomly initialize centers
  2. *k* clusters are created by associating every observation with the nearest mean
  3. Centroid of each of the *k* clusters becomes the new mean
  4. Repeat steps 2 and 3 until stopping rule



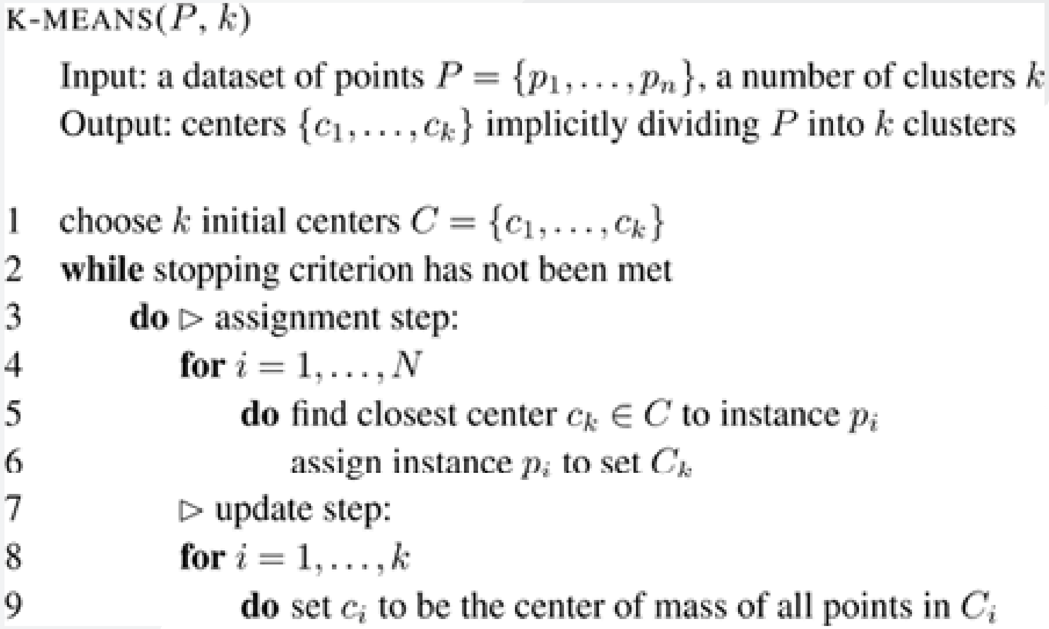
K-mean clustering



K-mean clustering

- Advantage
  - simple to implement
  - very fast
    - linear complexity *O*(*n*)
- Disadvantage
  - Must select number of clases
  - Random start point
  - Identify ‘spherical clusters’

K-mean clustering pseudocode



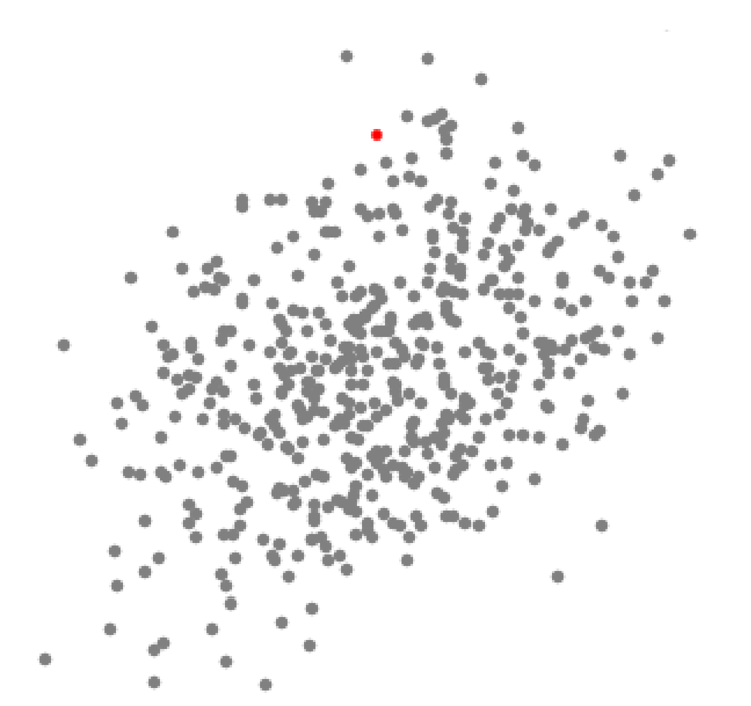
```
K-MEANS($P$, $k$)
    Input: a dataset of points $P = \{p_1, \ldots, p_n\}$, a number of clusters $k$
    Output: centers $\{c_1, \ldots, c_k\}$ implicitly dividing $P$ into $k$ clusters

    choose $k$ initial centers $C = \{c_1, \ldots, c_k\}$
    while stopping criterion has not been met
        do ▷ assignment step:
            for $i = 1, \ldots, N$
                do find closest center $c_k \in C$ to instance $p_i$
                    assign instance $p_i$ to set $C_k$
            ▷ update step:
            for $i = 1, \ldots, k$
                do set $c_i$ to be the center of mass of all points in $C_i$
```

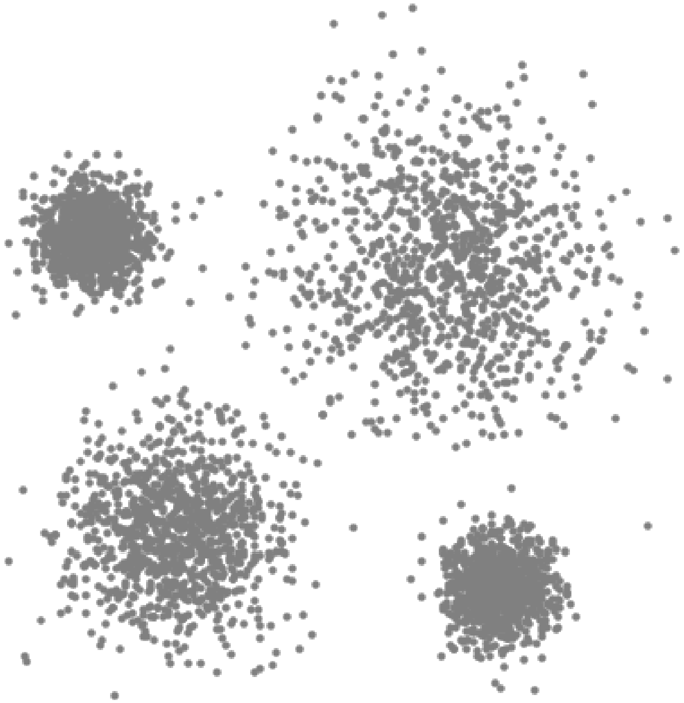
Mean-Shift Clustering

- Steps
  1. circular sliding window centered at a point C (randomly selected)
  2. every iteration, the sliding window is shifted towards regions of higher density by shifting the center point to the mean of the points within the window
  3. until there is no direction at which a shift
  4. process of steps 1 to 3 is done with many sliding windows until all points lie within a window

Mean-Shift Clustering



Mean-Shift Clustering

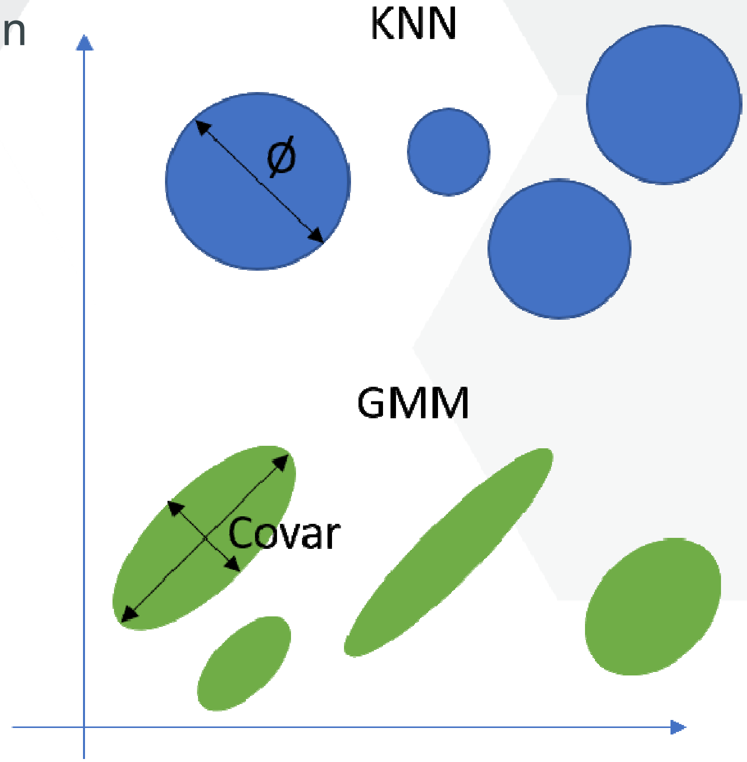


Mean-Shift Clustering

- Advantage
  - Not-necessary to select number of clusters
  - Fast computation
- Disadvantage
  - Must select kernel range r
  - Identify ‘spherical clusters’

Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)

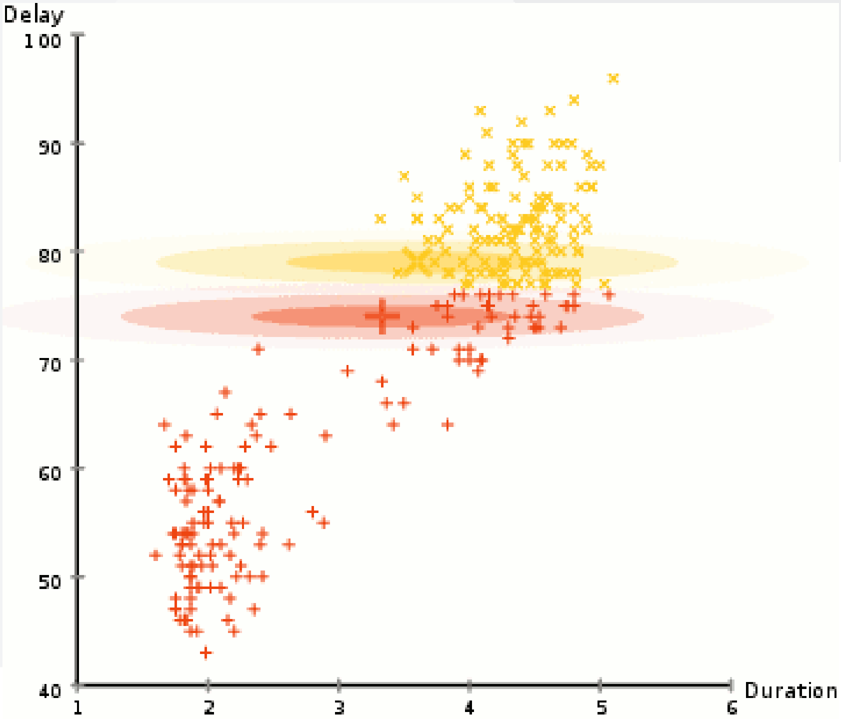
- Similar to K-means clustering
  - each cluster is not defined only by its mean but also variance
  - Gaussian normal curve in each dimension



Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)

- KM
  1. select a number of clusters to use and randomly initialize centers
  2. *k* clusters are created by associating every observation with the nearest mean
  3. centroid of each of the *k* clusters becomes the new mean
  4. repeat steps 2 and 3 until stop rule
- EM with GMM
  1. select a number of clusters to use and randomly initialize Gaussian distribution parameters
  2. Using Gaussian distributions for each cluster, compute the probability that each data point belongs to a particular cluster
  3. compute a new set of parameters of Gaussian distributions to maximize the probabilities of data points within the clusters
     - Weighted sum of individual
  4. repeat steps 2 and 3 until convergence

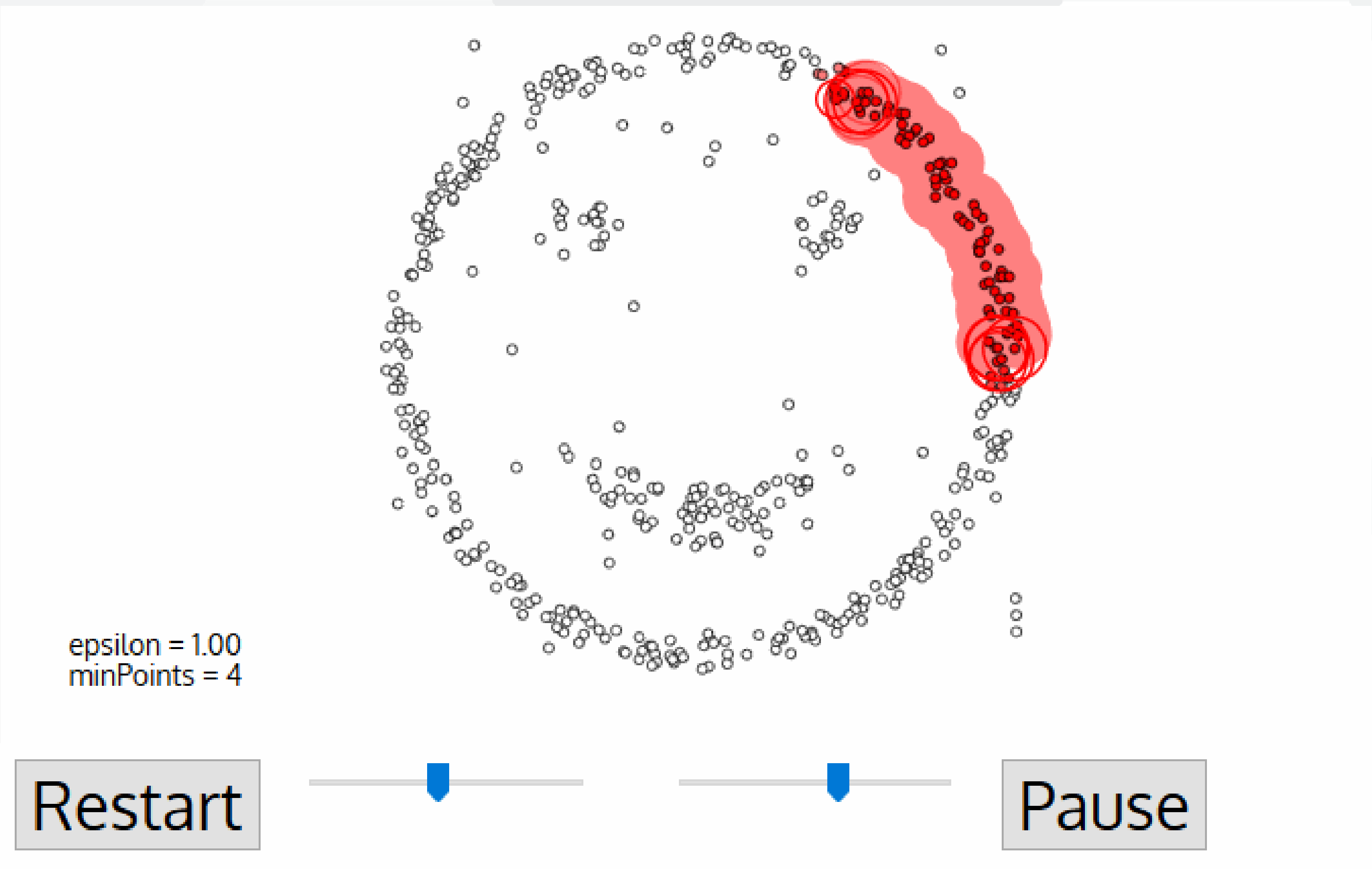
Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)



Density-Based Spatial Clustering of Applications with Noise

1. DBSCAN begins with an arbitrary starting data point that has not been visited. The neighborhood of this point is extracted using a distance epsilon ε (All points which are within the ε distance are neighborhood points).
2. If there are a sufficient number of points (according to minPoints) within this neighborhood then the clustering process starts and the current data point becomes the first point in the new cluster. Otherwise, the point will be labeled as noise (later this noisy point might become the part of the cluster). In both cases that point is marked as "visited".
3. For this first point in the new cluster, the points within its ε distance neighborhood also become part of the same cluster. This procedure of making all points in the ε neighborhood belong to the same cluster is then repeated for all of the new points that have been just added to the cluster group.
4. This process of steps 2 and 3 is repeated until all points in the cluster are determined i.e all points within the ε neighborhood of the cluster have been visited and labeled.
5. Once we're done with the current cluster, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. This process repeats until all points are marked as visited. Since at the end of this all points have been visited, each point will have been marked as either belonging to a cluster or being noise.

Density-Based Spatial Clustering of Applications with Noise



epsilon = 1.00
minPoints = 4

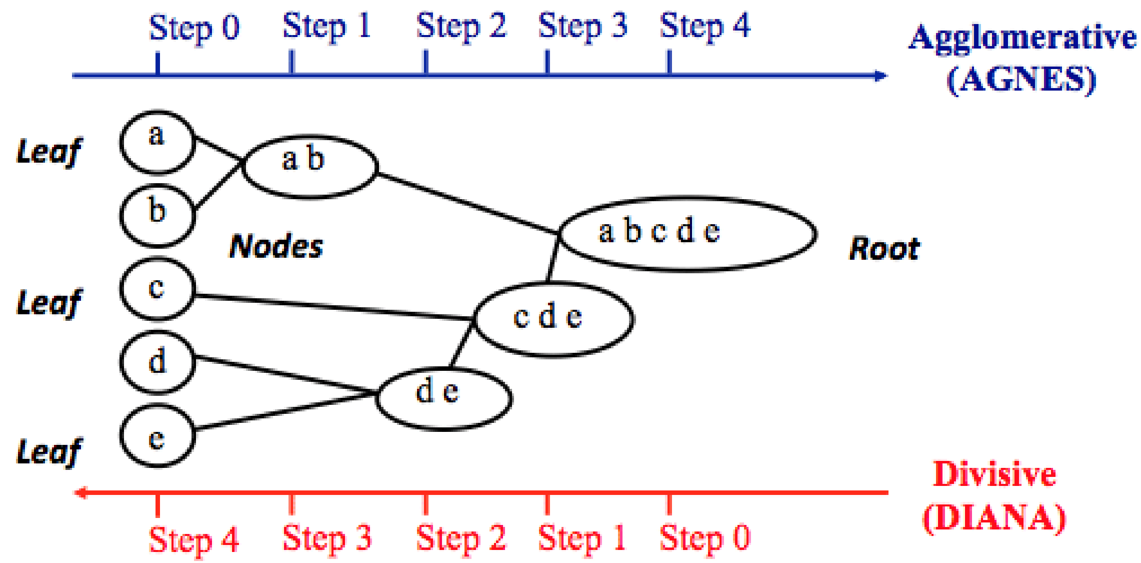
Density-Based Spatial Clustering of Applications with Noise

- Advantage
  - Not-necessary to select number of clusters
  - Identifies outliers as noise
  - Work on distance matrix

- Disadvantage
  - Doesn’t perform as well as others when the clusters are of varying density
  - Drawback also occurs with very high-dimensional data

Hierarchical Clustering

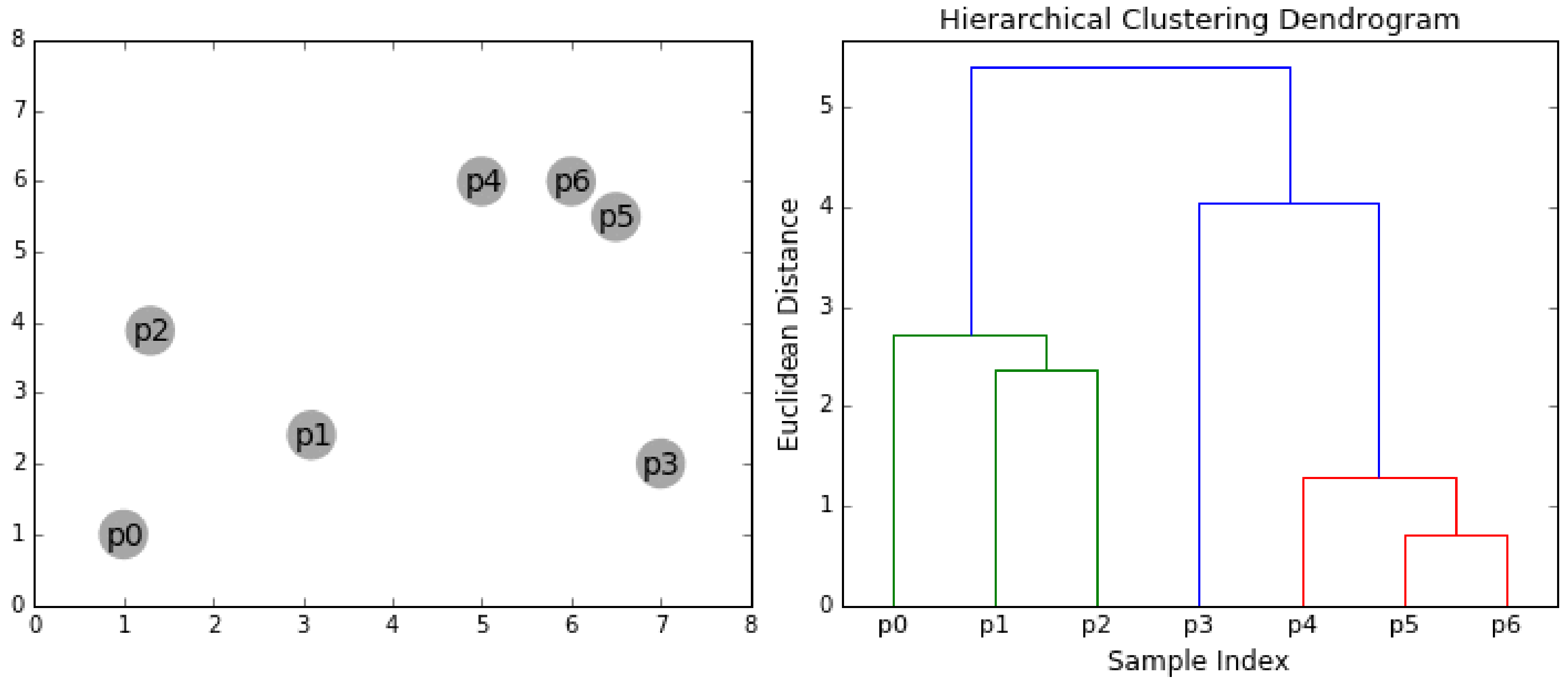
- top-down or bottom-up
  - Agglomerative or divisive



Agglomerative Hierarchical Clustering

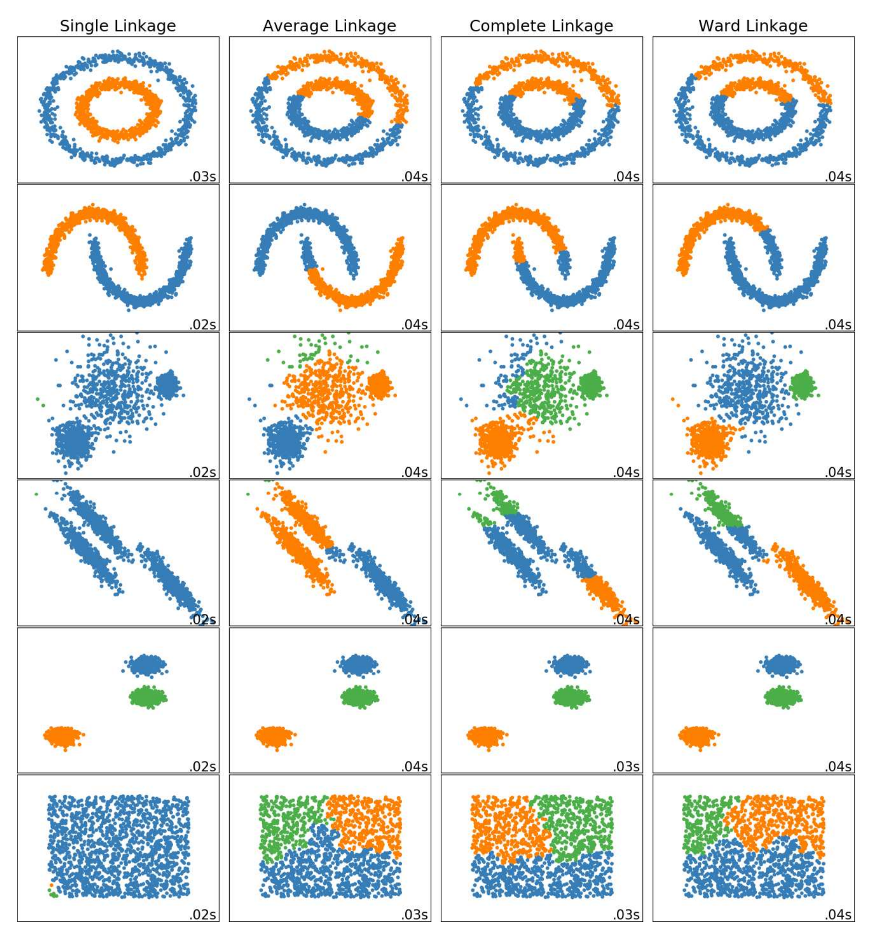
- Steps
  1. Select smallest distance from distance matrix (selects two data points)
  2. Combine selected two clusters into one new data point. Recompute distances to all other data points.
  3. Repeated 1 and 2 until we end up with one cluster or decide to stop.

Agglomerative Hierarchical Clustering



Agglomerative Hierarchical Clustering

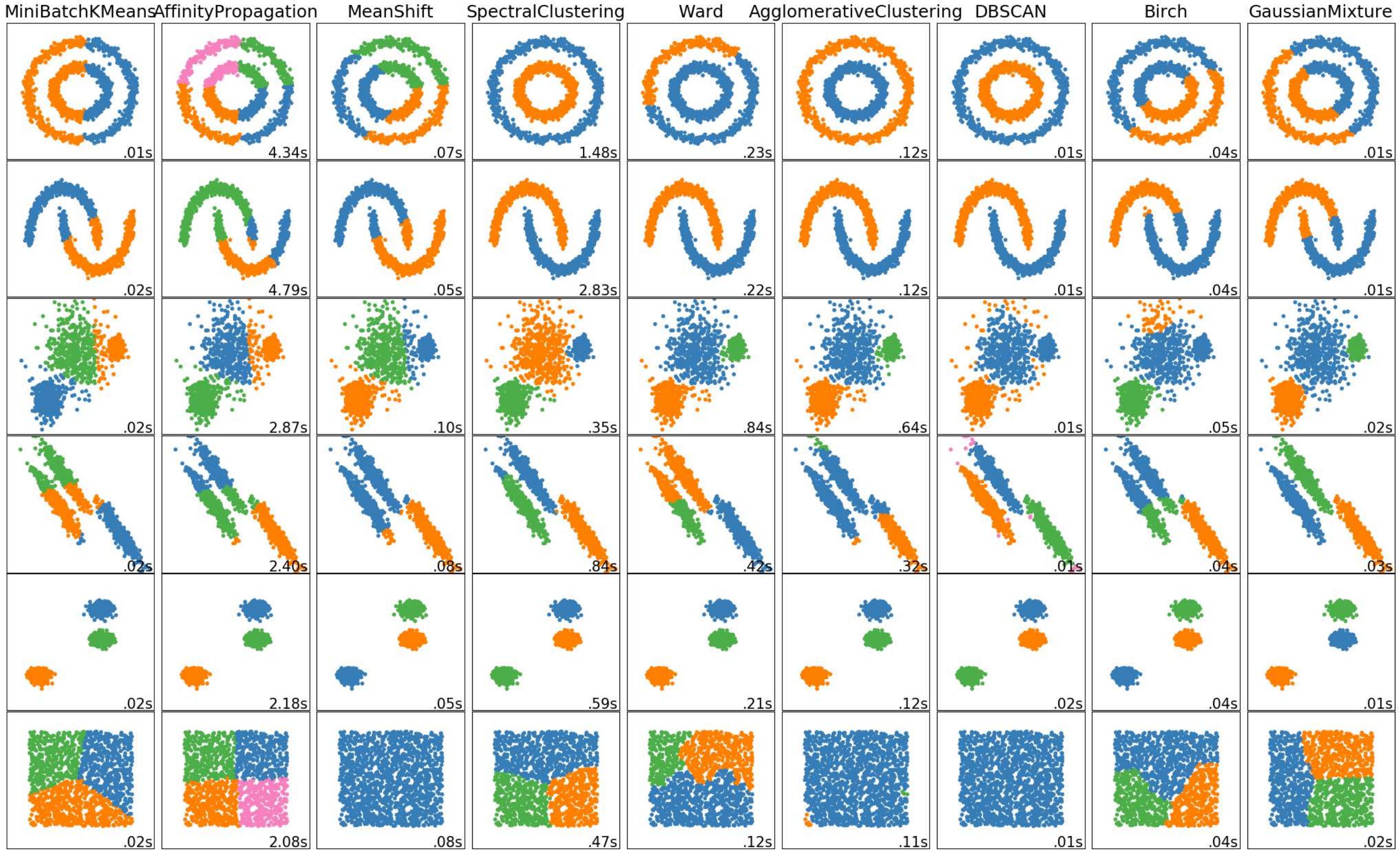
- Different agglomeration function:
  - Max = Complete linkage
  - Min = Single linkage
  - Mean = Average linkage
  - Ward linkage
    - Ward's minimum variance criterion minimizes the total within-cluster variance.



Agglomerative Hierarchical Clustering

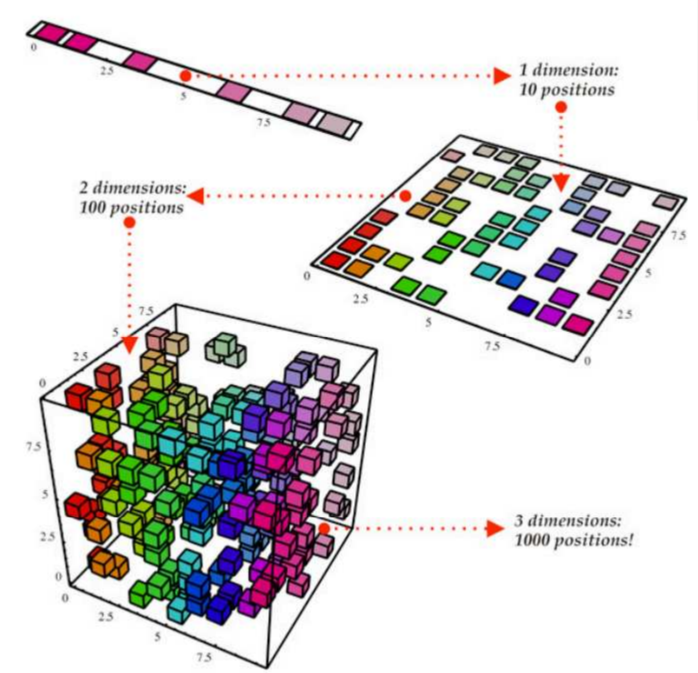
- Advantage
  - Get all ‘number of clusters’
  - Uncover underlying hierarchy
  - Work on distance matrix
  - Agglomeration method can modulate results
- Disadvantage
  - High computational complexity
    - complexity $O(n^3)$
  - Still simple cluster shapes

Clustering overview



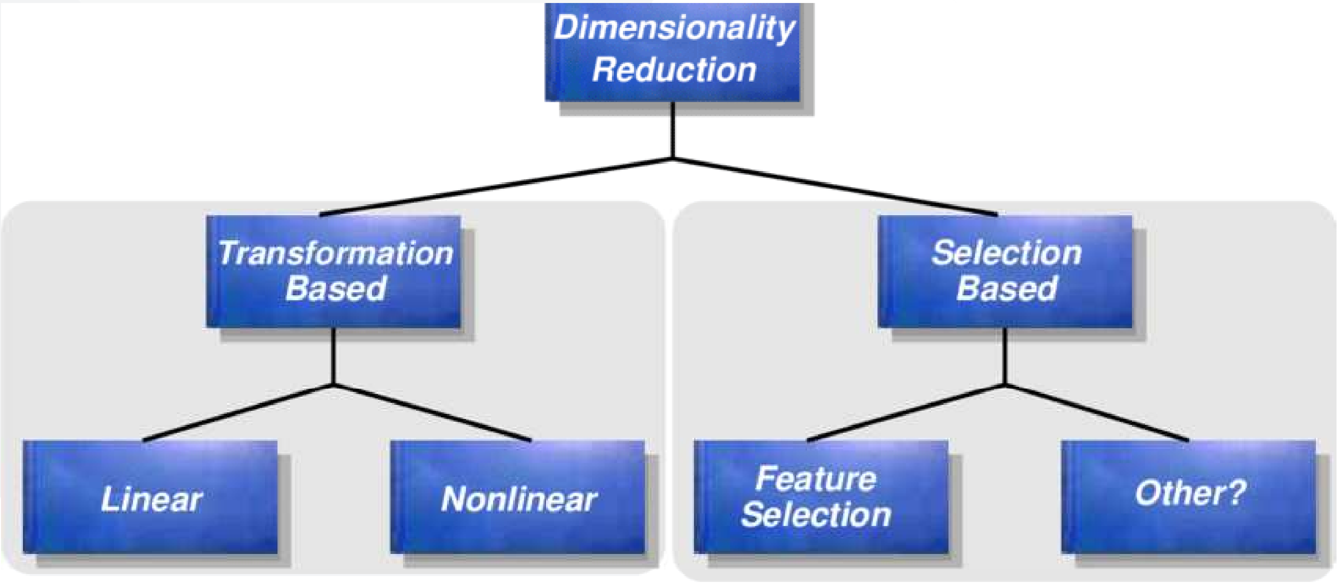
Dimension reduction

- Why?



Dimension reduction

- Feature selection
- Feature projection

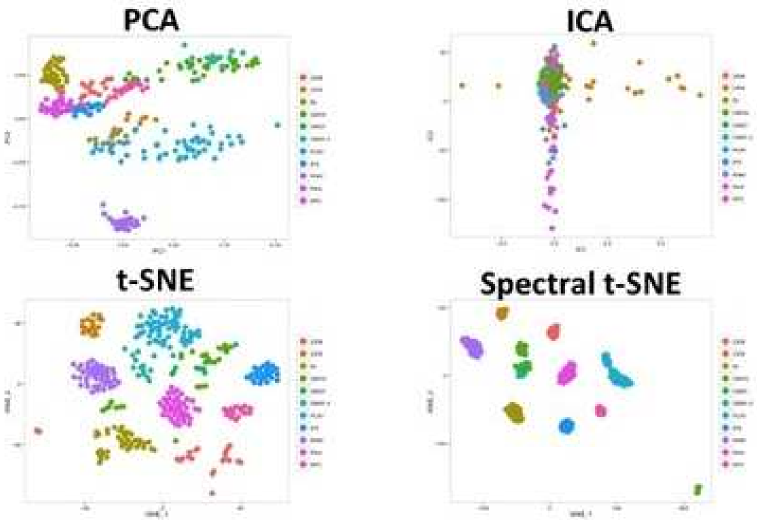


Feature selection

- Subset selection method
  - Exhaustive
  - Greedy forward selection
  - Genetic algorithm
- Evaluation metric type
  - Wrapper methods
    - Build full model and test
  - Filter methods
    - mutual information
    - Pearson product-moment correlation coefficient
  - Embedded methods

Feature projection

- Principal Component Analysis (PCA)
- Non-negative matrix factorization
- Independent Component Analysis (ICA)
- t-SNE



PCA

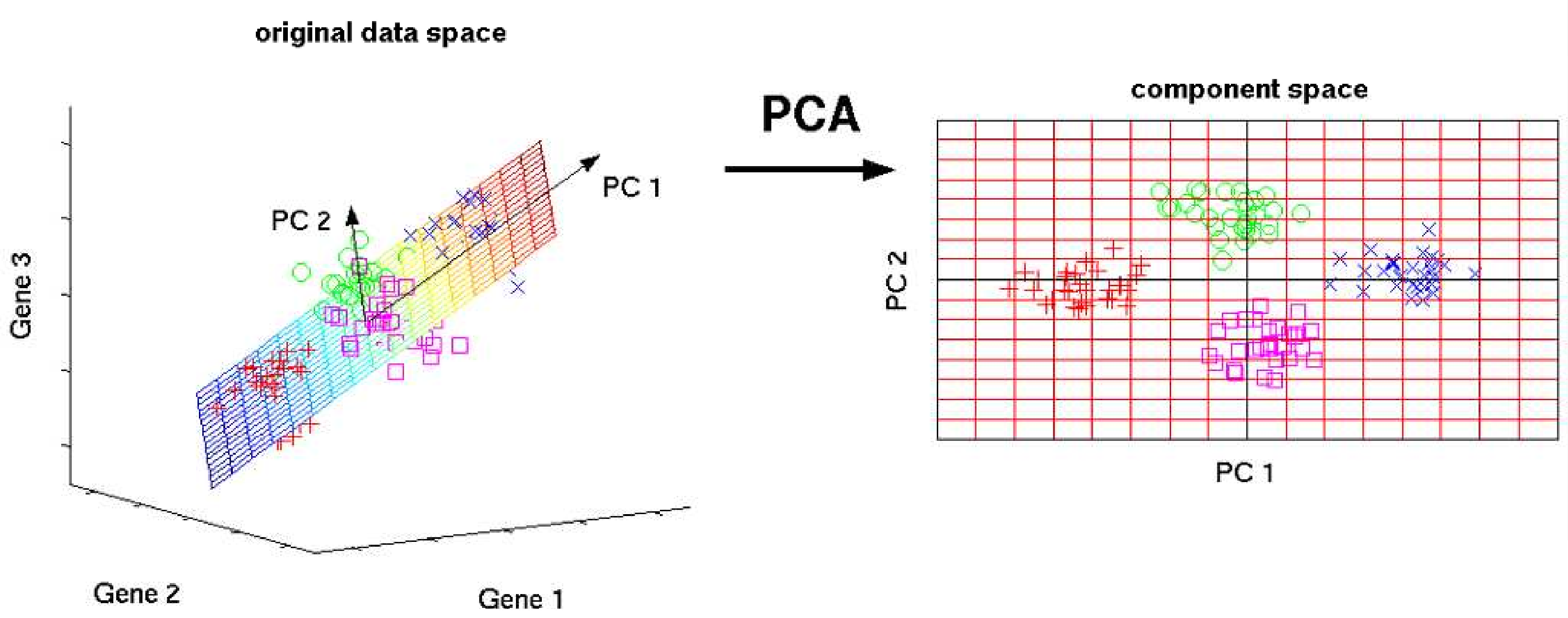
ICA

t-SNE

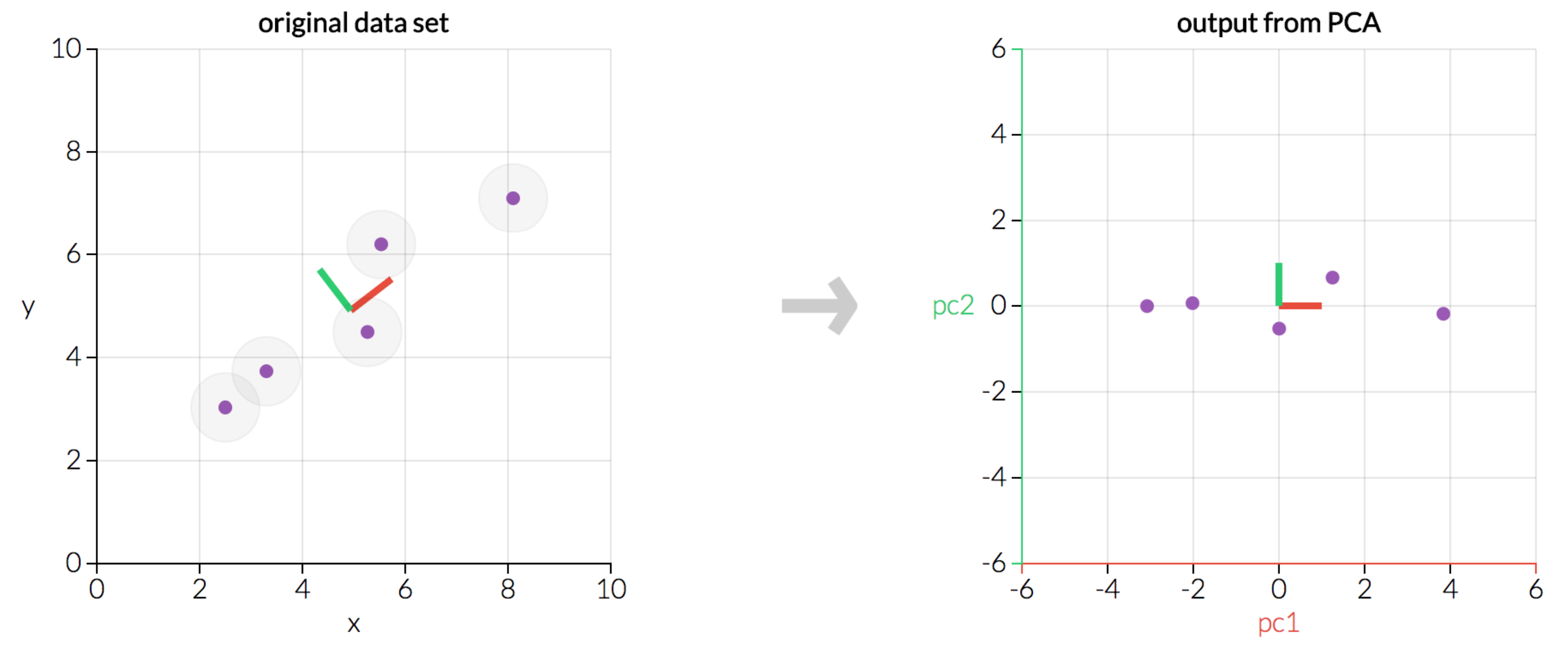
Spectral t-SNE

PCA

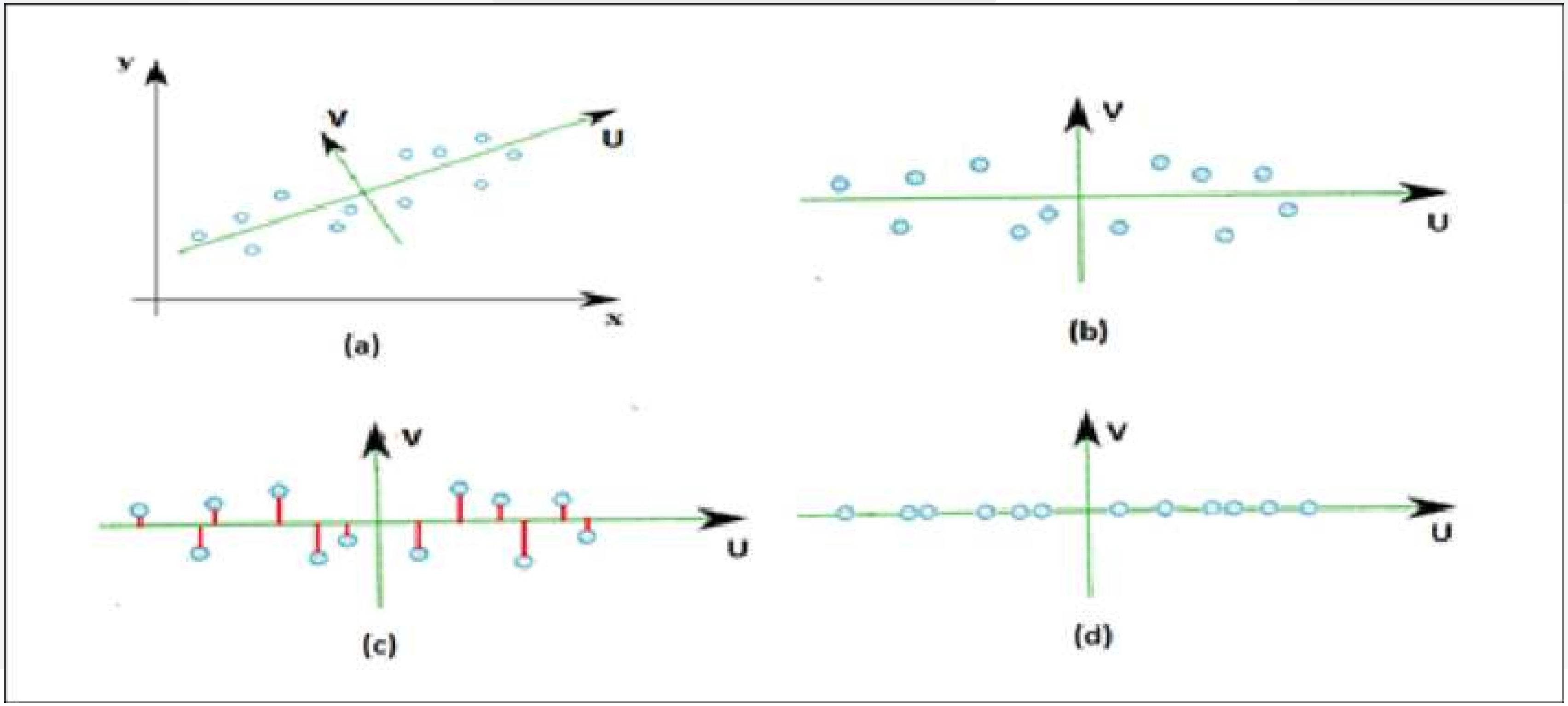
- Highlight variance
- Mainly used to visualize data better



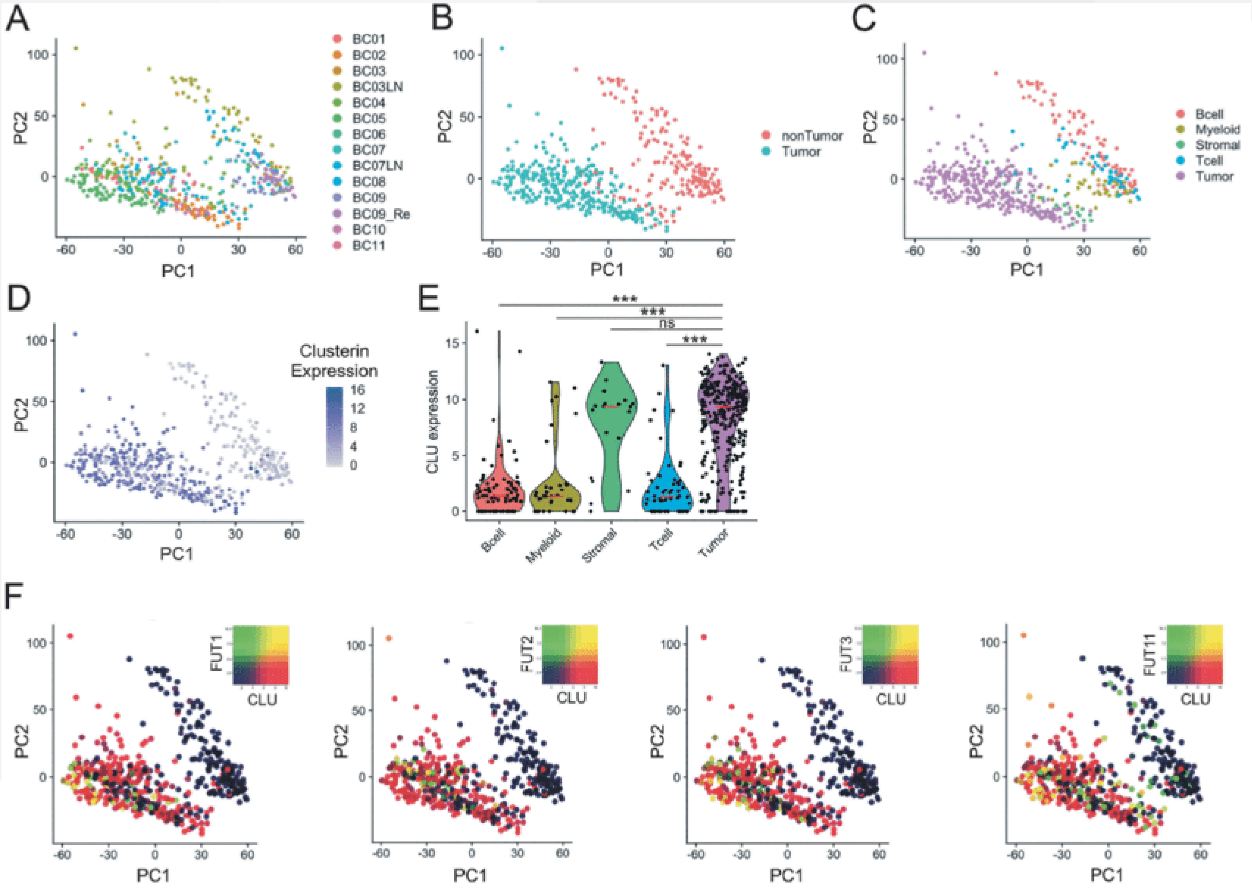
PCA


original data set
10
8
6
4
2
0
y
0
2
4
6
8
10
x
output from PCA
6
4
2
0
-2
-4
-6
pc2
-6
-4
-2
0
2
4
6
pc1

PCA



PCA


A
B
C
D
E
F
PC1
PC2
BC01
BC02
BC03
BC03LN
BC04
BC05
BC06
BC07
BC07LN
BC08
BC09
BC09_Re
BC10
BC11
nonTumor
Tumor
Bcell
Myeloid
Stromal
Tcell
Tumor
Clusterin Expression
16
12
8
4
0
CLU expression
***
***
ns
***
FUT1
FUT2
FUT3
FUT11
CLU
-60
-30
0
30
60
100
50
0
15
10
5

PCA

- Warnings:
  - Check variance explained
  - Check normality of your data-points
    - Relatively higher values can ‘capture’ the whole PCA

CEITEC

@CEITEC_Brno

Thank you for your attention!
60 minutes lunch break.

www.ceitec.eu