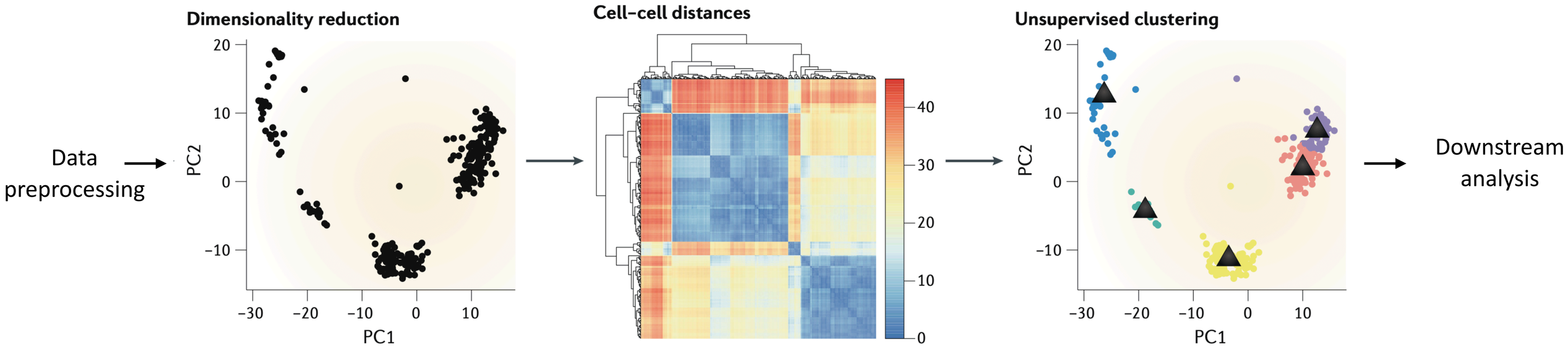
Introduction to Bioinformatics (GS011143)

2025 Spring

Statistics Lecture 3

Instructor: Lulu Shang (Department of Biostatistics, MDACC)

Overview



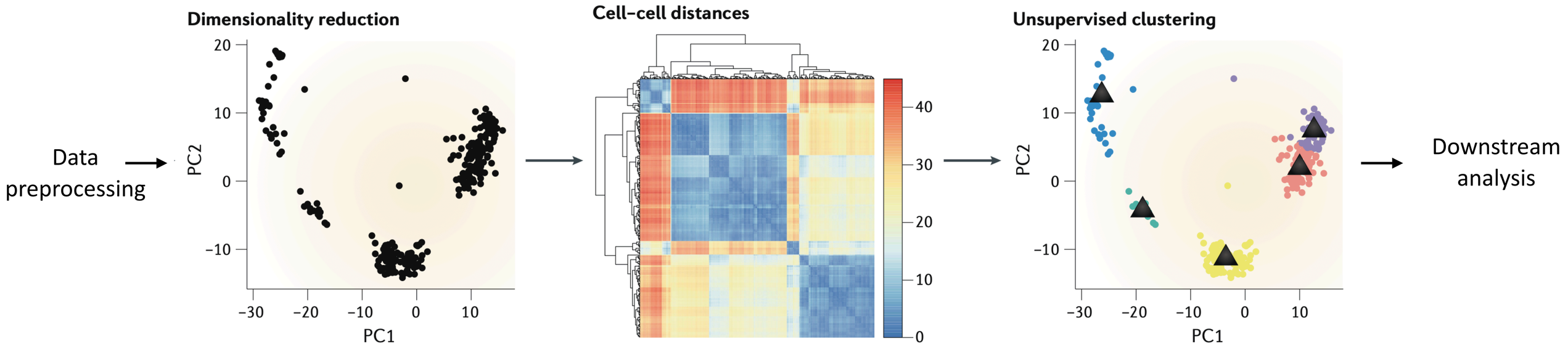
Example data analysis workflow for scRNA-seq

Figure from Kiselev et al. 2019

Unsupervised learning/Exploratory analysis

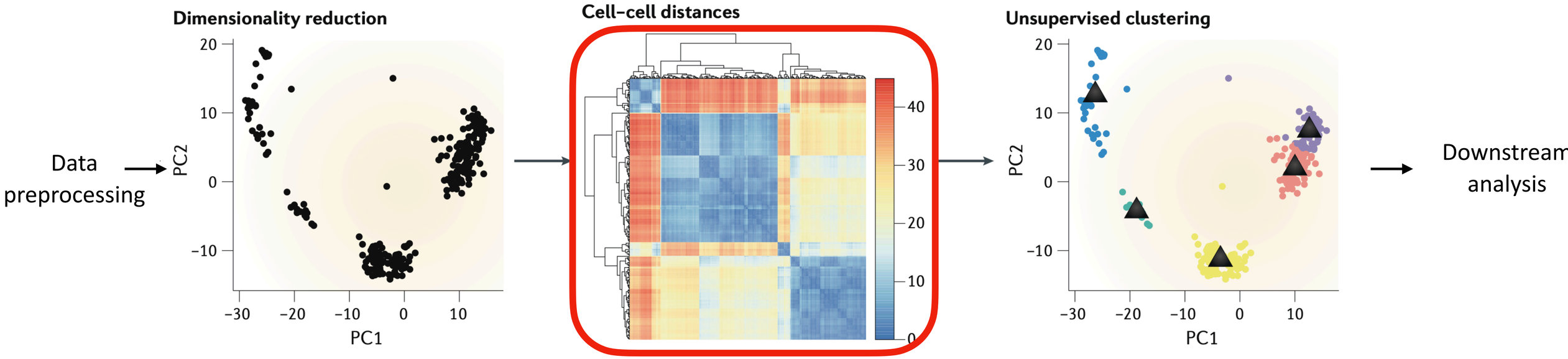
The goal of unsupervised learning is to find hidden patterns in unlabeled data

- Finding correlation patterns/distance among features (network)
- Dimension reduction
- Clustering



Example data analysis workflow for scRNA-seq

Correlation & Distance



Correlation

- Multivariate data may include redundant information
- Correlation is defined as the quantification of the degree to which two random variables are related/associated
- Not directional
- Pearson Correlation Coefficient, given a pair of random variables (*X*, *Y*)

$$r = \frac{\mathbb{E}[(X - u_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- r is ranged from -1 and 1
- When we have r=1 or -1, there is exact positive/negative linear relationship

Correlation

$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$(x_3, y_3)$$
$$\vdots$$
$$(x_n, y_n)$$

- Multivariate data may include redundant information
- Correlation is defined as the quantification of the degree to which two random variables are related/associated
- Not directional
- Suppose we have observed data x and y with length n
- **Sample** Pearson Correlation Coefficient

$$r = \frac{1}{n-1} \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}$$

- r is ranged from -1 and 1
- When we have r=1 or -1, there is exact positive/negative linear relationship

Correlation

$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$(x_3, y_3)$$
$$\vdots$$
$$(x_n, y_n)$$

- Multivariate data may include redundant information
- Correlation is defined as the quantification of the degree to which two random variables are related/associated
- Not directional
- Suppose we have observed data x and y with length n
- **Sample** Pearson Correlation Coefficient

$$r = \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

- r is ranged from -1 and 1
- When we have r=1 or -1, there is exact positive/negative linear relationship

Correlation

$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$(x_3, y_3)$$
$$\vdots$$
$$(x_n, y_n)$$

- Multivariate data may include redundant information
- Correlation is defined as the quantification of the degree to which two random variables are related/associated
- Not directional
- Suppose we have observed data x and y with length n
- **Sample** Pearson Correlation Coefficient

$$r = \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

z-score of x_i z-score of y_i

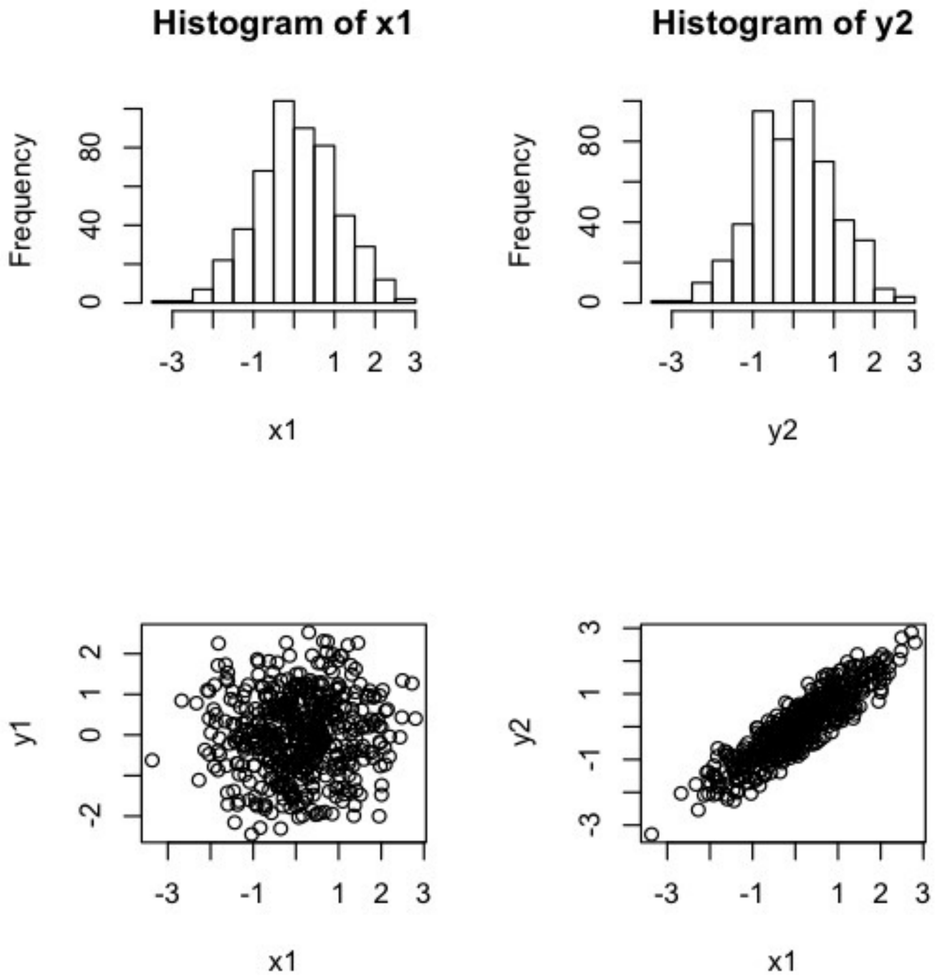
- r is ranged from -1 and 1
- When we have r=1 or -1, there is exact positive/negative linear relationship

Simulating Example

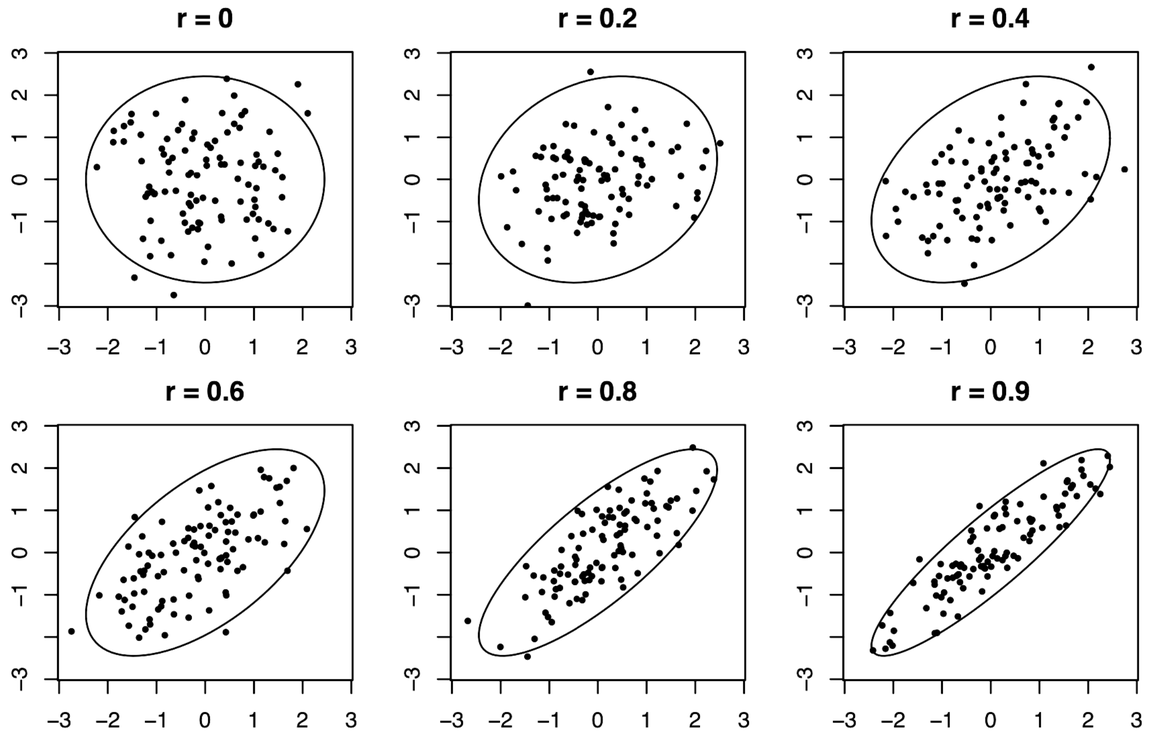
```r
set.seed(2707)
x1 <- rnorm(500,0,1)
y1 <- rnorm(500,0,1)

y2 <- 2*x1 + y1 # y2 is linearly related with x1 and y1
y2 <- y2-mean(y2) # center y2
y2 <- y2 / sd(y2) # scale y2
par(mfrow=c(2,2),mar=c(4.5,4.5,1,1))
hist(x1)
hist(y2)
```

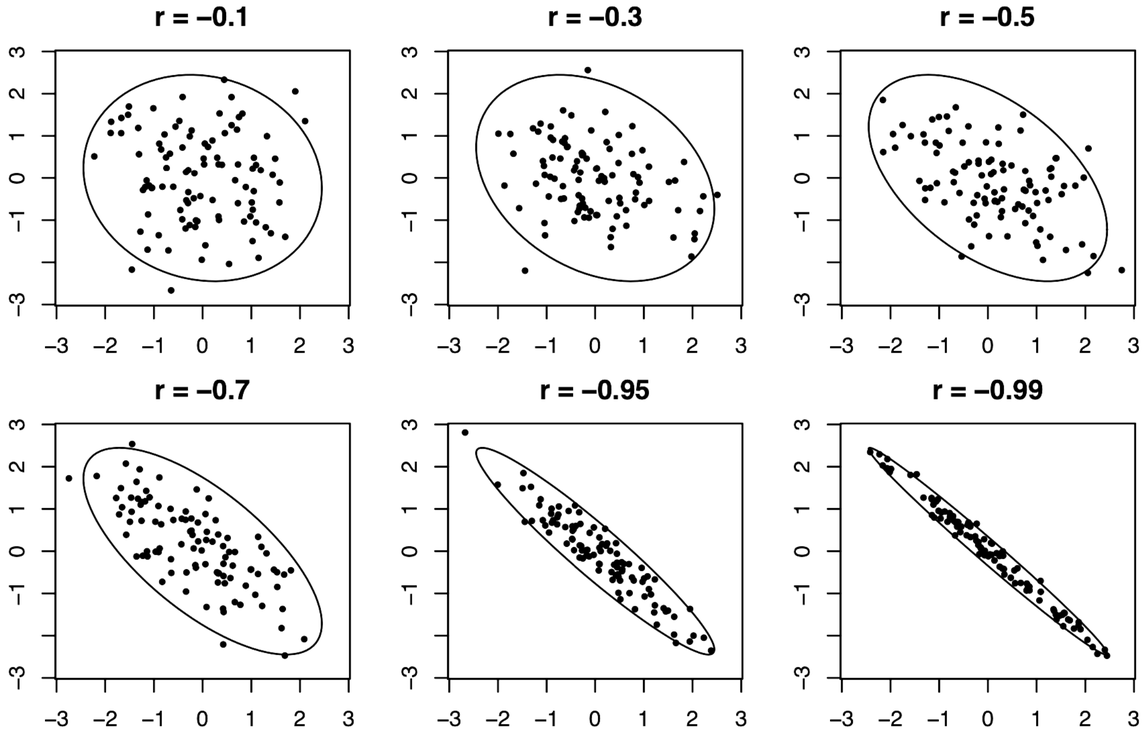
```r
plot(x1, y1)
plot(x1, y2)
```



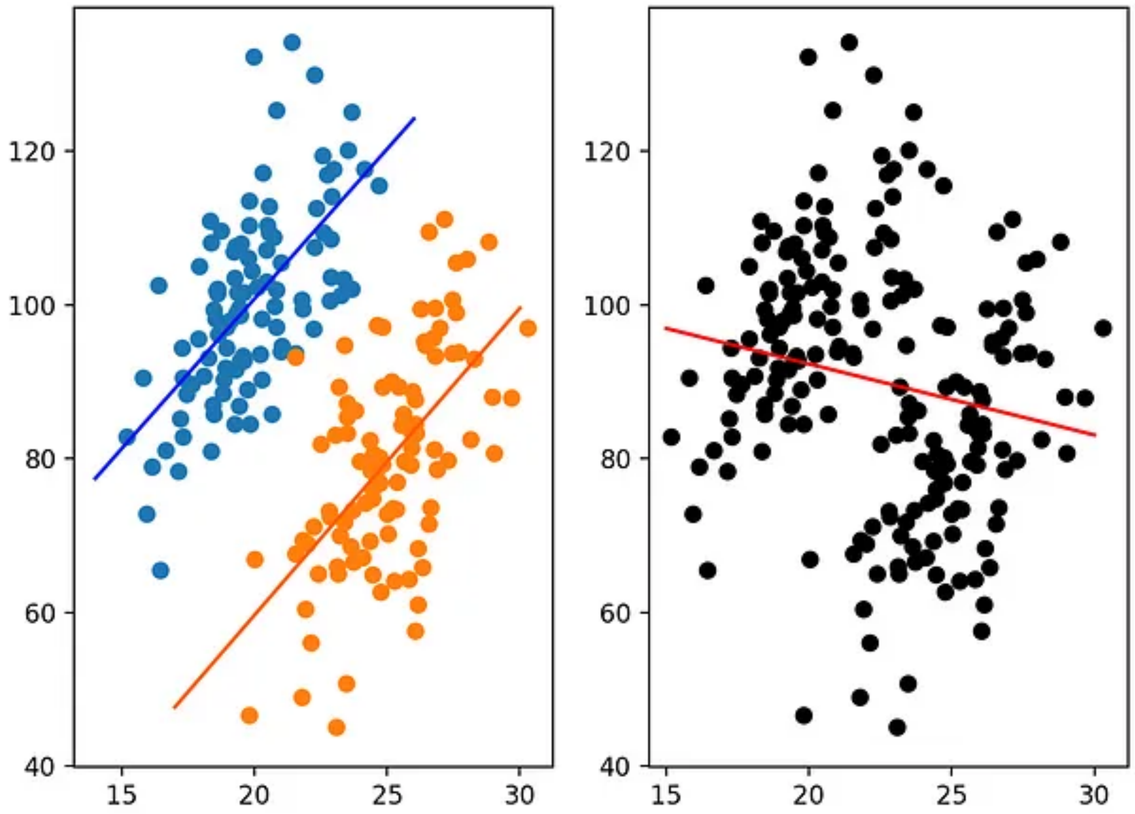
Positive correlations



Negative correlations



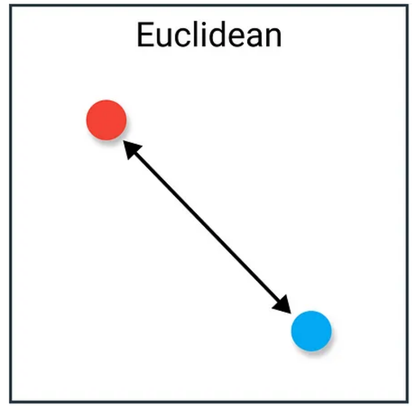
Simpson’s paradox

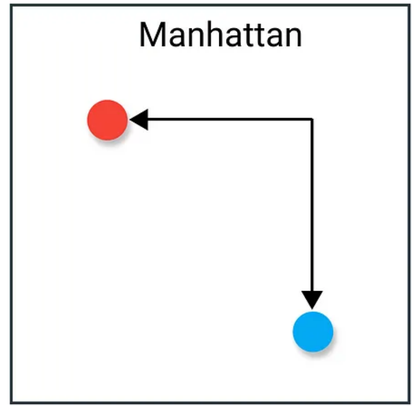


- An overall r can be misleading when data points are clustered.

Distance Metrics

- Distance metrics are used in supervised and unsupervised learning to calculate similarity in data points.
- Let $x = (x_1, \ldots, x_p)^T$ and $y = (y_1, \ldots, y_p)^T$
- Euclidian Distance: $d_{xy} = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$
- Manhattan Distance: $d_{xy} = \sum_{i=1}^{p} |x_i - y_i|$
- 1-abs(correlation), proportional to Euclidian distance square for standardized data

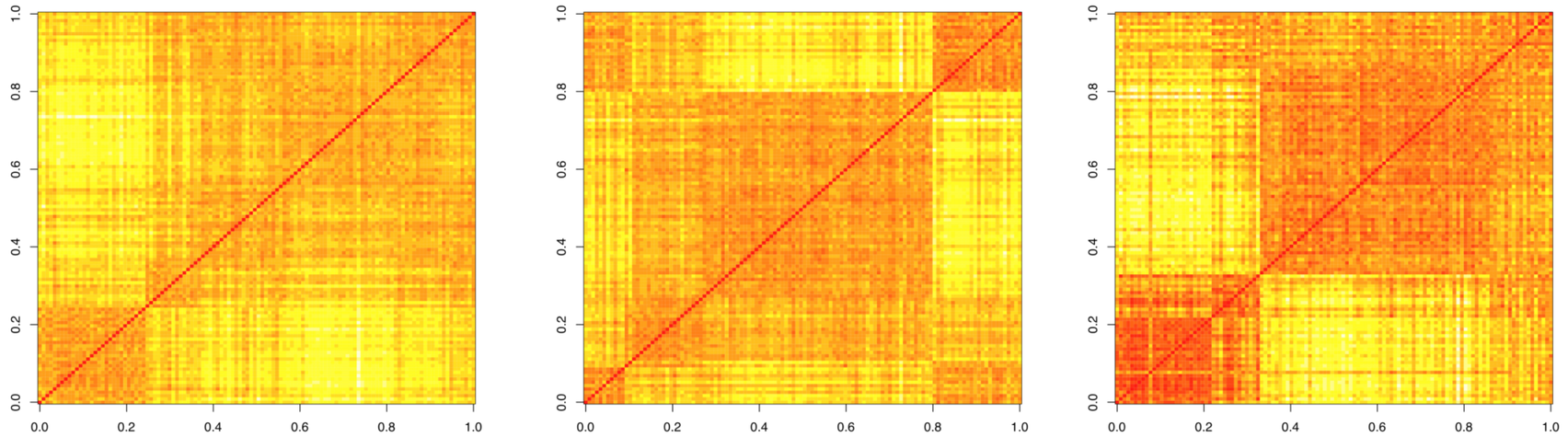




. . .

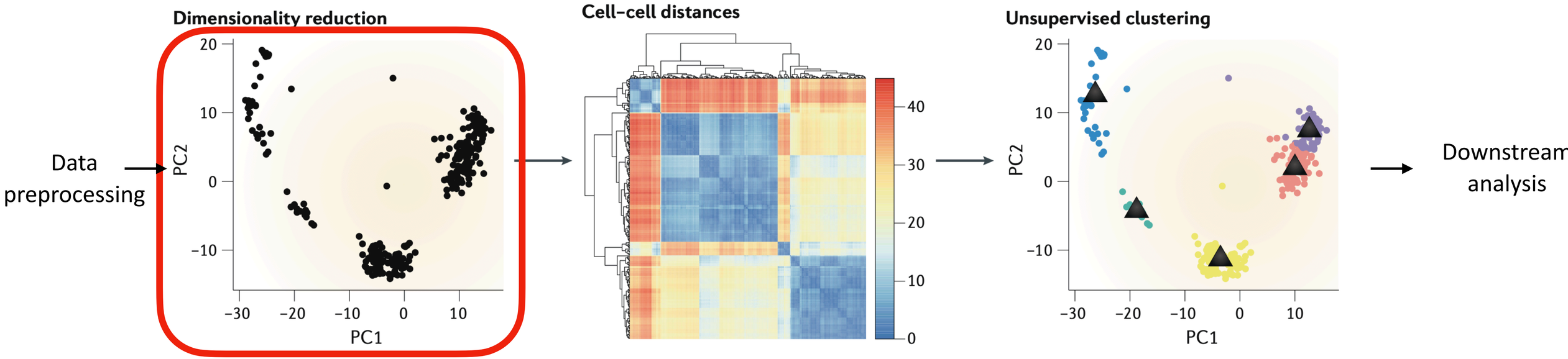
and more

Distance metrics compared



Conclusion: ***distance matters!***

Dimension Reduction



Principal Component Analysis (PCA)

We have variables that display strong pairwise correlation → we can reasonably think that we can reduce dimensionality of the data without losing too much information.

Key Ideas:

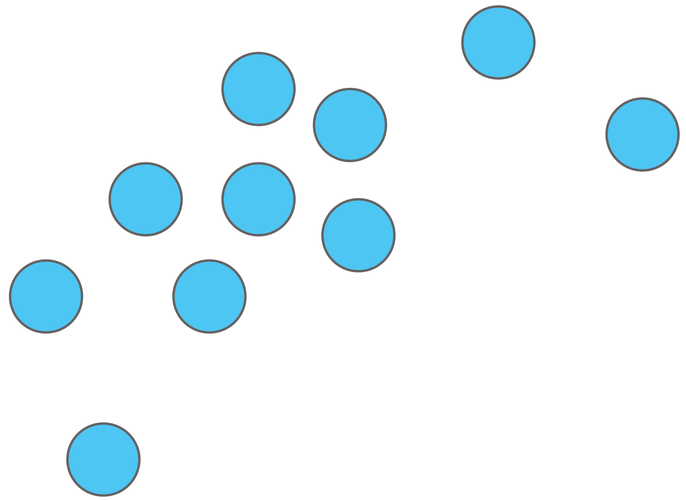
- Reduce dimesionality of the data
- Avoid loss of relevant information

In other words:

- *transform* a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components
- At the same time, keep the variability of original data

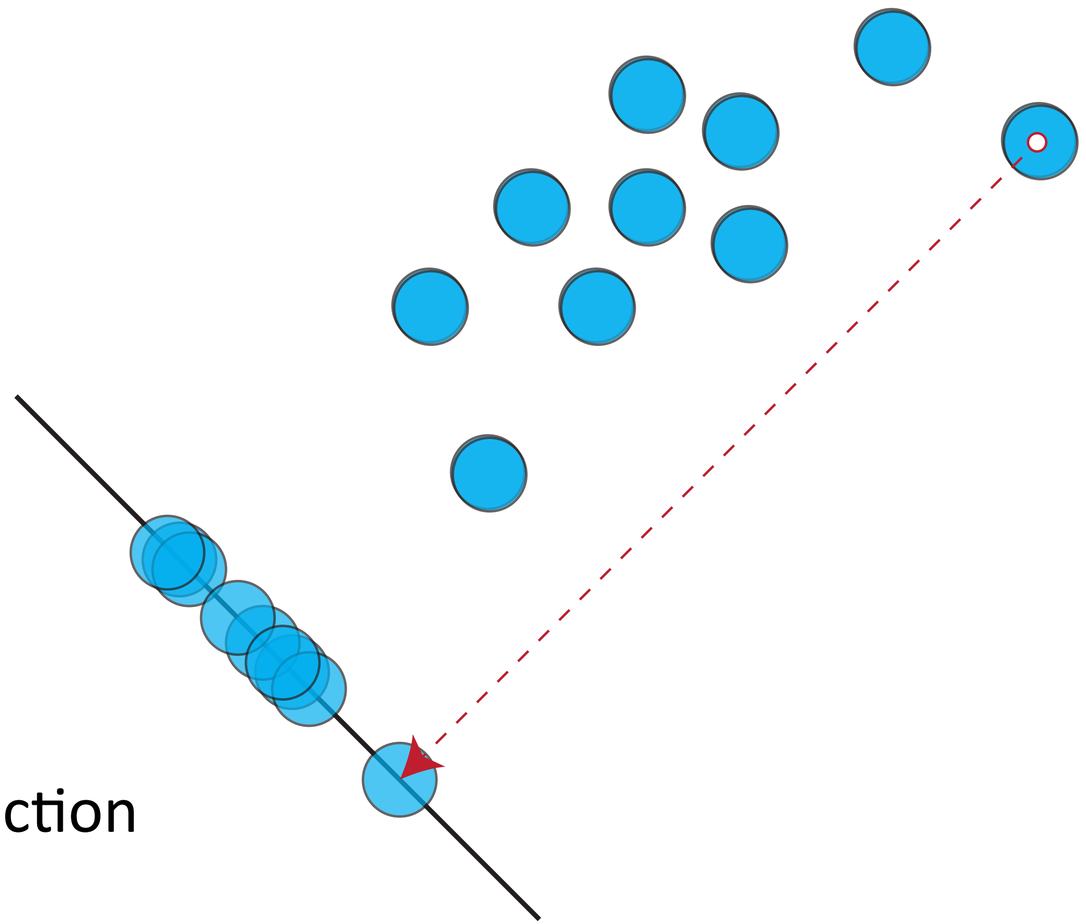
Principal Component Analysis (PCA)

PCA seeks to find the axis that explains the maximum amount of variance in the data.



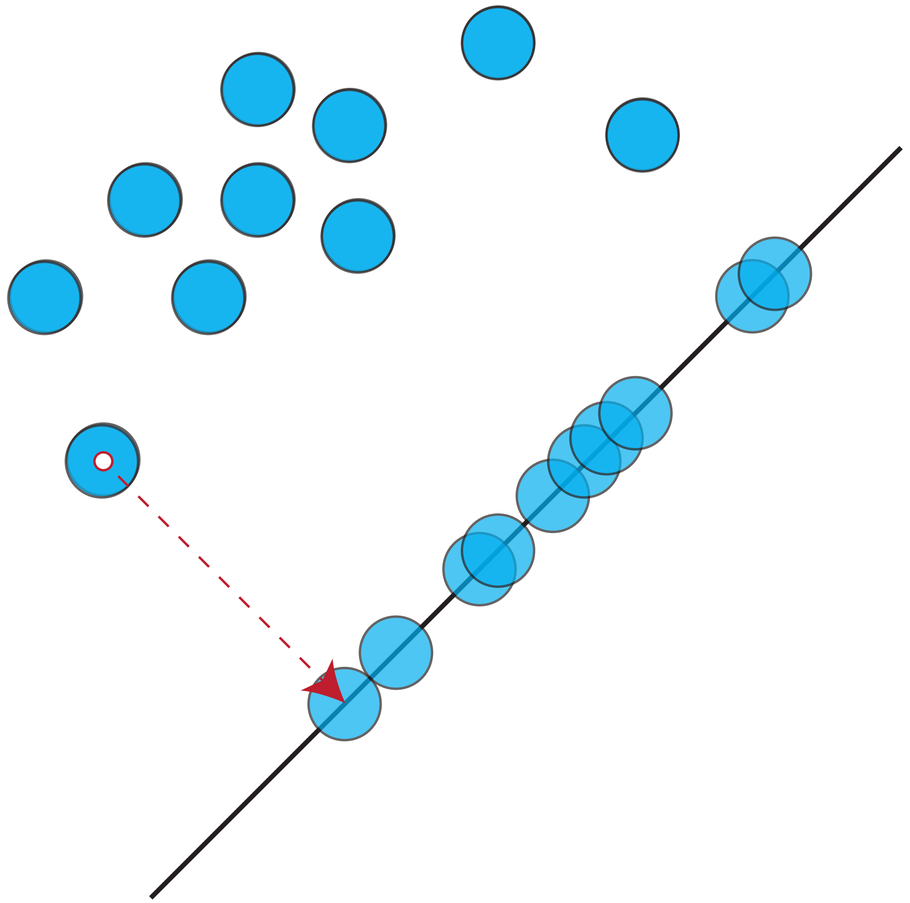
Data points

Principal Component Analysis (PCA)



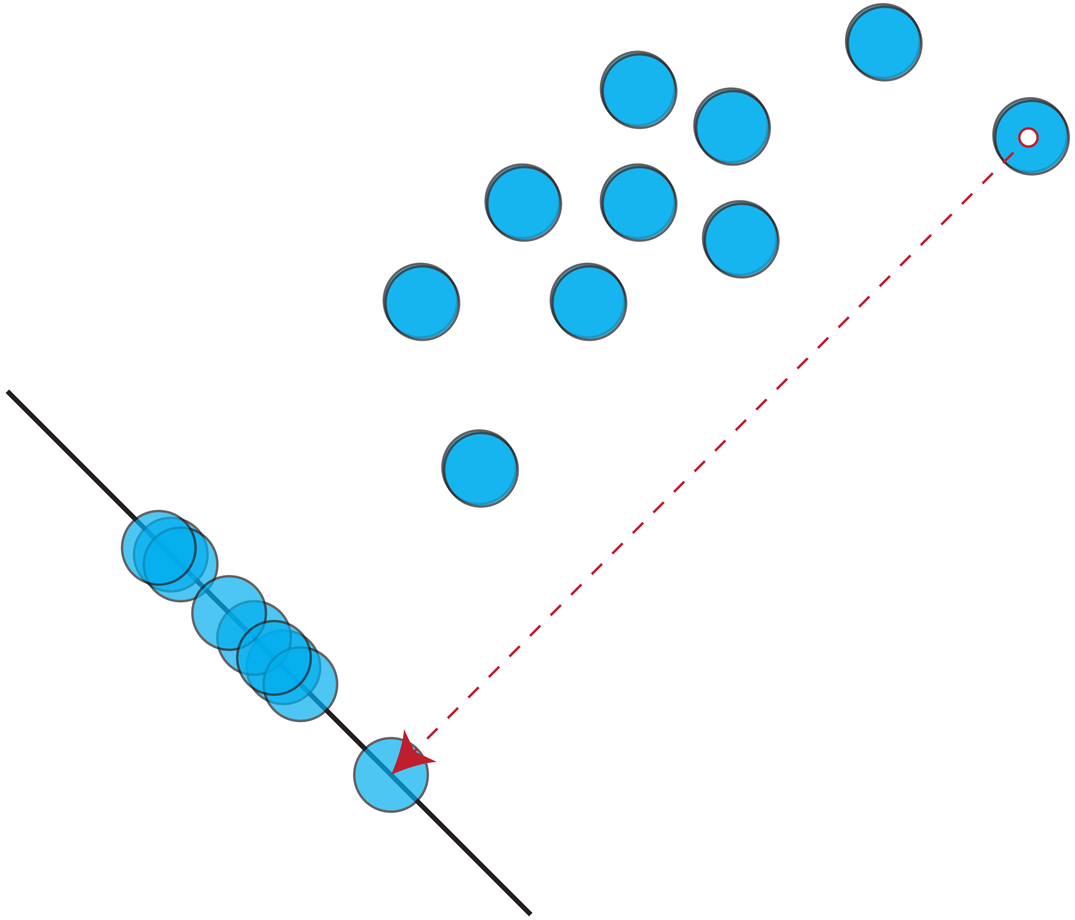
Do projection

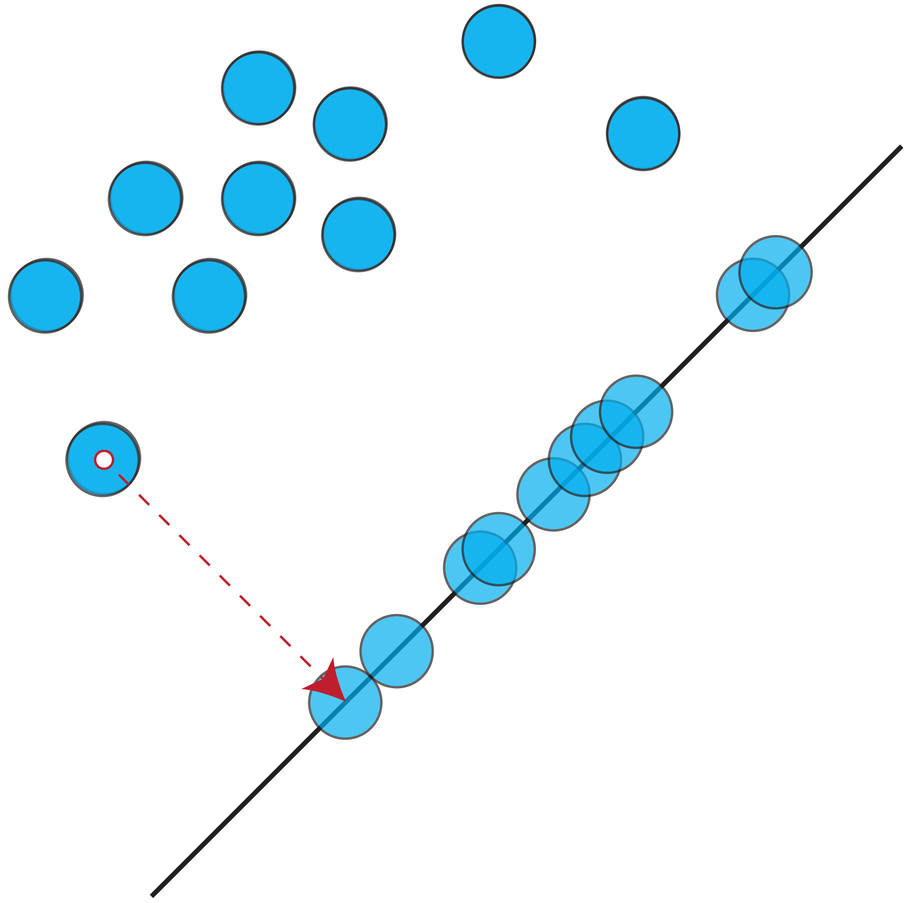
Principal Component Analysis (PCA)



Or you can

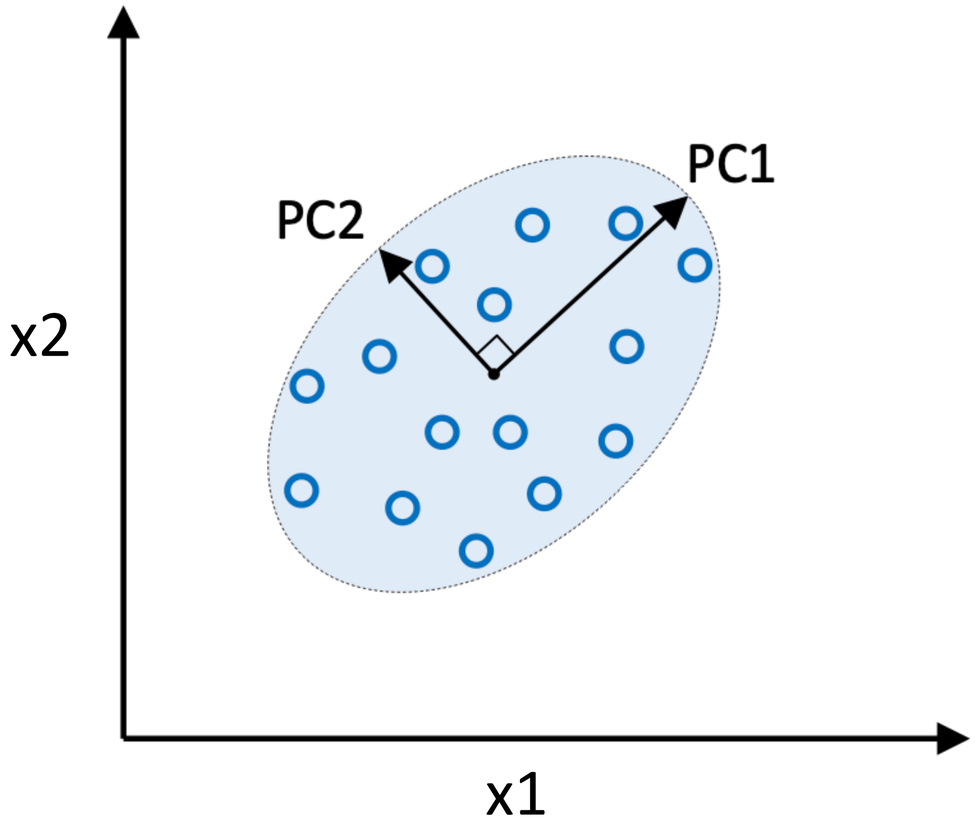
Principal Component Analysis (PCA)





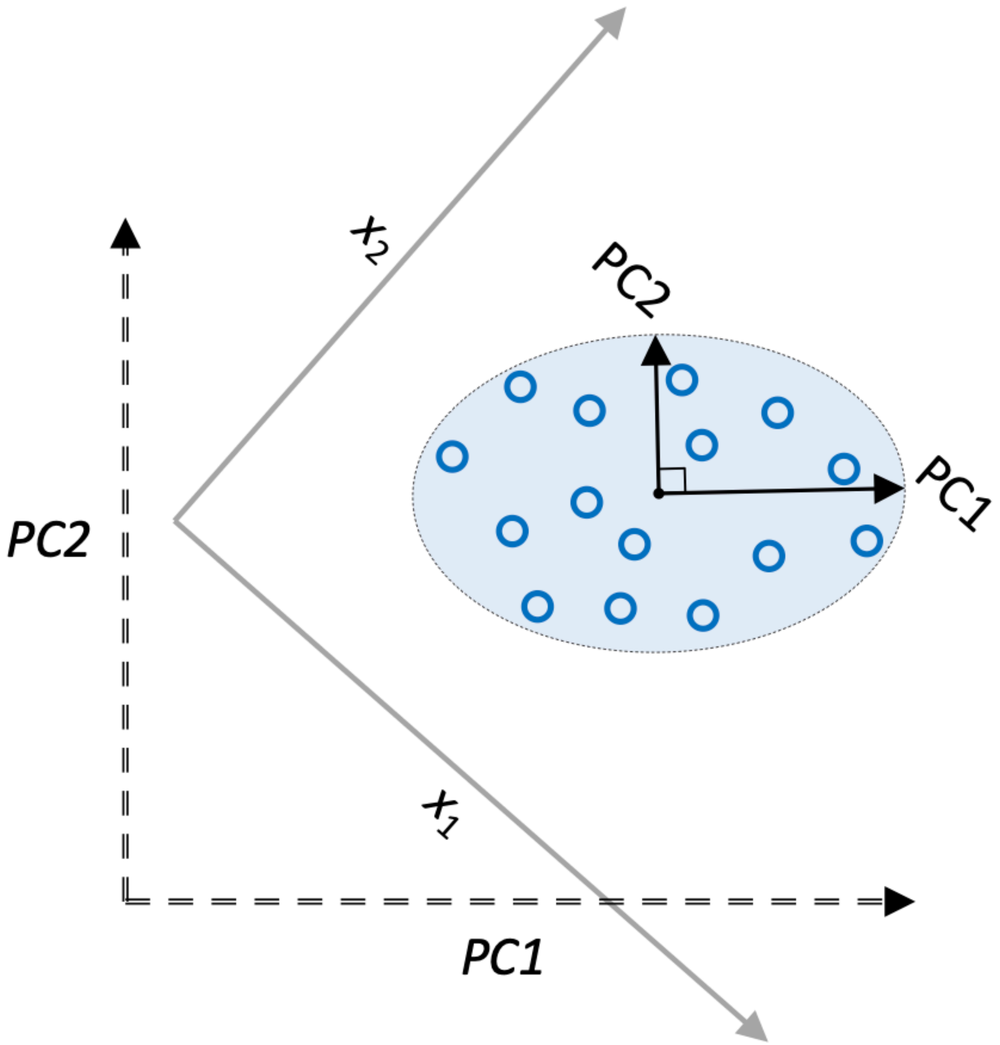
Which one preserves more variability in the data?

Principal Component Analysis (PCA)



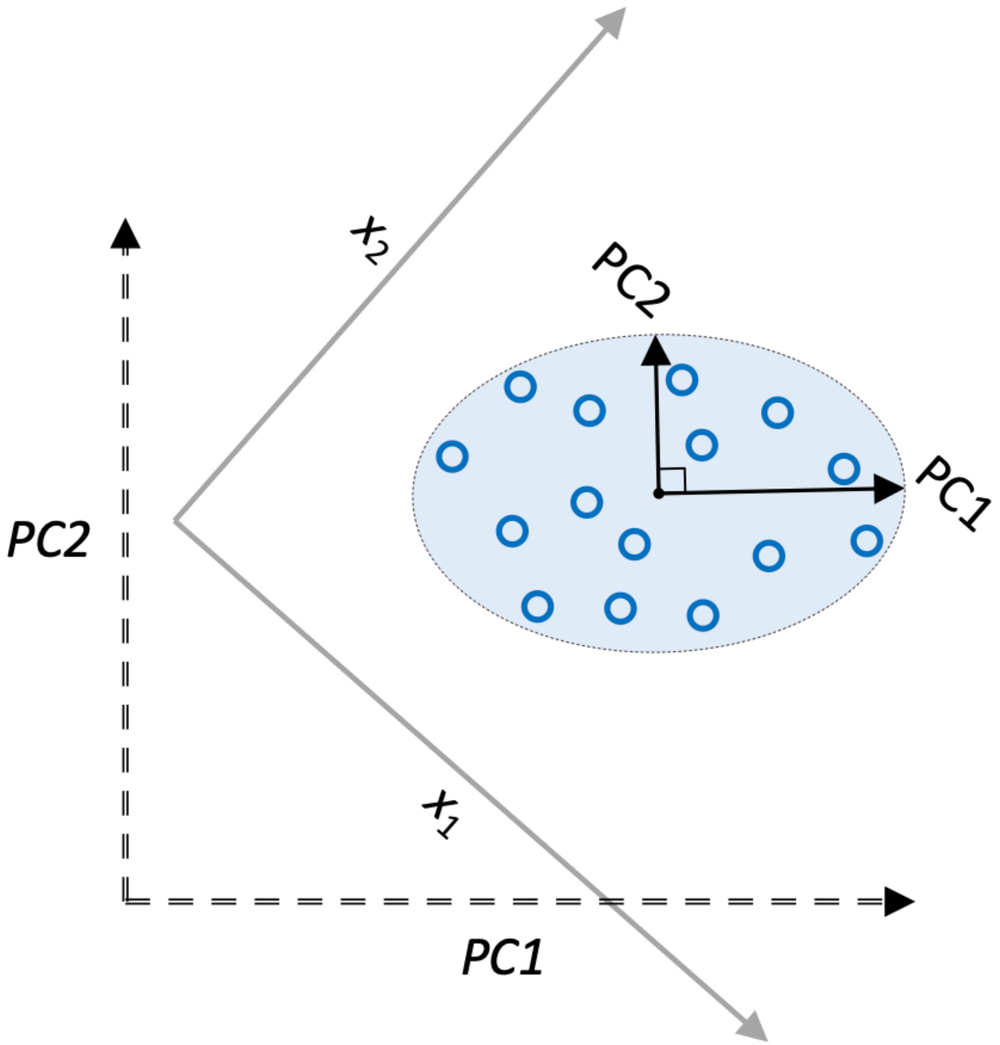
1. Find directions of maximum variance

Principal Component Analysis (PCA)

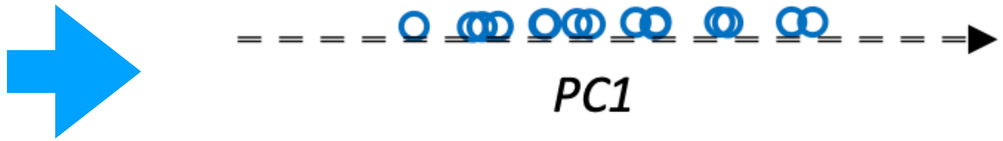


1. Find directions of maximum variance
2. Transform features onto directions of maximum variance

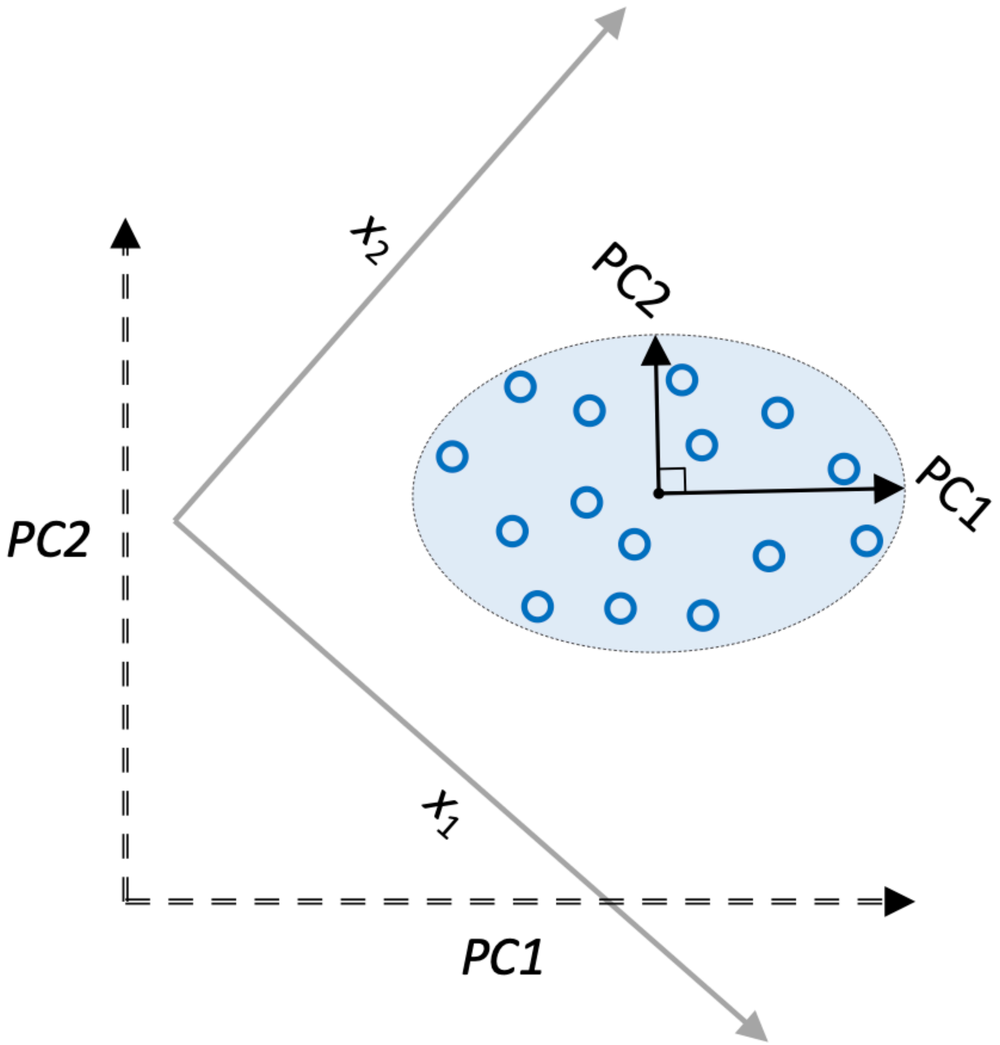
Principal Component Analysis (PCA)



1. Find directions of maximum variance
2. Transform features onto directions of maximum variance
3. Usually consider a subset of vectors of most variance (dimensionality reduction)

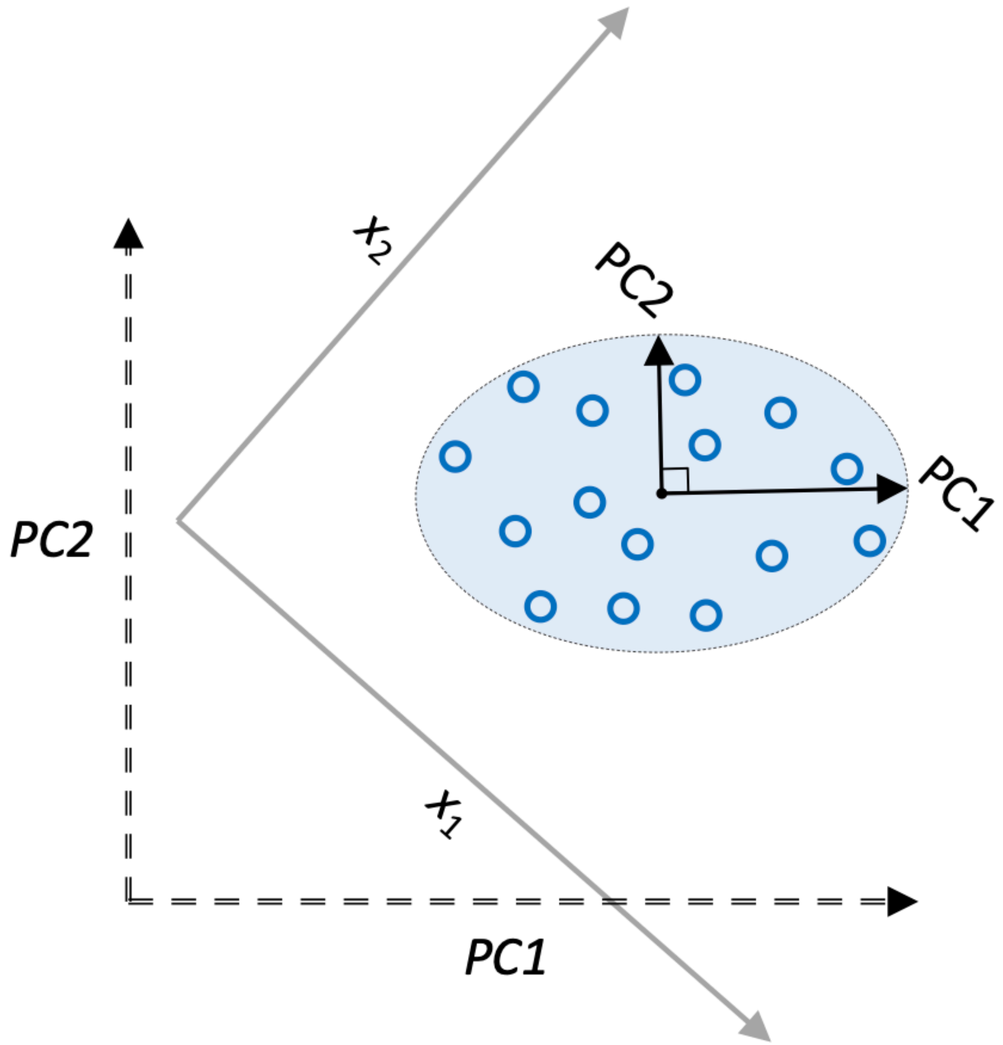


Principal Component Analysis (PCA)



- The **first principal component**
  (1) passes through the multidimensional mean
  (2) minimizes the sum of the squares of the distances of the points from the line, which is maximizing the variance along its direction

Principal Component Analysis (PCA)



- The **second principal component** is calculated in the same way as the first, with the additional constraint that it is orthogonal to the first principal component, effectively capturing the largest variance in the direction that is not already explained by the first principal component
- cov(PC1,PC2)=0
- Repeated until we get min(p,n) principal components

Principal Component Analysis (PCA)

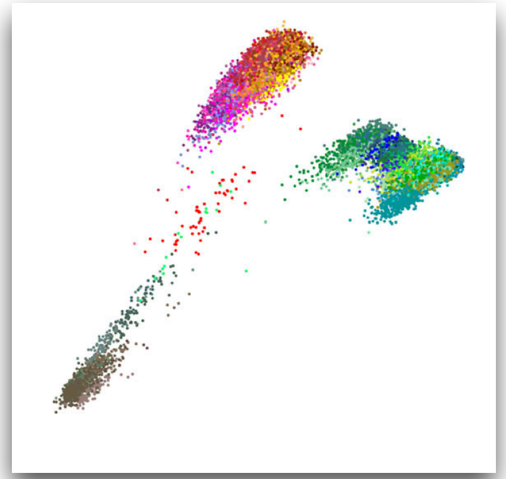
PCA is sensitive to the scaling of the variables.

- This is because it decomposes the covariance matrix, not the correlation.
- This means that whenever the different variables have different units (like temperature and mass), PCA is a somewhat arbitrary method of analysis
- Before performing PCA, one necessary step is to **standardize** each measurement vector for a variable to have mean 0 and sd 1.

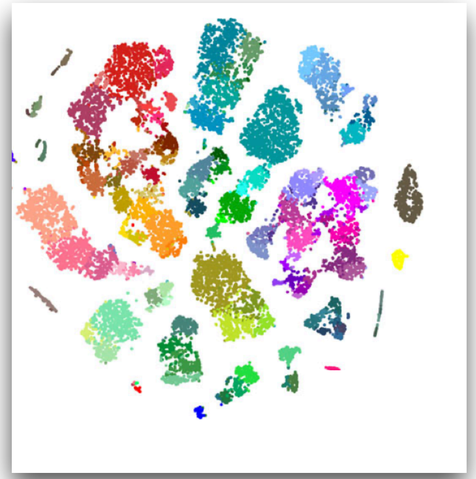
More examples are in the workshop slides.

t-SNE (t-distributed stochastic neighbor embedding)

- An **unsupervised**, **non-linear** technique primarily used for dimension reduction and visualizing high-dimensional data.
- Newer than PCA (PCA: 1933, t-SNE: 2008)
- PCA seeks to maximize variance and preserves large pairwise distances but can lead to poor visualization, especially when dealing with non-linear manifold structures.
- **t-SNE only preserves small pairwise distances or local similarities**



PCA



t-SNE, default

t-SNE

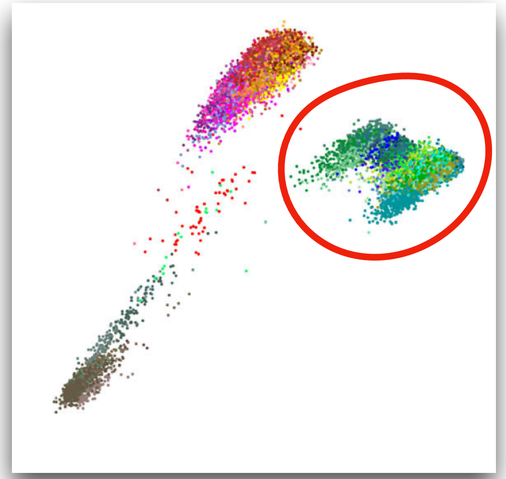
multi-scale, PCA init, high learning rate

Figure from Kobak et al. 2019

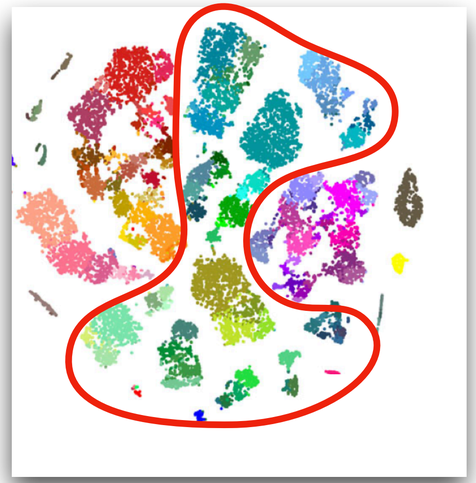
t-SNE (t-distributed stochastic neighbor embedding)

- An **unsupervised**, **non-linear** technique primarily used for dimension reduction and visualizing high-dimensional data.
- Newer than PCA (PCA: 1933, t-SNE: 2008)
- PCA seeks to maximize variance and preserves large pairwise distances but can lead to poor visualization, especially when dealing with non-linear manifold structures.
- **t-SNE only preserves small pairwise distances or local similarities**

Be careful with the interpretation of distance on t-SNE plots.



PCA



t-SNE, default

t-SNE

multi-scale, PCA init, high learning rate

UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction)

- A newer technique by Mclnnes et al. (2018)
- More computationally efficient than tSNE and PCA
- Similar to tSNE, UMAP also uses graph layout algorithms to arrange data in low-dimensional space.
- UMAP constructs a high-dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible
- UMAP builds something called a “fuzzy simplicial complex”, which is a representation of a weighted graph, with edge weights representing the likelihood that two points are connected.
- Parameter “radius”, the points are connected when their radii overlap (small value lead to isolated clusters, large value may make all points connected).
- UMAP choose the radius based on the distance of each point to their nearest neighbour.

Play with example datasets and compare between PCA, t-SNE, UMAP: https://projector.tensorflow.org/

t-SNE and UMAP

- The biggest **differences**: the balance between local and global structure.
  - The global structure is often better preserved in UMAP than t-SNE, i.e., the inter-cluster relations are potentially more meaningful than in t-SNE.
- **Same**: any given axis or distance in lower dimensions of both t-SNE and UMAP isn't directly interpretable as the way of techniques such as PCA.
- Many packages provide implementation of t-SNE and UMAP (e.g., Seurat, scater)
  - In R, there are packages Rtsne and umap

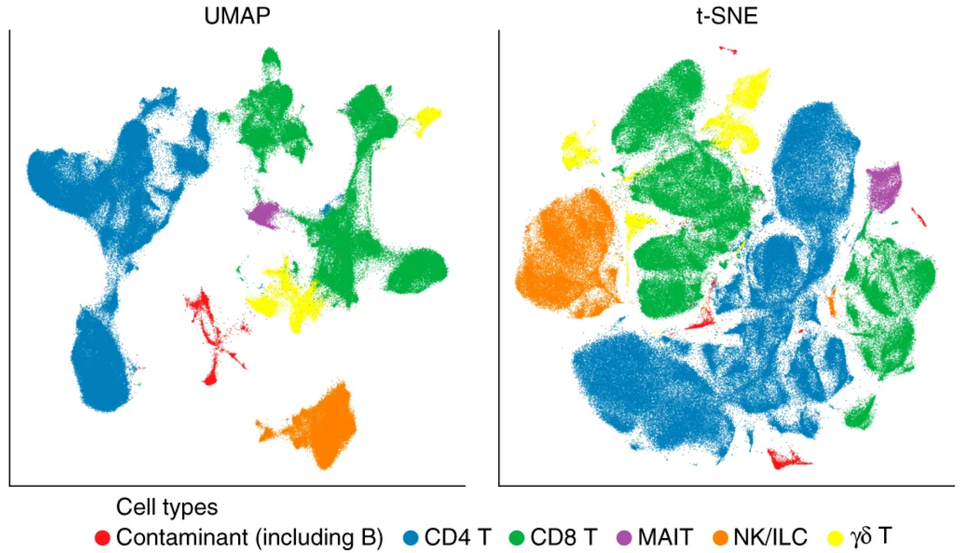
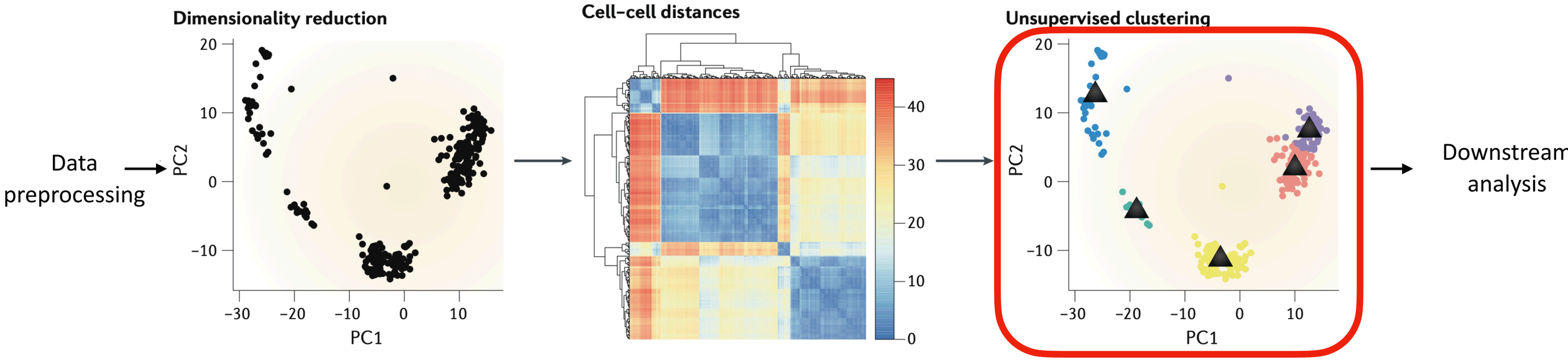


Figure from Becht et al. 2019

Cluster Analysis



Cluster Analysis

Clustering is the task of grouping a set of objects in a way that objects in the same group (=cluster) are more similar to each other than to those in other groups.

- We have two approaches
  - Partitioning methods
    - K-means
    - K-medoids (partitioning around medians)
  - Hierarchical methods
    - nested clusters

K-Means clustering

Each of the K clusters has a cluster center, called centroid (mean)

- Input of a partitioning based method
  - data matrix
  - distance function
  - number of groups
- Output
  - group assignment of every object

K-Means clustering

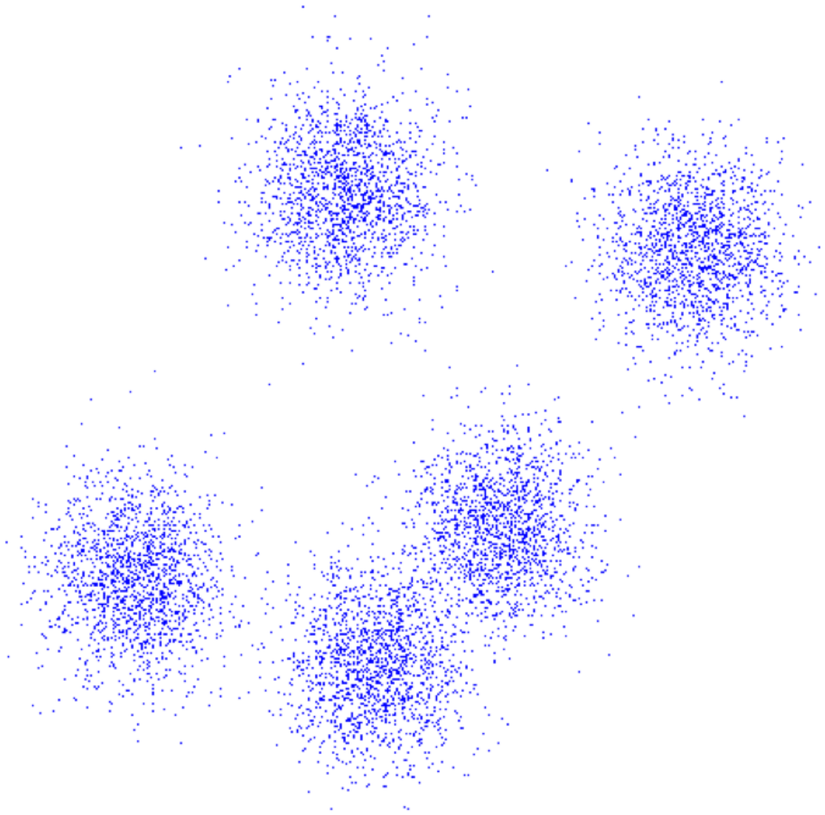
An iterative clustering algorithm

– Initialize: Pick *K* random points as cluster centers

– Alternate:

- Assign data points to closest cluster center
- Change the cluster center to the average of its assigned points

– Stop when: no points' assignments change; or no change of centroids; or minimum decrease in the sum of squared error, where error is defined by distance between each data point to the centroid



K-Means clustering

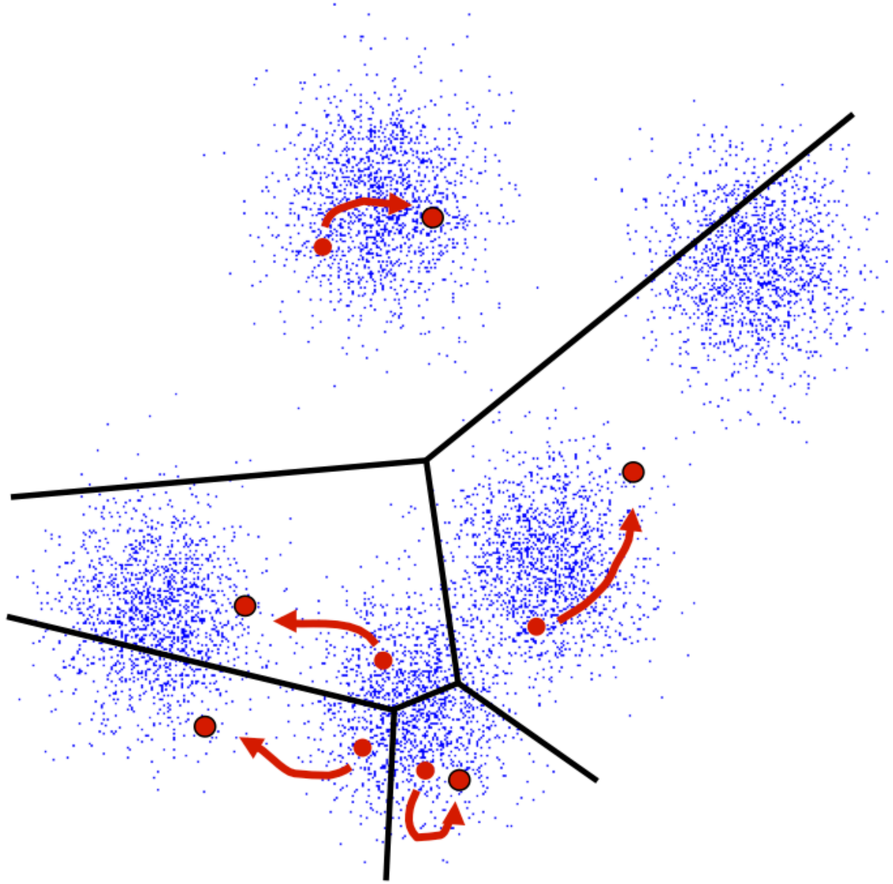
An iterative clustering algorithm

– Initialize: Pick *K* random points as cluster centers

– Alternate:

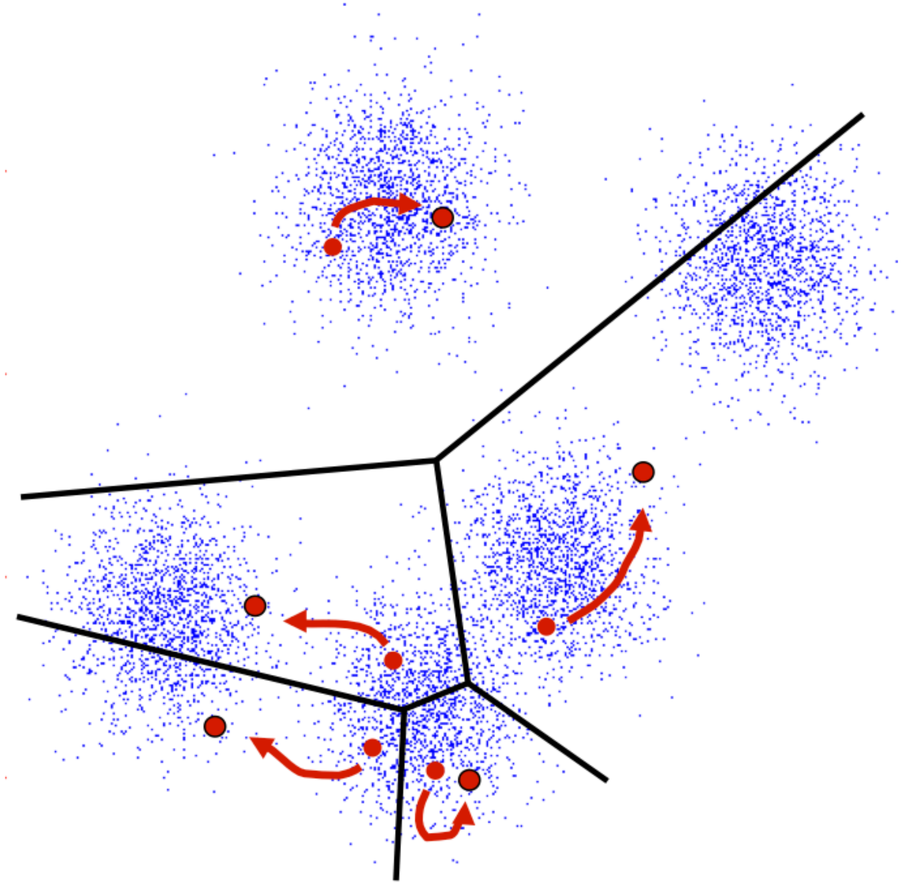
- Assign data points to closest cluster center
- Change the cluster center to the average of its assigned points

– Stop when: no points' assignments change; or no change of centroids; or minimum decrease in the sum of squared error, where error is defined by distance between each data point to the centroid



K-Medoids clustering

- One drawback of using K-Means is the fact that its centroids (mean) are sensitive to outliers. This can potentially impact the formed clusters.
- **K-Medoids** has mitigated that sensitivity by not relying on centroids, it forms clusters based on the distance to medoids (medians).



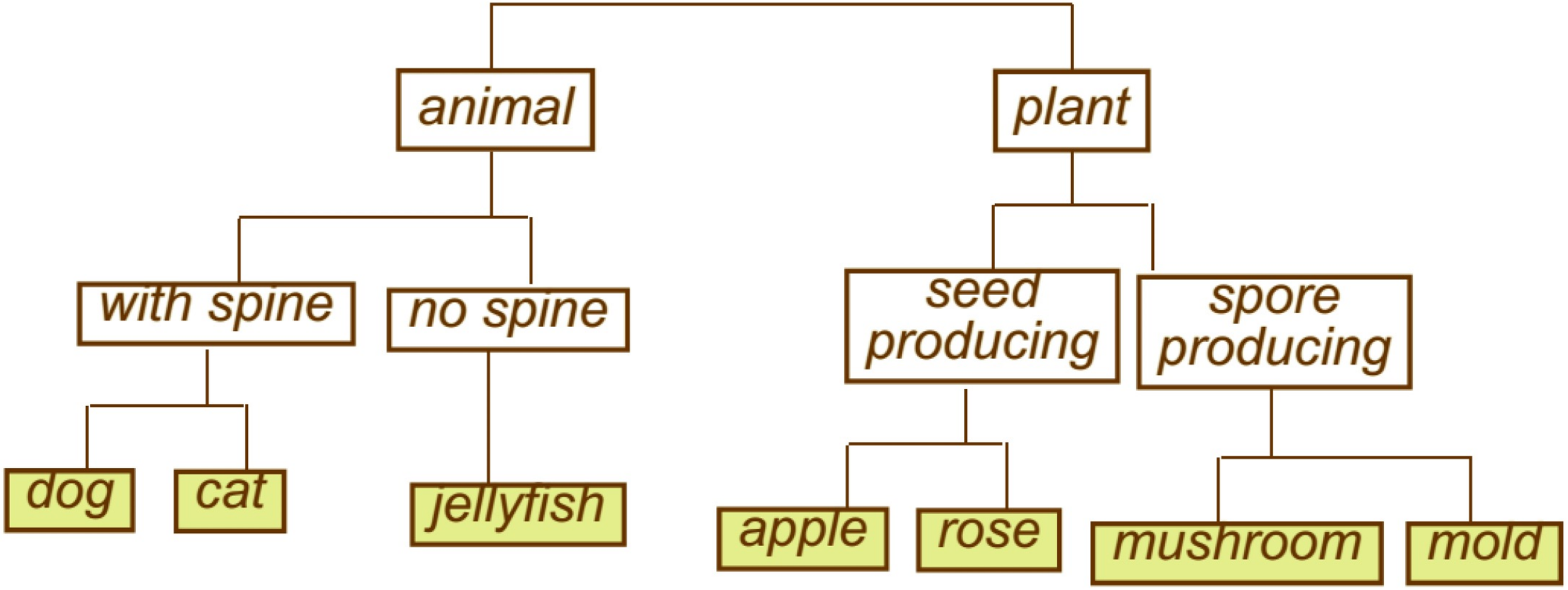
K-Means/K-Medoids clustering

Advantages	**Disadvantages**
Number of groups is well defined	Have to choose the number of groups
A clear, deterministic assignment of an object to a group	Sometimes objects do not fit well to any cluster
Simple algorithms for inference	Can converge on locally optimal solutions and often require multiple restarts with random initializations

Hierarchical clustering

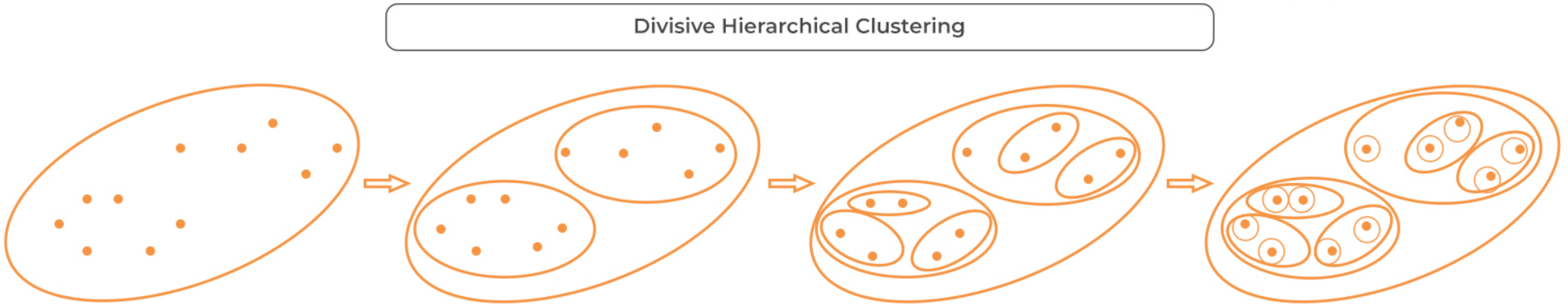
- The results of applying K-means or K-medoids clustering algorithms depend on the **choice for the number of clusters** to be searched and a starting centroid assignment
- Hierarchical clustering do not require such specifications
- They require user to **specify a measure of dissimilarity** between groups of observations
- They produce hierarchical representations in which the clusters at each level of the hierarchy are created by merging/dividing clusters at the next lower/higher level.

Example: Biological Taxonomy



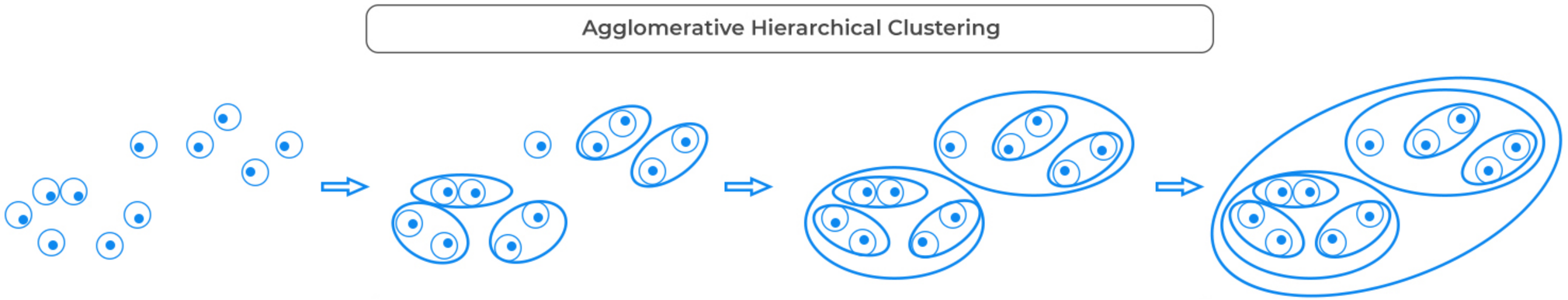
Types of Hierarchical Clustering

- 1. Divisive (top down) clustering
  - Starts with all data points in one cluster, the root, then
    - Splits the root into a set of child clusters. Each child cluster is recursively divided further
    - Stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point



Types of Hierarchical Clustering

- 2. Agglomerative (bottom up) clustering
  - The dendrogram is built from the bottom level (each data point) by
    - Merging the most similar (or nearest) pair of clusters
    - Stopping when all data points are merged into a single cluster (i.e., the root cluster)

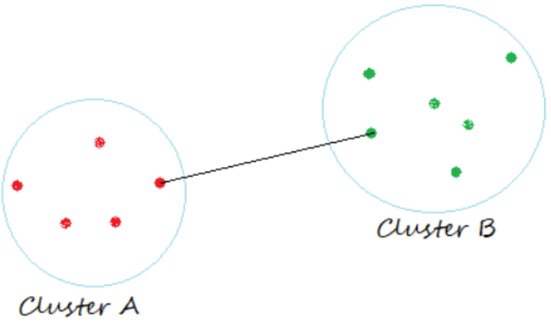


Hierarchical Clustering

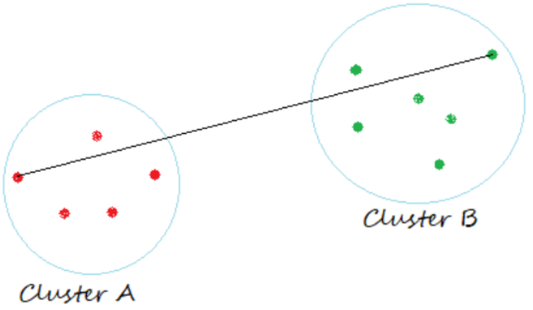
- Input of hierarchical clustering
  - distance matrix: to measure distance between single objects
  - **linkage method**: to measure distance between clusters
- Output
  - Dendrogram
    - a tree that defines the relationships between objects and the distance between clusters
    - a nested sequence of clusters

Linkage Methods

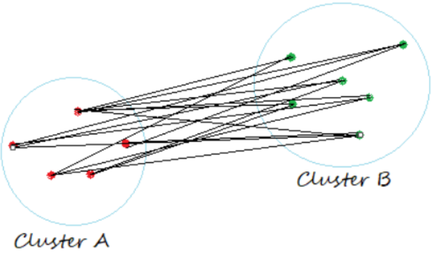
- Single Linkage: measures the closest pair of points



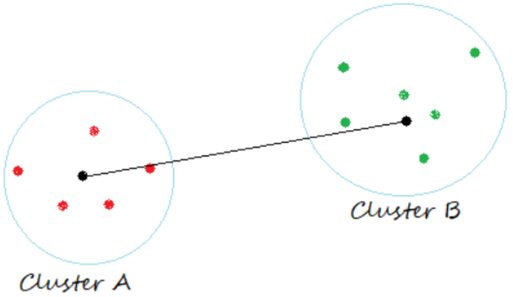
- Complete Linkage: measures the farthest pair of points



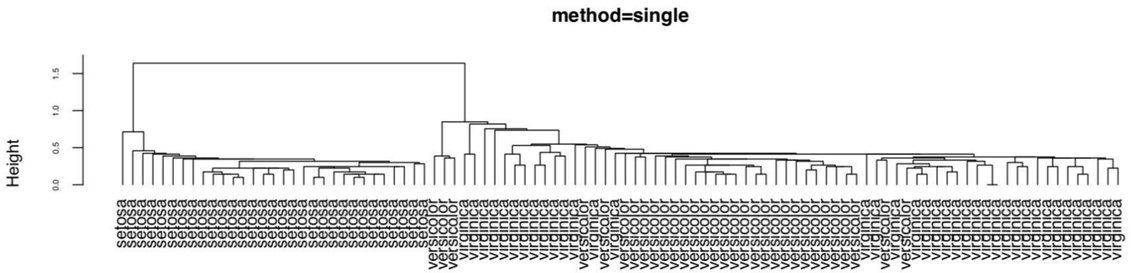
- Average Linkage: measures the average dissimilarity over all pairs

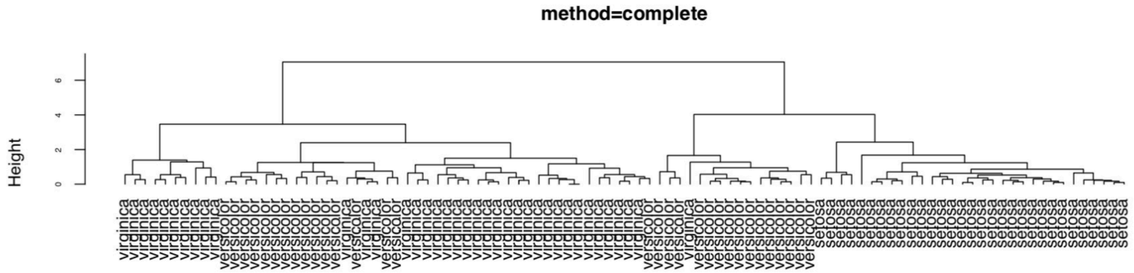


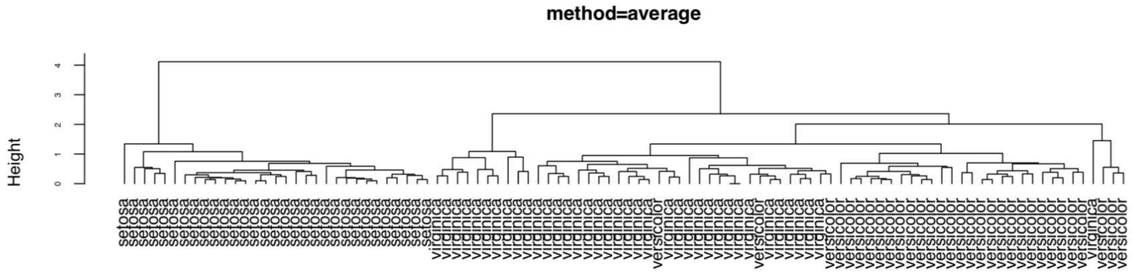
- Centroid Linkage: measures centroids

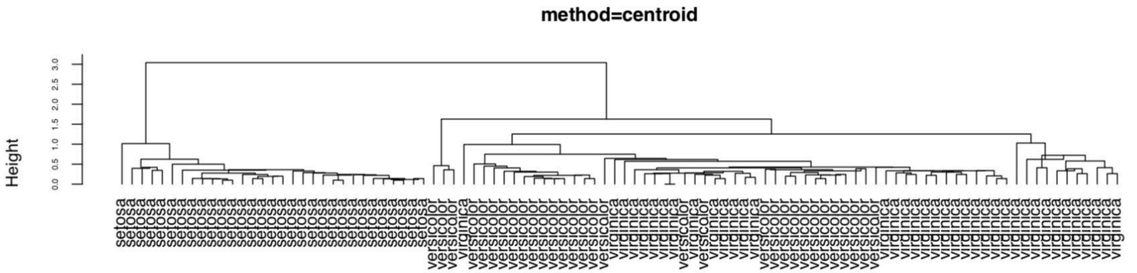


Linkage Methods









Linkage matters!

Hierarchical clustering

Advantages	**Disadvantages**
There may be small clusters nested inside large ones	Clusters might not be naturally represented by a hierarchical structure
No need to specify number groups ahead of time	Its necessary to 'cut' the dendrogram in order to produce clusters
Flexible linkage methods	Bottom up clustering can result in poor structure at the top of the tree. Early joins cannot be 'undone'