

# Bioinformatical analysis of omics expression data

## Part 3



Dr. Michael Turewicz<sup>1,2</sup>

<sup>1</sup>Institut für Klinische Biochemie und Pathobiochemie,  
Deutsches Diabetes-Zentrum, Leibniz-Zentrum für Diabetesforschung an der Heinrich-Heine-Universität, Düsseldorf, Deutschland

<sup>2</sup>Deutsches Zentrum für Diabetesforschung (DZD), München-Neuherberg, Deutschland

# Course schedule

- Part 1 (25.10.23)
  - Introduction (omics, example data, programming)
  - Data preprocessing (data inspection, normalization, missing values)
  - Exercises: R programming tutorial (part 1)
- Part 2 (08.11.23)
  - Differential expression analysis (statistics, volcano plot)
  - Exercises: R programming tutorial (part 2)
- **Part 3 (15.11.23)**
  - **Machine learning I: Clustering (clustering, PCA)**
  - **Exercises: Customized hierarchical clustering & PCA in R**
- Part 4 (22.11.23)
  - Overrepresentation analysis (GO, Reactome)
- Part 5 (29.11.23)
  - Network analysis (STRING, Cytoscape)
- Part 6 (06.12.23)
  - Machine learning II: Classification algorithms

# Review of previous part

## • Finding interesting candidates

- Omics data: thousands of measured biomolecules
- Which interesting? → Depends on study design!
- Often: candidates with differential group-specific values
  - fold change (high difference between gr. means)
  - p-value (statistically significant diff. betw. gr. means)

## • Fold change (FC)

- If A = mean of group 1 and B = mean of group 2
- Definition:  $FC = B/A$
- Often log2-FCs used

## • P-value (e.g. from t-test)

- Omics data usually log-transformed for t-test
- $H_0$  (mean equality) very unlikely for gr. means & variances of a candidate → p-value below 0.05
- Multiple testing: p-value adjustment needed

## • Visualization: volcano plot

- $\log_2(FC)$  vs.  $-\log_{10}(p\text{-value})$

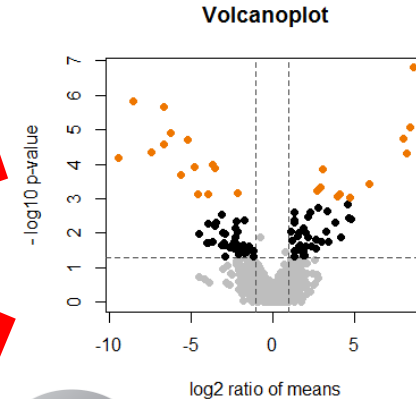
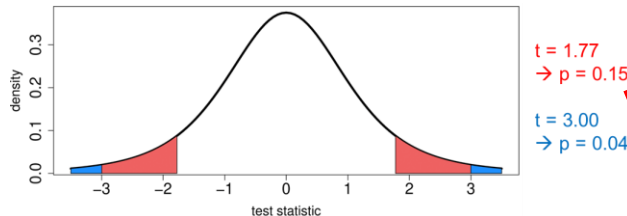
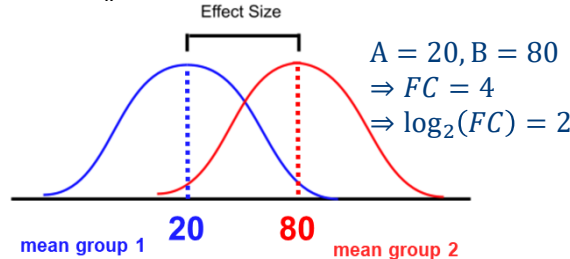
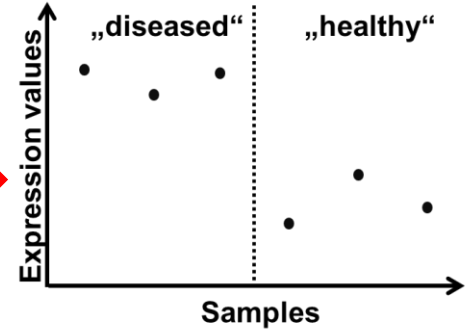
## • Programming (example: R)

- R tutorial part 2: plots in R, for-loop, volcano

Columns: p samples (p = sample number)

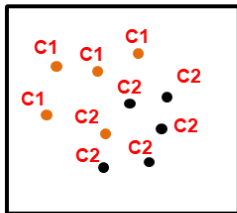
Protein IDs	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity	LFQ intensity
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A0A024GZKX.ADA	103120000	62530000	10640000	87570000	73140000	118490000	122110000	147700000	57634000	44445000										
A0A024GZKX.ADA	57040000	112420000	0	0	58015000	0	0	8130000	77254000	64144000										
A0A024GZKX.ADA	212130000	67509000	33130000	61434000	37726000	27660000	28360000	348950000	36724000											
A0A024GZKX.ADA	44839000	54076000	0	0	0	0	15540000	0	0	0										
A0A024GZKX.ADA	271330000	0	0	0	415720000	211880000	17040000	0	77600000	0										
A0A024GZKX.ADA	272030000	180710000	201930000	346480000	258450000	170690000	174750000	60781000	0	14780000										
A0A024GZKX.ADA	0	0	0	0	44555000	0	0	0	0	0										
A0A024GZKX.ADA	0	1034100000	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	91743000	0	248030000	0	0	273300000	0	0	0	0										
A0A024GZKX.ADA	0	0	103170000	0	339500000	79520000	912910000	79830000	0	41472000										
A0A024GZKX.ADA	0	50063000	0	0	24838000	0	120130000	0	0	0										
A0A024GZKX.ADA	0	0	34257000	0	0	0	0	0	0	0										
A0A024GZKX.ADA	181270000	142910000	339380000	278150000	416830000	415110000	125020000	537730000	0	217760000										
A0A024GZKX.ADA	154300000	161730000	223600000	0	236070000	270420000	0	0	0	0										
A0A024GZKX.ADA	89043000	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	20478000	61343000	107980000	137710000	112150000	130811000	412130000	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	400410000	405610000	108000000	0	115210000	125020000	178190000	0	615820000	361940000										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	146510000	94722000	156260000	35300000	85495000	111990000	174500000	85534000	0	0										
A0A024GZKX.ADA	29382000	51434000	75176000	11130000	62744000	38730000	109770000	15424000	35400000	47704000										
A0A024GZKX.ADA	57670000	0	0	0	45773000	63687000	65827000	41502000	0	58971000										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	1034200000	1366900000	1836300000	800110000	776780000	527890000	2051500000	1147500000	1184500000	980880000										
A0A024GZKX.ADA	22722000	0	0	0	74922000	72538000	41113000	0	30181000	0										
A0A024GZKX.ADA	0	0	0	0	138150000	0	0	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										
A0A024GZKX.ADA	0	0	0	0	0	0	0	0	0	0										

rows: n proteins (n = protein number)

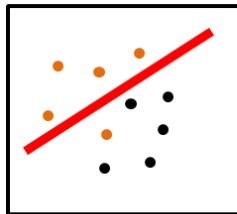


# Machine learning: tasks & example algorithms

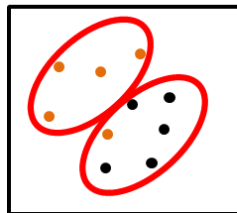
- **Classification** (output = class label): Support vector machines, decision trees, neural networks...



- **Regression** (output = (continuous) numbers): Support vector regression, regression trees...



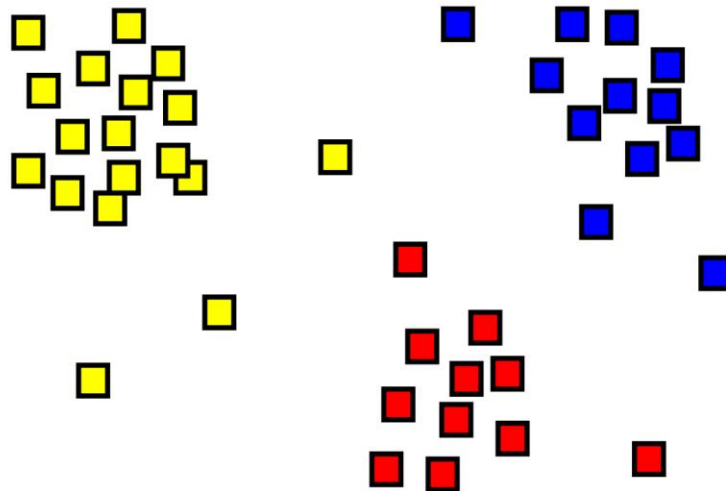
- **Clustering** (output = clusters of data points): k-Means, hierarchical clustering, ...



→ Discussed today!

# What is clustering?

- ▶ Clustering is the grouping of objects into groups (= clusters)...
- ▶ in a way that all objects inside a specific cluster are more similar to each other than to objects in all other clusters.

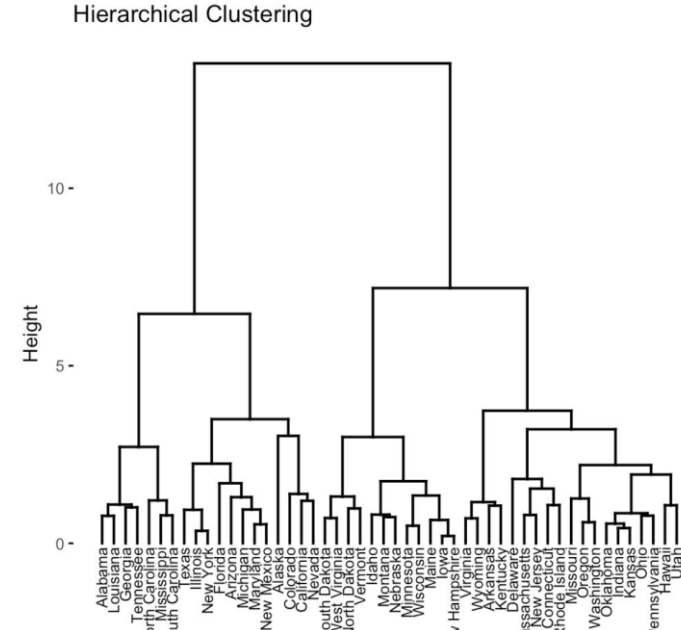


# Clustering algorithms

- ▶ There are many published clustering algorithms & algorithm variants.
- ▶ They can be categorized into four major groups of algorithms:
  - ▶ Hierarchical clustering (e.g., single linkage, complete linkage, average linkage)
  - ▶ Centroid-based clustering (e.g., k-means)
  - ▶ Distribution-based clustering (e.g., EM clustering)
  - ▶ Density-based clustering (e.g., DB SCAN)
- ▶ In this course only hierarchical clustering will be discussed.

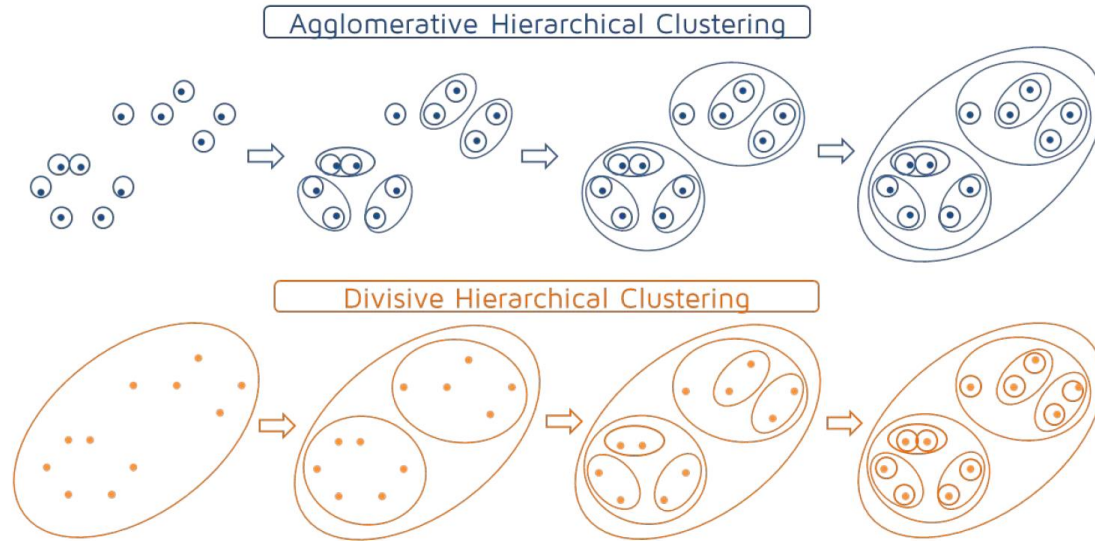
# Hierarchical clustering (HC): main idea & dendrograms

1. Each data point starts as its own cluster.
2. The distance between all pairs of clusters is computed.
3. Then the closest clusters are merged.
4. Steps 2. - 3. are repeated until all clusters are merged into a single cluster.



# HC: agglomerative vs. divisive approach

Note: In this course only agglomerative methods are discussed.





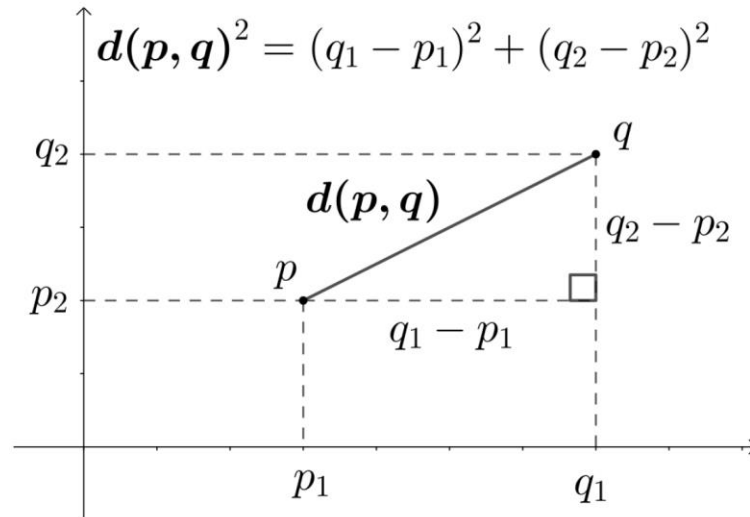
# HC: distance functions

- ▶ Distance functions compute distances between pairs of vectors.
- ▶ Distances between pairs of vectors are necessary to obtain distances between clusters.
- ▶ In this course, we discuss euclidean, manhattan & correlation-based distances.

# HC: euclidean distance

- ▶ The distance from classical geometry in school.

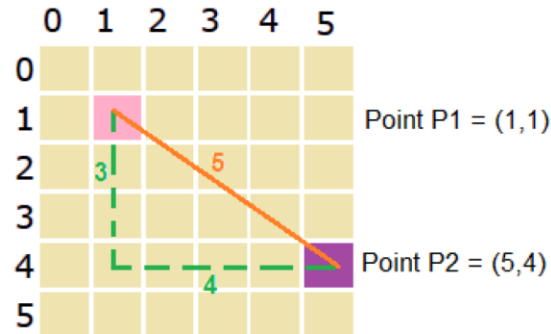
- ▶  $d(p, q) := \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$



# HC: Manhattan distance

- ▶ Also called taxicab distance or city block distance.

- ▶ 
$$d(a, b) := \sum_{i=1}^n |a_i - b_i|$$



Euclidean distance =  $\sqrt{(5-1)^2 + (4-1)^2} = 5$

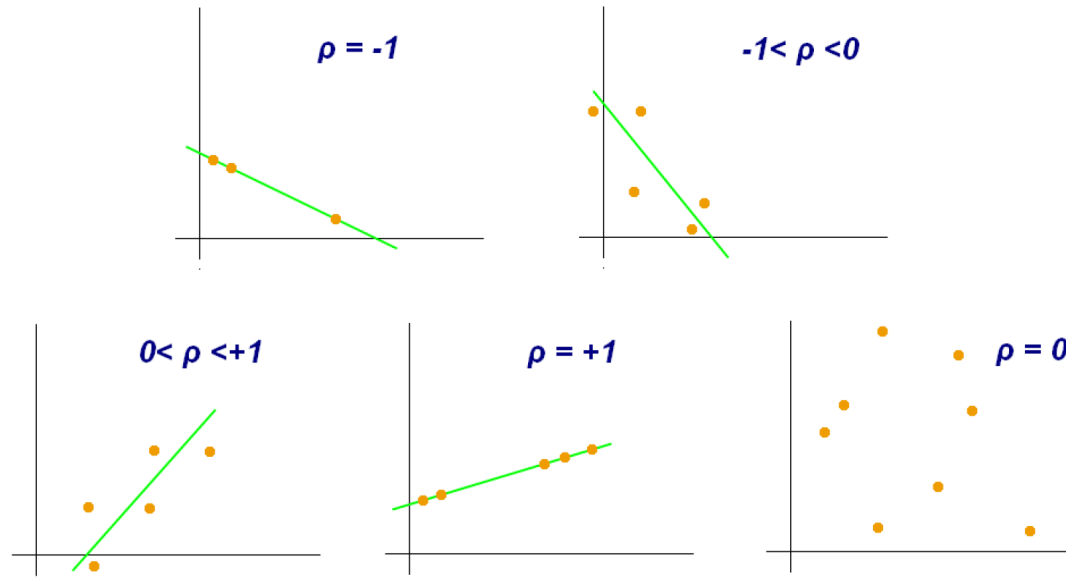
Manhattan distance =  $|5-1| + |4-1| = 7$

**DDZ**  
Deutsches Diabetes-Zentrum



# HC: correlation-based distance

- ▶ Using correlation as distance measure.
- ▶  $d(a, b) := (1 - \text{cor}(a, b))/2$ , where  $\text{cor}(a, b)$  is Pearson's correlation coefficient between vectors  $a$  and  $b$ .

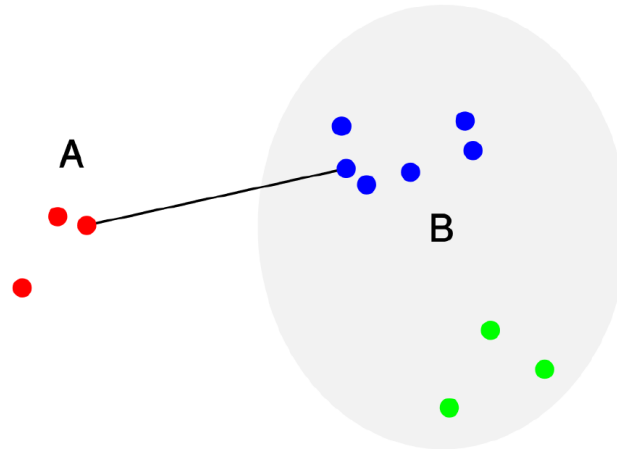


# HC: linkage methods

- ▶ Linkage methods compute the distance between clusters.
- ▶ To this end, they use the distance methods between single cluster elements (e.g., euclidean).
- ▶ They are crucial for the decision which clusters should be merged.
- ▶ In this course we will discuss: Single, complete and average linkage.

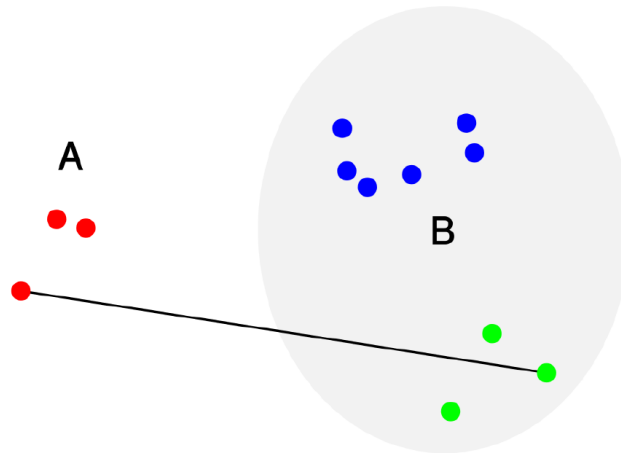
# HC: single linkage

- ▶ Minimal distance between all mixed pairs from both clusters.
- ▶  $D_{single} := \min_{a \in A, b \in B} \{d(a, b)\}.$



# HC: complete linkage

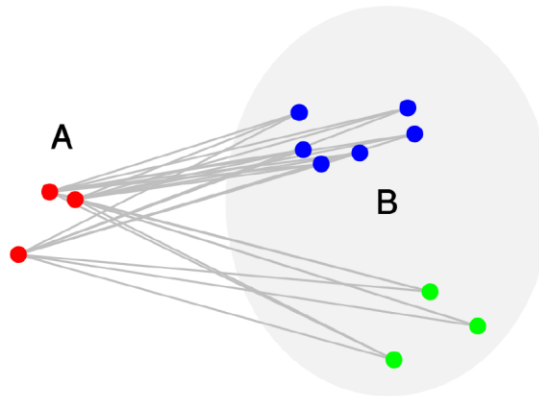
- ▶ Maximal distance between all mixed pairs from both clusters.
- ▶  $D_{complete} := \max_{a \in A, b \in B} \{d(a, b)\}$ .





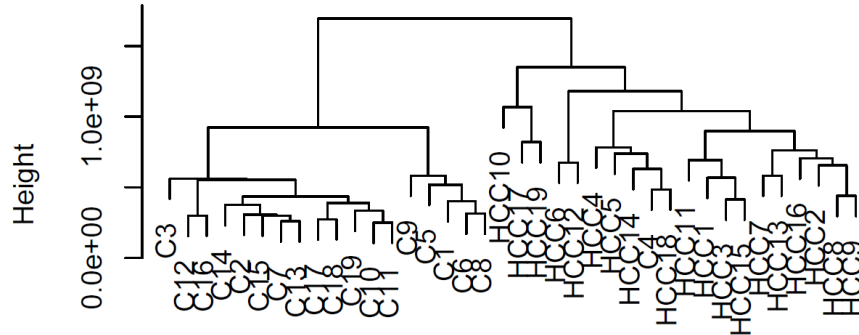
# HC: average linkage

- ▶ Unweighted pair group method with arithmetic mean (UPGMA).
- ▶ Average distance between all mixed pairs from both clusters.
- ▶  $D_{average} := \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a, b).$



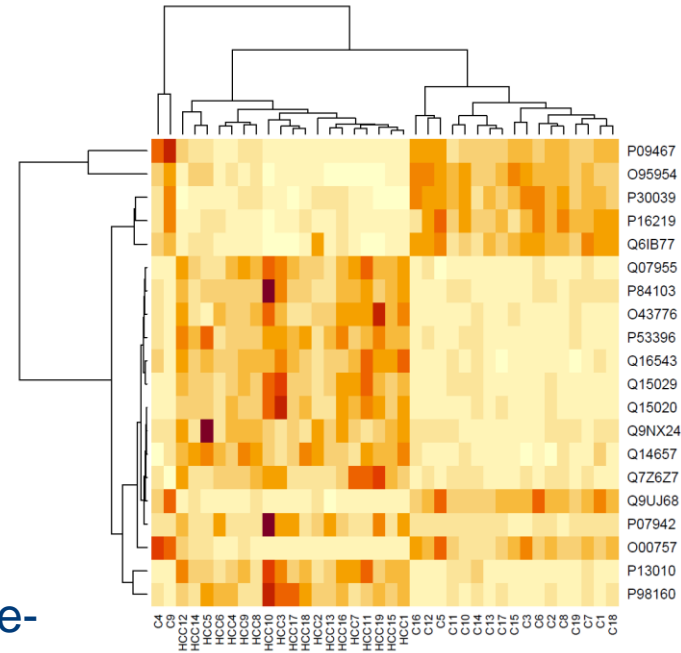
# HC: visualization

**Dendrogram**



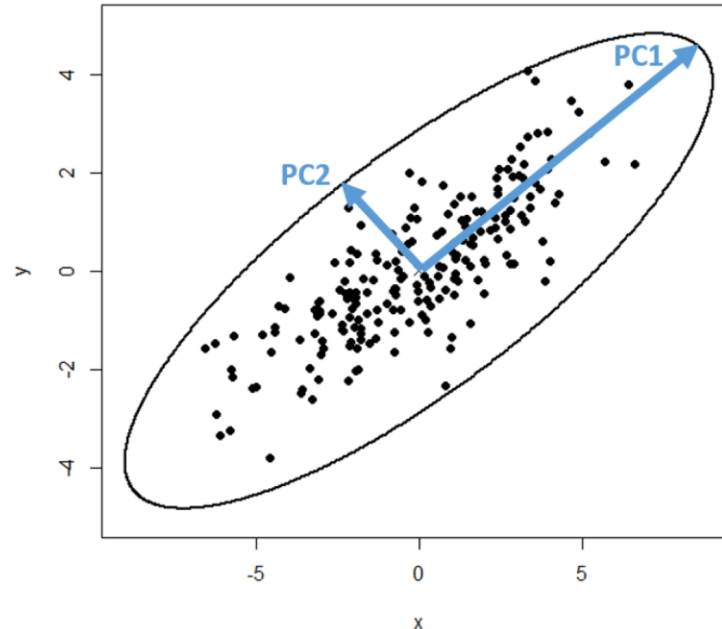
- Heat maps provide a more informative visualization,
- Dendrograms for both clustering can be shown including expression values: Sample-wise and biomolecule-wise clustering.

**Heat map**



# Principal component analysis (PCA)

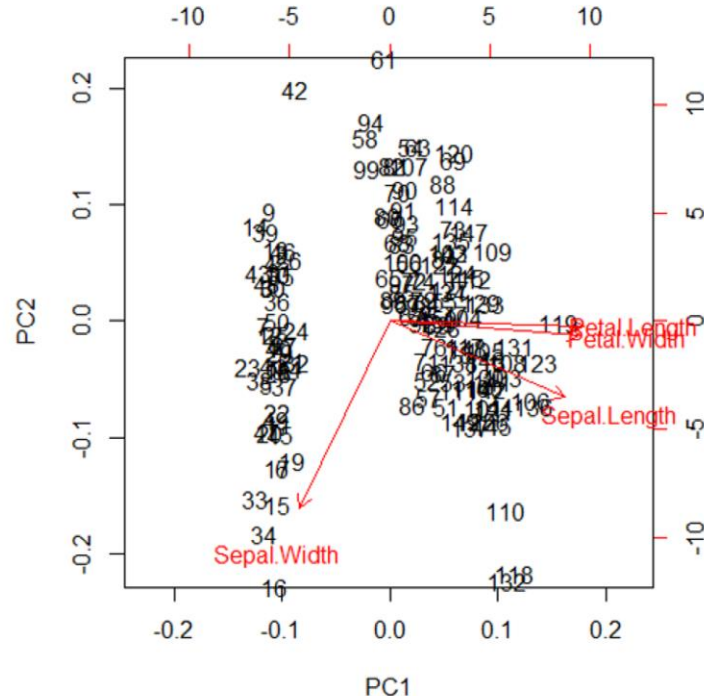
- ▶ transform  $n$  variables ( $X_1, \dots, X_n$ ) into  $n$  Principal Components
- ▶ Principal Components (PCs) are linear combinations of the original variables:  $PC = w_1X_1 + w_2X_2 + \dots + w_nX_n$
- ▶ PC1 has the largest possible variance, PC2 the largest variance and of vectors orthogonal to PC1, ...
- ▶ 2-dimensional PCA-Plots as an overview over the whole data set



# PCA biplot

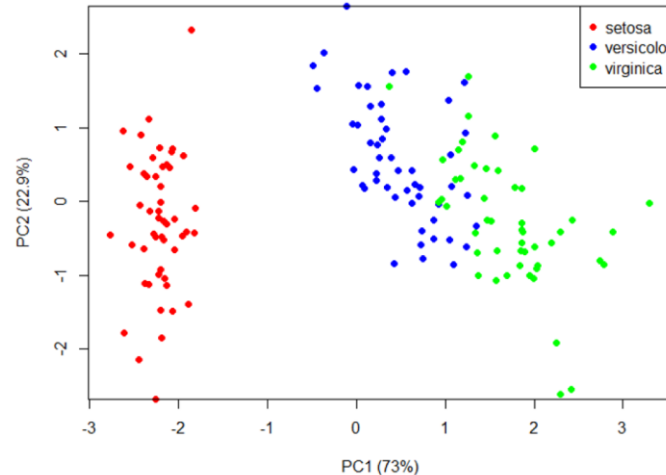
```
pca <- prcomp(iris[,1:4],  
scale. = TRUE)  
biplot(pca)
```

- ▶ shows first and second PC of observations as a scatterplot
- ▶ direction of original axes (variables)
- ▶ not suitable for proteomics data (too many variables)



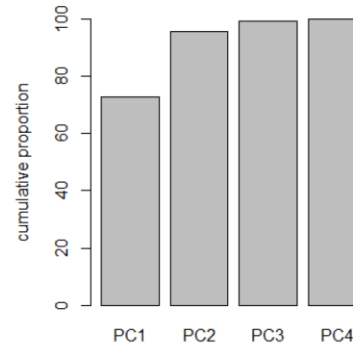
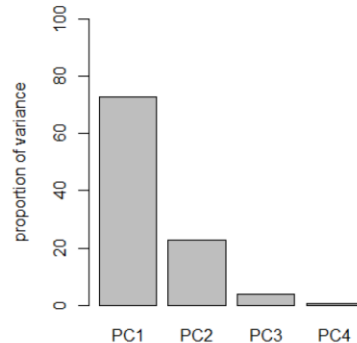
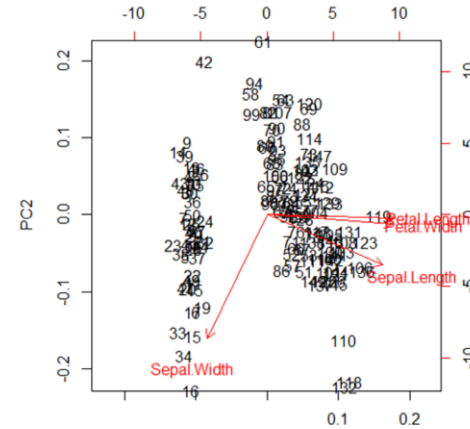
# PCA plot

```
summ <- summary(pca)
plot(pca$x[,1], pca$x[,2], pch = 16,
     col = rep(c("red", "blue", "green"), each = 50),
     xlab = paste0("PC1 (", round(100*summ$importance[2,1], 1), "%)"),
     ylab = paste0("PC2 (", round(100*summ$importance[2,2], 1), "%)"))
legend("topright", col = c("red", "blue", "green"), pch = 16,
      legend = levels(iris$Species))
```



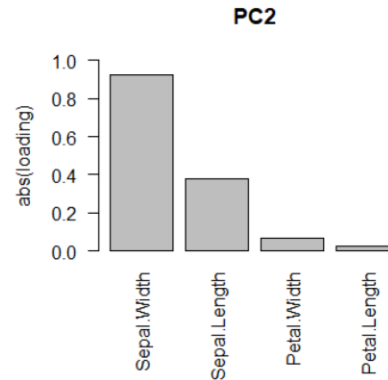
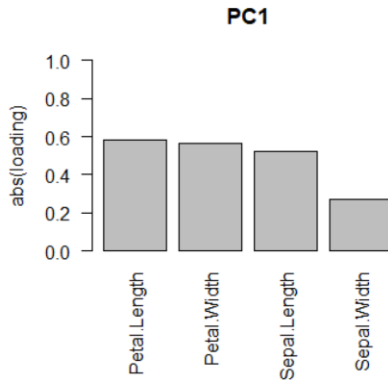
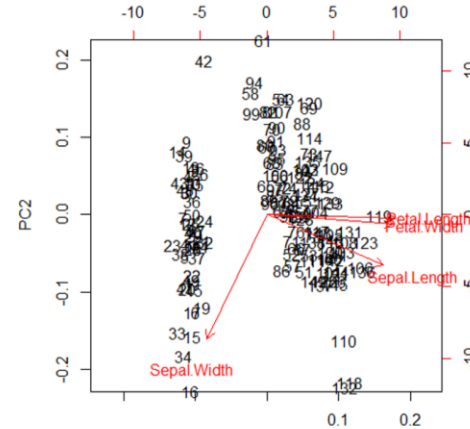
# PCA: explained variance

- ▶ each PC explains a portion of the total variance in the dataset (PC1 the most, PC2 the 2nd most, ...)
- ▶ the higher the variance in PC1 and PC2 combined, the more complete is the overview over the dataset in the 2D plot



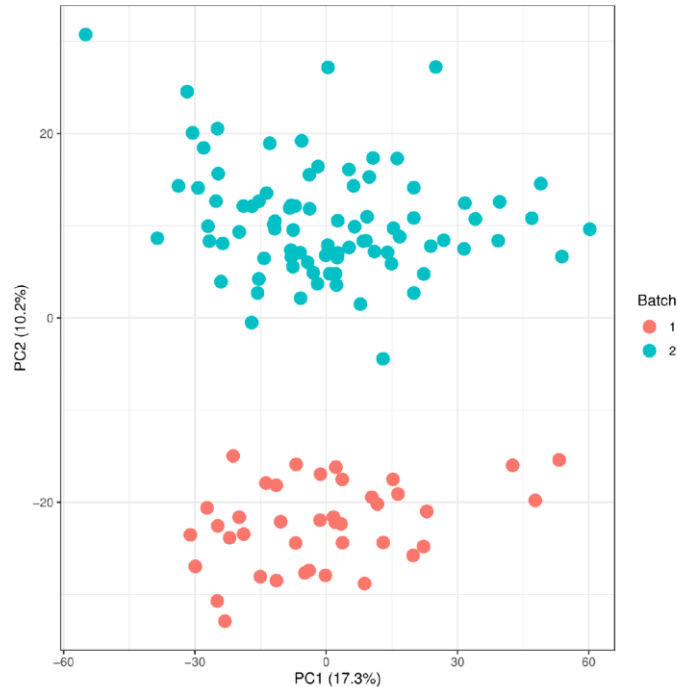
# PCA: loadings

- ▶ Loadings are the weights used in the linear combinations
- ▶ show how important a certain variable is for a certain PC
- ▶ called rotation by prcomp



# PCA for quality control

- ▶ overview over whole data set
- ▶ detection of outlier points
- ▶ detection of batch effects





# Hands on part!

# Exercises

- **Exercise 3**

- <https://drive.google.com/drive/folders/1vmewprs0gkpakU8idbgtexDIwmGVUJz3?usp=sharing>
- Use our example dataset from GitHub for the following exercises
- **Exercise 3.1:** Perform hierarchical clustering of samples using only differential features and generate dendrograms with 6 different combinations of linkage (single, complete, average) and distance (Euclidean, Manhattan) methods.
- **Exercise 3.2:** Visualize the above method combinations for hierarchical clustering with heat maps.
- **Exercise 3.3:** Perform & visualize PCA of samples using only differential features. Represent different groups with different colors and different individuals (assumption: Basal1 & Insulin1 are same individual) with different symbols.
- Please send me your solutions as an “.R”-file

# Thank you!