Bioinformatical analysis of omics expression data Part 3

Dr. Michael Turewicz[1,2]

[1]Institut für Klinische Biochemie und Pathobiochemie,
Deutsches Diabetes-Zentrum, Leibniz-Zentrum für Diabetesforschung an der Heinrich-Heine-Universität, Düsseldorf, Deutschland
[2]Deutsches Zentrum für Diabetesforschung (DZD), München-Neuherberg, Deutschland

Course schedule

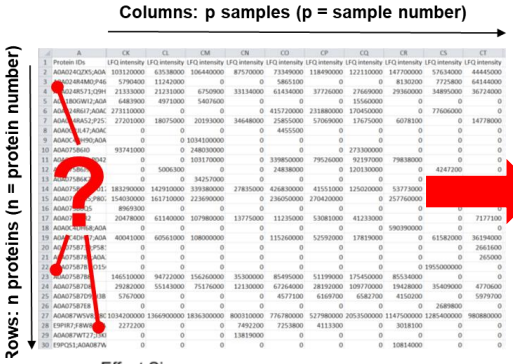
DDZ Deutsches Diabetes-Zentrum

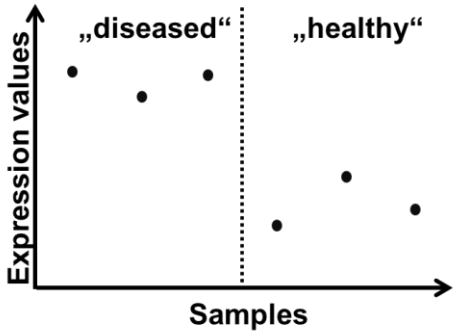
- Part 1 (25.10.23)
  - Introduction (omics, example data, programming)
  - Data preprocessing (data inspection, normalization, missing values)
  - Exercises: R programming tutorial (part 1)
- Part 2 (08.11.23)
  - Differential expression analysis (statistics, volcano plot)
  - Exercises: R programming 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

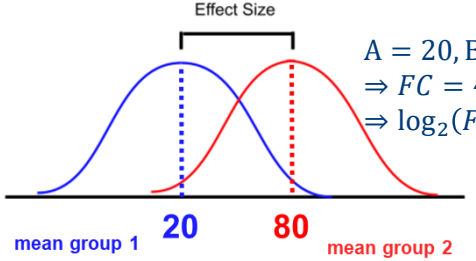
Recap of previous part

Deutsches Diabetes-Zentrum

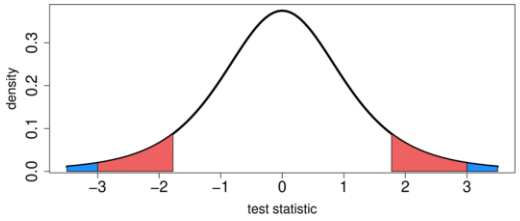
- **Finding interesting candidates**
  - Omics data: thousands of measured biomolecules
  - Which interesting? → Depends on study design!
  - Often: candidates with differential group-specific values
    - fold change (high difference between gr. means)
    - p-value (statistically significant diff. 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**
  - log2(FC) vs. –log10(p-value)
- **Programming (example: R)**
  - R tutorial part 2: plots in R, for-loop, volcano





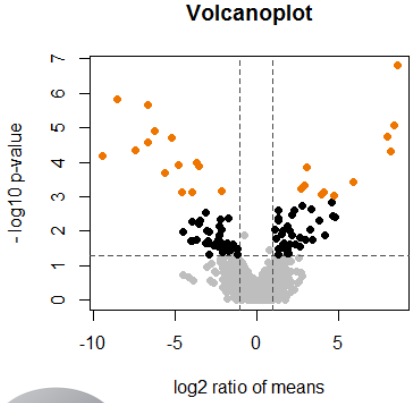


$A = 20, B = 80$
$\Rightarrow FC = 4$
$\Rightarrow \log_2(FC) = 2$



t = 1.77
→ p = 0.15

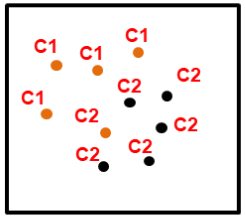
t = 3.00
→ p = 0.04



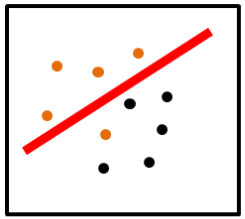
Machine learning: tasks & example algorithms

DDZ Deutsches Diabetes-Zentrum

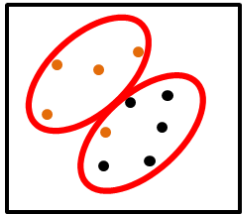
- **Classification** (output = class label): Support vector machines, decision trees, neural networks…



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



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

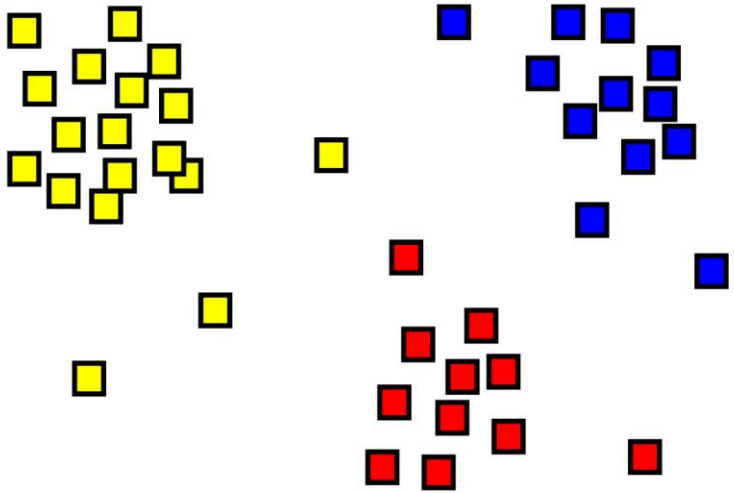


→ **Discussed today!**

What is clustering?

DDZ
Deutsches Diabetes-Zentrum

- 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.



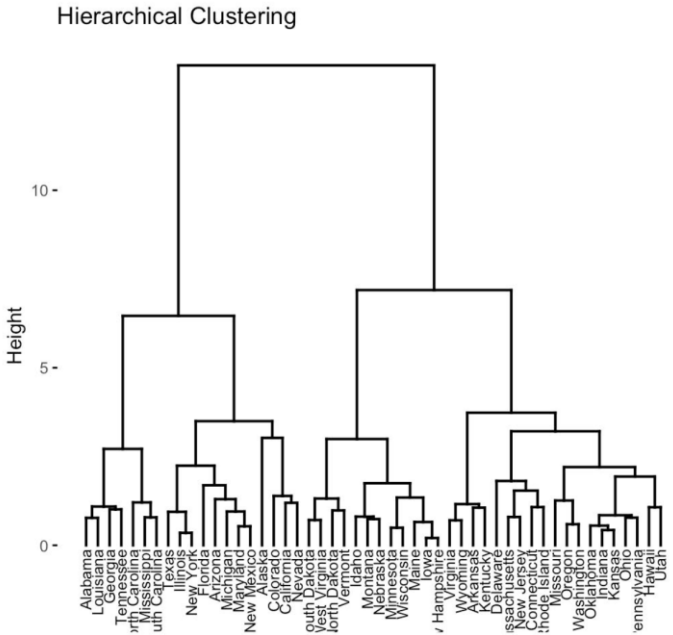
Clustering algorithms

- 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

DDZ Deutsches Diabetes-Zentrum

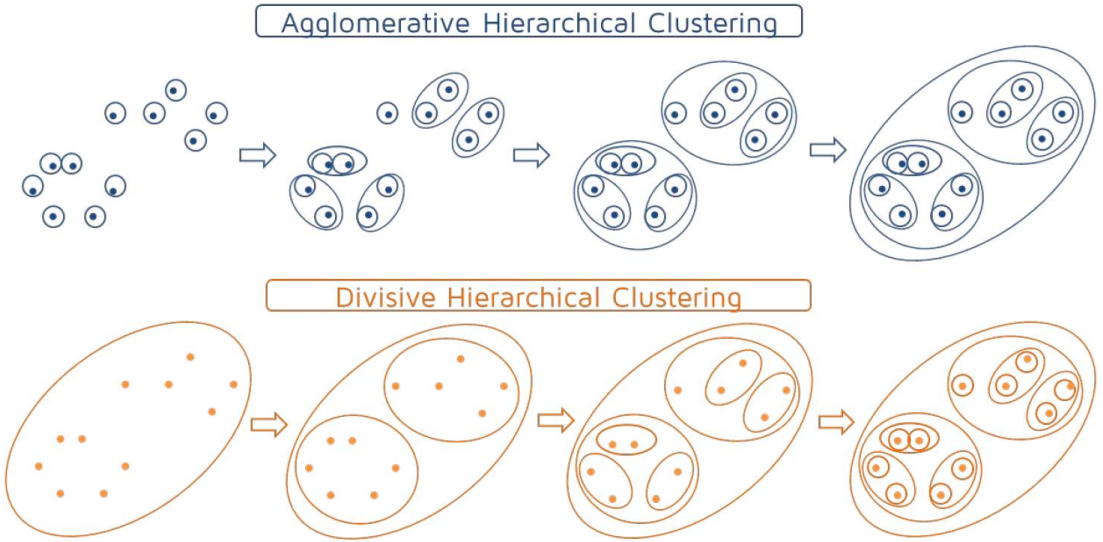
1. Each data point starts as its own cluster.
2. The distance between all pairs of clusters is computed.
3. Then the closest clusters are merged.
4. Steps 2. - 3. are repeated until all clusters are merged into a single cluster.



HC: agglomerative vs. divisive approach

DDZ
Deutsches Diabetes-Zentrum

Note: In this course only agglomerative methods are discussed.



HC: distance functions

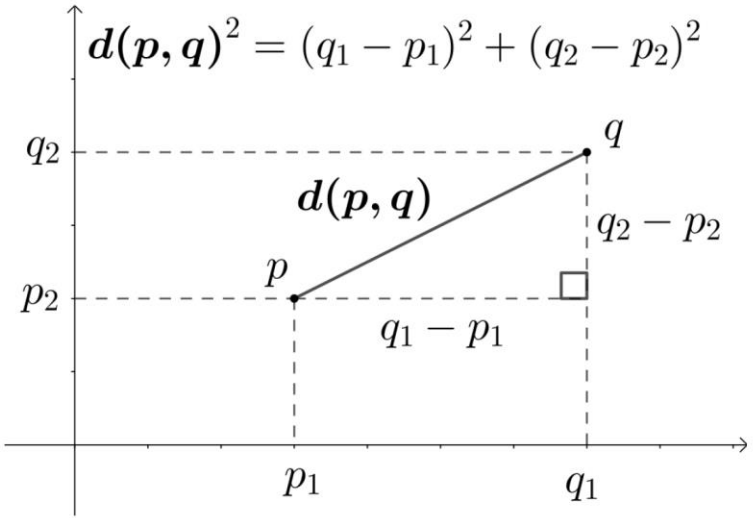
DDZ
Deutsches Diabetes-Zentrum

- 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

DDZ Deutsches Diabetes-Zentrum

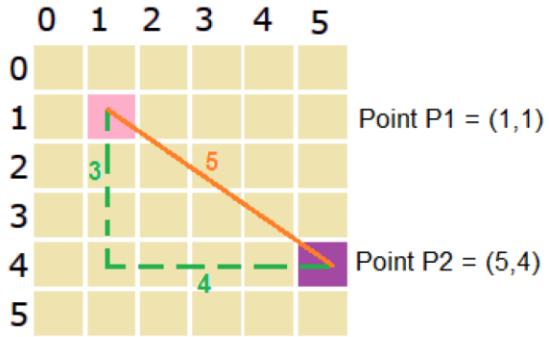
- The distance from classical geometry in school.
- $d(p,q) := \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$



HC: Manhattan distance

DDZ Deutsches Diabetes-Zentrum

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

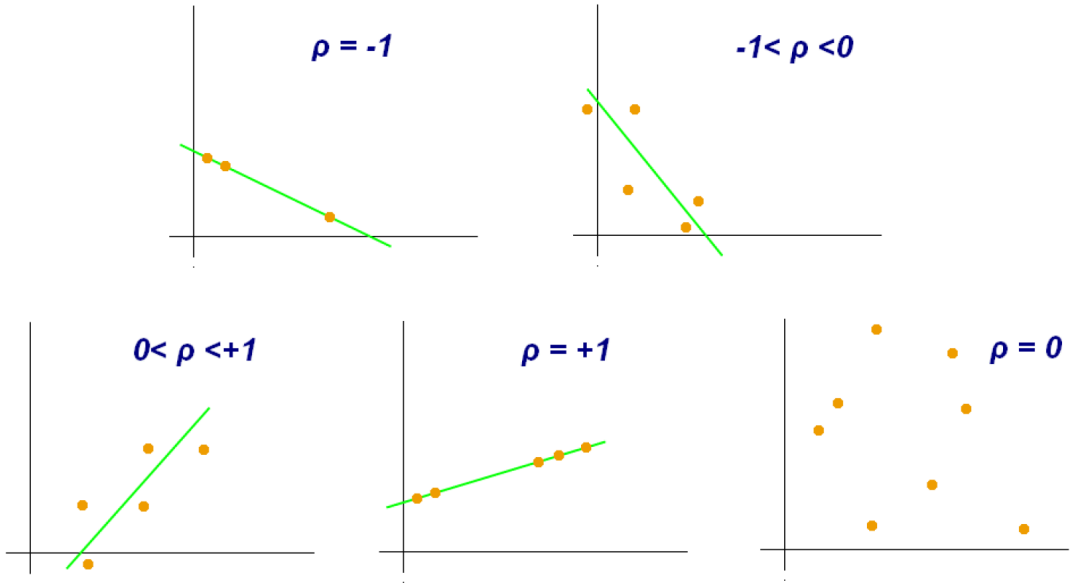
HC: euclidean vs. Manhattan

DDZ
Deutsches Diabetes-Zentrum

HC: correlation-based distance

DDZ Deutsches Diabetes-Zentrum

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

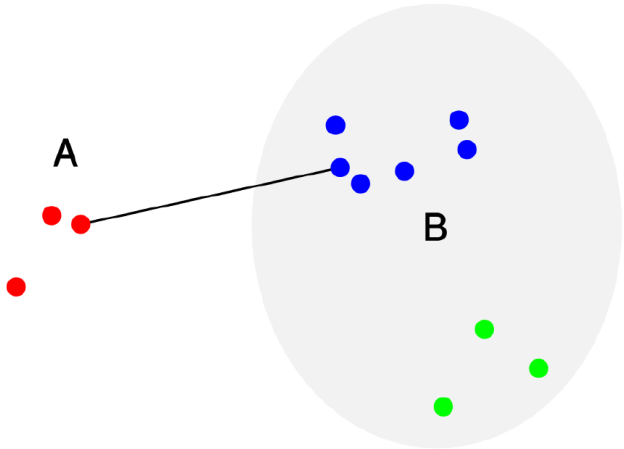


HC: linkage methods

- 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.

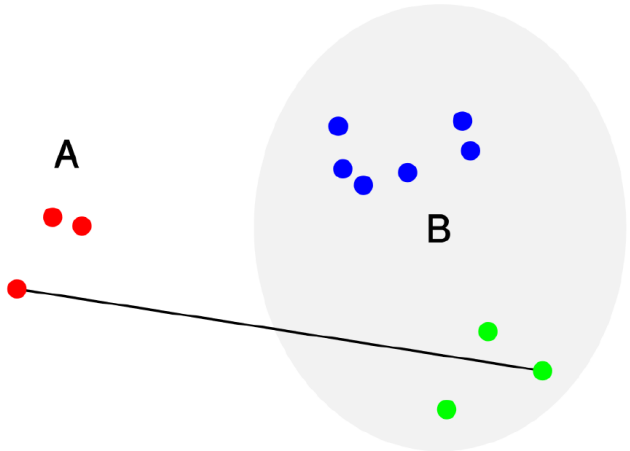
HC: single linkage

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



HC: complete linkage

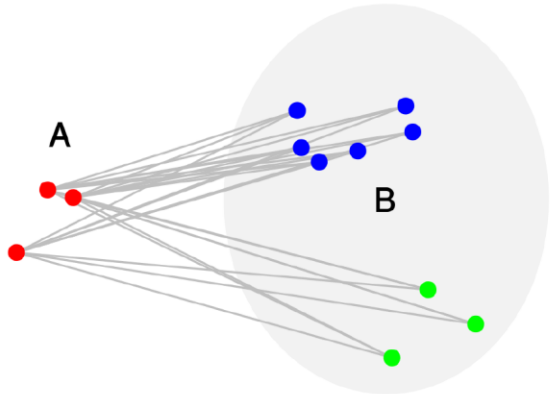
- Maximal distance between all mixed pairs from both clusters.
- $D_{complete} := \max_{a \in A, b \in B} \{d(a, b)\}$.



HC: average linkage

DDZ Deutsches Diabetes-Zentrum

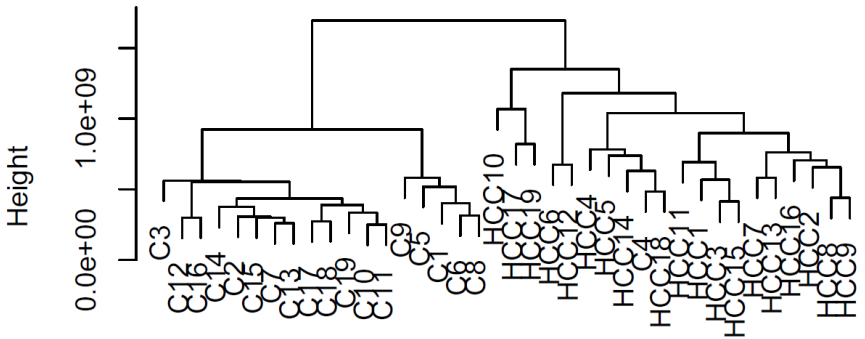
- Unweighted pair group method with arithmetic mean (UPGMA).
- Average distance between all mixed pairs from both clusters.
- $D_{average} := \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$.



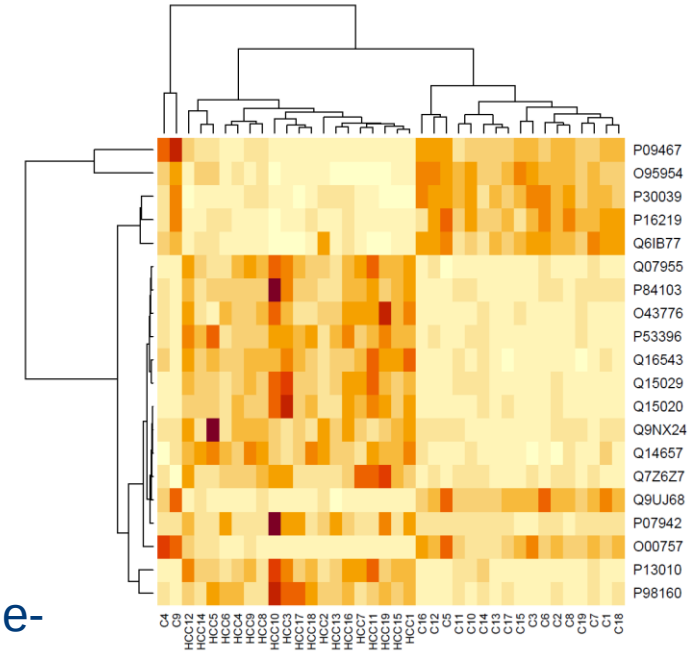
HC: visualization

Deutsches Diabetes-Zentrum

Dendrogram



Heat map



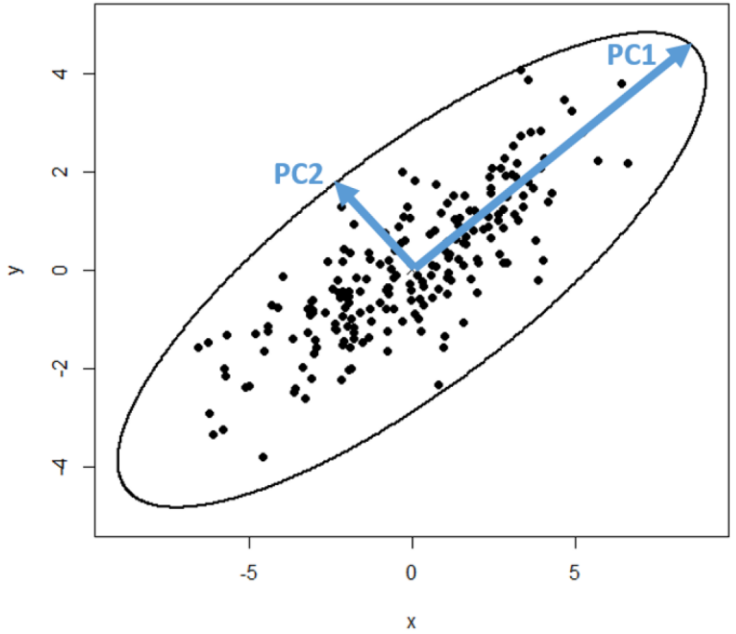
- Heat maps provide a more informative visualization,
- Dendrograms for both clustering can be shown including expression values: Sample-wise and biomolecule-wise clustering.

Principal component analysis (PCA)

DDZ Deutsches Diabetes-Zentrum

- transform n variables $(X_1, ..., X_n)$ into n Principal Components
- Principal Components (PCs) are linear combinations of the original variables: $PC = w_1X_1 + w_2X_2 + ... + w_nX_n$

- 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

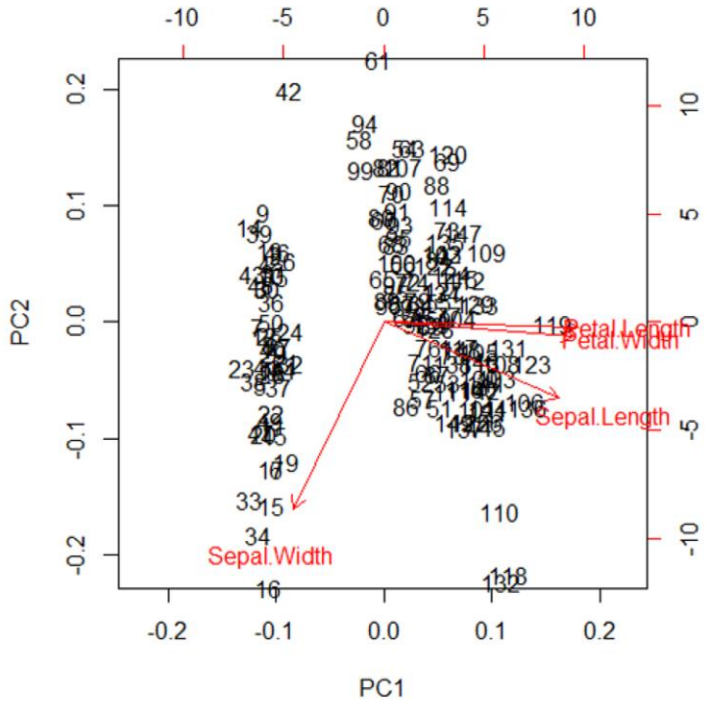


PCA biplot

DDZ Deutsches Diabetes-Zentrum

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

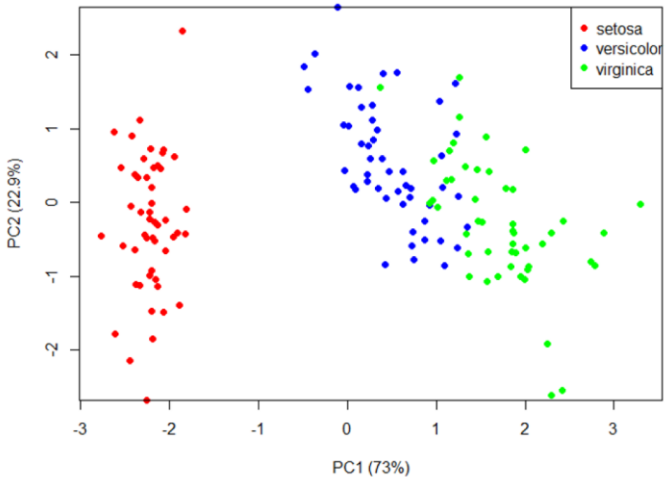
- shows first and second PC of observations as a scatterplot
- direction of original axes (variables)
- not suitable for proteomics data (too many variables)



PCA plot

Deutsches Diabetes-Zentrum

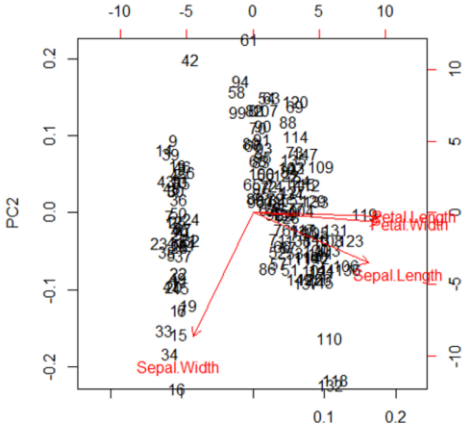
```
summ <- summary(pca)
plot(pca$x[,1], pca$x[,2], pch = 16,
  col = rep(c("red", "blue", "green"), each = 50),
  xlab = paste0("PC1 (", round(100*summ$importance[2,1], 1), "%)"),
  ylab = paste0("PC2 (", round(100*summ$importance[2,2], 1), "%)"))
legend("topright", col = c("red", "blue", "green"),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

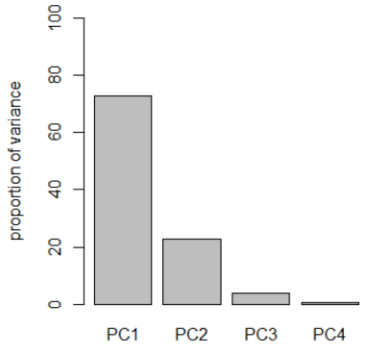


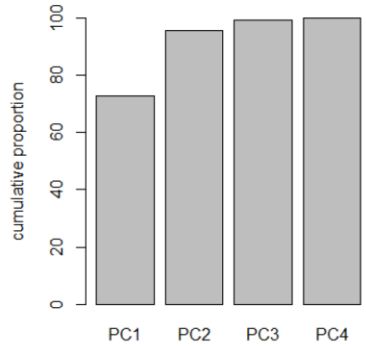
PCA: explained variance

DDZ Deutsches Diabetes-Zentrum

- 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



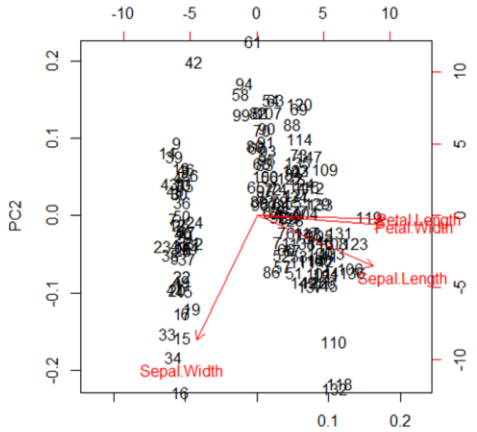


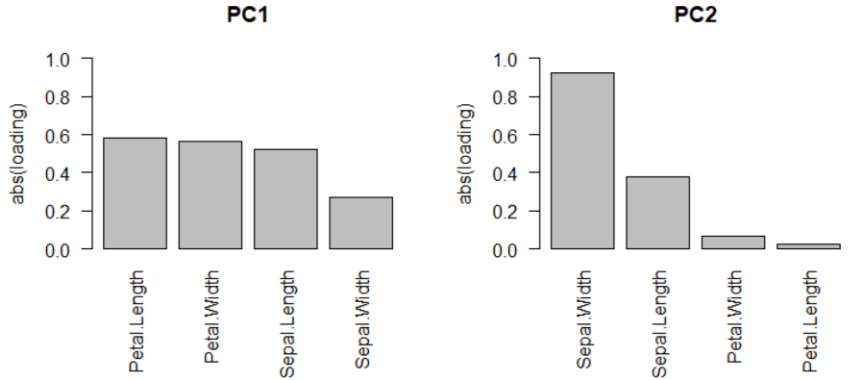


PCA: loadings

DDZ
Deutsches Diabetes-Zentrum

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

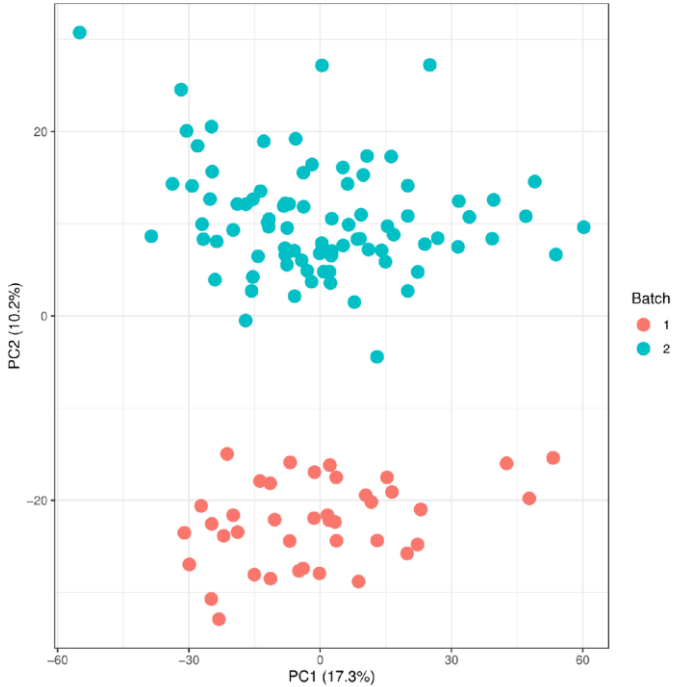




PCA for quality control

DDZ Deutsches Diabetes-Zentrum

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



Hands on part!

Exercises

DDZ Deutsches Diabetes-Zentrum

- **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!