Freie Universität Berlin

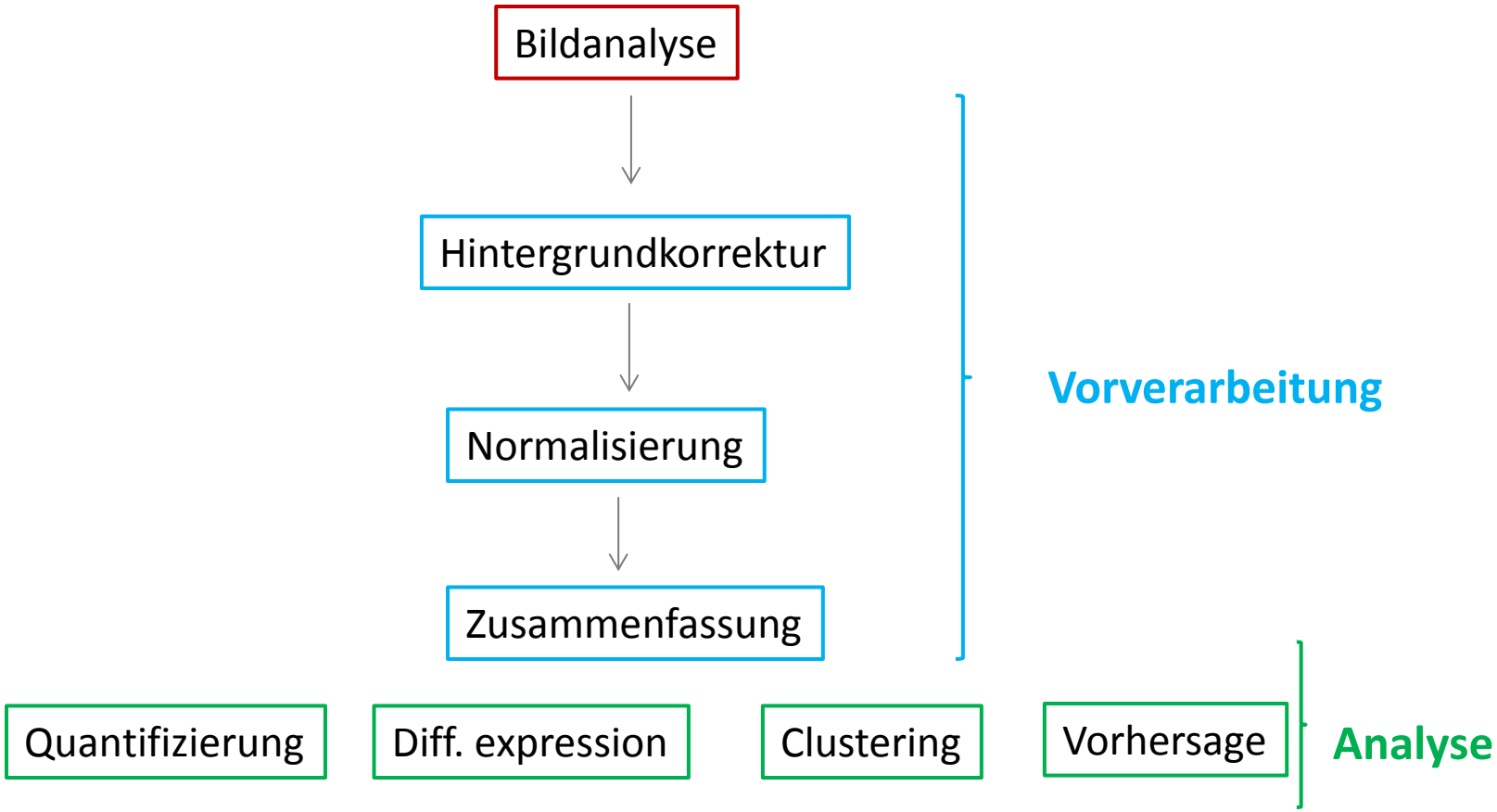
MAX-PLANCK-GESELLSCHAFT

AlgoBio WS 16/17
Clustering of gene expression data

Annalisa Marsico

09.01.2017

Pipeline für die Mikroarray-Analyse



Clustering

Clustering organisiert Datenpunkte, welche ‘nah’ sind, in Gruppen

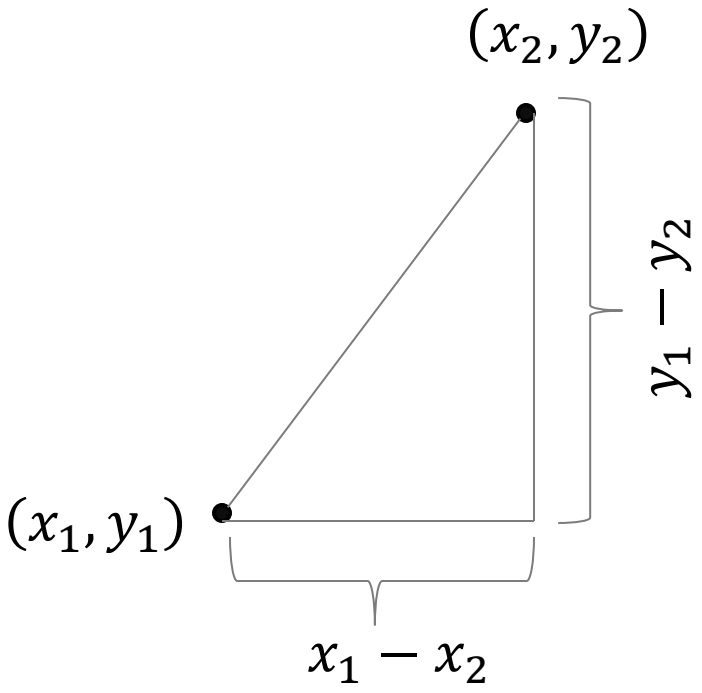
Was bedeutet es für Gene oder Proben ‚nah‘ zu sein?

Wir müssen ‘nah’ definieren -> Wir müssen ein Abstandsmaß definieren

Zwei Gene sind nahe, wenn sie eine ähnliche Expression über Proben haben

Zwei Proben sind nahe, wenn ihre Genexpressionsprofile ähnlich sind

Euklidischer Distanz

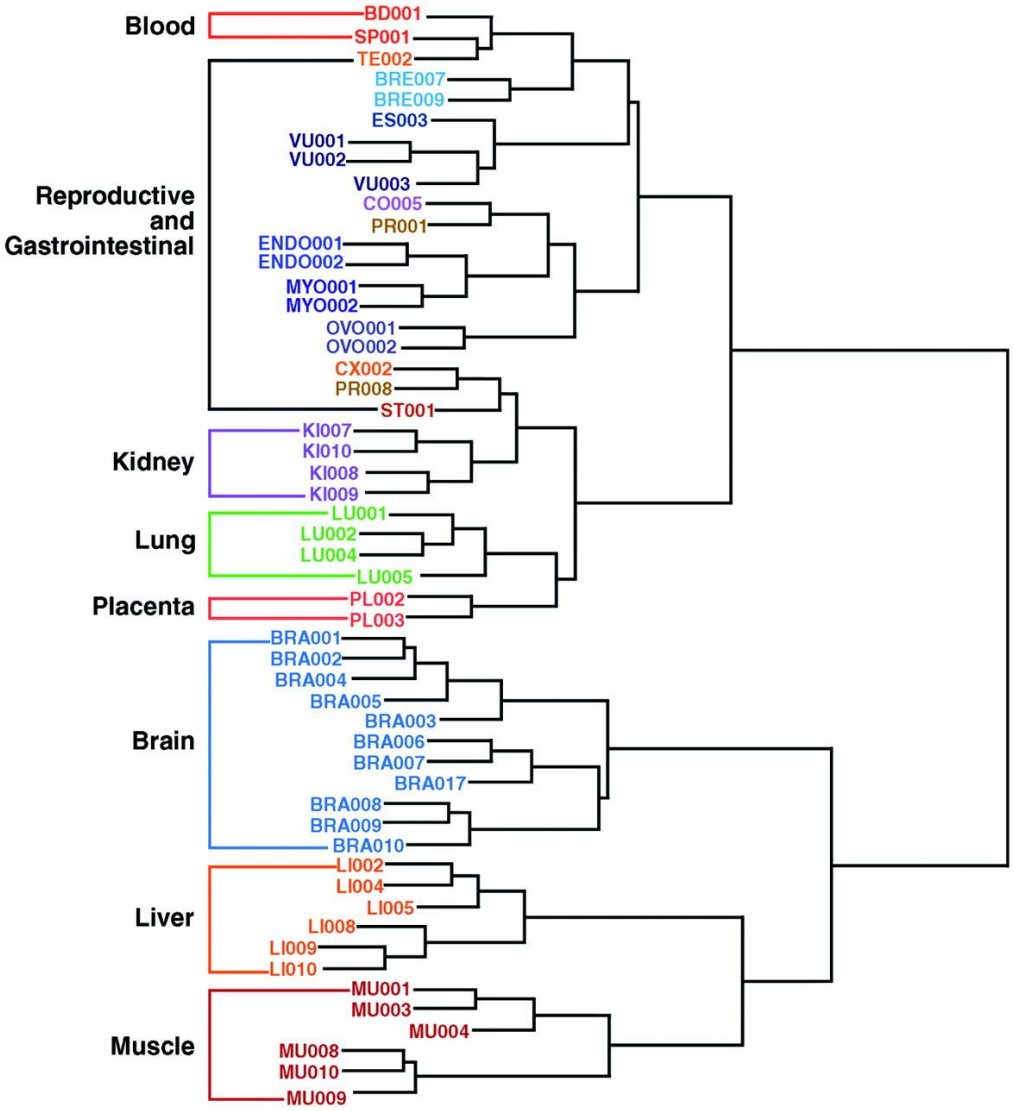


$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

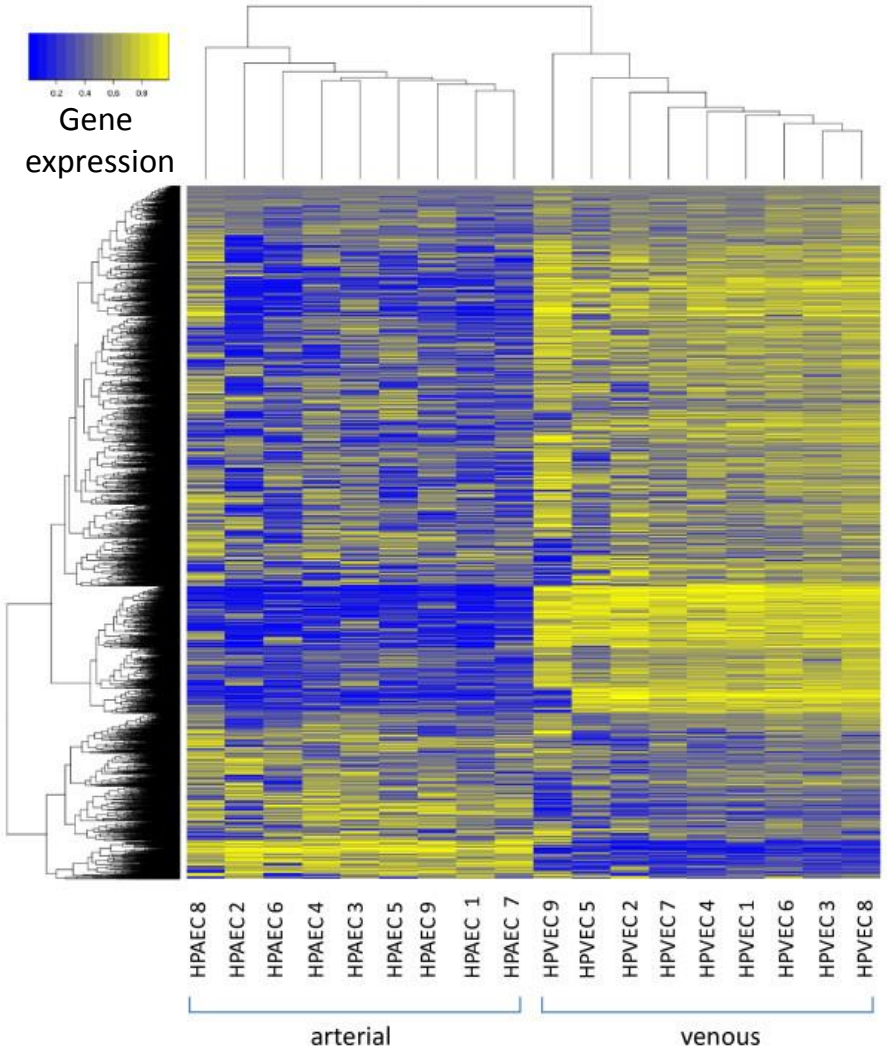
In einem zweidimensionalen Raum

Wenn wir die Distanz zwischen zwei Proben auf 20000 Genen basierend berechnen wollen, dann müssen wir die Formel auf über 20000 Dimensionen erweitern.

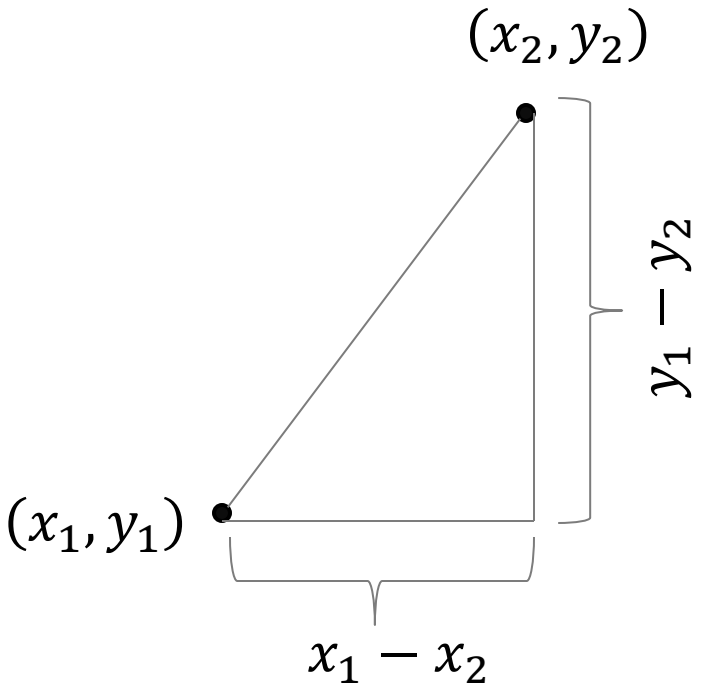
Clustering von Proben auf Genexpression basierend



Heatmap - Beispiel



Manhattan Distanz



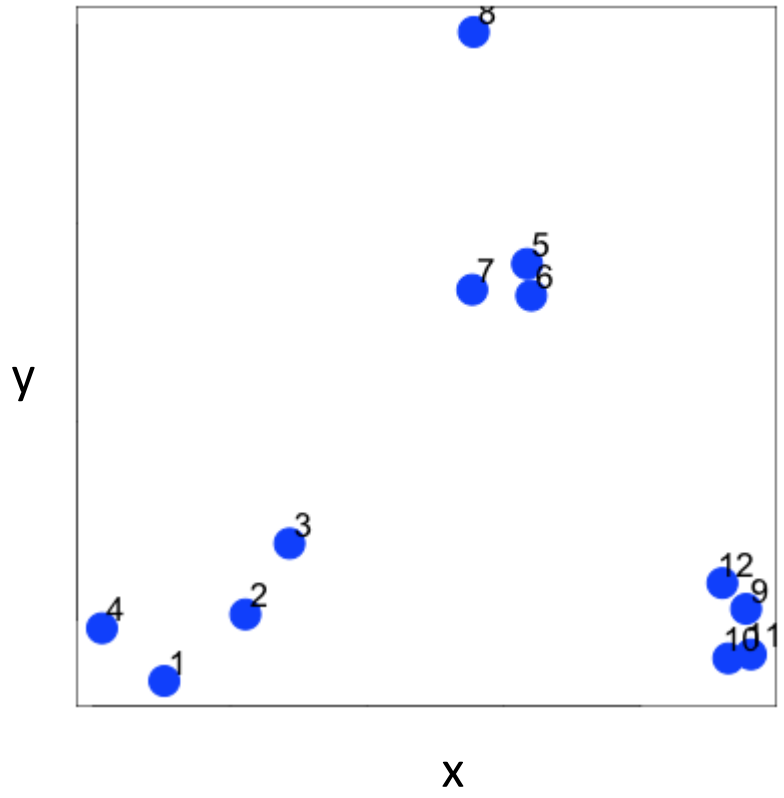
$$d = |x_1 - x_2| + |y_1 - y_2|$$

In einem zweidimensionalen Raum

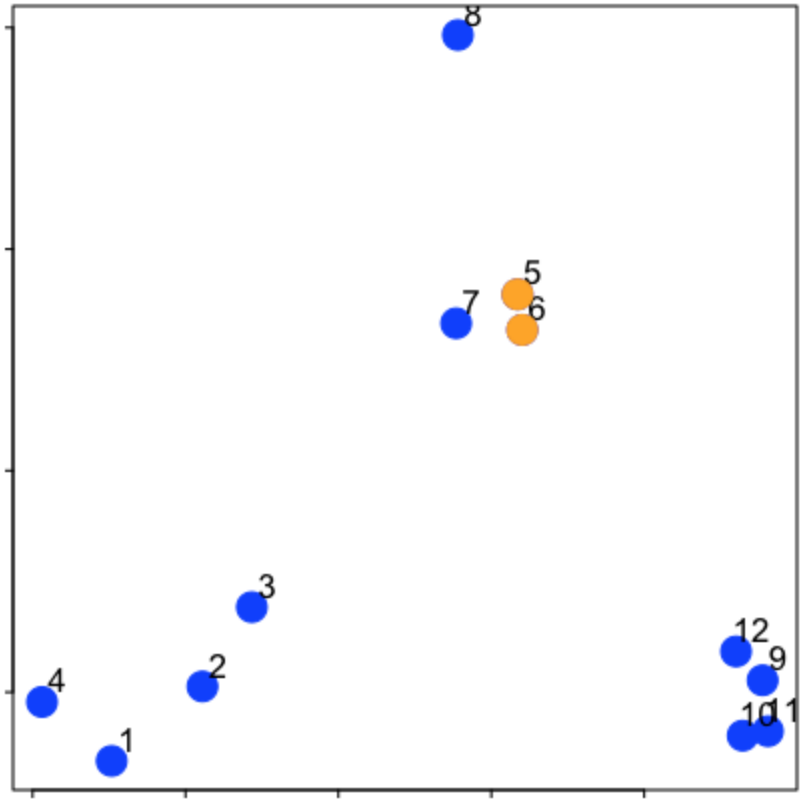
Sie ist robuster als die euklidische Distanz

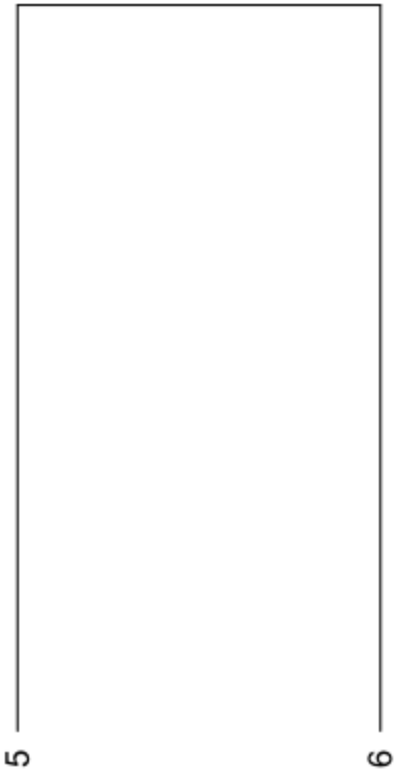
Hierarchisches Clustering

- Finde die "nächstgelegenen" Punkte (kleinste Euklidische Distanz)
- Vereinigen
- Wiederholen



Hierarchisches Clustering

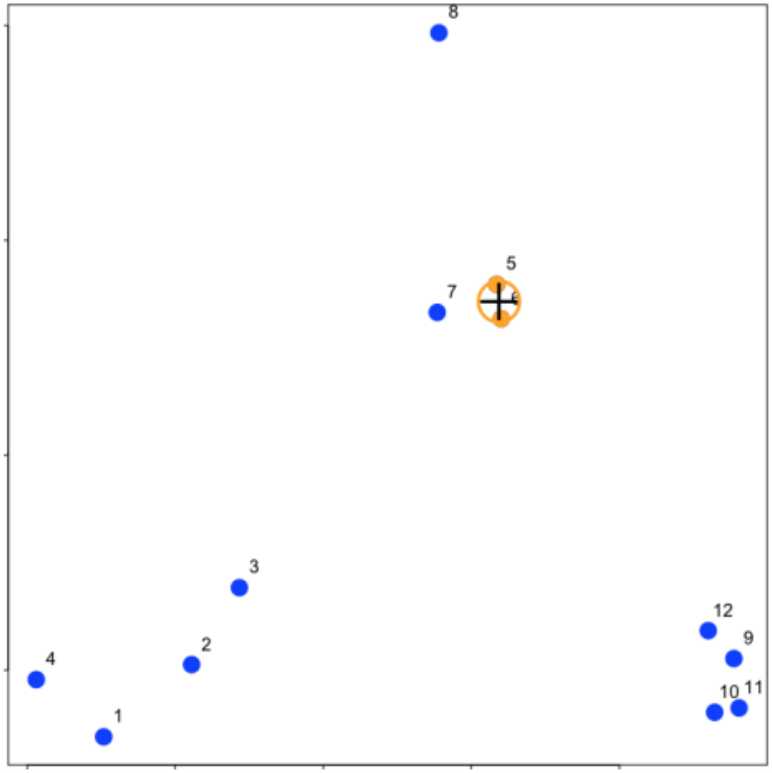




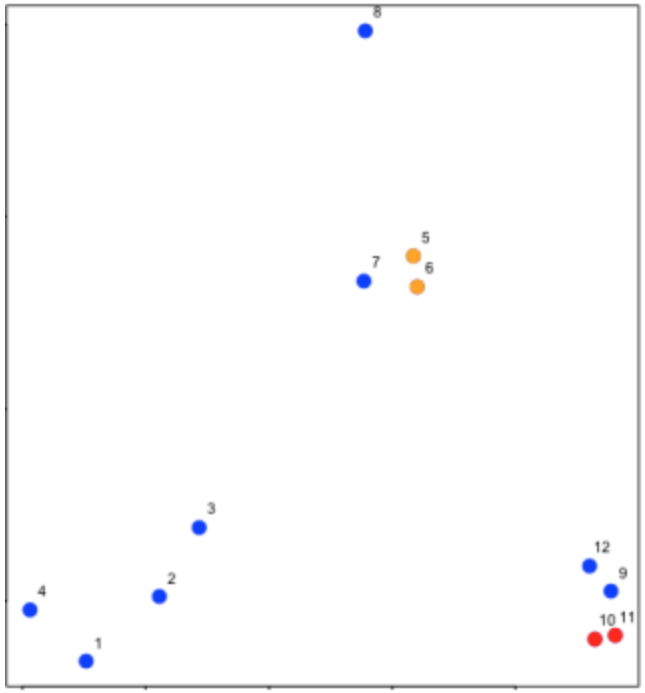
Hierarchisches Clustering

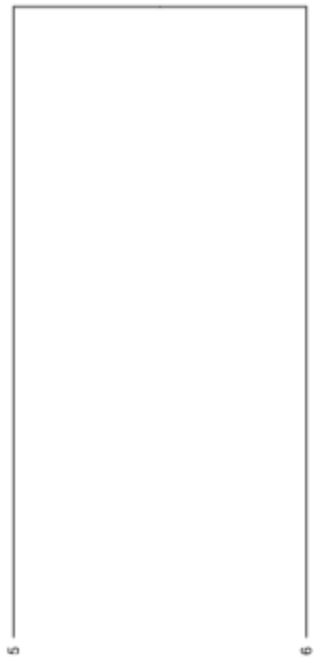
- Finde die nächsten Punkte -> schwieriger, weil 5 und 6 zusammen verschmolzen wurden

- Es gibt verschiedene Möglichkeiten, sie zusammenzuführen:
  - Messe den Abstand zu den Mittelpunkt Koordinaten (***average linkage***)
    e.g. $\sqrt{(x_7 - \bar{x}_{5,6})^2 + (y_7 - \bar{y}_{5,6})^2}$
  - ***Single Linkage***: nehme den Mindestabstand zu den verschmolzene Punkten
  - ***Complete Linkage***: nehme den maximalen Abstand zu den verschmolzene Punkten

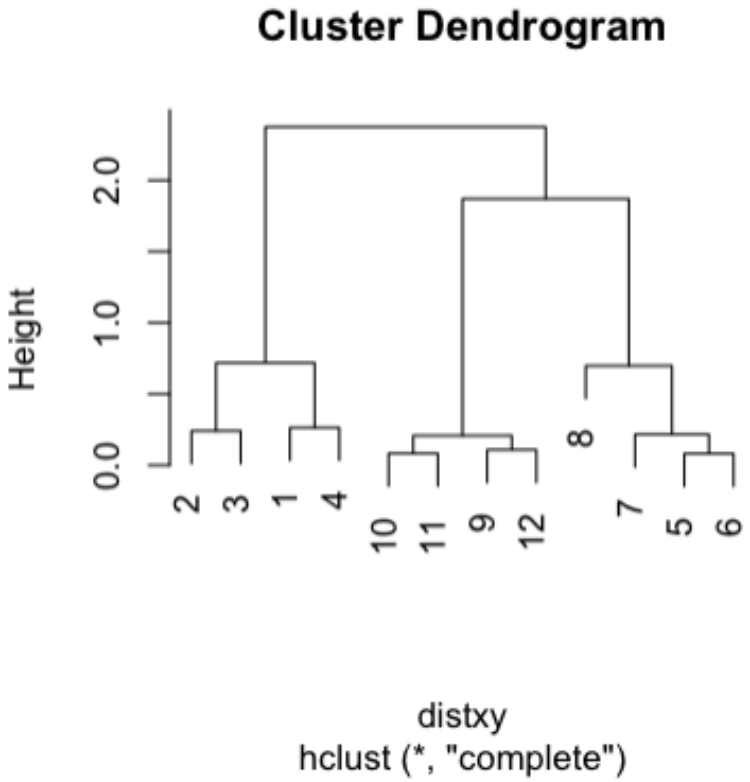


Hierarchisches Clustering



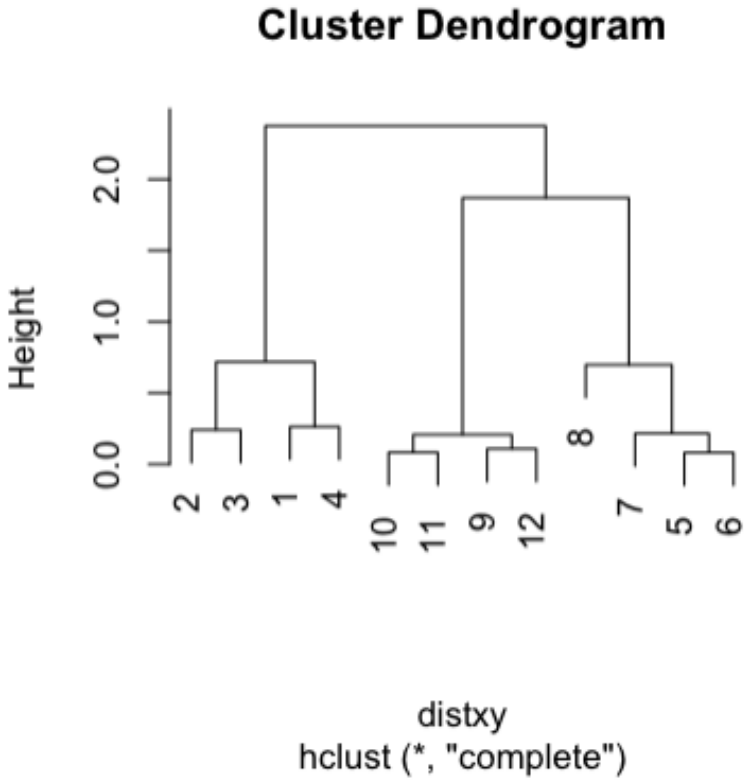


Hierarchisches Clustering



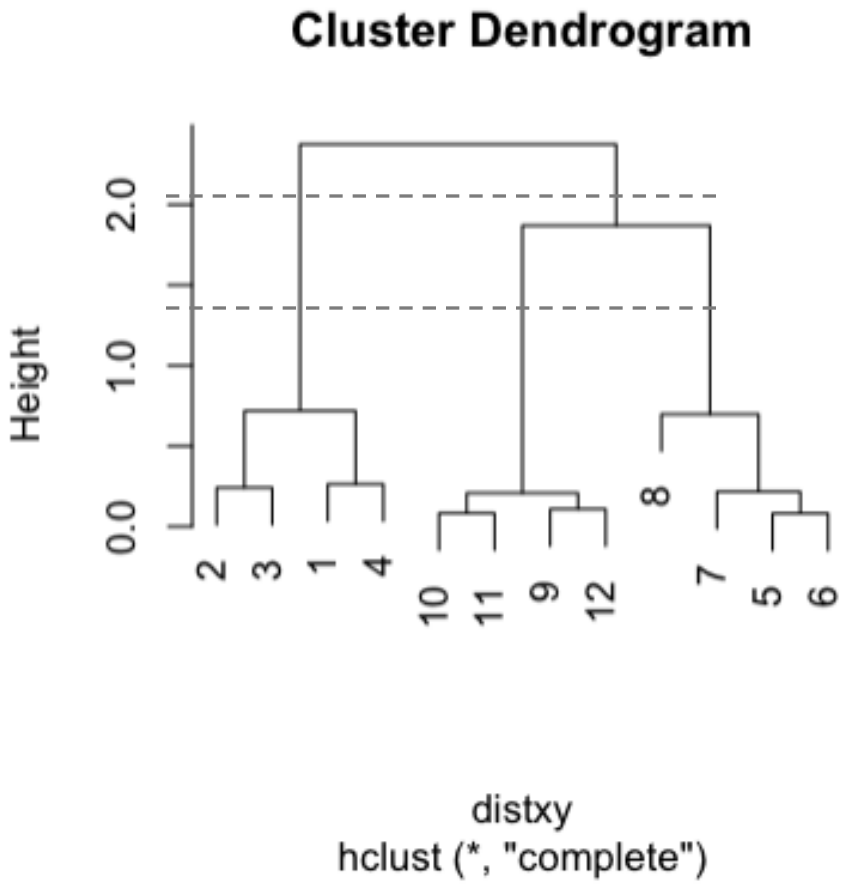
Hierarchisches Clustering

Aber wie bilden wir eigentlich Cluster?



Hierarchisches Clustering

Wir müssen eine Höhe auswählen und ‚schneiden‘

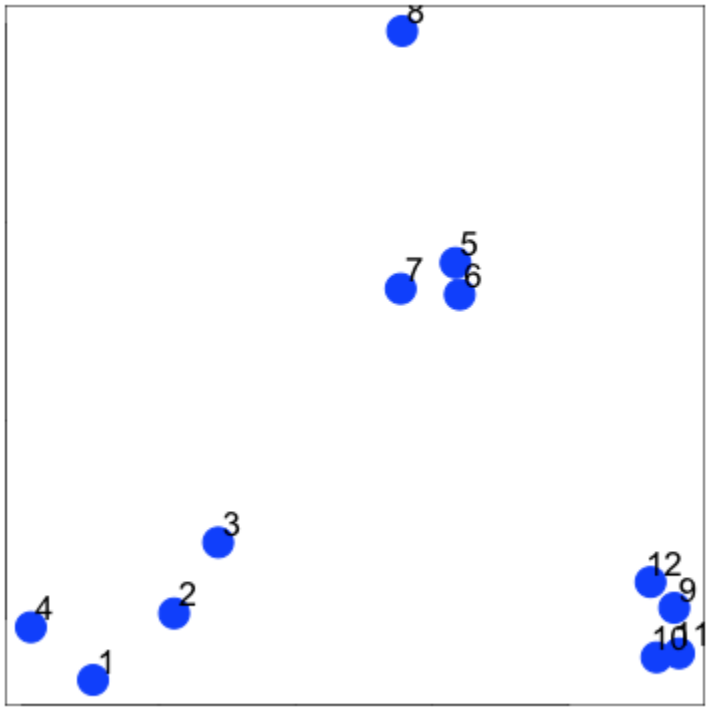


K-means Clustering

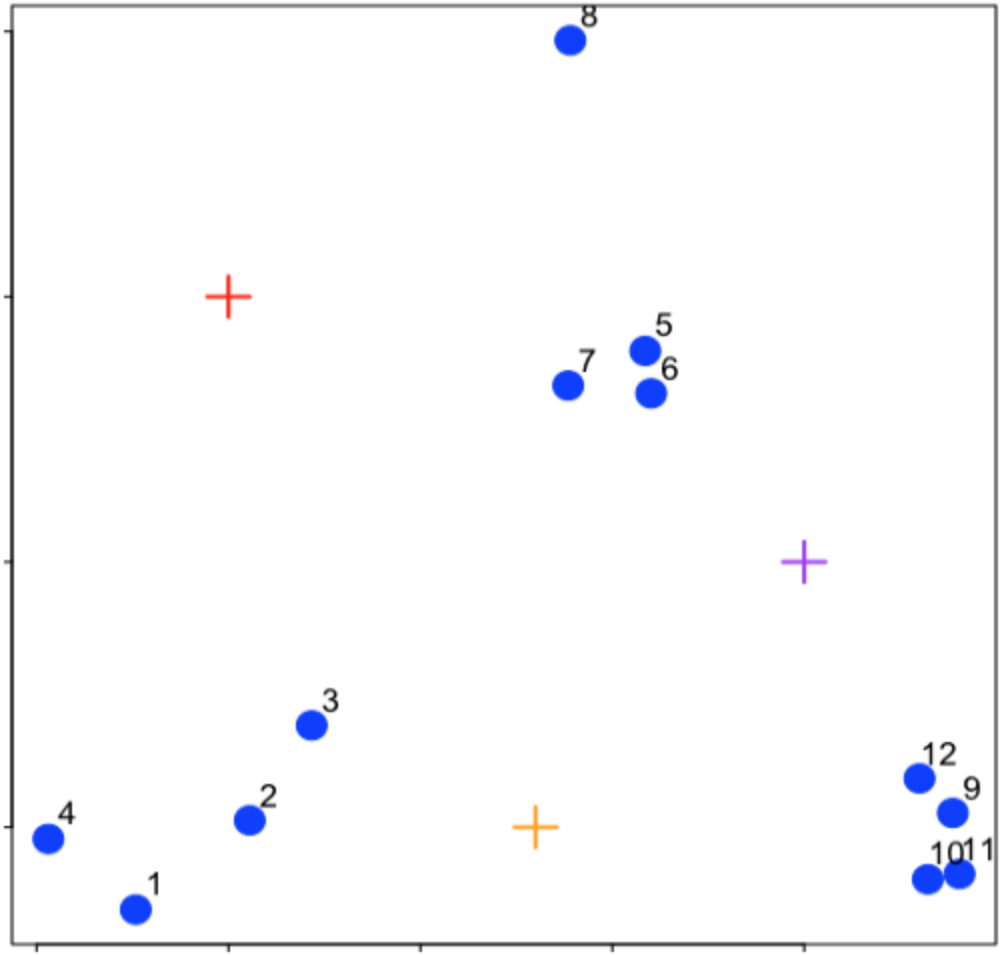
- Initialisiere die Cluster-Zentren
- Weise Werte den naheliegensten Zentren zu
- Die Zentren basierend auf den Werten aktualisieren
- Werte den Zentren neu zuordnen
- Wiederholen bis zur Konvergenz

K-means Clustering

Dies sind die Datenpunkte, die wir gruppieren möchten

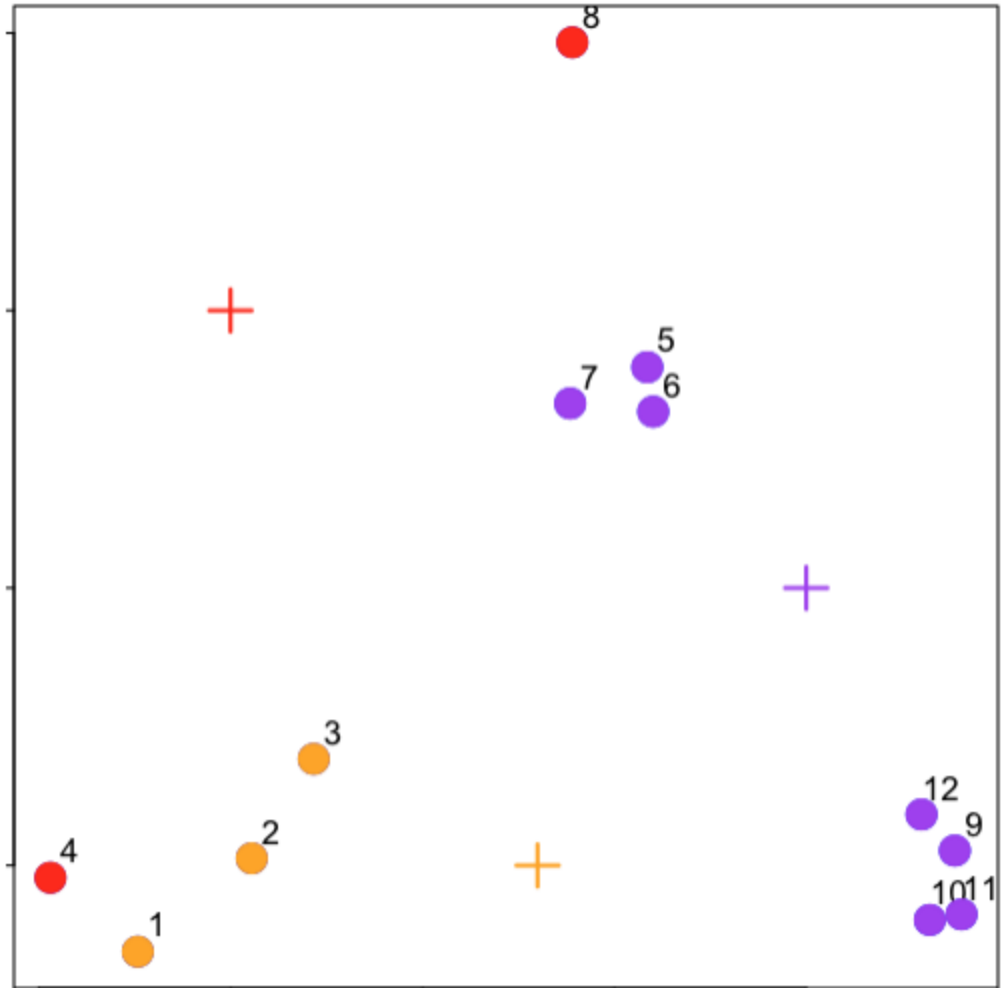


K-means Clustering



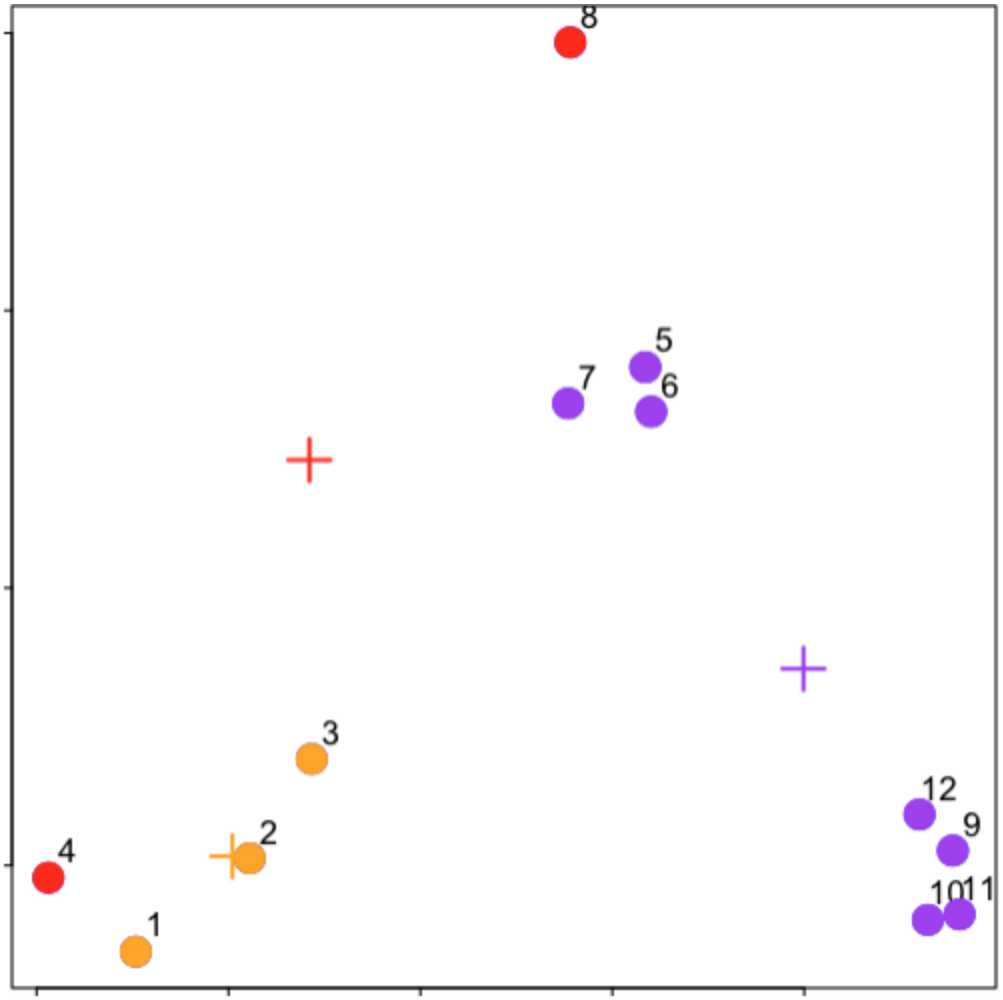
Wir beginnen mit dem Erraten der Positionen der Zentren

K-means Clustering



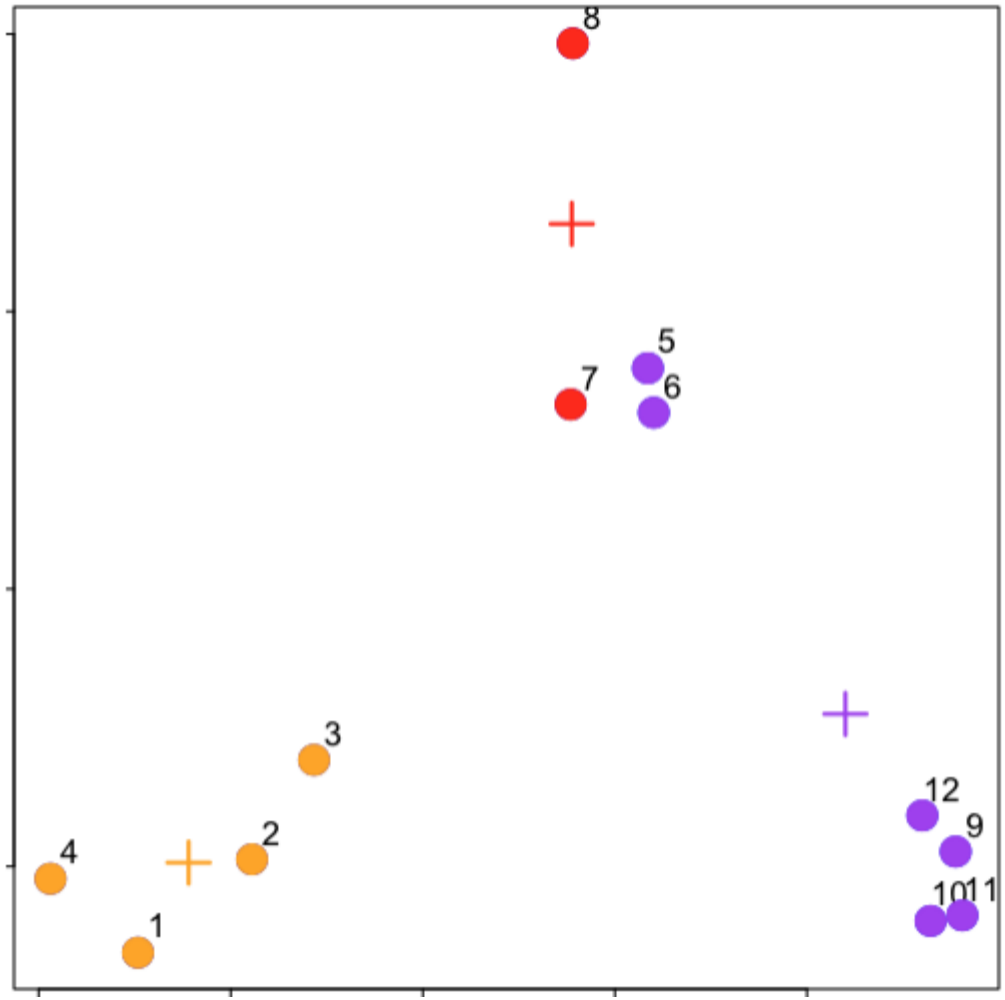
Weise alle Punkte den naheliegensten Zentren zu (durch die Benutzung der euklidischen oder Manhattan Distanz)

K-means Clustering

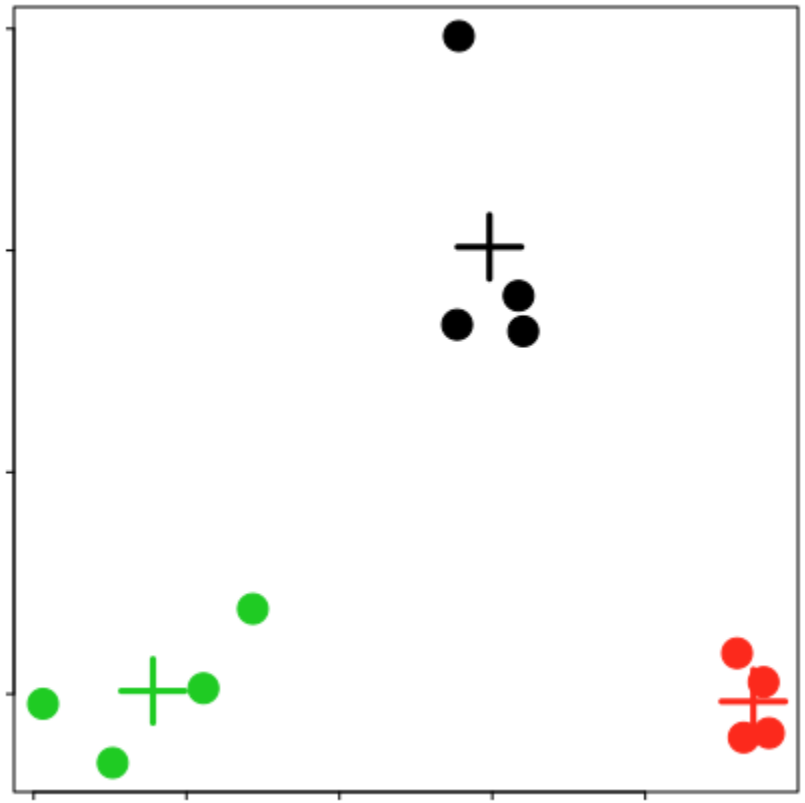


Berechne die Koordinaten der Zentren (Mittelpunkte) basierend auf den zugewiesenen Punkten.

K-means Clustering



K-means Clustering



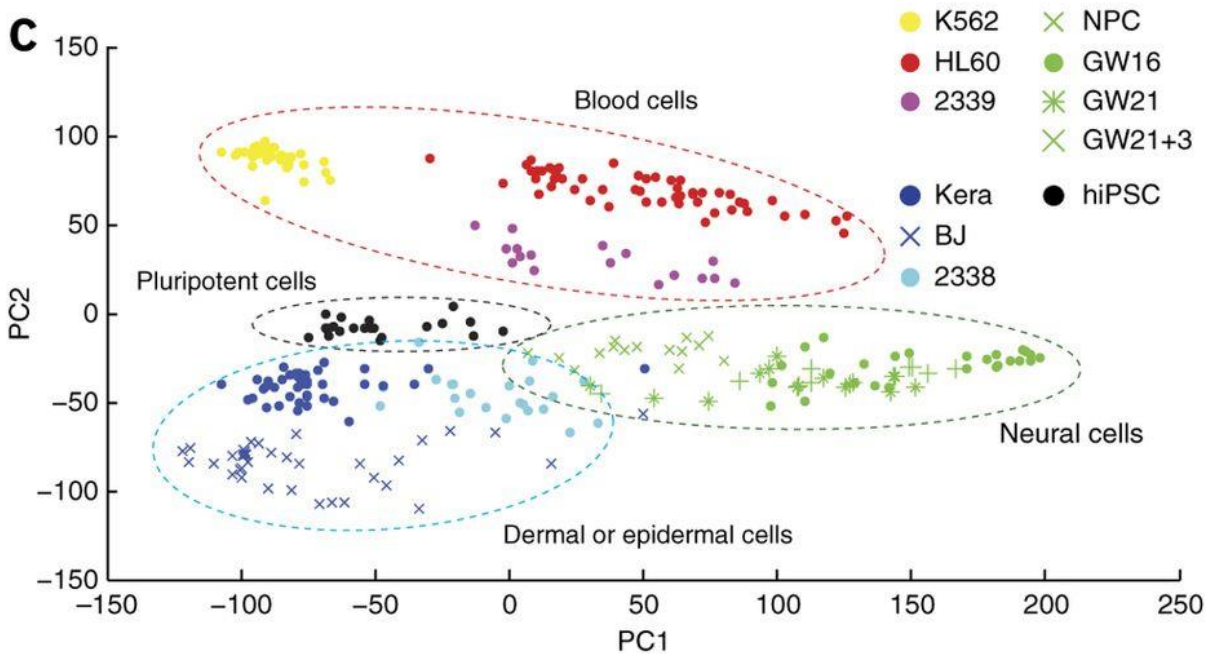
Stoppe, wenn die Cluster sich nicht mehr ändern oder die Koordinaten der Zentren nicht mehr variieren.

Hauptkomponentanalyse – Principal Component Analysis (PCA)

Beginnen wir mit einem Beispiel..

Dieses PCA-Diagramm zeigt Cluster von Zelltypen.
Die Expression von etwa 10000 Genen pro Sample wurde gemessen.

Jeder Punkt repräsentiert eine Probe und ihr Transkriptionsprofil.
Die allgemeine Idee ist, dass Zellen mit ähnlichen Expressionsprofilen clustern sollten

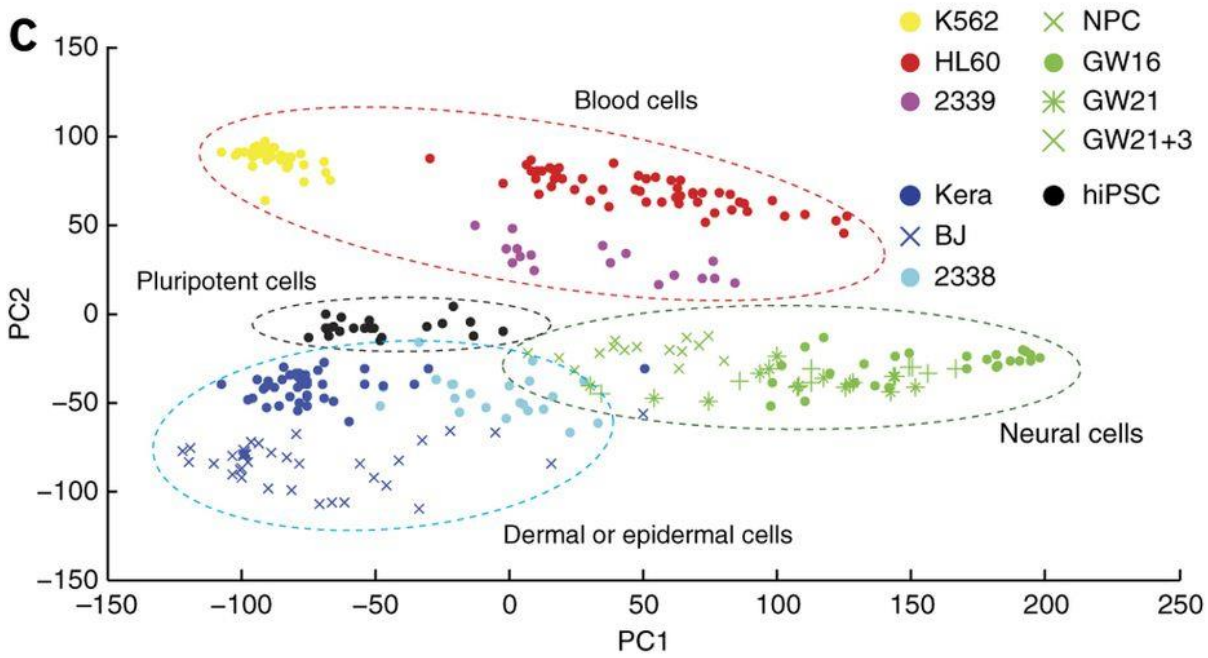


Pollen et al., Nature Biotechnology 2014

Beginnen wir mit einem Beispiel..

Dieses PCA-Diagramm zeigt Cluster von Zelltypen.
Die Expression von etwa 10000 Genen pro Sample wurde gemessen.

Frage: Sind alle Dimensionen / Features / Gen-Messungen super-wichtig oder sind einige wichtiger als andere?



Pollen et al., Nature Biotechnology 2014

Beginnen wir mit einem Beispiel..

Dieses PCA-Diagramm zeigt Cluster von Zelltypen.
Die Expression von etwa 10000 Genen pro Sample wurde gemessen.

Frage: Sind alle Dimensionen / Features / Gen-Messungen super-wichtig oder sind einige wichtiger als andere?

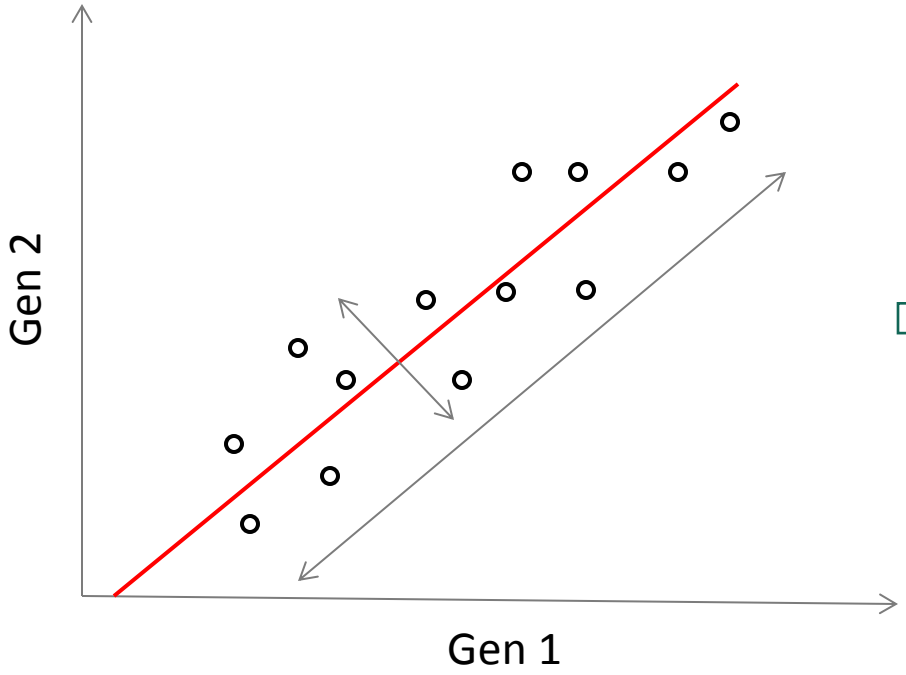
Jedes Gen fügt eine andere Dimension hinzu, aber einige Dimensionen sind wichtiger als andere.

Was hat das mit PCA zu tun?
PCA nimmt eine Datenmenge mit vielen Dimensionen (z.B. vielen Genen) und ‘glättet’ sie in 2 oder 3 Dimensionen, so dass wir uns in der Darstellung auf die Unterschiede zwischen den Genen konzentieren können.
→ Wir fangen die größte Variabilität der Daten ein.

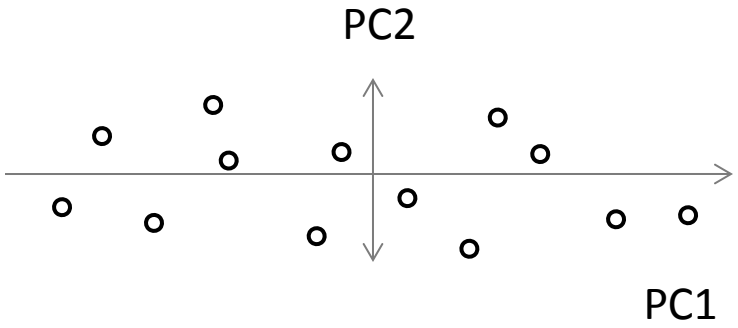
Pollen et al., Nature Biotechnology 2014

Beginnen wir mit einem Beispiel..

Die maximale Variation der Punkte ist entlang der diagonalen Linie



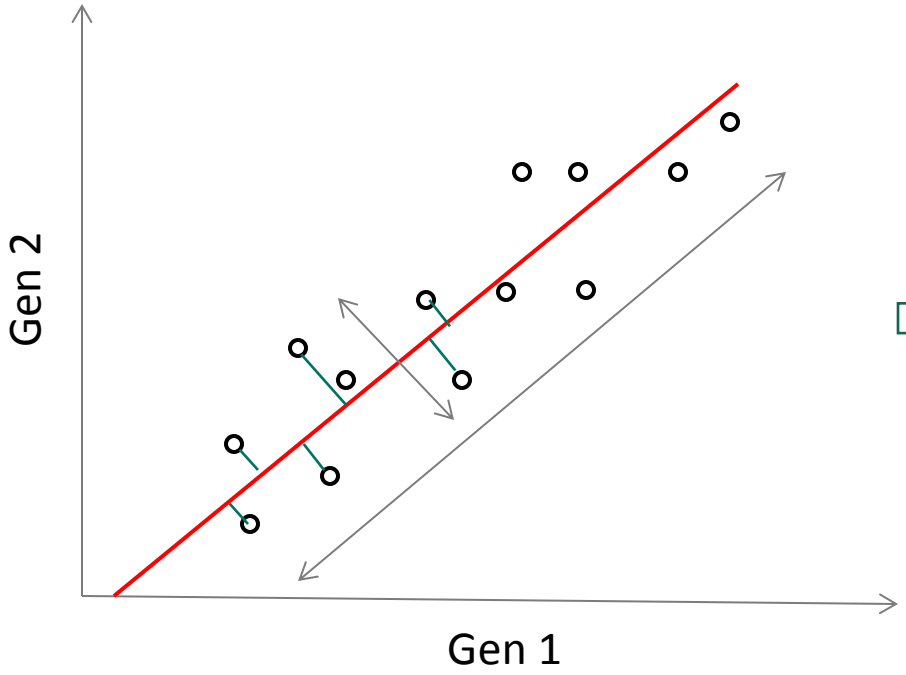
Wenn wir den Graphen drehen, bilden diese beiden Linien neue Achsen, die die linke / rechte und ober- / untere Variation leichter sichtbar machen.



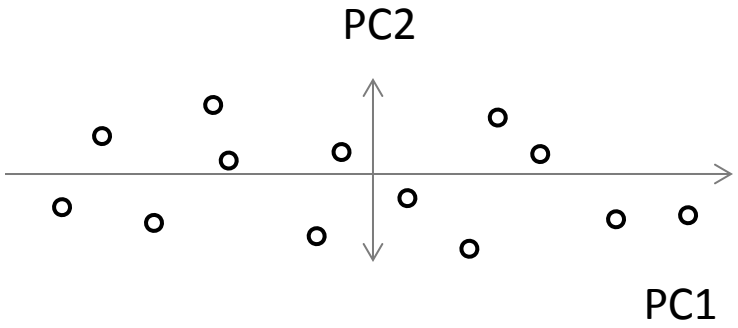
Diese beiden "neuen" oder "gedrehten" Achsen, die die Variation der Daten beschreiben, sind "Hauptkomponenten". PC1 ist die Achse, die die größte Variation überspannt; PC2 ist die Achse, die die zweite Variation überspannt.

Beginnen wir mit einem Beispiel..

Die maximale Variation der Punkte ist entlang der diagonalen Linie



Wenn wir den Graphen drehen, bilden diese beiden Linien neue Achsen, die die linke / rechte und ober- / untere Variation leichter sichtbar machen.



Mit anderen Worten: Ich möchte eine Linie finden, auf der die Punkte so projiziert werden, dass die Distanz der Punkte zur Linie klein sind (***Projektionsfehler***) und die Varianz maximal ist.

PCA – Formale Definition

Generell möchten wir die Datendimensionalität von n bis k reduzieren. Wir wollen die Vektoren $u^{(1)}, \ldots . u^{(k)}$ finden, auf die die Daten projiziert werden, um die Varianz zu maximieren und den Projektionsfehler zu minimieren.

Solche Vektoren werden Principal Components (PCs) genannt und PCA gibt uns einen formalen Weg, sie zu finden.

PCA – Schritt 1

Datenvorverarbeitung

m Datenpunkte; n Features oder Gene

- Für jedes Feature j berechne $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$
- Ersetze $x_j^{(i)}$ durch $x_j^{(i)} - \mu_j / s_j$, um zu erzwingen, dass die Features vergleichbare Wertebereiche haben

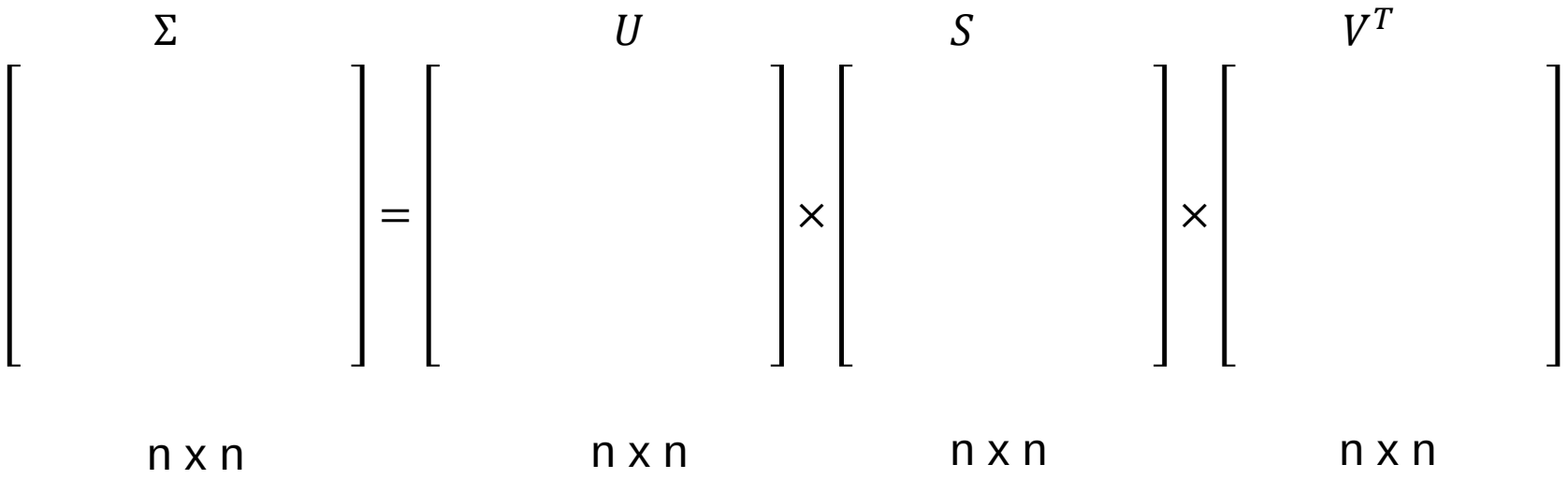
PCA – Schritt 2

- Berechne Σ , die ***Kovarianzmatrix*** der Daten $\Sigma = \frac{1}{m} X X^T$
- Berechne die ***Eigenvektoren*** der Matrix Σ mit ***Singular Value Decomposition (SVN)***

```
[U,S,V]=svd(sigma) or eigen(sigma)
```

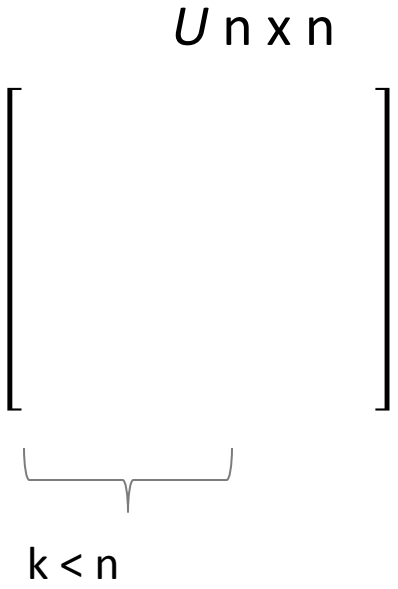
Mehr an der Tafel!

PCA – Schritt 2 – Details der Zersetzung



PCA – Schritt 2 - Details

Was uns interessiert, ist Matrix U, eine orthogonale Matrix, die die Eigenvektoren (Hauptkomponenten) des neuen Systems enthält, auf dem ich meine Daten projizieren möchte.



Wenn wir die Dimension von n auf k reduzieren wollen, müssen wir nur die ersten k Vektoren von U nehmen

$$u^{(1)} \ldots . . u^{(k)}$$

PCA – Schritt 2 - Details

Matrix S ist eine Diagonalmatrix n x n, deren Diagonaleingaben die ***Singular Values*** sind, die vom höchsten zum niedrigsten zählen. Viele dieser Einträge sind Null.

S n x n

$$\sigma_k \neq 0$$

$$\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}$$

Die ≠0 Werte von S entsprechen die Varianz jeder Komponente.
Die *singulären Werte* sind auch die *Eigenwerte,* die den Eigenvektoren in U entsprechen.

PCA – Schritt 3

- Projektiere die Punkte zu neuen Koordinaten
- Gehe von $x \in R^n$ nach $z \in R^k$ (vereinfachte Darstellung der Daten)
- Für jeden Punkt/Probe j

  $z_j = U_{reduced}^T \times X^j$

  k x 1 k x n n x 1
- Für alle Proben:

  $Z = U_{reduced}^T \times X$

  k x m k x n n x m
- Die Zeilen von Z sind die Hauptkomponenten; die Spalten von Z sind die Koordinaten der Punkte in neuen System

PCA Algorithmus (Zusammenfassung)

- Organiziere die Daten in einer Matrix X (n x m)
- Subtrahiere zu jedem Punkt den Mittelwert $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$
- Berechne die SVD oder die Eigenvektoren der Kovarianzmatrix Σ
- Nehme $U_{reduced}$ und berechne neue Koordinate $Z = U_{reduced}^T \times X$