

Hierarchische Clusteranalyse

Marvin Meusel, Markus Fleischauer

22.01.2016

Eingabe und Zielstellung

Eingabe

- eine Menge von Objekten
- Abstände zwischen Objekten
- keine Kenntnis über Anzahl oder Struktur der Cluster

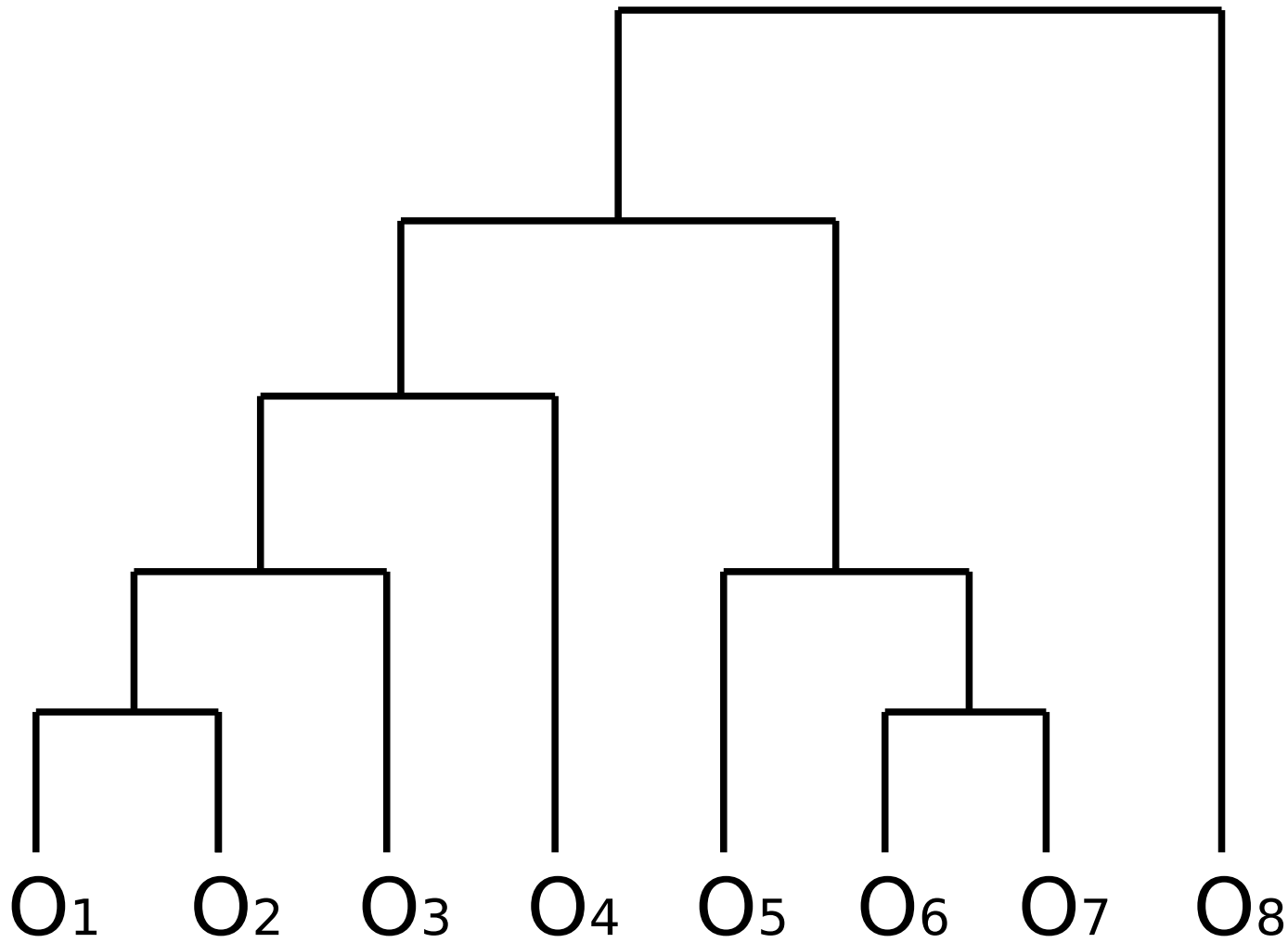
Zielstellung

- Daten Strukturieren → die Cluster finden

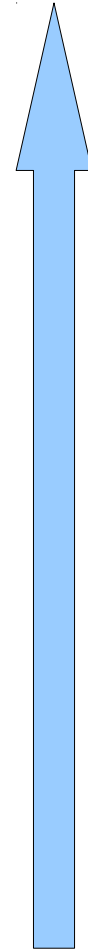
Cluster?

- Gruppe von Objekten mit gemeinsamen Eigenschaften

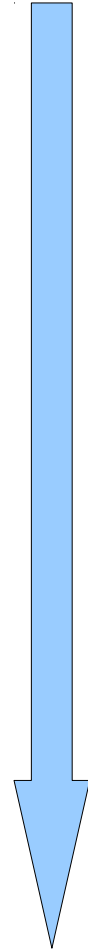
Hierarchisches Clustern



bottom-up agglomerativ



top-down divisiv



Die Distanzmatrix D

$D[i][j]$ = Distanz zwischen Objekt i und j

O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	
0	2	1	5	5	8	9	5	O ₁
	0	5	9	1	3	5	7	O ₂
		0	8	2	4	4	9	O ₃
			0	6	3	2	8	O ₄
				0	9	7	4	O ₅
					0	2	1	O ₆
						0	5	O ₇
							0	O ₈

Metriken

Metrik - Abstandsfunktion

- Positive Definitheit: $D_{i,j} \geq 0$ und $D_{i,j} = 0 \Leftrightarrow i = j$
- Symmetrie: $D_{i,j} = D_{j,i}$
- Dreiecksungleichung: $D_{i,j} \leq D_{i,k} + D_{k,j}$

Additive Baummetrik (4-Punkte Bedingung)

- Für alle i, j, k gilt:

$$D_{i,j} + D_{k,l} \leq \max\{D_{i,k} + D_{j,l}, D_{i,l} + D_{j,k}\}$$

Ultrametrik (3-Punkte Bedingung)

- Für alle i, j, k gilt: $D_{i,j} \leq \max\{D_{i,k}, D_{j,k}\}$

Pair Group Methods with Arithmetic Mean

Eingabe

- $n \times n$ Distanzmatrix D

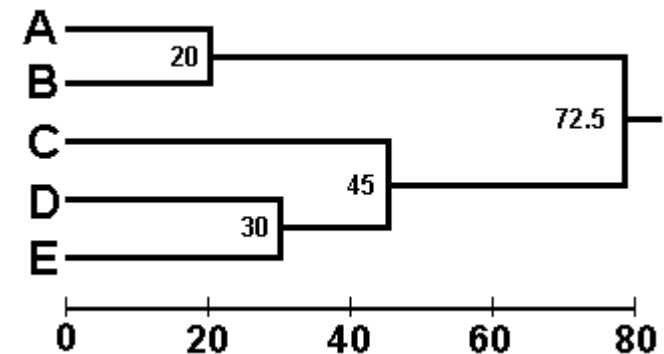
	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	
O ₁	0	2	1	5	5	8	9	5	O ₁
O ₂		0	5	9	1	3	5	7	O ₂
O ₃			0	8	2	4	4	9	O ₃
O ₄				0	6	3	2	8	O ₄
O ₅					0	9	7	4	O ₅
O ₆						0	2	1	O ₆
O ₇							0	5	O ₇
O ₈								0	O ₈

Ausgabe

- gewurzelter Baum mit Distanzen als Kantenlabel

Ablauf

- Zu Beginn pro Taxon ein Cluster
- In jedem Schritt



- Suche minimalen Eintrag $D[i][j]$ mit $i \neq j$
- Fasse die Cluster i und j zum Cluster k zusammen, lege neuen Knoten k im Baum an
- Berechne fehlende die Distanzen in D

Neighbour Joining (NJ)

Eingabe

- $n \times n$ Distanzmatrix D

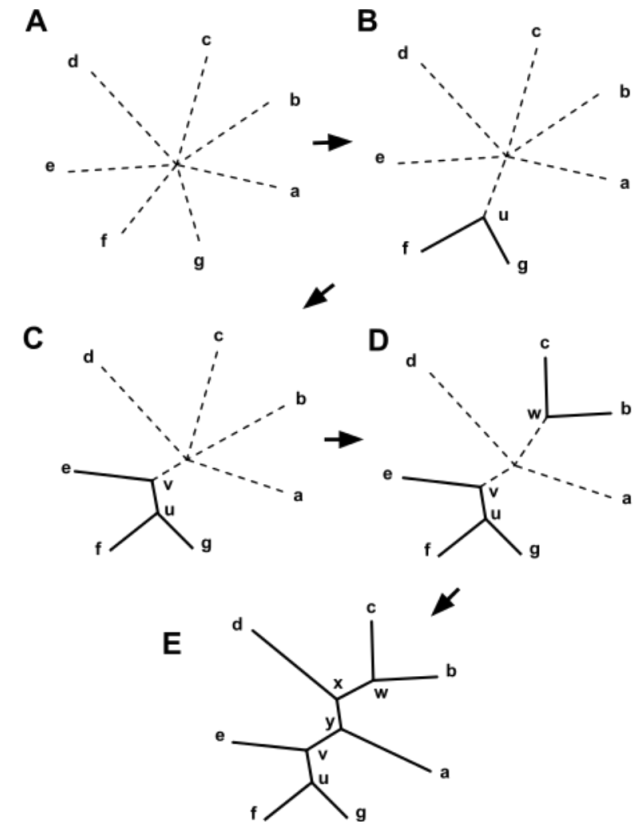
	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	
O ₁	0	2	1	5	5	8	9	5	O ₁
O ₂		0	5	9	1	3	5	7	O ₂
O ₃			0	8	2	4	4	9	O ₃
O ₄				0	6	3	2	8	O ₄
O ₅					0	9	7	4	O ₅
O ₆						0	2	1	O ₆
O ₇							0	5	O ₇
O ₈								0	O ₈

Ausgabe

- ungewurzelter Baum mit Distanzen als Kantenlabel

Ablauf

- Start mit „Sternbaumtopologie“
- in jedem Schritt
 - Finde Clusterpaar $i \neq j$ zum zusammenfassen
 - Fasse die Cluster i und j zum Cluster u zusammen
 - Füge neuen Knoten u in Baum und Matrix D ein
- Berechne fehlende die Distanzen in D



UPGMA und WPGMA

Neuberechnung der Distanzmatrix

- entferne Cluster i und j , füge neues Cluster k ein
- berechne die Distanzen von k zu den bestehenden Clustern
 - Distanz des neuen Clusters zu einem bereits bestehenden Cluster f

- WPGMA
$$D_{x,k} = D_{k,x} = \frac{D_{i,k} + D_{j,k}}{2}$$

- UPGMA
$$D_{x,k} = D_{k,x} = \frac{|i| \cdot D_{i,K} + |j| \cdot D_{j,k}}{|i| + |j|}$$

Neighbour Joining (NJ)

Geeignetes Clusterpaar auswählen

- Es müssen die Durchschnittlichen Distanzen von jedem Taxon zu jedem anderen Taxon berechnet werden.

$$r_i = \frac{1}{N-2} \sum_{k=1}^N D_{i,k}$$

- Wir berechnen eine Zwischenmatrix M.

$$M_{i,j} = D_{i,j} - (r_i + r_j)$$

- Suche minimalen Eintrag $M[i][j]$ mit $i \neq j$
- Fasse die Cluster i und j zum Cluster u zusammen, lege neuen Knoten im Baum an

Neighbour Joining (NJ)

Neuberechnung der Distanzmatrix

- Kantenlängen von i und j zu u berechnen sich wie folgt:

$$v_{i,u} = \frac{D_{i,j} + r_i - r_j}{2}$$

$$v_{j,u} = D_{i,j} - v_{i,u}$$

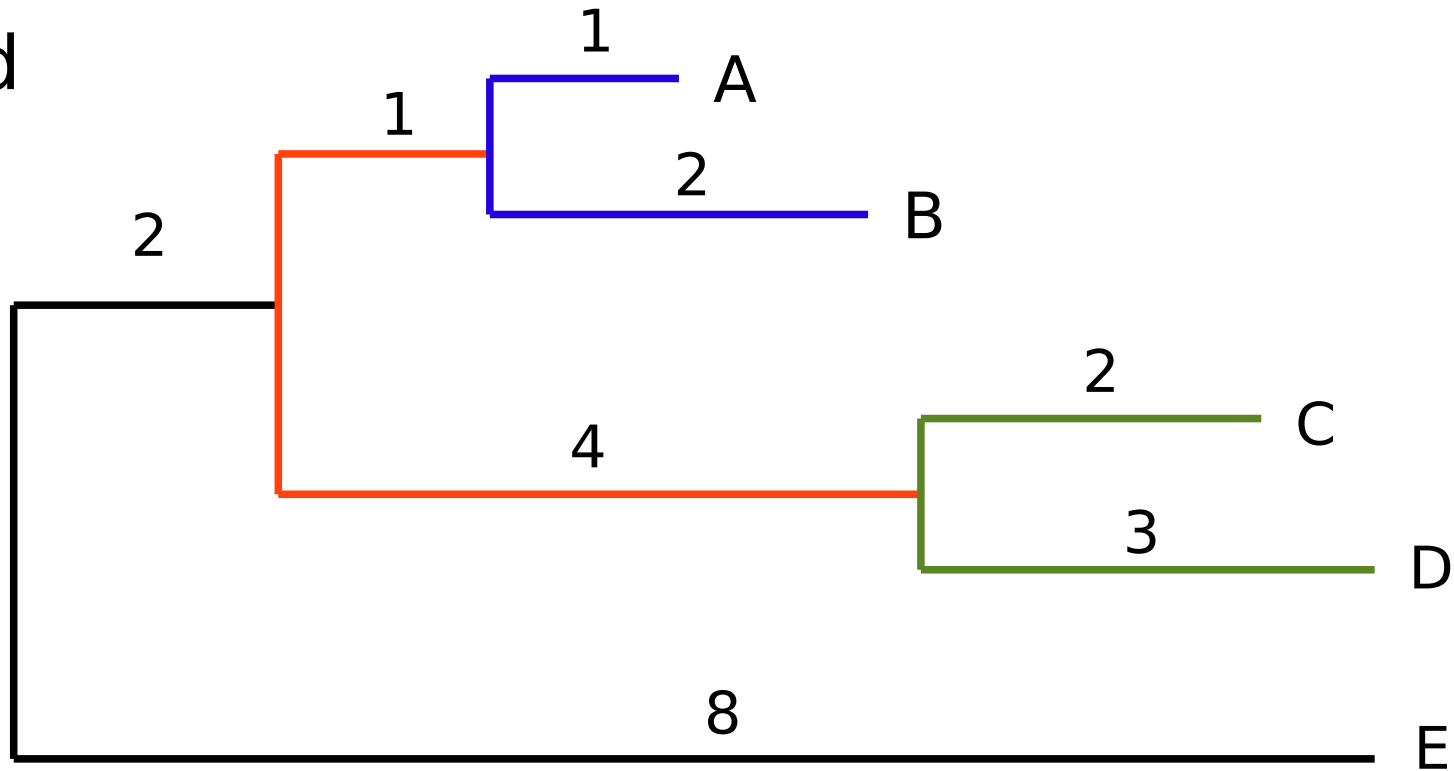
- Neues Cluster u zur Matrix D hinzufügen
- Fehlende Abstände berechnen durch:

$$D_{u,k} = \frac{D_{i,k} + D_{j,k} - D_{i,j}}{2}$$

- Knoten i und j aus D entfernen

Das Newick Format

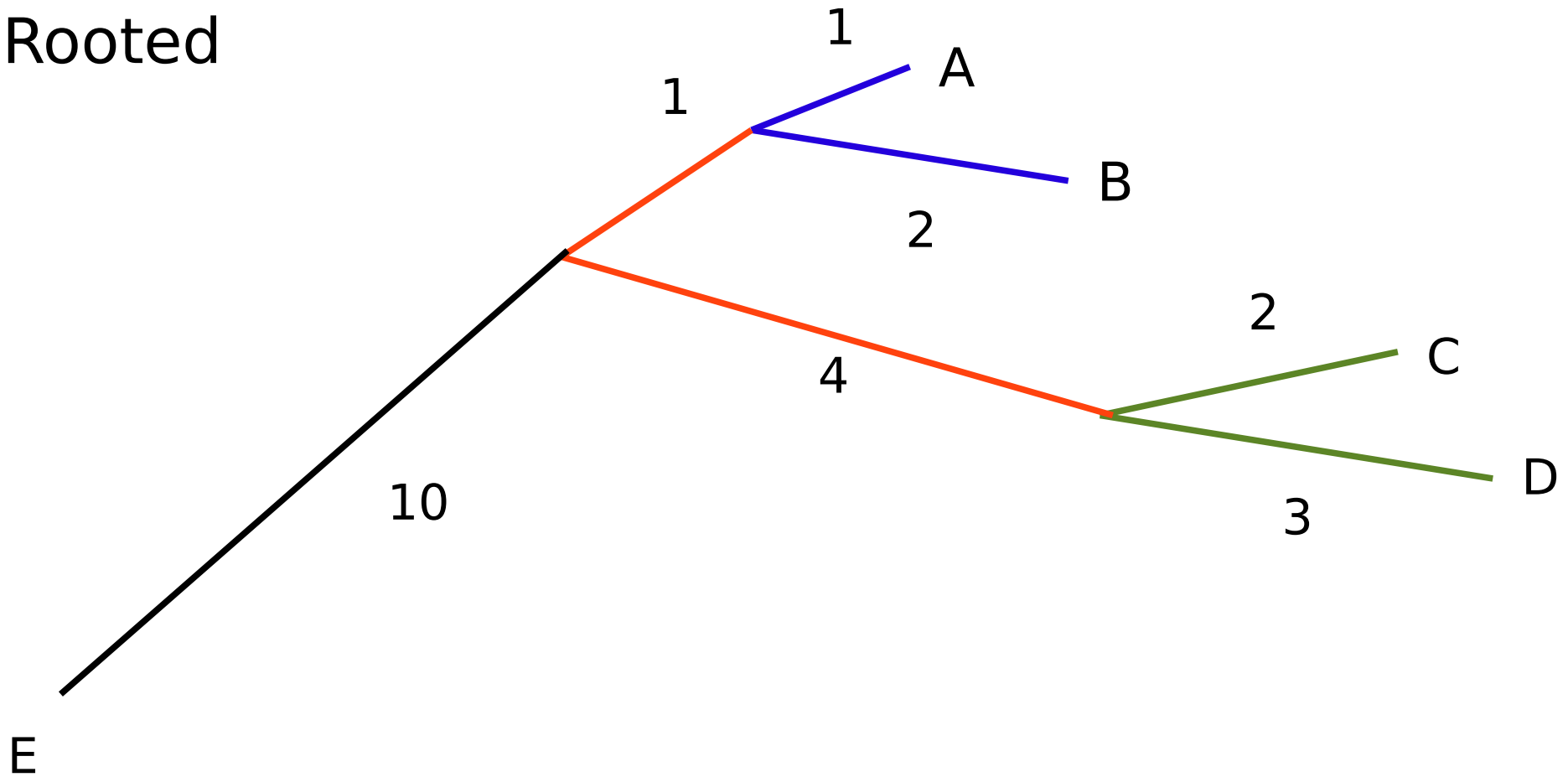
Rooted



```
((A:1,B:2):1,(C:2,D:3):4):2,E:8);
```

Das Newick Format

UnRooted



```
((A:1,B:2):1,(C:2,D:3):4,E:10);
```

Anforderungen an die Implementierung

- Einlesen einer Datei mit mehreren DNA-Sequenzen im multiple Fasta Format, Berechnung einer Distanzmatrix aus paarweisen Edit-Distanzen (Needleman-Wunsch) und Ausgabe der Matrix.
- Einlesen einer Distanzmatrix. Konstruktion eines Stammbaumes mit UPGMA, WPGMA und NJ. Ausgabe im Newick Format.
- Bestimmung des Metrik-Typs einer Distanzmatrix. Testet damit jeweils die eingelesene und erzeugte Matrix!

Aufgaben

Aufgaben

- Implementierung der Klassen mit den entsprechenden Algorithmen
- API Dokumentation mit javadoc
- Projektmanagement mit Gradle
- Unit-Tests
- Benutzerinterface als Kommandozeilenprogramm (CLI) + README
- Testläufe und Testdaten mit Protokollieren der Ergebnisse (Testdaten und Aufgabenblatt auf der Kurswebsite)