

**Vorlesung: Phylogenetik  
Wintersemester 2010/2011**

**Übungen**

Weihnachtszettel

**Übung 9, Abgabe: 13.01.2011**

1. **Additive Bäume** (3 Punkte)  
Beschreibe das *Least-Squares*-Kriterium und erkläre, wie es zur Rekonstruktion einer Phylogenie verwendet werden kann. (5–8 Sätze)
2. **Neighbor Joining** (4 Punkte)  
Gegeben sei die Distanzmatrix:

	A	B	C	D	E
A	0	5	7	4	5
B		0	5	3	5
C			0	3	6
D				0	8
E					0

Benutze den *Neighbour Joining*-Algorithmus um aus der Matrix einen phylogenetischen Baum zu konstruieren.

3. **Wiederholung Additive Metriken, Ultrametrien** (4 Punkte)
  - (a) Eine Metrik  $d$  ist eine additive Metrik auf einer Objektmenge  $O$ , wenn die *Vier-Punkt-Bedingung* für **jedes** 4-er Tupel von Elementen aus  $O$  gilt. Finde eine Metrik auf einer fünfelementigen Objektmenge, bei der es sowohl 4-er Tupel gibt, die diese Bedingung erfüllen, als auch solche, die sie nicht erfüllen. Zeige, dass diese Metrik nicht additiv sein kann, indem du versuchst einen eindeutigen additiven Baum für  $O$  zu rekonstruieren. (Merke: Für eine additive Metrik  $d^M$  existiert ein eindeutiger Baum  $T$  für den gilt:  $d^T = d^M$ .)
  - (b) Eine Metrik  $d$  ist eine Ultrametrik auf einer Objektmenge  $O$ , wenn die *Drei-Punkt-Bedingung* für **jedes** 3-er Tupel von Elementen aus  $O$  gilt. Finde eine Metrik auf einer vierelementigen Objektmenge, bei der es sowohl 3-er Tupel gibt, die diese Bedingung erfüllen, als auch solche, die sie nicht erfüllen. Zeige, dass diese Metrik nicht ultrametrisch sein kann, indem du versuchst einen eindeutigen ultrametrischen Baum für  $O$  zu rekonstruieren. (Merke: Für eine Ultrametrik  $d^M$  existiert ein eindeutiger ultrametrischer Baum mit  $d^M = d^T$ , für den gilt, dass die Distanz von der Wurzel zu jedem Blatt gleich ist.)

## Zusatzaufgaben

Die Punkte für diese Aufgaben sind Zusatzpunkte, werden also nicht in den Pflichtteil der Übungen mit eingerechnet. Ihr könnt sie aber nutzen um die benötigte Prozentzahl zu erreichen.

Für die Aufgaben mit Programmieranteil solltet ihr die Programme, sowie einen Beispielaufruf mit Ergebnis bis zum 13.01.2011 zum Vorlesungsbeginn an eure Tutoren schicken. Die Standardprogrammiersprachen sind Java oder Haskell, wenn ihr andere nutzen wollt spricht das bitte vorher mit eurem Tutor ab.

Unkommentierte Programme werden selbstverständlich nicht korrigiert.

### 1. Newick Bäume (4 Punkte)

Schreibe ein Programm, das als Eingabe eine Anzahl von Taxa erhält und alle möglichen Topologien binärer Bäume mit dieser Anzahl Blätter ausgibt. Das Programm soll die Topologien für gewurzelte **oder** ungewurzelte Bäume im Newick-Format ausgeben.

Gib sowohl das Programm als auch einen Beispielaufruf mit 6 Taxa und dessen Ergebnis ab.

Beispiel (ungewurzelt):

```
>topologie 4
```

```
(1,2)(3,4)
```

```
(1,3)(2,4)
```

```
(1,4)(2,3)
```

Beispiel (gewurzelt):

```
>topologie 3
```

```
(1,(2,3))
```

```
(2,(1,3))
```

```
(3,(2,1))
```

### 2. Minimale Spannbäume (3 Punkte)

Schreibe ein Programm, das als Eingabe einen Graphen  $G$  erwartet und einen Minimalen Spannbaum von  $G$  ausgibt.

Beispiel:

```
>spannbaum (A,B,C,D,E,F) ((A,B)4,(A,E)2,(A,F)2,(B,C)3,(C,D)4,(C,E)3,(C,F)1,(D,E)3,(E,F)3)
```

```
minimaler Spannbaum: (A,B,C,D,E,F) ((A,F)2,(A,E)2,(B,C)3,(D,E)3)
```

Gib sowohl das Programm als auch einen Beispielaufruf (für einen Graphen mit mind. 6 Knoten) und das Ergebnis ab.

### 3. Wiederholung Algorithmen (12 Punkte)

Beantworte für die Algorithmen: *Perfect Phylogeny* (Kapitel 3), *Hartigan* (Kapitel 4), *Greedy Sequential Addition* (Kapitel 5), *Neighbor-Joining* (Kapitel 6) und *Minimal Evolution* (Kapitel 6) die Fragen:

Was berechnet der Algorithmus? (Was ist die Ein- und was ist die Ausgabe?)

Unter welchen Bedingungen ergibt es Sinn diesen Algorithmus einzusetzen?

Wie lässt sich der Algorithmus in den Gesamtzusammenhang der Rekonstruktion von phylogenetischen Bäumen einordnen?

Nutzt er eine Heuristik, wenn ja: welche?

Wie ist die Laufzeit des Algorithmus?