Bioinformatik I, WS99/00
Uebungsblatt 6

Besprechung am 19.01.2000

Quartet Puzzling

Manche Verfahren zur Konstruktion phylogenetischer Bäume, z.B. Felsensteins Methode, sind für größere Stammbäume nicht mehr effizient einzusetzen. Mit der Idee des sogenannten *Quartet Puzzling* können solche Verfahren dennoch auch für die Konstruktion großer phylogenetischer Bäume genutzt werden. In der Übungsaufgabe wird diese Vorgehensweise am Beispiel von UPGMA mit Quartet Puzzling demonstriert. Dazu soll insbesondere der Quartet Puzzling Schritt implementiert werden.

Quartet Puzzling ist die Konstruktion eines (nicht gewurzelten) Baumes aus allen seinen Teilbäumen mit je 4 Blättern (*Quartet Trees*). Kombiniert man ein Verfahren zur Stammbaumkonstruktion mit Quartet Puzzling, berechnet man also zunächst sämtliche Quartet Trees nach diesem Stammbaumkonstruktionsverfahren und fügt diese dann im Quartet Puzzling Schritt zum gesamten Baum zusammen.

Ein *Quartet Tree* definiert eine Beziehung auf seinen vier Blättern (also zwischen vier Spezies) in der Weise, dass jeweils zwei Spezies näher miteinander verwandt sind als mit den anderen beiden. Das kann man als Nachbarschaftsrelation interpretieren.

$$N(a, b; c, d)$$

steht dafür, dass a und b näher miteinander verwandt sind als mit c oder d. Diese Relation muss auch für einen gesamten Baum gelten, von dem der Quartet Tree Teilbaum ist.

Der Algorithmus für Quartet Puzzling geht aus von einem beliebigen Quartet Tree aus der Liste der gegebenen Quartet Trees. Iterativ wird dann dem Baum in einem Schritt jeweils ein Blatt hinzugefügt. Dazu sucht man eine Kante an der der Baum erweitert werden kann, so dass möglichst wenige (oder keine) Nachbarschaftsrelationen verletzt werden. Für diese Suche bekommt jede Kante des Baumes einen Wert, zunächst 0. Für jede Nachbarschaftsrelation (jeden Quartet Tree) die den neu hinzuzufügenden Knoten enthält, erhöht man den Wert aller Kanten die die Relation verletzen um 1, d.h. eigentlich werden nur die Werte der Kanten auf dem Pfad von a nach b erhht, falls die Relation N(a,b;c,e) ist und e eingefügt werden soll. An einer Kante die nach diesem Verfahren den kleinsten Wert hat, fügt man das Blatt ein. (D.h. ein neuer interner Knoten wird erzeugt, der diese Kante teilt. Außerdem fügt man eine Kante von diesem Knoten zum neuen Blatt ein.).

Aufgabe

Für die Kombination mit UPGMA gehen wir von einer $n \times n$ Distanzmatrix von n Spezies aus. Aus dieser Matrix sollen sämtliche Distanzmatrizen für je 4 Spezies erzeugt werden. Aus diesen Distanzmatrizen berechnet ein vorgegebenes Programm die Nachbarschaftsrelationen. Diese Nachbarschaftsrelationen sollen von einem weiteren Programm eingelesen werden, das daraus per Quartet Puzzling einen Baum konstruiert.

Implementieren Sie zunächst die Erzeugung der 4×4 Matrizen aus einer $n \times n$ Matrix, so dass das Programm Ausgaben in folgender Art liefert:

```
a b c d
0 3 4 4
3 0 3 3
4 3 0 2
4 3 2 0

a b c e
0 4 4 2
4 0 2 4
4 2 0 4
2 4 4 0

. . .
```

Das heißt für jede 4er Gruppe von Spezies werden zunächst die (Identifikatoren der) vier Spezies ausgegeben und dann deren Distanzmatrix.

Über die Webseite der Vorlesung finden Sie ein Programm `phy.c`, das aus dieser Ausgabe die Nachbarschaftsrelationen berechnen kann. Angenommen Ihr Programm heißt `generate-quartets` und liest eine $n \times n$-Matrix aus der Datei `distmat`, dann erhalten Sie die Nachbarschaftsrelationen mit

```
generate_quartets distmat | phy -u -q
```

als Liste in der Art

```
c d a b
a e b c
.
.
.
```

Wobei jede Zeile einem Tuple `N(a,b;c,d)` entspricht. Zum Beispiel enstspricht die erste Zeile `N(c,d;a,b)`.

Implementieren Sie schliesslich ein Programm, dass für solche Tuple das Quartet Puzzling durchführt. (Hinweis: Der Baum kann als doppeltverketteter acyclischer Graph durch mit Zeigern verknüpfte Knoten implementiert werden. Um z.B. alle Kanten auf 0 zu setzen oder Kantenwerte zu erhöhen kann man dann die Struktur rekursiv durchlaufen.)