

Übungen zur Vorlesung
**Einführung in die angewandte
Bioinformatik**
Sommersemester 2009

Übungsblatt 8
Bearbeitungszeit:
18.06.2009

Aufgabe 8.1 – Phylogenie: Neighbor-joining vs den Rest

In dieser Aufgabe sollen Sie erneut Methoden zur Rekonstruktion von phylogenetischen Bäumen vergleichen. Es geht dieses Mal zusätzlich zu Neighbor-joining und UPGMA um *Maximum Parsimony*. Laden Sie sich zunächst die Datei 1ba.phy von der Vorlesungswebseite herunter. Sie enthält DNA-Sequenzen im passenden Dateiformat. Benutzen Sie das Mobyli-Portal (verlinkt von der Vorlesungswebseite, funktioniert nur mit dem Firefox vernünftig), um folgende Schritte durchzuführen:

- Berechnen Sie paarweise Distanzen der Sequenzen mit dem Programm *dnadist*.
- Leiten Sie das Ergebnis von *dnadist* an das *neighbor*-Programm weiter, um sowohl einen Baum zu erhalten, der mit Neighbor-joining erzeugt wurde, als auch einen, der mit UPGMA erzeugt wurde. Die Möglichkeit dazu haben Sie auf der Ausgabeseite von *dnadist*.
- Benutzen Sie das Programm *dnapars*, um mit der Maximum-Parsimony-Methode einen phylogenetischen Baum der Sequenzen zu erhalten. Warum ist es nicht nötig, vorher *dnadist* zu benutzen, um die Distanzen zu berechnen? _____

Wie hoch sind die Kosten des Maximum-Parsimony-Baums? _____

Die Bäume, die Sie erhalten, werden zunächst als reiner Text angezeigt. Wenn Sie möchten, können Sie die Bäume noch grafisch anzeigen lassen. Schauen Sie dazu unter "Tree file" in das Menü vor "further analysis".

Unterscheiden sich die UPGMA- und Maximum-Parsimony-Bäume? Wenn ja, worin?

Unterscheiden sich der Neighbor-Joining- und der Maximum-Parsimony-Baum? Wie erklären Sie den Unterschied?

Aufgabe 8.2 – Scharf nachdenken

Nehmen Sie an, Sie sind auf einem Bauernhof und hören Hufgeklapper, ohne die Quelle zu sehen. Welche Vermutung sollten Sie als erstes haben?

- a) Es handelt sich um ein oder mehrere Pferde.
- b) Es handelt sich um ein oder mehrere Zebras.

Nach welchem Prinzip denken Sie? _____

Aufgabe 8.3 – Taxonomiedatenbank des NCBI

Benutzen Sie die Taxonomie-Datenbank des NCBI, um sich über die wissenschaftlichen Namen einiger Organismen zu informieren. Schlagen Sie die Namen folgender Organismen nach:

Maus _____
Katze _____
Hund _____
Hefe (engl. yeast) _____

Sehen Sie sich auch mindestens einen Eintrag genauer an.

Was verbirgt sich hinter folgenden Namen?

Arabidopsis thaliana _____
Drosophila melanogaster _____
Caenorhabditis elegans _____
Vulpes vulpes _____
Danio rerio _____

Schauen Sie sich noch einmal genauer den Eintrag zur Maus an, insbesondere den *Genetic code*. Gegeben das DNA-Stück ATAGCGGTCTCT, welche Aminosäuresequenzen entstehen daraus, wenn Sie diese Sequenz a) aus einem Chromosom der Maus und b) aus dem Mitochondrion einer Mauszelle erhalten haben?

Aufgabe 8.4 – \mathcal{O} -Notation

In dieser Aufgabe sollen Sie die Laufzeit eines Algorithmus analysieren, der Matrizenmultiplikationen durchführt.

Gegeben seien zwei Matrizen A und B , die jeweils n Zeilen und n Spalten haben (die Matrizen sind also quadratisch); die Matrizen sollen ganze Zahlen enthalten. Das Ziel ist es, die Matrix $C = A \cdot B$ auszurechnen. Rufen Sie sich das Standardverfahren zur Matrizenmultiplikation in Erinnerung. Wenn Sie davon ausgehen, dass sowohl die Addition als auch die Multiplikation von ganzen Zahlen jeweils einen Rechenschritt kosten, wie viele Rechenschritte sind nötig, um einen Eintrag der gesuchten Matrix C zu ermitteln?

_____ Wie viele Rechenschritte sind nötig, um alle Einträge von C zu ermitteln? _____ Wie wird dies in \mathcal{O} -Notation angegeben? _____ Können Sie jetzt sagen, wie die Problemkomplexität der Matrixmultiplikation ist? Wenn ja, wie lautet diese (in \mathcal{O} -Notation)?

Aufgabe 8.5 – Komplexität

Folgende drei Probleme seien gegeben:

- Problem A ist in P.
- Problem B ist in NP.
- Problem C ist NP-schwer.

Bewerten Sie folgende Aussagen. Die möglichen Antworten sind: Ist nie korrekt, ist manchmal korrekt, ist immer korrekt.

Problem	nie	manchmal	immer
Die Komplexität von A ist $\mathcal{O}(n^2)$			
A lässt sich schneller lösen als B			
C ist nicht lösbar.			
Lässt sich C in polynomieller Zeit lösen, so auch B.			
Nicht berechenbare Probleme sind genau diejenigen, zu denen noch kein Algorithmus gefunden wurde.			