

# Übungen zur Vorlesung Sequenzanalyse I

Universität Bielefeld, WiSe 2009/2010

Prof. Dr. Jens Stoye · Dipl.-Inform. Nils Hoffmann

<http://wiki.techfak.uni-bielefeld.de/gi/GILectures/2009winter/SequenzAnalyse>

**Blatt 3 vom 29.10.2009**

**Abgabe in einer Woche vor Beginn der Vorlesung.**

## Aufgabe 1 Edit-Operationen

(2 Punkte)

Berechne  $E(\text{AGACCT}, \text{CS}_A\text{CDCS}_G)$  und gib alle Zwischenschritte an.

## Aufgabe 2 Edit-Sequenz und -Distanz

(3 Punkte)

Überführe den String  $x = \text{MEGAPHON}$  in den String  $y = \text{MINITON}$  durch Anwendung der *Edit*-Funktion.

1. Gib eine möglichst kurze Edit-Sequenz an, die  $x$  in  $y$  überführt.
2. Gib eine der längsten Edit-Sequenzen an, die  $x$  in  $y$  überführt.
3. Berechne den Wert der Edit-Distanz  $D(x, y)$ .

## Aufgabe 3 Hammingdistanz

(2 Punkte)

Bei der Translation werden Aminosäuren durch Basentriplets codiert.

*Beispiel:* Methionin wird nur durch AUG codiert, Arginin hingegen wird durch CGU, CGC, CGA, CGG, AGA und AGG codiert.<sup>1</sup>

1. Gegeben seien zwei codierende DNA-Sequenzen der Länge  $n = 3m$ ; ihre Hamming-Distanz betrage  $h \in [0, n]$ . Wie groß kann die Hamming-Distanz  $k$  der zugehörigen Aminosäuresequenzen der Länge  $m$  minimal und maximal sein? Gib einfache Sequenzbeispiele für die Extremfälle an.
2. Jetzt seien zwei Aminosäuresequenzen der Länge  $m$  mit Hammingdistanz  $k$  gegeben. Wie groß kann die Hammingdistanz  $h$  von zugehörigen codierenden DNA-Sequenzen der Länge  $n = 3m$  minimal und maximal sein? Gib wiederum Beispiele an.

## Aufgabe 4 Edit-Distanz

(3 Punkte)

Auf dem letzten Zettel sollte eine JAVA-Klasse `HammingDistance` implementiert werden, die auf zwei Strings gleicher Länge angewendet werden kann.

1. Implementiere für Strings unterschiedlicher Länge eine JAVA-Klasse `EditDistance`, die die in Kapitel 3.6 des Skripts definierte Rekursion zur Berechnung der Edit-Distanz verwendet. Die Distanz wird in der Methode

```
public double getDistance(String a, String b)
```

berechnet.

2. Da es nun zwei Klassen, mit der gleichen Methodensignatur gibt, sollte man die gemeinsame Funktionalität in ein Interface auslagern. Lege dafür ein JAVA-Interface `StringDistance` an, welches als einzige Methode

```
public abstract double getDistance(String a, String b);
```

deklariert. Die Klassen `HammingDistance`, sowie `EditDistance`, müssen nun dieses Interface implementieren.

3. Gegeben sind die Sequenzen  $s_1 = \text{Der Hunne Attila}$ ,  $s_2 = \text{Die Stadt Manila}$  und  $s_3 = \text{Die Henne Ottili}$ . Berechne mit den von dir implementierten Methoden alle paarweisen Hamming- und Edit-Distanzen und gib diese an. Unterscheide dabei Groß- und Kleinbuchstaben, z.B.  $b \neq B$ .

---

<sup>1</sup>Mehr dazu unter: [http://de.wikipedia.org/wiki/Genetischer\\_Code](http://de.wikipedia.org/wiki/Genetischer_Code)