Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, WS 2012/2013
Prof. Dr. Jens Stoye · Nina Luhmann · Linda Sundermann

http://wiki.techfak.uni-bielefeld.de/gi/Teaching/2012winter/SequenzAnalyse

Übungsblatt 5 vom 16.11.2012
Abgabe in zwei Wochen vor Beginn der Vorlesung

Aufgabe 1 (Gapkosten) (2 Punkte)

1. Erkläre, welche Arten von Gapkostenfunktionen es gibt und wie sie berechnet werden.

Aufgabe 2 (Affine Gapkosten) (6 Punkte)

1. Warum sollten die Kosten für eine *gap extension* nicht höher als die Kosten für ein *gap open* gewählt werden, also $e \leq d$?

2. Zeige, dass affine Gapkosten subadditiv sind.

3. Berechne ein optimales globales Alignment mit affinen Gapkosten von den Sequenzen $x =$ `TGAAATCG` und $y =$ `GACG` effizient mit Hilfe des Gotoh-Algorithmus und gib dessen Gesamtscore an. Verwende dabei: Score für Match = 2, Score für Mismatch = 0, Kosten für Gap-open $d = 2$, sowie Kosten für Gap-extension $e = 0.5$.

Aufgabe 3 (Suboptimale Alignments) (7 Punkte)

1. Beschreibe in eigenen Worten, was man unter *überlappenden* Alignments versteht.

2. Warum ist man bei der Bestimmung von suboptimalen Alignments in erster Linie an *nicht überlappenden* Alignments interessiert? Welche Probleme möchte man vermeiden?

3. Beschreibe kurz die Funktionsweise des Waterman-Eggert-Algorithmus.
   Durch welchen Trick kann man in der Praxis die Laufzeit verkürzen?

4. Gegeben sei $x =$ `TACC`, $y =$ `TGAC`, berechne ein lokales Alignment mit dem Smith-Waterman-Algorithmus und das erste nicht überlappende, suboptimale Alignment nach *Waterman-Eggert*. Verwende dazu Scores mit $\text{score}(\mathcal{C}) = 3$, $\text{score}(\mathcal{S}_{a,c}) = 1$ für $a \neq c$ und $\text{score}(\mathcal{I}_c) = \text{score}(\mathcal{D}_c) = -1$.

Aufgabe 4 (Approximatives Stringmatching) (4 Punkte)

1. Finde die Endpositionen aller Vorkommen des Patterns $x =$ `CACA` im Text $y =$ `TTATCACATACCCG` mit maximal $k = 1$ Fehlern. Verwende die Cutoff-Variante von *Sellers'* Algorithmus mit Einheits**kosten** und markiere die *last essential indices*.

2. Gib zu jeder gefundenen Endposition alle zugehörigen Alignments an.

3. Warum macht es Sinn, in der Praxis nicht *alle* gefundenen Endpositionen auszugeben? (Stichwort: *Runs*.) Welche Endpositionen würde man im obigen Beispiel nicht ausgeben?