

Prof. Dr. Knut Reinert
Thimo Wellner
Sascha Meiers

Institut für Informatik
AG Algorithmische Bioinformatik

Algorithmen und Datenstrukturen in der Bioinformatik

6. Programmieraufgabe WS 13

Abgabe Montag, 03.02., 15:00 Uhr per SVN

In dieser Programmieraufgabe werden Sie C-optimale Schnittstellen für ein multiples Alignment berechnen. Mit diesen ließe sich mit ein wenig mehr Aufwand ein vollständiges multiples Alignment erstellen.

- Lesen sie eine beliebige Anzahl von durch Leerzeichen getrennten Sequenzen ein
- Berechnen Sie für alle Paare von Sequenzen s und t die additional cost Matrix, indem Sie
 - die globale Alignmentmatrix D der Sequenzen berechnen,
 - die globale Alignmentmatrix D^{rev} der umgedrehten Sequenzen berechnen und
 - die additional cost Matrix gemäß der Formel $C(i, j) = D(i, j) + D^{rev}(n - i, m - j) - d(s, t)$ berechnen (n, m sind die Längen der Sequenzen. Siehe auch Definition 11.2 im Skript).
 - Beachten Sie dass hier Kosten minimiert werden, und nicht Scores maximiert. Passen Sie ihrern Needleman-Wunsch-Algorithmus entsprechend an. Distanzmaß ist die Edit-Distanz (Gaps oder Mismatches kosten 1).
- Gegeben $c_1 = \lfloor n_1/2 \rfloor$, finden Sie die Schnittpositionen c_2, \dots, c_k mit minimalen Zusatzkosten in dem Sie über die möglichen Schnittpositionen mittels eines rekursiven Branch-and-Bound-Algorithmus enumerieren (siehe auch P-VL Folien)
 - Eine Schittposition ist die Länge der Sequenz links vom Schnitt.
 - Setzen Sie alle paarweisen Gewichte $w_{ij} = 1$.
 - Die Branch-and-Bound-Funktion sollte u.A. die minimalen bisher gefundenen Kosten übergeben bekommen und die rekursive Suche dann abbrechen, wenn die Kosten zu hoch werden.

Ihrem Programm werden mehrere Sequenzen über die Kommandozeile übergeben und sie sollen die *additional costs* sowie ihre gefunden Schnittpositionen ausgeben:

```
sascha@sascha $ ./aufgabe6 CT AGT G
0
1,2,1
```

Praktikumshinweise Beachten Sie die Hinweise unter <https://www.mi.fu-berlin.de/w/ABI/AlDaBiWS13Praktikum>.