



Genomanalyse

3. Übungsblatt WS 2009

Abgabe: Dienstag 3.11.2009 vor der Übung, Besprechung: in der Übung

Bitte geben Sie auf Ihren abgegebenen Lösungen Name und Matrikelnummer an und arbeiten Sie in Gruppen von 2 oder 3 Personen zusammen.

Aufgabe 5: [2+4+2+2=10 Punkte] Overlap Alignment

Beim *overlap alignment* werden Gaps an den Enden nicht bestraft, es darf aber an den Enden immer nur eine Sequenz überhängen (nicht wie beim lokalen Alignment beide Sequenzen):

```
xxXXX   xxXXXxx   XXXxx   XXX
  YYYyy   YYY   yyYYY   yyYYYyy
```

Nur die Großbuchstaben sind aligniert, evtl. mit Gaps. Natürlich kann auch nichts überhängen. Entwickeln Sie, ausgehend von den in der Vorlesung behandelten Methoden, einen Algorithmus, der das Overlap Alignment Problem für zwei Sequenzen mit linearen Gapkosten löst.

Behandeln Sie dabei die folgenden Fragen:

- Rekursionsformel
- Initialisierung
- Wo und wie liest man den optimalen Score aus der Tabelle ab?
- Wo beginnt und endet der Traceback, und wie geht er überhaupt vonstatten?

Als Zusatzfrage (für alle, denen das zu einfach war, freiwillig und ohne Bewertung, aber für Ruhm und Ehre) können Sie dasselbe Problem auch für affine Gapkosten lösen.

Aufgabe 6: [10 Punkte] Longest Increasing Subsequence

Beim *longest increasing subsequence* Problem geht es um das Folgende: Gegeben ist eine Liste von Paaren $(p_i, q_i) \in \mathbb{N}^2, i = 1, \dots, n$. Gesucht ist eine Folge i_1, \dots, i_k von maximaler Länge k , so dass gilt: $j < j' \Rightarrow p_{i_j} < p_{i_{j'}} \wedge q_{i_j} < q_{i_{j'}}$.

Konkret könnten die p und q zum Beispiel die Positionen von verwandten Genen in verschiedenen Genomen sein. Gesucht ist dann eine möglichst große Teilmenge von Genen, die in beiden Genomen in derselben Reihenfolge stehen, was manchmal eine durchaus interessante Frage ist. ;-)

Entwickeln Sie einen Algorithmus für dieses Problem. Verwenden Sie dynamische Programmierung. Machen Sie sich wegen der Laufzeit wenig Sorgen, tatsächlich geht es in $O(n \log n)$, aber Ihre Lösung muss nicht ganz so schnell sein.

Zusatzfrage (außerhalb Bewertung): Die Paare sind jetzt gewichtet: $(p_i, q_i) \in \mathbb{N}^2, w_i \in \mathbb{N}$, und das Ziel ist, $\sum_{j=1}^k w_{i_j}$ für die ausgewählten Paare zu maximieren.
