Prof. Dr. Knut Reinert
Enrico Siragusa
Sascha Meiers
Christoph Hartmann

Institut für Informatik
AG Algorithmische Bioinformatik

Freie Universität Berlin

Algorithmen und Datenstrukturen in der Bioinformatik
Fünftes Übungsblatt WS 10/11

Abgabe Montag, 21.11., 15:00 Uhr

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer:

Niveau I

Aufgabe 1: Alignment mit affinen Gapkosten

Benutzen Sie den Algorithmus von **Gotoh** um das optimale globale Alignment der Sequenzen $S_1 =$ `AT` und $S_2 =$ `AGGT` zu finden. Die Scores seien: Match 5, Mismatch -4, Gap opening -8 und Gap Extension -1

		A	**G**	**G**	**T**
	M I_x I_y				
A	M I_x I_y				
T	M I_x I_y				

Niveau II

Aufgabe 2: Hirschberg-Algorithmus, Teil I

Um ein Alignment mit linearem Speicherplatzaufwand zu finden, bedient man sich eines Dïvide-and-Conquer-Ansatzes. In jedem Schritt teilt man die beiden Sequenzen an der optimalen Stelle, so lange bis man nur noch maximal ein Zeichen in jeder Teilsequenz hat. Durch Aneinanderfügen dieser Teilalignments erhält man das optimale Alignment. Im Skript (Kapitel 3.21) und auf Wikipedia werden für das Auffinden der optimalen Schnittposition zwei verschiedene Ansätze vorgestellt.

a) Lesen Sie die beiden oben verlinkten Methoden und versuchen Sie die verschiedenen Ansätze zu verstehen.

b) Benutzen Sie den von Ihnen favorisierten Ansatz, um herauszufinden an welchen Positionen man die beiden Sequenzen $S_1 =$ `ACGGAATT` und $S_2 =$ `CCGCATGA` teilen kann, sodaß das Aneinanderfügen der beiden Teilalignments ein optimales Alignment bildet. Scores: Match 5, Mismatch -3, Gap -2.

Aufgabe 3: Hirschberg-Algorithmus, Teil II

Gegeben sei die Funktion

```
int AlignmentPos(int aStart, int aEnd, int bStart, int bEnd);
```

die ein globales zwischen $A_{\texttt{aStart,aEnd}}$ und $B_{\texttt{bStart,bEnd}}$ (ohne Traceback) berechnet (d.h. zwischen zwei Teilsequenzen von A und B, die durch die gegebenen Indizes begrenzt sind).

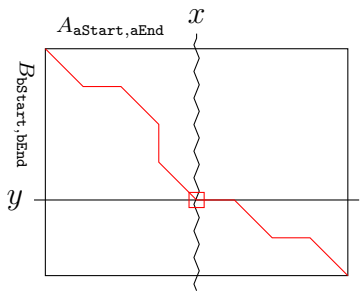

Abbildung 1: Alignment der Teilsequenzen $A_{\texttt{aStart,aEnd}}$ und $B_{\texttt{bStart,bEnd}}$ mit Traceback (in rot).

Wie man in Abbildung **??** sehen kann, läuft das optimale Alignment durch den Punkt (x, y). x ist fest gegeben als $x = \lfloor \frac{\texttt{aEnd}-\texttt{aStart}}{2} \rfloor$. y ergibt sich dann automatisch durch das Alignment. Die Funktion `AlignmentPos` gibt diesen Wert y zurück.

Gesucht ist eine rekursive Funktion, die Divide-And-Conquer anwendet, um mithilfe der obigen Funktion das komplette Alignment der Sequenzen A und B zu berechnen (siehe Hirschberg-Algorithmus, Skript S. 3022–3024).

Hinweis: Um die *formatierte Ausgabe* des Alignments müssen Sie sich keine Gedanken machen. Das Problem lässt sich dann rekursiv in drei Zeilen Pseudocode lösen.

Programmieraufgabe (Abgabe Montag, 28.11.2010, 15:00)

P-Aufgabe 3: Implement the Smith-Waterman algorithm in C++ and output one optimal local alignment along with its score. Both sequences x and y to be aligned have to be taken from command line. The exercise has to follow the following input/output format:

```
user@tetrahymena:~$ ./aufgabe3 IMISSMISSISSIPPI MYMISSISAHIPPIE
MISSISS-IPPI
||||||  ||||
MISSISAHIPPI
score:42
user@tetrahymena:~$
```

Hints: Use the following scoring scheme: Match: +5; Mismatch/Insert/Delete: -4. When more than one pair (i', j') leads to the optimum value $F(i, j)$, make your backtracking choice in the following order:

a) Mismatch ($\nwarrow$)

b) Insert in y ($\leftarrow$)

When more alignments are co-optimal, choose the leftmost one, i.e. that one with minimum (i', j').

Remember the material at `https://www.mi.fu-berlin.de/w/ABI/AlDaBiWS11`.