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

Institut für Informatik
AG Algorithmische Bioinformatik

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

Abgabe Montag, 12.12., 15:00 Uhr

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer:

Niveau I

Aufgabe 1: BLAST

Gegeben ist ein Pattern $P =$ `HILAM` sowie eine Datenbank $D =$ `ILHIVAVHIIM`

a) Finden Sie alle k-mere, die zu einem beliebigen k-mer aus P einen Score von mindestens 13 aufweisen. Die Fenstergröße ist auf $k = 3$ festgelegt. Benutzen Sie zum Scoring die BLOSUM62-Matrix.

b) Erstellen Sie aus der Liste der k-mere einen Aho-Corasick-Automaten. Markieren Sie dabei die finalen Knoten und zeichnen Sie alle nicht-trivialen Supply-Links ein, also jene die nicht zur Wurzel zurückgehen.

c) Suchen Sie die *seeds*, also alle Vorkommen eines der Pattern in der Datenbank D mit dem Aho-Corasick Algorithmus. Als Ergebnis genügt es die Stellen zu markieren.

d) Erweitern sie die gefundenen seeds nach rechts oder links, sofern dies ohne gaps möglich ist. Welcher Bereich liefert das beste lokale Alignment ohne gaps?

Aufgabe 2: Aho-Corasick Automaten und Pseudocodes

Mit Hilfe des *Aho-Corasick Automaten* wird in BLAST das multiple Stringmatching-Problem gelöst. Der Automat basiert auf einem *Trie*, der durch sogenannte *Supply-Links* erweitert wird. Der Supply-Link eines Knotens v zeigt auf das längste Suffix des Textes

a) Machen Sie sich mit den Pseudocodes zur Erstellung des Tries (Kapitel 5.7, Seite 5004) und zur Erweiterung des Tries durch Supply-Links (Kapitel 5.8, Seite 5007) vertraut. Erläutern Sie, was die Symbole δ und S bedeuten.

b) Bei der Suche kann es vorkommen, dass wenn ein Suchwort p gefunden wird, gleichzeitig auch ein anderes Suchwort s enthalten ist. In diesem Fall ist s ein Suffix von p. Um nun beim Finden von p auch einen Treffer für s auszugeben, muss sichergestellt werden, dass s in der Menge der finalen Suchwörter von p enthalten ist, also dass $s \in F(p)$. Wie wird das im Algorithmus umgesetzt?

c) Schreiben Sie den Pseudocode für die Suche nach einer Reihe von Pattern P in einem Text $T = T_1 \dots T_n$. Die Suchwörter sind gegeben mit $P = p^1, \dots, p^r$, wobei $p^i = p^i_1 p^i_2 \dots p^i_{m_i}$ für $1 \leq i \leq r$. Sie können annehmen, dass diese Pattern bereits in einem fertigen Trie $trie = (\delta, F, S)$ vorliegen. Der Algorithmus soll alle Positionen in T bestimmen, an denen mindestens eines der Pattern aus P auftaucht und sich an der Terminologie der beiden erwähnten Pseudocodes orientieren. Sie können zur Ausgabe einfach ein *output* statement verwenden.