Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, WS 2015/2016
Prof. Dr. Jens Stoye · M.Sc. Linda Sundermann

http://wiki.techfak.uni-bielefeld.de/gi/Teaching/2015winter/SequenzAnalyse

Übungsblatt 6 vom 08.12.2015
Abgabe in einer Woche vor Beginn der Vorlesung.

Aufgabe 1 (Maximale Repeats) (3+3* Punkte)

Lies den Abschnitt 7.6.3 im Skript über das effiziente Auffinden von maximalen Repeats in einem Text s mit Hilfe des Suffixbaums von s.

1. Stelle den Suffixbaum von $s =$ `ATACATGGCATATG` auf.
2. **Satz:** In jedem String der Länge n gibt es höchstens n maximale Repeats.

   Argumentiere unter Berücksichtigung des Suffixbaums, warum diese Aussage korrekt ist. Bedenke: Es stimmt nicht, dass an jeder Position nur ein maximales Repeat beginnen oder enden kann.
3. Finde alle maximalen Repeats in $s =$ `ATACATGGCATATG` unter Verwendung des im Skript geschilderten Algorithmus. Beschreibe dein Vorgehen beim Annotieren des Suffixbaums aus Aufgabenteil 1.

Lösung

1. ...
2. Wir betrachten den Suffixbaum eines Strings s der Länge n. Dieser hat höchstens n innere Knoten, deren Pfadlabel den Substrings von s entsprechen, die mindestens zweimal in s vorkommen. Somit kann s auch höchstens n maximale Repeats besitzen.

Aufgabe 2 (Maximal-Matches Distanz) (2 Punkte)

Berechne die Links-Rechts-Partition von $t =$ ATGCATAATG bezüglich s aus Aufgabe 1. Inwiefern ist der Suffixbaum von s dazu nützlich? Welche Maximal-Matches Distanz $\delta(t\|s)$ ergibt sich und warum?

Aufgabe 3 (Manber-Myers Algorithmus) (4 Punkte)

1. Führe den Manber-Myers Algorithmus schrittweise für den String $s =$ `GATAAAGATAAGA` aus und gib das Suffixarray *pos* für s an.
2. Wie viele Phasen braucht man für einen String der Länge 13 maximal?

Aufgabe 4 (Suffixarray) (4 Punkte)

Im Skript ist in Abschnitt 8.3.3 angegeben, wie sich die Arrays *rank* und *lcp* aus dem Suffixarray *pos* effizient berechnen lassen.

1. Implementiere zwei Funktionen, die das *rank*- und *lcp*-Array berechnen, wenn das Array *pos* gegeben ist. Die Funktionen sollen so in ein Programm eingebettet sein, dass der Benutzer nur das *pos*-Array übergeben muss.

   Verwende eine Programmiersprache, die mit deinem Tutor abgesprochen ist und sende ihm deinen Quellcode per Email zu. Beachte, dass die Indizierung mit 0 beginnen soll.
2. Berechne mit deinem Programm das *rank*- und das *lcp*-Array für $s =$ `ATACAATCTCTAT` und gib diese an.

Aufgabe 5 (Geometrische Projektion des Alignments) (2 Punkte)

Gegeben sei das multiple Alignment der Sequenzen $s_1 =$ `TATA`, $s_2 =$ `CTAT`, $s_3 =$ `GTAA`:

```
-  T  A  T  A
C  T  A  T  -
G  T  A  -  A
```

Zeichne das Alignment im 3-dimensionalen Raum und die jeweiligen Projektionen auf den 2-dimensionalen Unterräumen in die Abbildung auf der Rückseite ein.