Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, SS 2018
Dr. Daniel Dörr

https://gi.cebitec.uni-bielefeld.de/teaching/2018summer/sa

Übungsblatt 8 vom 07.06.2018
Abgabe 14.06.2018

Aufgabe 1 (Manber-Myers Algorithmus) (3 Punkte)

1. Führe den Manber-Myers Algorithmus schrittweise für den String $s =$ `CTGTTTCTGTTC` aus und gib das Suffixarray *pos* für s an.

2. Wie viele Phasen braucht man für einen String der Länge 12 maximal?

Aufgabe 2 (Suffixarray) (6 Punkte)
Verwende zur Implementierung der in folgenden Teilaufgaben beschriebenen Funktionen die Programmiersprache Java und sende ihm deinen Quellcode per Email zu.

1. Implementiere eine Funktion, die das *pos*-Array für eine beliebige DNA-Sequenz berechnet. Beachte, dass $\$ < A < C < G < T$ und die Indizierung mit 0 beginnen soll.

2. 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.

3. Wie lauten das *rank*- und *lcp*-Array für $s =$ `CTACCCTCACAATCCATTCCTTATCTCTCC`? Gib einen Beispielaufruf für dein Programm mit dem *pos*-Array von s an. Falls du die Programme nicht implementiert hast, führe die Berechnung der Arrays von Hand aus.

Aufgabe 3 (Burrows-Wheeler Transformation) (4 Punkte)
Gegeben sei der String $t =$ `TGCGCTGCGGCTCG$`.

1. Was ist die Burrows-Wheeler Transformation? Beschreibe ihre zentrale Idee in Bezug auf die Eigenschaft natürlicher Sprache.

2. Berechne die Burrows-Wheeler Transformierte $bwt(t)$ für t.

3. Schreibe $rle(bwt(t))$ als komprimierten String mit Hilfe von *run-length encoding* auf. Fasse dabei nur Buchstaben zusammen, die mindestens 3 mal hintereinander vorkommen.