

# Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, SS 2017

Prof. Dr. Jens Stoye · M.Sc. Tizian Schulz

<https://gi.cebitec.uni-bielefeld.de/teaching/2017summer/sa>

**Übungsblatt 8 vom 29.06.2017**

**Abgabe am Dienstag, den 11.07.2017**

## Aufgabe 1 (Suffixarray)

(6 Punkte)

Verwende für die folgenden Teilaufgaben eine Programmiersprache, die mit deinem Tutor abgesprochen ist 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  $s = \text{TATGTTAGAGATA}$ ? 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 2 (Burrows-Wheeler Transformation)

(4 Punkte)

Gegeben sei der String  $t = \text{ATGTGATGGTGA}\$$ .

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.