Prof. Dr. Knut Reinert
Sandro Andreotti
FB Mathematik und Informatik

Freie Universität Berlin

# Fortgeschrittene Algorithmen in der Bioinformatik (P4): Sequence and Structure Analysis Abschlussklausur SS 08

Name: Matrikelnummer:

*Viel Erfolg!*

| A1 | A2 | A3 | A4 | A5 | A6 | $\sum$ |
|---|---|---|---|---|---|---|
| 17 | 12 | 14 | 15 | 10 | 22 | 90 |
| | | | | | | |

## Aufgabe 1: Ukkonen, Myers Bitvector — 6 + 7 + 4 = 17 Punkte

**(a)** Beschreiben Sie den Idee von Ukkonen zum Berechnen eines Edit-Distanz alignments mit maximal $k$ Fehlern

**(b)** Argumentieren Sie, warum in Ukkonens Algorithmus die Aktualisierung des Zählers $lact$, der auf die letzte aktive Zelle zeigt, amortisiert nur Laufzeit $O(n)$ hat.

**(c)** Welche Bitvektoren benutzt der Myers Bitvektor-Algorithmus zum Berechnen eines Edit-Distanz alignments?

## Aufgabe 2: PEX, Pidgeonhole Principle — 3 + 3 + 6 = 12 Punkte

Ein Text $T$ der Länge 1000 enthält ein Vorkommnis $Occ$ eines Patterns $P$ der Länge 50 mit höchstens 5 Fehlern. In wieviele Stücke muss man $P$ zerlegen (also $P = P_1P_2 \dots P_k$), um sicherzustellen, dass ...

**(a)** $T$ eines der Stücke exakt (ohne Fehler) enthält?

**(b)** $T$ eines der Stücke mit höchstens zwei Fehlern enthält?

**(c)** Beweisen sie das Lemma:
Sei $Occ$ ein approximatives Vorkommen eines strings in einem pattern $P$ mit $k$ Fehlern. Sei weiter $P = p^1, \dots, p^j$ die Konkatenation von Teilen von $P$ sowie $a_1, \dots, a_j$ nichtnegative, ganze Zahlen so dass $A = \sum_{i=1}^{j} a_i$. Dann existiert ein $i \in 1, \dots, j$, so dass es einen substring in $Occ$ gibt der $p^i$ mit $\lfloor a_i k/A \rfloor$ Fehlern matched.

## Aufgabe 3: Komparative RNA-Analyse — 6 + 3 + 5 = 14 Punkte

In der Vorlesung wurde der LARA-Algorithmus zur komparativen RNA-Analyse besprochen der auf einer ILP Formulierung beruht.

**(a)** Geben Sie die ILP Formulierung an. Erläutern Sie die Bedeutung der Variablen und Ungleichungen.

**(b)** Lara verwendet Lagrangian Relaxierung um das ILP zu lösen. Was ist die Idee von Langragian Relaxation?

**(c)** Welche Ungleichungen werden bei Lara relaxiert?

---

**Aufgabe 4: Motif Finding, EM-Algorithmus** **3 + 12 = 15 Punkte**

---

**(a)** Definieren sei das "planted $(l, d)$-motif" problem.

**(b)** Führen Sie einen Schritt des EM-Algorithmus so wie in der Vorlesung durch. Gegeben seien die Beobachtungen $x = x_1, x_2, x_3$:

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $x_1$ | A | **C** | **A** | **G** | C | A |
| $x_2$ | A | G | G | **C** | **A** | **G** |
| $x_3$ | T | **C** | **A** | **G** | T | C |

Berechnen Sie die fehlenden Startpositioten des verborgenen Motifs und repräsentieren Sie dise durch eine Matrix $w$ wobei $w_{ij}$ die Wahrscheinlichkeit ist, dass das Motif an Position $j$ in Sequenz $i$ anfängt. Gegeben sei dabei das anfängliche Motif:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| A | 0.25 | 0.1 | 0.5 | 0.2 |
| C | 0.25 | 0.4 | 0.2 | 0.1 |
| G | 0.25 | 0.2 | 0.1 | 0.5 |
| T | 0.25 | 0.3 | 0.2 | 0.2 |

Berechnen Sie $w$.

---

**Aufgabe 5: Suffix arrays** **3 + 7 = 10 Punkte**

---

**(a)** Definieren sie ein suffix array für einen string und die lcp Tabelle.

**(b)** Mit der lcp Tabelle kann man in dem Suffix array schnell suche. Allerdings gibt es $O(n^2)$ viele lcp Werte. Erläutern Sie, dass man nicht alle diese Werte braucht. Wieviele braucht man?

---

**Aufgabe 6: Chaining** **5 + 7 + 5 + 5 = 22 Punkte**

---

**(a)** Welche Operationen unterstützt die beim Chaining-Algorithmus verwendete Datenstruktur? Beschreiben Sie jeweils kurz, was sie bewirken bzw. was ihr Ergebnis ist.

**(b)** Beschreiben Sie den Chaining-Algorithmus für $L_1$-Gap-Kosten.

**(c)** Die $L_1$ Kosten sind nicht sehr realistisch. In der Vorlesung wurde die "sum-of-pair" Kosten ebenfalls vorgestellt. Wie sind Sie definiert?

**(d)** Beschreiben Sie die Idee, wie der Algorithmus die sum-of-pair Kosten mit Hilfe von RMQ berechnet? (Hinweis: Man kann eine RMQ nicht direkt sondern erst nach einer Umformung anwenden).