Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Volker Heun

Wintersemester 2019/20
Semestralklausur
13. Februar 2020

Algorithmische Bioinformatik II

Vorname	Name	Matrikelnummer

Reihe	Platz	Unterschrift

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses dieser Semestralklausur unter Verwendung meiner Matrikelnummer im Internet zu. *Ja* ☐ *Nein* ☐

_____
(Unterschrift)

Allgemeine Hinweise zur Semestralklausur

- Vor der Prüfung ist diese Seite mit Vornamen, Namen, Matrikelnummer, Reihe und Platz **leserlich mit Druckbuchstaben** zu versehen und zu unterschreiben.
- Bitte **nicht** in *roter* oder *grüner* Farbe bzw. **nicht** mit Bleistift schreiben.
- Der Studentenausweis und ein amtlicher Lichtbildausweis sind bereit zu halten.
- Die reine Bearbeitungszeit beträgt **120 Minuten**.
- Es sind insgesamt 40 Punkte zu erreichen, zum Bestehen sind 17 Punkte nötig.

Viel Erfolg!

Hörsaal verlassen von ..... bis ..... von ..... bis .....

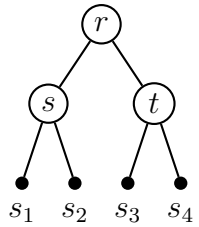
Vorzeitig abgegeben um .....

	Hz	A1	A2	A3	A4	A5	$\sum$
Erstkorrektur							
Nachkorrektur							
Zweitprüfer							

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 1 (8 Punkte)

Berechne für den rechts angegebenen vollständigen Baum ein optimales **uniform** geliftetes Alignment mittels der dynamischen Programmierung.



d	s_1	s_2	s_3	s_4
s_1	0	1	5	6
s_2		0	2	7
s_3			0	3
s_4				0

Lösungsskizze (nicht ausreichend für die volle Punktzahl)

$$
\begin{aligned}
D[s,s_1] &= (D[s_1,s_1])+(d(s_1,s_2)+D[s_2,s_2]) = 0+(1+1) = 1\\
D[s,s_2] &= (d(s_2,s_1)+D[s_1,s_1])+(D[s_2,s_2] = (1+0)+0 = 1\\
D[t,s_3] &= (D[s_3,s_3])+(d(s_3,s_4)+D[s_4,s_4]) = 0+(3+0) = 3\\
D[t,s_4] &= (D[s_3,s_3]+d(s_4,s_3))+(D[s_4,s_4]) = (0+3)+0 = 3\\
D[r,s_1] &= (D[s,s_1])+(d(s_1,s_3)+D[t,s_3]) = 1+(5+3) = 9\\
D[r,s_2] &= (D[s,s_2])+(d(s_2,s_4)+D[t,s_4]) = 1+(7+3) = 11\\
D[r,s_3] &= (d(s_3,s_1)+D[s,s_1])+(D[t,s_3]) = (5+1)+3 = 9\\
D[r,s_4] &= (d(s_4,s_2)+D[s,s_2])+(D[t,s_4]) = (7+1)+3 = 11
\end{aligned}
$$

Damit sind die Lösungen:

$$
\begin{aligned}
r &= s_1|s_3\\
s &= s_1\\
t &= s_3
\end{aligned}
$$

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 2 (8 Punkte)

Verwende den Algorithmus von Carrillo und Lipman zur Berechnung eines Sequenzen-Alignments zwischen zwei Sequenzen $s = TATA$ und $t = ATG$. Hierzu sind für das Distanzmaß die **Gap-Kosten** von 3 und **Mismatch-Kosten** von 2 zu verwenden. Die **globale obere Schranke** für die Distanz von s und t ist mit 9 vorgegeben.

Hinweis: In der Übung wurde dies für 3 oder mehr Sequenzen implementiert, natürlich funktioniert das Verfahren auch mit nur 2 Sequenzen.

Gib die kombinierte **Prefix-/Suffix-Matrix** $P + S$ und dessen Herleitung an und **markiere alle Zellen**, die in den **Heap** aufgenommen wurden. Gib dabei ebenfalls die Berechnung der verwendeten **obere Schranke** im Relevanz-Test für das Sequenzpaar (s, t) an.

Lösungsskizze (nicht ausreichend für die volle Punktzahl)

P		A	T	G
	0	3	6	9
T	3	2	3	6
A	6	3	4	5
T	9	6	3	6
A	12	9	6	5

S		A	T	G
	5	8	11	12
T	2	5	8	9
A	5	2	5	6
T	6	5	2	3
A	9	6	3	0

$P+S$		A	T	G
	5	*11*	17	21
T	*5*	*7*	*11*	15
A	*11*	*5*	*9*	*11*
T	15	*11*	*5*	*9*
A	21	15	*9*	*5*

Die in den Heap aufgenommen Elemente sind rot bzw. kursiv dargestellt.

Die im Relevanz-Test verwendete obere Schranke für s und t lautet:

$$C_{s,t} := C - \sum_{(s_i,s_j)\neq(s,t)} d(s_i, s_j) = C - 0 = 9 - 0 = 9.$$

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 3 (8 Punkte)

Bestimme für die folgenden Blöcke von Sequenzen die zugehörigen Häufigkeiten $H(a, b)$ für die BLOSUM50-Matrix.

$$
\begin{array}{ll}
s_1^{(1)} = \text{CABCC} & s_1^{(2)} = \text{CBBCACB} \\
s_2^{(1)} = \text{BAACB} & s_2^{(2)} = \text{CCBCABC} \\
s_3^{(1)} = \text{CCABB} & s_3^{(2)} = \text{BCBBABB} \\
s_4^{(1)} = \text{CAACB} & s_4^{(2)} = \text{ABAACBB}
\end{array}
$$

Lösungsskizze (nicht ausreichend für die volle Punktzahl)

Die Partitionierung nach mindestens 50%-Sequenzähnlichkeit ergibt:

$$
\begin{array}{lrcl}
\text{Block 1:} & [1:4] & = & [1:4] \\
\text{Block 2:} & [1:4] & = & [1:3] \cup \{4\}
\end{array}
$$

Dabei sind im ersten Block u.a. folgende Ähnlichkeiten von mindestens 50%: $s_1^{(1)}$ mit $s_4^{(1)}$, $s_2^{(1)}$ mit $s_4^{(1)}$ und $s_3^{(1)}$ mit $s_4^{(1)}$. Im zweiten Block sind u.a. folgende Ähnlichkeiten von mindestens 50%: $s_1^{(2)}$ mit $s_2^{(2)}$ und $s_2^{(2)}$ mit $s_3^{(2)}$, aber nicht $s_4^{(2)}$ zu einer anderen Sequenz im zweiten Block..

Somit ist nur Block 2 auszuwerten:

$$
\begin{aligned}
H(A,A) &= \frac{0+0+0+0+0+0+0}{3 \cdot 1} = \frac{0}{3} = 0 \\
H(A,B) &= \frac{1+0+3+1+0+0+0}{3 \cdot 1} = \frac{5}{3} = 1.\overline{6} \\
H(A,C) &= \frac{2+0+0+2+3+0+0}{3 \cdot 1} = \frac{7}{3} = 2.\overline{3} \\
H(B,B) &= \frac{0+2+0+0+0+4+4}{3 \cdot 1} = \frac{10}{3} = 3.\overline{3} \\
H(B,C) &= \frac{0+2+0+0+0+1+1}{3 \cdot 1} = \frac{4}{3} = 1.\overline{3} \\
H(C,C) &= \frac{0+0+0+0+0+0+0}{3 \cdot 1} = \frac{0}{3} = 0
\end{aligned}
$$

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 4 (8 Punkte)

Wir betrachten eine Münze, wobei mit Wahrscheinlichkeit $p \in (0, 1]$ Kopf erscheint und und mit Wahrscheinlichkeit $1 - p$ Zahl. Sei X die Zufallsvariable, die zählt, wie oft die Münze geworfen werden muss bis Kopf erscheint, dann gilt

$$\mathrm{Ws}[X = N] = p \cdot (1 - p)^{N-1}.$$

a) Gib die allgemeinen Formeln sowohl für den Maximum-Likelihood-Schätzer als auch den Maximum-A-Posteriori-Schätzer an.

b) Angenommen die Münze wurde N-mal geworfen, bis das erste Mal Kopf erschien. Bestimme die Likelihood-Funktion für dieses Ergebnis und gib dann den Maximum-Likelihood-Schätzer für p an.

c) Angenommen die Münze wurde N-mal geworfen, bis das erste Mal Kopf erschien. Bestimme die Posteriori-Wahrscheinlichkeit für dieses Ergebnis bezüglich des Parameterraums $p \in (0, 1]$, wobei der Prior $f(p) = 2p$ ist und gib dann den Maximum-A-Posteriori-Schätzer für p an.

Lösungsskizze (nicht ausreichend für die volle Punktzahl)

a) Der Maximum-Likelihood-Schätzer ist gegeben durch $\operatorname{argmax}\{L(p) : p \in (0, 1]\}$, wobei $L(p) = \mathrm{Ws}[x \mid p]$ ist (hier gilt also $L(p) = \mathrm{Ws}[N \mid p]$).

Der Maximum-A-Posteriori-Schätzer ist gegeben durch $\operatorname{argmax}\{f(p \mid x) : p \in (0, 1]\}$, wobei $f(p \mid x)$ die Posteriori-Wahrscheinlichkeit mit $f(p \mid x) = \frac{f(p) \cdot \mathrm{Ws}[X=x|p]}{\mathrm{Ws}[x]}$ ist (hier gilt also $f(p \mid N) = \frac{f(p) \cdot \mathrm{Ws}[X=N|p]}{\mathrm{Ws}[N]}$, wobei die Evidenz $\mathrm{Ws}[N]$ unabhängig von p ist).

b) Es gilt:

$$L(p) = \mathrm{Ws}[N \mid p] = p \cdot (1 - p)^{N-1}$$

Um das Extremum zu bestimmen, leiten wird die Log-Likelihoodfunktion nach p ab:

$$\begin{aligned}
\frac{d \ln(L(p))}{dp} &= \frac{d}{dp} \ln\left(p \cdot (1 - p)^{N-1}\right) \\
&= \frac{d}{dp} \left(\ln(p) + (N - 1) \ln(1 - p)\right) \\
&= \frac{d \ln(p)}{dp} + (N - 1) \frac{d \ln(1 - p)}{dp} \\
&= \frac{1}{p} - \frac{N - 1}{1 - p}
\end{aligned}$$

Für das Maximum muss $\frac{1}{p} - \frac{N-1}{1-p} = 0$ sein, also $(1 - p) - p(N - 1) = 0$ und somit $pN = 1$. Somit ist $p = \frac{1}{N}$ der Maximumlikelihood-Schätzer, da für $p \in \{0, 1\}$ gilt, dass $L(p) = 0$, und somit bei $\frac{1}{N}$ das Maximum angenommen werden muss. Alternativ kann man die zweite Ableitung prüfen.

c) Es gilt:

$$f(p \mid N) = \frac{f(p) \cdot \mathrm{Ws}[X = N \mid p]}{\mathrm{Ws}[N]} = \frac{2p \cdot p \cdot (1-p)^{N-1}}{\mathrm{Ws}[N]}$$

Um das Extremum zu bestimmen, logarithmieren wir die Gleichung und leiten nach p ab:

$$\begin{aligned}
\frac{d \ln(L(p))}{dp} &= \frac{d}{dp} \ln\left(\frac{2p \cdot p \cdot (1-p)^{N-1}}{\mathrm{Ws}[N]}\right) \\
&= \frac{d}{dp}\left(\ln(2) + 2\ln(p) + (N-1)\ln(1-p) - \ln(\mathrm{Ws}[N])\right) \\
&= 2\frac{d \ln(p)}{dp} + (N-1)\frac{d \ln(1-p)}{dp} - \frac{d\ \mathrm{Ws}[X=N]}{dp} \\
&\quad \text{da } \mathrm{Ws}[X=N] \text{ nach Definition der Evidenz unabhängig von } p \\
&= \frac{2}{p} - \frac{N-1}{1-p}
\end{aligned}$$

Für das Maximum muss $\frac{2}{p} - \frac{N-1}{1-p} = 0$ sein, also $2(1-p) - p(N-1) = 0$ und somit $pN = 2 - p$. Somit ist $p = \frac{2}{N+1}$ der MAP-Schätzer, da für $p \in \{0, 1\}$ gilt, dass $f(p \mid N) = 0$, und somit bei $\frac{2}{N+1}$ das Maximum angenommen werden muss. Alternativ kann man die zweite Ableitung prüfen.

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 5 (8 Punkte)

MAX3CUT

Eingabe: Ein ungerichteter Graph $G = (V, E)$
Lösung: Eine Partition V_1, V_2, V_3 von V, d.h. $V_1 \cup V_2 \cup V_3 = V$ und $V_i \cap V_j = \emptyset$ für alle $i \neq j \in [1:3]$
Optimum: Maximiere $\sum_{i=1}^{3} \sum_{j=i+1}^{3} |(V_i \times V_j) \cap E|$.

Hierbei ist $(V_i \times V_j) = \{\{v, w\} : v \in V_i \wedge w \in V_j\}$.

Anschaulich ist die Anzahl der Kanten zu maximieren, die zwischen den Mengen der Partition der Knoten verlaufen.

a) Zeige, dass MAX3CUT $\in \mathcal{NPO}$.

b) Konstruiere eine polynomielle 3-Approximation für MAX3CUT.

Hinweis: Korrektheitsbeweise und Laufzeitanalyse nicht vergessen.

Lösungsskizze (nicht ausreichend für die volle Punktzahl)

a) Zuerst muss in polynomieller Zeit entscheidbar sein, ob die Eingabe einen ungerichteten Graphen beschreibt. Ein ungerichteter Graph mit n Knoten besitzt offensichtlich eine Eingabegröße von $s(n) = \Omega(n)$. Mit den üblichen Realisierungen von Graphen ist das sicherlich in Zeit $O(|V|^2) = O(n^2) = O((s(n))^2)$ möglich.

   Weiter muss gezeigt werden, dass eine Lösung polynomiell in der Eingabegröße beschränkt ist. Für jede Lösung $(V_1, V_2, V_3) \subseteq V^3$ gilt jedoch, dass deren Beschreibung in $O(|V|) = O(n) = O(s(n))$ möglich ist.

   Weiterhin muss das Maß einer Lösung in polynomieller Zeit berechenbar sein. Hier ist das Maß einer Lösung $(V_1, V_2, V_3) \subseteq V^3$ die Anzahl der Kanten E, die zwischen verschiedenen Teilmengen verläuft. Dies ist durch vollständige Enumeration der Kanten ($|E| = O(n^2)$) und Durchsuchen der Endpunkte in (V_1, V_2, V_3) pro Kanten in zeit $O(n)$ möglich. Insgesamt ist der Zeitbedarf $O(n^3) = O((s(n))^3)$, das Maß also in polynomieller Zeit berechenbar.

b) Wir durchlaufen alle Knoten des Graphen in beliebiger Reihenfolge (Durchlaufen einer Adjazenzmatrix oder von Adjazenzlisten). Zu Beginn sind V_1', V_2' und V_3' leere Mengen. Für jede Knoten nehmen wir den Knoten in die Menge V_c', für die die meisten Kanten zwischen V_c' und V_i' mit $i \in [1:3] \setminus \{c\}$ verlaufen. Die Kanten zu Knoten in $V \setminus (V_1' \cup V_2 \cup V_3')$ sind dabei irrelevant.

   Die Laufzeit ist sicherlich polynomiell (bspw. $O(n^2)$ bei Verwendung einer Adjazenzmatrix und einem Vektor zur Darstellung der Menge V').

   Nach Konstruktion bildet das Mengensystem (V_1', V_2', V_3') immer eine Partition. Zum Schluss ist (V_1', V_2', V_3') die gesuchte Partition. Wenn der i-te Knoten $v \in V_c'$ für ein $c \in [1:3]$ aufgenommen wird, ist die Anzahl C der geschnitten Kanten lokal maximal. In einer optimalen Lösung werden maximal $i - 1$ Kanten geschnitten, bei der Konstruktion der Partition mindestens $\frac{i=1}{3}$. Da der Graph ungerichtet ist, werden am Endes das Algorithmus immer mindestens ein Drittel der zu v adjazenten Kanten durch die Partition $(V_1', V_2', V'3)$ geschnitten. Damit ist die Approximationsgüte 3.