Algorithmen und Datenstrukturen (für Bioinformatik)

Freie Universität Berlin, Institut für Informatik
Dr. Clemens Gröpl
Wintersemester 2004/2005

Klausur

21. Februar 2005

Name, Vorname: (A)

Matrikelnummer:

1		4
2		10
3		6
4		10
5		5
6		5
7		10
8		10
9		6
10		10

11		8
12		8
13		9
14		3
15		6
16		8
17		12
Σ		130

Note:

Aufgabe 1. 2+(1+1)=4 Punkte

(a) Erklären Sie kurz den Unterschied zwischen Hamming- und Editdistanz.

(b) In der Vorlesung wurde bei affinen Gapkosten die *gap opening penalty* mit $d \geq 0$ und die *gap extension penalty* mit $e \geq 0$ bezeichnet. Nennen Sie jeweils ein biologisches Anwendungsbeispiel, wo man d und e zweckmäßigerweise so wählen wird, dass gilt:

(i) $d > e$:

(ii) $d < e$:

Aufgabe 2. 10 Punkte

Berechnen Sie für die Sequenzen $S =$ `gtac` (horizontal) und $T =$ `agcta` (vertikal) ein lokales Alignment nach dem Algorithmus von Smith-Waterman. Ein Match trägt 3 zum Score des Alignments bei, ein Mismatch -5 und ein Space -2.

1. Berechnen Sie die DP-*Matrix*, und
2. tragen Sie dabei jeweils *alle* Traceback-Kanten ein (also eventuell mehrere pro Zelle).
3. Heben Sie *die beim Traceback benutzten* Kanten besonders hervor, und
4. notieren Sie das resultierende *lokale Alignment*. Geben Sie auch die jeweiligen *Start- und End-Positionen* in den beiden Sequenzen an. Zum Beispiel für die Sequenzen `caabaacc` und `babcaaa` wäre ein mögliches lokales Alignment: $\begin{array}{r|c|l} \texttt{ca} & \texttt{ab-aa} & \texttt{cc} \\ \hline \texttt{b} & \texttt{abcaa} & \texttt{a} \end{array}$.

Karopapier …

2. Versuch …

Aufgabe 3. 3+2+1=6 Punkte

Der Algorithmus von Smith-Waterman (local alignment) hat eine Variante für beliebige Gapkosten. Die Eingabesequenzen seien mit x, y bezeichnet. Die Kosten für einen Gap der Länge g seien $\gamma(g) \in \mathbb{Z}$, und die Kosten für Matches und Mismatches seien gegeben durch $s : \Sigma^2 \to \mathbb{Z}$.

(a) Geben Sie die Rekursionsformel an. (Nichts weiter.)

(b) Geben Sie eine Abschätzung für die Laufzeit an, und begründen Sie diese.

(c) Geben Sie eine Abschätzung für den Speicherplatzbedarf an, und begründen Sie diese.

Aufgabe 4. 10 Punkte

Berechnen Sie für die zwei Sequenzen $S = \texttt{ctag}$ (horizontal) und $T = \texttt{cgtgag}$ (vertikal) ein globales Alignment entsprechend dem A*-Algorithmus. Ein Match kostet 0, ein Mismatch 1 und ein Indel 3. Als untere Schranke für die Distanz zum Ziel soll (wie gehabt) die „Abweichung von der Zieldiagonalen“ verwendet werden, hier also

$$\ell\big((i,j),(m,n)\big) = \big|(i-j)-(m-n)\big| \cdot 3\,.$$

(Mit der „3“ für Indel.) Dabei ist (i,j) der aktuell berechnete Eintrag und (m,n) das Ziel.

1. - Berechnen Sie die DP *Matrix* für globales Alignment für S und T
   - einschließlich der *Traceback-Kanten*.
   - Lassen Sie Einträge frei, denen nie ein Wert zugewiesen wird. Wenn die Prioritätswarteschlange mehrere Einträge mit dem gleichen Wert enthält, können Sie selbst entscheiden, welcher als nächstes extrahiert werden soll.
   - Unterstreichen Sie alle Einträge, die aus der Prioritästwarteschlange entfernt werden, bevor die optimale Lösung gefunden wurde.
   - Lassen Sie so viel Platz um die Einträge, dass sie, wenn ein Eintrag (beim „Relaxieren“) aktualisiert wird, den **[**alten Eintrag**]** in Klammern setzen können – also nicht durchstreichen – und den neuen Eintrag daneben schreiben können. Markieren Sie alte Traceback-Kanten irgendwie, wenn Sie ihre Gütigkeit verlieren.
2. Kennzeichnen Sie die Kanten, die beim Traceback *abgelaufen* werden. (Fett oder bunt zeichnen – oder geben Sie Koordinaten der durchlaufenen Knoten an.)
3. Schreiben Sie *das Alignment* auf.

Karopapier …

Aufgabe 5. 3+1+1=5 Punkte

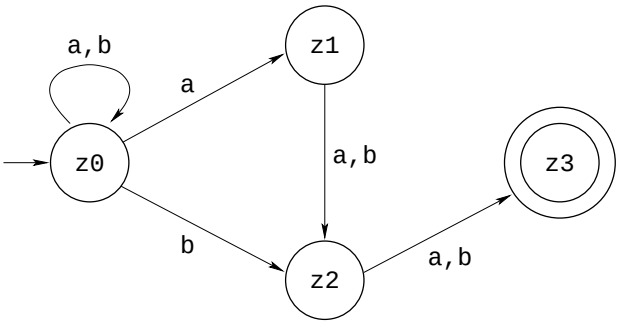
Welche Methoden muss die Priority Queue in Dijkstras Algorithmus (in der Anwendung für das paarweise Sequenzalignment) unterstützen? Welche Bedingungen muss das Scoring Schema unbedingt erfüllen? Was ist der Wertebereich der Prioritäten?

Aufgabe 6. 5 Punkte

Erklären Sie, wie der FastA-Algorithmus Hashing verwendet, um effizient so genannte *hot-spots* zu finden. (Weiter nichts!)

Aufgabe 7. 8+2=10 Punkte

Gegeben ist der folgende NFA M:



(a) Wandeln Sie den NFA in einen DFA M' um. Verwenden Sie dazu die „Potenzmengenkonstruktion“, erzeugen Sie dabei aber nur die Zustände, die tatsächlich erreicht werden können.

(b) Beschreiben Sie mit Worten die von M akzeptierte Sprache.

Aufgabe 8. 4+2+4=10 Punkte

(a) Beschreiben Sie allgemein den Algorithmus, um eine reguläre Grammatik $G = (V, \Sigma, P, S)$ in einen NFA $M = (Z, \Sigma, \delta, U_0, E)$ umzuwandeln. Sie können annehmen, dass $\varepsilon \notin L(G)$.

(b) Erklären Sie, was in (a) zu tun ist, falls $\varepsilon \in L(G)$.

(c) Wandeln Sie konkret die folgende reguläre Grammatik in einen NFA um: $G = (V, \Sigma, P, S)$, wobei $V = \{S, T, U\}$, $\Sigma = \{a, b\}$ und

$$
\begin{aligned}
P := \{ & \\
& S \to aS, \\
& S \to bS, \\
& S \to aT, \\
& S \to bU, \\
& U \to aT, \\
& U \to bT, \\
& T \to a, \\
& T \to b \\
\}\,. &
\end{aligned}
$$

Aufgabe 9. 2+2+2=6 Punkte

Seien α, β reguläre Ausdrücke und M_α, M_β NFAs mit $L(M_\alpha) = L(\alpha)$ und $L(M_\beta) = L(\beta)$. Beweisen Sie, dass dann für die folgenden regulären Ausdrücke ebenfalls NFAs existieren, indem Sie jeweils eine Konstruktionsvorschrift für ein M_γ mit $L(M_\gamma) = L(\gamma)$ angeben.

(a) $\gamma = (\alpha \mid \beta)$

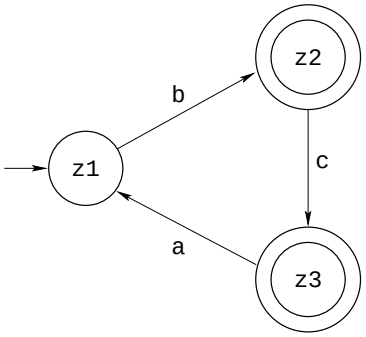
(b) $\gamma = \alpha\beta$

(c) $\gamma = (\alpha)^*$

Wichtig: Sie dürfen dabei ε-Übergänge verwenden, können aber gleichzeitig davon ausgehen, dass M_α, M_β keine ε-Übergänge enthalten. Die Elimination von ε-Übergängen muss hier also nicht erklärt werden.

Aufgabe 10. 10 Punkte

Konstruieren Sie mittels des Algorithmus aus der Vorlesung einen regulären Ausdruck für den folgenden DFA. (Alle nicht eingezeichneten Kanten führen in einen „Fehlerzustand“ z_4, der hier der Einfachheit halber weggelassen wurde.) Geben Sie jeweils $\gamma_{i,j}^k$ an, auch für die Zwischenergebnisse.



Aufgabe 11. 2+2+4=8 Punkte

Gegeben sei ein multiples Alignment, geschrieben in Form einer Matrix $A = (a_{i,j})$, wobei der Zeilenindex $i = 1, \ldots, n$ die Sequenz und der Spaltenindex $j = 1, \ldots, m$ die Position bezeichnet.

(a) Was versteht man unter der Projektion eines multiplen Alignments von n Sequenzen auf zwei Sequenzen i_1, i_2? (Gefragt ist die Definition.)

(b) Wir nehmen ferner an, dass wir bereits über eine Bewertungsfunktion für paarweise Alignments verfügen. Was versteht man in diesem Zusammenhang unter einem *WSOP-Score für* A, und welche weiteren Angaben benötigt man dazu noch? (Gefragt ist wiederum einfach die Definition.)

(c) Wenn wir in der Bewertungsfunktion für den paarweisen Sequenzvergleich lineare Gapkosten zugrundelegen, wie kann man dann mittels paarweisem Alignment eine untere Schranke für den WSOP Score von A berechnen? Begründen Sie ihre Antwort. (Gefragt ist ein Beweis.)

Aufgabe 12. 3+3+2=8 Punkte

(a) Erklären Sie (kurz!) den Ablauf des K-means Algorithmus.

Welches Distanzmaß wird beim K-means Algorithums zugrundegelegt?

(b) Was ist beim K-medoid Algorithmus anders?

Welche Eigenschaften muss das Distanzmaß haben, damit man den K-medoid Algorithmus durchführen kann?

(c) Vergleichen Sie die Laufzeiten beider Algorithmen.

Aufgabe 13. 2+(2+2+3)=9 Punkte

(a) Gegeben ist eine Index-Menge I und eine Dissimilarity Matrix $D = (d_i, j)$, wobei $i, j \in I$. Geben Sie die Formeln für den Cluster-Abstand zwischen $A, B \subseteq I$ an für:

(i) group average, $d_{\text{GA}}(A, B) =$

(ii) single linkage, $d_{\text{SL}}(A, B) =$

(iii) complete linkage, $d_{\text{CL}}(A, B) =$

(b) Gegeben sind Datenpunkte a, b, c, d, e, f, g mit den folgenden Abständen:

$$a \overset{16}{\text{———}} b \overset{2}{\text{———}} c \overset{12}{\text{———}} d \overset{4}{\text{———}} e \overset{13}{\text{———}} f \overset{8}{\text{———}} g$$

wobei sich die restlichen Abstände durch kürzeste Pfade ergeben. Berechnen Sie mit den in der Vorlesung behandelten agglomerativen Clustering-Verfahren die einzelnen Cluster, wie sie nach und nach zusammengefasst werden, bis nur noch drei Cluster vorhanden sind. Geben Sie für jeden Schritt die Cluster an und für die jeweils zusammengefassten Cluster deren Abstand. Es bietet sich an, ein Dendrogramm zu zeichnen. Beim letzten Schritt (von 4 auf 3 Cluster) geben Sie bitte alle 6 paarweisen Abstände an. (Der Rechenweg muss ersichtlich sein.)

(i) single linkage

(ii) complete linkage

(iii) group average

Aufgabe 14. 1+1+1=3 Punkte

Geben Sie die Definitionen für die folgenden Begriffe:

(a) Ultrametrische Distanz

(b) Additive Distanz

(c) (Allgemeine) Distanz

Aufgabe 15. 2+4=6 Punkte

Gegeben sei eine metrische Distanz D. Dann sind die folgenden Aussagen äquivalent.

(A): D ist eine Ultrametrik.

(B): Es gibt einen additiven Baum für die Matrix $2D$, der einen Knoten r enthält, der von allen anderen Taxa denselben Abstand hat.

Beweisen Sie beide Richtungen separat, indem Sie jeweils konstruktiv vorgehen:

(a) (A) $\Rightarrow$ (B)

(b) (B) $\Rightarrow$ (A)

Aufgabe 16. 8 Punkte

Berechnen Sie mit dem Algorithmus aus der Vorlesung den ultrametrischen Baum für die folgende Distanzmatrix D. Geben Sie bei den rekursiven Aufrufen jeweils an, für welche Mengen von Taxa sie erfolgen, und welchen Sub-Baum sie ergeben haben. Verwenden Sie bei der rekursiven Aufteilung jeweils das Taxon mit dem kleinsten Index als Pivot.

D	l:0	l:1	l:2	l:3	l:4	l:5	l:6
k:0	0	3	1	2	3	3	1
k:1	3	0	3	3	1	2	3
k:2	1	3	0	2	3	3	1
k:3	2	3	2	0	3	3	2
k:4	3	1	3	3	0	2	3
k:5	3	2	3	3	2	0	3
k:6	1	3	1	2	3	3	0

Aufgabe 17. 3+3+6=12 Punkte

(a) Wir betrachten eine Markoffkette mit den Zuständen $0, p, q$ (0 ist der Start- und Endzustand) und den folgenden Übergangswahrscheinlichkeiten:

$$A := \begin{array}{|c|ccc|} \hline & 0 & p & q \\ \hline 0 & 0 & 0.7 & 0.3 \\ p & 0.5 & 0.1 & 0.4 \\ q & 0.3 & 0.7 & 0 \\ \hline \end{array}$$

Bestimmen Sie die Wahrscheinlichkeit, dass die Markoffkette genau die Zustände pq durchläuft (und danach stoppt).

(b) Wir erweitern nun die Markoffkette zu einem Hidden Markov Model mit den Ausgabesymbolen X, Y und den folgenden Emissionswahrscheinlichkeiten:

$$e := \begin{array}{|c|cc|} \hline & X & Y \\ \hline p & 0.7 & 0.3 \\ q & 0.2 & 0.8 \\ \hline \end{array}.$$

Bestimmen Sie die Wahrscheinlichkeit, dass genau das Zeichen Y ausgegeben wird, und das HMM danach stoppt.

(c) *(Extra-Aufgabe für alle, die am Schluss noch Zeit haben :-))*
Bestimmen Sie mit dem Viterbi-Algorithmus den wahrscheinlichsten Zustandspfad für die Ausgabesequenz YX (genau diese 2 Zeichen).

Algorithmen und Datenstrukturen (für Bioinformatik)

Freie Universität Berlin, Institut für Informatik
Dr. Clemens Gröpl
Wintersemester 2004/2005

Klausur

21. Februar 2005

Name, Vorname: (B)

Matrikelnummer:

1		4
2		10
3		6
4		10
5		5
6		5
7		10
8		10
9		6
10		10

11		8
12		8
13		9
14		3
15		6
16		8
17		12
Σ		130

Note:

Aufgabe 1. 2+(1+1)=4 Punkte

(a) Erklären Sie kurz den Unterschied zwischen Hamming- und Editdistanz.

(b) In der Vorlesung wurde bei affinen Gapkosten die *gap opening penalty* mit $d \geq 0$ und die *gap extension penalty* mit $e \geq 0$ bezeichnet. Nennen Sie jeweils ein biologisches Anwendungsbeispiel, wo man d und e zweckmäßigerweise so wählen wird, dass gilt:

(i) $e > d$:

(ii) $d > e$:

Aufgabe 2. 10 Punkte

Berechnen Sie für die Sequenzen $S =$ `actg` (horizontal) und $T =$ `ctagc` (vertikal) ein lokales Alignment nach dem Algorithmus von Smith-Waterman. Ein Match trägt 3 zum Score des Alignments bei, ein Mismatch -5 und ein Space -2.

1. Berechnen Sie die DP-*Matrix*, und
2. tragen Sie dabei jeweils *alle* Traceback-Kanten ein (also eventuell mehrere pro Zelle).
3. Heben Sie *die beim Traceback benutzten* Kanten besonders hervor, und
4. notieren Sie das resultierende *lokale Alignment*. Geben Sie auch die jeweiligen *Start- und End-Positionen* in den beiden Sequenzen an. Zum Beispiel für die Sequenzen `caabaacc` und `babcaaa` wäre ein mögliches lokales Alignment: $\begin{array}{r|c|l} \texttt{ca} & \texttt{ab-aa} & \texttt{cc} \\ \hline \texttt{b} & \texttt{abcaa} & \texttt{a} \end{array}$.

Karopapier … 2. Versuch …

Aufgabe 3. 3+2+1=6 Punkte

Der Algorithmus von Smith-Waterman (local alignment) hat eine Variante für beliebige Gapkosten. Die Eingabesequenzen seien mit x, y bezeichnet. Die Kosten für einen Gap der Länge g seien $\gamma(g) \in \mathbb{Z}$, und die Kosten für Matches und Mismatches seien gegeben durch $s : \Sigma^2 \to \mathbb{Z}$.

(a) Geben Sie die Rekursionsformel an. (Nichts weiter.)

(b) Geben Sie eine Abschätzung für die Laufzeit an, und begründen Sie diese.

(c) Geben Sie eine Abschätzung für den Speicherplatzbedarf an, und begründen Sie diese.

Aufgabe 4. 10 Punkte

Berechnen Sie für die zwei Sequenzen $S = \texttt{gtac}$ (horizontal) und $T = \texttt{gtgagc}$ (vertikal) ein globales Alignment entsprechend dem A*-Algorithmus. Ein Match kostet 0, ein Mismatch 1 und ein Indel 3. Als untere Schranke für die Distanz zum Ziel soll (wie gehabt) die „Abweichung von der Zieldiagonalen“ verwendet werden, hier also

$$\ell\big((i,j),(m,n)\big) = \big|(i-j)-(m-n)\big| \cdot 3\,.$$

(Mit der „3“ für Indel.) Dabei ist (i,j) der aktuell berechnete Eintrag und (m,n) das Ziel.

1. - Berechnen Sie die DP *Matrix* für globales Alignment für S und T
   - einschließlich der *Traceback-Kanten*.
   - Lassen Sie Einträge frei, denen nie ein Wert zugewiesen wird. Wenn die Prioritätswarteschlange mehrere Einträge mit dem gleichen Wert enthält, können Sie selbst entscheiden, welcher als nächstes extrahiert werden soll.
   - Unterstreichen Sie alle Einträge, die aus der Prioritästwarteschlange entfernt werden, bevor die optimale Lösung gefunden wurde.
   - Lassen Sie so viel Platz um die Einträge, dass sie, wenn ein Eintrag (beim „Relaxieren“) aktualisiert wird, den **[**alten Eintrag**]** in Klammern setzen können – also nicht durchstreichen – und den neuen Eintrag daneben schreiben können. Markieren Sie alte Traceback-Kanten irgendwie, wenn Sie ihre Gütigkeit verlieren.
2. Kennzeichnen Sie die Kanten, die beim Traceback *abgelaufen* werden. (Fett oder bunt zeichnen – oder geben Sie Koordinaten der durchlaufenen Knoten an.)
3. Schreiben Sie *das Alignment* auf.

Karopapier …

Aufgabe 5. 5 Punkte

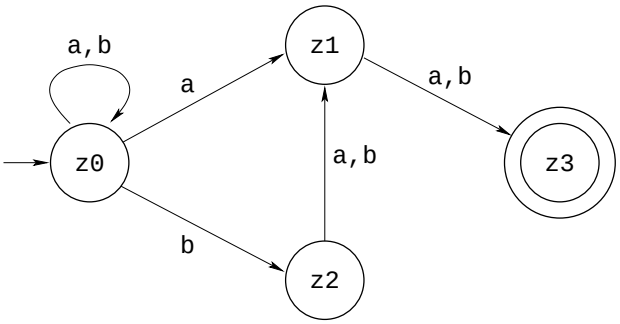
Erklären Sie, wie der FastA-Algorithmus Hashing verwendet, um effizient so genannte *hot-spots* zu finden. (Weiter nichts!)

Aufgabe 6. 3+1+1=5 Punkte

Welche Methoden muss die Priority Queue in Dijkstras Algorithmus (in der Anwendung für das paarweise Sequenzalignment) unterstützen? Welche Bedingungen muss das Scoring Schema unbedingt erfüllen? Was ist der Wertebereich der Prioritäten?

Aufgabe 7. 8+2=10 Punkte

Gegeben ist der folgende NFA M:



(a) Wandeln Sie den NFA in einen DFA M' um. Verwenden Sie dazu die „Potenzmengenkonstruktion“, erzeugen Sie dabei aber nur die Zustände, die tatsächlich erreicht werden können.

(b) Beschreiben Sie mit Worten die von M akzeptierte Sprache.

Aufgabe 8. 4+2+4=10 Punkte

(a) Beschreiben Sie allgemein den Algorithmus, um eine reguläre Grammatik $G = (V, \Sigma, P, S)$ in einen NFA $M = (Z, \Sigma, \delta, U_0, E)$ umzuwandeln. Sie können annehmen, dass $\varepsilon \notin L(G)$.

(b) Erklären Sie, was in (a) zu tun ist, falls $\varepsilon \in L(G)$.

(c) Wandeln Sie konkret die folgende reguläre Grammatik in einen NFA um: $G = (V, \Sigma, P, S)$, wobei $V = \{S, T, U\}$, $\Sigma = \{a, b\}$ und

$$
\begin{aligned}
P := \{ & \\
& S \to aS, \\
& S \to bS, \\
& S \to aT, \\
& S \to bU, \\
& T \to aU, \\
& T \to bU, \\
& U \to a, \\
& U \to b \\
\}\,. &
\end{aligned}
$$

Aufgabe 9. 2+2+2=6 Punkte

Seien α, β reguläre Ausdrücke und M_α, M_β NFAs mit $L(M_\alpha) = L(\alpha)$ und $L(M_\beta) = L(\beta)$. Beweisen Sie, dass dann für die folgenden regulären Ausdrücke ebenfalls NFAs existieren, indem Sie jeweils eine Konstruktionsvorschrift für ein M_γ mit $L(M_\gamma) = L(\gamma)$ angeben.

(a) $\gamma = (\alpha \mid \beta)$

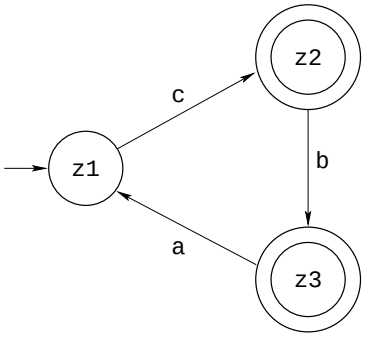
(b) $\gamma = \alpha\beta$

(c) $\gamma = (\alpha)^*$

Wichtig: Sie dürfen dabei ε-Übergänge verwenden, können aber gleichzeitig davon ausgehen, dass M_α, M_β keine ε-Übergänge enthalten. Die Elimination von ε-Übergängen muss hier also nicht erklärt werden.

Aufgabe 10. 10 Punkte

Konstruieren Sie mittels des Algorithmus aus der Vorlesung einen regulären Ausdruck für den folgenden DFA. (Alle nicht eingezeichneten Kanten führen in einen „Fehlerzustand“ z_4, der hier der Einfachheit halber weggelassen wurde.) Geben Sie jeweils $\gamma_{i,j}^k$ an, auch für die Zwischenergebnisse.



Aufgabe 11. 2+2+4=8 Punkte

Gegeben sei ein multiples Alignment, geschrieben in Form einer Matrix $A = (a_{i,j})$, wobei der Zeilenindex $i = 1, \ldots, n$ die Sequenz und der Spaltenindex $j = 1, \ldots, m$ die Position bezeichnet.

(a) Was versteht man unter der Projektion eines multiplen Alignments von n Sequenzen auf zwei Sequenzen i_1, i_2? (Gefragt ist die Definition.)

(b) Wir nehmen ferner an, dass wir bereits über eine Bewertungsfunktion für paarweise Alignments verfügen. Was versteht man in diesem Zusammenhang unter einem *WSOP-Score für* A, und welche weiteren Angaben benötigt man dazu noch? (Gefragt ist wiederum einfach die Definition.)

(c) Wenn wir in der Bewertungsfunktion für den paarweisen Sequenzvergleich lineare Gapkosten zugrundelegen, wie kann man dann mittels paarweisem Alignment eine untere Schranke für den WSOP Score von A berechnen? Begründen Sie ihre Antwort. (Gefragt ist ein Beweis.)

Aufgabe 12. 3+3+2=8 Punkte

(a) Erklären Sie (kurz!) den Ablauf des K-means Algorithmus.

Welches Distanzmaß wird beim K-means Algorithums zugrundegelegt?

(b) Was ist beim K-medoid Algorithmus anders?

Welche Eigenschaften muss das Distanzmaß haben, damit man den K-medoid Algorithmus durchführen kann?

(c) Vergleichen Sie die Laufzeiten beider Algorithmen.

Aufgabe 13. 2+(2+2+3)=9 Punkte

(a) Gegeben ist eine Index-Menge I und eine Dissimilarity Matrix $D = (d_i, j)$, wobei $i, j \in I$. Geben Sie die Formeln für den Cluster-Abstand zwischen $A, B \subseteq I$ an für:

(i) complete linkage, $d_{\text{CL}}(A, B) =$

(ii) single linkage, $d_{\text{SL}}(A, B) =$

(iii) group average, $d_{\text{GA}}(A, B) =$

(b) Gegeben sind Datenpunkte a, b, c, d, e, f, g mit den folgenden Abständen:

$$a \overset{8}{\text{———}} b \overset{13}{\text{———}} c \overset{4}{\text{———}} d \overset{12}{\text{———}} e \overset{2}{\text{———}} f \overset{16}{\text{———}} g$$

wobei sich die restlichen Abstände durch kürzeste Pfade ergeben. Berechnen Sie mit den in der Vorlesung behandelten agglomerativen Clustering-Verfahren die einzelnen Cluster, wie sie nach und nach zusammengefasst werden, bis nur noch drei Cluster vorhanden sind. Geben Sie für jeden Schritt die Cluster an und für die jeweils zusammengefassten Cluster deren Abstand. Es bietet sich an, ein Dendrogramm zu zeichnen. Beim letzten Schritt (von 4 auf 3 Cluster) geben Sie bitte alle 6 paarweisen Abstände an. (Der Rechenweg muss ersichtlich sein.)

(i) single linkage

(ii) complete linkage

(iii) group average

Aufgabe 14. 1+1+1=3 Punkte

Geben Sie die Definitionen für die folgenden Begriffe:

(a) Ultrametrische Distanz

(b) Additive Distanz

(c) (Allgemeine) Distanz

Aufgabe 15. 2+4=6 Punkte

Gegeben sei eine metrische Distanz D. Dann sind die folgenden Aussagen äquivalent.

(A): D ist eine Ultrametrik.

(B): Es gibt einen additiven Baum für die Matrix $2D$, der einen Knoten r enthält, der von allen anderen Taxa denselben Abstand hat.

Beweisen Sie beide Richtungen separat, indem Sie jeweils konstruktiv vorgehen:

(a) (A) $\Rightarrow$ (B)

(b) (B) $\Rightarrow$ (A)

Aufgabe 16. 8 Punkte

Berechnen Sie mit dem Algorithmus aus der Vorlesung den ultrametrischen Baum für die folgende Distanzmatrix D. Geben Sie bei den rekursiven Aufrufen jeweils an, für welche Mengen von Taxa sie erfolgen, und welchen Sub-Baum sie ergeben haben. Verwenden Sie bei der rekursiven Aufteilung jeweils das Taxon mit dem kleinsten Index als Pivot.

D	l:0	l:1	l:2	l:3	l:4	l:5	l:6
k:0	0	3	1	1	3	3	2
k:1	3	0	3	3	2	2	3
k:2	1	3	0	1	3	3	2
k:3	1	3	1	0	3	3	2
k:4	3	2	3	3	0	1	3
k:5	3	2	3	3	1	0	3
k:6	2	3	2	2	3	3	0

Aufgabe 17. 3+3+6=12 Punkte

(a) Wir betrachten eine Markoffkette mit den Zuständen $0, p, q$ (0 ist der Start- und Endzustand) und den folgenden Übergangswahrscheinlichkeiten:

$$A := \begin{array}{|c|ccc|} \hline & 0 & p & q \\ \hline 0 & 0 & 0.7 & 0.3 \\ p & 0.5 & 0.1 & 0.4 \\ q & 0.3 & 0.7 & 0 \\ \hline \end{array}$$

Bestimmen Sie die Wahrscheinlichkeit, dass die Markoffkette genau die Zustände qp durchläuft (und danach stoppt).

(b) Wir erweitern nun die Markoffkette zu einem Hidden Markov Model mit den Ausgabesymbolen X, Y und den folgenden Emissionswahrscheinlichkeiten:

$$e := \begin{array}{|c|cc|} \hline & X & Y \\ \hline p & 0.7 & 0.3 \\ q & 0.2 & 0.8 \\ \hline \end{array}.$$

Bestimmen Sie die Wahrscheinlichkeit, dass genau das Zeichen X ausgegeben wird, und das HMM danach stoppt.

(c) *(Extra-Aufgabe für alle, die am Schluss noch Zeit haben :-))*
Bestimmen Sie mit dem Viterbi-Algorithmus den wahrscheinlichsten Zustandspfad für die Ausgabesequenz XY (genau diese 2 Zeichen).