INSTITUT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR EFFIZIENTE ALGORITHMEN

Skriptum

zur Vorlesung

Algorithmische Bioinformatik I/II

gehalten im Wintersemester 2001/2002

und im Sommersemester 2002 von

Volker Heun

Erstellt unter Mithilfe von:

Peter Lücke – Hamed Behrouzi – Michael Engelhardt

Sabine Spreer – Hanjo Täubig

Jens Ernst – Moritz Maaß

14. Mai 2003

Version 0.96

Vorwort

Dieses Skript entstand parallel zu den Vorlesungen *Algorithmische Bioinformatik I* und *Algorithmische Bioinformatik II*, die im Wintersemester 2001/2002 sowie im Sommersemester 2002 für Studenten der Bioinformatik und Informatik sowie anderer Fachrichtungen an der Technischen Universität München im Rahmen des von der Ludwig-Maximilians-Universität und der Technischen Universität gemeinsam veranstalteten Studiengangs Bioinformatik gehalten wurde. Einige Teile des Skripts basieren auf der bereits im Sommersemester 2000 an der Technischen Universität München gehaltenen Vorlesung *Algorithmen der Bioinformatik* für Studierende der Informatik.

Das Skript selbst umfasst im Wesentlichen die grundlegenden Themen, die man im Bereich Algorithmische Bioinformatik einmal gehört haben sollte. Die vorliegende Version bedarf allerdings noch einer Ergänzung weiterer wichtiger Themen, die leider nicht in den Vorlesungen behandelt werden konnten.

An dieser Stelle möchte ich insbesondere Hamed Behrouzi, Michael Engelhardt und Peter Lücke danken, die an der Erstellung des ersten Teils dieses Skriptes (Kapitel 2 mit 5) maßgeblich beteiligt waren. Bei Sabine Spreer möchte ich mich für die Unterstützung bei Teilen des siebten Kapitels bedanken. Bei meinen Übungsleitern Jens Ernst und Moritz Maaß für deren Unterstützung der Durchführung des Übungsbetriebs, aus der einige Lösungen von Übungsaufgaben in dieses Text eingeflossen sind. Bei Hanjo Täubig möchte ich mich für die Mithilfe zur Fehlerfindung bedanken, insbesondere bei den biologischen Grundlagen.

Falls sich dennoch weitere (Tipp)Fehler unserer Aufmerksamkeit entzogen haben sollten, so bin ich für jeden Hinweis darauf (an heun@in.tum.de) dankbar.

München, im September 2002 Volker Heun

Inhaltsverzeichnis

Mehrfaches Sequenzen Alignment

4.1 Distanz- und Ähnlichkeitsmaße

In diesem Kapitel wollen wir uns mit der gleichzeitigen Ausrichtungen mehrerer (mehr als zwei) Sequenzen beschäftigen.

4.1.1 Mehrfache Alignments

Zuerst müssen wir mehrfache Alignments sowie deren Distanz bzw. Ähnlichkeit analog wie im Falle paarweiser Sequenzen Alignments definieren.

Definition 4.1 *Seien* $s_1, \ldots, s_k \in \Sigma^*$. *Eine Folge* $\overline{s}_1, \ldots, \overline{s}_k$ *heißt* mehrfaches Sequenzen Alignment (MSA) *für die Sequenz* $s_1, \ldots, s_k$, *wenn gilt:*

- $|\overline{s}_1| = \cdots = |\overline{s}_k| = n$,
- $\overline{s}_{1,i} = \overline{s}_{2,i} = \cdots = \overline{s}_{k,i} \quad \Rightarrow \quad \overline{s}_{1,i} \neq -$,
- $\overline{s}_j|_\Sigma = s_j$ *für alle* $j \in [1:k]$.

4.1.2 Alignment-Distanz und -Ähnlichkeit

Definition 4.2 *Sei* $w : \overline{\Sigma}^k \to \mathbb{R}_+$ *eine Kostenfunktion für ein Distanzmaß bzw. Ähnlichkeitsmaß eines* k*-fachen Sequenzen Alignments* $(\overline{s}_1, \ldots, \overline{s}_k)$ *für* $s_1, \ldots, s_k$, *dann ist*

$$w(\overline{s}_1, \ldots, \overline{s}_k) := \sum_{i=1}^{n} w(\overline{s}_{1,i}, \ldots, \overline{s}_{k,i})$$

mit $n = |\overline{s}_1|$ *die Distanz bzw. Ähnlichkeit des Alignments* $(\overline{s}_1, \ldots, \overline{s}_k)$ *für* $s_1, \ldots, s_k$.

Wie im Falle paarweiser Sequenzen Alignments sollte die Kostenfunktion wieder den wesentlichen Bedingungen einer Metrik entsprechen. Die Kostenfunktion w sollte wiederum symmetrisch sein:

$$w(a_1, \ldots, a_k) = w(a_{\pi_1}, \ldots, a_{\pi_k})$$

für eine beliebige Permutation $\pi = (\pi_1, \ldots, \pi_k) \in S([1 : k])$. Weiter sollte die Dreiecks-Ungleichung gelten:

$$\begin{aligned} & w(a_1, \ldots, a_i, \ldots, a_j, \ldots, a_k) \\ & \qquad \leq \; w(a_1, \ldots, a_i, \ldots, x, \ldots, a_k) + w(a_1, \ldots, x, \ldots, a_j, \ldots a_k). \end{aligned}$$

Weiterhin sollte auch wieder die Definitheit gelten:

$$w(a_1, \ldots, a_k) = 0 \quad \Leftrightarrow \quad a_1 = \ldots = a_k.$$

Eine Standardkostenfunktion ist die so genannte *Sum-of-Pairs-Funktion*:

$$w(a_1, \ldots, a_k) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} \tilde{w}(a_i, a_j),$$

wobei $\tilde{w} : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$ eine gewöhnliche Kostenfunktion für Alignment- oder Ähnlichkeitsmaße eines paarweisen Alignments ist. Hierbei nehmen wir jedoch an, dass $\tilde{w}(-, -) = 0$.

Dies impliziert den Abstand oder Ähnlichkeit für mehrfache Alignments:

$$w(\overline{s}_1, \ldots, \overline{s}_k) := \sum_{i=1}^{k} \sum_{j=i+1}^{k} \tilde{w}(\overline{s}_i, \overline{s}_j).$$

Definition 4.3 *Ein mehrfaches Sequenzen Alignment* $(\overline{s}_1, \ldots, \overline{s}_k) \in \overline{\Sigma}^n$ *für* $s_1, \ldots, s_k \in \Sigma^*$ *heißt optimal, wenn*

$$w(\overline{s}_1, \ldots, \overline{s}_k) = \min\{w(\overline{t}_1, \ldots, \overline{t}_k) \mid (\overline{t}_1, \ldots, \overline{t}_k) \text{ ist ein MSA für } s_1, \ldots, s_k\}.$$

Dann ist $d_w(s_1, \ldots, s_k) := w(\overline{s}_1, \ldots, \overline{s}_k)$ *die mehrfache Alignment-Distanz bzw. -Ähnlichkeit von* $s_1, \ldots, s_k$.

Wir merken hier noch an, dass wir im Folgenden meist als Kostenfunktion die oben erwähnte Sum-of-Pairs-Kostenfunktion verwenden werden. Das zugehörige Distanz bzw. Ähnlichkeitsmaß wird dann oft auch als Sum-of-Pairs-Maß oder kurz SP-Maß bezeichnet.

4.2 Dynamische Programmierung

In diesem Abschnitt verallgemeinern wir die Methode der Dynamischen Programmierung von paarweisen auf mehrfache Sequenzen Alignments. Aufgrund der großen Laufzeit ist dieses Verfahren aber eher von theoretischem Interesse.

4.2.1 Rekursionsgleichungen

Im Folgenden sei $D[\vec{x}]$ für $\vec{x} = (x_1, \ldots, x_k) \in \mathbb{N}_0^k$ der Wert eines optimalen mehrfachen Sequenzen Alignments für $s_{1,1} \cdots s_{1,x_1}$, $s_{2,1} \cdots s_{2,x_2}$, $\ldots$ $s_{k-1,1} \cdots s_{k-1,x_{k-1}}$ und $s_{k,1} \cdots s_{k,x_k}$. Der folgende Satz lässt sich analog wie für das paarweise Sequenzen Alignment beweisen.

Theorem 4.4 *Seien $s_1, \ldots, s_k \in \Sigma^*$. Es gilt für $\vec{x} \in [1 : |s_1|] \times \cdots \times [1 : |s_k|]$:*

$$D[\vec{x}] := \min\{D[\vec{x} - \vec{\eta}] + w(\vec{x} \bullet \vec{\eta}) \mid \vec{\eta} \in [0 : 1]^k \setminus \vec{0}\}.$$

Hierbei ist

$$(x_1, \ldots, x_k) \bullet (\eta_1, \ldots, \eta_n) = (s_{1,x_1} \otimes \eta_1, \ldots, s_{k,x_k} \otimes \eta_k) \text{ mit } \begin{array}{l} a \otimes 0 = - \\ a \otimes 1 = a \end{array} \text{ für } a \in \Sigma.$$

Nun stellt sich noch die Frage, wie die Anfangswerte für $\vec{x} \in [0 : n]^k \setminus [1 : n]^k$ eines solchen mehrfaches Sequenzen Alignments aussehen. Dies wird am Beispiel von drei Sequenzen im folgenden Bild erklärt.

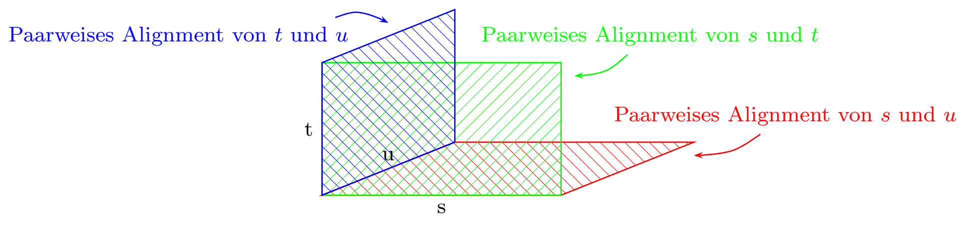

Abbildung 4.1: Skizze: Anfangswerte für ein 3-faches Sequenzen-Alignment

Für ein 3-faches Sequenzen Alignment von s_1, s_2 und s_3 mit $n_i = |s_i|$ für alle $i \in [1 : 3]$ wollen wir noch explizit die Rekursionsformeln und Anfangsbedingungen angeben. Es gilt dann für eine globales mehrfaches Sequenzen Alignment mit $(i, j, k) \in [1 : n_1] \times [1 : n_2] \times [1 : n_3]$:

$$D[0, 0, 0] = 0,$$

$$
\begin{aligned}
D[i,0,0] &= D[i-1,0,0] + w(s_i^1,-,-),\\
D[0,j,0] &= D[0,j-1,0] + w(-,s_j^2,-),\\
D[0,0,k] &= D[0,0,k-1] + w(-,-,s_k^3),\\
D[i,j,0] &= \min\left\{\begin{array}{c} D[i-1,j,0] + w(s_i^1,-,-),\\ D[i,j-1,0] + w(-,s_j^2,-),\\ D[i-1,j-1,0] + w(s_i^1,s_j^2,-) \end{array}\right\},\\
D[i,0,k] &= \min\left\{\begin{array}{c} D[i-1,0,k] + w(s_i^1,-,-),\\ D[i,0,k-1] + w(-,-,s_k^3),\\ D[i-1,0,k-1] + w(s_i^1,-,s_k^3) \end{array}\right\},\\
D[0,j,k] &= \min\left\{\begin{array}{c} D[0,j-1,k] + w(-,s_j^2,-),\\ D[0,j,k-1] + w(-,-,s_k^3),\\ D[0,j-1,k-1] + w(-,s_j^2,s_k^3) \end{array}\right\},\\
D[i,j,k] &= \min\left\{\begin{array}{c} D[i-1,j,k] + w(s_i^1,-,-),\\ D[i,j-1,k] + w(-,s_j^2,-),\\ D[i,j,k-1] + w(-,-,s_k^3),\\ D[i-1,j-1,k] + w(s_i^1,s_j^2,-),\\ D[i-1,j,k-1] + w(s_i^1,-,s_k^3),\\ D[i,j-1,k-1] + w(-,s_j^2,s_k^3),\\ D[i-1,j-1,k-1] + w(s_i^1,s_j^2,s_k^3) \end{array}\right\}.
\end{aligned}
$$

Hierbei ist $w : \overline{\Sigma}^3 \to \mathbb{R}_+$ die zugrunde gelegte Kostenfunktion. Für das Sum-Of-Pairs-Maß gilt dann: $w(x,y,z) = w'(x,y) + w'(x,z) + w'(y,z)$, wobei $w' : \overline{\Sigma}^2 \to \mathbb{R}_+$ die Standard-Kostenfunktion für Paare ist.

Die Übertragung auf z.B. semi-globale oder lokale mehrfache Alignments sei dem Leser zur Übung überlassen.

4.2.2 Zeitanalyse

Für die Zeitanalyse nehmen wir an, dass $|s_i| = \Theta(n)$ für alle $i \in [1 : k]$ gilt. Wir überlegen uns zuerst , dass die gesamte Tabelle $\Theta(n^k)$ viele Einträge besitzt. Für jeden Eintrag ist eine Minimumsbildung von $2^k - 1$ Elemente durchzuführen, wobei sich jeder Wert in Zeit $\Theta(k^2)$ berechnen lässt (wenn wir dass SP-Maß zugrunde legen). Insgesamt ist der Zeitbedarf also $O(k^2 * 2^k * n^k)$.

Dies ist leider exponentiell und selbst für moderat große k inakzeptabel. Für $k = 3$ ist dies gerade noch verwendbar, für größere k in der Regel unpraktikabel (außer die Sequenzen sind sehr kurz).

Leider gibt es für die Berechnung eines mehrfachen Sequenzen Alignment kein effizientes Verfahren. Man kann nämlich nachweisen, das die Entscheidung, ob eine gegebene Menge von Sequenzen, ein mehrfaches Alignment besitzt, das eine vorgegebene Distanz unterschreitet (oder Ähnlichkeit überschreitet), $\mathcal{NP}$-hart ist. Nach gängiger Lehrmeinung lassen sich $\mathcal{NP}$-harte Probleme nicht in polynomieller Zeit lösen, so dass eine Berechnung optimaler mehrfacher Sequenzen Alignments praktisch nicht effizient lösbar ist.

4.3 Alignment mit Hilfe eines Baumes

Da die exakte Lösung eines mehrfachen Alignments, wie eben angedeutet, in aller Regel sehr schwer lösbar ist, wollen wir uns mit so genannten *Approximationen* beschäftigen. Hierbei konstruieren wir Lösungen, die nur um einen bestimmten Faktor von einer optimalen Lösung entfernt ist.

4.3.1 Mit Bäumen konsistente Alignments

Dazu definieren wir zuerst mit Bäumen konsistente Alignments.

Definition 4.5 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ und sei $M = (\overline{s}_1, \ldots, \overline{s}_k)$ ein mehrfaches Sequenzen Alignment für S. Das Paar $(\overline{s}_i, \overline{s}_j)$ heißt* von M induziertes paarweises Alignment.

In der Regel werden wir bei induzierten Alignments annehmen, dass Spalten, die nur aus Leerzeichen $-$ bestehen, gestrichen werden, da wir bei paarweisen Sequenzen Alignments solche Spalten verboten haben. Wir bemerken hier noch einmal, dass die Distanz bzw. Ähnlichkeit dadurch nicht verändert wird, da wir hier für die Kostenfunktion annehmen, dass $w(-, -) = 0$ gilt.

Definition 4.6 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ und sei $T = (S, E)$ ein Baum. Ein mehrfaches Sequenzen Alignment $(\overline{s}_1, \ldots, \overline{s}_k)$ für S ist* konsistent mit T*, wenn jedes induzierte paarweise Sequenzen Alignment $(\overline{s}_i, \overline{s}_j)$ für $(s_i, s_j) \in E$ optimal ist.*

A	G	G	T	-
A	G	C	T	-
A	G	-	T	A
A	G	A	T	A
A	G	A	T	-
-	G	A	T	A

Abbildung 4.2: Skizze: mehrfaches Alignment, das mit einem Baum konsistent ist

4.3.2 Effiziente Konstruktion

Lemma 4.7 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ und sei $T = (S, E)$ ein Baum. Ein mehrfaches Sequenzen Alignment für S, das konsistent zu T ist, lässt sich in Zeit $O(kn^2)$ konstruieren, wobei $|s_i| = \Theta(n)$ für $i \in [1 : k]$.*

Beweis: Wir führen den Beweis mittels Induktion über k.

Induktionsanfang ($\boldsymbol{k = 0, 1, 2}$): Hierfür ist die Aussage trivial.

Induktionsschritt ($\boldsymbol{k \to k+1}$): Ohne Beschränkung der Allgemeinheit sei s_{k+1} ein Blatt von T und s_k adjazent zu s_{k+1} in T.

Nach Induktionvoraussetzung existiert ein mehrfaches Alignment $(\overline{s}_1, \ldots, \overline{s}_k)$ für $s_1, \ldots, s_k$, das konsistent zu T ist. Dieses wurde in Zeit $O(kn^2)$ konstruiert.

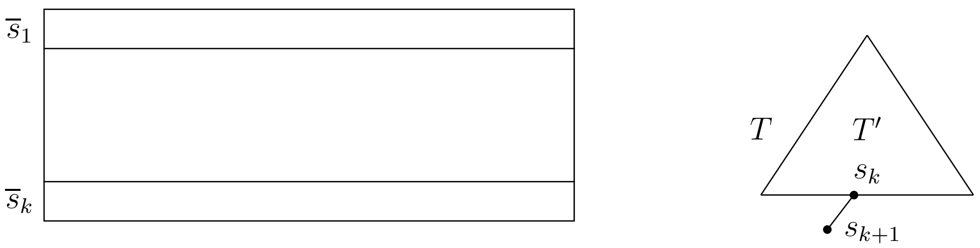

Abbildung 4.3: Skizze: Induktionsvoraussetzung

Wir berechnen ein optimales paarweises Alignment $(\tilde{s}_k, \tilde{s}_{k+1})$ von s_k mit s_{k+1} in Zeit $O(n^2)$. Dann erweitern wir das mehrfaches Sequenzen Alignment um das Alignment $(\tilde{s}_k, \tilde{s}_{k+1})$ wie in der Abbildung angegeben. Dazu müssen wir im Wesentlichen nur

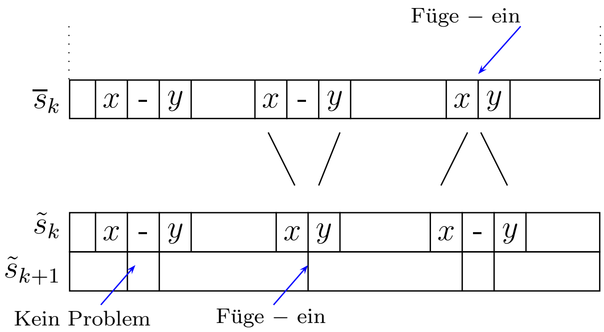

Abbildung 4.4: Skizze: Erweiterung des mehrfaches Sequenzen Alignments

die Zeile $\tilde{s}_{k+1}$ hinzufügen, wobei wir in der Regel sowohl in $\tilde{s}_{k+1}$ als auch im bereits konstruierten mehrfachen Sequenzen Alignment Leerzeichen einfügen müssen. Diese bestimmen sich im Wesentlichen aus dem Paar $(\overline{s}_k, \tilde{s}_k)$. Wir fügen im Prinzip so wenig wie möglich Leerzeichen hinzu, so dass $w(\overline{s}_k, \tilde{s}_k) = 0$ wird. ■

Somit können wir mehrfache Alignments, die zu Bäumen konsistent sind sehr effizient konstruieren. Im Weiteren wollen wir uns damit beschäftigen, wie gut solche mehrfachen Alignments sind.

4.4 Center-Star-Approximation

In diesem Abschnitt wollen wir ausgehend von dem im letzten Abschnitt vorgestellten Verfahren zur Konstruktion von mehrfachen Sequenzen Alignments mit Hilfe von Bäumen einen Algorithmus vorstellen, der ein mehrfaches Sequenzen Alignment bestimmter Güte konstruiert.

4.4.1 Die Wahl des Baumes

Bei der Center-Star-Methode besteht die Idee darin, den Baum T so zu wählen, dass er einen Stern darstellt. Also $T \cong$ ☼. Das Problem besteht nun darin, welche Sequenz als Zentrum des Sterns gewählt werden soll.

Definition 4.8 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ. Die Sequenz s_c mit $c \in [1 : k]$ heißt* Center-String*, wenn $\sum_{j=1}^{k} d(s_c, s_j)$ minimal ist.*

4.4.2 Approximationsgüte

Sei M_c das mehrfache Sequenzen Alignment, das zu T (dem Stern mit Zentrum s_c) konstruiert ist. Dann bezeichne $D(s_i, s_j)$ den Wert des durch M_c induzierten Alignments für s_i und s_j. Es gilt

$$\begin{aligned} D(s_i, s_j) &\geq d(s_i, s_j), \\ D(s_c, s_j) &= d(s_c, s_j), \\ D(M_c) &= \sum_{i=1}^{k} \sum_{j=i+1}^{k} D(s_i, s_j). \end{aligned}$$

Lemma 4.9 *Es gilt:*

$$D(s_i, s_j) \leq D(s_i, s_c) + D(s_c, s_j) = d(s_i, s_c) + d(s_c, s_j)$$

Beweis: Der Beweis folgt unmittelbar aus der folgenden Abbildung unter Berücksichtigung, dass für w die Dreiecksungleichung gilt. ∎

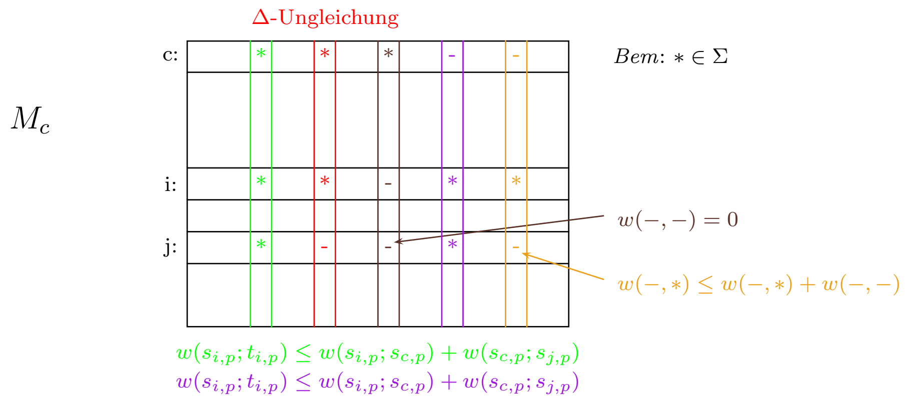

Abbildung 4.5: Skizze: Beweis

Erinnerung: $D(s,t) = w(\overline{s}, \overline{t}) = \sum_{i=1}^{|\overline{s}|} w(\overline{s_i}, \overline{t_i})$.

Sei M^* ein optimales mehrfaches Sequenzen Alignment für S und sei $D^*(s_i, s_j)$ der Wert des durch M^* induzierten paarweisen Alignments für s_i und s_j. Dann gilt:

$$d(s_1, \ldots, s_k) = D(M^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} D^*(s_i, s_j).$$

Theorem 4.10 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen und $T = (S, E)$ ein Stern, dessen Zentrum der Center-String von S ist. Sei M_c ein mehrfaches Sequenzen Alignment, das zu T konsistent ist, und M^* ein optimales mehrfaches Sequenzen Alignment von S. Dann gilt:*

$$\frac{D(M_c)}{D(M^*)} \leq 2 - \frac{2}{k}.$$

Beweis: Zuerst eine Vereinfachung:

$$\begin{aligned} D(M^*) &= \sum_{i=1}^{k} \sum_{j=i+1}^{k} D^*(s_i, s_j) = \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} D^*(s_i, s_j), \\ D(M_c) &= \sum_{i=1}^{k} \sum_{j=i+1}^{k} D(s_i, s_j) = \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} D(s_i, s_j). \end{aligned}$$

Dies folgt aus der Tatsache, dass $D(s_i, s_i) = 0 = D^*(s_i, s_i)$ sowie $D(s_i, s_j) = D(s_j, s_i)$ und $D^*(s_i, s_j) = D^*(s_j, s_i)$.

Dann gilt für den Quotienten:

$$\begin{aligned} \frac{D(M_c)}{D(M^*)} &= \frac{\frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} D(s_i, s_j)}{\frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} D^*(s_i, s_j)} \\ & \quad \text{da } D(s_i, s_i) = 0 \\ &= \frac{\sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} D(s_i, s_j)}{\sum_{i=1}^{k} \sum_{j=1}^{k} D^*(s_i, s_j)} \\ & \quad \text{mit Lemma 4.9 und } D^*(s_i, s_j) \geq d(s_i, s_j) \\ &\leq \frac{\sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} [d(s_i, s_c) + d(s_c, s_j)]}{\sum_{i=1}^{k} \underbrace{\sum_{j=1}^{k} d(s_i, s_j)}_{\text{minimal für} i=c}} \\ & \quad \text{Nach Wahl von } s_c \\ &\leq \frac{(k-1) \sum_{i=1}^{k} d(s_i, s_c) + (k-1) \sum_{j=1}^{k} d(s_c, s_j)}{\sum_{i=1}^{k} \sum_{j=1}^{k} d(s_c, s_j)} \\ &= \frac{2(k-1)}{k} * \underbrace{\frac{\sum_{i=1}^{k} d(s_i, s_c)}{\sum_{j=1}^{k} d(s_c, s_j)}}_{=1} \end{aligned}$$

$$= \frac{2k-2}{k}$$
$$= 2 - \frac{2}{k}.$$

■

4.4.3 Laufzeit für Center-Star-Methode

Wie groß ist die Laufzeit der Center-Star-Methode? Für die Bestimmung des Centers müssen wir für jede Sequenz die Summe der paarweisen Distanzen zu den anderen Sequenzen berechnen. Dies kostet pro Sequenz $(k-1) \cdot O(n^2) = O(kn^2)$. Für alle Sequenzen ergibt sich daher $O(k^2n^2)$. Für die Konstruktion des mehrfachen Sequenzen Alignments, das konsistent zum Stern mit dem gewählten Center-String als Zentrum ist, benötigen wir nur noch $O(kn^2)$.

Der Gesamtzeitbedarf ist also $O(k^2n^2)$, wobei die meiste Zeit für die Auswahl des Zentrums verbraucht wurde.

Theorem 4.11 *Die Center-Star-Methode liefert eine $(2-\frac{2}{k})$-Approximation für ein optimales mehrfaches Sequenzen Alignment für k Sequenzen der Länge $\Theta(n)$ in Zeit $O(k^2n^2)$.*

4.4.4 Randomisierte Varianten

Wir wollen im Folgenden zeigen, dass man nur einige Zentren ausprobieren muss und dann bereits der beste der ausprobierten Zentrum schon fast eine 2-Approximation liefert. Wir können also die Laufzeit noch einmal senken.

Theorem 4.12 *Für r sei $C(r)$ die erwartete Anzahl von Sternen, die zufällig gewählt werden müssen, bis das beste mehrfache Sequenzen Alignment, der mit der Center-Star-Methode und den gewählten Zentren, eine $(2+\frac{1}{r-1})$-Approximation ist. Dann ist $C(r) \leq r$.*

Für den Beweis (und den nächsten Satzes) benötigen wir das folgende Lemma.

Theorem 4.13 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen. Es existieren mehr als $\left\lfloor \frac{k}{r} \right\rfloor$ Sterne mit $M(i) \leq \frac{2r-1}{r-1}M$. Hierbei ist $M(i) := \sum_{j=1}^{k} d(s_i, s_j)$ und $M := \min\{M(i) \mid i \in [1:k]\}$.*

Beweis: Wir berechnen zuerst den Mittelwert von $M(i)$:

$$\begin{aligned}
\frac{1}{k}\sum_{i=1}^{k} M(i) &= \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k} d(s_i,s_j) \\
&\quad \text{mit } d(s_i,s_i)=0 \\
&= \frac{1}{k}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(s_i,s_j) \\
&\quad \text{mit Hilfe der Dreiecksungleichung} \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(s_i,s_c)+d(s_c,s_j) \\
&\quad \text{da } s_c \text{ ist Zentrum eines optimalen Sterns} \\
&= \frac{1}{k}2(k-1)M \\
&< 2M
\end{aligned}$$

Wir führen den Beweis des Lemmas mit Hilfe eines Widerspruchsbeweises.

Annahme: Es existieren maximal $\left\lfloor \frac{k}{r} \right\rfloor$ Sterne mit $M(i) \leq \frac{2r-1}{r-1}M$.

Dann gilt:

$$\begin{aligned}
2M &> \frac{1}{k}\sum_{i=1}^{k} M(i) \\
&\quad \text{mit Hilfe der Widerspruchsannahme} \\
&\geq \frac{1}{k}\left(\frac{k}{r}M + \left(k-\frac{k}{r}\right)\frac{2r-1}{r-1}M\right) \\
&= M\left(\frac{1}{r} + \underbrace{\left(1-\frac{1}{r}\right)}_{=\frac{r-1}{r}}\frac{2r-1}{r-1}\right) \\
&= \frac{M}{r}(1+2r-1) \\
&= 2M.
\end{aligned}$$

Also gilt $2M < 2M$, was offensichtlich der gewünschte Widerspruch ist. ■

Beweis von Satz 4.12: Im Folgenden gelte ohne Beschränkung der Allgemeinheit, dass $M(1) \leq M(2) \leq \cdots M(k)$. Sei $c := \left\lfloor \frac{k}{r} \right\rfloor + 1$, dann gilt aufgrund von Lemma 4.13:

$$M(c) = \varepsilon \cdot M \text{ mit } \varepsilon \in \left[1 : \frac{2r-1}{r-1}\right]$$

Damit gilt:

$$
\begin{aligned}
\frac{D(M_c)}{D(M^*)} &\leq \frac{\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} D(s_i,s_j)}{\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} D^*(s_i,s_j)} \\
&\quad \text{da } D(s_k,s_j)=0 \\
&\leq \frac{\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} D(s_i,s_j)}{\sum_{i=1}^{k}\sum_{j=1}^{k} D^*(s_i,s_j)} \\
&\quad \text{da } D(s_i,s_j) \leq d(s_i,s_c)+d(s_c,s_j) \text{ und } D(s_i,s_j) \geq d(s_i,s_j) \\
&\leq \frac{\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} [d(s_i,s_c)+d(s_c,s_j)]}{\sum_{i=1}^{k}\sum_{j=1}^{k} d(s_i,s_j)} \\
&\leq \frac{2\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(s_i,s_c)}{\sum_{i=1}^{k}\sum_{j=1}^{k} d(s_i,s_j)} \\
&\quad \text{da } M(i) \geq M \text{ für alle } i \text{ und } M(i) \geq \varepsilon M \text{ für } i > c \geq \tfrac{k}{r} \\
&\leq \frac{2(k-1)M(c)}{\sum_{i=1}^{k} M(i)} \\
&\leq \frac{2(k-1)\cdot\varepsilon\cdot M}{\frac{k}{r}\cdot M + \frac{kr-k}{r}\cdot\varepsilon\cdot M} \\
&\leq \frac{2r(k-1)\cdot\varepsilon}{k+k(r-1)\cdot\varepsilon} \\
&\leq \frac{2r(k-1)}{k/\varepsilon + k(r-1)} \\
&\quad \text{da } \varepsilon \leq \tfrac{2r-1}{r-1} \\
&\leq \frac{2r(k-1)}{k\frac{r-1}{2r-1} + k(r-1)} \\
&\leq \frac{2r(k-1)(2r-1)}{k(r-1)+k(r-1)(2r-1)} \\
&\leq \frac{2r(k-1)(2r-1)}{2rk(r-1)} \\
&\leq \frac{(k-1)}{k}\cdot\frac{(2r-1)}{(r-1)} \\
&\leq \frac{(2r-1)}{(r-1)} \\
&\leq 2+\frac{1}{(r-1)}
\end{aligned}
$$

■

Theorem 4.14 *Wählt man p Sterne (d.h. ihre Zentren) zufällig aus, dann ist das beste mehrfache Sequenzen Alignment, das von diesen Sternen generiert wird, eine* $(2+\frac{1}{r-1})$*-Approximation mit der Wahrscheinlichkeit größer gleich* $1-\left(\frac{r-1}{r}\right)^p$.

Beweis: Es gilt mit Lemma 4.13

$$\mathrm{Ws}[\underbrace{\text{schlechter Stern}}_{\text{liefert keine } (2+\frac{1}{r-1})\text{-Approximation}}] \leq \frac{k-\frac{k}{r}}{k} = \frac{r-1}{r}.$$

Daraus folgt sofort

$$\mathrm{Ws}[\text{Es werden p schlechte Sterne gewählt}] \leq \left(\frac{r-1}{r}\right)^p,$$

und somit

$$Ws[\text{Einer der p Sterne liefert } (2+\frac{1}{r-1})\text{Approximation}] \geq 1-\left(\frac{r-1}{r}\right)^p.$$

■

Bsp:

Approximation	Fehler	p
2,2 ($r=6$)	$< 1\%$	≈ 13
2,1 ($r=11$)	$< 1\%$	≈ 25

4.5 Konsensus eines mehrfachen Alignments

Nachdem wir nun Möglichkeiten kennen gelernt haben, wie wir mehrfache Sequenzen Alignments effizient konstruieren können, wollen wir uns jetzt damit beschäftigen, wie wir daraus eine Referenz-Sequenz (einen so genannten Konsensus-String) ableiten können.

4.5.1 Konsensus-Fehler und Steiner-Strings

Definition 4.15 *Seien $S = \{s_1, \ldots, s_k\}$ Sequenzen über Σ und $s' \in \Sigma^*$ eine beliebige Zeichenreihe. Der* Konsensus-Fehler *von s' zu S ist definiert durch*

$$E_S(s') := \sum_{j=1}^{k} d(s', s_j).$$

Ein optimaler Steiner-String *s^* für S ist eine Sequenz aus Σ^* mit minimalem Konsensus-Fehler*

$$E_S(s^*) := \min\{E_S(s') \mid s' \in \Sigma^*\}.$$

Im Allgemeinen ist s^* nicht eindeutig und es gilt $s^* \notin S$. Dennoch kann s^* in einigen wenigen Fällen durchaus eindeutig sein bzw. $s^* \in S$ sein.

Lemma 4.16 *Sei $S = \{s_1, \ldots, s_k\}$ und d sei eine Metrik. Dann existiert ein $s' \in S$ mit*

$$\frac{E_S(s')}{E_S(s^*)} \leq 2 - \frac{2}{k},$$

wobei s^* ein Steiner-String für S ist.

Beweis: Wähle $s_i \in S$ beliebig, aber fest.

$$\begin{aligned}
E_S(s_i) &= \sum_{j=1}^{k} d(s_i, s_j) \\
&\quad \text{da } d(s_i, s_i) = 0 \\
&= \sum_{\substack{j=1 \\ j \neq i}}^{k} d(s_i, s_j) \\
&\quad \text{aufgrund der Dreiecksungleichung} \\
&\leq \sum_{\substack{j=1 \\ j \neq i}}^{k} d(s_i, s^*) + \sum_{\substack{j=1 \\ j \neq i}}^{k} d(s^*, s_j) \\
&= (k-1)d(s_i, s^*) + E_S(s^*) - d(s^*, s_i) \\
&= (k-2)d(s_i, s^*) + E_S(s^*).
\end{aligned}$$

Sei s_i nun so gewählt, dass $d(s_i, s^*) \leq d(s_j, s^*)$ für alle $j \in [1 : k]$. Dann gilt:

$$\begin{aligned} E_S(s^*) &= \sum_{j=1}^{k} d(s^*, s_j) \\ &\geq \sum_{j=1}^{k} d(s^*, s_i) \\ &= k \cdot d(s^*, s_i) \end{aligned}$$

Fassen wir beide Zwischenergebnisse zusammen, dann gilt:

$$\begin{aligned} \frac{E_S(s_i)}{E_S(s^*)} &\leq \frac{(k-2)d(s_i, s^*) + E_S(s^*)}{E_S(s^*)} \\ &= 1 + \frac{(k-2)d(s_i, s^*)}{E_S(s^*)} \\ &\leq 1 + \frac{(k-2)d(s_i, s^*)}{k \cdot d(s^*, s_i)} \\ &= 1 + \frac{k-2}{k} \\ &= 2 - \frac{2}{k}. \end{aligned}$$

■

Da für den Center-String s_c gilt, dass $\sum_{j=1}^{k} d(s_u, s_j)$ für $i = c$ minimal wird:

$$\sum_{j=1}^{k} d(s_i, s_j) \geq \sum_{j=1}^{k} d(s_c, s_j) = E_S(s_c).$$

Damit gilt

$$E_S(s_c) \leq E_S(s_i),$$

wobei s_i aus Lemma 4.16 ist. Somit erhalten wir das folgende Korollar:

Korollar 4.17 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ und sei s_c ein Center-String von S und s^* ein optimaler Steiner-String für S, dann gilt*

$$\frac{E_S(s_c)}{E_S(s^*)} \leq 2 - \frac{2}{k}.$$

Theorem 4.18 *Für r sei $C(r)$ die erwartete Anzahl von Sternen (Zentren), die zufällig gewählt werden müssen, bis der beste Konsensus-Fehler bis auf den Faktor $(2 + \frac{1}{r-1})$vom Optimum entfernt ist. Dann gilt $C(r) \leq r$.*

Zum Beweis des Satze benötigen wir das folgende Lemma.

Lemma 4.19 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen über Σ . Es existieren mehr als $\left\lfloor \frac{k}{r} \right\rfloor$ Sterne mit $E_S(s_i) \leq \frac{2r-1}{r-1} E_S(s^*)$. Hierbei ist $E_S(s_i) := \sum_{j=1}^{k} d(s_i, s_j)$ und s^* ist ein Konsensus-String.*

Beweis: Wir berechnen zuerst den Mittelwert von $E(s_i)$:

$$
\begin{aligned}
\frac{1}{k}\sum_{i=1}^{k} E_S(s_i) &= \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k} d(s_i, s_j) \\
&\quad \text{mit } d(s_i, s_i) = 0 \\
&= \frac{1}{k}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(s_i, s_j) \\
&\quad \text{mit Hilfe der Dreiecksungleichung} \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(s_i, s^*) + d(s^*, s_j) \\
&\quad \text{da } s^* \text{ ein optimaler Steiner-String} \\
&= \frac{1}{k} 2(k-1) E_S(s^*) \\
&< 2E_S(s^*)
\end{aligned}
$$

Wir führen den Beweis des Lemmas mit Hilfe eines Widerspruchsbeweises.

Annahme: Es existieren maximal $\left\lfloor \frac{k}{r} \right\rfloor$ Sterne mit $E_S(s_i) \leq \frac{2r-1}{r-1} E_S(s^*)$.

Dann gilt:

$$
\begin{aligned}
2E_S(s^*) &> \frac{1}{k}\sum_{i=1}^{k} E_S(S_i) \\
&\quad \text{mit Hilfe der Widerspruchsannahme} \\
&\geq \frac{1}{k}\left(\frac{k}{r} E_S(s^*) + \left(k - \frac{k}{r}\right)\frac{2r-1}{r-1} E_S(s^*)\right)
\end{aligned}
$$

$$
\begin{aligned}
&= E_S(s^*)\left(\frac{1}{r} + \underbrace{(1-\frac{1}{r})}_{=\frac{r-1}{r}}\frac{2r-1}{r-1}\right)\\
&= \frac{E_S(s^*)}{r}(1+2r-1)\\
&= 2E_S(s^*).
\end{aligned}
$$

Also gilt $2E_S(s^*) < 2E_S(s^*)$, was offensichtlich der gewünschte Widerspruch ist. ■

Beweis von Satz 4.18: Es gelte ohne Beschränkung der Allgemeinheit

$$E_S(s_1) \leq E_S(s_2) \leq \cdots \leq E_S(s_k).$$

Mit $c = \lfloor \frac{k}{r} \rfloor + 1$ gilt $E_S(S_c) \leq \frac{2r-1}{r-1} \cdot E_S(s^*)$. ■

Damit stellen Steiner-Strings also eine Möglichkeit dar, für eine Folge von Sequenzen eine Referenz-Sequenz zu generieren.

4.5.2 Alignment-Fehler und Konsensus-String

Jetzt stellen wir eine weitere Methode vor, die auf mehrfachen Sequenzen Alignments basiert.

Definition 4.20 *Sei $M = (\overline{s}_1, \ldots, \overline{s}_k)$ ein mehrfaches Sequenzen Alignment für $S = \{s_1, \ldots, s_k\}$. Das* Konsensus-Zeichen *an der Position i ist das Zeichen $x \in \overline{\Sigma}$ mit*

$$\sum_{j=1}^{k} w(x, \overline{s}_{j,i}) = \min\left\{\sum_{j=1}^{k} w(a, \overline{s}_{j,i}) \; : \; a \in \overline{\Sigma}\right\} =: \; \delta_M(i).$$

Zur Erinnerung: $w : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$ war eine Kostenfunktion, bei der für mehrfache Sequenzen Alignments $w(-,-) = 0$ gilt.

Definition 4.21 *Der* Konsensus-String $\mathcal{S}_M$ *eines mehrfachen Sequenzen Alignments M für S ist $\mathcal{S}_M := s_1 \cdots s_m$, wobei s_i das Konsensus-Zeichen an der Position i ist.*

Definition 4.22 *Sei* $S = \{s_1, \ldots, s_k\}$ *eine Menge von Sequenzen über* Σ *und sei* $M = (\overline{s}_1, \ldots, \overline{s}_k)$ *ein mehrfaches Sequenzen Alignment für* S *mit* $n = |\overline{s}_1| = \cdots = |\overline{s}_k|$, *dann ist der* Alignment-Fehler *einer Sequenz* $s \in \Sigma^n$ *definiert als*

$$E_M(s) = \sum_{j=1}^{k} \sum_{i=1}^{n} w(s_i, \overline{s}_{j,i}),$$

wobei w *wieder die zugrunde liegende Kostenfunktion ist. Speziell gilt dann*

$$E_M(\mathcal{S}_M) = \sum_{i=1}^{m} \delta_M(i).$$

Definition 4.23 *Sei* $S = \{s_1, \ldots, s_k\}$ *eine Menge von Sequenzen. Das optimale* Konsensus-MSA *für* S *ist ein mehrfaches Sequenzen Alignment* M *für* S *mit minimalem Alignment-Fehler*

4.5.3 Beziehung zwischen Steiner-String und Konsensus-String

Wir haben jetzt mehrere Definitionen für einen *Konsensus-String* kennen gelernt:

- Steiner-String (ohne Mehrfaches Sequenzen Alignment!);
- Konsensus-String für ein Mehrfaches Sequenzen Alignment M (die Optimalität wird hierbei bezüglich minimalem Alignment-Fehler definiert);
- Konsensus-String für ein Mehrfaches Sequenzen Alignment M (die Optimalität wird hierbei bezüglich des Sum-of-Pairs-Maßes definiert).

Theorem 4.24 *i) Sei* s' *der Konsensus-String eines optimalen Konsensus-MSA für* S, *dann ist* $s'|_\Sigma$ *ein optimaler Steiner-String für* S.
ii) Ist M *ein mehrfaches Sequenzen Alignment für* $S \cup \{s^*\}$, *das konsistent zu einem Stern mit Zentrum* s^* *ist, dann ist* M *ohne die Zeile für* s^* *ein optimales Konsensus-MSA, wenn* s^* *ein optimaler Steiner-String für* S *ist.*

Beweis: Sei $S = \{s_1 \ldots s_k\}$ und sei $M = (\overline{s}_1, \ldots, \overline{s}_k)$ ein *beliebiges* mehrfaches Sequenzen Alignment für S. Sei $\mathcal{S}_M$ der Konsensus-String für M.

Für das induzierte paarweise Alignment von $\mathcal{S}_M$ mit $\overline{s}_j$ gilt:

$$D(\mathcal{S}_M, \overline{s}_j) \;\geq\; d(\mathcal{S}_M, s_j)$$

Also gilt

$$\begin{aligned}
E_M(\mathcal{S}_M) &= \sum_{i=1}^{n} \delta_M(i) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k} w(\mathcal{S}_{M,i}, \overline{s}_{j,i}) \\
&= \sum_{j=1}^{k} \sum_{i=1}^{n} w(\mathcal{S}_{M,i}, \overline{s}_{j,i}) \\
&= \sum_{j=1}^{k} D(\mathcal{S}_M, \overline{s}_j) \\
&\geq \sum_{j=1}^{k} d(\mathcal{S}_M, s_j) \\
&= E_S(\mathcal{S}_M) \\
&\geq E_S(s^*).
\end{aligned}$$

Hierbei ist s^* ein optimaler Steiner-String für S. Prinzipiell gilt also, dass der Alignment-Fehler des Konsensus-Strings für M mindestens so groß ist wie dessen Konsensus-Fehler.

Sei jetzt M^* ein mehrfaches Sequenzen Alignment für $S \cup \{s^*\}$, das konsistent zu einem Stern mit Zentrum s^* ist, wobei s^* ein optimaler Steiner-String für S ist.

Für das induzierte Alignment $\overline{s^*}$ mit $\overline{s}_j$ gilt:

$$D(\overline{s^*}, \overline{s}_j) = d(s^*, s_j), \tag{4.1}$$

da M konsistent zu einem Stern mit s^* als Zentrum ist.

Sei M das mehrfache Sequenzen Alignment, das aus M^* durch Streichen von $\overline{s^*}$ entsteht. Mit $n = |\overline{s^*}|$ gilt:

$$\begin{aligned}
E_M(\overline{s^*}) &= \sum_{i=1}^{n} \sum_{j=1}^{k} w(\overline{s^*}_i, \overline{s}_{j,i}) \\
&= \sum_{j=1}^{k} \underbrace{\sum_{i=1}^{n} w(\overline{s^*}_i, \overline{s}_{j,i})}_{D_{M^*}(s^*, s_j)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{k} D_{M^*}(\overline{s^*}, s_j) \\
&\quad \text{mit Hilfe der Gleichung 4.1} \\
&= \sum_{j=1}^{k} d(s^*, s_j) \\
&= E_S(s^*).
\end{aligned}
$$

Es gibt also ein mehrfaches Sequenzen Alignment M für S, dessen Alignment-Fehler des zugehörigen Konsensus-Strings gleich dem Konsensus-Fehler von S ist.

Damit folgt die zweite Behauptung des Satzes, da M ein optimales Konsensus-MSA ist, da $E_M(s^*)$ minimal ist.

Die erste Behauptung folgt, da es eine Konsensus-MSA mit Alignment-Fehler $E_S(s^*)$ gibt und da für ein optimales Konsensus-MSA $\bar{M}$ gilt:

$$E_S(s^*) = E_{\bar{M}}(\mathcal{S}_{\bar{M}}) \overset{Beh.}{\geq} E_S(\mathcal{S}_{\bar{M}}|_\Sigma) \geq E_S(s^*).$$

Also ist $\mathcal{S}_{\bar{M}}|_\Sigma$ auch ein optimaler Steiner-String. ■

Es gilt:

$$\frac{E_S(s_c)}{E_S(s^*)} \leq 2 - \frac{2}{k},$$

wobei s_c ein Center-String von S ist und s^* ein optimaler Steiner-String ist.

Fassen wir das Ergebnis dieses Abschnitts noch einmal zusammen.

Theorem 4.25 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen. Das mehrfache Sequenzen Alignment M_c für S, das mit Hilfe der Center-Star-Methode konstruiert wurde, hat eine Sum-of-Pairs-Distanz, die maximal $(2 - \frac{2}{k})$ vom Optimum entfernt ist, und es hat einen Alignment-Fehler, der maximal $(2 - \frac{2}{k})$ vom Optimum entfernt ist.*

4.6 Phylogenetische Alignments

Im letzten Abschnitt haben wir gesehen, wie wir mit Hilfe mehrfacher Sequenzen Alignments, die zu Sternen konsistent sind, eine Approximation für ein optimales mehrfaches Sequenzen Alignment oder einen Konsensus-String konstruieren können. Manchmal ist für die gegebenen Sequenzen ja mehr bekannt, zum Beispiel ein phylogenetischer Baum der zugehörigen Spezies. Diesen könnte man für die Konstruktion von Sequenzen Alignments ja ausnutzen.

4.6.1 Definition phylogenetischer Alignments

Wir werden jetzt Alignments konstruieren, die wieder zu Bäumen konsistent sind. Allerdings sind jetzt nur die Sequenzen an den Blättern bekannt und die inneren Knoten sind ohne Sequenzen. Dies folgt daher, da für einen phylogenetischen Baum in der Regel nur die Sequenz der momentan noch nicht ausgestorbenen Spezies bekannt sind, und das sind genau diejenigen, die an den Blättern stehen. An den inneren Knoten stehen ja die Sequenzen, von den Vorahnen der bekannten Spezies, die in aller Regel heutzutage ausgestorben sind.

Definition 4.26 *Sei* $S = \{s_1, \ldots, s_k\}$ *eine Menge von Sequenzen über* Σ *und* $T = (V, E)$ *ein Baum mit*

$$V = I \cup B, \text{ wobei } \begin{array}{ll} I = \{v \in V | deg(v) > 1\} & \textit{(innere Knoten)} \\ B = \{v \in V | deg(v) = 1\} & \textit{(Blätter)} \end{array}$$

Weiterhin existiert eine Bijektion $\varphi : B \to S$ *mit*

$$\varphi : B \to S : v \mapsto s_v$$

Einen solcher Baum T *heißt* konsistent *zu* S.

Ein phylogenetisches mehrfaches Sequenzen Alignment *(kurz:* PMSA*) ist eine Zuordnung von Zeichenreihen aus* Σ^* *an* I*, d.h.*

$$\varphi : I \to \Sigma^* : v \mapsto s_v$$

und ein mehrfaches Sequenzen Alignment, das mit T *konsistent ist.*

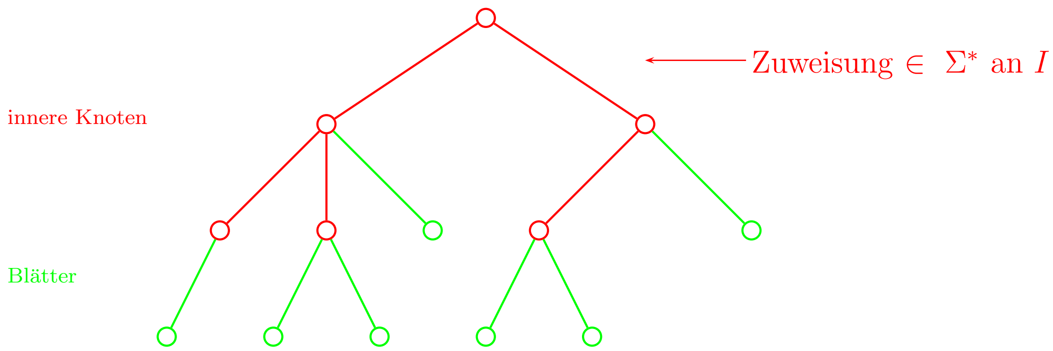

Abbildung 4.6: Skizze: Phylogenetisches mehrfaches Sequenzen Alignment

Für $(v, w) \in E(T)$ bezeiche $D_M(v, w) := d(s_v, s_w)$ die Alignment-Distanz des induzierten Alignments aus einem PMSA M für die zu v und w zugeordneten Sequenzen. Da das mehrfache Sequenzen Alignment zu T konsistent ist, entspricht diese Alignment-Distanz des induzierten Alignments der Alignment-Distanz des optimalen paarweisen Sequenzen Alignments.

Definition 4.27 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen und T ein zu S konsistenter Baum. Für ein PMSA M für S, das zu T konsistent ist, bezeichnet*

$$D_M(T) := \sum_{e \in E(T)} D_M(e) = \sum_{(v,w) \in E(T)} D_M((v,w))$$

die Distanz eines PMSA.

Definition 4.28 *Ein optimales phylogenetisches mehrfaches Sequenzen Alignment ist ein phylogenetisches mehrfaches Sequenzen Alignment M für T, das $D_M(T)$ minimiert.*

Leider ist auch hier wieder die Entscheidung, ob es ein phylogenetisches mehrfaches Sequenzen Alignment mit einer Distanz kleiner gleich D gibt, ein $\mathcal{NP}$-hartes Problem. Wir können also auch hierfür wieder nicht auf ein effizientes Verfahren für eine optimale Lösung hoffen.

4.6.2 Geliftete Alignments

Um wieder eine Approximation generieren zu können, betrachten wir so genannte geliftete Alignments. Ähnlich wie wir bei der Center-Star-Methode für das Zentrum eine Sequenz aus der Menge S wählen, werden wir uns bei der Zuordnung der Sequenzen an die inneren Knoten auch wieder auf die Sequenzen aus S beschränken. Wir schränken uns sogar noch ein wenig mehr ein.

Definition 4.29 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen und T ein zu S konsistenter Baum. Ein phylogenetisches mehrfaches Sequenzen Alignment heißt* geliftet, *wenn für jeden inneren Knoten $v \in I$ gilt, dass ein Knoten $w \in V(T)$ mit $s_v = s_w$ und $(v, w) \in E(T)$ existiert.*

4.6.3 Konstruktion eines gelifteten aus einem optimalem Alignment

Nun zeigen wir, wie wir aus aus einem optimalen phylogenetischen mehrfachen Sequenzen Alignment ein geliftetes konstruieren. Dies ist an und für sich nicht sinnvoll, da wir mit einem optimalen Alignment natürlich glücklich wären und damit an dem gelifteten kein Interesse mehr hätten. Wir werden aber nachher sehen, dass uns diese Konstruktion beim Beweis der Approximationsgüte behilflich sein wird.

Definition 4.30 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen und T ein zu S konsistenter Baum. Ein Knoten v heißt geliftet, wenn ein Knoten $w \in V(T)$ mit $s_v = s_w$ und $(v, w) \in E(T)$ existiert.*

Zu Beginn sind alle Blätter geliftet. Wir betrachten jetzt einen Knoten v, so dass alle seine Kinder geliftet sind und werden diesen Knoten selbst liften.

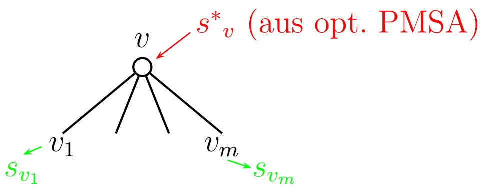

Abbildung 4.7: Skizze: Liften eines Knotens

Wir ersetzen jetzt die Zeichenreihe $s^*{}_v$ des Knotens v durch die Zeichenreihe s_{v_j} seines Kindes v_j, wobei $d(s_{v_j}, s^*{}_v) \leq d(s_{v_i}, s^*{}_v)$ für alle $i \in [1 : m]$ gelten soll. Wir ersetzen also die Zeichenreihe s^*_v des Knotens v durch die Zeichenreihe eines seiner Kinder, die zu s^*_v am nächsten ist (im Sinne der Alignment-Distanz).

4.6.4 Güte gelifteter Alignments

Definition 4.31 *Sei $S = \{s_1, \ldots, s_k\}$ eine Menge von Sequenzen und sei und sei T ein zu S konsistenter Baum. Ist T^* ein optimales PMSA M^* für S und T, dann gilt für das aus T^* konstruierte geliftete PMSA M_L T^L:*

$$D_{M_L}(T^L) \leq 2 \cdot D_{M^*}(T^*)$$

Beweis: Wir betrachten zuerst eine Kante $(v, w) \in E(T)$. Gilt $s_v = s_w$, dann ist logischerweise $D(v, w) = d(s_v, s_w) = 0$. Ist andernfalls $s_v \neq s_w$, dann gilt

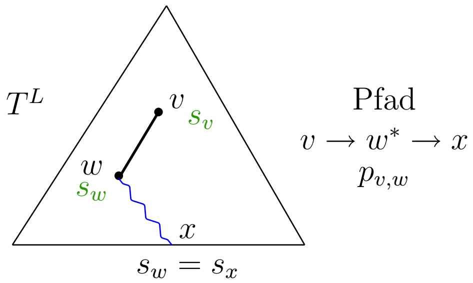

Abbildung 4.8: Skizze: Kante (v, w) definiert Pfad $p_{v,w}$

$D(v, w) = d(s_v, s_w) \leq d(s_v, s^*{}_v) + d(s^*{}_v, s_w)$. Aufgrund des Liftings gilt außerdem: $d(s^*{}_v, s_v) \leq d(s^*{}_v, s_w)$. Somit erhalten wir insgesamt:

$$D(v, w) = d(s_v, s_w) \leq d(s_v, s^*{}_v) + d(s^*{}_v, s_w) \leq 2 \cdot d(s^*{}_v, s_w).$$

Betrachten wir eine Kante (v, w) in T^L, wobei w ein Kind von v ist. Diese Kante definiert in T^L einen Pfad $p_{v,w}$ von w zu einem Blatt x , indem wir vom Knoten w aus immer zu dem Kind gehen, das ebenfalls mit der Sequenz s_w markiert ist. Letztendlich landen wir dann im Blatt x. Dieser Pfad ist eindeutig, da wir angenommen haben, dass die Sequenzen aus S paarweise verschieden sind.

Betrachten wir jetzt diesen Pfad $p_{v,w}$ mit $v \to w = w_1 \to \cdots \to w_\ell = x$ in einem optimalen phylogenetischen mehrfachen Sequenzen Alignment, dann gilt aufgrund der Dreiecksungleichung:

$$\begin{aligned} d(s^*{}_v, s_w) &\leq d(s^*{}_v, s^*{}_{w_1}) + d(s^*{}_{w_1}, s^*{}_{w_2}) + \cdots + d(s^*{}_{w_{\ell-1}}, \underbrace{s^*{}_x}_{s^*_x = s_{w_\ell}}) \\ &\leq D^*(p_{v,w}) \end{aligned}$$

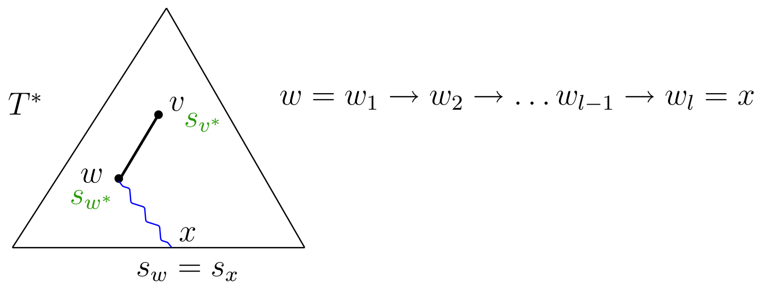

Abbildung 4.9: Skizze: Pfad $p_{v,w}$ im optimalen Baum T^*

Insgesamt erhalten wir dann

$$D(v, w) \leq 2d({s^*}_v, s_w) \leq 2D^*(p_{v,w}).$$

Wir können also die Distanz des gelifteten phylogenetischen Alignments geschickt wie folgt berechnen: Für alle grünen Kanten e gilt $D(e) = 0$. Wir müssen also nur

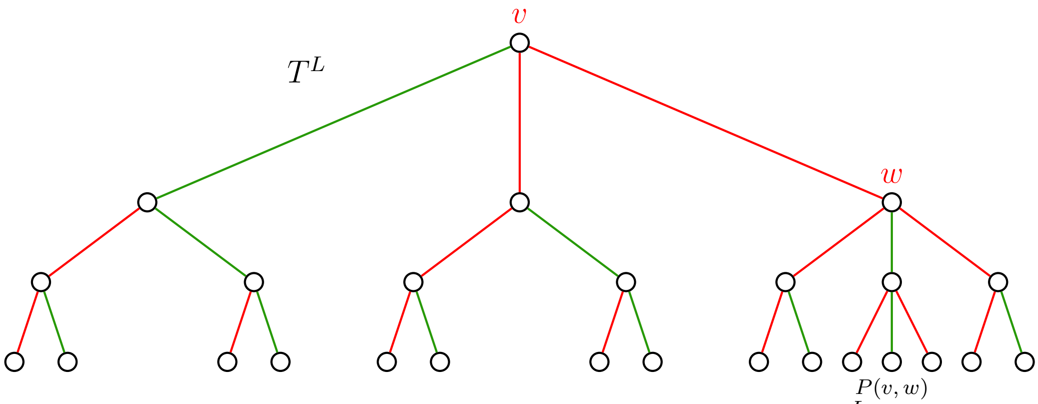

Abbildung 4.10: Skizze: Beziehung Kantengewichte in T^L zu T^*

die roten Kanten in T^L aufaddieren. Jede rote Kante (v, w) in T^L korrespondiert zu einem Pfad $p_{v,w}$ in T^*. Es gilt weiter, dass für alle Kanten (v, w), für die in T^L $D(v, w) > 0$ ist, die zueghörigen Pfade disjunkt sind. Somit kann die Summe der roten Kantengewichte durch die doppelte Summe aller Kantengewichte im Baum T^* abgeschätzt werden. Damit ergibt sich sich nun Folgendes:

$$\begin{aligned}
D(T^L) &= \sum_{(v,w)\in E(T)} D(v, w) \\
&= \sum_{\substack{(v,w)\in E(T) \\ D(v,w)>0}} \underbrace{D(v, w)}_{\leq 2D^*(p_{v,w})} \\
&\leq \underbrace{2 \cdot \sum_{\substack{(v,w)\in E(T) \\ D(v,w)>0}} D^*(p_{v,w})}_{\leq D(T^*)} \\
&\leq 2 \cdot D(T^*).
\end{aligned}$$

■

Damit haben wir gezeigt, dass es ein geliftetes phylogenetisches Sequenzen Alignment gibt, dass höchstens um den Faktor zwei vom Optimum entfernt ist. Wenn wir jetzt ein optimales geliftetes phylogenetisches mehrfaches Sequenzen Alignment konstruieren, so gilt dies natürlich auch für dieses.

4.6.5 Berechnung eines optimalen gelifteten PMSA

Wir wollen jetzt mit Hilfe der Dynamischen Programmierung ein optimales phylogenetisches mehrfaches Sequenzen Alignment konstruieren. Dazu stellen wir zunächst wieder einmal eine Rekursionsgleichung auf. Hierfür bezeichnet $D(v, s)$ den Wert eines besten gelifteten PMSA für den am Knoten v gewurzelten Teilbaum, so dass v mit der Sequenzen $s \in S$ markiert ist (dabei muss natürlich s an einem der Kinder von v bereits vorkommen). Es gilt dann:

$$D(v,s) = \begin{cases} \displaystyle\sum_{(v,w)\in E(T)} d(s, s_w) & \text{wenn alle Kinder von } v \text{ Blätter sind} \\ \displaystyle\sum_{(v,w)\in E(T)} \min\left\{ d(s,s') + D(w,s') : \begin{array}{c} s' \text{ ist eine Markierung} \\ \text{eines Blattes in } T_v \end{array} \right\} & \end{cases}$$

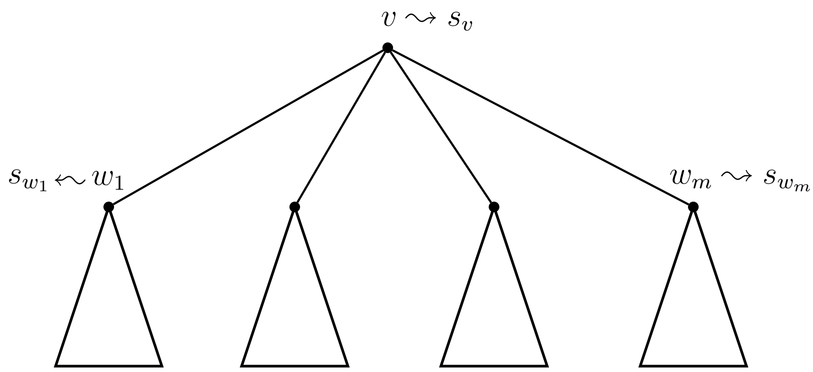

Abbildung 4.11: Skizze: Berechnung von $D(v, s)$

Während des Preprocessings ist es nötig, für alle Paare $(s, s') \in S^2$ $d(s, s')$ zu berechnen. Dafür ergibt sich folgender Zeitbedarf:

$$O(\sum_{i=1}^{k}\sum_{j=1}^{k}|s_i| * |s_j|) = O(\underbrace{\sum_{i=1}^{k}|s_i|}_{N}\underbrace{\sum_{j=1}^{k}|s_j|}_{N}) = O(N^2)$$

Nach dem Preprocessing kann jede Minimumbildung in konstanter Zeit erfolgen. Wir müssen uns nur noch überlegen, wie oft das Minimum gebildet wird. Dies geschieht für jede Baumkante (v, w) und jedes Paar von Sequenzen (s, s') genau einmal. Da ein binärer Baum, bei dem es keine Knoten mit genau einem Kind gibt, höchstens so viele innere Knoten wie Blätter besitzt, hat der Baum maximal $O(k)$ Knoten. Weiterhin gilt, dass jeder Baum weniger Kanten als Knoten besitzt und es somit maximal $O(k)$ Kanten in T gibt. Offensichtlich gibt es k^2 Paare von Sequenzen aus

S. Also gilt insgesamt für die Laufzeit: $O(N^2 + k^3)$. Fassen wir das Ergebnis noch zusammen.

Theorem 4.32 *Sei $S = \{s_1, \ldots, s_k\}$ eine k-elementige Menge von Sequenzen über Σ mit $N = \sum_{i=1}^{k} |s_i|$ und sei und sei T ein zu S konsistenter Baum. Ein PMSA für S und T, dessen Distanz maximal um 2 von einem optimalen PMSA für S und T abweicht, kann in Zeit $O(N^2 + k^3)$ konstruiert werden.*

Mit etwas Aufwand kann man den Summanden k^3 noch auf k^2 drücken.

Literaturhinweise

A.1 Lehrbücher zur Vorlesung

Peter Clote, Rolf Backofen: *Introduction to Computational Biology*; John Wiley and Sons, 2000.

Richard Durbin, Sean Eddy, Anders Krogh, Graeme Mitchison: *Biological Sequence Analysis*; Cambridge University Press, 1998.

Dan Gusfield: *Algorithms on Strings, Trees, and Sequences — Computer Science and Computational Biology*; Cambridge University Press, 1997.

David W. Mount: *Bioinformatics — Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2001.

Pavel A. Pevzner: *Computational Molecular Biology - An Algorithmic Approach*; MIT Press, 2000.

João Carlos Setubal, João Meidanis: *Introduction to Computational Molecular Biology*; PWS Publishing Company, 1997.

Michael S. Waterman: *Introduction to Computational Biology: Maps, Sequences, and Genomes*; Chapman and Hall, 1995.

A.2 Skripten anderer Universitäten

Bonnie Berger: *Introduction to Computational Molecular Biology*, Massachusetts Institute of Technology, http://theory.lcs.mit.edu/~bab/01-18.417-home.html;

Bonnie Berger, *Topics in Computational Molecular Biology*, Massachusetts Institute of Technology, Spring 2001, http://theory.lcs.mit.edu/~bab/01-18.418-home.html;

Paul Fischer: *Einführung in die Bioinformatik* Universität Dortmund, Lehrstuhl II, WS2001/2002, http://ls2-www.cs.uni-dortmund.de/lehre/winter200102/bioinf/

Richard Karp, Larry Ruzzo: *Algorithms in Molecular Biology*; CSE 590BI, University of Washington, Winter 1998. http://www.cs.washington.edu/education/courses/590bi/98wi/

Larry Ruzzo: *Computational Biology*, CSE 527, University of Washington, Fall 2001; http://www.cs.washington.edu/education/courses/527/01au/

Georg Schnittger: *Algorithmen der Bioinformatik*, Johann Wolfgang Goethe-Universität Frankfurt am Main, Theoretische Informatik, WS 2000/2001, http://www.thi.informatik.uni-frankfurt.de/BIO/skript2.ps.

Ron Shamir: *Algorithms in Molecular Biology* Tel Aviv University, http://www.math.tau.ac.il/~rshamir/algmb.html; http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html.

Ron Shamir: *Analysis of Gene Expression Data, DNA Chips and Gene Networks*, Tel Aviv University, 2002; http://www.math.tau.ac.il/~rshamir/ge/02/ge02.html;

Martin Tompa: *Computational Biology*, CSE 527, University of Washington, Winter 2000. http://www.cs.washington.edu/education/courses/527/00wi/

A.3 Lehrbücher zu angrenzenden Themen

Teresa K. Attwood, David J. Parry-Smith; *Introduction to Bioinformatics*; Prentice Hall, 1999.

Maxime Crochemore, Wojciech Rytter: *Text Algorithms*; Oxford University Press: New York, Oxford, 1994.

Martin C. Golumbic: *Algorithmic Graph Theory and perfect Graphs*; Academic Press, 1980.

Benjamin Lewin: *Genes*; Oxford University Press, 2000.

Milton B. Ormerod: *Struktur und Eigenschaften chemischer Verbindungen*; Verlag Chemie, 1976.

Hooman H. Rashidi, Lukas K. Bühler: *Grundriss der Bioinformatik — Anwendungen in den Biowissenschaften und der Medizin*,

Klaus Simon: *Effiziente Algorithmen für perfekte Graphen*; Teubner, 1992.

Maxine Singer, Paul Berg: *Gene und Genome*; Spektrum Akademischer Verlag, 2000.

Lubert Stryer: *Biochemie*, Spektrum Akademischer Verlag, 4. Auflage, 1996.

A.4 Originalarbeiten

Kellogg S. Booth, George S. Lueker: Testing for the Consecutive Ones property, Interval Graphs, and Graph Planarity Using PS-Tree Algorithms; *Journal of Computer and System Science*, Vol.13, 335–379, 1976.

Ting Chen, Ming-Yang Kao: On the Informational Asymmetry Between Upper and Lower Bounds for Ultrametric Evolutionary Trees, *Proceedings of the 7th Annual European Symposium on Algorithms, ESA'99*, Lecture Notes in Computer Science 1643, 248–256, Springer-Verlag, 1999.

Richard Cole: Tight Bounds on the Complexity of the Boyer-Moore String Matching Algorithm; *SIAM Journal on Computing*, Vol. 23, No. 5, 1075–1091, 1994.
s.a. *Technical Report*, Department of Computer Science, Courant Institute for Mathematical Sciences, New York University, TR1990-512, June, 1990, http://csdocs.cs.nyu.edu/Dienst/UI/2.0/Describe/ncstrl.nyu_cs%2fTR1990-512

Martin Farach, Sampath Kannan, Tandy Warnow: A Robust Model for Finding Optimal Evolutionary Trees, *Algorithmica*, Vol. 13, 155–179, 1995.

Wen-Lian Hsu: PC-Trees vs. PQ-Trees; *Proceedings of the 7th Annual International Conference on Computing and Combinatorics, COCOON 2001*, Lecture Notes in Computer Science 2108, 207–217, Springer-Verlag, 2001.

Wen-Lian Hsu: A Simple Test for the Consecutive Ones Property; *Journal of Algorithms*, Vol.43, No.1, 1–16, 2002.

Haim Kaplan, Ron Shamir: Bounded Degree Interval Sandwich Problems; *Algorithmica*, Vol. 24, 96–104, 1999.

Edward M. McCreight: A Space-Economical Suffix Tree Construction Algorithm; *Journal of the ACM*, Vol. 23, 262–272,1976.

Moritz Maaß: *Suffix Trees and Their Applications*, Ausarbeitung von der Ferienakademie '99, Kurs 2, Bäume: Algorithmik und Kombinatorik, 1999. http://www14.in.tum.de/konferenzen/Ferienakademie99/

Esko Ukkonen: On-Line Construction of Suffix Tress, *Algorithmica*, Vol. 14, 149–260, 1995.

Index

Symbole

A

B

S

T