

Multiples Alignment in der Praxis : T-Coffee

okay

Anmerkungen siehe Email

Ausarbeitung von
Aylin Ören und Ramona Richter

Inhaltsverzeichnis

1. Einleitung

1.1 Multiples Sequenz-Alignment

2. ClustalW

3. T-Coffee

- 3.1 Primär Bibliothek
- 3.2 Erweiterte Bibliothek
- 3.3 Progressive Alignment Strategie
- 3.4 Komplexität
- 3.5 Weitere Tools

Seitenzahlen!

4. Quellen

1. Einleitung

Zur Berechnung multipler Sequenz-Alignments werden immer effizientere und exaktere Algorithmen entwickelt. Der am häufigsten genutzte Algorithmus ist hierbei ClustalW, welcher einige Fehlerkomponenten jedoch nicht vollständig abdeckt. ^{Quelle} 1998 wurde ClustalW in einem erweiterten Algorithmus mit eingebunden, T-Coffee. (Quelle) Dieser komplexe Algorithmus nutzt verschiedene Alignments zur Fehlervermeidung. In Kapitel 2 findet sich eine kurze Erläuterung zu ClustalW, dies dient als Einstieg. In Kapitel 3 folgt dann eine genaue Beschreibung des Algorithmus von T-Coffee.

Was ist im dem Kapitel enthalten?

1.1 Multiple-Sequenz-Alignment

Die Frage, die sich einleitend stellt, ist nun wozu diese Algorithmen benötigt werden. Um dies zu klären werden zunächst multiple Sequenz-Alignments erklärt. Das Multiple Sequenz Alignment ist eine Erweiterung des Alignments von zwei Sequenzen. Hierbei werden bisher nicht bekannte biologische Verwandtschaften aus Sequenzähnlichkeiten abgeleitet. Durch den Vergleich mehrerer Sequenzen (multiples Alignment) ist es nun möglich auch kurze, vorerst unbemerkte, Muster in Sequenzen zu finden und somit biologisch relevante Muster auch in stark ähnelnden Sequenzen zu identifizieren.

Multiple Alignments werden zum Beispiel genutzt um phylogenetische Bäume zu konstruieren, diese dienen dem Vergleich einer Sequenz mit Proteinfamilien. Im DNA-Bereich werden multiple Sequenz-Alignments genutzt, um Primer zu bestimmen oder DNA-Sequenzen zu assemblieren.

2. ClustalW

Das Prinzip von ClustalW beruht darauf, dass alle Sequenzen von einer Ursequenz ausgehen bzw. daran orientiert sind. Dementsprechend werden, ausgehend vom Alignment, dem Sequenzpaar mit dem höchsten Score alle weiteren Sequenzen des Alignments zugeordnet.

ClustalW ist ein progressiver Algorithmus und besteht aus drei wesentlichen Schritten, Berechnung des paarweises Alignments und dessen Score, Berechnung des Guide-Trees und Bildung des Gesamtalignments entlang des Guide-Trees [1]. Zur besseren Verständlichkeit werden folgende Sequenzen als begleitendes Beispiel fungieren:

Sequenz A: GARFIELDTHELASTFATCAT

Sequenz B: GARFIELDTHEFASTCAT

Sequenz C: GARFIELDTHEVERYFASTCAT

Sequenz D: THEFATCAT

Um die Ähnlichkeit der Sequenzen untereinander feststellen zu können, werden zuerst alle paarweisen Alignments gebildet, diese werden dann mittels einer Scorefunktion bewertet (siehe Abb.1). In unserem Beispiel werden Matches mittels Farben gekennzeichnet.

GARFIELDTHELASTFATCAT	
GARFIELDTHEFASTCAT - - -	A → B, Score 88
GARFIELDTHELASTFA - TCAT	
GARFIELDTHEVERYFASTCAT	A → C, Score 61
GARFIELDTHELASTFATCAT	
- - - - - THEFATCAT	A → D, Score 66
GARFIELDTHE - - - - FASTCAT	
GARFIELDTHEVERYFASTCAT	B → C, Score 94
GARFIELDTHEFASTCAT	
- - - - - THEFA - TCAT	B → D, Score 88
GARFIELDTHEVERYFASTCAT	
- - - - - THEFATCAT	C → D, Score 44

Abb.1: paarweise Alignments und dessen Scores [1]

Im nächsten Schritt wird aus den gewonnenen Werten, mittels Neighbor-Joining Verfahren, der Guide-Tree entwickelt (siehe Abb.2). Nun aligniert man die Blätter entlang des Guide-Trees, hierbei werden zuerst die Sequenzen berücksichtigt, die einander am nächsten sind, bzw. von einem gemeinsamen Zweig ausgehen. In unserem Beispiel ist zusehen, dass A und D zuerst aligniert werden, dann dazu B und zum Schluss C. (siehe Abb.2)

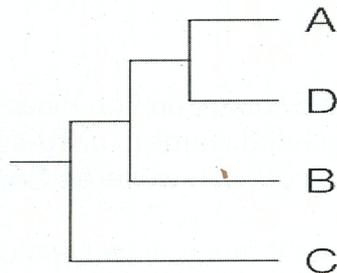


Abb.2: Guide-Tree [1]

Im letzten Schritt wird nun das Gesamtalignment gebildet. Hierzu werden die errechneten Ähnlichkeiten aus dem Guide-Tree herangezogen (siehe Abb.3). Dementsprechend werden zuerst A und D aligniert. Hierbei zeigen sich am Ende des Alignmentpaares Matches, FATCAT. Als nächstes wird B hinzugenommen. Da A und D nicht mehr verschiebbar sind, wird B, ungeachtet der entstehenden gaps, mit möglichst vielen Matches aligniert. Zum Schluss muss nun C noch hinzugefügt werden. Hierzu gilt nun wieder das weder A noch D und zudem auch B verschieden werden können. C wird somit an einen festen Sequenzblock aligniert, mit Hinsicht auf optimale Matches. Im Gesamtalignment fällt nun auf, dass die vorher hohe Anzahl an Matches, die man bei den paarweisen Alignments (siehe Abb.1) gesehen hat, nicht mehr vorhanden sind, obwohl die Beispiel Sequenzen sich untereinander sehr ähneln.

⊛ *Zitat nicht einfach nur nennen, sondern besser in den Kontext einbetten, zum Bsp. einleiten*

1. A → D

```
GARFIELDTHELASTFATCAT
-----THEFATCAT
```

2. AD → B

```
GARFIELDTHELASTFATCAT
-----THEFATCAT
GARFIELDTHEFASTCAT---
```

3. ABD → C

```
GARFIELDTHELASTFA-TCAT
GARFIELDTHEFASTCA-T---
-----THEFA-TCAT
GARFIELDTHEVERYFASTCAT
```

Abb. 3: Bildung des Gesamtalignments in 3 Schritten [1]

Das erhoffte Ergebnis (siehe Abb.4) zeigt lediglich wenig Übereinstimmung mit dem erhaltenen Ergebnis(siehe Abb.3). Die Anzahl der Matches ist deutlich höher und die Anordnung der Sequenzen sieht intuitiver aus. Die Frage die sich nun stellt ist, warum das Ergebnis so offensichtlich abweicht.

```
GARFIELDTHELASTFA-TCAT
GARFIELDTHE----FASTCAT
GARFIELDTHEVERYFASTCAT
-----THE----FA-TCAT
```

Abb.4: Gesamtalignmentergebnis durch die Berechnung mittels T-Coffee [1]

Diese Frage führt zu den Nachteilen von ClustalW, welche eine solche Fehlpaarung hervorrufen.

ClustalW nutzt lediglich globale Alignments, welche häufig zu „zerrissenen“ Alignments führen. Eine Korrektur Funktion ist zudem nicht vorgesehen, einmal gepaarte Alignments oder gesetzte Lücken (Gaps) können nicht mehr verschoben werden, „Once a gap, always a gap“ [5] somit kann eine **A**nfänglich suboptimale Fehlpaarung das Gesamtalignment negativ verschieben.

3. T-Coffee

T-Coffee, **A**bkürzung für **T**ree-based **C**onsistency **O**bjective **F**unction **F**or alignm**E**nt **E**valuation, ist ein multiples Sequenz-Alignment Programm. Es wurde 1998 von Cédric Notredame, Liisa Holm und Desmond G. Higging entwickelt, [2].

Prinzipiell gibt es lediglich zwei Verbesserungen gegenüber ClustalW, jedoch sind diese ausschlaggebend. **⊛**

Bei ClustalW fehlten lokale Informationen, um die Anzahl der Matches zu erhöhen, zudem fehlte eine Art „Zukunftswissen“, um Fehlpaarungen zu vermeiden. Aus diesem Grund nutzt T-Coffe nicht nur globale, sondern auch **lokale** lokale Alignments, um somit die Anzahl der Matches zu erhöhen. Zudem werden erweiterte Bibliotheken genutzt um ein „Zukunftswissen“ zu erlangen.

Der grobe Aufbau von T-Coffe (siehe Abb.5) lässt sich wie folgt erklären: Zuerst werden zwei primäre Bibliotheken gebildet (Primary Library), diese werden mittels Gewichtung zu einer gemeinsamen zusammengefasst. Diese wird zur erweiterten Bibliothek (Extended Library) erweitert, wobei hier der Vergleich der Sequenzpaare mit weiteren Sequenzen

erfolgt. Zum Schluss wird ein progressives Alignment gebildet, welches dann zum Gesamtalignment führt.

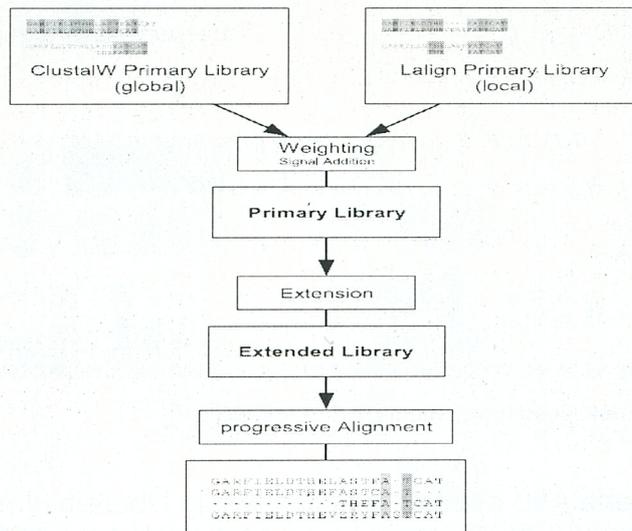


Abb. 5: Struktur von T-Coffee im Überblick [1]

3.1 Primär Bibliothek

Die Primär Bibliothek unterteilt sich zunächst in die ClustalW Primär Bibliothek, in der globale Alignments genutzt werden, und der Lalign Primär Bibliothek, in der die zehn besten lokalen Alignments errechnet werden. Um diese Bibliotheken nun zu einer primären Bibliothek zu vereinen, werden die gewonnenen Information gewichtet, dies geschieht durch die Berechnung der prozentualen Identität der Sequenzpaare. Diese wird durch die Anzahl der Buchstaben der hinzugefügten Sequenz, geteilt durch die Anzahl der erhaltenen Matches zwischen den gepaarten Sequenzen, errechnet. Dabei werden nur alignierte Buchstaben und keine gaps berücksichtigt. Daraus resultiert nun nicht nur eine Gewichtung der gesamten Sequenzpaaren, sondern eine der einzelnen Buchstabenpaare. Jeder Buchstabenpaarung wird somit ein Gewicht, berechnet aus allen Sequenzpaaren, zugewiesen, welche sich an den Matches orientiert. Je mehr Matches, desto höher ist die Gewichtung. Daraus bildet sich dann die Primär Bibliothek.

Im Beispiel (siehe Abb.6) sind nun die Sequenz A und D durch ein globales Alignment und ein lokales Alignment dargestellt. Beim globalen (oben) ist eine Gewichtung von 66 zu sehen, also 9 Buchstaben, wovon 6 Matches sind. Beim lokalen (unten) ist sogar eine Gewichtung von 100 zu sehen, also 9 Buchstaben, wovon 9 Matches sind. Somit hätten die schwarz umkreisten T's ein Gewicht von 100, die rot umkreisten jedoch von 166, da wir in allen Sequenzpaaren ein Match haben.

Aus diesen Informationen bildet sich dann die primär Bibliothek.

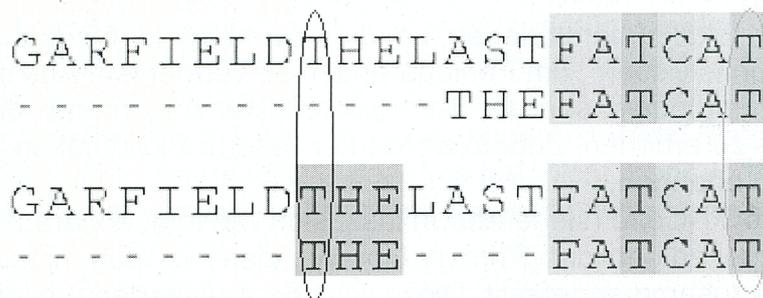


Abb. 6: globales (oben) und lokales (unten) Alignment der Beispielsequenzen A und D [1]

* hier sagt man auch, dass man nur alle G's miteinander
ander vergleichen wollen, in einer späteren Be-
schreibung wird das aber nicht mehr deutlich

3.2 Erweiterte Bibliothek

Primär

Der zweite wichtige Schritt ist die Erweiterung der Gesamtbibliothek, welche durch die Benutzung eines heuristischen Algorithmus genannt „library extension“, erreicht wird. Die Grundidee besteht darin die Informationen derart zu kombinieren, dass das Gesamtgewicht einige Informationen, die in der gesamten Bibliothek enthalten sind, reflektiert, und zwar für jedes zugewiesene Paar. Mit Hilfe der erweiterten Bibliothek soll herausgefunden werden, wie die Sequenzen als Paar untereinander zusammenpassen und wie gut das Sequenzenpaar jeweils zu einer der übrigen Sequenzen passt. Dazu werden Triplets betrachtet. In dem folgenden Beispiel sollen drei Sequenzen (A, B, C, D) herangezogen werden.

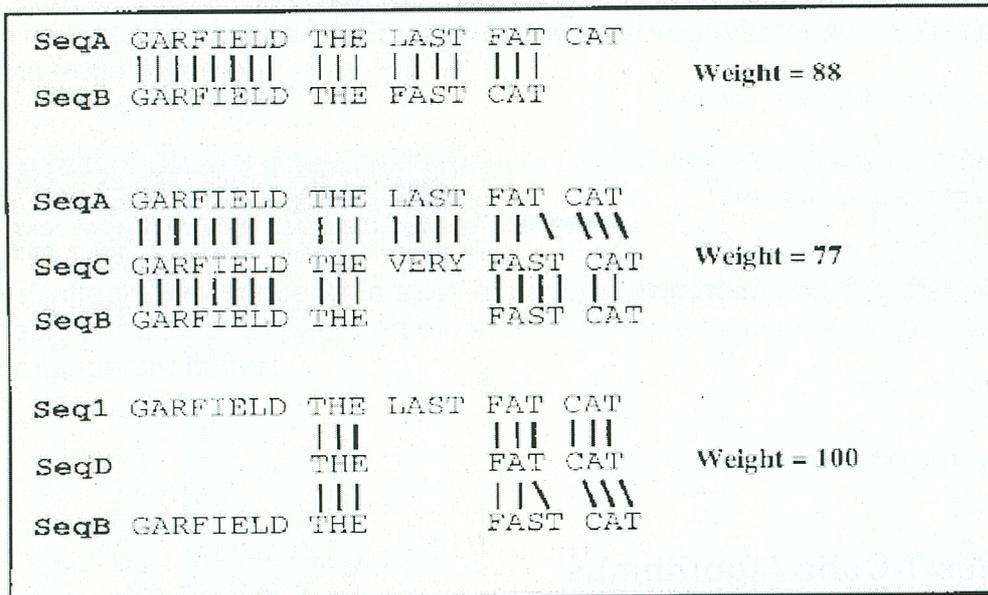


Abb.7: Library extension für ein Sequenzenpaar, 3 mögliche Alignments von den Sequenzen A und B werden gezeigt. [2]

Was bedeutet das?

In der Abbildung 7 ist der Prozess zur Erweiterung der Basisbibliothek zu sehen. In diesem Beispiel wird das Sequenzenpaar A – B aus der primären Bibliothek betrachtet. Es ist zu beachten, dass es nur der Vergleich A(G) und B(G) beschrieben wird. Diesem Sequenzenpaar wurde schon eine Gewichtung zugeschrieben, welches einer prozentualen Identität von 88 entspricht. Als nächstes wird ein Triplet gebildet und es wird die Ähnlichkeiten der Sequenzen A und B über C betrachtet. Die Verbindungslinien zwischen den Sequenzen lassen sich als Bindungskräfte interpretieren. Die Sequenzen A und C bzw. C und A werden miteinander aligniert. Somit steht fest, dass es ein Alignment von den Sequenzen A und B über C gibt. Wir kennen das Gewicht von dem Sequenzenpaar A – C aus der Primärbibliothek, welches 77 ist wie auch das Gewicht von dem Sequenzenpaar C – B, welches 100 ist. Man nimmt das kleinste Gewicht 77, welches ähnlicher zum Minimum ist. Dieser neue Wert wird nun in der erweiterten Bibliothek zu dem Wert aus der Primärbibliothek hinzugefügt bzw. addiert. Somit ergibt sich ein Gesamtgewicht von 165 für das Sequenzenpaar A – B. Nun muss aber noch die Ähnlichkeit dieses Sequenzenpaares auch über D betrachtet werden. Dies erfolgt nach dem gleichen Prinzip.

Bei der Betrachtung des Triplets A, D, B fällt auf, dass die Sequenzen A und B über D wenig Ähnlichkeit haben, damit die Sequenz D auch wenig Informationen über das

Sequenzenpaar hergibt. Auch hier ist zu beachten, dass dies für den Vergleich mit dem „G“ von dem Wort „GARFIELD“ gilt. Dies bedeutet also, dass es für andere AS-Paare Informationen geben kann. Damit verbunden spielt das Gewicht auch keine weitere Rolle für dieses Paar. Dieser Prozess wird für alle übrigen Sequenzpaare ebenfalls durchgeführt (A-C, A-D, B-C, B-D).

3.3 Progressive Alignment Strategie

Der letzte Schritt ist die progressive Alignment Strategie. Es soll nun das multiple Alignment gebildet werden. Dazu bilden die paarweisen Alignments eine Distanzmatrix zwischen allen Sequenzen. Daraus wird dann der Guide-Tree erstellt, wobei eine Methode namens Neighbor-Joining genutzt wird. Beim Erstellen des multiplen Alignments wird auf die Gewichte aus der primären Bibliothek zugegriffen, danach wird schrittweise aligniert. Dieser Vorgang und das Ergebnis ist der Abbildung 8 zu entnehmen.

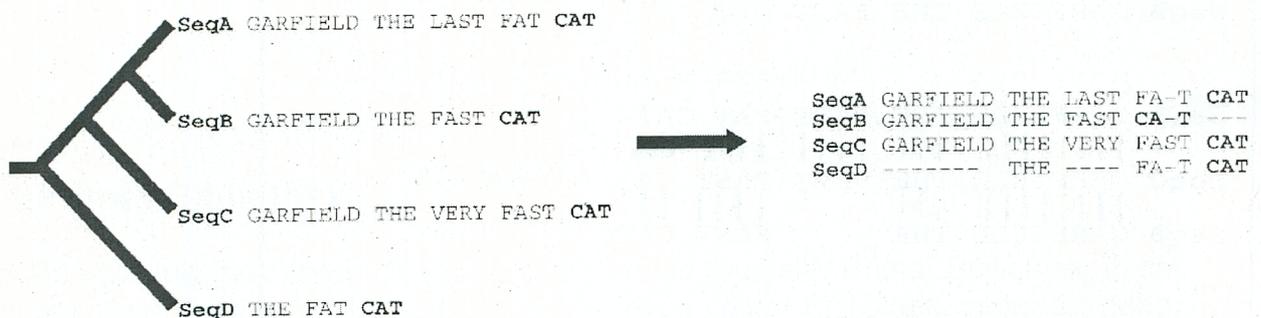


Abb. 8: Erstellung des Guide-Tree und multiplen Alignments. [2]

3.3 Komplexität des T-Coffe Algorithmus

Die Komplexität des gesamten Ablaufs beträgt:

$$O(N^2L^2) + O(N^3L) + O(N^3) + O(NL^2).$$

Dabei kann man die einzelnen Teile verschiedenen Vorgängen zuweisen. Zu dem Laufzeitverhalten allgemein ist zu sagen, dass N für die Anzahl der Sequenzen und L für die durchschnittliche Länge steht. Die Laufzeit zur Berechnung der paarweisen Bibliotheken beträgt $O(N^2L^2)$. N^2 ist die Bildung aller Paare und für jedes Paar werden L Positionen verglichen (L^2). Die Extension berechnet eine Laufzeit von $O(N^3L)$. Es handelt sich um ein Triplet, daher N^3 und L , da einmal über die Sequenz verglichen wird. Die Komplexität zur Berechnung des Guide-Tree ist $O(N^3L)$. Der letzte Teil der Gesamtkomplexität $O(NL^2)$ kommt durch die Bildung des progressiven Alignments zustande. In diesem Fall hat man N Sequenzen, woraus dann durch Alignierung das multiple Alignment gebildet wird, daher L^2 . Es wurde analysiert, dass oft $L > N$ ist, somit fallen die Komplexitäten $O(N^3L) + O(N^3)$ automatisch raus und man erhält die gleiche Laufzeit wie die von ClustalW. Dadurch, dass T-Coffe mit zwei Primärbibliotheken arbeitet, zum einen mit Lalign und zum anderen mit ClustalW, ist der Algorithmus von T-Coffe zweimal so langsam wie der Algorithmus von ClustalW.

3.4 Weitere Tools

Es werden noch viele verschiedene Tools von den T-Coffee Entwicklern angeboten. Einige davon werden nun erklärt.

M-Coffee: M-Coffee ist ein spezieller Modus von T-Coffee. Er ermöglicht es, dass die Ausgabe verschiedener Multiple-Sequenz-Alignment-Pakete wie z.B. Muscle, ClustalW, Mafft, ProbCons, etc. kombiniert werden können. Das Programm markiert die Regionen im Alignment, in denen die verschiedenen Einzelprogramme übereinstimmen. Regionen mit einer hohen Übereinstimmung sind meist besser aligniert.

3D-Coffee: Dieser spezieller Modus von T-Coffee ermöglicht die Verknüpfung von Sequenz und Struktur in einem Alignment. Die strukturbasierten Alignments können mit Hilfe von gebräuchlichen Struktur-Alignment-Programmen wie z.B. Tmalign, Mustang und sap erstellt werden.

Expresso: Dies ist eine Erweiterung von 3D-Coffee. Hier werden strukturelle Templates automatisch mit Blast identifiziert.

R-Coffee: Dies ist ebenfalls auch ein spezieller Modus von T-Coffee, welcher für RNAs designed wurde. Er generiert ein multiples Sequenzalignment unter Benutzung von Sekundärstrukturen.

4. Quellen

- [1] www2.informatik.hu-berlin.de/Forschung_Lehre/.../MSA_TCoffee.pdf
(24.06.2013)
- [2] Cedric Notredame, Desmond G. Higgins, Jaap Heringa: *A Novel Method for Fast and Accurate Multiple Sequence Alignment*, J. Mol. Biol. (2000) 302, 205-217h
- [3] http://www.bioinf.uni-freiburg.de/Lehre/Theses/DA_Joachim_Krempel.pdf
(24.06.2013)
- [4] http://abi.inf.uni-tuebingen.de/Teaching/Old/SS10/BILW/folien-1/10_MultiplesAlignment_II_PSIBLAST_3up.pdf (24.06.2013)
- [5] Feng, D.-F. & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25, 351-360