Georg-August-Universität
Göttingen
Zentrum für Informatik

ISSN 1612-6793
Nummer ZFI-BM-2005-28

Bachelorarbeit

im Studiengang „Angewandte Informatik“

Paarweise und multiple Alignments mit TBLASTX und DIALIGN

Dirk Pöhler

am Institut für
Mikrobiologie und Genetik
Abteilung für Bioinformatik

Bachelor- und Masterarbeiten
des Zentrums für Informatik
an der Georg-August-Universität Göttingen

26. September 2005

Georg-August-Universität Göttingen
Zentrum für Informatik

Lotzestraße 16-18
37083 Göttingen
Germany

Tel.	+49 (5 51) 39-1 44 14
Fax	+49 (5 51) 39-1 44 15
Email	office@informatik.uni-goettingen.de
WWW	www.informatik.uni-goettingen.de

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Göttingen, den 26. September 2005

Bachelorarbeit

Paarweise und multiple Alignments mit TBLASTX und DIALIGN

Dirk Pöhler

26. September 2005

Betreut durch Prof. Dr. Morgenstern
Institut für
Mikrobiologie und Genetik
Abteilung für Bioinformatik
Georg-August-Universität Göttingen

Inhaltsverzeichnis

1. Einleitung

Eine Aufgabe der Bioinformatik ist das Auffinden von Homologien zwischen unterschiedlichen Organismen mit Hilfe von sogenannten Alignments. Ein Alignment dient dem Vergleich zweier oder mehrerer DNA- oder Proteinsequenzen, um die funktionale oder evolutionäre Ähnlichkeit der Sequenzen untereinander zu untersuchen. (siehe Abb. 1) Dazu werden Abschnitte der jeweiligen Sequenzen, die eine gewisse Sequenzähnlichkeit zueinander haben, im Alignment übereinander

Gegeben:
S_1= AGGTCCGGT
S_2= GGTACG

Zwei mögliche Sequenzalignments von vielen:

Alignment A

S_1	A	G	G	T	C	C	G	G	T
S_2	-	G	G	T	-	A	C	G	-

Alignment B

S_1	-	-	-	A	G	G	T	C	C	G	G	T
S_2	G	G	T	A	C	G	-	-	-	-	-	-

Abb. 1 Beispiel für DNA-Alignments

In der Abbildung werden von den vielen möglichen Sequenzalignments, die aus den zwei DNA - Sequenzen S_1=aggtccggt und S_2=ggtagg errechnet werden können, nur zwei exemplarisch dargestellt. In diesem Beispiel wird ein sogenanntes Match durch eine Spalte mit schwarzen Buchstaben im Alignment dargestellt. Ein Mismatch wird durch eine Spalte mit blauen Buchstaben dargestellt. „-“ bezeichnet eine Lücke (Gap).

ausgerichtet. Mit Hilfe von Softwareprogrammen werden Alignments von Sequenzen berechnet. Die Untersuchung von funktionalen Stellen in Genomsequenzen verschiedener Spezies basiert auf Genomalignments. Allerdings ist der Vergleich von biologischen Sequenzen sehr rechenaufwendig und benötigt viel Hauptspeicher. Effiziente Algorithmen und Methoden, um biologisch sinnvolle Alignments zu erstellen, werden nach wie vor entwickelt.

Das Alignmentprogramm, um das es in der vorliegenden Arbeit geht, heißt *DIALIGN* [1]. Hierbei handelt es sich um ein Multiples Alignment Programm, das verwendet wird, um von mindestens zwei DNA- oder Protein- Sequenzen Alignments zu bilden. Die ursprüngliche Version von *DIALIGN* ist nicht dafür implementiert worden, sehr lange Sequenzen miteinander zu alignen. Dadurch ist im Hinblick auf die Laufzeit keine effiziente Berechnung eines Alignments aus Sequenzen genomischer

Größe möglich. Mittels einer speziellen Option können *DIALIGN* sogenannte Anker übergeben werden, wodurch das Programm dazu „gezwungen“ wird, bestimmte Regionen gemäß den Ankern zu alignen. Die Laufzeit von *DIALIGN* kann durch diese Option verbessert werden, da der Alignmentsuchraum reduziert wird. Um *DIALIGN* die Anker bereit zu stellen, kann z.B. das schnelle lokale Alignmentprogramm namens *CHAOS* [2] verwendet werden.

Ein anderes lokales Alignmentprogramm ist *TBLASTX* [3], das in der vorliegenden Arbeit als Alternative zu *CHAOS* Verwendung findet. Der Sequenzvergleich bei *TBLASTX* wird nicht wie bei *CHAOS* auf Nukleotidebene, sondern auf Proteinebene durchgeführt. Dazu werden *TBLASTX* DNA-Sequenzen übergeben, die alle jeweils in die sechs möglichen Leseraster (Frames) übersetzt werden. *TBLASTX* liegt ein sehr effizienter Algorithmus zugrunde, der es zu einem schnellen Tool macht. Zur Laufzeitoptimierung wird in dieser Arbeit *DIALIGN* eine Menge von Ankern übergeben, die aus dem von *TBLASTX* berechneten Output erstellt werden. Ein Ziel der vorliegenden Arbeit ist die Implementierung einer Schnittstelle zwischen *TBLASTX* und *DIALIGN*. Ein weiteres Ziel ist die Bestimmung eines allgemein gültigen Schwellwertes für die von *TBASTX* berechneten Anker, um für Kombinationen von Sequenzen verschiedener Spezies biologisch hochwertige Anker zu erhalten. Dieser Schwellwert wurde mittels ausführlichen Tests mit annotierten DNA-Sequenzen von der *ENSEMBL*-Datenbank [4] ermittelt. Dabei sollten die an *DIALIGN* übergebenen Anker möglichst nicht die biologische Qualität des von *DIALIGN* zu erzeugenden Sequenzalignments reduzieren, jedoch die Laufzeit optimieren.

2. Biologische Grundlagen

2.1. DNA

Die Desoxyribonukleinsäure (DNS, engl. *deoxyribonucleic acid, DNA*) ist ein Makromolekül, das in der Vererbung als Träger der Information dient. Makromoleküle sind große Moleküle, die aus vielen unterschiedlichen oder gleichen Bausteinen bestehen.

Ein DNA-Makromolekül besteht aus zwei komplementären antiparallelen DNA-Strängen, die eine Doppel-Helix bilden. DNA-Stränge sind Polymere aus Desoxyribonukleotiden (sogenannte Nukleotide), den Grundbausteinen der DNA. Nukleotide bestehen aus einem Phosphat-Rest, einem Zucker-Rest und einer von vier unterschiedlichen Basen. Die Nukleotide unterscheiden sich lediglich durch ihre jeweilige Base: Adenin, Cytosin, Guanin oder Thymin. Die genetische Information ist in der Sequenzabfolge der Basen kodiert.

Mutationen stellen eine Veränderung in der Nukleotidsequenz zu ihrer ursprünglichen Sequenz dar. Diese Veränderungen sind vererbbar. Mutationen wandeln die genetische Information ab. Solche Mutationen bilden den Ausgangspunkt für die Veränderung von Lebewesen im Rahmen der Evolution. Bereiche der DNA, die eine essentielle biologische Funktion haben, werden im Laufe der Evolution in der Regel stärker konserviert als andere Bereiche der DNA.

2.2. Proteine

Proteine sind Makromoleküle und gehören zu den Grundbausteinen aller Zellen. Sie bestehen aus einzelnen Bausteinen, 20 verschiedenen Aminosäuren. Das bedeutet Proteine sind somit die Verkettung von Aminosäuren.

Der Vorgang, bei dem die "Sprache der Gene" in die "Sprache der Proteine" übersetzt wird, nennt sich Translation. Ein Protein wird dabei aus einer in der Abfolge von Nukleotiden codierten Information in der Zelle hergestellt. Bei der Translation codieren drei Nukleotide (ein Triplett oder Codon) eine Aminosäure. Die genetische Sprache besitzt vier Buchstaben: Adenin, Guanin, Cytosin und Uracil. Die Proteinsprache besteht aus 20 Buchstaben, den 20 Aminosäuren. Bei der Proteinsprache sind die Verknüpfungsmöglichkeiten weitaus größer.

Ein Gen ist ein Abschnitt auf der DNA, der die Grundinformationen zur Herstellung einer RNA (engl. *ribonucleic acid*) enthält. Gene kodieren unter anderem mRNA (Boten-RNA), aus denen Proteine translatiert werden. Bei Eukaryoten werden jedoch nicht alle Bereiche der Nukleotidsequenz in

Proteine übersetzt, sondern nur die sogenannten Exons, die innerhalb von Genen liegen (siehe Abb. 2). Exons bezeichnen die kodierenden Bereiche, die genetische Information enthalten. Diese manifestiert sich in Proteinen. Nicht codierende Bereiche in Genen werden Introns und außerhalb von Genen werden sie intergenische Regionen genannt. Proteine haben unterschiedliche und sehr wichtige Bedeutungen für den Organismus. Sie regeln als Enzyme und Hormone den gesamten Stoffwechsel. Dabei beschleunigen und ermöglichen sie chemische Reaktionen oder steuern die Vorgänge im Körper. Die wichtigsten Funktionen von Proteinen sind der Aufbau der Zelle und die Reparatur des Gewebes.

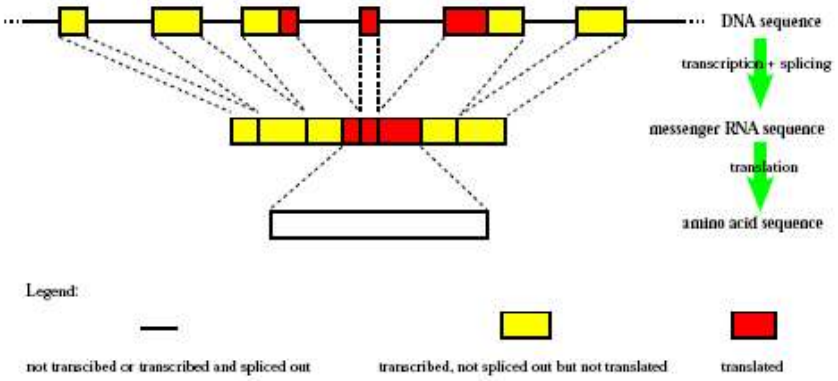

Abb. 2 Vereinfachte Genexpression bei Eukaryoten

Nur die roten Bereiche werden als Exons bezeichnet, da nur sie in Proteine übersetzt werden. [5]

2.3. Paarweise und multiple Alignments

Ein Alignment dient dem Vergleich zweier oder mehrerer DNA- oder Proteinsequenzen. Es zeigt die Homologien der beteiligten Sequenzen zueinander auf. Ein Alignment mit zwei beteiligten Sequenzen wird paarweises Alignment und mit mehr als zwei Sequenzen multiples Alignment genannt.

Elemente bezeichnen im Folgenden entweder Basen oder Aminosäuren, je nachdem ob Nukleotid- oder Proteinsequenzen vorliegen. Beim paarweisen Alignment ordnet man die Elemente einer Sequenz Elementen einer anderer Sequenz so zu, dass die Reihenfolge der jeweiligen Elemente der Sequenzen erhalten bleibt und jedes Element der einen Sequenz einem Element oder einer „Lücke“, einem sogenannten Gap, der anderen Sequenz zugeordnet ist. Dabei sollten die einander zugeordneten Elemente identisch oder möglichst ähnlich sein. Zudem darf keine Spalte im Alignment nur aus Gaps bestehen. Bei der Berechnung eines multiplen Alignments verfährt man mit mehr als zwei Sequenzen analog zur Berechnung eines paarweisen Alignments.

Zur Erstellung von Alignments werden Alignmentprogramme verwendet, denen die zu alignierenden Sequenzen übergeben werden. Bei der Ausführung von unterschiedlichen Alignmentprogrammen mit denselben Sequenzen erhält man eine große Menge an unterschiedlichen Alignments. Dies liegt daran, dass Programmen unterschiedliche Algorithmen zur Berechnung von Alignments zu Grunde liegen. Die verschiedenen Algorithmen der Programme basieren jeweils auf unterschiedlichen Bewertungsgrundlagen.

Es gibt lokale und globale Sequenzalignment-Werkzeuge. Ein globales Alignment, stellt ein Alignment dar, das alle am Alignment beteiligten Sequenzen komplett enthält. In der vorliegenden Arbeit bezeichnet ein lokales Alignment ein paarweises lokales Alignment. In diesem Sinne ist ein lokales Alignment, ein Alignment zwischen einem Sequenzabschnitt der einen Sequenz und einem Sequenzabschnitt einer anderen Sequenz.

2.4. Homologe Gene

Ein Gen, das für ein Protein kodiert, enthält die Beschreibung der Aminosäuresequenz eines Proteins. Gene sind Träger von Erbinformationen, die an Nachkommen weitergegeben werden können. Es wird zwischen paralogen und orthologen Genen unterschieden. Gene werden als paralog zueinander bezeichnet, wenn sie sich in demselben Organismus befinden. Orthologe Gene sind Gene in verschiedenen Spezies, die vom selben Vorläufergen im letzten gemeinsamen Vorfahren der verglichenen Spezies abstammen.

Haben zwei Gene denselben Vorfahren, so nennt man sie homolog, dabei spielt es keine Rolle ob sie ortholog oder paralog sind.

3. Alignment Programme für lange DNA-Sequenzen

3.1. *AVID*

AVID [6] ist ein paarweises globales Alignmentprogramm, das ganze Genomsequenzen miteinander alignt. Als Input werden *AVID* zwei DNA-Sequenzen im *fasta*-Format (Siehe Anhang Abb. i) übergeben. Der Output ist ein globales Alignment im *avx*-Format (Siehe Anhang Abb. ii).

Die *AVID* – Methode:

Zuerst werden die beiden übergebenen Sequenzen zu einer Sequenz verbunden und anschließend maximale Repeats in diesen Sequenzen mittels Suffixbäumen gesucht. Diese maximalen Repeats werden als Ankerpunkte verwendet. Anschließend werden diese Ankerpunkte auf Überschneidungen untereinander (Konsistenzprüfung) überprüft. Nach der Konsistenzprüfung wird entschieden, ob genügend Anker vorliegen, um 50 % der Länge des Alignments abzudecken. Falls genügend Anker vorliegen, werden die Sequenzen zwischen den Ankerpunkten gesplittet und rekursiv neue Ankerpunkte mittels Suffixbäume gesucht. Andernfalls wird mittels des Needleman–Wunsch Algorithmus [7] ein optimales Alignment gebildet. Ist der zu alignierende Bereich zu groß für den Needleman-Wunsch Algorithmus wird ein triviales Alignment gebildet. Bei der Bildung des trivialen Alignments werden einfach Nullen an die zu alignierenden Stellen geschrieben.

AVID ist schnell, speichereffizient und kann mit großen genomischen Regionen mit mehr als einer Milliarde Basenpaaren umgehen. [6]

3.2. *LAGAN*

LAGAN [8] ist ein Alignmentprogramm zur Bestimmung des globalen Alignments für große DNA-Sequenzen. Als Input werden *LAGAN* zwei DNA-Sequenzen im *fasta*-Format (Siehe Anhang Abb. i) übergeben. Der Output besteht aus einem globalen Alignment im *fasta*-Format (Siehe Anhang Abb. iv).

Der *LAGAN* – Algorithmus, der aus drei Schritten besteht wird anhand von Abb. 3 erläutert.

Schritt 1. Berechnung der lokalen Alignments mit *CHAOS* (siehe Kapitel 4.3) zwischen den zwei DNA-Sequenzen (Siehe Abb. 3 B). Jedem lokalen Alignment wird ein Gewicht zugewiesen.

Schritt 2. *LAGAN* berechnet eine geordnete Teilmenge von lokalen Alignments mit maximalem Gewicht und fügt sie zu einer ungefähren globalen Karte zusammen, wobei rekursiv maximale Ketten von lokalen Alignments gebildet werden (Siehe Abb. 3 C).

Schritt3. Berechnung des globalen Alignments. Dafür werden sogenannte Boxen berechnet, die um die

maximalen lokalen Alignments liegen. Mittels dynamischer Programmierung wird das optimale Alignment innerhalb der Boxen gebildet (Siehe Abb. 3 D). Daraus ergibt sich das globale Alignment (Siehe Abb. 3 A).

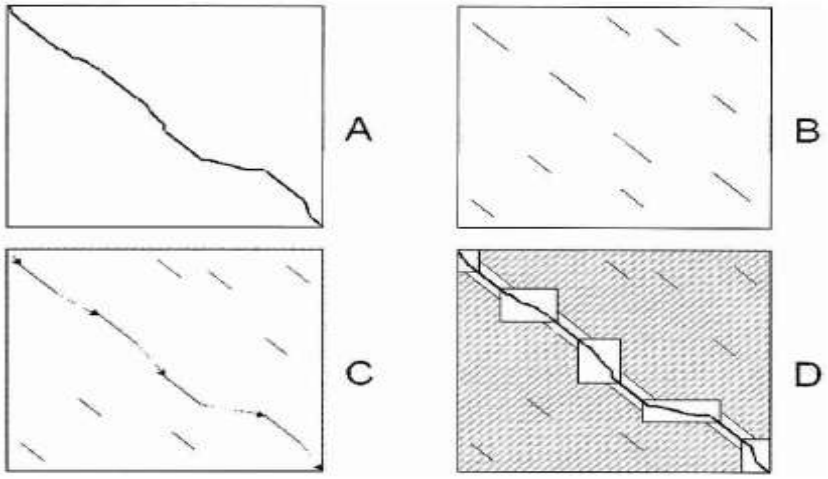

Abb. 3Der LAGAN-Algorithmus

(A)Ein globales Alignment zwischen zwei Sequenzen kann dargestellt werden als der Pfad von der oberen linken bis zur unteren rechten Ecke der Alignmentmatrix. (B)Zuerst sucht LAGAN lokale Alignments zwischen zwei Sequenzen. (C) LAGAN berechnet eine geordnete Teilmenge von lokalen Alignments mit maximalem Gewicht. (D)LAGAN beschränkt den Suchraum für ein optimales Alignment. [8]

Weitere Erläuterung siehe Text oberhalb der Abbildung.

4. *CHAOS - DIALIGN*

4.1. *DIALIGN*

4.1.1. DIALIGN Einleitung

DIALIGN [1] ist ein weit verbreitetes Softwareprogramm für multiple DNA- und Proteinsequenz-Alignments. Das Programm kombiniert lokale und globale Alignmenteigenschaften, wodurch es Sequenzen korrekt alignen kann, die mit den traditionellen Methoden nicht korrekt zu alignieren sind. Damit sind Familien von Sequenzen gemeint, bei denen lokale Ähnlichkeiten räumlich durch unverwandte Sequenzen getrennt sind. Die durch *DIALIGN* berechneten Alignments sind häufig biologisch aussagekräftige Ergebnisse.

DIALIGN basiert auf sogenannten Fragmenten, auf deren Grundlage das Outputalignment berechnet wird. Fragmente sind lokale paarweise Sequenzalignments, die keine Gaps beinhalten (siehe Abb.4). Solche Fragmente erscheinen in der sogenannten dot-Matrix als Diagonale. Der Name *DIALIGN* steht für das „Diagonalen-Alignment“ [9]. Eine dot-Matrix Analysis ist in erster Linie eine Methode zum Vergleich zweier Sequenzen in Hinblick auf ein mögliches Alignment zweier Zeichen zwischen den beiden Sequenzen.

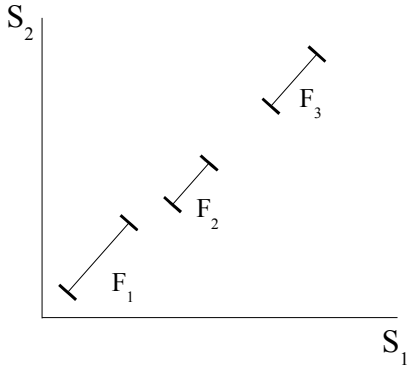

Abb.4 DIALIGN Fragmente

In der Abbildung werden Fragmente F1, F2 und F3 zu den Sequenzen S_1 und S_2 dargestellt.

4.1.2. DIALIGN-Algorithmus

Die in der vorliegenden Arbeit verwendete Version von *DIALIGN,* ist *DIALIGN2.2*. Diese Version basiert auf dem *DIALIGN2*-Algorithmus. Voraussetzung für den *DIALIGN2*-Algorithmus ist eine Score-Matrix. Bei der in dieser Arbeit verwendeten Version handelt es sich um eine *BLOSUM*-Matrix [10]. Diese Matrix listet einen Score s_{ij} für die Alignierung von jedem Aminosäurepaar i und j auf.
Die von *DIALGN* berechneten Outputalignments haben einen Score, der im Folgenden *DIALIGN*-Alignment-Score genannt wird. Der *DIALIGN*-Alignment-Score bezeichnet die Summe der Gewichte aller in dem von *DIALIGN* berechneten Outputalignment enthalten Fragmente.

Der *DIALIGN* 2 - Algorithmus wird nun beschrieben.
Der Algorithmus läuft in vier Phasen [1] ab:

1. Paarweise Alignments bilden
2. Overlap-Gewichte bestimmen
3. Fragmente nach ihrem Overlap-Gewicht sortieren
4. Konsistenzprüfung

1. Paarweise Alignments bilden

In diesem Algorithmusabschnitt werden optimale paarweise Alignments (Fragmente) bestimmt. Durch eine Gewichtungsfunktion für die Fragmente werden signifikante lokale Gemeinsamkeiten in den zu alignierenden Sequenzen gefunden. *P'(l,m)* ist die Wahrscheinlichkeit ein Fragment der Länge *l* mit mindestens *m* Matches zu finden, innerhalb einer Vergleichsmatrix von zwei zufälligen Sequenzen mit derselben Länge, wie die Originalsequenz. Das Gewicht *w(F)* des Fragmentes *F* wird durch

$$w(F) = -\ln(P'(l,m)) \quad \text{beschrieben,}$$

wobei die Werte von P', die größer als 10^{-5} sind, durch Experimente ermittelt wurden. Für kleinere Werte gilt die Approximation

$$P'(l,m) \approx l_1 * l_2 * P(l,m) \quad .$$

Dabei ist *P(l,m)* die Wahrscheinlichkeit, dass ein gegebenes Fragment der Länge *l* mindestens *m* Matche enthält [10]. l_1 und l_2 sind die Längen der Sequenzen.

2. Overlap-Gewichte bestimmen

Ein Overlap tritt auf, wenn zwischen mehr als zwei Sequenzen Fragmente gebildet werden, die einen Sequenzabschnitt gemeinsam involvieren (siehe Abb. 5) [9]. Ein solcher von zwei Fragmenten *F1* und

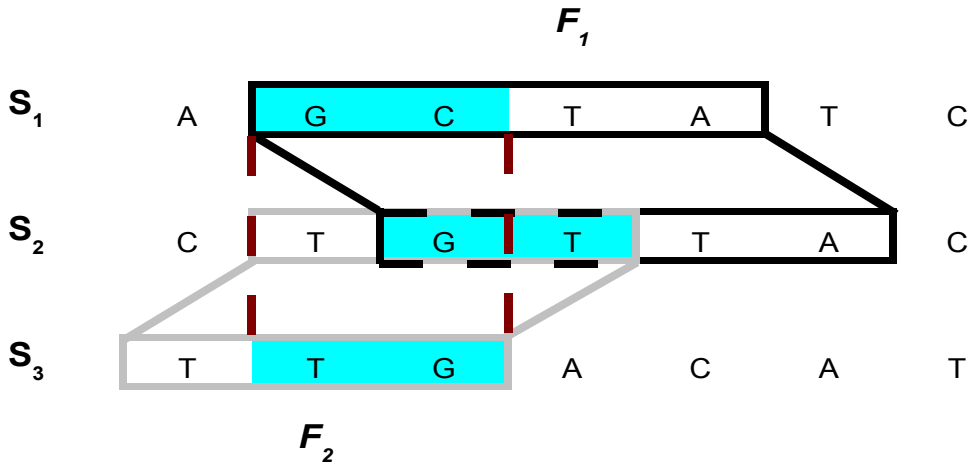

Abb. 5 Overlap von Fragmenten

Die Abbildung zeigt drei DNA-Sequenzen S_1 = AGCTATC, S_2 = CTGTTAC und S_3 = TTGACAT. In den Sequenzen werden die Fragmente F_1 (schwarz) und F_2 (grau) dargestellt Das Fragment F_1 besteht aus den Sequenzabschnitten „GCTA" der Sequenz S_1 und „GTTA" aus S_2. Das Fragment F_2 besteht aus den Sequenzabschnitten „TGT" der Sequenz S_2 und „TTG" der Sequenz S_3. Beide Fragmente involvieren den Sequenzabschnitt „GT" von S_2. Der von den Fragmenten F_1 und F_2 involvierte Sequenzabschnitt impliziert ein Fragment F_n (rot gestrichelt) zwischen den Sequenzabschnitten „GC" von S_1 und „TG" von S_3.

F_2 involvierter Sequenzabschnitt impliziert ein Fragment F_n. Das Gewicht $w(F_n)$ eines Fragments F_n entspricht dem Gewicht $\tilde{w}(F_1, F_2)$ der Fragmente *F1* und *F2* folgendermaßen:

$$\tilde{w}(F_1, F_2) := w(F_n)$$

Sind die Fragmente F_1 und F_2 identisch oder haben keinen Overlap, gilt:

$$\tilde{w}(F_1, F_2) := 0$$

Für ein beliebiges Fragment *F* wird das Overlap-Gewicht *w'(F)* folgendermaßen berechnet:

$$w'(F) = w(F) + \sum_{E \in K} \tilde{w}(F, E) \quad ,$$

wobei *K* die Menge aller Fragmente bezeichnet. Je größer der Overlap ist, desto signifikanter ist das Fragment.

3. Fragmente nach ihrem Overlap-Gewicht sortieren

Die Fragmente werden nun absteigend nach ihrem Overlap-Gewicht sortiert und nach einer Greedy-Strategie in ein multiples Alignment überführt. Das Fragment mit dem höchsten Overlap-Gewicht wird das erste Fragment.

4. Konsistenzprüfung

Nach bestandener Konsistenzprüfung wird das Fragment dem multiplen Alignment hinzugefügt. Bei inkonsistenten Zuordnungen ist es nicht mehr möglich die Sequenzen in Spalten anzuordnen. Als Beispiel für inkonsistente Mengen siehe Abb. 6 . Fragmente, die zu der wachsenden Menge von konsistenten Fragmenten inkonsistent sind, werden verworfen. Die Summe der verworfenen Fragmente wird der „Gesamtscore der nicht konsistenten Fragmente“ genannt.

Abb.6 Zwei unkonsistente Mengen von Fragmenten

In dieser Abbildung handelt es sich um Aminosäuresequenzen. Es werden unterschiedliche Fragmente gezeigt, wie z.B. „I“ und „A“ alignt mit „L“ und „A“. Die linke sowie die rechte Menge, der dargestellten Mengen, ist inkonsistent. Auf der linken Seite der Abbildung wird das „F“ der dritten Sequenz zwei unterschiedlichen Aminosäuren der ersten Sequenz zugewiesen. Auf der rechten Seite werden Aminosäuren über Kreuz zugewiesen.[11]

Terminierung des Algorithmus

Der Algorithmus wird iterativ wiederholt, bis keine neuen Fragmente gefunden werden, die zur bereits berechneten Menge an konsistenten Fragmenten konsistent sind.

Wird kein Fragment mehr gefunden, wird das multiple Alignment im letzten Schritt formatiert. Dabei werden alle Fragmente in Spalten angeordnet und benötigte Lücken eingefügt.

4.1.3. DIALIGN -Parameter

Im Folgenden werden die Parameter erläutert, mit denen *DIALIGN* im Rahmen dieser Arbeit ausgeführt wird.

Die Parameter lauten: „-lgs -ta -fa -col_score“

Das Parameter „-lgs“ ist speziell für lange genomische Sequenzen und kombiniert mehrere Parameter miteinander. Diese Parameter sollen die Berechnung des Output-Alignments unter anderem durch gewisse Einschränkungen beschleunigen. Außerdem werden die Kriterien erhöht, die ein Fragment erfüllen muss, um von *DIALIGN* in das Output-Alignment aufgenommen zu werden. Im Folgenden werden die Auswirkungen des Parameters „-lgs“ auf die von *DIALGN* durchzuführenden Berechnungen beschrieben. Durch die Begrenzung der maximalen Länge der Fragmente auf 30 bp wird die Berechnung beschleunigt, wobei die Qualität des Alignments beeinträchtigt werden kann. Es werden nur die Fragmente bei der Berechnung des Alignments berücksichtigt, die am Anfang des Fragments mindestens zwei Matche aufweisen können. Dadurch wird zwar die Berechnung des Alignments beschleunigt, jedoch wird die Sensitivität verringert. Ein Fragment muss mindestens einen Alignment-Score von zwei haben, um bei der Berechnung berücksichtigt zu werden.

Es wird ein File mit dem Status der aktuellen Berechnungsposition, an der sich *DIALIGN* bei der Berechnung des Alignments gerade befindet, angelegt und zur Laufzeit aktualisiert.

Durch das Parameter „-ta“ wird das Outputalignment im *DIALIGN*-Alignment-Format ausgegeben (siehe Anhang Abb. ix) .

Die folgenden Parameter sind zur Erstellung von Output-Files, die von dem Visualisierungstool *ABC* [12] zur graphischen Darstellung des von *DIALIGN* berechneten Alignments benötigt werden. Mittels des Parameters „-fa“ wird neben dem Output-Alignment-File im *DIALIGN*-Alignment-Format das Alignment im *fasta*-Format (siehe Anhang Abbildung iv) ausgegeben. Mit dem Parameter „-col_score“ wird ein zusätzliches File erzeugt, in dem die Ähnlichkeit jeder Spalte im Alignment auf einer Skala von eins bis zehn enthalten ist.

4.2. Ankerpunkte

Ein Ankerpunkt „verankert“ Bereiche gleicher Länge in zwei Sequenzen miteinander. Das bedeutet bei der Berechnung des Alignments dieser Sequenzen, dass die verankerten Bereiche nicht mehr

voneinander getrennt werden können, da sie „verankert“ sind. Mit „verankern“ ist die feste Ausrichtung der am Ankerpunkt beteiligten Sequenzbereiche zueinander gemeint. Die „Verankerung“ von zwei Sequenzbereichen impliziert nicht die Alignierung dieser Bereiche.
Um einen Ankerpunkt zu charakterisieren benötigt man folgende Informationen: Die Bezeichnung der beiden Sequenzen die den Ankerpunkt beinhalten. Zudem benötigt man die beiden Startpositionen des zu verankernden Bereiches auf der jeweiligen Sequenz, die Länge des zu verankernden Bereiches und das Gewicht für den Anker.
Es sind zwei Sequenzen S_1 und S_2 gegeben. A_1 und A_2 bezeichnen zwei Sequenzabschnitte in der zugehörigen Sequenz. Der Sequenzabschnitt A_1 hat die Startposition a_1 auf S_1 und A_2 hat die Startposition a_2 auf S_2 .
Bei der Angabe des Ankerpunktes

P : S_1 S_2 a_1 a_2 <Länge> <Gewicht>

würde *DIALIGN* bei der Angabe dieses Ankerpunktes dazu „gezwungen“ werden, den gesamten Sequenzabschnitt links von Sequenzbereich A_1 nur mit dem Sequenzabschnitt links von Sequenzbereich A_2 zu alignieren und umgekehrt (siehe Abb. 7). Dabei kann der Alignmentsuchraum maximal um den Faktor zwei reduziert werden. Die Übergabe des Ankerpunktes an *DIALIGN* bedeutet nicht, dass die zwei durch den Ankerpunkt beschriebenen Bereiche miteinander aligniert werden.

Bei den von *DIALIGN* berechneten Fragmenten handelt es sich genau genommen um Ankerpunkte im Alignment(siehe Kapitel 4.1.2). Damit ist ein Ankerpunkt für die Bildung des Alignments an sich und für die Suche einer optimalen Kette von Fragmenten unter Berücksichtigung der Ankerpunkte definiert.
Wenn ein Ankerpunkt P konsistent zu dem Alignment ist, das ohne diesen Ankerpunkt P berechnet wurde, dann hat der Ankerpunkt P keinen Effekt auf das resultierende Alignment. Die übergebene Menge an Ankerpunkten wird vor der Einbeziehung in die Berechnung des Alignments auf Konsistenz überprüft. Dabei wird eine Greedy – Methode verwendet.

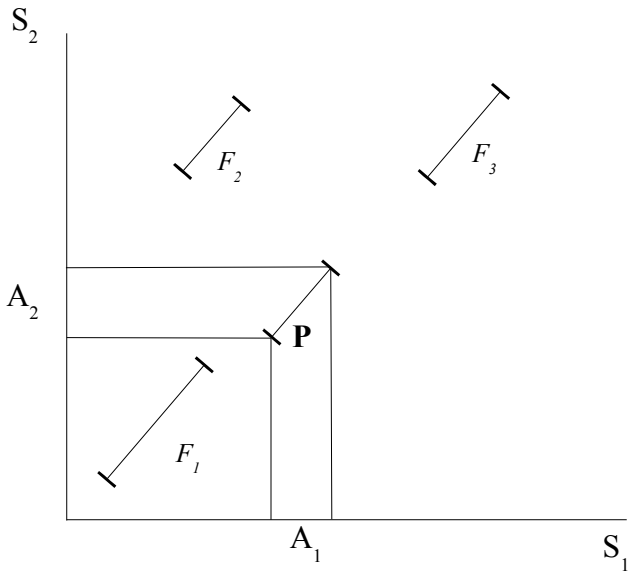

Abb.7 Graphische Darstellung eines Ankerpunktes P

S_1 und S_2 sind zwei Sequenzen. A_1 und A_2 bezeichnen zwei Sequenzabschnitte in den zugehörigen Sequenzen. F1, F2 und F3 sind Fragmente (siehe Kapitel 4.1.1.). P ist ein Ankerpunkt. Der durch den Ankerpunkt beschriebene Bereich muss nicht aligniert werden, jedoch kann der Sequenzabschnitt links von Sequenzbereich A_1 nur mit dem Sequenzabschnitt links von Sequenzbereich A_2 aligniert werden und umgekehrt.

Durch die Ankeroption kann der Benutzer auf das resultierende Alignment Einfluss nehmen. In erster Linie wird die Option benutzt, um die Laufzeit zu optimieren. Dazu müssen die Anker im biologischen Sinne qualitativ hochwertig sein.

Die Übergabe der Anker an *DIALIGN* wird durch die Option „-anc“ ermöglicht. Hierbei wird *DIALIGN* eine Datei übergeben, in der die Ankerpunkte in einem bestimmten Format (siehe Anhang Abb. iii) stehen.

4.3. *CHAOS*

4.3.1. Einleitung

CHAOS [2] ist ein schnelles Suchwerkzeug für Datenbanken, das eine Liste von lokalen paarweisen Sequenzähnlichkeiten erstellt. Diese werden von *DIALIGN* verwendet, um die finale Berechnung des Alignments zu beschleunigen. *CHAOS* kann auch als selbstständiges Programm zum Auffinden lokaler Sequenzähnlichkeiten verwendet werden. CHAOS werden zwei DNA-Sequenzen im *fasta*-Format (siehe Anhang Abb.i) als Input übergeben. *DIALIGN* ist jedoch ein multiples Alignmentprogramm, deshalb berechnet *CHAOS* für alle möglichen Sequenzpaare die Liste von lokalen Alignments und

wählt die besten Kandidaten aus [2].

Der Zusammenschluss der Alignmentprogramme *CHAOS* und *DIALIGN* wird die *CHAOS-DIALIGN* Methode genannt.

4.3.2. CHAOS-Algorithmus

Der *CHAOS* Algorithmus arbeitet mit der Verkettung von paarweise ähnlichen Regionen, wobei eine Region von jeder der zwei Input-Sequenzen stammt. Solche Paare von Regionen werden Seeds genannt [13]. Die Seeds sind Paare von Wörtern der Länge k mit mindestens n identischen Basenpaaren (bp).

Ein Seed wird mit einem anderen verkettet, wenn erstens die Indizes des einen Seeds in beiden Sequenzen höher sind, als die Indizes in dem anderen Seed und zweitens die Seeds eine gewisse Distanz nicht überschreiten. Der Score einer Kette berechnet sich durch die Summe aller identischen Basenpaare in ihr. Nachdem die maximalen Ketten berechnet wurden, weist *CHAOS* jeder Kette einen Wert unter Berücksichtigung von Match- und Mismatch-Strafen für die Zeichen von jedem Seed zu. Für zwei Seeds, die durch die Basen x und y in der ersten und der zweiten Sequenz getrennt sind, wird eine Lückenstrafe, die proportional zu |x-y| ist, hinzu gezogen. Die Ketten, die unterhalb von einem Grenzwert T liegen, werden verworfen. Bei Ketten, die einen Score oberhalb von T haben, werden die Seeds nach beiden Richtungen lückenlos ausgedehnt und es wird eine geeignete Stelle gesucht, um genau eine Lücke der Länge |x-y| einzufügen. Diese Methode ermöglicht eine effiziente Berechnung von lokalen Alignments.

5. *TBLASTX* und *BLAST*

5.1. *BLAST*

5.1.1. Einleitung

BLAST [14] ist ein bekanntes Alignment-Tool, das in verschiedenen Ausprägungen frei verfügbar ist. Das Prinzip hinter den unterschiedlichen Versionen ist immer gleich. Die Grundidee des *BLAST*-Algorithmus ist, dass lokale Alignments sehr wahrscheinlich kurze Bereiche exakter Identität oder Bereiche mit einem sehr hohen Score enthalten. Diese Bereiche werden zu längeren Alignments in beide Richtungen ausgebaut [15].

Bei *TBLASTX* läuft die Berechnung der lokalen Alignments auf Proteinebene ab. Dadurch liegt der Berechnung eine komplexere Statistik als auf Nukleotidebene zugrunde. Die höhere Komplexität begründet sich unter anderem durch das größere Alphabet und damit durch eine komplexere Scoring-Matrix (siehe Kapitel 5.2).

5.1.2. Blast-Algorithmus

Der in diesem Kapitel beschriebene *BLAST*-Algorithmus bezieht sich auf die in der vorliegenden Arbeit verwendeten Version von *TBLASTX* (siehe Kapitel 5.2).

Der aus drei Schritten bestehende *Blast*-Algorithmus wird zunächst beschrieben, anschließend wird ein Einblick in die Grundlagen des statistischen Scoring-Schemas von Blast gegeben.

Gegeben sind eine Suchsequenz S, eine Datenbank DB = $\{D_1,...,D_n\}$ und eine Substitutionsmatrix. Diese Matrix listet einen Score s_{ij} für die Alignierung von jedem Aminosäurepaar i und j auf [14]. Hier wird eine *BLOSUM62*-Substitutionsmatrix (siehe Anhang Abb. v) verwendet.

Die drei Schritte des *BLAST*-Algorithmus sind:

1. Bestimmung der Teilwörter der Suchsequenz
2. Suche von Hits gegen die Datenbank
3. Ausdehnung der Hits

<u>Bestimmung der Teilwörter der Suchsequenz</u>

Anfangs werden aus der Eingabesequenz alle Substrings $S_1,..,S_{m-k-1}$ der Länge k gebildet, wobei m die Länge der Eingabesequenz ist. Für die k-mere wird k = 4 als default-Wert verwendet. Bei Proteinsequenzen ist der default-Wert normalerweise k = 3, aber da in diesem Fall eine *BLOSUM62*-

Matrix verwendet wird, ist der default-Wert k = 4. Zum Beispiel sehe die Menge M der k-mere mit k = 2 für die Sequenz S = agtta wie folgt aus: M = {ag, gt, tt, ta}.

Suche von Hits gegen die Datenbank

Die von *formatdb* erzeugte Datenbank wird nach allen exakten Matches mit den Teilwörtern $S_1,..,S_m$ durchsucht. Darüber hinaus werden für jedes S_i alle Positionen mit allen Aminosäuren permutiert, um die Sensitivität zu steigern. Ist S_i mit einem Bereich derselben Länge in DB alignt und hat dieses Alignment einen Score größer als der Grenzwert T, so nennt man dieses Alignment einen Hit. Im folgenden wird die „Two Hit" Methode von *BLAST2* beschrieben.

Die Distanz $D(x_1,y_1)$ zwischen zwei Hits x und y auf derselben Diagonale ist die Differenz ihrer jeweils ersten Koordinaten. Findet man zwei Hits x und y mit $D(x_1,y_1) < A$ auf derselben Diagonale im Dotplot, wobei A ein Grenzwert ist und diese Hits sich zudem nicht überlappen, werden sie bei der weiteren Suche berücksichtigt. Der Grenzwert T ist wegen der beschriebenen „Two Hit" Methode des *BLAST 2* Algorithmus niedriger als bei der älteren *BLAST*-Version, da die meisten gefundenen Hits aufgrund des Grenzwertes A verworfen werden. Zudem wird jeder Hit, der den zuletzt berechneten überlappt, ignoriert. Dies wird effizient durch Arrays umgesetzt, wobei für jede Diagonale ein Array existiert, welches die ersten Koordinaten des jeweils zuletzt gefundenen Hits beinhaltet. Sobald es Hits gibt, die die eben genannten Kriterien erfüllen, werden sie im nächsten Algorithmusschritt zu verlängern versucht [14].

Ausdehnung der Hits

Dieser Algorithmusschritt beansprucht 90 % der gesamten von *BLAST* benötigten Berechnungszeit.

Jeder Diagonalenabschnitt, der durch die oben beschriebenen zwei Hits begrenzt wird, wird bidirektional solange ausgedehnt, bis sich der Score nicht mehr verbessert. Bei der Ausdehnung werden keine Gaps erlaubt. Diese optimalen lokalen Alignments werden HSP's (high scoring pairs) genannt. Das HSP mit dem höchsten Score zwischen zwei Sequenzen bezeichnet man als MSP (Maximal Segment Pair). Definiert ist ein MSP als Segmentpaar mit dem höchsten Score von allen möglichen Segmentpaaren zweier Sequenzen, das durch zwei Sequenzen und einer Scoring-Matrix berechnet werden kann [14].

Die statistischen Grundlagen, die der von *BLAST* verwendeten Substitutionsmatrix zugrunde liegen, sollen nun kurz erläutert werden. Grundlage ist ein Proteinmodell, bei dem die Aminosäure i zufällig an

allen Positionen mit der Hintergrundwahrscheinlichkeit P_i auftreten kann. Es wird verlangt, dass der Erwartungswert für zwei zufällige Aminosäuren

$$\sum_{i,j} P_i P_j s_{ij}$$

negativ ist, wobei s_{ij} der Score für die Alignierung von jedem Aminosäurepaar i und j ist. Gegeben sind P_i und s_{ij}, hinzu kommen zwei kalkulierbare Parameter λ und *K*, die so genannten Karlin-Altschul Parameter. Diese Parameter konvertieren den nominellen HSP Score in einen normalisierten Score. Durch den statistischen Blickwinkel werden alle unterschiedlichen „Gewichtungssysteme" direkt vergleichbar. Der normalisierte HSP Score S' wird durch

$$S' = \frac{(\lambda S - \ln K)}{(\ln 2)}$$

beschrieben. Dieser Score hat die Einheit „bits". Bei der hier verwendeten BLOSUM62 – Matrix für lückenlose lokale Alignments werden die Parameter λ = 0,3176 und *K*= 0,134 zur Berechnung verwendet [14].

5.2. *TBLASTX*

TBLASTX (Translated Basic Local Alignment Search Tool X) [3] ist ein schnelles Alignmentprogramm, welches lokale paarweise Alignments für zwei DNA-Sequenzen berechnen kann. *TBLASTX* erhält als Eingabe eine Nukleotidsequenz und sucht in Nukleotidsequenzen einer gegebenen Datenbank. Sowohl die Datenbanksequenz, als auch alle „Suchsequenzen" werden in alle sechs Leserahmen („six-frame translation") übersetzt. Damit ist eine Übersetzung der DNA-Sequenz in alle sechs möglichen Aminosäuresequenzen gemeint. Dabei wird der Sense und der Antisensestrang jeder DNA-Sequenz jeweils in die drei Leserahmen übersetzt. Dies bedeutet, das die Suche von einer Eingabesequenz gegen eine Datenbank 36 Sequenzvergleiche benötigt. Dieser sehr große Rechenaufwand ist angemessen, da es sich um unbekannte Sequenzen handelt, von denen man nicht genau weiß, welcher der Leserahmen der Richtige ist. Aufgrund des degenerierten Codes sind die Sequenzunterschiede auf Nukleotidebene größer als auf Proteinebene, z.B. wird Leucin auf DNA-Ebene durch 6 verschiedene Tripletts und auf Proteinebene nur durch Leucin kodiert, d.h. auf Proteinebene sind kodierende Bereiche besser konserviert. Damit liefert *TBLASTX* gute Voraussetzungen, Ankerpunkte in Exons zu berechnen.

Mittels des Datenbanktools *formatdb* [16] wird die zum Sequenzvergleich benötigte Datenbank

erzeugt. Das Alignment selbst wird mit dem *BLAST 2 (Basic Local Alignment Search Tool 2)* Algorithmus berechnet.

5.3. Datenbanktool: *formatdb*

BLAST benötigt eine Vorverarbeitung der Sequenzdatenbank, bevor die *BLAST*-Suche gestartet werden kann. Hierzu werden die Input-Sequenzen mittels *formatdb* [16] in ein spezielles *Blast*-Format umgewandelt und zusätzliche Index-Strukturen angelegt.
In dem für diese Arbeit verwendeten Fall führt *formatdb* mit Nukleotidsequenzen im *fasta*-Format (Siehe Anhang Abb.i) zuerst eine six-frame-translation (siehe Kapitel 5.2) durch und erstellt anschliessend aus den sechs erhaltenen Proteinsequenzen eine Datenbank im *BLAST*format. Die dabei erzeugten Dateien mit den Endungen *.nsi* und *.nsd* enthalten Indizes, die ein Abbild des Sequenzbezeichners auf die jeweilige Sequenz in der Datenbank sind. Die Sequenzdaten sind in einer Datei mit der Endung *.nsq* abgelegt.
Alle von *formatdb* angelegten Files liegen im Binärformat vor [12].

5.4. Vergleich von *TBLASTX* und *CHAOS*

CHAOS benötigt weniger Zeit zur Berechnung der lokalen Alignments als *TBLASTX*, da bei *TBLASTX* zuerst die DNA-Sequenzen jeweils in alle sechs Aminosäuresequenzen mittels der six-frame-translation übersetzt werden müssen. Bei *TBLASTX* werden durch die verschiedenen Leserahmen mehr Sequenzvergleiche als bei *CHAOS* durchgeführt. Auf Proteinebene hat *TBLASTX* bessere Voraussetzungen die lokalen Alignments in Exons zu finden, da die kodierenden Bereiche auf Proteinebene besser konserviert sind, als auf Nukleotidebene.

6. Schnittstelle zwischen *TBLASTX* und *DIALIGN*

6.1. Einleitung

Die Implementierung der Schnittstelle zwischen *TBLASTX* und *DIALIGN* besteht vorwiegend aus Textformatierungen. Deshalb wurde die Schnittstelle auch vollständig in der Programmiersprache Perl implementiert. Perl ist für seine Anwenderfreundlichkeit hinsichtlich Formatierungen jeglicher Art bekannt. Zudem stellt Perl das Zusatzpaket BioPerl [17] für die Bioinformatik zur Verfügung, das den Umgang mit großen Datensätzen erleichtert.

Außer den durchzuführenden Textformatierungen müssen die Parameter von *TBLASTX* und *DIALIGN* bestimmt werden. Die Festlegung der Parameter erfolgt unter zwei Kriterien: Erstens soll die Laufzeit von *DIALIGN* verbessert und zweitens der numerische Score, den *DIALIGN* seinem erstellten Alignment gibt, beibehalten werden.

Die Verwendung von *TBLASTX* wird dadurch motiviert, dass die Ankerpunkte, die an *DIALIGN* übergeben werden, biologisch qualitativ hochwertig sein müssen. Sind sie es nicht, wird *DIALIGN* dazu „gezwungen“, ein minderwertiges Alignment zu bilden. Am besten geeignet wären Ankerpunkte, die verifizierten Homologien entsprechen, und somit nachweislich biologisch richtig sind. Jedoch ist diese Vorgehensweise bei nicht annotierten Sequenzen nicht möglich. Deshalb werden zur Bestimmung einer Menge von Ankerpunkten lokale Alignmentprogramme verwendet. Es kann dabei nicht mit Sicherheit gesagt werden, ob die berechneten Anker biologisch richtig sind. Es werden nur die lokalen Alignments verwendet, die einen sehr hohen numerischen Score über einem gewissen Grenzwert haben. Dieser Grenzwert ist meist so hoch, dass nur wenige lokale Alignments über dem Grenzwert liegen. Andererseits sollen die Ankerpunkte die Laufzeit von *DIALIGN* optimieren, wobei gilt: je mehr Anker vorliegen, desto mehr wird der Suchraum bei der Berechnung der Alignments eingeschränkt (siehe Kapitel 4). Die Idee ist nun mit dem Grenzwert für den numerischen Score der von *TBLASTX* berechneten lokalen Alignments, die als Ankerpunkte verwendet werden sollen, „so weit herunter zu gehen“, dass die Qualität des von *DIALIGN* berechneten Alignments beibehalten wird und der Geschwindigkeitsvorteil auf Grund der größeren Anzahl von Ankern maximal ist. Dadurch, dass *TBLASTX* die lokalen Alignments, wie in Kapitel 5 beschrieben, auf Proteinebene berechnet, könnte die biologische Qualität der berechneten Anker, trotz des niedrigeren Grenzwertes, für ein biologisch sinnvolles Alignment ausreichend sein. Die Methode zur Berechnung des Outputalignments mit *TBLASTX* und *DIALIGN* wird *TBLASTX-DIALIGN* genannt.

6.2. Implementierung der Schnittstelle in Perl

Die Schnittstelle besteht aus aus einem Perlskript *anker.pl* und zwei Perlmodulen *format.pm* und *datenbank.pm*.

Der Input der Schnittstelle ist eine Datei im *fasta*-Format (siehe Anhang Abb. i), welche die zu alignierenden Sequenzen enthält. Außerdem müssen der Pfad, in dem die übergebene Datei liegt (Pfad 1) und der Pfad in dem *TBLASTX* und *formatdb* installiert sind (Pfad 2), übergeben werden. Der Aufruf der Schnittstelle lautet:

perl anker.pl <Datei im fasta-Format> <Pfad1> <Pfad2>

Der Output, der von der Schnittstelle erzeugt wird, ist eine Datei im von *DIALIGN* benötigtem Ankerformat (siehe Anhang Abbildung iii). Nun werden zuerst die beiden Module und anschließend das Perlskript *anker.pl* beschrieben, welches das Hauptprogramm darstellt.

Modul datenbank

Das Modul *datenbank* beinhaltet die Methode *db( )*. Diese Methode erzeugt die für den Aufruf von *TBLASTX* benötigte(n) Datenbank(en) und führt anschließend den (die) *BLAST*aufruf(e) gegen die Datenbank(en) durch.

Zuerst wird jede einzelne Sequenz aus der Input-Datei in eine eigene Datei geschrieben. Parallel werden Sequenzen, die länger als 100000 bp lang sind, in Sequenzstuecke mit einer Länge von 100000 bp zerlegt. Diese Teilstücke haben die Startposition d-1 * 100000 mit d $\in$ 1,...,m Sequenzstücke.

Zum Beispiel wird eine Sequenz, der Länge 234000 in zwei Sequenzen 100000 und eine Sequenz der Länge 34000 zerlegt. Die Startpositionen der einzelnen Teilsequenzen in ihren ursprünglichen Sequenzen wird in einer Variablen vermerkt. In diesem Beispielfall stellt Null die erste, 100000 die zweite und 200000 die dritte Startposition der jeweiligen Teilsequenz dar. Dieser sogenannte Cutoff der Sequenzen wird bei 100000 bp durchgeführt, weil bei jeder *BLAST*-Suche nur 200 HSP's gesucht werden und durch die Aufteilung der Sequenzen folglich maximal d-1 * 200 HSP's und lokale Alignments mit geringerer Sequenzähnlichkeit gefunden werden können. Außerdem entstehen durch diesen betriebenen Aufwand bei der *BLAST*-Suche keine längeren Laufzeiten. Anschließend werden aus den n nicht unterteilten Sequenzen n-1 Datenbanken mittels des Tools *formatdb* erzeugt. Bei multiplen Alignments wird bei 1,...,n Sequenzen beim i-ten Aufruf von *formatdb* eine Datenbank erstellt, die die Sequenzen i+1,...,n beinhaltet.

Bei n Sequenzen, die alle kürzer als 100000 bp sind, werden n-1 *BLAST-A*ufrufe durchgeführt. Bei

1,...,n Sequenzen , wird die i-te Sequenz gegen die Datenbank „geblastet“, welche die i+1,...,n Sequenz enthält. Bei Sequenzen mit einer Länge über 100000 bp werden alle Teilsequenzen der i-ten Sequenz einzeln gegen die Datenbank, welche die i+1,...,n Sequenz enthält, „geblastet“. Bei dieser Vorgehensweise werden alle möglichen paarweisen Sequenzalignments gebildet und es entstehen keine redundanten Daten. Die nicht mehr benötigten Datenbank-Files werden gelöscht.

Modul *format*

Das Modul *format* beinhaltet die Methoden *anc( )* und *write( )*. In der Methode *anc( )* wird das von *TBLASTX* erzeugte *.out*-File (siehe Anhang Abbildung vi) in das von *DIALIGN* benötigte *.anc*-File formatiert.

Das Output-File von *TBLASTX* enthält lokale paarweise Sequenzalignments, die nach dem Score sortiert sind. Anhand eines Grenzwertes für den Score werden die lokalen Alignments, die als Anker verwendet werden sollen, selektiert. Auf Grund der durch die unterschiedlichen Testdatensätze erzielten Ergebnisse, wurde ein Grenzwert von 150 bits zur Auswahl der lokalen Alignments festgelegt (siehe Kapitel 6.3.).

Erfüllt der Score eines Alignments den Grenzwert, wird nicht das ganze Alignment als Anker verwendet, sondern nur die Bereiche, die in Anfrage- und Datenbanksequenz identische Aminosäuren enthalten. Diese Information, der Ähnlichkeit der Aminosäuren zueinander, ist bei *TBLASTX* in einer extra Zeile enthalten. Diese Zeile wird zur Erstellung der Anker verwendet (siehe Abb. 7). Das Alignment liegt auf Proteinebene vor, die Ankerpunkte sollen jedoch für die Sequenzen auf Nukleotidebene verwendet werden und müssen deshalb umgerechnet werden. Bei der Umrechnung wird die Regel verwendet, dass ein Triplett eine Aminosäure kodiert, also eine Aminosäure durch drei Basenpaare dargestellt wird.

In der Methode *write( )* werden die von der Methode *anc( )* berechneten Ankerpunkte in den *.anc*-File, der *DIALIGN* übergeben werden soll, geschrieben.

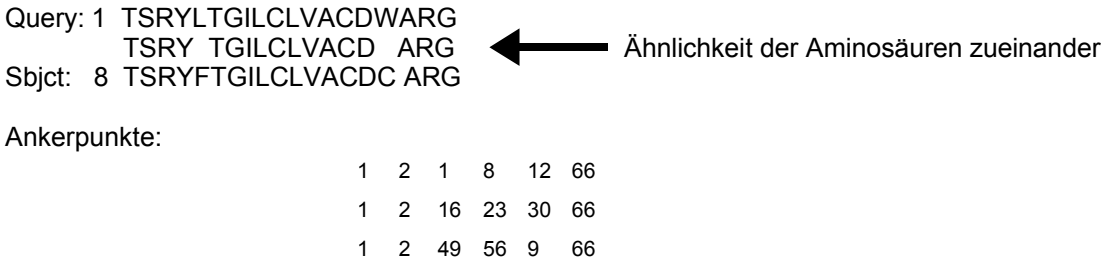

Abb. 8 Berechnung der Ankerpunkte anhand des TBLASTX-Outputs

Diese drei Ankerpunkte würden sich bei einem Alignment - Score von 66 bits aus den zwei Sequenzen, die durch „Query“ und „Subjt“ gegeben sind, ergeben.
Erklärung des Ankerformates, siehe Anhang Abbildung iii.

anker.pl

Das Perlskript *anker.pl* dient der ganzen Schnittstelle als Hauptprogramm. Zuerst wird die Methode *db()* mit dem File im *fasta*-Format aufgerufen, womit nach Ausführung der Methode die von *TBLASTX* errechneten lokalen Alignments im *.out*-Format (siehe Anhang Abb. vi) vorliegen. Anschließend wird für jedes von *TBLASTX* erzeugte *.out*-Format File die Methode *format( )* aufgerufen, welche die Formatierung von den paarweise lokalen Alignments zu den Ankerpunkten vornimmt. Bei der Übergabe des *TBLASTX* Outputs an *format( )* wird zudem die oben beschriebene Startposition der Suchsequenz („Query“) des zugehörigen *TBLASTX* Outputs angegeben.Nun wird *DIALIGN* mit den zu alignierenden Sequenzen im *fasta*-Format und dem Ankerfile, welches zuvor berechnet wurde, aufgerufen.

In dem Skript *anker.pl* wird direkt am Anfang eine Datei angelegt und geöffnet, in die der Ausführungsstart des Programms anker.pl geschrieben wird. In der letzten Zeile des Skripts wird dieser Datei das Ausführungsende mitgeteilt. Damit liegt die genaue Ausführungszeit des gesamten Vorgangs zur Erstellung des Alignments aus den übergebenen Sequenzen vor. Diese Zeit beinhaltet alle Schritte zur Berechnung der Anker und Ausführung und Berechnung des Alignments mittels *DIALIGN*.

6.3. Ankerselektion

Eine Teilmenge der Menge von *TBLASTX* berechneten lokalen paarweisen Alignments soll nun bestimmt werden. Diese Teilmenge wird im Folgenden als Ankermenge bezeichnet. Die Alignments, die in der Ankermenge enthalten sein sollen, müssen eine gewisse biologische Qualität aufweisen. Im Folgenden werden nun die Kriterien erläutert, nach denen die Qualität der Anker beurteilt wird, soweit es möglich ist. Diese „Qualität“ muss zur Selektion der Anker in Bezug zum *TBLASTX*-Score gesetzt

werden.

Zur Beurteilung, ob die Elemente der Ankermenge diese biologische Qualität aufweisen, wird zum einen anhand von annotierten Daten der Ensembl-Datedenbank [4] überprüft (siehe Kapitel 7). Dabei werden *TBLASTX* unterschiedliche Datensätze übergeben und anschließend die Ergebnisse der Berechnung darauf hingehend untersucht, ob die berechneten lokalen Alignments in den Exons der beiden beteiligten Sequenzen liegen. Zum anderen wird *DIALIGN* mit unterschiedlichen Ankermengen bzw. mit unterschiedlichen *.anc*-Files zu einem Datensatz aufgerufen. Hierzu wird die im vorherigen Kapitel beschriebene Schnittstelle mit unterschiedlichen Grenzwerten für *TBLASTX* ausgeführt. Die Ergebnisse dieser Programmausführungen werden miteinander verglichen. Zuerst wird überprüft, ob die lokalen Alignments in den orthologen Genen der beiden beteiligten Sequenzen liegen, bzw. in den Exons des orthologen Gens.

Bei der Betrachtung der Ergebnisse steht der *TBLASTX*-Score der verwendeten Anker im Vordergrund, weil durch ihn eine Selektion der als biologisch sinnvoll erachteten Alignments möglich wird. Um dabei eine allgemeine Aussage treffen zu können, werden unterschiedliche Datensätze verwendet, deren Unterschiede nicht nur in der Anzahl und der Länge der Sequenzen des einzelnen Datensatzes, sondern auch in dem Verwandtschaftsgrad der Sequenzen untereinander liegen. Die Auswertung der Ergebnisse ergab, dass die lokalen Alignments ab einem *TBLASTX*-Score von 150 bits fast immer in den kodierenden Bereichen liegen. Ein paar „Ausreißer“ gibt es bei diesem Grenzwert. Jedoch gibt es diese „Ausreißer“ bei höherem Score ebenfalls. Die Anzahl der „Ausreißer“ bei einem Score unter 150 bits deutlich zu.

Beim zweiten Teil des Tests zur Selektion der Anker, wird *DIALIGN* mit unterschiedlichen Ankermengen zu demselben Datensatz aufgerufen. Die einzelnen Ankermengen, die *DIALIGN* übergeben werden, sind Teilmengen der von *TBLASTX* zu diesem Datensatz berechneten Ankermenge. Diese Teilmengen unterscheiden sich durch den *TBLASTX* – Score, der in ihnen enthaltenen lokalen Alignments. Es wird für jede Menge ein anderer Grenzwert bezüglich des *TBLASTX*–Scores festgelegt, den jedes lokale Alignment der jeweiligen Menge erfüllen muss.

Nachdem *DIALIGN* mit den unterschiedlichen Ankermengen jeweils mit den verschiedenen Datensätzen ausgeführt wurde, werden die Ergebnisse verglichen (siehe Tabelle 1-3). Dabei wird jeweils die Laufzeit betrachtet, die zur Erstellung des gesamten globalen Alignments benötigt wird. Zudem kommt die Betrachtung der an *DIALIGN* jeweils übergebenen Ankermenge.

Datensatz 1						
Grenzwert für die Anker	50	100	150	200	300	400
Laufzeit in Minuten	5	5	5	27	27	27
DIALIGN-Alignment-Score	1068,38	1068,38	1068,38	1104,42	1104,42	1104,42
Gesamtscore der nicht konsistenten Fragmente	2,53	2.53	2,53	17,25	17,25	17,25
Anzahl der Anker	1015	287	83	0	0	0
Anzahl der inkonsis. Anker	437	137	24	0	0	0
inkons. Anker in %	43,05	47,74	28,92	0	0	0

Tabelle 1: Ergebnisse der mehrfachen DIALIGN-Ausführung mit verschiedenen Ankermengen zu dem Datensatz 1

Die Tabelle zeigt die Resultate der mehrfachen Ausführung von DIALIGN mit dem Datensatz 1 (siehe Kapitel 7). Bei den einzelnen Ausführungen wurden jeweils unterschiedliche Ankermengen verwendet. Die verschiedenen Ankermengen haben einen unterschiedlichen Grenzwert für die Anker. Der Grenzwert bezieht sich auf den TBLASTX-Score eines jeden Ankers und muss mindestens von dem Anker erfüllt sein, um in die jeweilige Menge aufgenommen zu werden. Je höher der Grenzwert ist, um so hochwertiger ist die Qualität der bei dieser DIALIGN-Ausführung verwendeten Anker. Die angegebene Laufzeit beinhaltet den Vorgang zur Erstellung des Alignments aus den übergebenen Sequenzen, inklusive aller Schritte zur Berechnung der Anker und der Ausführung und Berechnung des Alignments mittels DIALIGN. Der DIALIGN-Alignment-Score ist ein numerischer Score, den DIALIGN berechnet hat. Er gibt die „Qualität" des berechneten Alignments an. Der Gesamtscore der nicht konsistenten Fragmente wird in Kapitel 4.1.2 erläutert.

Datensatz 2						
Grenzwert für die Anker	50	100	150	200	300	400
Laufzeit in Minuten	7	10	12	16	51	78
DIALIGN-Alignment-Score	36141,57	36278,96	36355,37	36460,26	36515,24	36608,99
Gesamtscore der nicht konsistenten Fragmente	195,84	299,06	362,63	440,07	407,06	398,16
Anzahl der Anker	15999	8357	5325	3959	2545	1857
Anzahl der inkonsis. Anker	3848	325	212	156	37	36
inkons. Anker in %	24	3,99	3,98	3,94	1,45	1,94

Tabelle 2: Ergebnisse der mehrfachen DIALIGN-Ausführung mit verschiedenen Ankermengen zu dem Datensatz 2

Erläuterung siehe Tabelle 1.

Datensatz 3						
Grenzwert für die Anker	50	100	150	200	300	400
Laufzeit in Minuten	9	9	9	9	10	15
DIALIGN-Alignment-Score	41934,81	41934,81	41934,81	41934,81	42202,85	42672,2
Gesamtscore der nicht konsistenten Fragmente	0	0	0	0	0	0
Anzahl der Anker	8592	8592	8592	8592	7591	3680
Anzahl der inkonsis. Anker	0	0	0	0	0	0
inkons. Anker in %	0	0	0	0	0	0

Tabelle 3: Ergebnisse der mehrfachen DIALIGN-Ausführung mit verschiedenen Ankermengen zu dem Datensatz 3
Erläuterung siehe Tabelle 1.

Bevor *DIALIGN* die Berechnung des Output-Alignments beginnt, wird die Ankermenge auf Konsistenz überprüft. Bei der Erstellung der konsistenten Menge an Ankerpunkten, bedient sich *DIALIGN* der Greedy Strategie und nimmt bei zwei Ankern die zueinander inkonsistent sind, den Anker in die konsistente Menge auf, der den höheren Score hat.

In dieser Arbeit wird der Anteil der inkonsistenten Anker an der *DIALIGN* übergebenen Ankermenge überprüft. Dies ist von Interesse, weil die konsistenten Anker als die richtigen Anker angesehen werden. Denn ist eine Ankermenge konsistent, sind die Anker in ihr kollinear. Allgemein bedeutet kollinear, dass mindestens drei Punkte auf einer Geraden liegen. In dem vorliegenden Fall bedeutet es, dass die Anker eine nicht überlappende Kette bilden. Hingegen müsste bei einem hohen Anteil an inkonsistenten Ankern überlegt werden, ob die Methode diese Anker zu bestimmen, die Richtige ist. Bestimmt man eine Menge von Ankern zufällig, ist der Anteil an inkonsistenten Ankern sehr groß. (Siehe Tabelle 4) Konsistente Anker werden im weiteren als richtige Anker angesehen. Bei den Datensätzen, bei denen inkonsistente Anker vorlagen, nahm der prozentuale Anteil an inkonsistenten Ankern unter einem *TBLASTX*-Score von 200 bits stark zu (siehe Abb. 9). Außerdem wird die Summe des *DIALIGN*-Scores der einzelnen inkonsistenten und der konsistenten Fragmente, unter Verwendung unterschiedlicher Ankermengen, berechnet (siehe Tabelle 1-3).

Zufällig generierte Ankerpunkte				
Datensatz	Anzahl der Anker	Anzahl inkons. Anker	inkons. Anker in %	*DIALIGN*-Alignment-Score
1	11	6	54.5	982.88
1	50	36	72.0	655.01
1	86	66	76.7	685.96

Tabelle 4 : Zufällige Ankerpunkte

DIALIGN wurde jeweils mit unterschiedlichen zufällig generierten Mengen von Ankerpunkten ausgeführt. Die unterschiedlichen Mengen von Ankerpunkten wurden zufällig mittels eines Perlskripts zu Datensatz 1 generiert. Die Tabelle zeigt den prozentualen Anteil an inkonsistenten Ankerpunkten in der jeweiligen Menge.

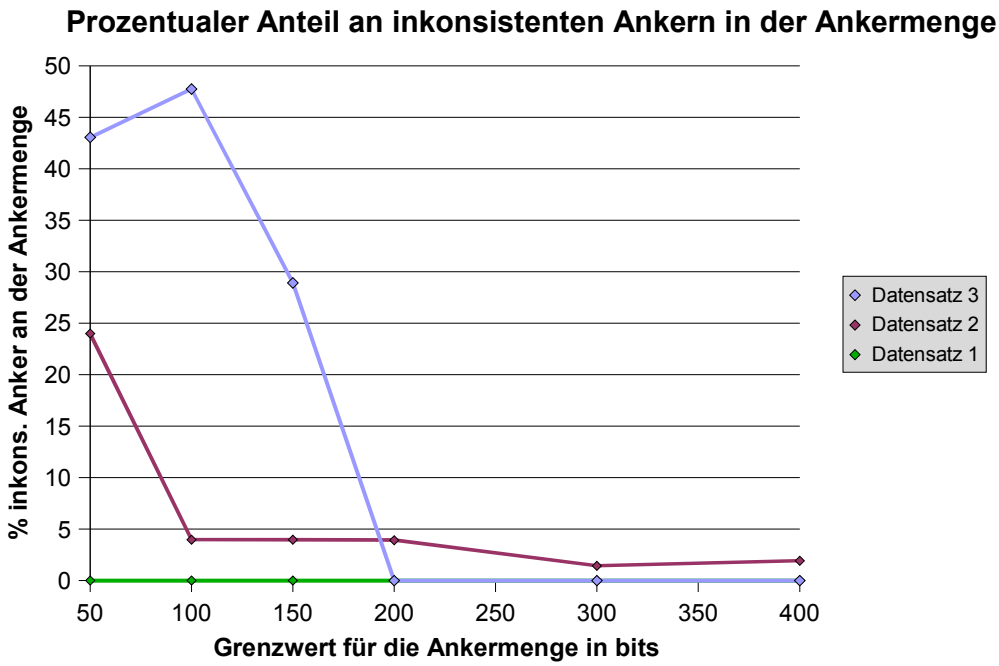

Abb. 9 Inkonsistente Anker in Prozent

DIALIGN wird mit unterschiedlichen Ankermengen jeweils mit den drei Datensätzen ausgeführt. Dabei unterscheiden sich die Ankermengen durch ihre Qualität von einander. Die Qualität wird durch den „Grenzwert für Ankermenge in bits“ angegeben, wobei der Grenzwert das Minimum des TBLASTX-Scores in bits angibt, den jeder Anker in dieser Menge mindestens haben muss. In der Abbildung wird der prozentuale Anteil der inkonsistenten Anker der verschiedenen Ankermengen für den jeweiligen Datensatz dargestellt. Einzelheiten zu den Datensätzen siehe Kapitel 7.

Der von *DIALIGN* berechnete Score ist nicht mit dem von *TBLASTX* berechneten Score zu verwechseln, da sie auf unterschiedlichen Berechnungsmethoden basieren. Die Summe der konsistenten Fragmente bzw. ihr von *DIALIGN* zugewiesener Score, ergibt den numerischen Score, den *DIALIGN* dem von ihm berechneten globalen Output-Alignment zuordnet, den sogenannten *DIALIGN*-Alignment-Score (siehe Kapitel 4.1.2). Damit können die von *DIALIGN* auf unterschiedlichen Ankermengen basierenden Alignments, in Hinblick auf ihre Qualität miteinander in Bezug gesetzt werden. Um so größer der berechnete numerische Score ist, desto besser wird die biologische Qualität der Alignments eingestuft. Die von *DIALIGN* berechneten Fragmente liegen nach der Ausführung von *DIALIGN* mit dem zusätzlichen Parameter „-ff“ als Outputfile mit der Endung *.frg* vor (siehe Anhang Abb. vii).

Die hier betrachteten drei Datensätze (siehe Kapitel 7) sind exemplarisch für alle durchgeführten Versuche hinsichtlich der Festlegung eines default-Wertes als Grenzwert für den *TBLASTX*-Score, den ein lokales Alignment haben muss, um als Anker Verwendung zu finden. Bei der Betrachtung der Laufzeit wird deutlich, dass die Grenze für den *TBLASTX*-Score zwischen 100 und 150 bits liegen sollte (siehe Abb. 9). Besonders effiziente Auswirkungen auf die Berechnung des Alignments hat dieser Grenzwert bei Sequenzen mit hoher Sequenzähnlichkeit. Abschließend wird die biologische Qualität, wie oben erklärt, für Datensätze mit naher und entfernter Sequenzverwandtschaft überprüft. Aus Abb.10 und Abb.11 wird ersichtlich, dass es kaum Abweichungen bei dem *DIALIGN*-Alignment-Score der jeweiligen Alignments gibt. Diese geringen Abweichungen des *DIALIGN*-Alignment-Scores waren unerwartet, da z.B. im zweiten Datensatz bei einem Grenzwert von 50 bits 15999 Anker und bei 400 bits lediglich 1857 von *DIALIGN* verwendet wurden. Auf Grund der minimalen Abweichungen der *DIALIGN*-Alignment-Scores, fällt diese Kategorie bei der Festlegung des Grenzwertes nicht ins Gewicht. In Anbetracht der Ergebnisse wird ein Grenzwert für die von *TBLASTX* berechneten lokalen Alignments von 150 bits festgelegt, den sie erfüllen müssen, um in die Ankermenge aufgenommen zu werden. Dieser Grenzwert von 150 bits rechtfertigt sich durch die oben beschriebenen kürzeren Laufzeiten von *DIALIGN* und der geringen Anzahl von inkonsistenten Ankern.

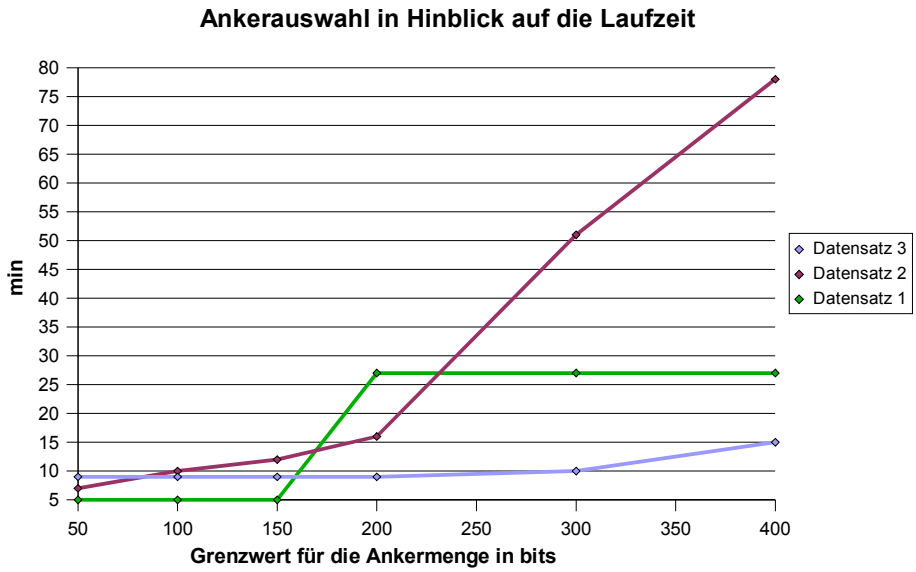

Abb. 10 Laufzeitvergleich

In der Abbildung wird die Laufzeit von DIALIGN mit den verschiedenen Ankermengen für den jeweiligen Datensatz dargestellt. Die angegebene Laufzeit bezieht sich auf die gesamte Zeit, die zur Erstellung des globalen Output-Alignment von DIALIGN benötigt wurde, inklusive der Zeit zur Berechnung und Formatierung der Ankerpunkte. Weitere Erläuterungen siehe Abb. 9.

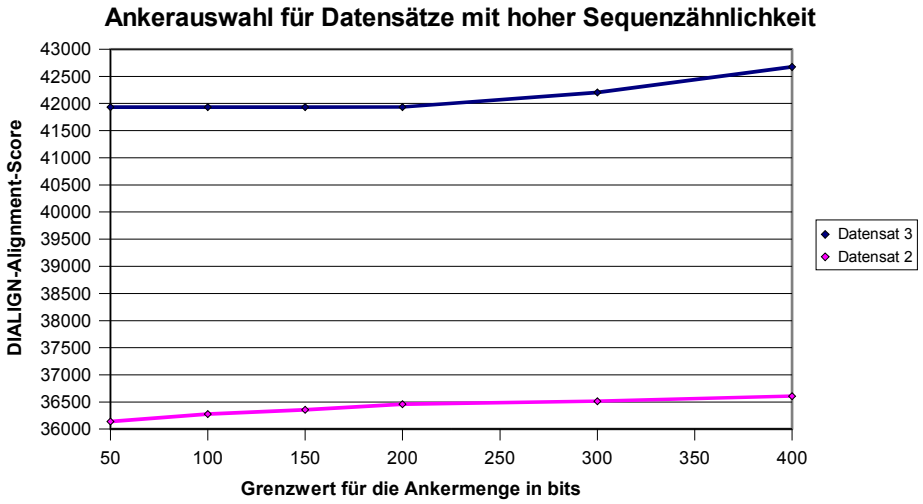

Abb. 11 Qualität der Alignments bei hoher Sequenzverwandtschaft

In der Abbildung wird die einzelne Qualität des von DIALIGN erstellten Output Alignments mit den verschiedenen Ankermengen für den jeweiligen Datensatz dargestellt. Die Datensätze beinhalten jeweils Sequenzen mit hoher Sequenzverwandtschaft zueinander. Weitere Erläuterungen siehe Abb. 9.

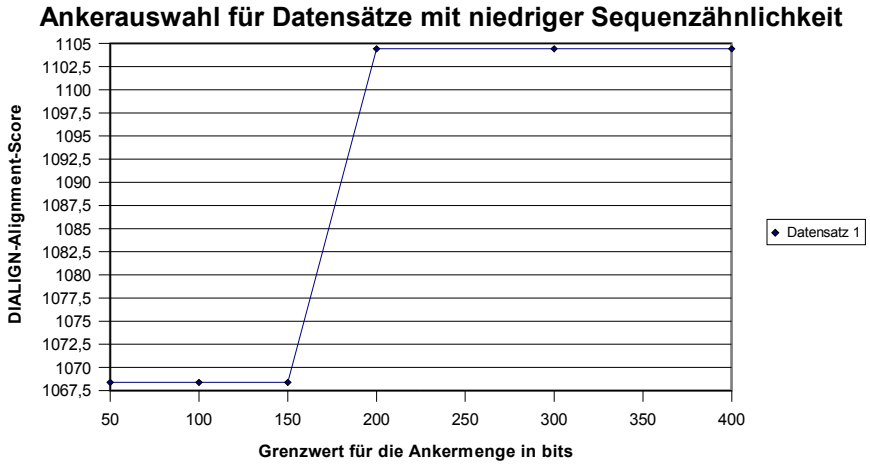

Abb. 12 Qualität der Alignments bei niedriger Sequenzverwandtschaft

In der Abbildung wird die einzelne Qualität des von DIALIGN erstellten Output Alignments mit den verschiedenen Ankermengen für den jeweiligen Datensatz dargestellt. Die Datensätze beinhalten jeweils Sequenzen mit niedriger Sequenzverwandtschaft zueinander. Weitere Erläuterungen siehe Abb. 9.

7. Annotierte Daten

7.1. *ENSEMBL* Datenbank

Die Datenbank *ENSEMBL* [4] (http://www.ensembl.org/) ist ein Bioinformatikprojekt, um mit den biologischen Informationen die Reihenfolge der Genome zu organisieren. *ENSEMBL* ist eine Datenbank, welche die Annotation der einzelnen Genome und die orthologen Verhältnisse zwischen ihnen beinhaltet [4].

Es stehen unter anderem gesamte Chromosomen von unterschiedlichen Spezies in diversen Formaten zum Download bereit. Es ist zudem möglich, beliebige DNA – Sequenzabschnitte von vorliegenden Genomen mit zugehöriger Annotation herunter zu laden. In der vorliegenden Arbeit wurden die Datensätze, die gleich vorgestellt werden sollen, ausschließlich von der *ENSEMBL* Datenbank bezogen. Die Annotationen zu den Datensätzen sind ebenfalls von *ENSEMBL*.

7.2. Verwendete Datensätze

In diesem Kapitel werden nun sechs unterschiedliche Datensätze vorgestellt, die repräsentativ für alle verwendeten Datensätze sind. Alle zu den jeweiligen Datensätzen gehörenden Sequenzen sind DNA-Sequenzen und liegen im *fasta*-Format vor. Ein Datensatz beinhaltet jeweils Sequenzen von unterschiedlichen Spezies. Mit Ausnahme des letzten Datensatzes, der zwei ganze Chromosomen von *Saccharomyces cerevisiä* enthält. Die Datensätze sind in zwei Kategorien unterteilt. Die erste Kategorie ist die der multiplen Alignments. Dieser Kategorie gehören die ersten beiden Datensätze an. Die übrigen vier Datensätze gehören der zweiten Kategorie an, die des paarweisen Alignments. Diese Kategorien zeigen Unterschiede bezüglich der Sequenzlänge auf. Die Berechnung von multiplen Alignments ist sehr rechenintensiv, deshalb sind die einzelnen Sequenzen dieser Kategorie nicht so lang (Sequenzlänge < 100000 bp), wie die der zweiten Kategorie. Ein wichtiger Punkt ist die Sequenzverwandtschaft der unterschiedlichen Sequenzen in einem Datensatz zueinander. Dabei wurden Datensätze mit hohem Verwandtschaftsgrad, wie z.B. Datensatz 3 (Mensch-Schimpanse) und Datensätze mit geringem Verwandtschaftsgrad, wie Datensatz 5 (Mensch-Ratte) verwendet. Die Sequenzen sind jedoch nicht nach Sequenzverwandtschaft ausgesucht worden und damit nicht nach der Spezies aus denen sie stammen, sondern nach orthologen Genen, die sie beherbergen. So wurde im ersten Datensatz das Gen Myoglobin in dem 22 Chromosom des Menschen lokalisiert und anschließend mittels *ENSEMBL* die Vorkommen dieses Gens in anderen Spezies gesucht. Alle

Datensätze wurden nach unterschiedlichen orthologen Genen zusammengestellt. In diesem Zusammenhang wurde auf den prozentuale Anteil der Exons auf die Länge der Sequenz geachtet. Die Datensätze sind so zusammengestellt worden, dass der Anteil in einigen höher als in anderen ist. Diese Vorgehensweise wurde gewählt, da es bei der Zusammenstellung der Datensätze, primär um die Bewertung der Ankerpunkte ging und diese vorwiegend anhand der Exons vorgenommen wurde.

Bei der Auswahl von Sequenzen mit wenigen Exons wurde folglich darauf geachtet, ein Gen zu verwenden, dass aus möglichst wenig Exons besteht. Gegebenenfalls wurden die Enden dieser Sequenzen ausgedehnt, um nicht kodierende Bereiche in die Sequenz mit aufzunehmen. Exons können auf dem Vorwärts- oder auf dem Rückwärtssrtrang liegen. Die *ENSEMBL*- Daten liegen alle in dem „Vorwärtsstrang-Format“ vor. Teilweise musste mittels eines Perlskriptes das inverse Komplement gebildet werden, um den Rückwärtsstrang der DNA- Sequenz zu erzeugen. Die Informationen über Anzahl und Lage der Exons in den Sequenzen sind unter andern Informationen in dem sogenannten *flat*-File(siehe Anhang viii) enthalten. In der folgenden Tabelle werden die Datensätze genauer vorgestellt, wobei unter anderem die Lage der Sequenzen in der jeweiligen Spezies angegeben wird.

Daten-satznr.	*Involvierte Seq.*	*Spezies*	*Lokalität der Sequenz im Organismus*	*Sequenz-länge (bp)*	*Anzahl der Exons laut Annotation*
1	1	Huhn	Chr. 1	3375	3
	2	Hund	Chr. 10	8907	6
	3	Mensch	Chr. 22	16591	5
	4	Maus	Chr. 15	35181	9
2	1	Mensch	Chr. 19	41348	33
	2	Schimpanse	Chr. 20	38063	30
	3	Kuh	Chr. 5	36140	37
	4	Ratte	Chr. 7	51749	34
3	1	Schimpanse	Chr. X	200000	8
	2	Mensch	Chr. Y	200000	24
4	1	Mensch	Chr. 1	100000	114
	2	Maus	Chr. 4	100000	120
5	1	Ratte	Chr. 20	3534667	1824
	2	Mensch	Chr. 21	4545503	1890
6	1	S_cerevsisae	Chr. 7	1090946	593 Gene
	2	S_cerevsisae	Chr. 15	1091287	606 Gene

Tabelle 4: Verwendete Datensätze
Die Tabelle zeigt die in dieser Arbeit verwendeten Datensätze. Die „Lokalität der Sequenz im Organismus“ gibt an, aus welchem Chromosom des betreffenden Organismus die jeweiligen Sequenzen stammen. Fuer die Sequenzen des letzten Datensatzes ist die Anzahl der Exons unbekannt. Stattdessen ist die Anzahl der Gene bekannt.

8. Diskussion der Ergebnisse

Dieses Kapitel behandelt die Bewertung der Qualität der Alignments, die von *DIALIGN* unter Verwendung der von *TBLASTX* zur Verfügung gestellten Ankerpunkte berechnet werden. Zudem wird die Zeit, die *TBLASTX-DIALIGN* (siehe Kapitel 6.1) für die Erstellung des Output-Alignments inklusive der Berechnung und Formatierung der Ankerpunkte benötigt, bestimmt.

Die Qualität eines von *TBLASTX-DIALIGN* berechneten Alignments wird anhand des *DIALIGN*-Alignment-Scores (siehe Kapitel 4.1.2) bewertet. Die von *DIALIGN* berechneten Output-Alignments ohne die Verwendung von Ankerpunkten werden ebenfalls mit dem *DIALIGN*-Alignment-Score bewertet. Damit können die von *DIALGN* (ohne Ankerpunkte) und die von *TBLASTX-DIALIGN* berechneten Alignments anhand des *DIALIGN*-Alignment-Scores miteinander verglichen werden. Um eine weitere Vergleichsmöglichkeit zu haben, werden die von *CHAOS-DIALIGN* (siehe Kapitel 4.3.1) berechneten Alignments mit denen von *TBLASTX-DIALIGN* berechneten Alignments verglichen. Zum Vergleich der von den drei Methoden zu den einzelnen Datensätzen jeweils berechneten Alignments wurde das Visualisierungstools *ABC* [12] verwendet. Mit Sequenzen sind im Folgenden immer DNA-Sequenzen gemeint. Die Ähnlichkeit der einzelnen Sequenzen zueinander wird mit *ABC* graphisch dargestellt. In der *ENSEMBL*-Annotation sind die genauen Sequenzabschnitte notiert, in denen sich die Exons der jeweiligen Gene befinden. Die in der *ENSEMBL*-Annotation nicht aufgeführten Bereiche zwischen den Exons eines Gens, werden im Folgenden als Introns bezeichnet. Es werden die von den drei Methoden berechneten Output-Alignments mittels *ABC* jeweils überprüft, ob Exons eines Gens G mit Exons des zu G orthologen Gens in einer anderen Sequenz aligniert wurden. Die Exons die miteinander alignt werden, müssen aus orthologen Genen stammen. Ansonsten ist ein lokales Alignment zwischen den Exons biologisch falsch. Außerdem wird überprüft, ob Introns eines Gens nur mit Introns eines orthologen Gens alignt wurden. Außer der Bewertung der Qualität der von *TBLASTX-DIALIGN* berechneten Alignments wird das Laufzeitverhalten der Methode in Bezug zu dem Laufzeitverhalten von *DIALIGN* und *CHAOS-DIALIGN* gesetzt.

Im Folgenden werden die unterschiedlichen Datensätze jeweils mit *CHAOS-DIALIGN*, *TBLASTX-DIALIGN* und *DIALIGN* aligniert. Dabei wird jeweils dieselbe *DIALIGN* Version benutzt. Zudem wird der *DIALIGN*-Aufruf jeweils mit folgenden Parametern, die im Kapitel 4.1.3 näher erläutert sind, durchgeführt:

./dialign2-2 -lgs -ta -fa -col_score <Datensatz>

Werden *DIALIGN* bei der Programmausführung Ankern übergeben, wird der Parameter „-anc“ der eben beschriebenen Menge von Parametern hinzugefügt. *CHAOS* und *TBALSTX* werden mit ihren Default-Werten ausgeführt. Bei der Selektion einer Teilmenge der Menge von lokalen Alignments, die von *TBLASTX* berechnet wurden, wird ein Schwellwert von 150 bits verwendet (siehe Kapitel 6.3). Diese Teilmenge wird *DIALIGN* als Ankermenge übergeben. Die Ergebnisse der Alignmentmethoden mit den einzelnen Datensätzen (siehe Kapitel 7) werden im Folgenden diskutiert. Im ersten Datensatz werden von den drei Alignmentmethoden multiple Alignments erzeugt, wobei die Sequenzen durchschnittlich 16013,5 bp lang sind. Die im Datensatz 1 enthaltenen Sequenzen haben eine geringe Verwandtschaft zueinander. Außerdem befinden sich nur sehr wenige Exons in den jeweiligen Sequenzen. Bei dem Vergleich der von den drei Methoden berechneten Alignments gibt es Differenzen bei den Alignmentlängen. Jedoch zeigt die graphische Darstellung der einzelnen Alignments, dass die Exons bei den unterschiedlichen Alignments in den gleichen Bereichen im Alignment liegen. Der *DIALIGN*-Alignment-Score der drei berechneten Alignments zum Datensatz 1 zeigt keine großen Differenzen auf. Der *DIALIGN*-Alignment-Score für das von *TBLASTX-DIALIGN* berechnete Alignment beträgt 98,18% von dem von *DIALIGN* berechneten Score. Damit ist es vom numerischen Score her qualitativ fast gleichwertig. Der *DIALIGN*-Alignment-Score von *CHAOS-DIALIGN* liegt zwischen den Scores der beiden anderen Alignmentmethoden (siehe Abb. 13).

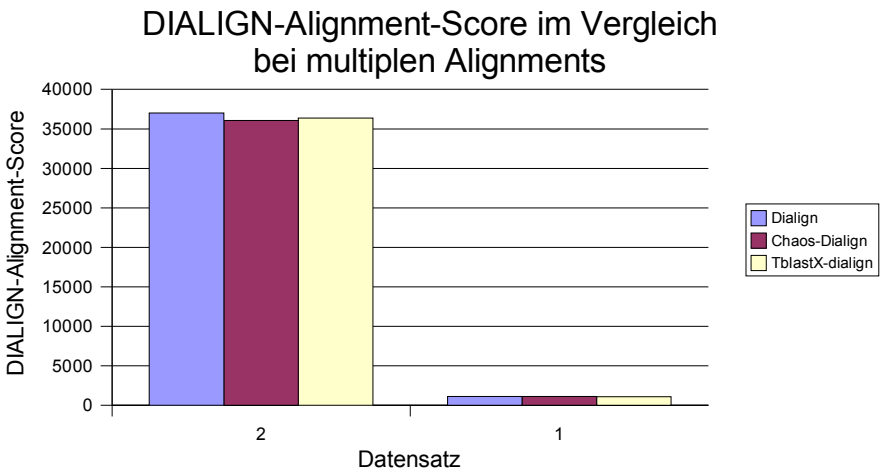

Abb. 13 Vergleich des DIALGN-Alignment-Scores der drei Alignmentmethoden bei multiplen Alignments

Die Abbildung zeigt den Vergleich der „Qualität" der von den Alignmentmethoden berechneten multiplen Alignments. Die Sequenzen von Datensatz 1 haben eine niedrige und die von Datensatz 2 eine hohe Sequenzähnlichkeit zueinander.

Weitere Erläuterungen zu den Datensätzen siehe Kapitel 7.

Die Relation der drei Alignmentmethoden mit Datensatz 1 untereinander sind repräsentativ für alle Datensätze in Hinblick auf ihr Laufzeitverhalten, wobei *TBLASTX-DIALIGN* und *CHAOS-DIALIGN* zeitlich beieinander liegen und *DIALIGN* deutlich länger für die Berechnung des globalen Alignments braucht (siehe Abb.14). Die einzelnen Ergebnisse zu Datensatz 1 sind in Tabelle 5 dargestellt.

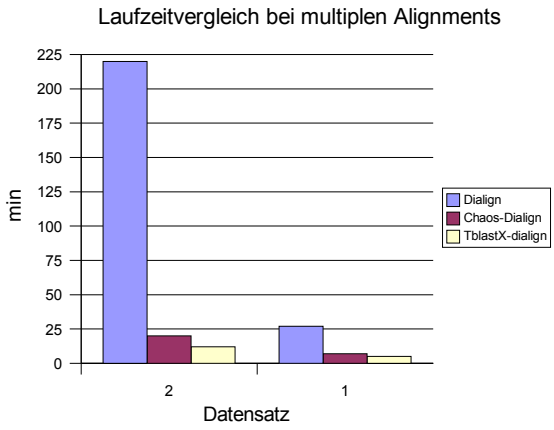

Abb. 14 : Laufzeitverhalten bei multiplen Alignments

Es wird das Laufzeitverhalten von den drei oben beschriebenen Alignmentmethoden bei multiplen Alignments mit hoher (Datensatz 2) und niedriger (Datensatz 1) Sequenzähnlichkeit dargestellt. Die angegebene Laufzeit bezieht sich auf die gesamte Zeit, die zur Erstellung des globalen Output-Alignment von DIALIGN benötigt wurde, inklusive der Zeit zur Berechnung und Formatierung der Ankerpunkte, insofern DIALIGN welche übergeben bekommt.

Datensatz 1			
	DIALIGN	***CHAOS-DIALIGN***	***TBLASTX-DIALIGN***
Alignmentlänge in bp	40722	43341	47785
Laufzeit	27 min	7 min	5 min
DIALIGN-Alignment-Score	1104,42	1094,66	1068.38
Gesamtscore der nicht konsistenten Fragmente	17,25	14,34	2,53
Anzahl verwendeter Anker	-	55	83

Tabelle 5 : Resultate der Programmasuführungen von den drei Alignmentmethoden mit Datensatz 1
Die Tabelle zeigt die Ergebnisse der drei Alignmentmethoden DIALIGN, CHAOS-DIALIGN und TBLASTX-DIALIGN mit dem Datensatz 1 (siehe Kapitel 7). Die angegebene Laufzeit bezieht sich auf die gesamte Zeit, die zur Erstellung des globalen Output-Alignment von DIALIGN benötigt wurde, inklusive der Zeit zur Berechnung und Formatierung der Ankerpunkte, insofern DIALIGN welche übergeben bekommt. Der Gesamtscore der nicht konsistenten Fragmente wird in Kapitel 4.1.2 erläutert.

Im Datensatz 2 haben die Sequenzen eine durchschnittliche Länge von 41825 bp, wobei die Verwandtschaft der Sequenzen untereinander enger ist, als bei Datensatz 1. Trotzdem sind die Relationen der Ergebnisse der drei Programme mit Datensatz 2 zu den Relationen der Ergebnisse von Datensatz 1 hinsichtlich Alignmentlänge, Laufzeit und *DIALIGN*-Alignment-Score ähnlich (siehe Tabelle 6). Der *DIALGN*-Alignment-Score des Alignments, das von *TBLASTX-DIALIGN* mit dem

Datensatz 2 berechnet wurde, beträgt 36355,37 und der von *CHAOS-DIALIGN* erzeugte 36079,37 (siehe Abb. 13). Damit ist der *DIALGN*-Alignment-Score des von *TBLASTX-DIALIGN* berechneten Alignments größer. Die Alignmentlänge des von *TBLASTX-DIALIGN* berechneten Alignments ist fast identisch mit der Länge des von *DIALIGN* berechneten Alignments. Es werden von *CHAOS* 767 Anker berechnet. Bei einer durchschnittlichen Sequenzlänge des zweiten Datensatzes von 41825 bp wurde durchschnittlich alle 54 Nukleotide ein Anker gefunden. Von *TBLASTX* wurden 5325 Anker gefunden. Damit wurde durchschnittlich alle 8 Nukleotide ein Anker gefunden. Nur 4% der von *TBLASTX* gefundenen Anker sind inkonsistent. Damit scheinen die von *TBLASTX* berechneten Anker eine hohe „Qualität" zu haben. Mittels *ABC* wird die Lage der Anker in den Exons der entsprechenden orthologen Gene vorgefunden. Teilweise wurden auch Introns zweier Sequenzen, die in orthologen Genen liegen, miteinander alignt.

Datensatz 2			
	DIALIGN	***CHAOS-DIALIGN***	***TBLASTX-DIALIGN***
Alignmentlänge in bp	67220	69548	67218
Laufzeit	220 min	20 min	12 min
DIALIGN-Alignment-Score	37028,61	36079,37	36355,37
Gesamtscore der nicht konsistenten Fragmente	279,77	304,68	362,63
Anzahl verwendeter Anker	-	767	5325

Tabelle 6 : Resultate der Programmausführungen von den drei Alignmentprogrammen mit Datensatz 2 Erläuterung siehe Tabelle 5

Die drei Alignmentmethoden werden jeweils mit den Datensätzen 3 und 4 ausgeführt. Die jeweils berechneten Outputalignments sind paarweise Alignments. Die Sequenzen im Datensatz 3 sind beide 200000 bp lang, haben eine enge Sequenzverwandtschaft zueinander und wenige Exons in den Sequenzen.Dabei sind die Sequenzen im Datensatz 4 beide 100000 bp lang, haben eine entferntere Sequenzverwandtschaft als die Sequenzen in Datensatz 3 und viele Exons in den Sequenzen (siehe Kapitel 7). Die Ergebnisse der Alignmentprogramme mit Datensatz 3 widersprechen den oben diskutierten Ergebnissen von Datensatz 1 und 2 nicht (siehe Tabelle 7). Der *DIALIGN*-Alignment-Score für das von *TBLASTX-DIALIGN* berechnete Alignment mit dem dritten Datensatz beträgt 97,04 % des von *DIALIGN* berechneten Scores. Für die Berechnung des Alignments benötigt *TBLASTX-DIALIGN* mit 9 Minuten die geringste Zeit von den drei Alignmentmethoden (siehe Abb.15).

Datensatz 3			
	DIALIGN	***CHAOS-DIALIGN***	***TBLASTX-DIALIGN***
Alignmentlänge in bp	252276	254516	253512
Laufzeit	89 min	15	9 min
DIALIGN-Alignment-Score	43214,71	39973,06	41934,81
Gesamtscore der nicht konsistenten Fragmente	0	0	0
Anzahl verwendeter Anker	-	329	8592

Tabelle 7 : Resultate der Programmasuführungen von den drei Alignmentprogrammen mit Datensatz 2

Erläuterung siehe Tabelle 5

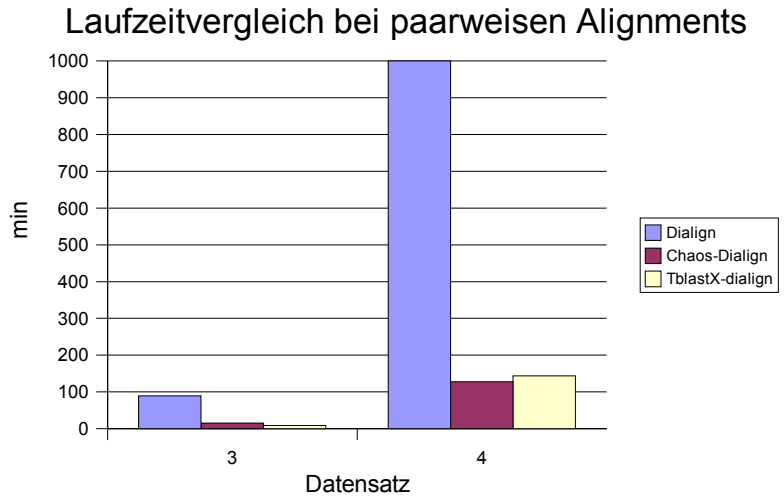

Abb. 15 Laufzeitvergleich von paarweisen Alignments

Es wird das Laufzeitverhalten von den drei oben beschriebenen Alignmentmethoden bei paarweisen Alignments mit hoher (Datensatz 3) und mit niedriger(Datensatz 4) Sequenzähnlichkeit dargestellt. DIALIGN hat für die Berechnung von Datensatz vier 11520 Minuten gebraucht, es werden in der Abbildung nur 1000 Minuten angezeigt. Weitere Erläuterungen siehe Abb. 14.

Bei den Alignments, die von den verschiedenen Methoden mit dem Datensatz 4 erzeugt wurden, gab es große Unterschiede in der jeweiligen Alignmentlänge. Die größte Längendifferenz weist das von *TBLASTX* berechnete Alignment mit einer Differenz von 247554 bp zu der Länge des von *DIALIGN* berechneten Alignments mit diesem Datensatz auf. Bei allen durchgeführten Testdurchläufen ist im Datensatz 4 die größte Abweichung zwischen dem *DIALIGN*-Alignment-Score des von *TBLASTX*-

DIALIGN und des von *DIALIGN* berechneten Alignments. Der *DIALIGN*-Alignment-Score für das von *TBLASTX-DIALIGN* berechnete Alignment mit dem vierten Datensatz beträgt 90,4 % des von *DIALIGN* berechneten Scores.

Datensatz 4			
	DIALIGN	**CHAOS-DIALIGN**	**TBLASTX-DIALIGN**
Alignmentlänge in bp	1331341	1259247	1578995
Laufzeit	11520	128	144
DIALIGN-Alignment-Score	10975,79	9982,09	9922,05
Gesamtscore der nicht konsistenten Fragmente	0	0	0
Anzahl verwendeter Anker	-	2035	18686

Tabelle 8 : Resultate der Programmasuführungen von den drei Alignmentprogrammen mit Datensatz 2
Erläuterung siehe Tabelle 5

Der *DIALGN*-Alignment-Score des Alignments, das von *TBLASTX-DIALIGN* mit dem Datensatz 3 erzeugt wurde, beträgt 41934,81und der von *CHAOS-DIALIGN* erzeugte 39973,06. Damit ist der *DIALGN*-Alignment-Score des von *TBLASTX-DIALIGN* berechneten Alignments um 1961,75 größer. Bei dem vierten Datensatz verhält es sich umgekehrt (siehe Abb. 16). Dabei ist es vermutlich nicht ausschlaggebend, ob es sich um multiple oder paarweise Alignments handelt, sondern vielmehr kommt es auf die Sequenzähnlichkeit und auf die Anzahl der gefundenen Anker in den Sequenzen an. Beziehungsweise kommt es auf die Anzahl der Anker in den unterschiedlichen Sequenzen mit niedriger Sequenzähnlichkeit an. Die von *TBLASTX* berechneten lokalen Alignments zu dem Datensatz 4 liegen überwiegend in den Exons der jeweiligen orthologen Gene der Sequenzen, die an dem Anker beteiligt sind. 10 % der von *TBLASTX* berechneten Anker liegen jedoch nicht in zwei orthologen Genen, sondern in zwei unterschiedlichen Genen.

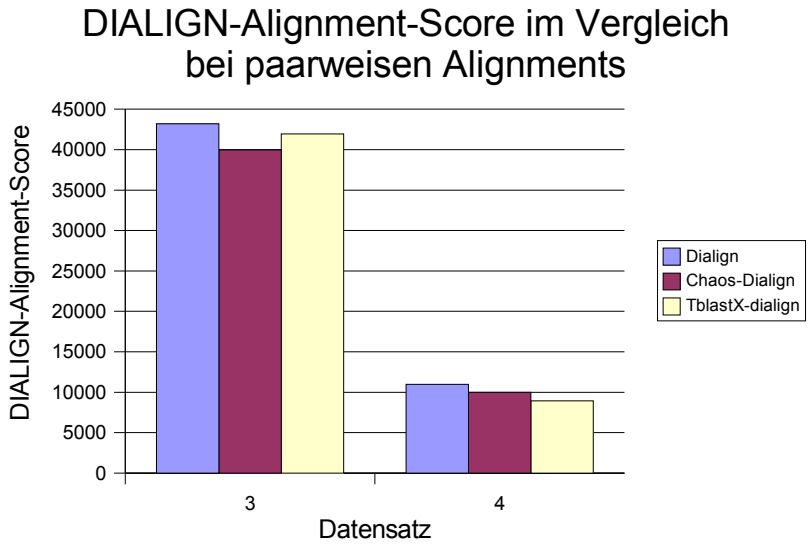

Abb. 16 DIALGN-Alignment-Score im Vergleich bei paarweisen Alignments

Vergleich der „Qualität" der von den Alignmentmethoden berechneten paarweisen Alignments. Datensatz 4 hat eine niedrige und Datensatz 3 eine hohe Sequenzähnlichkeit. Weitere Erläuterungen siehe Abb. 13.

Eine mögliche Erklärung für die vorliegenden Ergebnisse der Berechnungen von *TBLASTX-DIALIGN* mit dem Datensatz 4 ist die folgende.
Die Sequenzunterschiede von Exons auf Proteinebene in demselben Organismus können sehr gering sein. Liegt eine entfernte Sequenzverwandtschaft vor, unterscheiden sich die Sequenabschnitte der Exons eines orthologen Gens mehr voneinander als die Sequenzabschnitte zweier Exons unterschiedlicher Gene eines Organismus. Dadurch kann es zu falschen lokalen Alignments kommen. Man könnte den Schwellwert für die Selektion der Anker bei Sequenzen mit entfernter Verwandtschaft zueinander erhöhen. Der Schwellwert soll jedoch für Sequenzkombinationen aus allen Spezies verwendet werden.
Unter Verwendung eines Schwellwertes von 150 bits wurden von *TBASTX-DIALIGN* mit den Datensätzen 1-4 globale Alignments erzeugt, die zu den von *DIALIGN* erzeugten Alignments qualitativ gleichwertig waren. Die von *TBLASTX-DIALIGN* berechneten Alignments hatten durchschnittlich 95,6 % des *DIALGN*-Alignment-Scores des von *DIALGN* berechneten Alignments.

Bei Datensatz 5 und 6 konnte der *DIALIGN*-Alignment-Score von *DIALIGN* und *TBLASTX-DIALIGN*

nicht in Bezug gesetzt werden, weil *DIALIGN* nach zwei Wochen Zeit für die Berechnung des Outputalignments noch keine Ergebnisse geliefert hatte. Deswegen wurden die Berechnungen aus zeitlichen Gründen abgebrochen. Die Ergebnisse zu der jeweiligen Programmausführung von *TBLASTX-DIALIGN* mit den Datensätzen 5 und 6 sind in der Tabelle 8 dargestellt.
Die tabellarisierten Ergebnisse widersprechen den zu den Datensätzen 1 bis 4 diskutierten Ergebnissen nicht.

TBLASTX-DIALIGN		
	Datensatz 5	**Datensatz 6**
Alignmentlänge in bp	5796849	1859522
Laufzeit in Minuten	12660	250
DIALIGN-Alignment-Score	24193,28	12130,43
Gesamtscore der nicht konsistenten Fragmente	0	0
Anzahl verwendeter Anker	8775	22605

Tabelle 8 : TBLASTX-DIALIGN berechnet Alignments
Die Tabelle beinhatet die jeweiligen Ergebnisse zu den von TBLASTX-DIALIGN berechneten paarweisen Alignments mit den Datensätzen 5 und 6. Datensatz 6 beinhaltet Sequenzen mit einer jeweiligen Länge von 1 Millionen bp. Im Datensatz 5 haben die jeweiligen Sequenzen eine Länge von mehr als 3 Millionen bp. Weitere Erläuterungen siehe Tabelle 5.

9.Zusammenfassung

Die vorliegende Arbeit behandelt unter anderem die Implementierung einer Schnittstelle zwischen *TBLASTX* und *DIALGN*. Die an *DIALIGN* von *TBLASTX* übergebenen Anker sollen dabei möglichst nicht die biologische Qualität des von *DIALIGN* zu erzeugenden Sequenzalignments reduzieren, jedoch die Laufzeit optimieren. Es wurde anhand von annotierten Daten der *ENSEMBL*-Datenbank ein Schwellwert von 150 bits ermittelt. Diesen Wert müssen die von *TBLASTX* berechneten lokalen Alignments mindestens haben, um als Anker an *DIALIGN* übergeben zu werden. Die von *DIALIGN* unter Berücksichtigung der Ankerpunkte erzeugten Alignments sind zu den von *DIALGN* erzeugten Alignments ohne Anker mit den verwendeten Datensätzen qualitativ gleichwertig. Außerdem wird die Laufzeit von *DIALGN* durch die übergebenen Anker optimiert.

10. Fazit und Ausblick

Die vorliegenden Ergebnisse belegen, dass *TBLASTX-DIALIGN* eingesetzt werden kann, um im Vergleich zu *DIALIGN* das globale Sequenzalignment in kürzerer Zeit zu berechnen.

Die von *TBASTX-DIALIGN* berechneten Alignments und die von *DIALIGN* berechneten Alignments mit den verwendeten *ENSEMBL*-Datensätzen haben durchschnittlich eine Abweichung von 4,4 % des *DIALIGN*-Alignment-Score von einander. Damit sind die von *TBASTX-DIALIGN* berechneten Alignments zu den von *DIALIGN* berechneten Alignments gleichwertig.

Die Methode *TBLASTX-DIALIGN* kann als eine Alternative zu *CHAOS-DIALIGN* für die Berechnung von Sequenzalignments genomischer Größe verwendet werden.

11. Anhang

```
>11 dna:chromosome chromosome:NCBIM33:11:32195640:32196444:1
ATCTTTGAGCTCAGCCAGGGGGCAGAGCGCAAGGCTCAGTTCATTGAAGATGGCTCGGTC
CCAGGATGATCAGTGGCTGGTCCTTGCGCTCTGGAAGAAGATGGGCAGCAATGTCGGAAT
CTACACGACCGAGGCCTTGGAGAGGTACGCCCTCGGGTTCATCTTCCAACGGACTGCGGG
GCGAAGACGGGCTCTC...

>14 dna:chromosome chromosome:WASHUC1:14:13958598:13960030:1
ATGGCACTGACCCAAGCTGAGAAGGCTGCCGTGACCACCATCTGGGCAAAGGTGGCTACC
CAGATTGAGTCCATTGGGCTGGAATCACT...
```

Abbildung i : fasta-Format

```
> [1,60] + human region
GCTGGGCACAGTGGCTCACACCTATAATCCCAGCACTTTGGGAAACTGAGGTGGGTGGAT
> [1,60] + chimp region
GCTGGGCACGGTGGCTCACACCTATAATCCCAGCACTTTGGGAAACTGAGGTGGGCGGAT
> [61,120] + human region
CACCTGAGGTCTGGAGTTTGAGACCAGCGTGGCCAACATGGTGAAACCCCGTCTCTACTA
> [61,120] + chimp region
CACCTGAGGTTTGGAGTTTGAGACCAGCGTGGCCAACATGGTGAAACCCCGTCTCTACTA
...
```

Abbildung ii : avx-Format

```
1 2 33611 29954 9 1053
1 2 33635 29978 3 1053
1 2 33647 29990 3 1053
1 2 32344 28687 36 1020
1 2 32353 28696 90 1020
1 2 32446 28789 48 1020
1 2 32554 28897 66 1020
1 2 32563 28906 111 1020
...
```

Abbildung iii : .anc-Format für DIALIGN

Jede Zeile repräsentiert einen Anker. Die erste Spalte steht für die erste Sequenz und die Zweite für die zweite Sequenz die an dem Anker beteiligt sind. Die dritte Spalte steht für die Startposition des Ankers in der ersten Sequenz und die vierte Spalte analog für die zweite Sequenz. Die fünfte Spalte repräsentiert die Länge und die Sechste das Gewicht des Ankerpunktes.

```
>MMLV
pdadhtwYTDGSSLLQEGQRKAGAAVTTETeviwaKALDAG---T---SA
QRAELIALTQALKMAEgkk-LNVYTDSRYAFATAHIHGEIYRRRGLLTSE
GKEIKNKDE---ILALLKALFLPKRLSIIHCPGHQ------KGHSAEARG
NRMADQAARKAAITETPDTStll----------------------------
----------
>HEPB
rpglcQVFADAT------PTGWGLVMGHQRMR---GTFSAPLPIHt----
--AELLAACFArsrsgan---IIGTDN-----------------------
-------------SVVLSR--------------KYTSFPWLLGCAANWI-
LRGTSFVYVPSALNPADDPSrgrlglsrpllrlpfrpttgrtslyadsps
vpshlpdrvh
>ECOL
mlkqvEIFTDGSCLGNPGPGGYGAILRYRGRE---KTFSAGytrT---TN
NRMELMAAIVALEALKEHCEVILSTDSQYVRQGITQWIHNWKKRGWKTAD
KKPVKNVDlwqrLDAALGQ--------------HQIKWEWVKGHAGHPE-
NERCDELARAAAMNPTledtgyqvev----------------------
----------
```

Abbildung iv : Alignment im fasta-Format

Das im Beispiel dargestellte Alignment ist ein Protein-Alignment.

```
C   9
S  -1  4
T  -1  1  5
P  -3 -1 -1  7
A   0  1  0 -1  4
G  -3  0 -2 -2  0  6
N  -3  1  0 -2 -2  0  6
D  -3  0 -1 -1 -2 -1  1  6
E  -4  0 -1 -1 -1 -2  0  2  5
Q  -3  0 -1 -1 -1 -2  0  0  2  5
H  -3 -1 -2 -2 -2 -2  1 -1  0  0  8
R  -3 -1 -1 -2 -1 -2  0 -2  0  1  0  5
K  -3  0 -1 -1 -1 -2  0 -1  1  1 -1  2  5
M  -1 -1 -1 -2 -1 -3 -2 -3 -2  0 -2 -1 -1  5
I  -1 -2 -1 -3 -1 -4 -3 -3 -3 -3 -3 -3 -3  1  4
L  -1 -2 -1 -3 -1 -4 -3 -4 -3 -2 -3 -2 -2  2  2  4
V  -1 -2  0 -2  0 -3 -3 -3 -2 -2 -3 -3 -2  1  3  1  4
F  -2 -2 -2 -4 -2 -3 -3 -3 -3 -3 -1 -3 -3  0  0  0 -1  6
Y  -2 -2 -2 -3 -2 -3 -2 -3 -2 -1  2 -2 -2 -1 -1 -1 -1  3  7
W  -2 -3 -2 -4 -3 -2 -4 -4 -3 -2 -2 -3 -3 -1 -3 -2 -3  1  2 11
   C  S  T  P  A  G  N  D  E  Q  H  R  K  M  I  L  V  F  Y  W
```

Abbildung v : BLOSUM62 - Matrix

```
Query= 1
         (3375 letters)

Database: 1
           3 sequences; 60,679 total letters

Searching.done

                                                                    Score     E
Sequences producing significant alignments:                         (bits) Value

2                                                                        162   7e-70
3                                                                        157   2e-67
4                                                                        146   5e-64

>2
          Length = 8907

 Score =  162 bits (348), Expect(4) = 7e-70
 Identities = 67/86 (77%), Positives = 74/86 (86%)
 Frame = +1 / +2


Query: 2071 SASVHRLFHDHPETLDRFDKFKGLKTPDQMKGSEDLKKHGATVLTQLGKILKQKGNHESE 2250
          S   HRLF +HPETLD+FDKFK LKT D+MKGSEDLKKHG TVLT LG ILK+KG+HE+E
Sbjct: 4958 SPCAHRLFKNHPETLDKFDKFKHLKTEDEMKGSEDLKKHGNTVLTALGGILKKKGHHEAE 5137


Query: 2251 LKPLAQTHATKHKIPVKYLEVWKRTG 2328
          LKPLAQ+HATKHKIPVKYLEV  R G
Sbjct: 5138 LKPLAQSHATKHKIPVKYLEVGGRAG 5215


 Score =  122 bits (262), Expect(4) = 1e-55
 Identities = 57/86 (66%), Positives = 61/86 (70%)
 Frame = -2 / -1


Query: 2327 PVLFHTSRYLTGILCFVAWV*ARGFSSDS*LPFCFRILPSWVRTVAPCFFRSSEPFI*SG 2148
          P L  TSRYLTGILC VA   ARGFSS S* PF  R+ P  V TV PCFFRSSEPFI S
Sbjct: 5214 PALPPTSRYLTGILCLVACDWARGFSSAS*CPFFLRMPPRAVSTVLPCFFRSSEPFISSS 5035
```

Abbildung vi : TBLASTX-Output Format

```
ID   22     standard; DNA; HTG; 16591 BP.
XX
AC   chromosome:NCBI35:22:34327311:34343901:1
XX
SV   chromosome:NCBI35:22:34327311:34343901:1
XX
DT   26-JUL-2005
XX
DE   Homo sapiens chromosome 22 NCBI35 partial sequence 34327311..34343901
DE   annotated by Ensembl
XX
KW   .
XX
OS   Homo sapiens (Human)
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia;
..       ..       ..       .
..       ..       ..       .
..       ..       ..       .
FT   exon             complement(10599..10745)
FT                     /note="exon_id=ENSE00001402406"
FT   exon             complement(4121..4343)
FT                     /note="exon_id=ENSE00000653544"
FT   exon             complement(1..680)
FT                     /note="exon_id=ENSE00001106915"
FT   exon             complement(16428..16591)
FT                     /note="exon_id=ENSE00001413187"
FT   exon             complement(10400..10502)
```

Abbildung vii : flat-File von ENSEMBL

Dieser File beinhaltet die Lage der Exons in der jeweiligen Sequenz.

```
seq_len:   3534667   4545503
sequences:   20          21

   1) seq:   1   2  beg: 1118482 1501189 len:  51 wgt:  73.15 olw:  73.15 it: 1 cons
   2) seq:   1   2  beg: 1736800 2193545 len:  84 wgt:  46.01 olw:  46.01 it: 1 cons
   3) seq:   1   2  beg: 3505447 4447882 len:  90 wgt:  45.83 olw:  45.83 it: 1 cons
   4) seq:   1   2  beg: 2427044 3056173 len:  90 wgt:  45.83 olw:  45.83 it: 1 cons
   5) seq:   1   2  beg: 2999923 3810427 len:  90 wgt:  45.31 olw:  45.31 it: 1 cons
   6) seq:   1   2  beg: 2985637 3794478 len:  90 wgt:  42.22 olw:  42.22 it: 1 cons
   7) seq:   1   2  beg: 3496211 4437500 len:  87 wgt:  41.46 olw:  41.46 it: 1 cons
   8) seq:   1   2  beg:  751681  950650 len:  90 wgt:  41.21 olw:  41.21 it: 1 cons
   9) seq:   1   2  beg:  270166  309459 len:  90 wgt:  41.21 olw:  41.21 it: 1 cons
  10) seq:   1   2  beg:  607397  694353 len:  81 wgt:  40.52 olw:  40.52 it: 1 cons
  11) seq:   1   2  beg: 3518888 4524824 len:  90 wgt:  40.20 olw:  40.20 it: 1 cons
  12) seq:   1   2  beg: 1004618 1301723 len:  90 wgt:  39.70 olw:  39.70 it: 1 cons
  13) seq:   1   2  beg: 2447731 3085165 len:  90 wgt:  39.70 olw:  39.70 it: 1 cons
  14) seq:   1   2  beg: 3493763 4435108 len:  90 wgt:  39.70 olw:  39.70 it: 1 cons
  15) seq:   1   2  beg: 2482023 3145994 len:  87 wgt:  39.42 olw:  39.42 it: 1 cons
  16) seq:   1   2  beg:  602965  689682 len:  90 wgt:  39.20 olw:  39.20 it: 1 cons
  17) seq:   1   2  beg: 1559792 1999868 len:  90 wgt:  39.20 olw:  39.20 it: 1 cons
  18) seq:   1   2  beg: 1703565 2171587 len:  90 wgt:  39.20 olw:  39.20 it: 1 cons
  19) seq:   1   2  beg: 3245377 4099980 len:  90 wgt:  39.20 olw:  39.20 it: 1 cons
  20) seq:   1   2  beg: 1804620 2298413 len:  90 wgt:  38.20 olw:  38.20 it: 1 cons
```

Abbildung viii : .frg-Format des DIALIGN Outputs für Fragmente

```
   Aligned sequences:          length:
   ==================          =======

   1) X                       200001
   2) Y                       200001

   Average seq. length:       200001.0


   Please note that only upper-case letters are considered to be aligned.


   Alignment (DIALIGN format):
   ===========================

X                  1   cggccgctcg gcctcaccct cacatgcccg cggtaccacc tcggctgctc
Y                  1   cagaggggcc tgtggcggcc aggtcatcca ctttaaatcc tcctcagtac

                       0000000000 0000000000 0000000000 0000000000 0000000000

X                 51   cacgtcccac gcttgacacg caccttggcc accaggccca tgtcgtccat
Y                 51   gtgtacttta aaaagaactt cgtaaagagc ccgtcaccca aagcgcactg

                       0000000000 0000000000 0000000000 0000000000 0000000000

X                101   ggtggccctc tcgtgggggа accgggcgaa ggtgctctcg atcttgctga
Y                101   ataaaggcga caccctgact cccaacaagc tatttctgtt gagggcactg

                       0000000000 0000000000 0000000000 0000000000 0000000000

X                151   tgacggcatc cacgttgacg gagtcctgcg ggcgtcctct ggggaagtag
Y                151   agaaggcagc tccctgactc atcacaattc cagaagtcac agatacatgt

                       0000000000 0000000000 0000000000 0000000000 0000000000
```

Abbildung ix : DIALIGN-Format für das Output-Alignment

12. Literatur

[1] B. Morgenstern (1999) „*DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment*“. Bioinformatics 15, 211-218

[2] M.Brudno (2004), „*The CHAOS/DIALIGN WWW server for multiple alignment of genomic sequences*“, Nucleic Acids Research, Vol. 32,W41-W44, http://nar.oxfordjournals.org/ MI.

[3] SF Altschul (1990), „*Basic local alignment search tool*“, J Mol Biol.;215:403–410.

[4] T. Hubbard (2005), „*Ensembl 2005*“, Nucleic Acids Research, Vol. 33, Database issue D447–D453

[5] M. Stanke (2004); „*Algorithmen der Bioinformatik II Teile der Genvorhersage und Sequenzierung und Assemblierung“,* http://gobics.de/mario/AlgBioInfII/skript.pdf

[6] N. Bray (2003), „*AVID: A Global Alignment Program“.* Genome Res., 13:97.

[7] S. B. Needleman (1970), J. Mol. Biol. 48, 443–458.

[8] M. Brudno (2003), „*LAGAN and Multi-LAGAN: Efficient Tools for Large-Scale Multiple Alignment of Genomic DNA*“, Genome Res., 13(4): 721-731.

[9] B. Morgenstern (1996), „*Multiple DNA and protein sequence alignment based on segment-to-segment comparison*“, Proc. Natl. Acad. Sci. USA Vol. 93, pp. 12098-12103

[10] S. Henikoff (1992), „*Amino acid substitution matrices from protein blocks*“ Proc. Natl. Acad. Sci. USA 89: 10915–10919.

[11] B. Morgenstern (1998), „*Segment-based scores for pairwise and multiple sequence alignments*“, Proceedings ISMB'98, pp. 115-121

[12] G. M. Cooper (2004), „*ABC: software for interactive browsing of genomic multiple sequence alignment data*“, BMC Bioinformatics 5:192

[13] M. Brudno (2003), „F*ast and sensitive multiple alignment of large genomic sequences*“, BMC Bioinformatics , 4:66

[14] Stephen F. Altschul (1997), „*Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*“, Nucleic Acids Research, Vol. 25, No. 17, 3389–3402

[15] R. Durbin (1998), „*Biological sequence analysis*“, Cambrige University S.33ff

[16] formatdb, NCBI, http://www.ncbi.nlm.nih.gov/Class/BLAST/README.bls

[17] The Bioperl Project, http://bio.perl.org/