Albert-Ludwigs-Universität Freiburg
Lehrstuhl für Bioinformatik
Prof. Dr. Rolf Backofen

Alignmentverbesserungen mit Hilfe von Consensus-Dotplots

Masterarbeit

Erstgutachter: Prof. Dr. Rolf Backofen,
Lehrstuhl für Bioinformatik
Zweitgutachter: Junior-Prof. Dr. Olaf Ronneberger,
Image Analysis Group

Betreuerin:
Sita Lange, M.Sc.

von
Benjamin Schulz
schulzb@informatik.uni-freiburg.de
21. Juni 2011 - 27. Dezember 2011

Danksagungen

Professor Dr. Rolf Backofen
Ich danke Ihnen für die Möglichkeit an ihrem Lehrstuhl meine Masterarbeit schreiben zu dürfen und für dieses sehr interessante Thema.

Junior-Prof. Dr. Olaf Ronneberger
Ich danke Dir, dass du die Aufgabe des zweiten Gutachters übernommen hast.

Sita Lange
Vielen Dank für die Betreuung meiner Masterarbeit.

Sebastian Will
Vielen Dank für die nette Diskussionrunde zu Benchmark-Methoden.

Steffen Heyne
Danke für die hilfreichen Tipps und Ratschläge sowie die „last-minute“-Anmerkungen zu meiner Masterarbeit.

Inhaltsverzeichnis

1 Zusammenfassung

RNA-Alignments sind ein wichtiges Thema der Bioinformatik. Ein großes Problem ist, wenn Alignments Schwächen in Bezug auf das korrekte Alignment der Strukturen aufweisen, denn es hat sich gezeigt, dass die Sekundärstruktur deutlich stärker als die Basensequenz konserviert ist zwischen verwandten RNA-Sequenzen. Ein Alignment in dem die Strukturen zusammenpassend aligniert sind, ist damit sehr wichtig. In dieser Arbeit wurde ein Programm zur schnellen Verbesserung von in Bezug auf die Struktur minder qualitativen multiplen Alignments entwickelt. MAIC (=**M**ultiple-**A**lignment **I**mprovment through **C**onsensus-dotplots) führt, basierend auf einem Vergleich der Sequenz-Dotplots mit dem Consensus-Dotplot des Alignments, kleine Veränderungen durch, mit dem Ziel eine höhere Übereinstimmung zwischen den Dotplots zu erzeugen. Sequenz-Dotplots enthalten die Basenpaarwahrscheinlichkeiten einer einzelnen RNA-Sequenz, während Consensus-Dotplots die gemittelten Basenpaarwahrscheinlichkeiten aller Sequenzen eines Alignments enthalten.
Die durchgeführten Veränderungen stellen gemeinsame Strukturen der Sequenzen stärker heraus, was im Allgemeinen mit einer Erhöhung der Qualität des Alignments einher geht.

In diesem Zusammenhang werden zwei neue Methoden der Alignmentbewertung vorgestellt:
Die „Intrakorrelation“ basiert auf der Korrelation der Sequenz-Dotplots zum Consensus-Dotplot. Sie bewertet damit die Übereinstimmung der möglichen Sekundärstrukturen der Sequenzen innerhalb eines Alignments.
Die „Interkorrelation“ basiert ebenfalls auf der Korrelation von Dotplots, vergleicht jedoch zwei verschiedene Alignments der selben Sequenzen. Sie dient dem Vergleich eines berechneten Alignments zu einem Referenz-Alignment.

Die zwei neuen Bewertungsmethoden wurden zusammen mit einer Reihe verbreiteter Benchmark-Methoden für einen Vergleich aktueller multiple-Alignment-Programme benutzt.

Es konnte gezeigt werden, dass die Intrakorrelation ein gutes Maß für die Qualität eines Alignments ist. MAIC kann bei nur geringem Zeitaufwand strukturell schlechte Alignments, wie sie von reinen Sequenz-Alignment-Programmen erzeugt werden und auch teilweise in den Rfam-seed-Alignments vertreten sind, deutlich verbessern. Sequenz-Struktur-Alignment-Programme dagegen produzieren bereits strukturell hochwertige Alignments, die von MAIC kaum noch weiter verbessert werden können. Im Vergleich der Programme untereinander konnte sich mLocARNA als der eindeutige Sieger im Bezug auf alle Benchmark-Methoden profilieren.

2 English Abstract

RNA-alignments are an important topic in recent bioinformatics. A big problem is, when alignments are not perfectly aligned in reference to the structural similarity of the sequences. The secondary structure is much more conserved than the base sequence in many RNA-families, therefore it is quiet important to align the structure correctly. In this thesis, I present an algorithm to quickly improve the quality of multiple sequence alignments with respect of the aligned structure: MAIC (=**M**ultiple-**A**lignment **I**mprovment through **C**onsensus-dotplots). The approach is based on a comparison between the sequence-dotplots and the consensus-dotplot of the alignment. Sequence-dotplots contain the basepairprobabilities of a single RNA-Sequence. Consensus-dotplots contain the averaged probabilities of a complete alignment. MAIC increases the congruency between these dotplots through small changes in the alignment. These changes should improve the visibility of common structures and so increase the overall quality of the alignment.

Additionally, two new methods to rate the quality of an alignment are presented: The „intracorrelation“ is based on the correlation between the dotplots of the sequences and the consensus-dotplot.
The „intercorrelation“ rates an alignment based on the correlation between its consensus-dotplot and the consensus-dotplot of a reference-alignment.

These two methods, plus three well established benchmarking methods were used to compare recent multiple-alignment programs.

The results are showing, that the intracorrelation is a good measurement for the quality of alignments. MAIC can improve structurally bad alignments, which are for example produced by pure sequence alignment programs and occur in Rfam, with only small time cost. The alignments of sequence-structure-alignment programs have already a quiet high quality so that MAIC is not needed. In the comparison of the different programs, mLocARNA achieved the highest score in all benchmarks.

3 Einleitung

3.1 Bioinformatik

Lange Zeit wurde angenommen, dass Ribonukleinsäure (kurz RNA vom Englischen **r**ibo-**n**ucleic-**a**cid) hauptsächlich als temporärer Informationsspeicher dient, der, als Kopie der in der DNA gespeicherten Gene, die Grundlage für die Erzeugung von Proteinen ist. In jüngerer Zeit wurde aber immer klarer, dass RNA noch deutlich mehr Aufgaben in der Zelle übernimmt[19]. Neben den Proteinen übernehmen auch RNA-Sequenzen wichtige regulatorische und katalytische Aufgaben in der Zelle. Seither ist das Feld der RNA-Forschung massiv gewachsen. Die Bioinformatik versucht dabei unter anderem, die in der Zelle vorgehenden Prozesse, wie die Faltung der RNA-Moleküle, am Computer nachzubilden. Da dies gelingt, ist es möglich, vollautomatisch verschiedene RNA-Moleküle zu vergleichen und zu kategorisieren.

3.2 RNA

Ribonukleinsäure ist ein aus bis zu mehreren tausend Nukleotiden bestehendes Kettenmolekül mit wichtigen Aufgaben in der Zelle. Die Nukleotide werden durch ihre (Nukleo-)Base unterschieden. Vier verschiedene Basen können in der RNA vertreten sein: Adenin(A), Cytosin(C), Guanin(G) und Uracil(U). Unterschieden werden RNA-Moleküle durch die Abfolge dieser Basen in der Kette. Ein Nukleotid hat zwei Bindungstellen, an denen das Molekül eine Bindung mit anderen Nukleotiden eingehen kann. Nach der Position der Kohlenstoffatome im Molekül werden diese Bindungspositionen mit 3' und 5' bezeichnet. Wenn sich zwei Nukleotide verknüpfen um eine Kette zu bilden, verbindet sich immer eine 3' Position mit einer 5' Position. Durch diese eindeutige Orientierung kann einem RNA-Molekül eine Richtung zugewiesen werden (nämlich von 5' nach 3'), in der es zu lesen ist.

3.2.1 Nicht-kodierende RNA

Es gibt zwei große Gruppen von RNAs: Die kodierende RNA und die nicht-kodierende RNA. „Kodierend“ wird die RNA genannt, wenn sie als Bauplan für ein Protein fungiert. Diese Arbeit

konzentriert sich aber auf „nicht-kodierende“ RNA (kurz ncRNA vom Englischen „non-coding“). Diese enthält keinen Bauplan für ein Protein, sondern ist ein aktiver Bestandteil der Prozesse in der Zelle. Insbesondere in Bezug auf die Regulation der zellulären Prozesse fällt der ncRNA eine wichtige Aufgabe zu[19]. Studien haben gezeigt, dass rund 97%-98% der in den Zellen von Eukarioten erzeugten RNA nicht für Proteine kodiert. Es ist allerdings noch unklar, wie groß der Anteil dieser nicht-kodierenden RNA ist, der tatsächlich in irgend einer Weise eine Aufgabe hat, und wieviel davon nur funktionslose Reste sind, die bei anderen Prozessen anfallen[19].

3.2.2 RNA Struktur

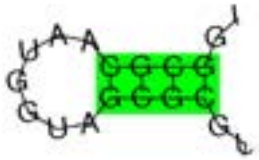
Nicht-kodierende RNA liegt im Allgemeinen in gefalteter Form in der Zelle. Man unterscheidet verschiedene Strukturebenen: Die Primärstruktur beschreibt die Abfolge der Basen im RNA-Molekül. Die Sekundärstruktur bezieht sich auf die zweidimensionale Faltung des Moleküls. Sie entsteht, weil sich Wasserstoffbrücken zwischen den Basen der Sequenz ausbilden. Dabei gibt es eine Tendenz zu bestimmten festen Paarungen: Adenin paart mit Uracil, Cytosin mit Guanin und etwas weniger oft auch Guanin mit Uracil. Andere Paarungen kommen nur selten vor. Die energetisch optimale Struktur nennt sich MFE-Struktur (MFE = minimale freie Energie). Viele RNA-Familien definieren sich durch ihre Sekundärstruktur. Ein bekanntes Beispiel hierfür ist die tRNA, die ein essentieller Bestandteil im Herstellungsprozess von Proteinen ist. Sie liegt meist in einer sogenannten Kleeblattstruktur (ähnlich Grafik 3.1) vor. Die Relevanz der Sekundärstruktur für die Funktion der RNA-Sequenz hat zur Folge, dass die Sekundärstruktur auch über große evolutionäre Distanz konserviert bleibt, während die Basensequenz (Primärstruktur) durch Mutationen bereits signifikante Unterschiede aufweisen kann[10].

Abbildung 3.1: Sekundärstruktur einer RNA-Sequenz

Es gibt verschiedene Möglichkeiten diese Faltung zu veranschaulichen. Eine weit verbreitete Methode ist die „Dot-Bracket“-Schreibweise. Dabei werden durch öffnende und schließende Klammern die Basenpaare identifiziert und Punkte zeigen ungepaarte Basen an. Die Struktur aus Grafik 3.1 würde in dieser Schreibweise folgendermaßen aussehen:

```
GGAUUCGUGGCGCAAUGGUAGCGCGUCUGACUCCAGAUCAGAAGGUUGCGUGUUCGAUUCACGUCGGGUUCA
(((((((..((((.......)))).((((((....).)))))).....((((.......))))))))))).
```

Wenn mehrere Basenpaare direkt hintereinander liegen, stabilisieren sie sich gegenseitig. Eine solche Struktur wird „Stem“ genannt. Ein Stem sieht zum Beispiel wie in diesem Auschnitt aus Grafik 3.1 aus:



Es existiert auch eine Tertiärstruktur, die die dreidimensionale Struktur des Moleküls beschreibt. Da ein Großteil der (heute bekannten) Funktion der RNA bereits durch die Primär- und insbesondere durch die Sekundärstruktur definiert ist, wird die Tertiärstruktur in dieser Arbeit nicht betrachtet. Im Allgemeinen ist sie sehr aufwändig zu berechnen.

3.2.3 Sekundärstrukturvorhersage

Die Sekundärstrukturvorhersage ist ein wichtiges Thema der Bioinformatik. Diverse Algorithmen wurden bereits entwickelt um diese Aufgabe zu erfüllen. Der Nussinov Algorithmus[23] ist einer der ersten Algorithmen um die Sekundärstruktur einer RNA-Sequenz zu berechnen. Er war noch recht einfach gehalten und versuchte schlicht die Struktur mit der maximalen Anzahl an Basenpaaren zu finden. Seither wurden viele Experimente durchgeführt um für verschiedene Teilstrukturen deren energetische Eigenschaften festzustellen. Basierend auf diesen Daten konnten komplexere Algorithmen, wie der Zuker Algorithmus[37] entwickelt werden, die nun nichtmehr einfach die Basenpaare zählen, sondern die Struktur finden, die energetisch optimal für diese Sequenz ist. Allerdings wurde erkannt, dass die energetisch optimale Struktur nicht immer die Struktur ist, in die sich die Sequenz in der Zelle faltet. Der nächste Entwicklungsschritt war daher ein Algorithmus, der nichtmehr eine Struktur berechnet, sondern für alle möglichen Basenpaare der Sequenz Wahrscheinlichkeiten angibt, mit der diese Basen gepaart sind. Dies kann zum Beispiel der McCaskill-Algorithmus[20]. McCaskill geht davon aus, dass die Energien der möglichen Strukturen einer Sequenz Boltzmann-verteilt sind. Dementsprechend verteilt er Gewichte für Teil-Strukturen. Damit ist es möglich alle Gewichte der Strukturen aufzusummieren, die ein Basenpaar enthalten und durch die Summe aller Gewichte zu teilen. Dieser Wert entspricht dann der Wahrscheinlichkeit dieses Basenpaars im Strukturensemble. Diese Basenpaarwahrscheinlichkeiten können in sogenannten Dotplots gespeichert werden (Kapitel 3.2.4), auf deren Basis dann auch suboptimale Strukturen identifiziert werden können.

Alle angesprochenen Faltungsalgorithmen schränken allerdings die möglichen Sekundärstrukturen dahingehend ein, dass keine sogenannten Pseudoknots erlaubt sind. Dies sind Strukturen, in

denen sich Basenpaare kreuzen. Eine solche Struktur ist in Grafik 3.2 zu sehen. Strukturen mit (beliebigen) Pseudoknots zu berechnen fällt in die Klasse der NP-vollständigen Probleme und ist damit derzeit nicht effizient lösbar. Die Einschränkung auf pseudoknot-freie Strukturen senkt die Problemkomplexität auf polynomielle Laufzeit.

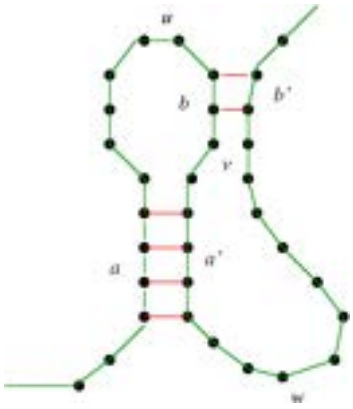


Abbildung 3.2: Teilstruktur mit einem einfachen Pseudoknot[26]

Die vorgestellten Algorithmen wurden in verschiedenen Programmen implementiert. In dieser Arbeit wurde RNAfold[13] aus dem ViennaRNAPackage[12] verwendet. Dieses Programm basiert auf dem Zuker- und dem McCaskill-Algorithmus um die MFE-Struktur und die Basenpaarwahrscheinlichkeiten zu berechnen.

3.2.4 Sequenz-Dotplots

In dieser Arbeit wird als Repräsentation der möglichen Sekundärstrukturen einer RNA-Sequenz ein sogenannter „Dotplot“ benutzt. Das ist eine diagonal geteilte quadratische Matrix, mit Werten zwischen 0 und 1. Ein Beispiel für einen Dotplot ist in Grafik 3.3 zu sehen. An den Kanten der Matrix wird die RNA-Sequenz abgetragen. In der linken unteren Hälfte der quadratischen Matrix wird die MFE-Struktur eingetragen. Für jedes Basenpaar in der MFE-Struktur wird ein Punkt an der entsprechenden Stelle in der Matrix eingezeichnet. Im Dreieck rechts oben werden die Basenpaarwahrscheinlichkeiten eingetragen. Jeder Punkt zeigt für das durch Spalte und Zeile definierte Paar von Basen die Wahrscheinlichkeit an, dass sie gepaart sind. Je größer der Punkt, desto höher ist die Wahrscheinlichkeit. In der graphischen Repräsentation wird meist die Wurzel der Wahrscheinlichkeit als Kantenlänge eines Punktes genommen, damit entspricht die Fläche des Punkts der Wahrscheinlichkeit.

Dotplots haben gegenüber der Dot-Bracket Schreibweise insbesondere den Vorteil, dass ein Dotplot nicht nur die eine optimale Struktur anzeigt, sondern über die Wahrscheinlichkeiten auch suboptimale Faltungen abgelesen werden können. Eine solcher Hinweis auf eine suboptimale

Faltung ist in Grafik 3.3 in grün markiert. Dieser Stem ist kein Teil der MFE-Struktur, die relativ großen Punkte zeigen aber an, dass Strukturen, die diesen Stem enthalten, ebenfalls recht wahrscheinlich sind. Eine, im Vergleich zur MFE-Struktur, energetisch nicht wesentlich schlechtere Faltung existiert hier also.

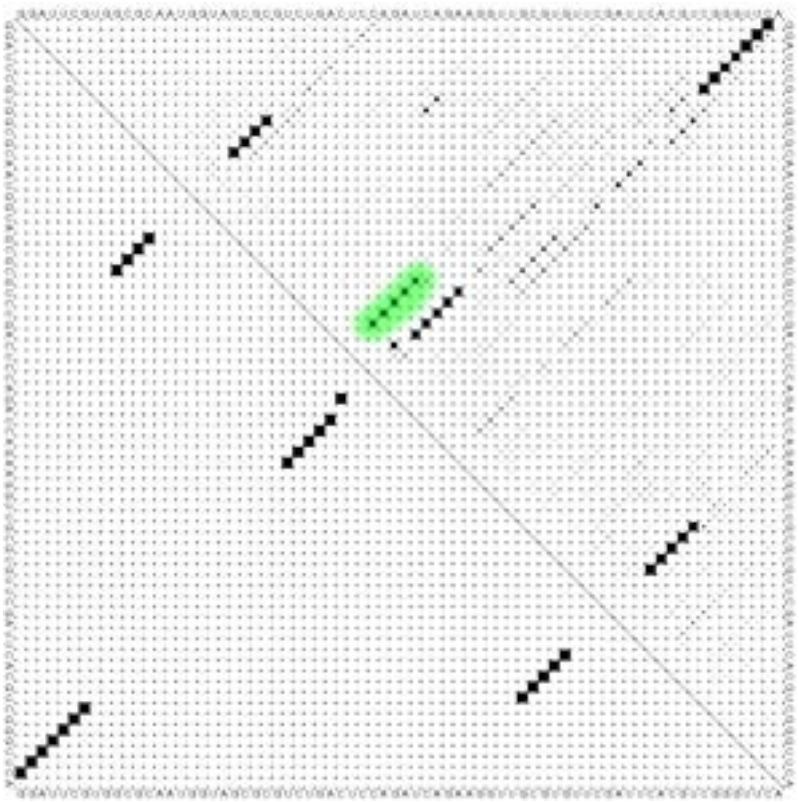


Abbildung 3.3: Dotplot der Sequenz aus Grafik 3.1

3.3 Alignments

Ein Alignment ist ein in der Bioinformatik häufig benutztes Verfahren, um Kettenmoleküle wie DNA, RNA und Proteine auf ihre sequenzielle und/oder strukturelle Ähnlichkeit hin zu vergleichen. Dies kann unter anderem benutzt werden, um evolutionäre Beziehungen zwischen RNA-Sequenzen verschiedener Tiere zu finden.

Bei paarweisen Alignments wird versucht zwei (RNA-)Sequenzen durch Einfügung von Gaps (englisch für Lücke) so zu verändern, dass gleiche oder verwandte Basen an den selben Stellen im Alignment liegen, beziehungsweise unter Berücksichtigung der Sekundärstruktur, dass Basenpaare von Strukturen, die in beiden Sequenzen auftauchen, an den selben Stellen im Alignment sind. Ein mögliches sequentielles Alignment könnte z.B. so aussehen:

```
---UAUU---GGGGG-UUGCCUUUAUGA-
UGGUAUUCCAGGG--CAUGCCUGUAUGAG
```

Bei multiple-Alignments, also Alignments, die aus mehr als zwei Sequenzen bestehen, hat man das Problem, dass die Spaltenbewertung zunehmend kompliziert wird. Außerdem ist eine Lösung mit dynamischer Programmierung dem Problem unterworfen, dass sich Laufzeit und Platzbedarf mit der Anzahl der Sequenzen potenziert: Die Komplexitätsklasse liegt in $O(n^k)$ wobei n die Länge des Alignments und k die Anzahl der Sequenzen ist. Damit lassen sich nur sehr wenige und nur kurze Sequenzen alignieren.

Die in dieser Arbeit benutzten multiple-Alignment-Programme, mit Ausnahme des in dieser Arbeit entwickelten Programms, setzten auf progressive Alignment-Algorithmen, mit deren Hilfe das Problem auf paarweise Alignments zurück geführt werden kann: Es werden Alignments aller Sequenzpaare berechnet und so (evolutionäre) Distanzen zwischen den Sequenzen ermittelt. Mit deren Hilfe kann ein (phylogenetischer) Baum erstellt werden, in dem ähnliche Sequenzen dicht zusammen sind und unterschiedliche Sequenzen weiter auseinander liegen. Entlang dieses Baumes werden dann Alignments berechnet, wobei das an einem Knoten aus den Subalignments entstehende neue Alignment im Allgemeinen entsprechend dem am besten passenden Paar von Sequenzen zusammen gebaut wird. Die Idee zu diesem Vorgehen stammt aus [6]. Die verschiedenen Programme unterscheiden sich hauptsächlich in der Art, wie genau die paarweisen Alignments durchgeführt werden und welche und wie aufwendige Vor- und Nachverarbeitungsschritte durchgeführt werden, um das Ergebnis zu optimieren.

Eine wichtige Frage ist, wie ein optimales Alignment aussieht. Es gibt keine allgemeingültigen Kriterien, anhand deren man fundiert sagen könnte, ob ein Alignment besser ist als das andere. Aus paarweisen Vergleichen nahe verwandter Sequenzen wurden verschiedene Mutationstabellen ermittelt, die einen Ansatz dafür bieten, wie wahrscheinlich Mutationen bestimmter Basen oder Basenpaare sind. Auf diesen basierend arbeiten die Alignment-Programme und versuchen Basen zu alignieren, die entweder gleich sind oder mit relativ hoher Wahrscheinlichkeit durch Mutationen ineinander übergehen. Abgesehen von der Basenübereinstimmung sollte aber auch die Faltung der Sequenzen möglichst so aligniert sein, dass Basenpaare, die in mehreren Sequenzen auftreten in den selben Spalten im Alignment stehen. Fraglich ist dann, was getan werden soll, wenn die Struktur ein Alignment vorschreibt, welches eine deutlich geringere Basenübereinstimmung besitzt, als andere alternative Alignments.

Im Rahmen dieser Arbeit kamen verschiedene Alignment-Programme zur Anwendung:

3.3.1 Sequenz-Alignment Methoden

Sequenz-Alignment-Programme betrachten nur die Basenfolge der (RNA-)Sequenzen und versuchen eine möglichst hohe Übereinstimmung in den Alignment-Spalten zu erzielen.

ClustalW [30] aligniert eine fast beliebig hohe Anzahl an Sequenzen mit Hilfe eines progressiven Alignments.

Mafft [15] benutzt ebenfalls ein progressives Alignment, im Gegensatz zu ClustalW beinhaltet Mafft aber auch einen iterativen Verarbeitungsschritt, der nach dem progressiven Alignment durchgeführt wird. Dabei wird versucht, die Schwächen eines progressiven Alignments auszugleichen. Solch eine Schwäche ist zum Beispiel, dass es eine Tendenz zu Alignment-Spalten gibt, in denen fast alle Sequenzen ein Gap enthalten. Durch iteratives Entfernen eines Teils der Sequenzen und neu-Alignierung dieser Sequenzen an das Alignment kann dieses Problem abgeschwächt werden.

3.3.2 Sequenz-Struktur-Alignment Methoden

Sequenz-Struktur-Alignment-Programme betrachten nicht nur die Basenfolge, sondern auch die Sekundärstruktur der (RNA-)Sequenzen. In dieser Arbeit wurden folgende Programme benutzt:

LARA [2] wurde im „Department of Mathematics and Computer Science“ der Freien Universität Berlin entwickelt. Paarweise Alignments werden durch einen graphenbasierten Algorithmus berechnet. Für multiple Alignments ruft LARA T-COFFEE[22] auf, an das Informationen über alle paarweisen Alignments übergeben werden.

mCARNA [24] wurde in einer Zusammenarbeit des Lehrstuhl für Bioinformatik der Universität Freiburg, dem Computation and Biology Lab, CSAIL, MIT, Cambridge und dem „Dipartimento di Matematica“ der Università Parma entwickelt. mCARNA ist ein constraintbasierter Algorithmus, mit dem Ziel Sequenzen mit beliebig hoher Faltungskomplexität alignieren zu können. Im Gegensatz zu den anderen Programmen ist mCARNA allerdings noch in einer frühen Prototypenphase und ist noch nicht veröffentlicht. Die in späteren Kapiteln gezeigten Ergebnisse von mCARNA spiegeln daher nicht die Leistungen des Programms wider, wenn es veröffentlicht wird, sondern dienen eher den Entwicklern als Feedback, wo sich das Programm momentan im Vergleich mit anderen Programmen einordnet.

mLocARNA [34] wurde am Lehrstuhl für Bioinformatik der Universität Freiburg entwickelt. Es basiert auf einem Sankoff-ähnlichen Algorithmus[28], mit RNAfold Dotplots als Basis für die möglichen Faltungen.

RAF [4] wurde im „Computer Science Department“ der Universität Stanford in Kalifornien entwickelt. Es berechnet intern zuerst nur sequenzbasierte Alignments, um eine Matrix über Alignment-Spalten-Wahrscheinlichkeiten zu bekommen. Diese Matrix wird um alle Werte bereinigt, die unterhalb eines Limits liegen, und dann als Basis für die eigentlichen Sequenz-Struktur-Alignments benutzt. Dieses Vorgehen hat den Vorteil, dass der Zeitaufwand für das Sequenz-Struktur-Alignment deutlich geringer ist, weil deutlich weniger mögliche Alignierungen betrachtet werden.

3.3.3 Consensus-Dotplot

Zu einem Alignment kann ein sogenannter „Consensus-Dotplot“ berechnet werden. Dies ist ein Dotplot, der nicht die Faltung einer einzelnen Sequenz und deren MFE-Struktur enthält, sondern die Faltungen und die Struktur des Alignments als Ganzes. Um die in dieser Arbeit verwendete Variante eines Consensus-Dotplots zu erzeugen, müssen zuerst die Dotplots der Sequenzen im Alignment zu sogenannten Gapped-Dotplots erweitert werden. Dafür werden an den Positionen der Sequenz, an denen sie einen Gap im Alignment enthält, leere Spalten und Zeilen in den Dotplot eingefügt. Da nun alle Dotplots gleich groß sind, können die Wahrscheinlichkeiten im Consensus-Dotplot durch Mitteln nach Formel 3.1 über die Sequenz-Dotplots berechnet werden:

$$\forall(i,j)(j<i, 0<i<\text{Alignment-Länge}) : DP(i,j) = \sqrt{\frac{1}{|Seq|}\sum_{s\in Seq}(DP_s(i,j))^2} \tag{3.1}$$

Meistens werden in Dotplots nicht die Basenpaarwahrscheinlichkeiten direkt gespeichert, sondern es wird die Wurzel der Wahrscheinlichkeiten genommen. Damit entsprechen die Werte in der Matrix der Kantenlänge der in der graphischen Repräsentation des Dotplots eingezeichneten Punkte. Für die Berechnung der Wahrscheinlichkeiten im Consensus-Dotplot müssen die Werte daher erst quadriert werden, bevor sie zusammengerechnet werden können.

Die MFE-Struktur im linken unteren Teil des Consensus-Dotplots wird in dieser Arbeit mit RNAalifold[3] berechnet; sie wird entsprechend dem Namenszusatz des Dotplots „Consensus-Struktur“ genannt. Wenn in der weiteren Arbeit von Consensus-Dotplots gesprochen wird, bezieht sich dies immer nur auf die Wahrscheinlichkeiten im rechten oberen Dreieck, nicht auf die Consensus-Struktur.

3.4 Dynamische Programmierung

Das in dieser Arbeit entwickelte Programm, sowie alle anderen benutzten multiple-Alignment-Programme basieren auf sogenannter dynamischer Programmierung, um paarweise Alignments zu berechnen. Dynamische Programmierung ist ein Ansatz zum lösen komplexer Probleme durch Aufspaltung der Aufgabe in kleine Teilprobleme. Aus den Lösungen der Teilprobleme kann dann die Lösung des eigentlichen Problem zusammengebaut werden. Idealerweise können für die Lösung Teilprobleme mehrfach verwendet werden, wodurch gegenüber einem naiven Lösungsweg Rechenaufwand eingespart werden kann.

Ein einfacher auf dynamischer Programmierung basierender Sequenz-Alignment-Algorithmus ist der Needleman-Wunsch-Algorithmus[21]. Er soll hier als Beispiel dienen, um den Ansatz der dynamischen Programmierung bei Alignments zu veranschaulichen. Der Algorithmus berechnet

einen Vergleichswert, der angibt, wie ähnlich zwei Sequenzen, bezogen auf die Basenübereinstimmung, sind. Die erzeugten Alignments maximieren die Sequenzübereinstimmung unter der Einschränkung, nicht zu viele Lücken einzufügen.

Zwei Sequenzen, a und b, sollen aligniert werden. Es wird eine Funktion w benötigt, die für zwei Zeichen aus den Sequenzen einen Ähnlichkeitswert angibt. Needleman-Wunsch benutzt hier eine binäre Funktion, die bei identischen Zeichen 1 zurück gibt und sonst -1. Außerdem wird eine Gapkosten-Funktion benötigt, die definiert, wie teuer es ist, einen Gap in eine der Sequenzen einzubauen. Der allgemeine Needleman-Wunsch-Algorithmus benutzt eine Funktion, die abhängig von der Länge des Gaps Kosten definiert. In diesem Beispiel wird darauf verzichtet und stattdessen die Kosten pro Gap einfach auf $g = -1$ gesetzt.

Zuerst wird eine Matrix M generiert, die $n+1$ Zeilen und $m+1$ Spalten hat, wobei n die Länge von a ist und m die Länge von b. Die erste Spalte und die erste Zeile werden mit Hilfe der Gapkosten g initialisiert:

$$\text{Initialisierung: } \forall i(0 < i \leq n) : M(i,0) = g \cdot i; \forall j(0 < j \leq m) : M(0,j) = g \cdot j$$

	-	**A**	**A**	**C**	**G**
-	0	-1	-2	-3	-4
A	-1				
C	-2				
A	-3				
C	-4				
G	-5				

Der Rest der Matrix wird mit Hilfe folgender Rekursionsgleichung gefüllt:

$$A(i,j) = \max \begin{cases} M(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ M(i-1,j) + g & \text{Gap in Sequenz } b \\ M(i,j-1) + g & \text{Gap in Sequenz } a \end{cases}$$

$$w(c,d) = \begin{cases} 1 & c = d \\ -1 & \text{sonst} \end{cases}$$

$$g = -1$$

Um den Wert eines Felds in der Matrix zu berechnen wird also die beste von drei Möglichkeiten genommen, entweder zwei Zeichen werden aligniert (Match/Mismatch) oder es wird ein Gap in die eine oder in die zweite Sequenz eingefügt.

	-	A	A	C	G
-	0	-1	-2	-3	-4
A	-1	1	0	-1	-2
C	-2	0	0	1	0
A	-3	-1	1	0	0
C	-4	-2	0	2	1
G	-5	-3	-1	1	3

Wenn die Matrix gefüllt ist, wird mit Hilfe eines Backtracking-Algorithmus das optimale Alignment aus der Matrix ausgelesen. Der Algorithmus sucht nach dem Berechnungspfad auf dem aus dem Wert in Feld $(0,0)$ das Ergebnis im rechten untersten Feld berechnet wurde. Ein solcher Pfad existiert, da jedes Feld immer in Abhängigkeit vom relativ dazu linken, oberen oder linken oberen Feld berechnet wurde. Das Backtracking beginnt im rechten untersten Feld der Matrix und läuft entlang des Pfades, wie die Felder berechnet wurden, zurück zum Feld $(0,0)$. Je nach dem welche der drei Rekursionsregeln für die Berechnung des Felds angewendet wurde, wird eine Spalte mit zwei alignierten Basen oder eine Spalte mit einem Gap in einer der zwei Sequenzen und in der anderen eine Base ausgegeben.

Das optimale Alignment aus obiger Matrix sähe im Beispiel dann so aus, wobei es von rechts nach links zusammengebaut wurde:

```
A-ACG
ACACG
```

Theoretisch kann ein solcher Ansatz auch für das Alignment von mehr als zwei Sequenzen benutzt werden. Allerdings wächst die nötige Matrix mit jeder Sequenz um eine Dimension und mit der Matrix wächst auch die Anzahl der zu unterscheidenden Fälle in der Rekursionsformel und die Funktion w muss um Fälle für mehr als zwei Basen erweitert werden. Rechenaufwand und Platbedarf steigen damit zwangsläufig sehr schnell an, weswegen andere Ansätze für das Alignment von mehr als zwei Sequenzen benutzt werden (Vergleich Kapitel 3.3).

3.5 Rfam

Rfam[11] ist eine Datenbank zur Klassifizierung von nicht-kodierender RNA. Die aktuelle Version (10.1) enthält rund 2,7 Millionen Sequenzen aufgeteilt auf 1973 Familien[27]. Eine Familie ist eine Sammlung von Sequenzen, bei denen mit gewisser Sicherheit gesagt werden kann, dass sie homolog sind. Zu jeder Familie gibt es ein sogenanntes seed-Alignment, welches die am besten passenden RNA-Sequenzen enthält und als Repräsentation dieser Familie dient. Es kann benutzt werden, wenn eine neu entdeckte RNA-Sequenz kategorisiert werden soll. Für die meisten RNA-Familien gilt dabei, dass die Sequenzen ihre Funktion durch die Sekundärsstruktur

gewinnen, die daher in allen Sequenzen dieser Familien sehr ähnlich ist und auch deutlich im seed-Alignment dieser Familie erkennbar sein sollte.

3.6 Themenstellung dieser Arbeit

Als wichtige Quelle für RNA-Familien existiert die Rfam-Bibliothek. Da die seed-Alignments in Rfam als Referenz dienen sollen um neue Sequenzen zu kategorisieren, ist es wichtig, dass die gemeinsame Struktur einer Familie deutlich im seed-Alignment hervor tritt. Momentan sind aber bei einem Teil der seed-Alignments die Sekundärstrukturen der Sequenzen ungenügend aligniert, sodass die gemeinsame Struktur der Familie nicht deutlich erkennbar ist. Das erste Ziel dieser Arbeit war es daher, einen Algorithmus zu entwickeln (MAIC), der schnell eine deutliche strukturelle Verbesserung eines Alignments herbeiführen kann. Der Augenmerk des Programms liegt darauf, die Sekundärstrukturen der RNA-Sequenzen im Alignment besser zur Deckung zu bringen, sodass eine gemeinsame Struktur deutlich und eindeutig hervor tritt.

Das zweite Ziel dieser Arbeit bestand darin, existierende Programme für multiple Alignments zu vergleichen. Zu diesem Zweck sollte eine Liste verschiedener aktuell benutzter Alignment-Programme zusammengestellt werden und verglichen werden. Außerdem wurden zu diesem Zweck zwei neue Bewertungsmaße für die Qualität eines Alignments entwickelt. Das eine Maß (Interkorrelation) dient dem Vergleich mit einem Referenz-Alignment. Es bewertet die Ähnlichkeit zwischen zwei Alignments der selben Sequenzen in Bezug auf die Übereinstimmung der Faltungen im Consensus-Dotplot. Das zweite Maß (Intrakorrelation) benötigt kein Referenz-Alignment, sondern bewertet die Qualität eines Alignments im Hinblick auf die Übereinstimmung der Faltungen der Sequenzen.

Mit Hilfe dieser neuen Bewertungsmaße und einer Reihe von bereits existierenden, weit verbreiteten Maßen sollten dann die Alignment-Programme verglichen werden und in einem weiteren Schritt untersucht werden, in welchen Fällen MAIC als Nachverarbeitungsschritt Sinn macht.

Außerdem wurde untersucht, ob die Intrakorrelation ein geeignetes Maß für die Bewertung eines Alignments ist.

3.7 Arbeiten zu verwandten Themen

3.7.1 Alignmentverbesserungen

Das zu lösende Problem, Alignments zu verbessern, ist sehr nah mit dem in der Bioinformatik intensiv diskutierten Alignment-Problem verwandt. Das Alignment-Problem befasst sich mit der Erzeugung eines Alignments aus beliebigen Sequenzen. Mögliche Lösungen dieses Problems

sind zum Beispiel mLocARNA[34] oder RAF[4]. Diese Programme gehen von nicht alignierten Sequenzen aus, um ein Alignment zu berechnen. Das hier entwickelte Programm wird dagegen auf bereits berechnete, aber minder qualitative Alignments angewendet, um sie zu verbessern. Nicht automatische Ansätze dafür sind zum Beispiel in [1] und [35] enthalten. Darin werden Alignment-Editoren beschrieben, die mögliche, schlecht passende Bereiche im Alignment identifizieren. Diese können dann bei Bedarf von Hand verbessert werden.
In dieser Arbeit wurde ein Algorithmus entwickelt, der Verbesserungen vollautomatisch durchführt.

3.7.2 Benchmarks von multiple-Alignment Programmen

Eine weit verbreitete Benchmark-Datenbank für alle bei RNA relevanten Bereiche ist Bralibase. Diese Datenbank teilt sich in drei Teilbereiche, BralibaseI[8], BralibaseII[9] und BralibaseIII[7]. Für diese Arbeit ist nur BralibaseII interessant, die zusammengestellt wurde, um multiple Sequenzalignments zu bewerten. BralibaseI und BralibaseIII sind nur für die Bewertung von RNA-Faltungsprogrammen beziehungsweise für die Bewertung von Homologie-suchenden Programmen relevant. In dieser Arbeit wird allerdings nicht die ursprüngliche BralibaseII verwendet, sondern Bralibase2.1[36], eine Erweiterung dieser Datenbank. Bralibase2.1 basiert auf der Rfam-Bibliothek, wobei verschiedene Auswahlkriterien zum Tragen kamen, die sicher stellen sollen, dass die als Referenz dienenden Alignments auch wirklich korrekt sind. Insgesamt 18990 Alignments sind in Bralibase2.1 enthalten. Sie bestehen aus jeweils zwischen 2 und 15 Sequenzen, wobei nicht ganz die Hälfte, 8976 Alignments, aus 2 Sequenzen bestehen. Die Längen der Alignments variieren zwischen 25 und 400 Basen mit einem Schwerpunkt auf kurzen Alignments bis 150 Basen.

Um die Qualität der Alignments zu bewerten, gibt es verschiedene Methoden. Weit verbreitet und auch in Bralibase2.1 eingesetzt sind der SumOfPairs-Score (SPS), der Structure Conservation Index (SCI) und Matthews Korrelationskoeffizient (MCC) [4, 15, 18, 29, 8, 2, 9, 16, 14]:

SumOfPairs Score (SPS) wie er in Bralibase 2.1 verwendet wird (berechnet mit dem compalignP Script von `http://www.biophys.uni-duesseldorf.de/bralibase/`). Der SPS berechnet die Übereinstimmung der Basen zwischen einem Referenz-Alignment und dem berechneten Alignment. Dafür werden alle Sequenzpaare angeschaut und der Quotient aus der Anzahl identischer Alignmentspalten in Referenz und Berechnung und der Anzahl aller Alignmentspalten berechnet.

Structure Conservation Index (SCI) Der SCI wird ebenfalls in Bralibase 2.1 verwendet. Er berechnet sich aus der minimalen freien Energie des Alignments dividiert durch das Mittel der freien Energien der Sequenzen. Zur Berechnung der freien Energien wird der RNAalifold- und der RNAfold-Algorithmus benutzt. Ein SCI nahe 0 bedeutet, dass keine ge-

meinsame Struktur im Alignment gefunden werden konnte, ein $SCI = 1$ zeigt eine perfekt konservierte Sekundärstruktur an, während ein SCI größer als 1 eine perfekt konservierte Struktur anzeigt, die außerdem von kompensatorischen oder konsistenten Mutationen unterstützt wird[36].

Matthews Korrelationskoeffizient (MCC) Der MCC ist ein Maß, wie stark die Consensusstruktur zwischen dem Referenz-Alignment und dem berechneten Alignment übereinstimmt. Der Wert berechnet sich, indem man die Anzahl kompatibler Basenpaare in beiden Strukturen durch die Gesamtzahl an Basenpaaren dividiert. Berechnet werden kann der MCC mit dem „compare_ct.pm“ Script von der Bralibase Homepage (`http://projects.binf.ku.dk/pgardner/bralibase/compare_ct.pm`).

3.8 Überblick

In diesem Kapitel wurden die für das Thema relevanten Hintergründe der Bioinformatik eingeführt und die Fragestellung erläutert. Im Folgenden wird nun auf die im Rahmen dieser Arbeit entwickelten Verfahren eingegangen. Daran schließt sich ein Benchmark-Kapitel an, in dem die Alignmentmethoden verglichen werden und der Anwendungsbereich des entwickelten Alignmentverbesserungsalgorithmus analysiert wird. Die Zusammenfassung der Ergebnisse und der Ausblick schließen die Arbeit ab.

4 Entwickelte Verfahren

In diesem Kapitel werden die in dieser Arbeit entwickelten Bewertungsmethoden und Verfahren vorgestellt.

4.1 Bewertungsmethoden

4.1.1 Die Intrakorrelation eines Alignments

Die Intrakorrelation ist ein in dieser Arbeit neu entwickeltes Verfahren, um die Güte eines Alignments zu bewerten. Die (durch Einfügen von Gaps auf die selbe Größe gebrachten) Dotplots der Sequenzen und der Consensus-Dotplot werden als eindimensionale Vektoren interpretiert, wobei in den Dotplots nicht vorhandene Punkte als Wahrscheinlichkeit 0 gesetzt werden. Bildlich gesprochen wird der Dotplot zeilenweise hintereinander gehängt, sodass statt einer quadratischen Matrix ein eindimensionaler Vektor entsteht. Zwischen jeweils zwei der Vektoren kann nun mit Formel 4.1 der Korrelationskoeffizient berechnet werden.

$$IntraCorr(x,y) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})*(y_i-\bar{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}*\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{4.1}$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.2}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.3}$$

x und y sind die als Vektoren interpretierten zwei Dotplots, zwischen denen der Korrelationskoeffizient berechnet werden soll, beide haben n Einträge. x_i und y_i sind die Werte an der Position i in den Vektoren x und y.

Ein Koeffizient wird für jede Sequenz, also dem Paar aus Sequenz-Dotplot und Consensus-Dotplot, berechnet. Dann wird über alle Koeffizienten gemittelt. Je höher der berechnete Wert ist, desto genauer stimmen die Faltungswahrscheinlichkeiten der Sequenzen überein. Die Intrakorrelation eines Alignments ist ein Maß dafür, wie gut die Strukturen der einzelnen Sequenzen zum Alignment passen.

Auch wenn ein Korrelationskoeffizient im Allgemeinen zwischen -1 und 1 liegt, ist bei dieser speziellen Anwendung der Korrelation ein Wert von unter 0.3 nur selten zu beobachten. Ein Wert von unter 0.5 deutet bereits deutlich auf ein schlechtes strukturelles Alignments oder recht unterschiedliche Faltungen der beteiligten Sequenzen hin.

4.1.2 Die Interkorrelation zweier Alignments

Die Interkorrelation wird zwischen zwei Alignments der selben Sequenzen berechnet. Zu beiden Alignments werden Consensus-Dotplots erzeugt. Dann werden iterativ alle Sequenzen durch gegangen. Für jede Sequenz werden „Teil-Consensus-Dotplots“ aus den zwei Consensus-Dotplots der Alignments gebildet. Ein Teil-Consensus-Dotplot ist ein Consensus-Dotplot bei dem alle Zeilen und Spalten entfernt wurden, bei denen die betrachtete Sequenz Gaps im Alignment enthält. Die Teil-Consensus-Dotplots der zwei Alignments für die selbe Sequenz sind gleich groß und können wie bei der Intrakorrelation als Vektoren interpretiert werden. Nun kann mit Formel 4.4 ein Korrelationskoeffizient berechnet werden. Auf diese Art werden Koeffizienten für jede Sequenz im Alignment berechnet. Die Interkorrelation ist dann das Mittel über die Korrelationskoeffizienten.

$$InterCorr(x,y) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})*(y_i-\bar{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}*\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{4.4}$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.5}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.6}$$

x und y sind die als Vektoren interpretierten zwei Dotplots, zwischen denen der Korrelationskoeffizient berechnet werden soll, beide haben n Einträge. x_i und y_i sind die Werte an der Position i in den Vektoren x und y.

Theoretisch skaliert ein Korrelationskoeffizient zwischen -1 und 1, da aber die Alignments aus den selben Sequenzen bestehen ist ein negativer Korrelationskoeffizient wahrscheinlich ausgeschlossen.

4.2 Der MAIC-Algorithmus

MAIC (=**M**ultiple-**A**lignment **I**mprovment through **C**onsensus-dotplots) wurde entwickelt, um fehlerhafte Seed-Alignments in Rfam[11] zu verbessern. Zweck des Programms ist es, Strukturen in den Dotplots der Sequenzen zu identifizieren und gemeinsame Strukturen eines Teils der Sequenzen im Alignment möglichst zu alignieren. Dafür wird jeweils der Dotplot einer Sequenz

genommen und die gefundenen Stems mit den Stems, die im Consensus-Dotplot zu finden sind, verglichen. Falls Sequenz-Stems nahe, aber nicht exakt, passend zu Consensus-Stems liegen, wird das Alignment dahingehend verändert, dass sie nun deckungsgleich liegen. Der Gewinn dabei ist, dass nun eine gemeinsame Consensus-Struktur der Sequenzen deutlicher hervor tritt. Das ist zum Beispiel in folgendem Alignment zu sehen:

```
RF01344:
AE004439.[...] GUUGUAGUUCCCUCUCUCAUUUCGCAGUGCUACAAU
MFE-Struktur:  (((((((((.....................)))))))))
CP000019.[...] GUUUUAACUCCCUUUCUCAUUUCGCAAUGCUACAAU
MFE-Struktur:  ..................................
AE004969.[...] GUUAUUGCUCCCGUUCUCAUUUCGCAGUGCUACAAU
MFE-Struktur:  ..((((((.............))))))......
AL157959.[...] GUUGUAGCUCCCUUUCUCAUUUCGCAGUGCUACAAU
MFE-Struktur:  (((((((((.....................)))))))))
CP000746.[...] GUUGUAGCUCCCUUUUUCAUUUCGCAGUGCUAUAAU
MFE-Struktur:  (((((((((.....................)))))))))

CS Struktur:   (((((((((.....................)))))))))
```

So wie das Alignment hier vorliegt, ist es zwar unter Betrachtung der Sequenzähnlichkeit gut aligniert, die Sekundärstruktur (in Dot-Bracket-Schreibweise, jeweils unter der Sequenz) weist aber bei der dritten Sequenz einen Unterschied auf.

Nach der Optimierung könnte das Alignment zum Beispiel so aussehen:

```
RF01344:
AE004439.[...] GUUGUAGUUCCCUCUCUCAUUUCGCAGUGCUAC-----AAU
MFE-Struktur:  (((((((((.....................)))))).....)))
CP000019.[...] GUUUUAACUCCCUUUCUCAUUUCGCAAUGCUAC-----AAU
MFE-Struktur:  ......................................
AE004969.[...] GUUAUUGCUCCCGUUCUCAUUUC-----GCAGUGCUACAAU
MFE-Struktur:  ..((((((.....................))))))......
AL157959.[...] GUUGUAGCUCCCUUUCUCAUUUCGCAGUGCUAC-----AAU
MFE-Struktur:  (((((((((.....................)))))).....)))
CP000746.[...] GUUGUAGCUCCCUUUUUCAUUUCGCAGUGCUAU-----AAU
MFE-Struktur:  (((((((((.....................)))))).....)))

CS Struktur:   (((((((((.....................)))))).....)))
```

Es liegt nun eine deutlich höhere Übereinstimmung zwischen den Sekundärstrukturen vor. Wie auch zu sehen ist, geht das allerdings teilweise auf Kosten der Sequenzübereinstimmung. Es ist immer abzuwägen, was wichtiger ist. Bei MAIC wurde der Schwerpunkt auf die Struktur gelegt. Falls es möglich ist, die Übereinstimmung der Struktur deutlich zu erhöhen, soll dies durchge-

führt werden, unabhängig davon, ob sich dies negativ auf die Übereinstimmung in der Sequenz auswirkt.
Dieser Ansatz wurde verfolgt, da die Rfam-Familien Sequenzen von evolutionär weit entfernten Lebewesen enthalten können. In diesen Fällen ist es durchaus möglich, dass weite Teil der Basenfolge durch Mutationen nicht mehr übereinstimmen, aber die Funktion gebende Struktur erhalten geblieben ist. Damit ist es deutlich wichtiger, die Struktur korrekt zu alignieren als die Basensequenz. Außerdem gibt es bereits eine ganze Reihe von Programmen, die versuchen die Sequenzübereinstimmung und die strukturelle Übereinstimmung gleichermaßen zu beachten. Dieser duale Ansatz geht allerdings stark auf Kosten der Geschwindigkeit. Auch mit heuristischen Methoden steigt die Laufzeit sehr schnell mit der Sequenzlänge und der Sequenzanzahl.

4.2.1 Bewertungsfunktionen

In MAIC wurden zwei Bewertungsfunktionen benutzt, die den Zweck haben, zu prüfen, ob die gemachten Veränderungen eine Verbesserung herbei geführt haben oder ob das entstandene Ergebnis schlechter ist als die Ausgangssituation. Diese sind:

- die in dieser Arbeit entwickelte **Intrakorrelation** des Alignments (Kapitel 4.1.1)
- die **minimale freie Energie (MFE)** des Alignments

Die MFE wird mit Hilfe von RNAalifold[3] berechnet. Sie ist immer negativ und je niedriger sie ist, desto stabiler ist die energetisch optimale Struktur des Alignments. Die Idee zur Benutzung der MFE als Bewertungsfunktion basiert auf der Verwendung des „Structure Conservation Index“ (SCI) als Benchmarkmethode in Bralibase2[9] und Bralibase2.1[36]. Der SCI berechnet sich, indem man über die MFE der Einzelsequenzen mittelt und die MFE des Alignments hierdurch teilt. Da MAIC nur auf genau einem Alignment arbeitet, kann der Divisionsschritt weg gelassen werden.

4.2.2 Programmablauf

MAIC iteriert so lange über alle Sequenzen, bis bei keiner Sequenz mehr etwas verbessert werden kann oder die maximale Anzahl an Iterationen erreicht ist. Ob eine Verbesserung eintrat, wird mit Hilfe der oben erläuterten Bewertungsfunktionen ermittelt. Die Sequenzen werden sortiert nach dem Korrelationskoeffizient ihrer Dotplots zum Consensus-Dotplot abgearbeitet, beginnend mit der schlechtest passenden Sequenz. Dies stellt sicher, dass die Minderheit schlecht alignierter Sequenzen an die Mehrheit angepasst werden, statt umgekehrt.

Der eigentliche Algorithmus geht in 4 Schritten vor.

1. Stems im Dotplot der betrachteten Sequenz und im Consensus-Dotplot finden

2. Paarweise die Stems der Sequenz mit den Stems des Consensus alignieren
3. Abarbeiten der Paarungen nach Bewertung und Einbau in das Alignment
4. Prüfen, ob das Alignment verbessert wurde, eventuell Verwerfen der Änderungen
5. weiter mit der nächsten Sequenz bei Schritt 1

Wenn eine Sequenz betrachtet wird, wird (falls nicht vorhanden) ein „gapped-Dotplot“ des Faltungsverhaltens dieser Sequenz erzeugt. Ein gapped-Dotplot ist ein normaler Dotplot, nur dass an den Stellen, an denen die Sequenz im Alignment ein Gap enthält leere Spalten und Zeilen in den Dotplot eingefügt wurden. Aus den Gapped-Dotplots aller anderen Sequenzen wird ein Consensus-Dotplot erzeugt.

Stems finden

Im Gapped-Dotplot der Sequenz und dem Consensus-Dotplot wird jeweils nach ausgeprägten Stems gesucht. Ausgeprägte Stems sind alle Basenpaare in den Dotplots, deren Wahrscheinlichkeit oberhalb eines Schwellwerts liegen. Standardmäßig ist dieser auf 0.4 gesetzt. Im nächsten Schritt werden zusammenhängende Basenpaare, die also im Dotplot in der Diagonalen benachbart sind, zu einem Stem zusammengefasst.

Paarweise Stem-Alignments

Alle Stems des Gapped-Dotplots werden mit allen in der Nähe liegenden Stems des Consensus-Dotplots paarweise aligniert. „Nähe“ heißt, dass nur wenige Basen zwischen den Mitten der Stems liegen. Standardmäßig ist dieser Wert auf maximal sieben Basen gesetzt. Der paarweise Stem-Alignment-Algorithmus wurde in dynamischer Programmierung verwirklicht und orientiert sich am Needleman-Wunsch Algorithmus[21, 32] mit einheitlichen Gapkosten, wie er auch in Kapitel 3.4 erläutert ist. Anders als bei Needleman-Wunsch werden aber nicht einzelne Basen, sondern Basenpaare aligniert, und anstatt zwei Sequenzen (aus Basenpaaren) zu vergleichen, wird eine Basenpaar-Sequenz mit einer Gemeinschaft von Basenpaar-Sequenzen aligniert.

Rekursionsschritt:
Es wird das Basenpaar an der momentan aktuellen Stelle in der Sequenz mit den Basenpaaren aller anderen Sequenzen an dieser Stelle verglichen und über die paarweisen Bewertungen gemittelt. Als Bewertung wird die Basenpaarähnlichkeit und die Basenpaarwahrscheinlichkeitsdifferenz verwendet.
Die Basenpaarähnlichkeit wird mit Hilfe der Ribosumscore-Matrix[17] bewertet, wenn alle 4 Basen vorliegen. Falls ein oder mehr Gaps im momentan betrachteten Quadrupel sind, wird die Ribosumscore-Einzelbasentabelle benutzt, wobei sie um eine Gap-Zeile und Spalte erweitert

wurde, die allen Alignments zwischen einer Base und einem Gap den Wert -1 zuweist (fest). Die verwendeten Ribosum-Matrizen sind im Anhang, Kapitel 7.4 zu finden. Alignment-Gaps werden mit -3 beziehungsweise -6 (je nachdem ob das Gap in der Sequenz oder im Restalignment eingefügt wird) bewertet. Bei der Basenpaarwahrscheinlichkeit wird die Differenz berechnet und auf eine Skala von 4 bis -4 normalisiert, mit 4 als „Wahrscheinlichkeiten identisch“ bis -4 „Wahrscheinlichkeiten maximal unterschiedlich“. Der Score basiert zu 80% auf dem Ribosumscore und zu 20% auf der Wahrscheinlichkeit. Es wird ein songenanntes „Semiglobales“-Alignment erzeugt, das bedeutet, dass Gaps am Anfang und am Ende des Alignments ohne Kosten zugelassen werden. Beim Backtracking wird daher nicht in der rechten unteren Ecke begonnen, sondern in dem Feld an der rechten oder unteren Kante mit dem höchsten Wert und der Backtracking-Algorithmus ist bereits fertig, wenn die linke oder obere Kante der Matrix erreicht ist, nicht erst in Feld $(0,0)$.

Die Rekursionsformel:
A bezeichnet die Stem-Alignment-Matrix, rs ist die Ribosumscore-Matrix und g sind verschiedene Kosten für das Einfügen eines Gaps. Die betrachte Sequenz ist a, alle anderen Sequenzen des multiplen Alignments sind in $Cons$ zusammengefasst.

$$\text{Initialisierung: } \forall i(0 < i \leq \text{Länge}(Stem_a)) : A(i,0) = 0; \forall j(0 < j \leq \text{Länge}(Stem_{Cons})) : A(0,j) = 0$$

$$A(i,j) = \max \begin{cases} A(i-1,j-1) & +\sum_{s \in Cons} 0.8 * (S((k+i-1)_a, (l-i-1)_a, (m+j-1)_s, (n-j-1)_s)) \\ & +0.2 * (4 - 8 * |DP_a(k+i-1, l-i-1) - DP_s(m+j-1, n-j-1)|) \\ A(i-1,j) & +g_{Cons} \\ A(i,j-1) & +g_{Sequenz} \end{cases}$$

$$g_{Cons} = -6; g_{Sequenz} = -3$$

$$S(a,b,c,d) = \begin{cases} rs(a,b,c,d) & \text{falls } a,b,c,d \text{ kein Gap} \\ rs(a,c) + rs(b,d) & \text{sonst} \end{cases}$$

$$\text{Stem-Alignment-Score} = \max \left(\max_y A(i,y), \max_x A(x,j) \right)$$

i, j sind die Indizes der Stem-Alignment-Matrix A; k, l ist die Position des ersten Basenpaars im betrachteten Stem der momentanen Sequenz; m, n ist die Position des ersten Basenpaars des Consensus-Stems. s ist eine Sequenz aus dem multiplen Alignment $Cons$. $(j+m-1)_s$ ist die Base an Position $(j+m-1)$ der Sequenz s. $rs(a,b,c,d)$ ist der Ribosumscore des Basenpaar-Paars (a,b) und (c,d), entsprechend ist $rs(a,c)$ der Ribosumscore der Basen a und c; falls a oder c ein Gap ist, ist der Ribosumscore -1, falls beide Positionen ein Gap enthalten, ist der Score 0. $DP_s(i,j)$ ist der Wert an der Position i, j im Dotplot der Sequenz s.

Änderung des Alignments

Nachdem alle paarweisen Stem-Alignments berechnet wurden, werden alle Stem-Alignments verworfen, bei denen die Stems überhaupt nicht aligniert sind, sondern durch Einfügen von Gaps am Anfang beziehungsweise am Ende hintereinander angeordnet sind. Stem-Alignments, die zu keiner Veränderung im Sequenz-Alignment führen, bekommen einen Bonus von 2 auf ihre Bewertung. Nun werden die Stem-Alignments nach der Bewertung sortiert abgearbeitet und, soweit miteinander kompatibel, zu einer Liste von durchzuführenden Veränderungen im multiplen Sequenz-Alignment umgebaut. Falls bereits Gaps im Alignment enthalten sind, diese maximal 10 Basen von zu verändernden Positionen entfernt sind und die Veränderung das Entfernen von Gaps in der Sequenz vorschlägt, werden diese Gaps entfernt.

Anhand eines Beispiels soll dieser Vorgang veranschaulicht werden. Verwendet wird das seed-Alignment RF01347 aus Rfam.

Das Alignment vor der Veränderung sieht wie folgt aus:

GUU**UCC**AU*CCCC*GUGA*GGGG*UAAA**GGA**AUUAAAAC

GU**UUC**CAU*CCCC*GUGA*GGGG*AAU**AAG**UGUUUUGAA
GU**UUC**CAU*CCCC*GUGA*GGGG*UAA**GAG**AUUAAAAAC

Die erste Sequenz des Alignments ist die momentan betrachtete und es wird versucht, zwischen den durch Fettdruck markierten Stems zu alignieren. Außerdem enthält das Alignment einen weiteren Stem (Kursivdruck, Position 9 bis 12 und 17 bis 20), der bereits korrekt aligniert ist und daher einen Bonus auf seine Bewertung bekommt (auch zu diesem Stem wurde ein Stem-Alignment berechnet, das wird hier aber nicht gezeigt). Dieser Stem definiert damit einen Bereich, der Priorität hat und nicht verändert werden sollte. Bereits passende Alignments werden aber nicht erzwungen verwendet, da es eine Verschiebung in dem Bereich geben könnte, deren Alignment so gut ist, dass es einen noch höheren Score hat als das Alignment plus Bonus. In einem solchen Fall wird statt dem momentan passenden, das neue Alignment bevorzugt.

Der Alignmentschritt wird übersprungen, da das Ergebnis relativ eindeutig sein dürfte: Die markierten Bereiche würden besser zusammen passen, wenn sie in der Sequenz jeweils um eine Base nach links wandern. Das Stem-Alignment, aufgeschrieben mit Positionstupeln der Basenpaare, sieht so aus:

```
Sequenz: (04,27), (05,26), (06,25)
Consens: (03,26), (04,25), (05,24)
```

Dies ist so zu lesen, dass das Stem-Alignment das Basenpaar $(04,27)$ in der Sequenz mit dem Basenpaar $(03,26)$ im multiple-Alignment(Consens) aligniert hat. Entsprechend die weiteren Positionen. Dies ist genau die bereits vermutete Verschiebung des linken und rechten Stem-Teils in der Sequenz jeweils um eine Position nach links (Veränderungen sind relativ zur momentan betrachteten Sequenz definiert). Aus dem Stem-Alignment und dem bereits korrekt alignierten

Stem (Position 9 bis 12 und 17 bis 20) wird nun ein Liste von Intervallen zusammengebaut, die definieren, wo Veränderungen der Sequenz erlaubt sind und durchgeführt werden sollen:

Position Anfang bis 4 links nichts, rechts 1 Gap entfernen

Position 6 bis 9 links 1 Gap einfügen, rechts nichts

Position 12 bis 17 links nichts, rechts nichts

Position 20 bis 25 links nichts, rechts 1 Gap entfernen

Position 27 bis Ende links 1 Gap einfügen, rechts nichts

Um die Stems passend zu machen, muss offensichtlich links von Position 4 ein Gap in der Sequenz entfernt werden und zwar am rechten Rand des Intervalls, zwischen Position 6 und 9 muss ein Gap am linken Rand eingefügt werden. Im Bereich zwischen Position 12 und 17 ist nichts zu tun, weil dort keine Stems gefunden wurden. Wie die Schreibweise bereits andeutet, können pro Intervall auch beidseitig Veränderungen definiert werden. In solchen Fällen kann teilweise optimiert werden. Wenn zum Beispiel auf der linken Seite ein Gap entfernt werden soll und rechts ein Gap eingefügt werden soll, muss im Endeffekt in diesem Intervall nichts gemacht werden. Diese Optimierung wird nur bei bei kleinen Intervallen, die weniger als 10 Positionen lang sind, durchgeführt.
Bei komplexeren Alignments kann es vorkommen, dass Stem-Verschiebungen inkompatibel zueinander sind. Daher wird in der Reihenfolge der Bewertungen die Intervall-Liste zusammen gebaut. Wenn ein Stem-Alignment nicht in die bereits angelegte Intervall-Liste passt, wird dieses Stem-Alignment verworfen.

Der Algorithmus prüft nun, ob bereits Gaps im Alignment vorhanden sind, die benutzt werden können, um die Veränderung durchzuführen. Zum Beispiel wird geprüft, ob links von Position 4 ein Gap in der Sequenz enthalten ist, der entfernt werden kann. Dies ist hier nicht so, also werden Gaps direkt an den Stem anschließend in die anderen Sequenzen eingefügt. Das fertig veränderte Alignment sieht dann so aus:

```
GUU-AUCCCCGUGAGGGGUAAA-AUUAAAAC
GU-UUCAUCCCCGUGAGGGGAAU-AAGUGUUUUGAA
GU-UUCAUCCCCGUGAGGGGUAA-GAGAUUAAAAAC
```

Prüfen auf Verbesserung

Nach dem Umbau des Alignments wird geprüft, ob die Intrakorrelation des neuen multiplen Alignments gestiegen ist und ob die minimale freie Energie konstant oder gesunken ist. Wenn dies der Fall ist, werden die Veränderungen akzeptiert, ansonsten verworfen. Danach wird mit der nächsten Sequenz weiter gemacht.

4.2.3 Weitere Optionen

Um den Speicherverbrauch und die Laufzeit zu minimieren, werden beim Einlesen der Dotplots nur Punkte gespeichert, deren Wahrscheinlichkeit höher als 0.05 ist. Insbesondere bei sehr langen Sequenzen führt dies zu einer deutlichen Geschwindigkeitserhöhung. Dies kann per Option aber auch auf einen anderen Wert oder auf 0 gesetzt werden.

Falls gewünscht, kann für die Stemsuche statt des Consensus-Dotplots die Consensusstruktur benutzt werden. Dann wird versucht, die Sequenzen möglichst gut an diese Struktur anzupassen. Das deutlich reduzierte Suchfeld führt zu einer leichten Beschleunigung. Die Consensusstruktur darf dabei Pseudoknots enthalten, diese werden beachtet.

4.2.4 Anwendungsbeispiel

In Grafik 4.1 ist ein mögliches Anwendungsszenario aufgezeigt. Es handelt sich dabei um das Seed-Alignment *RF01432* aus Rfam. Der linke Dotplot ist der Consensus-Dotplot bevor MAIC angewendet wurde. Zu erkennen sind in den rot markierten Bereichen (links oben, in der Mitte und rechts unten) eine Reihe von wahrscheinlichen Faltungen, die aber gegeneinander verschoben sind und keine gemeinsamen Stems bilden. Durch Veränderungen des Alignments könnte hier bewirkt werden, dass in diesen Bereichen die Faltungen zu jeweils einem Stem zusammen fallen. Auf der rechten Seite ist zu sehen, wie der Consensus-Dotplot nach Anwendung von MAIC aussieht. In allen drei Bereichen konnte MAIC gemeinsame Stems identifizieren, und das Alignment dahingehend verändern, dass diese nun aligniert sind und damit deutlich besser im Dotplot hervor treten. Auch zu sehen ist, dass nicht sämtliche Stems in den Bereichen nun zusammenfallen. Man könnte sich unter anderem fragen, ob der grün markierte Bereich nicht auch noch mit dem Stem rechts darunter hätte aligniert werden können. Dies ist allerdings nicht möglich, da der grüne Stem eine alternative Faltung der selben Sequenzen ist, die auch den anderen Stem ausbilden können. Gleiches gilt für die Punkte, die links oben scheinbar noch nicht korrekt aligniert sind.

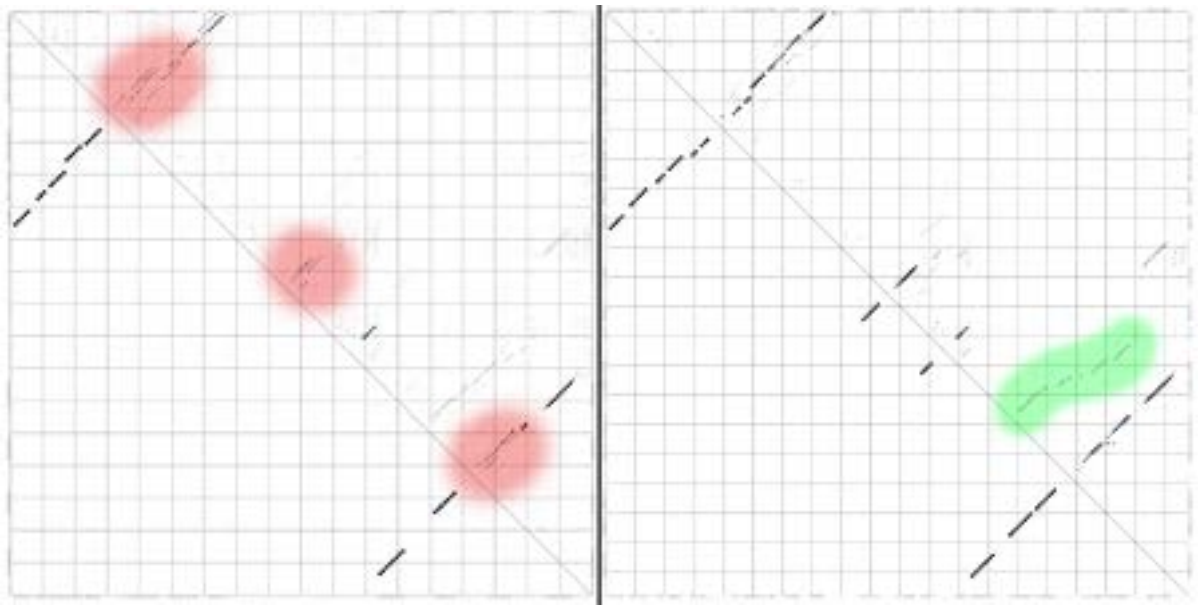


Abbildung 4.1: Veränderung durch MAIC im Consensus-Dotplot. Die roten Bereiche konnten deutlich verbessert werden. Der grüne Bereich kann nicht nach rechts unten auf den dortigen Stem verschoben werden, da er eine alternative Faltung darstellt, nicht ein ungünstiges Alignment

5 Methodenvergleich und Validierung

In diesem Kapitel werden verschiedene Analysen vorgestellt, die durchgeführt wurden:
Der erste Teil befasst sich mit dem Vergleich bisher existierender Multiple-Alignment Programme, wie diese sich bei den verschiedenen Benchmark-Methoden anordnen. Dieser Teil ist das Debüt der zwei in dieser Arbeit neu entwickelten Bewertungsmethoden, daher werden in diesem und den anderen Teilen nicht nur die Alignment-Programme verglichen, sondern es wird auch kritisch analysiert, wie sich die benutzten Bewertungsmethoden gegeneinander abgrenzen und welche Schlüsse sich daraus über die Verwendungsmöglichkeiten ziehen lassen. Als Datenbasis für diesen Teil wurde Bralibase2.1 verwendet.

Im zweiten Teil wurden die besten und schnellsten Programme aus dem ersten Teil genommen und basierend auf den von diesen Programmen produzierten Alignments MAIC ausgeführt. Ziel dieses Teils war es herauszufinden, ob und welche Alignment-Programme von MAIC als Nachverarbeitungsschritt profitieren können. Auch hierfür wurde Bralibase 2.1 als Datenbasis benutzt.

Der dritte Teil befasst sich mit der Verbesserung von Rfam Seed-Alignments. Ein Vergleich zwischen MAIC und mLocARNA wurde angestellt.

Der vierte Teil schließlich befasst sich mit der Intrakorrelation. Sie wird mit der mit LocARNA-P berechenbaren Reliability eines Alignments verglichen um zu entscheiden, ob die Intrakorrelation eine gute Alignment-Bewertungsmethode ist.

5.1 Benchmark Methoden

Es wurden fünf verschiedene Methoden verwendet, um die Güte eines Alignments zu bewerten. Drei Methoden durch den Vergleich mit einer Referenz, zwei Methoden ohne einen Vergleichspunkt:

SumOfPairs Score (SPS) wurde mit dem compalignP Script von `http://www.biophys.uni-duesseldorf.de/bralibase/`) berechnet.

Structure Conservation Index (SCI) wurde mit RNAz[31] berechnet.

Matthews Korrelationskoeffizient (MCC) wurde mit dem „compare_ct.pm“ Script berechnet, welches auf der Bralibase Homepage (`http://projects.binf.ku.dk/pgardner/bralibase/compare_ct.pm`) zur Verfügung gestellt wird ist.

Intrakorrelation wurde mit dem „IntraCorrelation.pl“ Script berechnet, einer Implementation des im Rahmen dieser Arbeit entwickelten Algorithmus gleichen Namens (Kapitel 4.1.1).

Interkorrelation wurde mit dem „InterCorrelation.pl“ Script berechnet, das eine Implementation des Interkorrelation-Algorithmus (Kapitel 4.1.2) ist.

Bei allen Methoden wurden die Standardparameter verwendet.

Entsprechend dem Vorgehen in [36] wurden SPS und SCI nicht einzeln betrachtet, sondern miteinander multipliziert um den sogenannten Braliscore zu erhalten. Dies erscheint sinnvoll, da der SPS nur auf die Basensequenz achtet, während der SCI die Struktur bewertet. Eine Kombination dieser zwei Werte beachtet damit beide Aspekte und sollte damit ein besseres Benchmark sein, als es die einzelnen Werte sind.

5.2 Analyse auf Bralibase 2.1

Im ersten Teil nun werden die Benchmark Methoden verwendet, um verschiedene Multiple-Alignment-Programme untereinander zu verglichen. Die Programme sind: mLocARNA[34], mCARNA[24], lara[2] und RAF[4] als Vertreter der Sequenz-Struktur-Alignment-Programme; Mafft[15] und ClustalW[30] um auch das Feld der reinen Sequenz-Alignment Programme abzudecken. Außerdem werden die Benchmark-Methoden kritisch betrachtet und untereinander verglichen.

Die Analysen wurden basierend auf der Bralibase 2.1-Datenbank[36] durchgeführt.

Im Gegensatz zu den Auswertungen z.B. in [36] werden in dieser Arbeit die Alignments nicht nach Sequenzanzahl aufgeteilt, sondern gemeinsam betrachtet.

Die Daten werden jeweils gegen den sogenannten APSI (=average pairwise sequence identity) abgetragen. Der APSI gibt die mittlere paarweise Basenübereinstimmung zwischen den Sequenzen eines Alignments an. Je niedriger der APSI ist, desto weniger Basen stimmen zwischen den Sequenzen überein. Der APSI grenzt damit ab zwischen Alignments aus Sequenzen, die kaum Unterschiede aufweisen und Alignments aus sehr unterschiedlichen Sequenzen. Diese Unterscheidung ist wichtig, denn je niedriger die Übereinstimmung zwischen den Sequenzen ist, desto wichtiger ist es, beim Alignment zusätzlich zu den Basen auch die Struktur zu betrachten, um die Sequenzen korrekt zu alignieren.

Für die Betrachtung der Ergebnisse muss angemerkt werden, dass mCARNA in Teilen sehr hohe Laufzeiten hat. Daher wurde bei mCARNA die Laufzeit pro paarweisem Alignment auf 100

Minuten beschränkt. Dies hat den Effekt, dass ein Teil der von mCARNA produzierten Alignments suboptimal sind, da eine weitere Verbesserung des Alignments das Zeitlimit überschritten hätte. Außerdem brach mCARNA bei etwa 5.7% der Alignments wegen des Zeitlimits ab, ohne ein Alignment zu produzieren. Die anderen Programme waren schnell genug, sodass dort das Setzen eines Zeitlimits nicht nötig war.

5.2.1 Braliscore

Beim Braliscore (Grafik 5.1) kann sich mLocARNA im niedrigen APSI Bereich klar vor den anderen Programmen positionieren. Mafft kann sich deutlich gegenüber ClustalW behaupten, bei beiden Programmen sinkt die Bewertung aber stark mit sinkendem APSI. Dies zeigt, wie wichtig eine Betrachtung der Struktur über die Sequenz hinaus sein kann. Oberhalb von einem APSI von 80 sind dagegen alle Programme (mit Ausnahme von mCARNA) annähernd gleich gut.

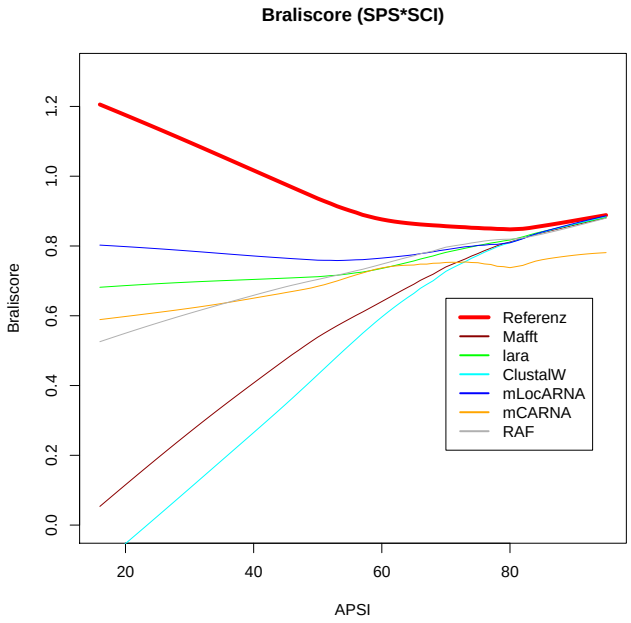


Abbildung 5.1: Braliscore gegen APSI

5.2.2 Interkorrelation

Die Interkorrelation (Grafik 5.2) deckt sich von der Aussage größtenteils mit dem Braliscore. Wieder ist mLocARNA in großen Teilen das beste Programm, und die reinen Sequenz-Alignment-Programme sind abgeschlagen weit dahinter, wobei auch hier Mafft deutlich besser ist als

ClustalW. Ein Unterschied ist, dass RAF hier etwa die selbe Leistung bringt, wie lara, während sich RAF bei Bralibase unterhalb von lara einordnet bei einem APSI unter 50.

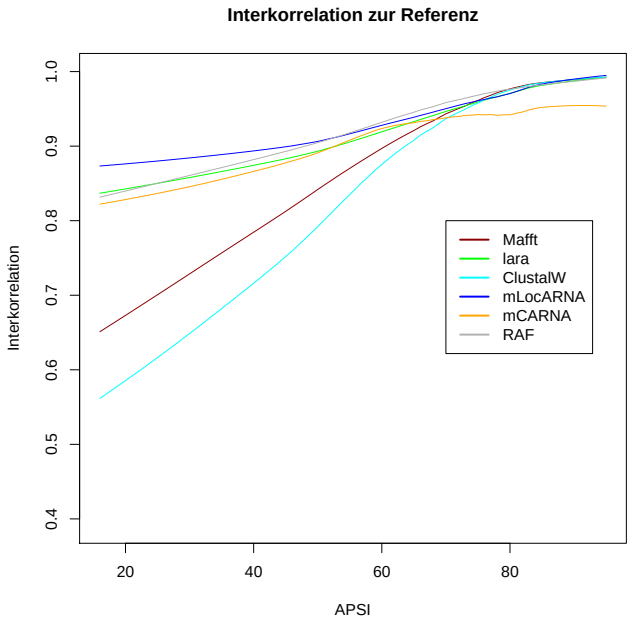


Abbildung 5.2: Interkorrelation gegen APSI

5.2.3 Intrakorrelation

Bei der Intrakorrelation (Grafik 5.3) liegen alle Sequenz-Struktur-Alignment-Programme sehr nahe an der Referenz und es lässt sich kein klarer Sieger feststellen. Die reinen Sequenz-Alignment-Programme zeigen die selbe Tendenz wie bei den vorherigen Grafiken: gute Übereinstimmung im hohen APSI Bereich, aber mit sinkendem APSI sinkt auch die Intrakorrelation unter die anderen Programme.

5.2.4 Matthews Korrelationskoeffizient (MCC)

Um die Consensus-Struktur des Alignments für die Berechnung des MCC zu bekommen, wurde in allen Fällen, in denen keine Struktur direkt ausgegeben wird RNAalifold [3] mit den Standardparametern aufgerufen. Dies war bei Mafft, ClustalW, lara und dem Referenz-Alignment der Fall.
Bei einem APSI von unter 50 zeigt sich beim MCC das selbe Bild wie bei den anderen Benchmarks: mLocARNA an erster Stelle, lara, RAF und mCARNA etwas dahinter relativ nahe beieinander und weit dahinter dann die Sequenz-Alignment-Programme, bei denen Mafft besser ist

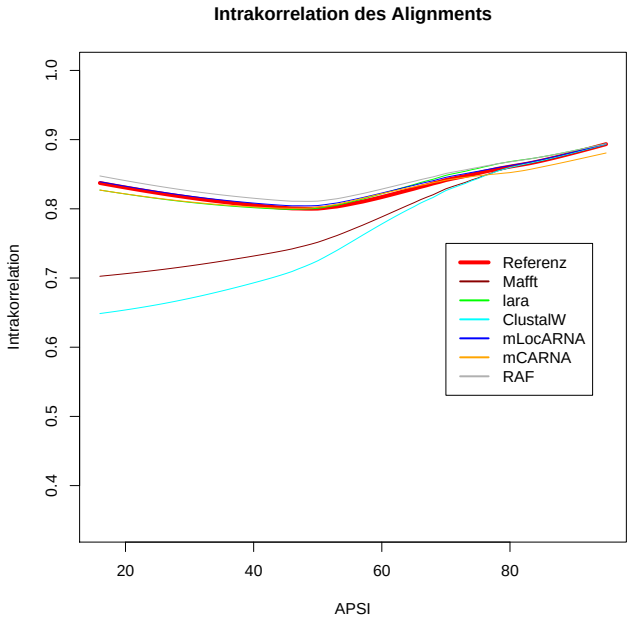


Abbildung 5.3: Intrakorrelation des Alignments gegen APSI

als ClustalW. Bei einem APSI über 50 allerdings sind die Sequenz-Alignment-Programme, Mafft und ClustalW, sowie lara deutlich besser als die anderen drei Programme.

Grafik 5.4 und Grafik 5.10, die sich ebenfalls mit dem MCC befasst, lassen gewisse Zweifel aufkommen, ob der MCC in seiner momentan vorliegenden Form, eine sinnvolle Benchmark-Methode ist. Die Grafiken scheinen insbesondere im hohen APSI-Bereich hauptsächlich zu zeigen, bei welchen Programmen der selbe Algorithmus wie beim Referenz-Alignment benutzt wurde um die Consensus-Struktur zu berechnen. Bei Mafft, lara, MAIC und ClustalW, genauso wie bei der Referenz, wurde RNAalifold mit den Standardparametern zur Berechnung benutzt, während die anderen Programme nicht genau diesen Algorithmus bzw. den Algorithmus mit etwas anderen Parametern benutzten.

Je nach Algorithmus und gewählter Parameter scheinen verschiedene Strukturen als optimal angesehen zu werden. In Grafik 5.5 ist dies deutlich zu sehen. Hier wurden für die Strukturberechnung des Referenz-Alignments statt der normalen Scores die Ribosumscores benutzt. Die Ribosumscores werden unter anderem bei mLocARNA bei der Strukturberechnung benutzt. Der Einfluss auf den MCC von mLocARNA ist deutlich. Der MCC ist also nur dann einsetzbar, wenn alle Consensus-Strukturen mit dem identischen Algorithmus und gleichen Parametern erzeugt werden. Dies macht einen Vergleich von Programmen, die intern auch direkt eine Consensus-Struktur berechnen, wie es bei RAF, mLocARNA und mCARNA der Fall ist, teilweise unsinnig: Entweder man ignoriert diesen Teil der Alignment-Programme und berechnet nochmal extra eine Struktur, damit der selbe Algorithmus mit den selben Parametern überall verwendet wird,

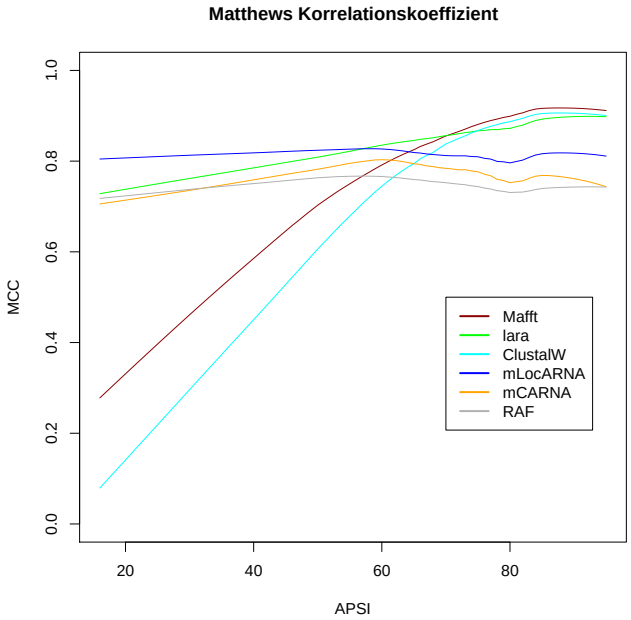


Abbildung 5.4: Matthews Korrelationskoeffizient gegen APSI

oder man hat Ergebnisse die nur sehr bedingt untereinander vergleichbar sind.

Außerdem scheint eine vollständige Fokussierung auf die Consensus-Struktur nicht zwingend ein guter Weg zu sein, um Alignments der selben Sequenzen zu vergleichen. Falls die Sequenzen nicht nur in einer, sondern in mehreren energetisch nahe beieinander liegenden Strukturen ähnlich sind, können bereits kleine Unterschiede im Alignment bewirken, dass eine andere Consensus-Struktur bevorzugt wird. Diese unterschiedliche Struktur führt dann dazu, dass der MCC des Alignments deutlich niedriger ausfällt, als es dem nur geringen Unterschied im Alignment angemessen wäre.
Es lässt sich sagen, dass man sich beim MCC im Gegensatz zu den anderen Benchmark-Methoden, deutlich genauer über die Grenzen des Benchmarks im Klaren sein muss, sonst werden die Ergebnisse möglicherweise falsch interpretiert.

5.2.5 Laufzeiten

In Tabelle 5.1 sind die durchschnittlichen und maximalen Laufzeiten, sowie der Median der Laufzeiten der Programme eingetragen. Es lässt sich erkennen, dass RAF mit deutlichem Vorsprung das schnellste der Sequenz-Struktur-Alignment-Programme ist und nur von Mafft und ClustalW als Vertreter der reinen Sequenz-Alignments geschlagen wird. mCARNA dagegen liegt deutlich abgeschlagen auf dem letzten Platz. In Grafik 5.6 ist die Laufzeit nach APSI aufgeschlüsselt. Hier lässt sich weiterhin erkennen, das sich lara im niedrigen und hohen APSI Bereich vor

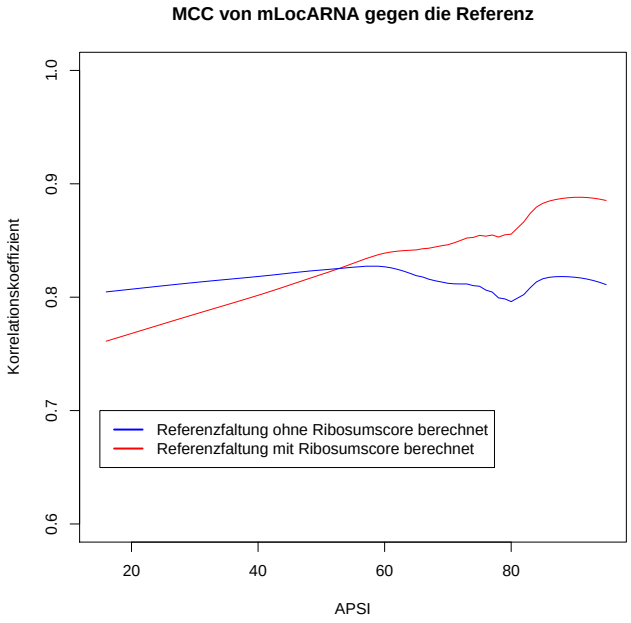


Abbildung 5.5: MCC von mLocARNA bei etwas anderen Parametern für die Consensus-Struktur-Berechnung der Referenz

mLocARNA positionieren kann, während im mittleren APSI Bereich beide ungefähr gleich auf liegen. Mafft und ClustalW liegen sehr nahe zusammen, sodass ein Geschwindigkeitsunterschied zwar vorhanden sein mag, aber anwendungstechnisch unerheblich ist.

5.2.6 Zusammenfassung

Zusammenfassend lässt sich aus der Betrachtung aller Benchmark-Methoden sagen, dass mLocARNA von den getesteten Programme die besten Alignments produziert. Mafft und ClustalW sind etwa gleich schnell, aber bei den produzierten Alignments ist Mafft in sämtlichen Benchmarks besser als ClustalW. Bei mCARNA muss noch an verschiedenen Stellen nachgebessert

Programm	mittlere Laufzeit	Maximale Laufzeit	Median
mCARNA	3127	537300	2
mLocARNA	46.39	6975	2.15
lara	16.24	1017	3
RAF	1.75	774	0.259
ClustalW	0.39	24	0.11
Mafft	0.33	27	0.19

Tabelle 5.1: Mittelwert, Maximum und Median der Laufzeiten der getesteten Programme in Sekunden, sortiert nach mittlerer Laufzeit

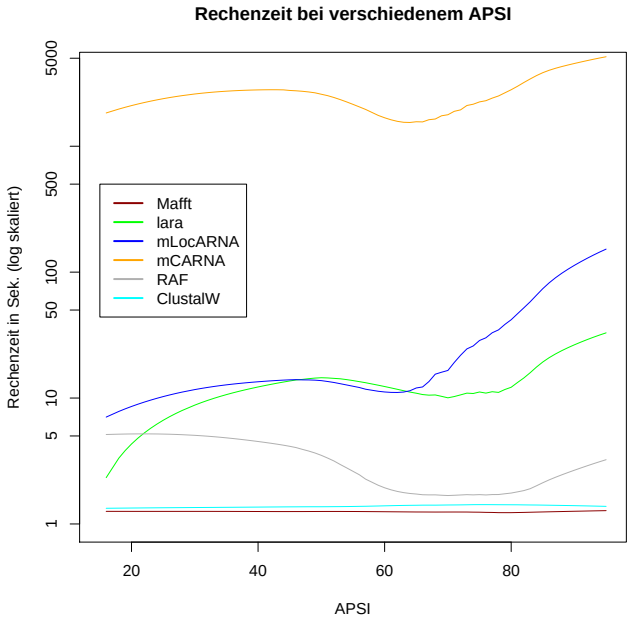


Abbildung 5.6: Laufzeiten gegen APSI

werden. Insbesondere bei der Geschwindigkeit sollte etwas gemacht werden, denn mCARNA ist momentan um Faktor 100 langsamer, als die anderen Programme. Die produzierten Alignments dagegen sind konkurrenzfähig zu lara und RAF. Hier sollte noch einmal betont werden, dass mCARNA noch in der Prototypenphase ist, starke Veränderungen und insbesondere Verbesserungen in zukünftigen Versionen sind damit sehr wahrscheinlich.

RAF konnte bei der Geschwindigkeit erstaunliches leisten und ist in Teilbereichen im Durchschnitt kaum langsamer als reine Sequenz-Alignment-Programme.

In Vergleich der Benchmark Methoden zeigt sich, dass die Aussagen von Braliscore und Interkorrelation sich in weiten Bereichen gleichen. Das ist, wenn man sich die doch sehr unterschiedliche Herangehensweise der Methoden betrachtet, ein gutes Indiz für die Stabilität und Aussagekraft beider Methoden. Beim MCC gibt es verschiedene Punkte, die ihn nicht unbedingt als allgemein empfehlenswerte Methode für den Vergleich verschiedener Alignments aus den selben Sequenzen erscheinen lassen.
Auf die Intrakorrelation wird weiter unten in Kapitel 5.5 im Detail eingegangen.

5.3 Analyse von MAIC auf Bralibase 2.1

In diesem Abschnitt wird nun untersucht, ob und wie MAIC, das in dieser Arbeit entwickelte Programm, die von anderen Programmen erzeugten Alignments verändert. Es soll die Frage beantwortet werden, ob eine Nachbearbeitung der Alignments mit MAIC eine Verbesserung herbei führt und wie sich diese Nachbearbeitung auf die Gesamtlaufzeit auswirkt.

Zur besseren Übersicht wurden nicht alle Programme aus dem vorherigen Teil wieder verwendet, sondern die Analyse hat sich auf mLocARNA als das beste Programm, RAF als das schnellste Sequenz-Struktur-Alignment-Programm und Mafft als das beste Sequenz-Alignment-Programm konzentriert.

5.3.1 Braliscore

Der Braliscore (Grafik 5.7) zeigt, dass bei den Sequenz-Struktur-Alignment-Programmen keine Verbesserung durch MAIC eintritt, während bei Mafft besonders im niedrigen APSI-Bereich Verbesserungen eintraten. Im hohen APSI Bereich dagegen tritt teilweise eine leichte Verschlechterung der Alignments ein.

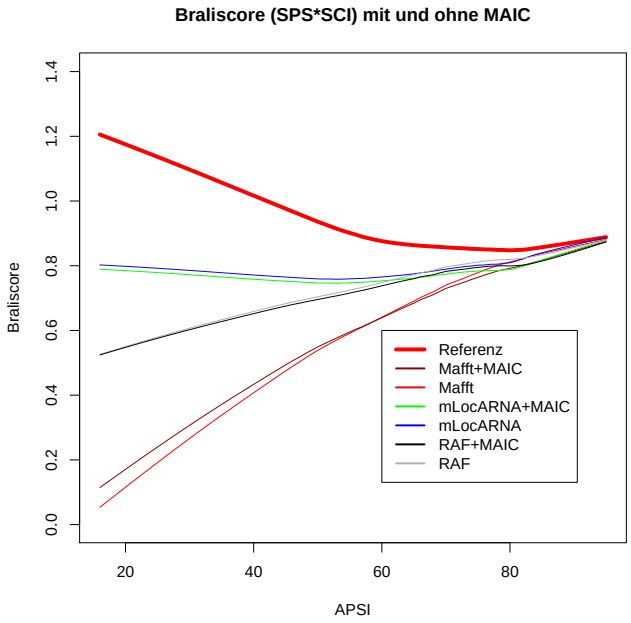


Abbildung 5.7: Braliscore gegen APSI

5.3.2 Interkorrelation

Die Interkorrelation (Grafik 5.8) zeigt ein Ähnliches Bild wie der Braliscore: bei mLocARNA und RAF sind keine Verbesserungen möglich, bei Mafft dagegen können deutliche Verbesserungen beobachtet werden. Ebenfalls wie bei Braliscore scheint es lechte Verschlechterungen bei einem APSI über 80 zu geben. Wenn man bedenkt, wie MAIC arbeitet, ist das nicht unbedingt überraschend. Je höher der APSI ist, desto interessanter und wichtiger wird es, auch die (ungepaarten) Basen korrekt zu alignieren, MAIC aber achtet nur auf die Basenpaare und deren Ausrichtung.

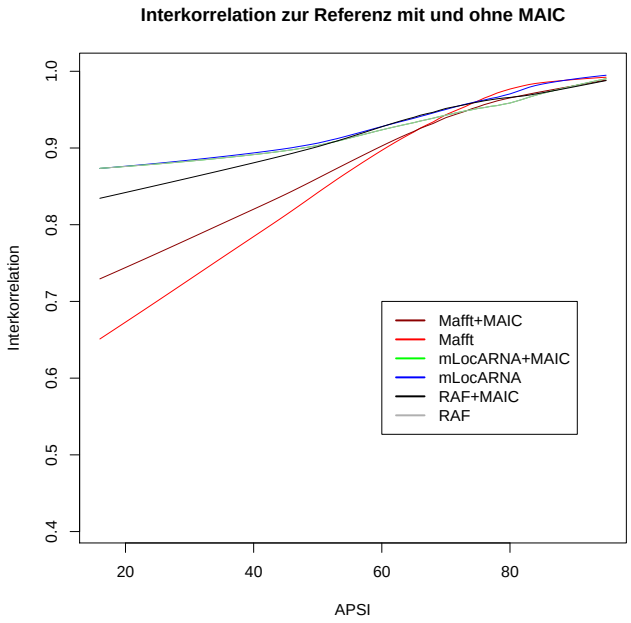


Abbildung 5.8: Korrelation zur Referenz gegen APSI

5.3.3 Intrakorrelation

Bei der Intrakorrelation 5.9 konnte bei allen Programmen und über den gesamten APSI Bereich eine leichte Verbesserung herbei geführt werden, wobei die deutlichste Steigerung auch hier bei Mafft ist. Zur besseren Unterscheidung der Kurven wurde die Referenzkurve nicht eingezeichnet, ihr Verlauf ist in Grafik 5.3 zu sehen.

5.3.4 Matthews Korrelationskoeffizient

Matthews Korrelationskoeffizient (Grafik 5.10) zeigt ein auf den ersten Blick erstaunlich anderes Bild. Hier kann MAIC eine mit dem APSI steigende sichtbare Verbesserung bei den Sequenz-

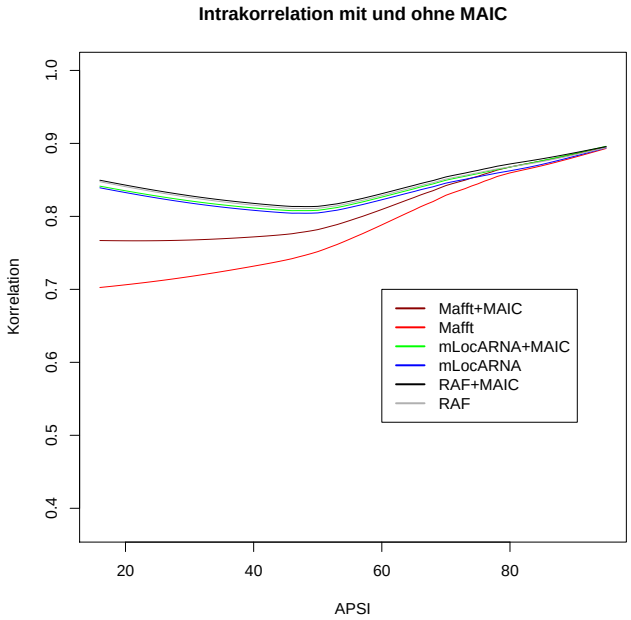


Abbildung 5.9: Intrakorrelation des Alignments gegen APSI

Programm	Laufzeit ohne MAIC	Laufzeit mit MAIC
mLocARNA	46 sek	52 sek
RAF	4 sek	7 sek
Mafft	0.3 sek	4.5 sek

Tabelle 5.2: mittlere Laufzeiten der getesteten Programme mit und ohne MAIC in Sekunden

Struktur-Alignments bewirken. Bei Mafft dagegen führt MAIC nur im unteren APSI Bereich zu einer Verbesserung, während es im oberen Bereich eine leichte Verschlechterung bewirkt. Bedenkt man die im vorherigen Teil bereits erwähnten Kritikpunkte am MCC, nämlich, dass er sehr stark vom Sekundärstruktur-Algorithmus abhängt und dass nur kleine Unterschiede zwischen den Alignments nicht zwingend mit einem hohen MCC einhergehen, stellt sich die Frage, welche Aussagekraft dieser Test eigentlich hat.

5.3.5 Laufzeiten

Die Tabelle 5.2 und die Grafiken 5.11 und 5.12 zeigen, dass der zeitliche Mehraufwand durch die Nachbearbeitung mit MAIC absolut betrachtet gering ist und auch bei langen Alignments oder Alignments mit hoher Sequenzanzahl nicht wesentlich steigt.

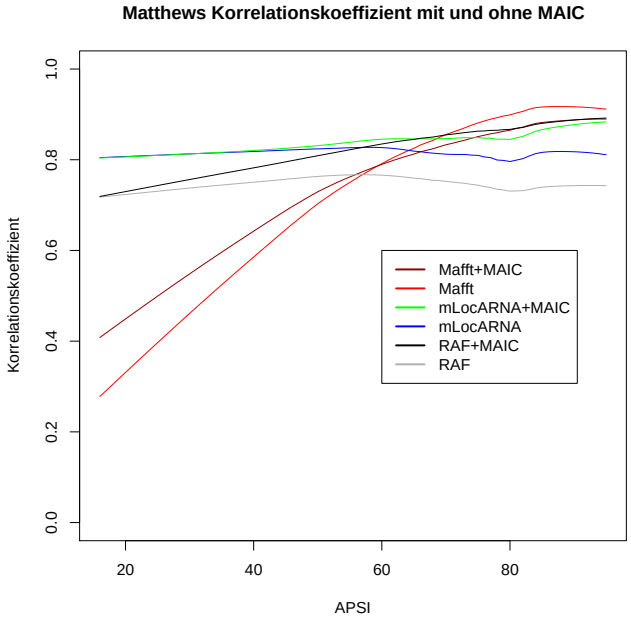


Abbildung 5.10: Matthews Korrelationskoeffizient gegen APSI

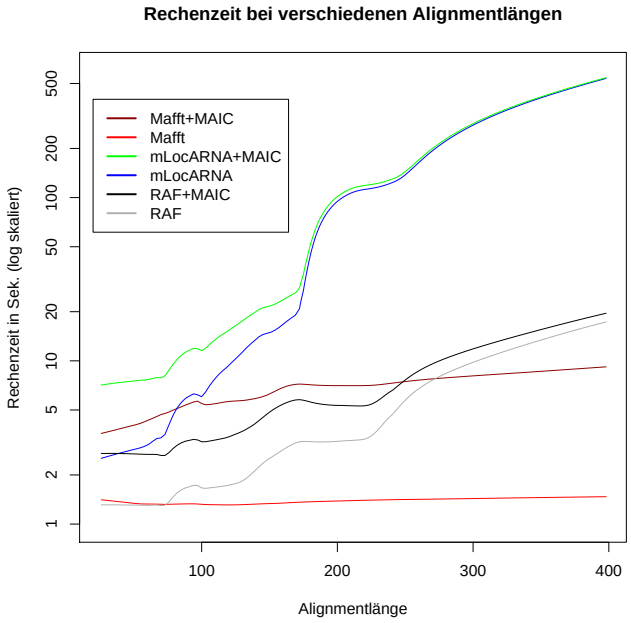


Abbildung 5.11: Laufzeiten gegen Alignmentlänge

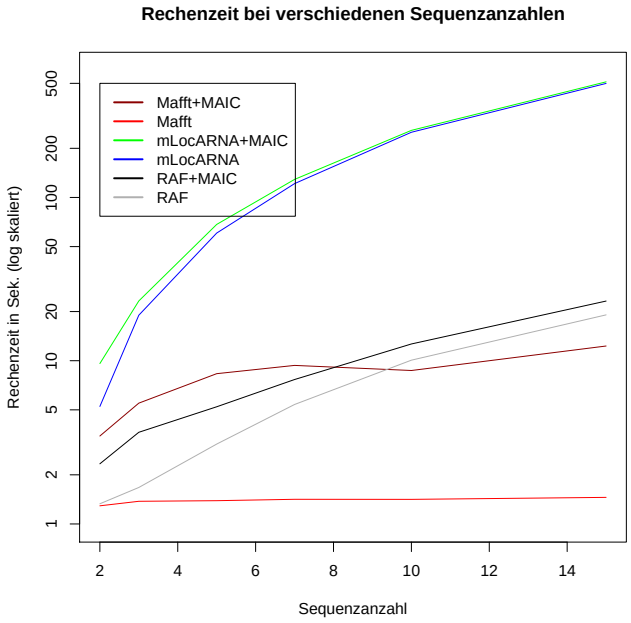


Abbildung 5.12: Laufzeiten gegen Sequenzanzahl im Alignment

5.3.6 Zusammenfassung

Hauptsächlich reine Sequenz-Alignment Programme wie Mafft können von MAIC als Nachverarbeitungsschritt profitieren, während die Sequenz-Struktur-Alignment-Programme bereits sehr hohe Qualität liefern und daher nur noch geringe Verbesserungen möglich sind.

Es ist zu bedenken, dass Bralibase nur einen kleinen Teil der möglichen Alignments beinhaltet. Insbesondere Alignments mit langen Sequenzen oder mit mehr als fünfzehn Sequenzen sind in Bralibase nicht enthalten. In diesen Bereichen sind Sequenz-Struktur-Alignment-Programme nur sehr eingeschränkt nützlich, da die Laufzeiten schnell stark steigen. Die Laufzeiten reiner Sequenz-Alignment-Programme steigen dagegen deutlich langsamer, die produzierten Alignments sind aber dementsprechend auch schlechter, da strukturelle Elemente komplett ignoriert werden. Die Analysen haben gezeigt, dass MAIC die mit Mafft erzeugten Alignments insbesondere im niedrigeren APSI Bereich deutlich verbessern kann. Die Qualität der laufzeitintensiven Sequenz-Struktur-Alignment-Programme kann im Allgemein nicht erreicht werden. Gleichzeitig können diese Verbesserungen im Mittel im einstelligen Sekundenbereich durchgeführt werden und zeigen nur eine schwache Tendenz zu einer Steigerung der Laufzeit mit Sequenzlänge und Sequenzanzahl.
Eine Kombination aus zum Beispiel Mafft und MAIC ist also bei größeren Alignments um mehrere Größenordnungen schneller als alle momentan existierenden Sequenz-Struktur-Alignmentprogramme. Sie ist dabei nur unwesentlich langsamer als Mafft alleine und gleichzeitig deutlich

besser.

Eine Einschränkung hat sich allerdings gezeigt: Alignments mit einem sehr hohen APSI (Höher als etwa 70-80) können von MAIC im Vergleich zur Referenz meist nicht verbessert werden sondern sie werden leicht schlechter. Dies ist allerdings auch der Bereich, in dem die reinen Sequenz-Alignment Programme genauso gute Ergebnisse liefern, wie die Sequenz-Struktur-Alignment Programme, sodass in diesen Bereichen kein Bedarf für Nachbearbeitungen vorhanden ist.

5.4 Analyse von MAIC auf Rfam

Im Folgenden wird die Qualität von MAIC auf der Rfam Bibliothek[11] geprüft. Es wurde untersucht, ob MAIC die vorhandenen Rfam Seed-Alignments verbessern kann.

Alle Seed-Alignments mit maximal 30 Sequenzen wurden verwendet. Von den insgesamt 1973 RNA-Familien in Rfam 10.1 sind das 1717 Alignments. 994 Alignments wurden dabei von MAIC verändert. Die Länge des Alignments wurde nicht beschränkt.

Im Gegensatz zu den Bralibase 2.1 Daten liegt hier keine Referenz vor, mit dem das Ergebnis verglichen werden könnte. Daher können in diesem Teil der nur die Benchmark Methoden benutzt werden, die ein Alignment unabhängig von einer Referenz bewerten. Von den benutzten Kriterien sind dies die Intrakorrelation und der „Structure Conservation Index“ (SCI). Um einen Vergleichswert zu bekommen wurde jeweils auch mLocARNA, welches sich in den vorherigen Kapiteln als das beste der getesteten Programme herausgestellt hat, aufgerufen. Der APSI basiert jeweils auf dem unveränderten Rfam-Alignment und wurde mit „alistat“ aus dem Squid/Hmmer Package[5] berechnet.

5.4.1 Intrakorrelation und Structure Conservation Index

In Grafik 5.13 ist zu sehen, dass MAIC die Intrakorrelation der Seed-Alignments deutlich steigern und an das Ergebnis von mLocARNA heran bringen kann.

Beim SCI dagegen tritt keine Veränderung ein (Grafik 5.14). Dies ist überraschend, da eigentlich zu erwarten wäre, dass eine Verbesserung der Stem-Übereinstimmung meist auch mit einer weiteren Senkung des Alignment-MFE einhergeht.

Der Grund dafüer liegt im speziellen MFE-Algorithmus, der bei der Berechnung des SCI benutzt wird. Dieser bewertet Gaps ungünstig. MAIC allerdings fügt gezielt Gaps in das Alignment ein, um die Stem-Übereinstimmung zu erhöhen. Die eigentlich sinkende freie Energie wird so durch den Gap-Malus in etwa kompensiert. Um also den SCI zu verbessern, müsste die Gapanzahl verringert oder zumindest konstant gehalten werden, statt sie zu erhöhen.

5.4.2 Laufzeiten

Es zeigt sich, dass im Durchschnitt und insbesondere bei langen Alignments MAIC nur einen Bruchteil der Rechenzeit von mLocARNA benötigt (siehe Tabelle 5.3, Grafik 5.15 und Grafik 5.16).

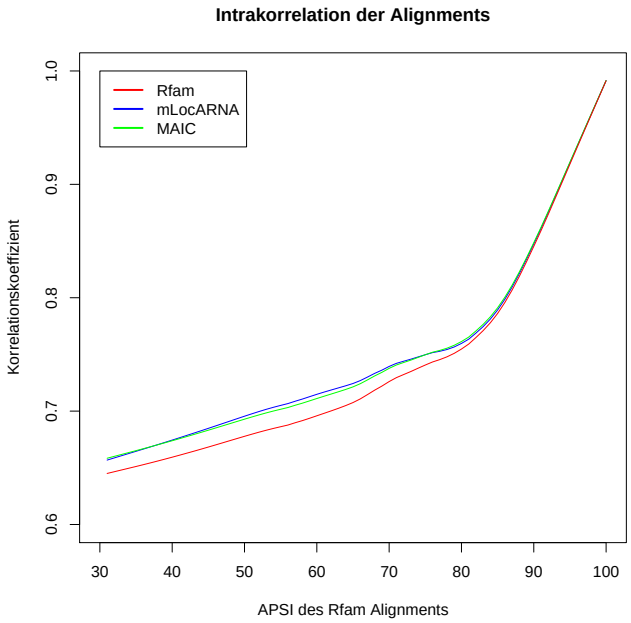


Abbildung 5.13: Intrakorrelation der Rfam Alignments vor und nach Aufruf von MAIC und nach Aufruf von mLocARNA

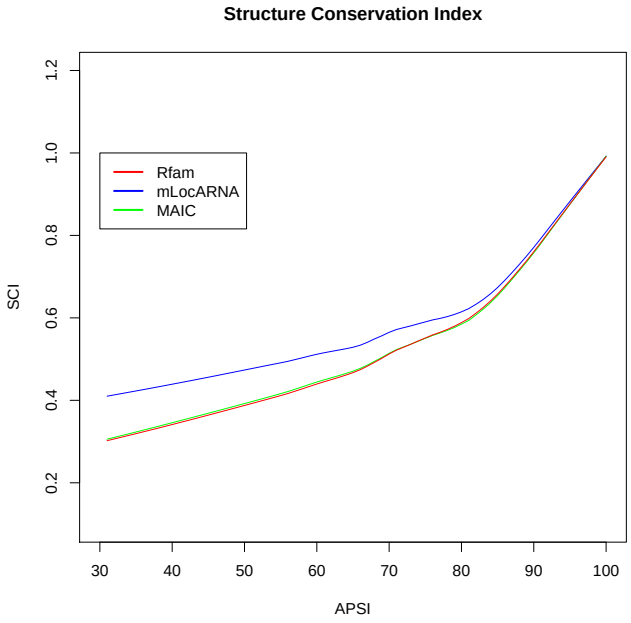


Abbildung 5.14: SCI der Rfam Alignments vor und nach Aufruf von MAIC und nach Aufruf von mLocARNA

Programm	mittlere Laufzeit	maximale Laufzeit
mLocARNA	379.7 sek	6557 sek
MAIC	4.8 sek	217 sek

Tabelle 5.3: mittlere und maximale Laufzeiten von mLocARNA und MAIC auf der Rfam Seed-Alignments mit maximal 30 Sequenzen

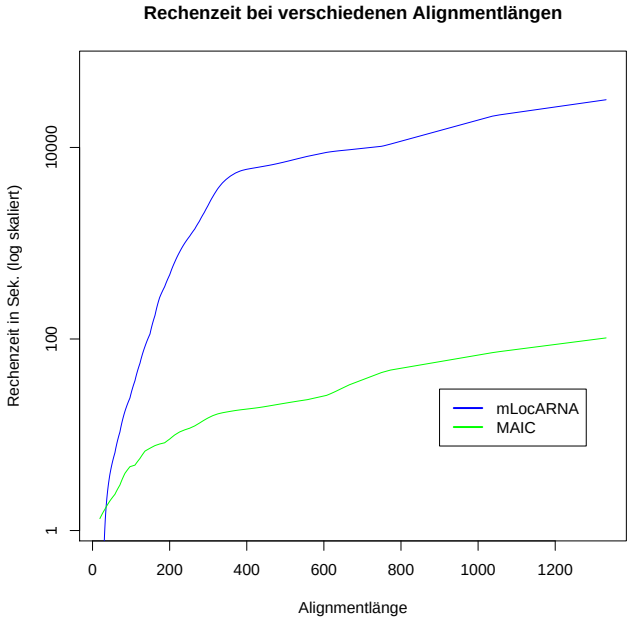


Abbildung 5.15: Rechenzeit von mLocARNA und MAIC auf den Rfam-Alignments bei verschiedenen Alignmentlängen

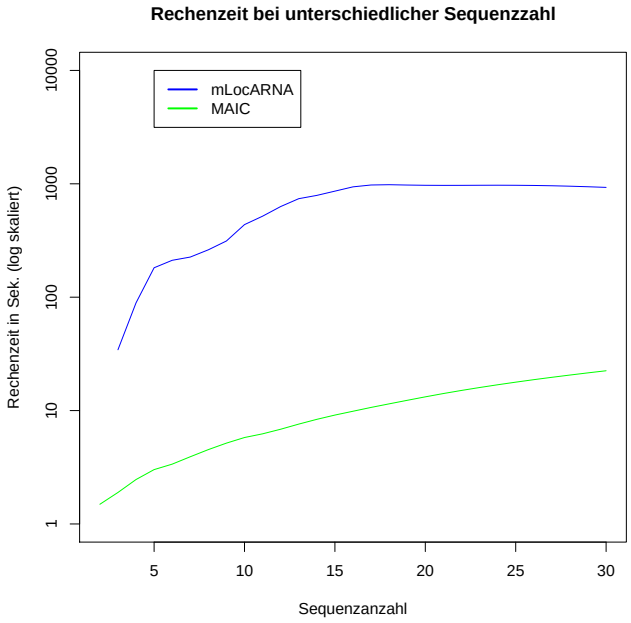


Abbildung 5.16: Rechenzeit von mLocARNA und MAIC auf den Rfam-Alignments bei unterschiedlicher Sequenzzahl

5.4.3 Zusammenfassung

Die Analyse auf der Rfam Datenbank zeigt, dass MAIC sehr schnell sichtbare Verbesserungen herbei führen kann. Wenn man die Intrakorrelation als Maßstab nimmt, wird ein ähnliches Niveau erreicht wie wenn man die Sequenzen ganz neu mit mLocARNA alignieren würde bei nur rund einem Hundertstel des Zeitaufwands. Damit ist das primäre Ziel von MAIC voll erreicht: Das Programm kann sehr schnell signifikante Verbesserungen in bereits existierenden Alignments herbei führen.

5.5 Analyse der Intrakorrelation

Die Intrakorrelation als in dieser Arbeit neu vorgestellte Bewertungsmethode für ein Alignment soll hier noch einmal genauer betrachtet werden, um eine fundierte Entscheidung treffen zu können, wie sinnvoll sie im Vergleich zu anderen Methoden ist.

Hierfür wird die Intrakorrelation mit den „STructure-based Alignment Reliabilities“(STAR) eines Alignments verglichen. STAR wird mit Hilfe von LocARNA-P[33] berechnet: Für ein multiples Alignment werden zu allen Sequenzpaaren für alle möglichen Alignmentspalten Wahrscheinlichkeiten, dass diese Alignmentspalte in einem Alignment auftritt, berechnet. Die Wahrscheinlichkeiten basieren auf der Boltzmann-Verteilung und der LocARNA-Bewertungsfunktion. Auf Basis der paarweisen Alignmentspalten-Wahrscheinlichkeiten kann LocARNA-P dann für ein multiples Alignment aus den Sequenzen eine „Reliability“ berechnen. Sie gibt damit an, wie verlässlich das Alignment ist. Von den verschiedenen Werten, die LocARNA-P ausgibt, wurde „Relibility 2/Col“ benutzt.

Der direkte Vergleich der schnell berechenbaren Intrakorrelation mit der sehr aufwendigen Reliability soll zeigen, ob die beiden Methoden zur Alignmentbewertung im Allgemeinen gleiche Aussagen über die getestet Alignments machen. Eine positive Antwort würde bedeuten, dass man ohne großen Verlust der Aussagekraft die Intrakorrelation als Bewertungsmethode benutzten kann, statt der Reliability.

Als Datenbasis für den Vergleich von Intrakorrelation und Reliability wurden die Bralibase2.1-Referenz-Alignments mit drei bis fünfzehn Sequenzen verwendet und die auf Bralibase2.1 von mLocARNA berechneten Alignments mit drei bis fünfzehn Sequenzen.

In Grafik 5.17 ist zu sehen, dass die Reliability und die Intrakorrelation gut korrelieren, sowohl bei den Referenz-Alignments aus der Bralibase 2.1 wie auch bei den (aus Bralibase berechneten) mLocARNA Alignments (Korrelationskoeffizient: 0.55 (Bralibase) bzw. 0.53 (mLocARNA)). Zu den Bralibase2.1-Referenz-Alignments wurde außerdem die Punktwolke der zugrundeliegenden Daten eingezeichnet. Zu erkennen ist, dass die Reliability größtenteils mit der Intrakorrelation korreliert, mit Ausnahme einem Bereich um den Wert 40, bei dem die Reliability eine lokale Häufung von Ergebnissen hat, die bei der Intrakorrelation nicht auftritt.

Im nächsten Schritt werden Fälle betrachtet, bei denen im Bralibase-Benchmark die Intrakorrelation annähernd identisch zwischen dem Programm-Alignment und dem Referenz-Alignment ist, aber die Interkorrelation deutliche Unterschiede aufweisen. Dies kann eventuell ein Hinweis darauf sein, dass die Alignments zwar verschieden, aber ähnlich optimal bzw. ähnlich wahrscheinlich, das heißt verlässlich („reliable“) sind. Im Allgemeinen ist anzunehmen, dass es neben der Referenz noch eine ganze Reihe weiterer Alignments geben kann, die genauso oder annähernd genauso optimal sind wie die Referenz. Ab einem bestimmten Punkt an Perfektion der Alignment-Programme ist eine Veränderung hin zu mehr Ähnlichkeit zur Referenz nur noch

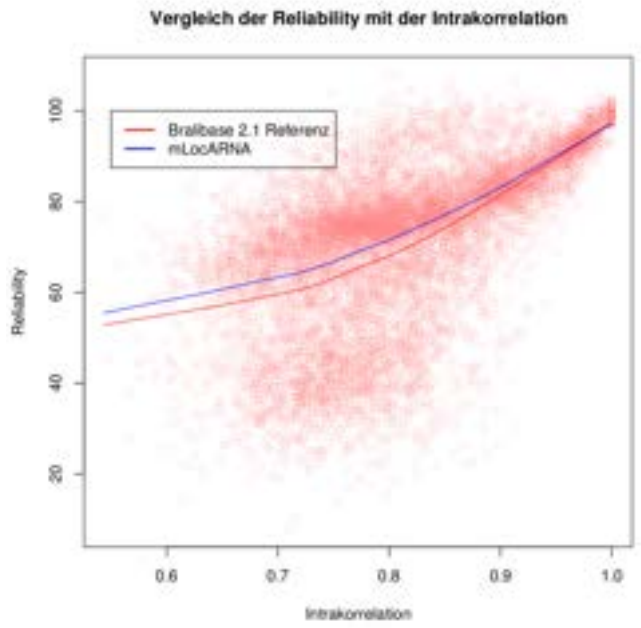


Abbildung 5.17: Reliability und Intrakorrelation bei Bralibase2.1-Referenz-Alignments und mLocARNA Alignments mit drei bis fünfzehn Sequenzen

durch spezifisches Wissen aus dritter Quelle über die Referenz oder die Sequenzen erreichbar, während die sequenzielle und strukturelle Information der Sequenzen selbst bereits vollständig und korrekt ausgeschöpft wurde. Daher soll geprüft werden, ob die Reliability auch die Intrakorrelation bestätigt, wenn man nicht schlicht über alle Daten mittelt, sondern nur Fälle betrachtet, in denen das mLocARNA-Alignment deutlich unterschiedlich zum Referenz-Alignment ist, die Intrakorrelation aber annähernd identisch ist.

Es werden von allen Alignments mit drei bis fünfzehn Sequenzen aus Bralibase2.1 das Hundertstel mit der niedrigsten Interkorrelation zwischen Referenz- und mLocARNA-Alignment gewählt. Von dieser Menge wurden dann die 50 Alignments genommen, bei denen die Differenz zwischen den Intrakorrelationen am niedrigsten ist. Der Ausschnitt aus der Datenmenge ist rot markiert in Grafik 5.18 zu sehen. Wenn die Reliabilities der Referenz-Alignments und die Reliabilities der mLocARNA-Alignments auf dieser Datenmenge annähernd gleich verteilt sind, bestätigt dies, dass LocARNA-P, in Übereinstimmung mit der Intrakorrelation, diese Alignments, obwohl der Unterschied zwischen erzeugtem Alignment und Referenz jeweils groß ist, für ähnlich gut befindet.

In Grafik 5.19 zeigt sich eine gute Übereinstimmung der Mittelwerte. Zur genaueren Prüfung wurde der Wilcox-Rangsummentest und der Kolmogorov-Smirnov-Test zwischen der Reliability der Referenz-Alignments und der Reliability der mLocARNA-Alignments angewendet. In Tabelle 5.4 sind die errechneten p-Werte eingetragen. Beide Werte sind im zweistelligen Prozentbe-

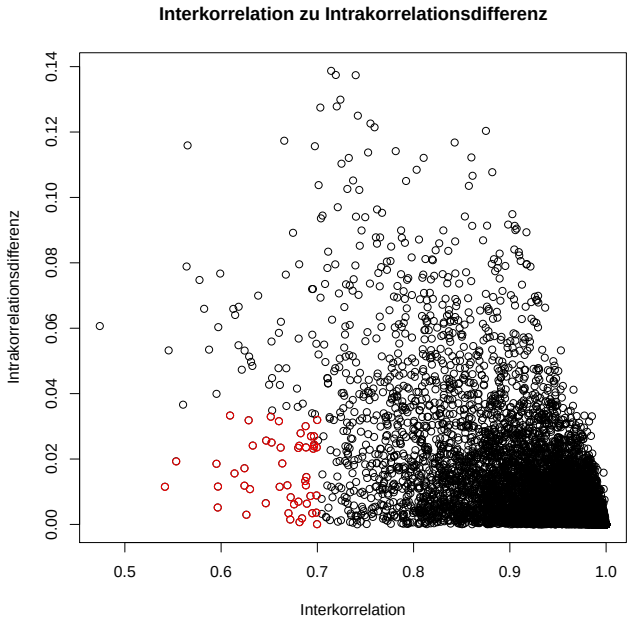


Abbildung 5.18: Der rot markierte Bereich wurde gewählt um zu prüfen, ob die Reliability die von der Intrakorrelation getroffene Aussage, dass Referenz und berechnetes Alignment trotz großem Unterschied zwischen den Alignments ungefähr gleich gut sind, bestätigt

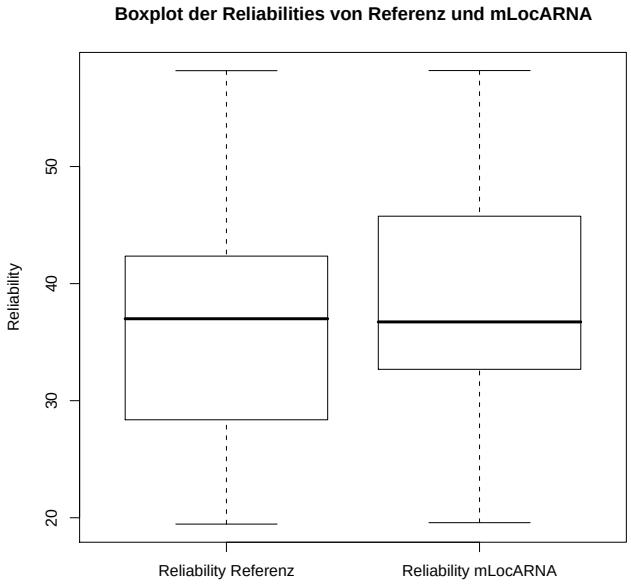


Abbildung 5.19: Boxplot der Reliabilities von mLocARNA und von den Bralibase-Referenzalignments

	p-Wert
KS-Test	0.5441
Wilcox-Test	0.1701

Tabelle 5.4: p-Werte des Kolmogorov-Smirnov-Tests und des Wilcox-Rangsummentest auf der Reliability

Schritt	Mittlere Laufzeit	Maximale Laufzeit
Reliability	101 sek	8133 sek
Intrakorrelation	0.64 sek	9.79 sek

Tabelle 5.5: Mittlere und maximale Gesamtlaufzeit von Reliability und Intrakorrelation bei Bralibase2.1 Alignments mit drei Sequenzen

reich, also sind die Stichproben gleich verteilt. Damit lässt sich schließen, dass, wie erhofft, die Reliability die qualitative Gleichheit dieser Alignments, die die Intrakorrelation gemessen hat, bestätigt.

Bei der Laufzeit kann sich die Intrakorrelation deutlich vor der Reliability einordnen (Tabelle 5.5). Wobei unterschieden werden muss, was an Vorberechnung bereits gemacht wurde. Die Berechnung der Reliability eines Alignments mit LocARNA-P benötigt eine vorberechnete Tabelle mit Spalten-Reliabilities des Alignments, auf deren Basis dann verschiedene Alignments der selben Sequenzen verglichen werden können. Wenn diese Tabelle vorliegt, kann schnell ein Alignment bewertet werden, ansonsten ist mit großem Zeitaufwand zu rechnen. Bei der Intrakorrelation kann die Berechnung durch Bereithalten der Sequenzdotplots signifikant beschleunigt werden, die ansonsten erst noch erzeugt werden müssten. Siehe zum Vergleich Grafik 5.20.

5.5.1 Zusammenfassung

Die Intrakorrelation ist ein neuartiges Maß, um die Güte eines Alignments festzustellen. Sie misst die strukturelle Übereinstimmung zwischen den Sequenzen eines Alignments. Die Struktur bezieht sich dabei auf den ganzen Dotplot, nicht nur auf die eine MFE-Sekundärstruktur. Ein schlechter Wert kann dabei zwei mögliche Gründe haben, entweder die Sequenzen sind relativ unterschiedlich in ihren Faltungspräferenzen oder die Sequenzen sind strukturell unpassend aligniert. Auf den Bralibase2.1-Daten, im direkten Vergleich zu einer Referenz, zeigt sich, dass eine Intrakorrelation von 1 im Allgemeinen nicht erreicht werden kann (Grafik 5.3). Es zeigt sich auch, dass alle Sequenz-Struktur-Programme sehr gut die Referenzkurve nachzeichnen und nur minimale Abweichungen nach oben oder unten zeigen. Auch MAIC kann keine deutliche Veränderung nach oben bewirken, obwohl die Intrakorrelation als Bewertungsfunktion eingesetzt wird (Grafik 5.9). Daraus lässt sich schließen, dass die Sequenz-Struktur-Alignments qualitativ

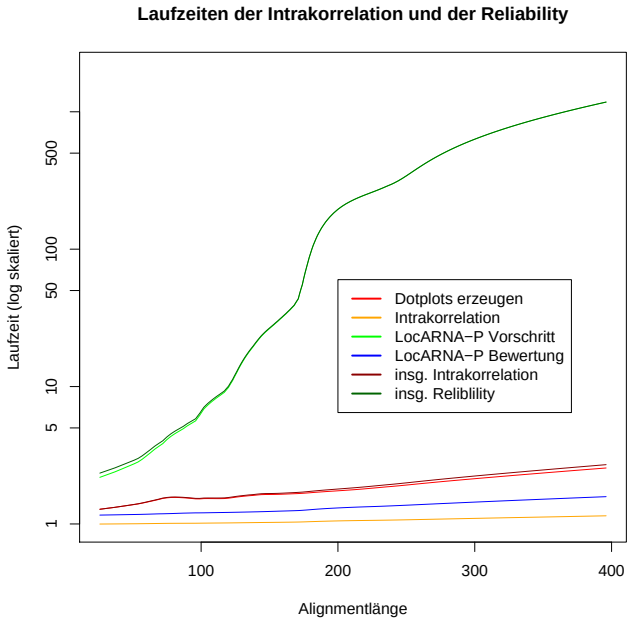


Abbildung 5.20: Laufzeiten der einzelnen Schritte von Reliability und Intrakorrelation bei Bralibase2.1 Alignments mit drei Sequenzen

ähnlich gute Alignments produzieren wie die Referenz. Das zeigt insbesondere auch, dass die Intrakorrelation ein geeignetes Maß für die Bewertung eines Alignments ist. Denn wenn ein Teil der berechneten Alignments eine deutlich höhere Intrakorrelation als die Referenz hätte, könnte dies nur eine Schlussfolgerung zulassen: Entweder die Referenzalignments sind dürftig, oder die Bewertungsmethode ist ungeeignet, denn sie würde dann ein eigentlich schlechteres Alignment besser bewerten als die Referenz. Die Qualität der Referenz-Alignments lässt sich zwangsläufig nicht objektiv bewerten. Es ist allerdings anzunehmen, dass bei der Zusammenstellung der Alignments aus Rfam nur Alignments genommen wurden, die keine offensichtlichen Fehler enthalten. Man kann also davon auszugehen, dass höchstens kleine Verbesserungen gegenüber der Referenz beobachtet werden sollten, wenn die Bewertungsmethode gut ist. Und genau dies ist der Fall.

Im Vergleich mit der Reliability, die mit LocARNA berechnet werden kann, zeigt sich, dass es eine hohe Korrelation zwischen der Intrakorrelation und der Reliability gibt. Das bedeutet, dass die Intrakorrelation ein ähnlich gutes Maß für die Bewertung eines Alignments ist, wie die Reliability und außerdem den Vorteil hat, signifikant schneller berechnet werden zu können.

6 Zusammenfassung der Ergebnisse

Die Ergebnisse dieser Arbeit zusammenfassend kann festgestellt werde, dass alle Sequenz-Struktur-Alignment Programme sehr gute Ergebnisse liefern, wobei mLocARNA das momentan beste der getesten Programme und RAF das schnellste der Programm, die auch die Struktur beachten, ist. Im weiteren konnte gezeigt werden, dass MAIC gut einzusetzen ist, wenn es darum geht, sehr lange oder sehr viele Sequenzen zu alignieren. Diese Szenarien sind mit Sequenz-Struktur-Alignment Programmen aufgrund sehr hoher Rechenzeiten nicht durchführbar. Eine Kombination von MAIC mit einem reinen Sequenz-Alignment-Programm dagegen ist sehr schnell. Der Vorteil gegenüber der Ausführung nur des Sequenz-Alignment-Programms ist, dass strukturelle Elemente trotzdem beachtet werden.
MAIC ist auch im ursprünglich im Thema festgelegten Einsatzfeld gut einsetzbar; wenn man also in einer Situation ist, in der man bereits ein Alignment hat, aber vermutet, dass es von eher schlechter Qualität ist und insbesondere in Bezug auf die Struktur wahrscheinlich nicht korrekt ist, kann man MAIC aufrufen und bekommt in kurzer Zeit ein wahrscheinlich besseres Alignment. Die Analyse konnte zeigen, dass die Alignments von Rfam teilweise auf das Niveau von mLocARNA verbessert werden konnten und zwar in deutlich kürzerer Zeit, als wenn man die Sequenzen ganz neu alignieren lässt.

Die meisten anderen Paper, in denen aktuelle Alignment-Programme verglichen werden, konzentrieren sich nur entweder auf den MCC oder auf SPS und SCI. In dieser Arbeit wurden dagegen alle drei Methoden inklusive zwei neuer Methoden benutzt. Dieses große Feld verschiedener Benchmarks erlaubt es unterschiedliche Aspekte der Programme nebeneinander zu vergleichen. Damit lässt sich eine deutlich fundiertere Aussage über die Qualität der Programme treffen. Außerdem konnte in kritischer Auseinandersetztung mit dem MCC Schwächen dieser Methode erkannt werden, die die Aussagekraft des MCC stark einschränken. Die Intrakorrelation konnte in ihrem Debüt als Benchmark-Methode überzeugen und im Vergleich zur Reliability, einer deutlich komplexer zu berechnenden Bewertung, zeigte sich eine deutliche Korrelation, die die Qualität der Methode unterstreicht.

6.1 Ausblick

Wie die Ergebnisse zeigen, gibt es bei den aktuellen Alignment-Programmen nur noch geringen Bedarf für einfache Verbesserungen der Qualität, wie sie MAIC durchführt. Sequenz-Struktur-Alignments sind schon über den Schritt hinaus, an dem solch ein Nachverarbeitungsschritt eine Verbesserung bewirkt. Jedoch gibt es bei der Geschwindigkeit noch Potential. Eine Kombination von Mafft und MAIC kann in kurzer Zeit auch sehr lange und sehr viele Sequenzen alignieren, während die meisten Sequenz-Struktur-Alignment-Programme hier sehr viel Zeit benötigen. Aber wie man an RAF sieht, wird in dieser Richtung intensiv gearbeitet und es konnten auch schon deutliche Geschwindigkeitssteigerungen erzielt werden.

Ein möglicher Ansatzpunkt für Verbesserungen bei MAIC ist die Anzahl an Gaps, die MAIC einbaut. Eventuell lässt sich durch leichte Veränderungen bewirken, dass weniger Gaps eingefügt werden oder Gaps wieder entfernt werden können und trotzdem die beobachtete Verbesserung der Alignments eintritt. Möglich ist dies zum Beispiel, wenn Gaps so über mehrere benachbarte Spalten verteilt sind, dass, die Spalten zusammen genommen, in jeder Sequenz ein Gap ist. Es wäre also möglich, durch kleine Verschiebungen eine Spalte nur aus Gaps zu bekommen, die dann entfernt werden kann. Der Haken daran und der Grund, dass dies bisher nicht in MAIC eingebaut wurde, ist, dass diese kleinen Verschiebungen möglicherweise genau die Erhöhung der strukturellen Übereinstimmung zwischen den Sequenzen, auf die MAIC abzielt, wieder rückgängig macht.

Großes Potential liegt in der Verwendung der Intrakorrelation als Benchmark Methode. Sie ist sowohl sehr schnell, wie auch theoretisch und praktisch geeignet.

7 Anhang

7.1 Verwendete Software

Alignment-Programme wurden in den folgenden Versionen und mit folgenden Parametern verwendet:

ClustalW [30] Version 1.83; Parameter: Ø

lara [2] Version 1.3.2; Parameter: „-o lara-1.3.2a/lara.params“

Mafft [15] Version 6.624b; Parameter: „–auto –quiet“

MAIC Version 2.0.0 bis 2.0.4; Parameter: „-clean“

mCARNA [24] Version 0.2.2; Parameter: „–time-limit=6000000 –max-diff 20“

mLocARNA [34] Version 1.6.2pre; Parameter: Ø

RAF [4] Version 1.00; Parameter: „predict“

GNU R [25] ist eine Statistik-Software, zu der es eine Reihe von Zusatzpaketen gibt. Die Grafiken in dieser Arbeit wurden mit R erstellt. Es wurde GNU R Version 2.8.1 verwendet (`http://www.r-project.org/`).

Perl ist eine plattformunabhängige Programmiersprache, mit der sich schnell kleine Skripte erstellen lassen. Der sehr einfache Umgang mit regulären Ausdrücken ist eine der Stärken dieser Sprache. Es wurde Version 5.8.8 und 5.12.4 verwendet (`http://www.perl.org/`).

RNAz [31] wurde verwendet um den SCI zu berechnen. Version 1.0 wurde benutzt (`http://www.tbi.univie.ac.at/~wash/RNAz/`)

ViennaPackage [12, 13] ist eine Bibliothek von Programmen, die in der „Theoretische Biochemie“ Gruppe der Universität Wien zusammengestellt wurde. Sie enthält insbesondere RNAfold, mit dem die Dotplots erstellt werden, und RNAalifold, mit dem die Consensustruktur und die minimale freie Energie des Alignments berechnet werden. Es wurde Version 1.8.5 verwendet (`http://www.tbi.univie.ac.at/RNA/`).

Rechnermodell	Anzahl Kerne/Rechner	Anzahl Rechner
Opteron 275/285	8	2
Xeon 5160	4	3
Opteron 2356	8	12

Tabelle 7.1: Verwendete Rechnermodelle und Anzahl der Rechenkerne im Cluster

7.2 Verwendete Hardware

Alle Alignments wurden auf dem Rechnercluster des Bioinformatik-Lehrstuhls der Technischen Fakultät Freiburg berechnet. Das Cluster besteht derzeit aus 17 Rechnern mit zusammen 124 Rechenkernen. Die verschiedenen Modelltypen sind in Tabelle 7.1 aufgelistet.

7.3 Grafikerzeugung

Alle durch Grafiken veranschaulichten Daten sind sehr großer Streuung unterworfen. Um weiche Kurven zu erhalten wurde daher die GNU R Funktion „lowess“ mit einem smoothing-Faktor von $\frac{1}{2}$ verwendet. Zur Veranschaulichung der großen Variation in den Daten ist in Grafik 7.1 beispielhaft die Kurve des Braliscore(SPS*SCI) der mLocARNA Ergebnisse auf der Bralibase Datenbank (rot) zusammen mit den zugrunde liegenden Datenpunkten (blau) eingezeichnet.

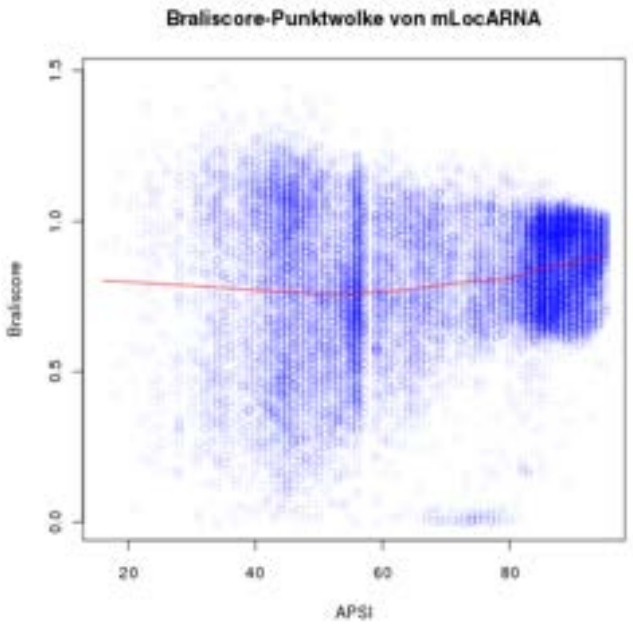


Abbildung 7.1: Braliscore gegen APSI: Vergleich zwischen der gezeichneten Kurve(rot) und den zugrunde liegenden Datenpunkten(blau)

	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
AA	-2.49	-7.04	-8.24	-4.32	-8.84	-14.37	-4.68	-12.64	-6.86	-5.03	-8.39	-5.84	-4.01	-11.32	-6.16	-9.05
AC		-2.11	-8.89	-2.04	-9.37	-9.08	-5.86	-10.45	-9.73	-3.81	-11.05	-4.72	-5.33	-8.67	-6.93	-7.83
AG			-0.80	-5.13	-10.41	-14.41	-4.57	-10.14	-8.61	-5.77	-5.38	-6.60	-5.43	-8.87	-5.94	-11.07
AU				4.49	-5.56	-6.71	1.67	-5.17	-5.33	2.70	-5.61	0.59	1.61	-4.81	-0.51	-2.98
CA					-5.13	-10.45	-3.57	-8.49	-7.98	-5.95	-11.36	-7.93	-2.42	-7.08	-5.63	-8.39
CC						-3.59	-5.71	-5.77	-12.43	-3.70	-12.58	-7.88	-6.88	-7.40	-8.41	-5.41
CG							5.36	-4.96	-6.00	2.11	-4.66	-0.27	2.75	-4.91	1.32	-3.67
CU								-2.28	-7.71	-5.84	-13.69	-5.61	-4.72	-3.83	-7.36	-5.21
GA									-1.05	-4.88	-8.67	-6.10	-5.85	-6.63	-7.55	-11.54
GC										5.62	-4.13	1.21	1.60	-4.49	-0.08	-3.90
GG											-1.98	-5.77	-5.75	-12.01	-4.27	-10.79
GU												3.47	-0.57	-5.30	-2.09	-4.45
UA													4.97	-2.98	1.14	-3.39
UC														-3.21	-4.76	-5.97
UG															3.36	-4.28
UU																-0.02

Tabelle 7.2: Ribosum-Matrix für Basenpaare

	A	C	G	U	-	X
A	2.22	-1.86	-1.46	-1.39	-1	0
C	-1.86	1.16	-2.48	-1.05	-1	0
G	-1.46	-2.48	1.03	-1.74	-1	0
U	-1.39	-1.05	-1.74	1.65	-1	0
-	-1	-1	-1	-1	0	0
X	0	0	0	0	0	0

Tabelle 7.3: Ribosum-Matrix für einzelne Basen

7.4 Ribosum-Matrizen

Die in dieser Arbeit verwendeten Abwandlungen der ursprünglichen Ribosum-Martrizen[17] sind in Tabelle 7.2 und 7.3 zu sehen. Tabelle 7.2 wird von MAIC für die Bewertungen von vollständigen Basen-Quadrupeln benutzt, während Tabelle 7.3 von MAIC benutzt wird, wenn das zu bewertende Basen-Quadtrupel ein oder mehrere Gaps enthält. Im Unterschied zur ursprünglichen Ribosum-Matrix wurde Tabelle 7.3 um eine Zeile und Spalte für Gaps („-“) und eine Zeile und Spalte „X“, als Wildcard für alle sonstigen Zeichen im Alignment, erweitert.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Ebbe S. Andersen, Allan Lind-Thomsen, Bjarne Knudsen, Susie E. Kristensen, Jakob H. Havgaard, Elfar Torarinsson, Niels Larsen, Christian Zwieb, Peter Sestoft, Jørgen Kjems, and Jan Gorodkin. Semiautomated improvement of RNA alignments. *RNA*, 13(11):1850–1859, November 2007.

[2] M. Bauer, G. W. Klau, and K. Reinert. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics*, 8(271), 2007.

[3] Stephan H. Bernhart, Ivo L. Hofacker, Sebastian Will, Andreas R. Gruber, and Peter F. Stadler. RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, 9:474, 2008.

[4] Chuong B. Do, Chuan-Sheng S. Foo, and Serafim Batzoglou. A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics (Oxford, England)*, 24(13), July 2008.

[5] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.

[6] Da-Fei Feng and Russell Doolittle. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987. 10.1007/BF02603120.

[7] Eva K. Freyhult, Jonathan P. Bollback, and Paul P. Gardner. Exploring genomic dark matter: A critical assessment of the performance of homology search methods on noncoding RNA. *Genome Research*, 17(1):117–125, January 2007.

[8] P. P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5(1):140, 2004.

[9] P. P. Gardner, A. Wilm, and S. Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res*, 33(8):2433–9+, 2005.

[10] Jan Gorodkin and Ivo L Hofacker. From structure prediction to genomic screens for novel non-coding rnas. *PLoS computational biology*, 7(8):e1002100, 2011.

[11] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*, 33(Database issue):D121–4, January 2005.

[12] Ivo Hofacker and Peter Stadler. Vienna RNA package. Paper as Print Copy, 1998.

[13] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, Sebastian L. Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, 125:167–188, 1994.

[14] Kazutaka Katoh and Hiroyuki Toh. Improved accuracy of multiple ncRNA alignment by incorporating structural information into a MAFFT-based framework. *BMC Bioinformatics*, 9(1):212, 2008.

[15] Kazutaka Katoh and Hiroyuki Toh. Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform*, 9(4):286–298, July 2008.

[16] Hisanori Kiryu, Yasuo Tabei, Taishin Kin, and Kiyoshi Asai. Murlet: a practical multiple alignment tool for structural rna sequences. *Bioinformatics*, 23(13):1588–1598, 2007.

[17] Robert Klein and Sean Eddy. Rsearch: Finding homologs of single structured rna sequences. *BMC Bioinformatics*, 4(1):44+, September 2003.

[18] Stinus Lindgreen, Paul P. Gardner, and Anders Krogh. MASTR: multiple alignment and structure prediction of non-coding RNAs using simulated annealing. *Bioinformatics (Oxford, England)*, 23(24):3304–3311, December 2007.

[19] J. S. Mattick. Challenging the dogma: the hidden layer of non-protein-coding rnas in complex organisms. *BioEssays*, 25(10):930–939, 2003.

[20] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for rna secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990.

[21] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[22] C. Notredame, D. G. Higgins, and J. Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, September 2000.

[23] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35(1):68–82, 1978.

[24] Alessandro Dal Palu, Mathias Möhl, and Sebastian Will. Alignnment of RNA with structures of unlimited complexity. In *Proceedings of the Workshop on Constraint Based Methods for Bioinformatics (WCB 2010)*, page 7, 2010.

[25] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[26] Jens Reeder and Robert Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5(1):104, 2004.

[27] Rfam-Team. Readme rfam 10.1. `ftp://ftp.sanger.ac.uk/pub/databases/Rfam/10.1/README`, June 2011.

[28] David Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.

[29] Yasuo Tabei, Hisanori Kiryu, Taishin Kin, and Kiyoshi Asai. A fast structural multiple alignment method for long RNA sequences. *BMC Bioinformatics*, 9(1):33, 2008.

[30] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[31] Stefan Washietl, Ivo L. Hofacker, and Peter F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459, February 2005.

[32] M. Waterman, T. Smith, and W. Beyer. Some biological sequence metrics. *Advances in Mathematics*, 20(3):367–387, June 1976.

[33] Sebastian Will, Tejal Joshi, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. LocARNA-P: Accurate boundary prediction and improved detection of structured RNAs. not yet published, 2010.

[34] Sebastian Will, Kristin Reiche, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Computational Biology*, 3(4):e65, 2007.

[35] Andreas Wilm, Kornelia Linnenbrink, and Gerhard Steger. ConStruct: Improved construction of RNA consensus structures. *BMC bioinformatics*, 9:219+, April 2008.

[36] Andreas Wilm, Indra Mainz, and Gerhard Steger. An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms for Molecular Biology*, 1(1):19, 2006.

[37] Michael Zuker and Patrick Stiegler. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.