Friedrich-Schiller-
Universität
Jena

Fakultät für Mathematik und Informatik

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Bioinformatiker

MuLoRA

Ein Ansatz für multiple, lokale RNA-Sequenz-Struktur-Alignments

Verfasser:	Wolfgang Otto
Aufgabensteller:	Prof. Dr. Rolf Backofen
Betreuer:	Prof. Dr. Rolf Backofen
	Dipl.-Math. Sven Siebert
Abgabetermin:	3. September 2005

Zusammenfassung

Für lange Zeit galten Ribonukleinsäuren in der Wissenschaft hauptsächlich als Botenstoffe bei der Umwandlung genetischer Informationen in Proteine. Doch mit der Entdeckung einer Vielzahl neuer Funktionen wanderten die RNAs in den letzten Jahren immer mehr in den Blickpunkt der Forschung, wobei vor allem nicht kodierenden RNAs und untranslatierten Bereichen in mRNAs bedeutende Rollen zukommen.

Die Funktionen der Ribonukleinsäuren werden dabei meist von konservierten Strukturen bestimmt, weshalb die Suche nach solchen Strukturbereichen mit Hilfe von multiplen Alignments in der Bioinformatik eine bedeutende Rolle spielt. Da Strukturen in RNAs aber nur selten auch auf Sequenzebene konserviert sind, muss diese Suche auch auf Strukturebene durchgeführt werden, wobei man sich aus Komplexitätsgründen auf die Sekundärstruktur beschränkt.

Allerdings können für einzelne RNA-Sequenzen viele thermodynamisch plausible Sekundärstrukturen vorhergesagt werden von denen fast alle keine funktionelle Relevanz besitzen. Deshalb reicht es bei der Suche nach Strukturmotiven nicht aus, nur die energetisch günstigsten Strukturen zu verwenden. Vielmehr sollten dabei alle plausiblen Strukturen betrachtet werden.

Ein weiterer wichtiger Punkt besteht darin, dass die gleichen strukturellen Motive in ansonsten vollkommen anderen strukturellen Umfeldern vorkommen, weshalb die Suche nach Motiven auch lokal sein sollte. Da ein Strukturmotiv aber aus mehreren unzusammenhängenden Teilsequenzen bestehen kann, ist dabei eine besondere strukturelle Form der Lokalität notwendig.

Da es bis jetzt keine Ansatz für ein multiples Sequenz-Struktur-Alignment mit einer auf Strukturen zugeschnittener Form von Lokalität gab, habe ich mich in meiner Arbeit diesem Problem angenommen und **MuLoRA** – einen Ansatz für **mu**ltiple, **lo**kale **R**NA-Sequenz-Struktur-**A**lignments – entwickelt.

Für die Berechnung eines multiplen Alignments aus einer Menge von Sequenzen verwende ich dabei einen progressiven Ansatz, welcher für jeden einzelnen Alignmentschritt die lokalen Sequenz-Struktur-Informationen innerhalb aller Sequenzen berücksichtigt.

Um diese Informationen zu erhalten, habe ich einen Algorithmus für paarweise lokale Alignments entwickelt, der für alle Paare der Eingabesequenzen die Teilstrukturen mit maximaler Ähnlichkeit in beiden Sequenzen sucht. Die Grundlage für die Bewertung der strukturellen Ähnlichkeit bilden dabei thermodynamische Informationen in Form von Basenpaarwahrscheinlichkeiten.

Aus dem so gewonnenen multiplen Alignment leite ich schließlich die Konsensussequenz und – wieder unter Verwendung der Basenpaarwahrscheinlichkeiten – die Konsensusstruktur ab und erhalte so die in den Sequenzen konservierte Struktur.

Der so entstandene Ansatz arbeitet effizient und zeigt bei Testdurchläufen sehr gute Ergebnisse.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Multiple Alignments . . . 3
  - 1.1.1 Bewertungsfunktionen . . . 5
  - 1.1.2 Berechnungsmethoden . . . 7
  - 1.1.3 Sequenz-Struktur-Ansätze . . . 8
- 1.2 Verwandte Arbeiten . . . 9
  - 1.2.1 Der Sankoff-Algorithmus . . . 9
  - 1.2.2 MARNA . . . 10
  - 1.2.3 pmmulti . . . 11
  - 1.2.4 RNA-forester . . . 12
- 1.3 Übersicht . . . 13

2 Vorbetrachtungen **15**
- 2.1 Formale Definitionen . . . 15
  - 2.1.1 Sequenz und Struktur . . . 15
  - 2.1.2 Alignments und Lokalität . . . 18
- 2.2 Problemstellungen . . . 23
  - 2.2.1 Das paarweise lokale Alignment-Problem . . . 24
  - 2.2.2 Die Konsensus-Probleme . . . 25

3 Der MuLoRA Ansatz **27**
- 3.1 Überblick . . . 27
- 3.2 Bestandteile . . . 30
  - 3.2.1 Basenpaarwahrscheinlichkeiten . . . 31
  - 3.2.2 Paarweise lokale Seqenz-Struktur-Alignments . . . 32
  - 3.2.3 Multiples Alignment . . . 40
  - 3.2.4 Konsensussequenz und -Struktur . . . 42
- 3.3 Komplexitätsbetrachtungen . . . 44
  - 3.3.1 Paarweiser lokaler Alignmentalgorithmus . . . 44
  - 3.3.2 Konsensus-Sequenz-Struktur-Algorithmus . . . 47
  - 3.3.3 Gesamtüberblick . . . 48

4 Ergebnisse **51**
- 4.1 Parameterabschätzung . . . 52
  - 4.1.1 Parametertraining . . . 52
  - 4.1.2 Basenpaarwahrscheinlichkeitsanalyse . . . 54
- 4.2 Anwendungen . . . 55
  - 4.2.1 Laufzeitanalyse . . . 57
  - 4.2.2 Konservierte Sequenzen . . . 59
  - 4.2.3 Unkonservierte Sequenzen . . . 60
  - 4.2.4 Motivsuche . . . 62

5 Zusammenfassung und Ausblick **63**

A Ergebnistabellen **65**
A.1 Konservierte Sequenzen 65
A.2 Unkonservierte Sequenzen 70

Literaturverzeichnis **73**

Kapitel 1

Einleitung

Für lange Zeit galt die Ribonukleinsäure in der Wissenschaft hauptsächlich als ein Bote für die DNA, bei der Umwandlung genetischer Informationen in Proteine. Doch mit der Entdeckung einer Vielzahl neuer Funktionen wanderte die RNA in den letzten Jahren immer mehr in den Blickpunkt der Forschung. Nicht-kodierende RNAs (*ncRNAs*) spielten dabei eine bedeutende Rolle [Edd01], [Sto02]. Anstatt translatierter Proteine produzieren sie funktionelle RNAs, welche in der Zelle an vielen wichtigen Prozessen beteiligt sind.

- Ribonukleinsäuren sind Bestandteile wichtigster zellulärer Komponenten. Zu diesen gehören das *Ribosom*, das *Spliceosom* und die *Telomerase*.
- Die *tRNAs* gewährleisten die Übersetzung des genetischen Codes.
- RNA-Enzyme, die *Ribozyme*, katalysieren Phosphatgruppentransfers und Peptidbindungsformationen. Zu ihnen gehören unter anderen *self-cleaving RNAs*, *self-splicing RNAs*, sowie *ribosomale RNAs*. [FW05].
- Kleine RNA-Moleküle – *smallRNAs* – sind an der Inhibition von Genen und der Epigenese beteiligt. Zwei wesentliche Gruppen von smallRNAs sind dabei *microRNAs* und *small interfering RNAs* [Bar04].

Aber auch die Bereiche innerhalb der mRNAs enthalten Elemente, die bei der Kontrolle der Proteinexpression wichtige Aufgaben übernehmen.

- *Riboswitch-Elemente*, auch als *Regulons* bezeichnet, binden an kleine Moleküle und steuern so die Aktivität der mRNA in Abhängigkeit dieser Moleküle [WCB02] oder Umgebungsbedingungen [NHB01].
- *SECIS-Elemente* steuern den Einbau der einundzwanzigsten proteinogenen Aminosäure Selenocystein [WSP97].
- *IRES-Elemente* beeinflussen den Einstiegspunkt des Ribosoms und damit den Translationsbeginn [KH05].
- *Iron-Response-Elemente* regulieren die translationale Effizienz in Abhängigkeit des Eisengehalts der Zelle [HK96].

Aufgrund dieser Entdeckungen und der damit gestiegenen Bedeutung der Ribonukleinsäuren wurde die Erforschung der RNA vom Science-Magazine zum wissenschaftlichen Durchbruch im Jahr 2002 gewählt [Cou02].

Wie bei Proteinen liegt der Schlüssel für diese Vielzahl von Funktionen in der Struktur, weshalb die automatische Suche nach funktionellen Strukturbereichen in der Bioinformatik eine bedeutende Rolle spielt. Da funktionelle Strukturen einem

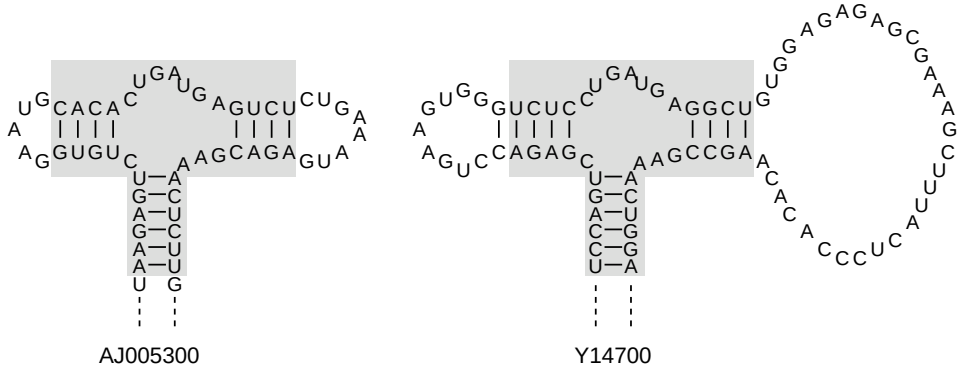

Abbildung 1.1: Zwei Vertreter der Typ III Hammerhead Ribozyme [PFM94]. AJ005300 ist ein Peach latent mosaic Viroid, während Y14700 aus einem Streptomyces Viridochromogen stammt. Hammerhead-Ribozyme gehören zu den kleinsten enzymatischen RNAs. Sie besitzen eine Endonuklease-Funktion und sind meist autokatalytisch. Der funktionelle Bereich ist grau hinterlegt.

hohen Selektionsdruck ausgesetzt sind, versucht man dazu mit Hilfe multipler Alignments konservierte Strukturen zu finden.

Dabei ist allerdings zu beachten, dass die gleiche Struktur von vielen verschiedenen Sequenzen ausgebildet werden kann. Abbildung 1.1 zeigt beispielsweise zwei Hammerhead Ribozyme. Obwohl sie beide die gleiche funktionelle Struktur besitzen (grau hinterlegt), sind sie durch komplementäre Basenaustausche sequenziell unterschiedlich.

Deshalb ist es im Gegensatz zu Proteinen wichtig, die Suche nach Strukturen auch auf Strukturebene und nicht nur auf Sequenzebene durchzuführen. Problematischerweise ist jedoch die Anzahl der strukturellen Freiheitsgrade in Polynukleotiden höher als in Polypetitden, wodurch das vollständige Strukturvorhersageproblem mindestens so schwer wie das Proteinfaltungsproblem ist. Aus diesem Grund beschränkt man sich im Allgemeinen auf die Sekundärstruktur, welche nur die Bindungen zwischen den Basen betrachtet.

Für einzelne RNA-Sequenzen können allerdings viele thermodynamisch plausible Sekundärstrukturen vorhergesagt werden, wobei fast alle keinerlei funktionelle Relevanz besitzen. Deshalb reicht es bei der Suche nach Strukturmotiven nicht aus, nur die energetisch günstigsten Strukturen mit den niedrigsten freien Energien[1] zu verwenden. Vielmehr sollten dabei alle plausiblen Strukturen betrachtet werden.

Ein weiteres Problem besteht darin, dass die gleichen strukturellen Motive in ansonsten vollkommen unverwandten Molekülen und damit auch in einem vollkommen anderen strukturellem Umfeld vorkommen. Deshalb sollte die Suche nach solchen Motiven auch lokal sein. Da ein Strukturmotiv aber nicht zwingend aus einer zusammenhängenden Teilsequenz besteht – die konservierten funktionellen Strukturbereiche in Abbildung 1.1 werden beispielsweise durch sowohl in der Länge als auch in der Sequenz unterschiedliche Loops unterbrochen – ist es nicht ausreichend, die Form von Lokalität zu übernehmen, wie sie bei reinen Sequenzen verwendet wird.

Damit ergeben sich für die Suche nach funktionellen Motiven in Ribonukleinsäuren insgesamt folgende Anforderungen: Erstens müssen bei der Suche mehrere Moleküle verglichen werden um konservierte Bereiche zu entdecken. Zweitens müssen dabei sowohl die Sequenz als auch mögliche Strukturen in Betracht gezogen werden um die fehlende Konserviertheit auf Sequenzebene auszugleichen. Und

[1] Die freie Energie wird im Deutschen auch als Gibbssche Freie Enthalpie bezeichnet.

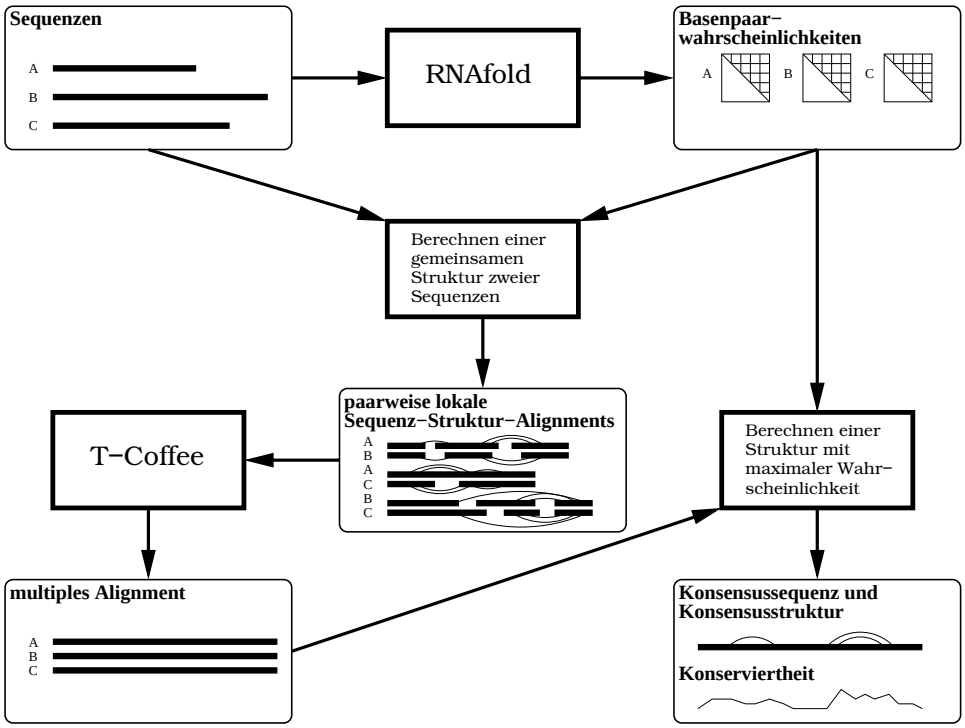

Abbildung 1.2: Schematische Darstellung des MuLoRA-Ablaufs. Quadratische Blöcke repräsentieren Berechnungen, während abgerundete Blöcke Datenstrukturen zeigen.

drittens sollten diese Sequenz-Struktur-Vergleiche in einer bestimmten strukturellen Art und Weise lokal sein, um so nur die Bereiche zu betrachten, welche konserviert sind.

In dieser Arbeit stelle ich einen Ansatz vor, der erstmalig all diesen Anforderungen gerecht wird. **MuLoRA** – einen Ansatz für **mu**ltiple, **lo**kale **R**NA-Sequenz-Struktur-**A**lignments – berechnet für eine Menge von Eingabesequenzen ein multiples Sequenz-Struktur-Alignment mit einer auf Strukturen zugeschnittener Form von Lokalität.

Für die Berechnung eines solchen multiplen Alignments verwende ich einen progressiven Ansatz, welcher für jeden einzelnen Alignmentschritt die lokalen Sequenz-Struktur-Informationen aller Eingabesequenzen berücksichtigt. Diese Informationen stammen aus paarweisen lokalen Alignments, welche für alle Sequenzpaare die Teilstrukturen mit maximaler Ähnlichkeit in beiden Sequenzen suchen. Die Grundlage für die Bewertung der strukturellen Ähnlichkeit bilden dabei die Wahrscheinlichkeiten der in der Struktur enthaltenen Basenpaare. Aus dem so gewonnenen multiplen Alignment wird schließlich die Konsensussequenz und – wieder unter Verwendung der Basenpaarwahrscheinlichkeiten – die Konsensusstruktur ermittelt. Abbildung 1.2 fasst das Verfahren noch einmal zusammen.

1.1 Multiple Alignments

In einen multiplen Alignment sind eine Menge von Sequenzen mit Hilfe eines neutralen Gap-Elementes '−' so angeordnet, dass homologe Basen in gemeinsamen Spalten stehen. Abbildung 1.3 zeigt beispielsweise ein multiples Alignment der Sequenzen von vier Iron-Response-Elementen. Die Homologie der Basen in einer Spalte soll-

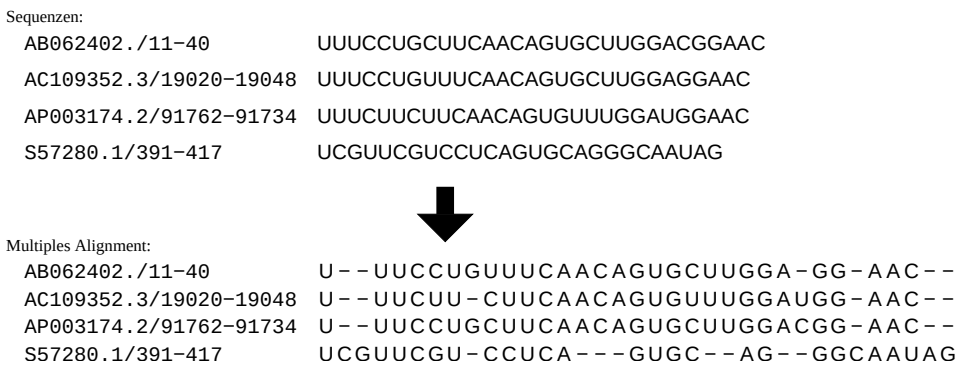

Abbildung 1.3: Multiples Alignment von vier Iron-Response-Elementen. Das Alignment beruht dabei nur auf den Sequenzinformationen und vernachlässigt eventuell enthaltene Strukturen komplett.

te sich dabei sowohl auf die Struktur, als auch auf den evolutionären Hintergrund beziehen. Idealerweise würden also die Basen innerhalb einer Spalte ähnliche strukturelle Positionen aufweisen und von einem gemeinsamen Vorfahren abstammen.

Die Aufgabe von multiplen Alignments besteht also im Organisieren, Visualisieren und Analysieren mehrerer Sequenzen. Damit spielen sie als Eingabe für viele nachfolgende Methoden der Datenanalyse eine wichtige Rolle, weshalb ihr Anwendungsbereich in der Biologie fast keine Grenzen kennt.

Allgemein könnte man multiple Alignments als eine Erweiterung der paarweisen Alignments betrachten. Alles was paarweise Alignments können, können auch multiple Alignments – allerdings um einiges besser, da sie viel genauere Informationen über die Verteilung an einzelnen Positionen liefern. So ist es beispielsweise möglich, dass gemeinsame funktionelle Bereiche in Proteinfamilien erst durch die Verstärkung eines multiplen Alignments klar erkennbar werden, da die gegenseitigen Ähnlichkeiten für sich zu schwach sind, um durch ein paarweises Alignment hervorgehoben zu werden. Oder um es in den anschaulichen Worten des Molekularbiologen Arthur M. Lesk auszudrücken: “One or two homologous sequences whisper – a full multiple alignment shouts out loud.”

Darüber hinaus bieten multiple Alignments aber noch Fähigkeiten, die über eine bloße Verallgemeinerung der paarweisen Alignments hinausgehen. Paarweise Alignments werden in der Regel verwendet, um für neue Sequenzen mit unbekannter Funktion möglichst ähnliche mit bekannter Funktion zu finden. Da man annimmt, dass ähnliche Sequenzen auch ähnliche Funktionen nachsichziehen, schließt man so auf die Funktion der neuen Sequenz.

Will man hingegen diesen Schluss umkehren, kommt man mit paarweisen Alignments nicht weiter. Es gibt jede Menge Fälle, in denen sehr unterschiedliche Sequenzen trotz allem eine ähnliche Funktion aufweisen. Hier können nun multiple Alignments ihre Stärke ausspielen. Hat man mehrere Sequenzen, die trotz großer syntaktischer Unterschiede funktionsverwandt sind, versucht man mit multiplen Alignments herauszufinden, wo gemeinsame Kernbereiche hoher Ähnlichkeit liegen. Diese Kernbereiche könnten die essentiellen Teile der Moleküle sein, über die sich die Funktion definiert. So gesehen entsprechen multiple Alignments eher den Zielsetzungen der Abstraktion, Regelfindung, Generalisierung oder Erklärung.

Neben der Identifikation von funktionsrelevanten und evolutionär konservierten Merkmalen gibt es jedoch noch eine Vielzahl von anderen Anwendungsmöglichkeiten. So bilden multiple Alignments auch die Grundlage für die Ableitung evolutionärer Historie aus DNA- oder Proteinsequenzen wie das Abschätzen von evolutionären Distanzen oder die Suche nach Hinweisen für Selektion. Die Verteilungs-

informationen der einzelnen Positionen ermöglichen die Charakterisierung und Repräsentation von Proteinfamilien und damit auch profilbasierte Datenbanksuchen. Der Mutual Information Content der Alignmentspalten liefert wichtige Informationen zur Struktur von Polypetiden bzw. -Nukleotiden und erst multiple Alignments ermöglichen ein überdeckendes Zusammenfügen der shot-gun Bruchstücke der Genomsequenzierung.

Für all diese Anwendungen spielen zwei Eigenschaften multipler Alignments eine wesentliche Rolle, nämlich die Qualität und der Ressourcenbedarf für deren Erstellung. Qualitativ hochwertige multiple Alignments werden von Biologen – komplett oder teilweise – per Hand erstellt. Dazu verwenden sie Expertenwissen über Sequenzevolution, welches aus viel Erfahrung stammt. Wichtige Faktoren beinhalten dabei die spezifischen Spaltenarten innerhalb von Alignments wie hoch konservierte Bereiche die sich mit erwarteten Insertions- und Deletionsmustern abwechseln, den Einfluss von Strukturelementen sowie die phylogenetischen Beziehungen innerhalb der Sequenzen. Der Nachteil dieser Prozedur besteht in ihrer Langwierigkeit. Deshalb unterliegen automatische multiple Alignments einer extensiven Erforschung in der Bioinformatik.

Die erste Aufgabe für einen erfolgreichen automatischen Ansatz besteht darin, eine Bewertungsfunktion zu entwickeln, so dass bessere Alignments einen besseren Score erhalten.

1.1.1 Bewertungsfunktionen

Das Ziel bei der Entwicklung einer Bewertungsfunktion ist es, soviel Expertenwissen wie möglich einfliesen zu lassen. Damit ergeben sich zwei Anforderungen: Da bestimmte Positionen stärker konserviert sind als andere, sollte die Bewertung einerseits positionsspezifisch sein und da die einzelnen Sequenzen durch einen phylogenetischen Baum verwandt sind, sollten sie andererseits nicht als unabhängig voneinander betrachtet werden.

Eine ideale Bewertungsmethode für multiple Alignments wäre deshalb ein komplett auf Wahrscheinlichkeiten basierendes Modell. Bei einem gegebenen phylogenetischen Baum würde sich die Wahrscheinlichkeit für ein multiples Alignment dann aus dem Produkt aller Wahrscheinlichkeiten der für die Erzeugung des Alignments notwendigen evolutionären Ereignisse multipliziert mit der Wahrscheinlichkeit der Ausgangssequenz ergeben. Gute strukturelle und evolutionäre Alignments wären dann diejenigen, die eine hohe Wahrscheinlichkeit aufweisen.

Leider wäre so ein Modell ziemlich komplex, denn sowohl die Wahrscheinlichkeiten der evolutionären Ereignisse als auch die positionsspezifischen Einschränkungen aufgrund der durch natürliche Selektion hervorgerufene Konserviertheit von funktionellen und strukturellen Elementen, wären von den evolutionären Distanzen zwischen allen Knoten des phylogenetischen Baumes abhängig.

Unglücklicherweise existieren jedoch nicht genügend Daten um so ein komplexes Modell zu parametrisieren. Und während beispielsweise für den Einfluss der Struktur unabhängige Bezugspunkte wie die Röntgen-Kristall-Analysen oder die NMR-Spektroskopie für die Gewinnung von neuen Daten existieren, gibt es für die Bestimmung der evolutionäre Geschichte einer Sequenzfamilie keine unabhängigen Quellen. Aus diesem Grund ist es notwendig, vereinfachende Annahmen zu machen.

In den meisten Fällen werden die Spalten eines Alignments als statistisch unabhängig betrachtet. Dadurch vereinfacht sich die Bewertungsfunktion zu einer Summe über alle Spaltengewichte.

Sei A^m ein multiples Alignment von m Sequenzen, $A^m[\,i\,]$ die i-te Spalte von A^m und χ_c eine Funktion, welche jeder Spalte aus A^m ein Gewicht zuordnet. Dann

ergibt sich die Bewertungsfunktion s wie folgt:

$$s\left(A^m\right) = \sum_i \chi_c\left(A^m\left[\,i\,\right]\right)$$

Eine weitere Vereinfachung besteht darin, den phylogenetischen Baum, dem die Sequenzen zugrunde liegen, zu vernachlässigen. Die evolutionäre Geschichte wird so nur indirekt aus den Sequenzdaten abgeleitet. Dadurch ist es zwar verhältnismäßig einfach, multiple Alignments von sehr ähnlichen Sequenzen zu erstellen aber auch unmöglich, ein eindeutig richtiges Alignment für entfernte Sequenzen – und damit für die interessanten Fälle – zu berechnen.

Als Beispiele für die vielen existierenden Bewertungsfunktionen sollen hier zwei Standartmethoden der Bioinformatik dienen.

Sum Of Pairs

Auch diese Methode verwendet keinen phylogenetischen Baum und nimmt eine statistische Unabhängigkeit der Spalten an. Der Sum-of-pairs-Score einer Spalte ist dann als Summe der Substitutiosscores über alle Paare von Einträgen innerhalb der Spalte definiert.

Sei A^m ein multiples Alignment von m Sequenzen, $A^m[\,g\,][\,i\,]$ der Eintrag der g-ten Zeile in der i-ten Spalte von A^m und χ_p eine Funktion, welche zwei Einträgen aus A^m ein Gewicht zuordnet. Dann ergibt sich die Bewertungsfunktion s_{sop} wie folgt:

$$s_{sop}\left(A^m\right) = \sum_i \sum_{g<h} \chi_p\left(A^m\left[\,g\,\right]\left[\,i\,\right], A^m\left[\,h\,\right]\left[\,i\,\right]\right)$$

Auf dem ersten Blick scheint das Addieren der paarweisen Substitutionsscores einer Spalte eine ganz natürliche Bewertung zu sein. Da aber Werte für Substitutionsscores wie etwa *PAM* oder *BLOSUM* aus der Annahme heraus abgeleitet wurden, dass die Sequenzen eines Alignments von einer gemeinsamen Vorgängersequenz abstammen, müsste diese Annahme auch für die Bewertung eines multiplen Alignments verwendet werden. Bei der Sum-of-pairs-Methode wird jedoch jede Sequenz so bewertet, als ob sie von den anderen Sequenzen abstammt, wodurch eine Überbewertung der evolutionäre Ereignisse entsteht.

Alignment entlang eines Baumes

Diese Methode geht von einen gegebenen phylogenetischen Baum zu den alignierten Sequenzen aus. Der Score eines multiplen Alignments ergibt sich dann aus der Summe der paarweisen Alignmentscores der im Baum benachtbarten Sequenzen.

Sei A^m ein multiples Alignment von m Sequenzen, T ein phylogenetischer Baum dieser m Sequenzen und $p\left(S_g, S_h\right)$ der Score eines paarweisen Alignments der Sequenzen S_g und S_h. Dann ergibt sich die Bewertungsfunktion s_{tree} wie folgt:

$$s_{tree}\left(A^m\right) = \sum_{(g,h)\in T} p\left(S_g, S_h\right)$$

Diese Kostenfunktion hat den Vorteil, dass man effizient zu jedem phylogenetischen Baum ein konsistentes multiples Alignment berechnen kann.

Die nächste Aufgabe für die Entwicklung eines automatischen Ansatz besteht nun darin, auf Grundlage der Bewertungsfunktion ein multiples Alignment zu berechnen bzw. zu optimieren.

1.1.2 Berechnungsmethoden

Der erste Ansatz würde nun darin bestehen, nach dem Vorbild des Needleman-Wunsch-Algorithmus für paarweise Alignments [NW70], ein optimales multiples Alignment zu berechnen.

Multidimensionales dynamisches Programieren

Die einzige Vorraussetzung für diesen Ansatz ist eine Bewertungsfunktion, die auf einer statistischen Unabhängigkeit der Spalten beruht. Dann ist es verhältnismäßig einfach, eine Rekursionsgleichung für $M(i_1, i_2, \ldots, i_m)$, den maximalen Score eines Alignments der Teilsequenzen $S_1[1, \ldots, i_1]$, $S_2[1, \ldots, i_2]$ bis $S_m[1, \ldots i_m]$, aufzustellen.

Sei χ_m eine Funktion, welche m Einträgen aus A^m einen Substitutionsscore zuordnet. Dann lässt sich M wie folgt definieren:

$$M(i_1, i_2, \ldots, i_m) = \begin{cases} M(i_1 - 1, i_2 - 1, \ldots, i_m - 1) + \chi(i_1, i_2, \ldots, i_m) \\ M(i_1, i_2 - 1, \ldots, i_m - 1) + \chi(-, i_2, \ldots, i_m) \\ M(i_1 - 1, i_2, \ldots, i_m - 1) + \chi(i_1, -, \ldots, i_m) \\ \qquad\qquad\qquad\qquad \vdots \\ M(i_1 - 1, i_2 - 1, \ldots, i_m) + \chi(i_1, i_2, \ldots, -) \\ M(i_1, i_2, \ldots, i_m - 1) + \chi(-, -, \ldots, i_m) \\ \qquad\qquad\qquad\qquad \vdots \\ M(i_1, i_2 - 1, \ldots, i_m) + \chi(-, i_2, \ldots, -) \\ \qquad\qquad\qquad\qquad \vdots \end{cases}$$

Allerdings ist dieser Ansatz für die meisten Bewertungsfunktionen NP-hart [Jus01]. Deshalb ist es beim heutigen Stand der Technik trotz verschiedener Branch-and-Bound Techniken recht zeitaufwendig, mehr als zehn längere Sequenzen optimal zu alignieren. Will man mehr, muss man auf heuristische Verfahren zurückgreifen.

Progressive Alignmentverfahren

Diese Ansätze sind die wohl am häufigsten verwendeten für die Erstellung eines multiplen Alignments. Sie beruhen auf der Idee der Alignment entlang eines Baumes-Berechnungsfunktion, indem sie das multiple Alignment durch eine Folge von paarweisen Alignments erstellen. Da sie die Bewertung des Alignments nicht von dem Optimierungsalgorithmus trennen und damit keine globale Bewertungsfunktion für die Korrektheit eines Alignments optimieren, sind sie heuristisch.

Die verschiedenen Ansätze unterscheiden sich dabei in der Art, wie die Reihenfolge der paarweisen Alignments bestimmt wird, sowie bei der Berechnung und Bewertung der einzelnen Alignmentschritte. Ausserdem erlauben einige Ansätze das Alignieren von zuvor berechneten Alignments gegeneinander, während andere nur einzelne Sequenzen gegen Alignments alignieren dürfen.

Als Beispiel soll hier einer der ersten Algorithmen für progressive Alignments dienen, der Feng-Doolittle-Algorithmus [FD87]. Dieser berechnet zuerst mit Hilfe von paarweisen Alignments eine Diagonalmatrix der $m(m-1)/2$ Distanzen zwischen allen Paaren der m zu alignierenden Sequenzen. Aus den Distanzen wird dann mit einem Cluster-Algorithmus ein einfacher phylogenetischer Baum abgeleitet, der als Leitfaden für die nun folgenden $m - 1$ progressiven Alignments dient.

Der zuerst in den Baum eingefügte Knoten wird mit seinen Kindern in der Reihenfolge der Distanzen aligniert, danach folgt der zweite in den Baum eingefügte

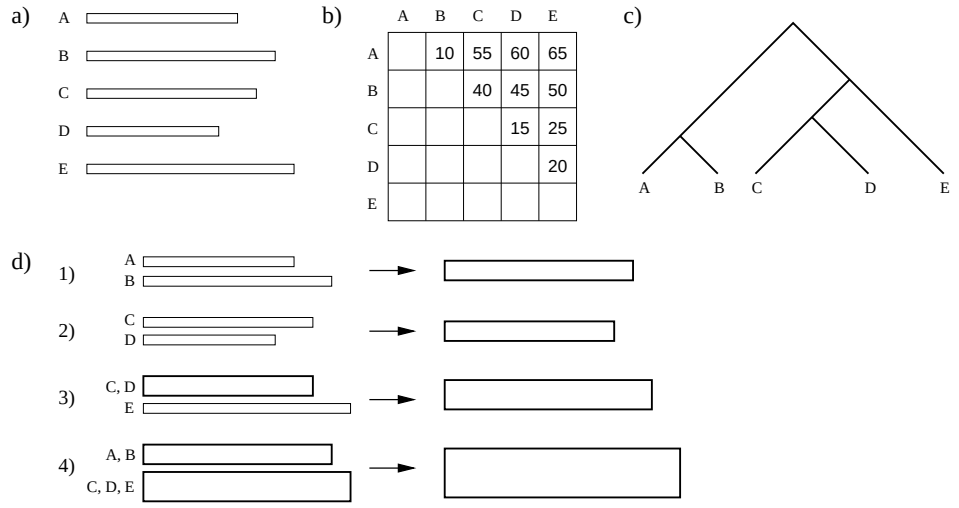

Abbildung 1.4: Progressives Alignment von fünf Sequenzen (a). Im ersten Schritt werden die Distanzen aller Paare von Sequenzen berechnet (b). Aus diesen wird dann im nächsten Schritt ein phylogenetischer Baum abgeleitet (c). Nach Vorgabe dieses Baumes werden schließlich die progressiven Alignments durchgeführt (d, 1–4), bis das multiple Alignment berechnet ist.

Knoten, und so weiter bis schließlich alle m Sequenzen aligniert sind. Dabei werden jeweils zwei Sequenzen, eine Sequenz und ein Alignment oder zwei Alignments aligniert. Abbildung 1.4 illustriert den Ablauf progressiver Alignments.

Dieser Ansatz geht allerdings von der Annahme aus, dass alle Sequenzen gleichmäßig und mit konstanter Geschwindigkeit evolvieren, was unrealistisch ist. Ein weiteres Problem ist, dass sich am Anfang gemachte Fehler durch das ganze multiple Alignment ziehen ("once a gap, always a gap"). Allerdings liefern progressive Ansätze trotz dieser Schwachpunkte in den meisten Fällen gute Ergebnisse, die vor allem schnell und effizient berechnet werden.

Weitere Verfahren

Neben den beiden vorgestellten Ansätzen für die Berechnung von multiplen Alignments existieren noch einige andere, die ich hier nur kurz erwähnen möchte:

- Motif-Suche Verfahren [HHS90]
- Stochastische Verfahren wie Hidden-Markov-Modelle [KBM94]
- Divide-and-Conquer Verfahren [Sto98]

Die hier vorgestellten Arbeiten gingen ursprünglich von reinen Seqeunzalignments aus. Diese sind – wie in der Einleitung schon erwähnt – bei Ribonuleinsäuren nicht allzu aussagekräftig. Deswegen wende ich mich nun der Struktur zu.

1.1.3 Sequenz-Struktur-Ansätze

Für die Repräsentation von RNA-Sekundärstrukturen gibt es mehrere Möglichkeiten und damit auch mehrere entsprechende Bewertungsfunktionen.

RNA-Sekundärstrukturen können beispielsweise als gelabelte oder ungelabelte Bäume dargestellt werden, zwischen denen dann eine Distanz berechnet wird. Arbeiten dazu existieren unter anderen von Shapiro *et al.* [SZ90], Shasha *et al.* [SWZ94] sowie Zhang [Zha96a, Zha96b].

Eine andere Möglichkeit für die Repräsentation von RNA-Strukturen beruht auf der Verwendung von stochastischen kontextfreien Grammatiken, welche eine Verallgemeinerung der Hidden-Markov-Modelle darstellen. SCFGs beschreiben dabei sowohl die gemeinsame Sequenz als auch die gemeinsame Struktur. Die Wahrscheinlichkeiten für die Produktionsregeln werden dabei aus einen Trainingsset gewonnen, weshalb die Anwendung von SCFGs auf relativ einfache Bewertungsschemen mit wenig Parametern begrenzt sind. Arbeiten zu diesem Ansatz gibt es beispielsweise von Sakakibara *et al.* [SBH94] und Brown [Bro00].

Andererseits gibt es auch einige Ansätze, welche die Ähnlichkeit auf der Sequenzebene betrachten und dabei Basenpaare als Einheiten behandeln. Dazu gehören unter anderen die Arbeiten von Sankoff [San85], Zuker *et al.* [ZS81], Zhang *et al.* [ZWM99] und Jiang *et al.* [JLM02].

1.2 Verwandte Arbeiten

Wie im Abschnitt über Sequenz-Struktur-Ansätze schon angedeutet, gibt es eine Vielzahl von unterschiedlichen Ansätzen für multiple Sequenz-Struktur-Alignments. Vier wichtige verwandte Ansätze dazu werde ich nun exemplarisch vorstellen.

1.2.1 Der Sankoff-Algorithmus

Einer der ersten Ansätze für multiple RNA-Alignments stammt von Sankoff [San85], welcher einen Algorithmus für das Alignieren von Sequenzen bei gleichzeitiger Vorhersage einer gemeinsamen Sekundärstruktur entwickelt hat.

Der Ansatz für das Sequenzalignment verwendet eine Distanzminimierung, während die Strukturvorhersage auf einem Loop-basierten Energiemodell beruht. Dabei geht Sankoff von der Hypothese aus, dass diejenige Sekundärstruktur welche über alle möglichen Paarungsvarianten die niedrigste Summe der freien Energien aller strukturellen Elemente hat, thermodynamisch am stabilsten ist und deswegen ausgebildet wird.

Um die freien Energien zu bestimmen wird die Struktur in verschiedene Loop-Formen und externe Basenpaare zerlegt. Abbildung 1.5 gibt eine Übersicht der verschiedenen Strukturbestandteile. Während externe Basenpaare keinen Beitrag zur freien Energie liefern, sind für *Hairpins*, *Bulges*, *interne Loops* und *Stems* die freien Energiebeiträge experimentell bestimmt. Um exponentielle Laufzeiten zu vermeiden werden *multiple Loops* durch eine lineare Funktion aproximiert.

Um nun das Falten mit dem Alignment kombinieren zu können, nutzt der Algorithmus eine Reihe von biologisch gerechtfertigten Einschränkungen. Diese laufen insgesamt darauf hinaus, dass strukturelle Elemente immer in beide RNAs eingefügt und alle äquivalenten Basenpaare der Elemente gegeneinander aligniert werden.

Insgesamt ergibt sich so eine Mischfunktion aus freier Energie und Alignmentkosten, die dann mittels dynamischer Programmierung optimiert werden kann. Der Nachteil dabei besteht aus den hohen Kosten. Für das paarweise Alignment beträgt die Laufzeit $O(n^6)$ und der Speicherbedarf $O(n^4)$ in Abhängigkeit von der Sequenzlänge n.

Die Berechnung multipler Alignments von m Sequenzen benötigt sogar exponentiele Laufzeit der Größenordnung $O(n^{3m})$. Diese kann allerdings durch die Begrenzung der Abstände zwischen den strukturbildenden Positionen auf $O(n^3 \cdot c^m)$ gesenkt werden, wobei c eine Konstante ist.

Ein weiteres Problem bildet der hohe Implementierungs- und Rechenaufwand des vollständig Loop-basierten Energiemodells. Wegen dieser Nachteile verwenden momentan vorhandene Softwarepakete nur modifizierte Versionen des Algorithmus.

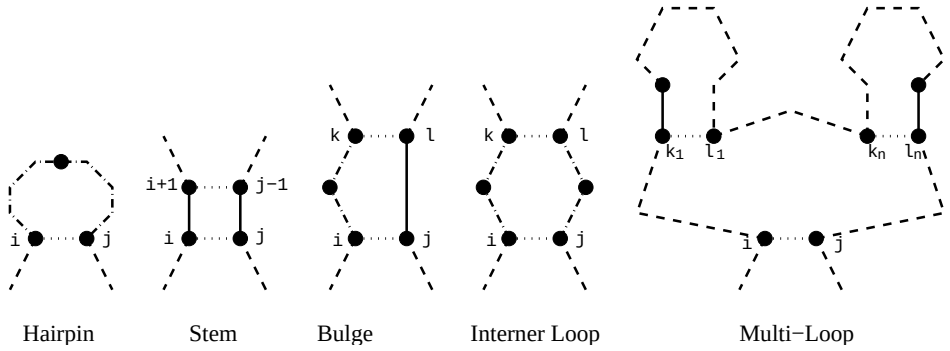

Abbildung 1.5: Die verschiedenen Sekundärstrukturbestandteile. Punkte repräsentieren einzelne Basen, gepunktete Verbindungen bedeuten eine Basenpaarbindung und Linien stellen das Polyphosphatrückgrad dar. Dabei stellen durchgezogene Linien Bereiche ohne Basen dar, gestrichelte Linien stellen Bereiche dar, in denen Basen und Basenpaare vorkommen können und gestrichelte Linien mit Punkten repräsentieren Bereiche die Basen enthalten können, in denen aber keine Basenpaare vorkommen.

1.2.2 MARNA

Der Ansatz von Siebert und Backofen [SB03] erstellt ein multiples globales Alignment von Ribonukleinsäuren mit bekannter Struktur auf Grundlage paarweiser Sequenz-Struktur-Vergleiche nach Jiang *et al.* [JLM02]. Dabei berechnet ein flexibles Bewertungsschema die Distanzen zwischen zwei RNA-Sequenz-Strukturen, wobei zwischen Edit-Operationen auf Basenpaaren und auf Basen unterschieden wird.

Insgesamt gibt es *arc-match*, *arc-mismatch*, *arc breaking*, *arc-altering* und *arc-removing* als Operationen auf Basenpaaren und *base-match*, *base-mismatch* und *base-deletion* als Operationen auf Basen. Abbildung 1.6 gibt eine Übersicht der einzelnen Edit-Operationen. Die Operationen auf einen Basenpaar, die beteiligten Basen und die Summe der jeweiligen Kosten definieren so strukturelle Komponenten.

Für die Berechnung des multiplen Alignments verwenden Siebert und Backofen *T-Coffee* [NHH00], ein fortschrittliches progressives Alignmentverfahren. Während herkömmliche progressive Ansätze unter ihrer geizigen Natur leiden – Fehler in den ersten Alignments können später beim Einfügen der restlichen Sequenzen nicht mehr verbessert werden – minimiert T-Coffee dieses Problem, indem es mit den Alignmentsschritten beginnt, die von den meisten anderen paarweisen Alignments bestätigt werden.

Für die Berechnung greift T-Coffee auf die Kanten der einzelnen paarweisen Alignments zurück. Kanten zwischen zwei Basen erhalten dabei ein Gewicht in Abhängigkeit der strukturellen Komponenten, an denen sie beteiligt sind. Dazu werden die Kosten der strukturellen Komponenten in Ähnlichkeiten transformiert und auf die beteiligten Kanten aufgeteilt. Die Verwendung von Ähnlichkeiten anstatt Distanzen hat dabei den Vorteil, dass RNAs besser bewertet und die Gewichte durch Addition gestärkt werden können. Alle restlichen Kanten zwischen einem Gap und einer Base erhalten ein Gewicht von Null.

Die so von T-Coffee erstellten multiplen Alignments spiegeln aufgrund der Kanteninformationen sowohl die sequentielle als auch die strukturelle Konserviertheit der Sequenzen wieder. Dabei benötigt der Algorithmus für die Berechnung eines multiplen Alignment von m Sequenzen mit einer durchschnittlichen Länge von n Nukleotiden $O\big(m^2 \cdot n^4\big) + O\big(m^3 \cdot n^2\big)$ Zeit.

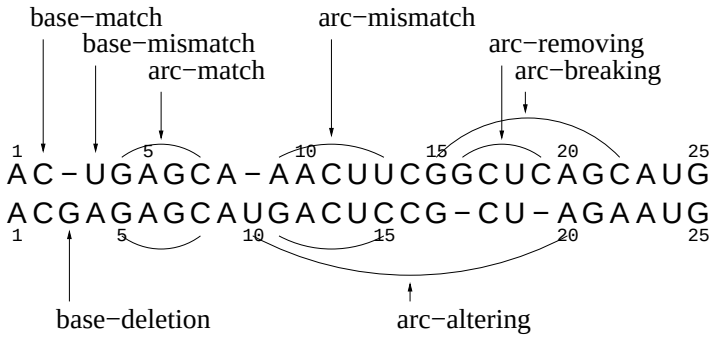

Abbildung 1.6: Die verschiedenen Edit-Operationen auf Basen und Basenpaaren nach Jiang [JLM02]

1.2.3 pmmulti

Auch dieses Programm von Hofacker *et al.* [HBS04] berechnet multiple Alignments progressiv. Dabei geht die Berechnung der paarweisen Alignments auf eine Variante des Sankoff-Algorithmus [San85] zurück, bei der das anspruchsvolle Loop-basierte Energiemodell durch ein einfacheres Modell ersetzt wurde.

In einen Vorverarbeitungsschritt werden dazu mit Hilfe des McCaskill-Algorithmus [McC90] die thermodynamischen Informationen in Form von Basenpaarwahrscheinlichkeiten über den Sequenzen berechnet. Damit reduziert sich das Problem auf paarweise Alignments von Basenpaarwahrscheinlichkeitsmatrizen, also dem finden einer Sekundärstruktur mit maximalen Gewicht, welche in den jeweiligen Matrizen enthalten ist.

Das Gewicht ergibt sich dabei aus der Summe der Basenpaargewichte, den Substitutionsgewichten von ungepaarten und gepaarten Basen sowie den Gap-Kosten. Über die Substitutionsgewichte kann man dabei Verteilungen basierend auf Covariation und Substitution einfliesen lassen bzw. kann man sie bei reinen Struktur-Alignments auch vernachlässigen.

Für die paarweisen Alignments braucht pmmulti so analog zu dem ursprünglichen Sankoff-Algorithmus $O(n^6)$ Zeit und $O(n^4)$ Speicher bei einer durchschnittlichen Sequenzlänge von n. Um diesen Bedarf zu verringern haben Hofacker *et al.* eine maximale Spannweite für die partiellen Alignments eingeführt, welche den Größenunterschied zwischen zwei Teilsequenzen begrenzt. Dadurch ist es möglich, die Komplexität bis auf $O(n^4)$ Zeit und $O(n^3)$ Speicher zu beschränken. Allerdings sind dabei für deutliche Verbesserungen sehr niedrigen Spannweiten notwendig. Dies allerdings schränkt wiederrum den Suchraum so sehr ein, dass optimale Ergebnisse leicht übersehen werden.

Aufgrund der immer noch recht hohen Kosten, greifen Hofacker *et al.* bei der Berechnung des multiplen Alignments auf einen einfachen, aber $O(n^2)$ schnellen Algorithmus zurück, um die $m(m-1)/2$ paarweisen Alignments zur Ableitung des phylogenetischen Baumes der m Sequenzen zu berechnen. Damit muss der kostspieligere Algorithmus nur noch für die $m-1$ progressiven Alignments verwendet werden. Allerdings erhöht sich damit die Gefahr, durch einen falschen phylogenetischen Baum auch ein falsches multiples Alignment zu erhalten.

Für die Berechnung der progressiven Alignments bei denen zuvor berechnete Alignments beteiligt sind, bestimmt pmmulti eine Konsensus-Paarwahrscheinlichkeitsmatrix der betroffenen Alignments. Diese ergibt sich dabei aus dem geometrischen Mittel der Paarwahrscheinlichkeitsmatrizen der alignierten Sequenzen.

Insgesamt benötigt pmmulti so mit der Einschränkung der maximalen Spannweite und unter Verwendung des schnellen Alignmentalgorithmus für die Berechnung

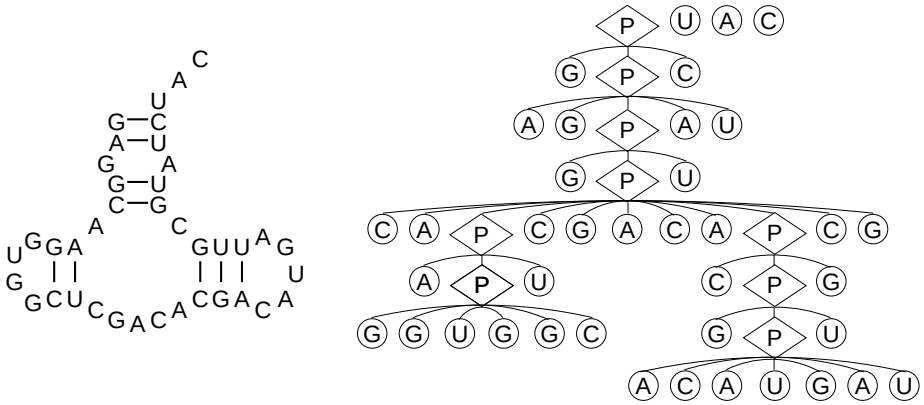

Abbildung 1.7: Beispiel für die Darstellung einer RNA Sekundästruktur als ein Wald von Bäumen. Basenpaare werden explizit durch P-Knoten dargestellt, wobei die äußeren Kinder die paarenden Basen darstellen.

des phylogenetischen Baumes $O(m^2 \cdot n^2) + O(m \cdot n^4)$ Zeit.

1.2.4 RNA-forester

Das Programm von Höchsmann *et al.* [HTG03] berechnet multiple Baum-Alignments von genesteten Sequenz-Strukturen. Eine Sequenz-Struktur wird dabei als ein Wald von Bäumen dargestellt, wobei zwei Arten von Knoten existieren. P-Knoten stehen für Basenpaarbindungen – ohne die beteiligten Basen – und werden mit P bezeichnet. B-Knoten symbolisieren die Basen und werden mit den entsprechenden Nukleotiden bezeichnet.

Die Eltern- und Nachkommensbeziehung wird durch die Positionen der Knoten innerhalb der Struktur festgelegt. Alle Knoten innerhalb eines Basenpaars bilden zusammen mit den Basen der Basenpaarbindung die Nachlommen des P-Knotens, welcher die entsprechende Basenpaarbindung repräsentiert. Die Reihenfolge der Nachkommen wird dabei durch die 5'–3'-Natur der RNA-Moleküle festgelegt, wobei die Basen der übergeordneten Basenpaarbinung ganz links bzw. ganz rechts stehen. Abbildung 1.7 zeigt ein Beispiel für die Repräsentation einer RNA durch einen Wald von Bäumen.

Um nun ein Baumalignment zweier Sequenz-Strukturen zu erhalten, werden die beiden den Sequenz-Strukturen entsprechenden Wälder durch eine Reihe von Edit-Operationen ineinander überführt. Das Löschen eines Knotens repräsentiert dabei eine Insertion oder Deletion, während der Austausch eines Knotens ein Match oder Mismatch beschreibt. Dabei ist es in Analogie zu der Arbeit von Jiang *et al.* [JLM02] durch die Aufteilung von Basenpaaren in P-Knoten und B-Knoten auch hier möglich, Ereignisse wie ein arc-altering oder ein arc-breaking zu behandeln. Abbildung 1.8 zeigt ein Beispiel eines Baumalignments.

Den Edit-Operationen sind dabei in Abhängigkeit der beteiligten Knoten Ähnlichkeitsscores zugeordnet. Damit ergibt sich das beste Baumalignment aus der Folge von Operationen mit der höchsten Summe über alle Scores. Diese Summe kann rekursiv aus Baumalignments von Teilwäldern abgeleitet werden, wodurch man mittels dynamischer Programmierung ein komplettes Baumalignment zweier Sequenz-Strukturen mit einer Länge von n in $O(n^4)$ Zeit und mit $O(n^4)$ Speicher berechnen kann[2].

[2]Die genaue Komplexität hängt dabei von der Anzahl der Knoten in den Wäldern und dem

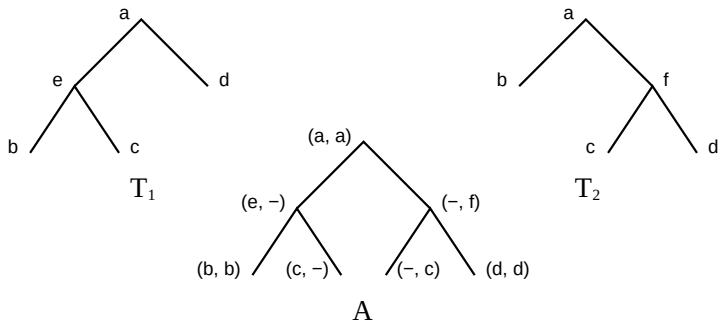

Abbildung 1.8: Beispiel für ein optimales Baumalignment zweier Bäume T_1 und T_2 (aus [JWZ95]).

Durch diese rekusive Zerlegung erhält man mit den Zwischenergebnissen auch gleichzeitig alle Alignments derjenigen Teilwälder, deren Wurzeln im ursprünglichen Wald nebeneinander lagen. Damit hat man neben dem Alignment beider Sequenz-Strukturen auch Alignments von allen Teilsequenz-Strukturen berechnet. Die beiden Teilsequenz-Strukturen mit dem höchsten Score entsprechen dann dem besten lokalen Alignment. Aus den paarweisen Alignments wird schließlich unter Verwendung eines progressiven Ansatzes ein mutilpes Alignment berechnet.

Die in diesem Ansatz verwendet Form von Lokalität entspricht dabei derjenigen, wie sie bei Sequenzen üblich ist: Alignments von fortlaufenden Teilsequenzen. Damit werden die strukturellen Zusammenhänge nicht mit in den Lokalitätsbegriff aufgenommen. Die multiplen Alignments besitzen sogar überhaupt keine lokalen Informationen, da bei deren Erstellung nur auf die globalen paarweisen Alignments zurückgegriffen wird.

Ausserdem haben Baumalignments generell den Nachteil, dass die Edit-Distanz nicht mehr mit der Alignmentdistanz übereinstimmt [JWZ95]. So lässt sich beispielsweise der Baum T_1 in Abbildung 1.8 mit einer Deletion von e und einer Insertion von f in den Baum T_2 überführen. Das optimale Alignment hingegen weist eine Distanz von zwei Insertionen und zwei Deletionen auf. Hinzu kommt, dass bestimmte Alignments nicht von Baumalignments dargestellt werden können, da immer nur Basenpaare auf Basenpaare und Basen auf Basen abgebildet werden. So ist es nicht möglich, ein Alignment zweier sich kreuzende Basenpaare wie es beispielsweise in Abbildung 1.6 vorkommt (die Basenpaare des arc-altering und des arc-removing kreuzen sich), darzustellen.

1.3 Übersicht

In Kapitel 2 werden die Vorraussetzungen für eine genauere Herleitung und Betrachtung des Ansatzes geschaffen. Zuerst werden dabei wichtige Begriffe erklärt und formalisiert. Besonderes Augenmerk liegt dabei auf dem Begriff der strukturellen Lokalität, da dies ein völlig neuen Ansatz bei multiplen Alignments darstellt. Anschliesend werden die konkreten Problemstellungen eingeführt und definiert.

Kapitel 3 wendet sich dann dem eigentlichen Algorithmus zu. Nach einer kurzen Übersicht über den Programmablauf und die Hintergründe werden die einzelnen Bestandteile des Ansatzes ausführlich beschrieben. Eine Komplexitätsanalyse der von mir entwickelten Algorithmen und des gesamten Ansatzes beendet das Kapitel.

Grad der Wälder ab. Seien F_1 und F_2 zwei Wälder, $|F_g|$ die Anzahl der Knoten in F_g und $deg(F_g)$ der Grad von F_g. Dann beträgt die genaue Komplexität $O(|F_1| \cdot |F_2| \cdot deg(F_1) \cdot deg(F_2) \cdot (deg(F_1) + deg(F_2)))$ Zeit und $O(|F_1| \cdot |F_2| \cdot deg(F_1) \cdot deg(F_2))$ Speicher.

Das darauf folgende Kapitel 4 widmet sich zuerst den Parametern. Der Schwerpunkt liegt dabei auf den Basenpaarwahrscheinlichkeiten. Nach einer Analyse der Verteilung in Ribonukleinsäuren wird eine Grenzwahrscheinlichkeit diskutiert, welche für natürliche Strukturen wichtige Wahrscheinlichkeiten von unwichtigen trennen soll. Anschliesend werden verschiedene Testdurchläufe präsentiert und die Ergebnisse mit anderen Programmen und den korrekten Ergebnissen aus einer Datenbank verglichen.

In Kapitel 5 beendet eine kurze Zusammenfassung schließlich die Arbeit.

Kapitel 2

Vorbetrachtungen

Bevor ich mich im nächsten Kapitel dem eigentlichen Programm zuwende, werden in diesem Kapitel die Problemstellungen konkretisiert, sowie die Definitionen aller wichtigen Begriffe des Ansatzes formuliert.

2.1 Formale Definitionen

Der erste Abschnitt widmet sich den Definitionen. Dabei werde ich anhand des Programmablaufes die Bedeutung der Begriffe zuerst vorstellen und anschließend formalisieren. Im Mittelpunkt steht dabei die strukturelle Lokalität, da diese Lokalitätsform bei multiplen Sequenz-Struktur-Alignments einen völlig neuen Weg darstellt.

2.1.1 Sequenz und Struktur

Die Eingabe für den Algorithmus bilden die Sequenzen der zu alignierenden Ribonukleinsäuren.

Definition 1 (Sequenz)
Sei R ein RNA-Molekül. Die primäre Sequenz S_R von R ist ein Wort über dem Alphabet Σ_N der Nukleotide.

$$S_R \in \Sigma_N^*,\ \Sigma_N = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$$

Weiterhin bezeichnet:

- *$S_R[\,i\,]$ den i-ten Buchstaben von S_R,*
- *$S_R[\,i, j\,]$ die Teilsequenz von der i-ten bis zur j-ten Position aus S_R und*
- *$|S_R|$ die Länge der Sequenz S_R.*

Aus den Sequenzen leitet MuLoRA zunächst die strukturellen Informationen der Ribonukleinsäuren ab. Dazu werden für alle Sequenzen die Basenpaarwahrscheinlichkeitsmatrizen berechnet. Diese enthalten für alle möglichen Basenpaare über der Sequenz die Wahrscheinlichkeiten, mit der diese Basenpaare in einer Struktur des RNA-Moleküls ausgebildet werden.

Definition 2 (Basenpaar)
Ein Basenpaar a über einer Sequenz S ist ein Tupel $(i, j) \in \{1 \ldots |S|\} \times \{1 \ldots |S|\}$ mit $i < j$.

Das linke Ende i von a wird dabei auch mit a^l bezeichnet und das rechte Ende j mit a^r. Weiterhin bezeichnet $p_s(a)$ die Basenpaarwahrscheinlichkeit von a über der Sequenz S.

Basenpaare bilden normalerweise ein Watson-Crick-Paar (`A`–`U`, `G`–`C`) oder Nicht-Standart-Paar (`G`–`U`) aus. Für den Algorithmus selbst spielt dies aber keine Rolle.

Die Basenpaare, welche sich über einer RNA-Sequenz ausbilden, bestimmen die Struktur, die das Molekül einnimmt. Die Menge aller möglichen Basenpaare über einer Sequenz repräsentiert damit auch alle möglichen Strukturen die ausgebildet werden können. Dabei ergibt natürlich nicht jede beliebige Teilmenge von Basenpaaren eine gültige Struktur.

Definition 3 (Strukturmenge)
Eine Strukturmenge P_R einer Ribonukleinsäure R ist eine Menge von Basenpaaren über der Sequenz S_R mit folgenden Eigenschaften:

1. $(i,j) \in P_R \wedge (i',j) \in P_R \Rightarrow i = i'$
   (j kann nur für ein Basenpaar rechtes Ende sein)
2. $(i,j) \in P_R \wedge (i,j') \in P_R \Rightarrow j = j'$
   (i kann nur für ein Basenpaar linkes Ende sein)
3. $(i,j) \in P_R \Rightarrow \forall k\,(k,i) \notin P_R \wedge \forall l\,(j,l) \notin P_R$
   (eine Base kann nicht gleichzeitig linkes Ende eines Basenpaares und rechtes Ende eines anderen Basenpaares sein)

In Abhängigkeit der Lage der Basenpaare in einer Struktur zueinander ordnet man die Basenpaarbeziehungen in verschiedene Klassen ein.

Definition 4 (Basenpaarbeziehungen)
Sei P_R eine Strukturmenge und (i,j) und (k,l) zwei Basenpaare aus P_R (o.B.d.A. sei $i < k$). Gilt:

- $i < j < k < l$, *bezeichnet man die Basenpaare als* unabhängig,
- $i < k < l < j$, *bezeichnet man die Basenpaare als* genestet,
- $i < k < j < l$, *bezeichnet man die Basenpaare als* gekreuzt.

Die Reihenfolge der genesteten Basenpaare einer Strukturmenge P_R definiert eine Halbordnung über P_R.

Definition 5 ($\preceq_P$)
Sei P eine Strukturmenge und (i,j) und (k,l) zwei Basenpaare aus P. Dann gilt $(k,l) \preceq_P (i,j)$ genau dann, wenn (i,j) das Basenpaar (k,l) umschließt bzw. (i,j) gleich (k,l) ist. Formal ausgedrückt bedeutet dies:

$$\preceq_P = \{((k,l),(i,j)) \mid (i,j) \in P \wedge (k,l) \in P \wedge i \leq k < l \leq j\}$$

Weiterhin bezeichne ich (i,j) als Nachfolger *der Base (k,l) bzw. (k,l) als* Vorgänger *der Base (i,j), falls $i < k < l < j$ gilt. Falls zusätzlich keine weitere Base $(m,n) \in P$ mit $i < m < k < l < n < j$ existiert, bezeichne ich (i,j) als* direkten Nachfolger *bzw. (k,l) als* direkten Vorgänger.

Insgesamt teilt man RNA-Strukturen in Abhängigkeit der Detailliertheit ihrer Abstraktion in drei Gruppen ein. Die Primärstruktur besteht nur aus den Sequenzdaten. Die Sekundärstruktur beinhaltet neben den Sequenzinformationen auch die Strukturmenge, also alle Basenpaare, die in der Struktur vorkommen. Die Tertiärstruktur enthält schließlich neben der Strukturmenge auch die räumlichen Informationen, also insgesamt die komplette dreidimensionale Struktur. Abbildung 2.1 zeigt ein Beispiel für die drei Einteilungen.

Da, wie in der Einleitung schon erwähnt, die Tertiärstruktur einer RNA mehr Freiheitsgrade als die eines Polypeptids hat, arbeitet MuLoRA mit Sekundärstrukturen.

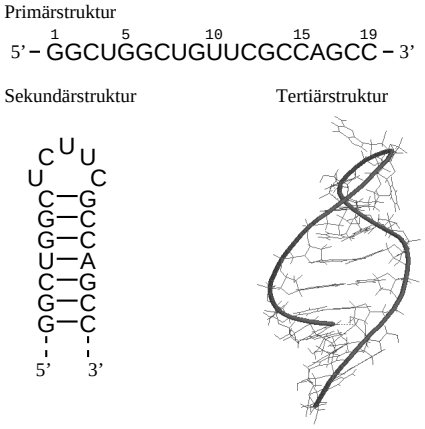

Abbildung 2.1: Darstellung des A-Loops der A-Site einer 23S rRNA als Primärstruktur, Sekundärstruktur und Tertiärstruktur.

Definition 6 (Sekundärstruktur)
Die Sekundärstruktur $\mathcal{S}$ *einer Ribonukleinsäure* R *ist ein Tupel der Sequenz und der Strukturmenge.*

$$\mathcal{S}_R = (S_R, P_R)$$

Sekundärstrukturen werden dabei in der Bioinformatik Aufgrund der Lage der Basenpaare innerhalb der Struktur zueinander noch einmal in verschiedene Klassen unterteilt.

Definition 7 (Sekundärstrukturklassen)
Sei R *eine RNA und* $\mathcal{S}_R$ *ihre Sekundärstruktur. Dann ist* $\mathcal{S}_R$*:*

- crossing *genau dann, wenn* P_R *mindestens ein Paar sich kreuzender Basenpaare enthält,*
- nestet *genau dann, wenn* P_R *nur aus unabhängigen oder genesteten Paaren von Basenpaaren besteht oder*
- plain *genau dann, wenn* $P_R = \emptyset$ *gilt.*

Abbildung 2.2 zeigt ein Beispiel für die Sekundärstrukturklassen. Die Zugehörigkeit einer Sekundärstruktur zu einer dieser Klassen hat Auswirkungen auf die Komplexität des Alignment-Problems. Im Allgemeinen unterscheidet man zwischen sechs Problemfällen, wenn es darum geht, eine Edit-Distanz zwischen zwei Sekundärstrukturen zu berechnen.

$$\begin{aligned} \textit{crossing} &\Leftrightarrow \textit{crossing} \\ \textit{crossing} &\Leftrightarrow \textit{nested} \\ \textit{crossing} &\Leftrightarrow \textit{plain} \\ \textit{nested} &\Leftrightarrow \textit{nested} \\ \textit{nested} &\Leftrightarrow \textit{plain} \\ \textit{plain} &\Leftrightarrow \textit{plain} \end{aligned}$$

Jiang *et al.* [JLM02] konnten zeigen, dass bereits das *crossing* $\Leftrightarrow$ *plain* Problem MAX SNP-hart ist und damit auch für die ersten beiden Problemfälle eine MAX SNP-härte folgt. Deshalb arbeitet mein Ansatz auch mit genesteten Strukturen, obwohl er theoretisch durch die Verwendung von Paarwahrscheinlichkeiten über allen möglichen Strukturklassen arbeiten könnte.

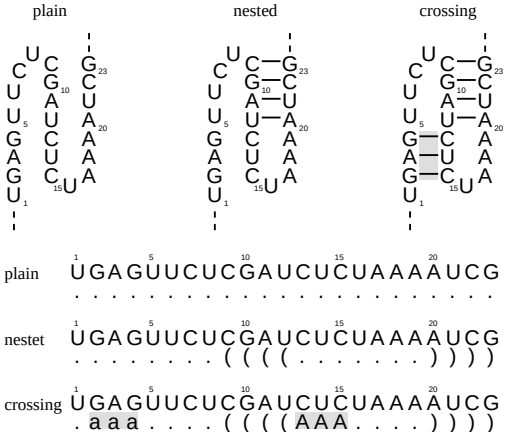

Abbildung 2.2: Das 3'-Ende der tRNA ähnlichen Turnip-Yellow-Mosaic-Virus-RNA in den drei Sekundärstrukturklassen. Der Pseudoknoten, welcher in der crossing-Sekundärstruktur durch das Ausbilden der grau hinterlegten Basenpaare entsteht, fungiert als Enhancer.

Für die Darstellung von Sekundärstrukturen verwende ich dabei Graphen $G = (V, E)$ mit $V = \{1 \dots |S|\}$ und $E = \{(i, i+1) \mid 1 \leq i < |S|\} \cup P$. Allerdings füge ich dabei für eine bessere Übersichtlichkeit anstatt der Positionsangaben die entsprechenden Basen ein und lasse die Kanten zwischen aufeinanderfolgenden Basen weg, sofern die Reihenfolge eindeutig zu erkennen ist.

Weiterhin verwende ich eine linearisierte Darstellungsform, welche aus der Sequenz und einem zusätzlichen Strukturstring der gleichen Länge besteht. Dieser kennzeichnet dabei in genesteten Strukturen alle linken Basenpaarenden mit '(', alle rechten Basenpaarenden mit ')' und alle ungepaarten Basen mit '.'.

Für genestete Sekundärstrukturen ist die Zuordnung zwischen dem linken Ende und dem rechten Ende eines Basenpaars eindeutig. Bei crossing-Sekundärstrukturen wäre das nicht der Fall. Aus diesem Grund werden da zu den Klammersymbolen zusätzlich Buchstaben verwendet, wobei kleine Buchstaben das linke Ende und große Buchstaben das rechte Ende eines Basenpaars kennzeichnen. Abbildung 2.2 zeigt ein Beispiel für die von mir verwendeten Darstellungsformen.

Damit sind erst einmal die wichtigsten Begriffe zu RNA-Molekülen definiert. Als nächstes wende ich mich den Alignments zu.

2.1.2 Alignments und Lokalität

Aus den Eingabesequenzen und den Sekundärstrukturinformationen in Form der Basenpaarwahrscheinlichkeiten werden als nächstes alle möglichen paarweisen Alignments berechnet. Dabei werden die Sequenzen mit Hilfe von neutralen Gap-Elementen '$-$' so angeordnet, dass sequenziell und strukturell ähnliche Bereiche in den gleichen Regionen stehen. Formal ausgedrückt bedeutet dies folgendes:

Definition 8 (paarweises Alignment)
Seien R_1 und R_2 zwei RNA-Moleküle, S_{R_1} bzw. S_{R_2} deren Sequenzen. Ferner sei ein Alignmentalphabet Σ_A wie folgt definiert:

$$\Sigma_A =_{def} (\{1 \dots |S_{R_1}|\} \cup \{-\}) \times (\{1 \dots |S_{R_2}|\} \cup \{-\}) \setminus \{(-,-)\}\,.$$

Dann ist ein paarweises Alignment A von S_{R_1} und S_{R_2} ein Wort über Σ_A, bei dem für jedes $g \in \{1, 2\}$ die Aneinanderreihung aller g-ten Elemente ungleich '$-$' der

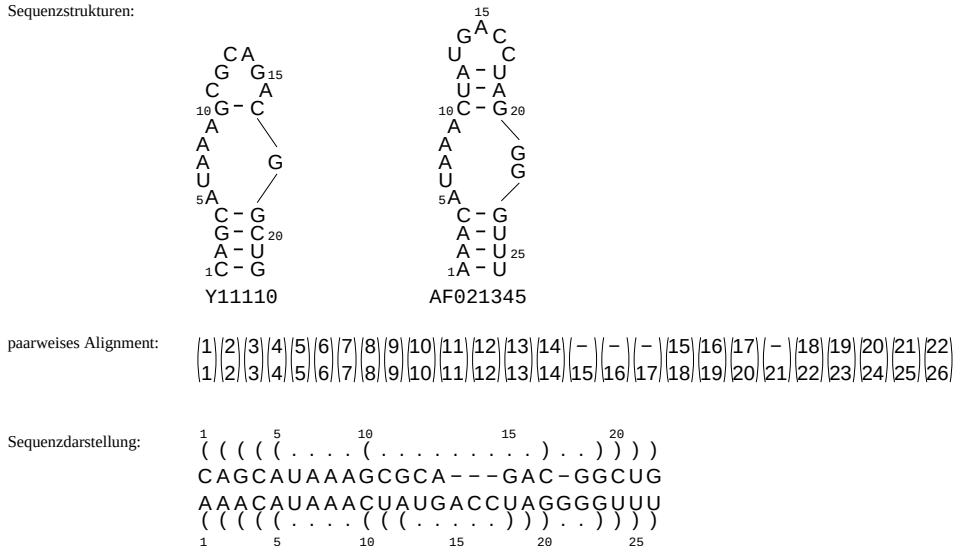

Abbildung 2.3: Paarweises Alignment der Sequenzen zweier SECIS-Motive.

einzelnen Kanten in der Reihenfolg ihres Auftretens in A ein Wort ergibt, welches aus aufsteigend sortierten Zahlen besteht.

Weiterhin bezeichnet:

- $A[\,i\,]$ *die i-te Kante von A,*
- $A[\,g\,][\,i\,]$ *das g-te Element mit* $g \in \{1,2\}$ *der i-ten Kante von A und*
- $|A|$ *die Anzahl der Kanten innerhalb des Alignments A.*

Die Elemente in einer Kante geben dabei die Positionen der Buchstaben innerhalb der entsprechenden Sequenzen an. Sollten dabei Gap-Symbole '−' vorkommen, werden an den entsprechenden Stellen keine Buchstaben eingefügt. Abbildung 2.3 zeigt ein Beispiel für ein Alignment der Sequenzen zweier SECIS-Motive. Für die Darstellung paarweiser Alignments in dieser Arbeit verwende ich die Sequenzen über denen das Alignment berechnet wurde. Dabei werden diese mit Hilfe des Gap-Symbols so angeordnet, dass alignierte Positionen übereinander stehen. Bei Alignments von Sekundärstrukturen gebe ich dabei zusätzlich noch den Strukturstring an.

Nach Definition 8 braucht ein paarweises Alignment A nicht für jede Position der Sequenzen S_{R_1} bzw. S_{R_2} eine Kante zu enthalten. Damit handelt es sich bei dieser Definition nicht um globale Alignments. Diese müssten zusätzlich sowohl für jedes $1 \leq i_1 \leq |S_{R_1}|$ als auch für jedes $1 \leq j_2 \leq |S_{R_2}|$ ein Kante (i_1, i_2) und eine Kante (j_1, j_2) enthalten.

Da das Alignment also nicht global sein muss, benötige ich noch Formalismen, welche Informationen über die im Alignment vorkommenden Sequenzabschnitte und Strukturen liefern:

Definition 9
Seien R_1 *und* R_2 *zwei Ribonukleinsäuren,* (S_{R_1}, P_{R_1}) *und* (S_{R_2}, P_{R_2}) *deren Sekundärstrukturen und A ein Alignment von* S_{R_1} *und* S_{R_2}*. Dann ist:*

- $\pi_g(A) = \left\{ i_g \in \left\{1 \ldots |S_{R_g}|\right\} \mid \exists\,(i_1, i_2) \;\; in\; A \right\}, g \in \{1,2\}$
  (Die Menge der in A alignierten Positionen der g-ten Sequenz)
- $P_g^A = \left\{ (i_g, j_g) \in P_{R_g} \mid \exists\,(i_1, i_2) \;\; in\; A \wedge \exists\,(j_1, j_2) \;\; in\; A \right\}, g \in \{1,2\}$
  (Die Menge der in A alignierten Basenpaare der g-ten Strukturmenge)

- $\Pi_g^A = \{1 \dots |S_{R_g}|\} \cap \left\{ i \mid \exists (i,j) \in P_{R_g} \vee \exists (j,i) \in P_{R_g} \right\}, g \in \{1,2\}$
  (Die Menge der Positionen der g-ten Sequenz, welche in A nicht an einem alignierten Basenpaar beteiligt sind)

Für die Darstellung alignierten Teilstrukturen verwende ich Motiv-Graphen. Diese bestehen nur aus denjenigen Basen und Basenpaaren, welche im Alignment durch Kanten repräsentiert werden.

Definition 10 (Motiv-Graph)
Seien R_1 und R_2 zwei RNAs und (S_{R_1}, P_{R_1}) und (S_{R_2}, P_{R_2}) deren Sekundärstrukturen. Ferner sei A ein Alignment von S_{R_1} und S_{R_2}. Dann sind die Motiv-Graphen $G_g^A = (V_g^A, E_g^A)$ von A mit $g \in \{1,2\}$ wie folgt definiert:

$$\begin{aligned} V_g^A &= \pi_g \\ E_g^A &= \left\{ (i, i+1) \mid i \in V_g^A \wedge i+1 \in V_g^A \right\} \cup P_g^A \end{aligned}$$

In Analogie zur Darstellung der Sekundärstrukturen füge ich auch bei den Motiv-Graphen die entsprechenden Basen für die Postionsnummern ein und lasse die Kanten zwischen aufeinanderfolgenden Basen weg, sofern die Reihenfolge eindeutig zu erkennen ist.

Damit wäre der Begriff des Alignments geklärt. Doch was genau macht nun ein lokales RNA-Sequenz-Struktur-Alignment aus? Welche Abhängigkeiten bestehen innerhalb von Molekülen und wie sollen sich diese Einschränkungen bemerkbar machen? Diesen und andere Fragen werde ich nun nachgehen, wobei ich mich an den Lokalitätsbegriff von Backofen und Will [BW04] halten werde.

In der Natur weisen Moleküle – seien dies nun Proteine, RNA oder DNA – oft nur in bestimmten Bereichen eine hohe Ähnlichkeit auf, während der Rest vollkommen divergent ist. Beispiele dafür sind Proteine, die über eine gleiche Domäne verfügen, erweiterte Bereiche von genomischer DNA oder strukturelle RNA-Motive. Aber auch Moleküle, die den selben evolutionären Ursprung besitzen, weisen mitunter nur noch in Teilbereichen eine nachweisbare Ähnlichkeit auf, da der Selektionsdruck nur für diese Bereiche hoch genug war um ein auseinanderdriften zu verhindern. Die beiden Sequenzen in Abbildung 2.3 bilden beispielsweise das gleiche strukturelle Motiv aus (siehe Abbildung 2.5), werden aber von divergenten Loops unterbrochen.

Durch diesen Hintergrund motiviert ist es sinnvoll, bei der Suche nach Motiven mit Hilfe von Alignments das Weglassen bestimmter Bereiche zu erlauben – oder andersherum ausgedrückt – nur bestimmte Bereiche zu alignieren.

Dabei können allerdings nicht beliebige Bereiche ausgelassen werden. Um biologisch sinnvolle Alignments zu erhalten, müssen die alignierten Bereiche in irgendeiner Form zusammenhängend sein. Bei RNAs, denen ich mich nun wieder zuwenden werde, wird der Zusammenhang zum einen von dem Polyphosphatrückgrad gebildet. Auf der Ebene der Sequenz-Alignments bestimmt diese Form des Zusammenhangs auch die herkömliche Definition lokaler Alignments als globale Alignments von Teilsequenzen. Dieser Definition liegt auch einer der bekanntesten Ansätze für lokale Sequenzalignments zugrunde, der Smith-Waterman-Algorithmus [SW81].

Eine Erweiterung dieses Ansatzes stellen Programme wie *BLAST* [AGM90] dar, welche gleich mehrere isolierte Paare von unabhängig alignierten und bewerteten Teilsequenzen liefern. Diese entsprechen dann einfach den k-besten nicht überlappenden lokalen Alignments.

Bei Sequenz-Struktur-Alignments gestaltet sich der Lokalitätsbegriff allerdings schwieriger. Neben den Polyphosphatrückgrad kommt hier zusätzlich noch die Sekundärstruktur als neuer Faktor bei dem Zusammenhang hinzu. Dieser wird nun neben den kovalenten Bindungen des Rückgrades auch durch die Wasserstoffbrückenbindungen der Basenpaare erreicht.

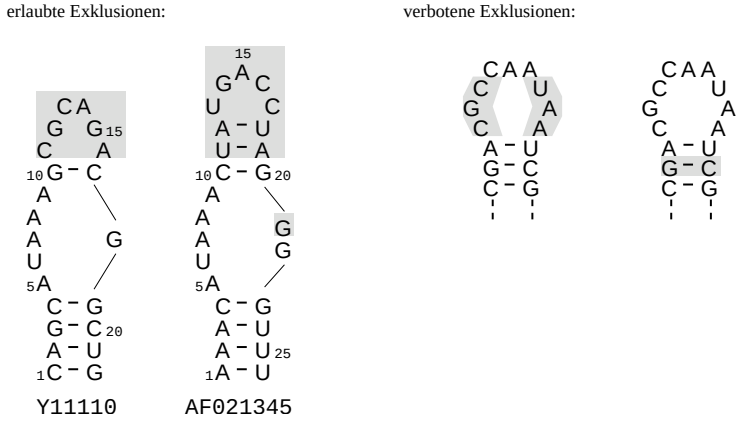

Abbildung 2.4: Die beiden linken Sequenzstrukturen zeigen mögliche Exklusionen (grau hinterlegt) für die beiden SECIS-Motive aus Abbildung 2.3. Die beiden rechten Sequenzstrukturen enthalten hingegen nicht erlaubte Exklusionen. In der ersten wurden in einem Loop zwei Exklusionen durchgeführt, wärend sich in der zweiten Sequenzstruktur die Exklusion außerhalb eines Loops befindet.

Damit erweitert sich die Definition lokaler Alignments bei Sekundärstrukturen von globalen Alignments von Teilsequenzen zu globalen Alignments von Teilstrukturen. Aus diesem Grund ist es jetzt auch möglich, aus zusammenhängenden Teilsequenzen wiederum kleinere Teilsequenzen auszuschließen.

Definition 11 (Exklusion)
Sei A ein Alignment zweier RNA-Sequenzen S_{R_1} und S_{R_2}. Eine Exklusion in S_{R_g} mit $g \in \{1, 2\}$ ist dann als ein Bereich $[\, u, v \,]$ definiert, der folgende Eigenschaften erfüllt:

1. $u \leq v$,
2. $u - 1 \in \pi_g\,(A)$,
3. $v + 1 \in \pi_g\,(A)$ *und*
4. $\{u, \ldots, v\} \cap \pi_g\,(A) = \emptyset$.

Allerdings darf durch eine Exklusion nicht der Zusammenhang der Teilsequenzen zerstört werden. Um dies zu gewährleisten, darf nur maximal eine Exklusion pro Loop – egal welche Art von Loop – durchgeführt werden. Abbildung 2.4 zeigt Beispiele für erlaubte und verbotene Exklusionen.

Obwohl durch Exklusionen mehrere, sequentiell nicht verbundene Teilstücke entstehen, hat ein Alignment dieser Sequenzbereiche jedoch nichts mit der bereits erwähnten Erweiterung auf die k-besten nicht überlappenden lokalen Alignments zu tun. Die durch die Basenpaare hervorgerufene Abhängigkeit verbietet in diesem Fall eine unabhängige Behandlung der Teilsequenzen. Natürlich ist aber auch für Sequenz-Struktur-Alignments eine Erweiterung auf die k-besten nicht überlappenden lokalen Alignments möglich. Dabei können Teilsequenzen jedoch nur dann unabhängig aligniert werden, wenn kein Basenpaar zwischen ihnen existiert.

Aus biologischer Sicht ist es bei Exklusionen jedoch nicht sinnvoll, beliebige Basenpaare für das Bilden einer Verbindung zwischen zwei ansonsten isolierten Teilsequenzen zuzulassen. Da das Ziel darin besteht, konservierte Sequenz-Struktur-Bereiche zu finden, sollten diese verbindenden Basenpaare ebenfalls konserviert sein.

Dabei gilt hier ein Basenpaar genau dann als konserviert, wenn dessen Anfangs und Endpositionen im paarweisen Alignment mit den entsprechenden Positionen

eines anderen Basenpaares gematcht sind. Damit lässt sich diese Form von Lokalität jedoch nur für Alignments und nicht für einzelne RNA-Moleküle definieren.

Definition 12 (lokales paarweises Alignment)
Seien R_1 und R_2 zwei Ribonukleinsäuren, (S_{R_1}, P_{R_1}) und (S_{R_2}, P_{R_2}) deren Sekundärstrukturen und A ein Alignment von S_{R_1} und S_{R_2}. Ferner sei ein konserviertes Basenpaar (i, j) aus P_g^A mit $g \in \{1, 2\}$ genau dann direkter Nachfolger einer Exklusion $[u, v]$ in S_{R_g}, wenn $i < u \leq v < j$ gilt und kein weiteres konserviertes Basenpaar (k, l) mit $i < k < u \leq v < l < j$ aus P_g^A existiert.

Dann ist A genau dann ein lokales paarweises Alignment, wenn es ohne Exklusionen ein globales Alignment zweier zusammenhängender Teilsequenzen von S_{R_1} und S_{R_2} ist und die folgenden drei Punkte erfüllt sind:

1. *Für jede Exklusion $[u_1, v_1]$ in S_{R_1} existiert ein konserviertes Basenpaar (i_1, j_1) aus P_1^A, welches direkter Nachfolger der Exklusion ist und keine andere Exklusion $[u'_1, v'_1]$ hat (i_1, j_1) als direkten Nachfolger.*
2. *Für jede Exklusion $[u_2, v_2]$ in S_{R_2} existiert ein konserviertes Basenpaar (i_2, j_2) aus P_2^A, welches direkter Nachfolger der Exklusion ist und keine andere Exklusion $[u'_2, v'_2]$ hat (i_2, j_2) als direkten Nachfolger.*
3. *Für jedes Basenpaar (i_g, j_g) aus P_g^A mit $g \in \{1, 2\}$ exisitert eine Kante (i_1, i_2) und eine Kante (j_1, j_2) in A.*

Durch die dritte Forderung wird die Behandlung von Basenpaaren als Einheit sichergestellt. Entweder sind beide Enden eines Basenpaars aligniert oder keines. Da der von mir verwendete Algorithmus zur Berechnung lokaler paarweiser Alignments die Struktur zusammen mit dem Alignment berechnet, wird diese Forderung von MuLoRA allerdings von vornherein erfüllt.

Definition 12 entspricht damit genau der Vorstellung von zusammenhängenden Teilstrukturen. In ihrer Arbeit beweisen Backofen und Will [BW04] dazu, dass ein Alignment A zweier RNA-Sequenzen S_{R_1} und S_{R_2} genau dann lokal ist, wenn dessen Motiv-Graphen G_1^A und G_2^A jeweils zusammenhängend sind. Als Beispiel zeigt Abbildung 2.5 ein lokales Alignment der beiden SECIS-Motive aus Abbildung 2.3, welchen die Exklusionen aus Abbildung 2.4 enthalten, sowie deren zusammenhängende Motiv-Graphen.

Nachdem MuLoRA alle lokalen paarweisen Alignments berechnet hat, besteht die nächste Aufgabe darin, aus den erhaltenen Alignmentkanten ein multiples Alignment abzuleiten. Dabei stellt ein multiples Alignment eine natürliche Erweiterung eines paarweisen Alignments auf mehr als zwei Ribonukleinsäuren dar.

Definition 13 (multiples Alignment)
Seien R_1 bis R_m RNA-Moleküle und S_{R_1} bis S_{R_m} deren Sequenzen. Ferner sei ein Alignmentalphabet Σ_{A^m} wie folgt definiert:

$$\Sigma_{A^m} =_{def} (\{1 \ldots |S_{R_1}|\} \cup \{-\}) \times \ldots \times (\{1 \ldots |S_{R_m}|\} \cup \{-\}) \setminus \{(-, \ldots, -)\}\,.$$

Dann ist ein multiples Alignment A^m von S_{R_1} bis S_{R_m} ein Wort über Σ_{A^m}, bei dem für jedes $g \in \{1, \ldots, m\}$ die Aneinanderreihung aller g-ten Elemente ungleich '$-$' der einzelnen Kanten in der Reihenfolg ihres auftretens in A^m ein Wort ergibt, welches aus aufsteigend sortierten Zahlen besteht.

Weiterhin bezeichnet:

- $A^m[\,i\,]$ *die i-te Kante von A^m,*
- $A^m[\,g\,][\,i\,]$ *das g-te Element mit $g \in \{1, \ldots, m\}$ der i-ten Kante von A^m und*
- $|A^m|$ *die Anzahl der Kanten innerhalb des Alignments A^m.*

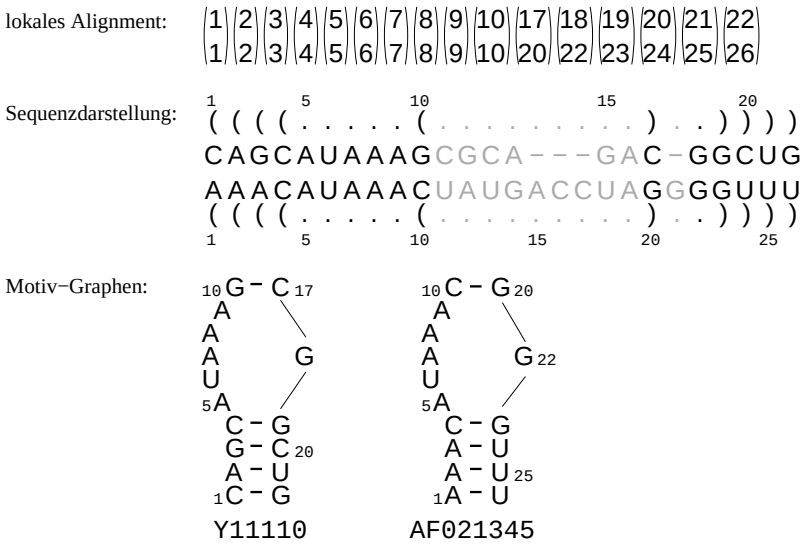

Abbildung 2.5: Ein lokales Alignment der beiden SECIS-Motive aus Abbildung 2.3. Die Sequenzdarstellung entspricht dabei einem globalen Alignment, wobei Alignmentkanten, welche im lokalen Alignment fehlen, grau gekennzeichnet sind. Die Motiv-Graphen bleiben dabei trotz der Exklusionen zusammenhängend.

Da nach dieser Definition nicht jede Sequenzposition der m Sequenzen in einer Kante vorkommen muss, brauchen die multiplen Alignments in Analogie zu Definition 8 der paarweisen Alignments ebenfalls nicht global sein.

Damit wäre es theoretisch möglich, die Lokalität eines multiplen Alignments durch das weglassen bestimmter Kanten zu definieren. Ich habe mich jedoch bei meinen Ansatz gegen diese Form der Lokalität entschieden und verwende statt dessen für jede Spalte des multiplen Alignments ein Maß für die Konserviertheit der in der Spalte alignierten Positionen. Genauer werde ich darauf im nächsten Kapitel eingehen.

Damit wären alle wichtigen Begriffe formalisiert und es kann nun zur Konkretisierung der Problemstellungen übergegangen werden.

2.2 Problemstellungen

Das Ziel des MuLoRA-Ansatzes ist es, aus einer Menge von RNA-Sequenzen konservierte Strukturen zu entdecken. Um dieses Ziel zu erreichen, ist es notwendig ein multiples Sequenz-Struktur-Alignment der Eingabesequenzen zu berechnen. Da allerdings multiple Sequenzalignments ein Spezialfall von multiplen Sequenz-Struktur-Alignments sind – nämlich diejenigen, bei denen die Strukturmenge P leer ist – folgt aus der NP-härte des multiplen Sequenzalignment-Problems [Jus01] auch die NP-härte des multiplen Sequenz-Struktur-Alignment-Problems. Aus diesem Grund kann kein Algorithmus exisitieren[1], welcher ein optimales multiples Sequenz-Struktur-Alignment in polynomialer Zeit berechnet.

Damit bleiben für einen verwendbaren Ansatz nur noch heuristischen Verfahren, wobei ich mich dabei aufgrund der Robustheit und der guten Ergebnisse für ein progressives Alignmentverfahren entschieden habe. Damit reduziert sich das eigentliche Problem der Bestimmung eines multiplen Alignments auf die Bestimmung

[1] Vorrausgesetzt $P \neq NP$

von optimalen paarweisen Alignments für die $m(m-1)/2$ verschiedenen Paare der m Eingabesequenzen.

2.2.1 Das paarweise lokale Alignment-Problem

Für die Berechnung der paarweisen Alignments unter der Berücksichtigung der Struktur existieren viele Möglichkeiten. Da MuLoRA konservierte Strukturen finden soll, habe ich mich für den Ansatz von Hofacker *et al.* [HBS04] entschieden. Dieser hat den Vorteil, dass er nicht mit einer starren vorgegebenen Struktur arbeitet, sondern nach der thermodynamisch stabilsten, konservierten – also in beiden Sequenzen enthaltenen – Struktur sucht und dazu alle möglichen genesteten Strukturen betrachtet. Die strukturellen Eigenschaften der Sequenzen werden dabei aus den Basenpaarwahrscheinlichkeiten gewonnen. Dadurch können aufwendig zu berechnende Energiemodelle, wie sie beispielsweise der Sankoff-Algorithmus [San85] verwendet, vermieden werden.

Allerdings berechnet der Ansatz von Hofacker nur globale Alignments. Da ich jedoch nur Teilstrukturen suche, ergeben sich zwei zusätzliche Anforderungen an die Bewertungsfunktion. Einerseits muss die Bewertungsfunktion auf Ähnlichkeiten anstatt auf Distanzen beruhen, da ansonsten immer das leere Alignment einen optimalen Score von Null hätte.

Andererseits muss sich die Bewertungsfunktion dabei sowohl aus positiven als auch negativen Beiträgen zusammensätzen, da eine Erweiterung oder Verkleinerung der optimalen Teilstruktur zu einem schlechteren Score führen muss. Bestünde die Bewertungsfunktion nur aus positiven Beiträgen, würde hingegen immer die komplette Struktur den maximalen Score erhalten.

Da das Bewertungschema des Hofacker-Ansatzes diese beiden Bedingungen erfüllt, ist es auch für lokale Alignments geeignet und ich kann es unverändert übernehmen.

Seien R_1 und R_2 zwei RNAs, $\mathfrak{S}_{R_1} = (S_{R_1}, P_{R_1})$ und $\mathfrak{S}_{R_2} = (S_{R_2}, P_{R_2})$ deren Sekundärstrukturen und A ein paarweises lokales Alignment von S_{R_1} und S_{R_2}. Ferner sei ρ_S eine Funktion, welche Basenpaaren über einer Sequenz S in Abhängigkeit ihrer Wahrscheinlichkeiten ein Gewicht zuordnet, σ und τ zwei Substitutionsfunktionen, γ der Strafscore für ein Gap und Γ_A die Anzahl der Gaps in A. Dann ergibt sich der paarweise lokale Alignment-Score s_{pla} von A wie folgt:

$$
\begin{aligned}
&s_{pla}(A, \mathfrak{S}_{R_1}, \mathfrak{S}_{R_2}) = \\
&\qquad \sum_{\substack{a_1 \in P_1^A, a_2 \in P_2^A \\ (a_1^l, a_2^l) \in A, (a_2^r, a_2^r) \in A}} \Big(\rho_{S_{R_1}}(a_1) + \rho_{S_{R_2}}(a_2) + \tau\big(S_{R_1}[a_1^l], S_{R_1}[a_1^r], S_{R_2}[a_2^l], S_{R_2}[a_2^r]\big) \Big) \\
&\qquad + \sum_{\substack{i_1 \in \Pi_1^A, i_2 \in \Pi_2^A \\ (i_1, i_2) \in A}} \Big(\sigma\big(S_{R_1}[i_1], S_{R_2}[i_2]\big) \Big) + \gamma \cdot \Gamma_A
\end{aligned}
\tag{2.1}
$$

Das Gewicht $\rho_S(a)$ eines Basenpaarses a über der Sequenz S sollte dabei nur von der Wahrscheinlichkeit $p_S(a)$ abhängen, mit der es über S ausgebildet wird. Da die Basenpaarwahrscheinlichkeiten über allen möglichen genesteten Strukturen einer Sequenz bestimmt werden, existieren jedoch viele Basenpaare mit einer Wahrscheinlichkeit größer Null, die in natürlichen Strukturen niemals vorkommen. Deshalb sollten solche Basenpaare eine negative Bewertung bekommen, wärend alle anderen eine positive Bewertungen erhalten. Damit wäre

$$\rho_S(a) = \log \frac{p_S(a)}{p_{rand}(a, S)}$$

wobei $p_{rand}(a, S)$ der Wahrscheinlichkeit entspricht, dass das Basenpaar a zufällig in der Sequenz S ausgebildet wird, eine ideale Bewertungsfunktion. Problematischer weise hängt p_{rand} jedoch von so vielen Parametern wie beispielsweise der Sequenzlänge, den Anfangs- und Endpositionen des Basenpaars, der Sequenzzusammensetzung und ähnlichem ab, das es nicht möglich ist, p_{rand} abzuschätzen.

Aus diesen Grund verwende ich statt p_{rand} die kleinsten Paarwahrscheinlichkeit p_{sig}, welche für eine Struktur noch signifikant ist. Damit ergibt sich letztendlich folgende Gewichtsfunktion:

$$\rho_S(a) = \log \frac{p_S(a)}{p_{sig}}$$

Auf die Bestimmung von p_{sig} und den restlichen Parametern der Bewertungsfunktion gehe ich ausführlich in Kapitel 4 ein.

Aufgrund der Bewertungsfunktion ist es nun möglich, das paarweise lokale Alignment-Problem zu definieren.

Definition 14 (paarweise lokale Alignment-Problem)
Seien S_{R_1} und S_{R_2} die Sequenzen zweier RNA-Moleküle R_1 und R_2. Dann besteht das paarweise lokale Alignment-Problem (PLA-Problem) daraus,

$$\arg\max_A \big\{ s_{pla}\,(A, \mathfrak{S}_{R_1}, \mathfrak{S}_{R_2}) \mid \\ \mathfrak{S}_A = (S_{R_1}, P_{R_1}) \text{ mit } P_{R_1} \text{ ist genestete Strukturmenge über } S_{R_1}, \\ \mathfrak{S}_B = (S_{R_2}, P_{R_2}) \text{ mit } P_{R_2} \text{ ist genestete Strukturmenge über } S_{R_2} \text{ und} \\ A \text{ ist paarweises lokales Alignment} \big\}$$

über S_{R_1} und S_{R_2} zu bestimmen.

Bei der Bestimmung des Ähnlichkeitsscores eines Alignments kann man im einfachsten Fall $\sigma = \tau = 0$ setzen und so die sequenzspezifischen Komponenten ausblenden. Damit würde das Ergebnis aus der wahrscheinlichsten Teilstruktur bestehen, welche in den beiden Sequenzen enthalten ist.

Aus den so berechneten paarweisen lokalen Alignments leitet das progressive Alignmentverfahren das multiple Alignment ab. Damit besteht das nächste Problem aus der Bestimmung der Konsensussequenz sowie der Konsensusstruktur.

2.2.2 Die Konsensus-Probleme

Die Konsensussequenz S_C besteht aus den Konsensusbuchstaben der einzelnen Spalten eines multiplen Alignments. Die Konsensusbuchstaben wiederrum bestehen im Allgemeinen aus dem häufigsten Buchstaben der Spalten. Etwas formaler Ausgedrückt bedeutet dies folgendes:

Definition 15 (Konsensussequenz-Problem)
Sei A^m ein multiples Alignment, Σ_C das Konsensusalphabet und $\nu(\omega_i, A^m[\,i\,])$ eine Funktion, welche die Ähnlichkeit zwischen einem $\omega_i \in \Sigma_C$ und der i-ten Spalte von A^m berechnet. Dann besteht das Konsensussequenz-Problem daraus, für jede Spalte i in A^m

$$\arg\max_{\omega_i \in \Sigma_C} \Big\{ \nu\big(\omega_i, A^m[\,i\,]\big) \Big\}$$

zu bestimmen.

Die Ähnlichkeitsfunktion ν kann dabei ziemlich beliebig definiert werden. Die in MuLoRA verwendete Funktion berechnet beispielsweise die Häufigkeit von ω_i in $A^m[\,i\,]$. Allerdings unterscheide ich bei der Bestimmung des Konsensusbuchstabens zusätzlich noch zwischen konservierten und unkonservierten Spalten und verwende

einen Grenzwert für die Häufigkeit. Genauer gehe ich darauf im entsprechenden Abschnitt in Kapitel 3 ein.

Das Konsensusstruktur-Problem besteht darin, eine Strukturmenge P_C über der Konsensussequenz S_C eines multiplen Alignments A^m zu finden, welche eine maximale Summe aller Basenpaargewichte der in ihr enthaltenen Basenpaare besitzt.

Definition 16 (Konsensusstruktur-Problem)
Sei A^m ein multiples Alignment, S_C dessen Konsensussequenz und $\vartheta_{S_C}(a)$ eine Funktion, welche einem Basenpaar über S_C ein Gewicht zuordnet. Dann besteht das Konsensusstruktur-Problem daraus,

$$\arg\max_{P_C} \Big\{ \sum_{a \in P_C} \vartheta_{S_C}(a) \Big\}$$

über alle genesteten Strukturmengen P_C zu bestimmen.

In Analogie zu den Basenpaargewichten der Eingabesequenzen hängen auch die Basenpaargewichte der Konsensussequenz von Basenpaarwahrscheinlichkeiten ab.

$$\vartheta_{S_C}(a) = \log \frac{p_{S_C}(a)}{p_{sig}}$$

Dabei gibt $p_{S_C}(a)$ die Wahrscheinlichkeit an, mit der sich a über S_C ausbildet. Auf die Berechnung dieser Wahrscheinlichkeiten gehe ich ebenfalls genauer in Kapitel 3 ein.

Damit sind nun alle Problemstellungen definiert. Im nächsten Kapitel folgt deren Lösung.

Kapitel 3

Der MuLoRA Ansatz

In diesem Kapitel werde ich mich nun dem eigentlichen Algorithmus zuwenden. Nach einem allgemeinen Überblick über die Funktionsweise zusammen mit den Hintergründen liegt dabei das Hauptaugenmerk auf den einzelnen Bestandteilen. Abschließend folgt eine Zeit- und Speicheranalyse.

3.1 Überblick

Wie in der Einleitung schon begündet, sollte ein Programm für die Suche nach konservierte Teilsstrukturen in einer Menge von RNA-Sequenzen drei wichtige Anforderungen erfüllen: Erstens müssen bei der Suche mehrere Moleküle verglichen werden, um konservierte Bereiche zu entdecken. Zweitens müssen dabei sowohl die Sequenz als auch die möglichen Strukturen in Betracht gezogen werden, um die fehlende Konserviertheit auf Sequenzebene auszugleichen. Und drittens sollten diese Sequenz-Struktur-Vergleiche in einer bestimmten strukturellen Art und Weise lokal sein, um so nur die Bereiche zu betrachten welche auch konserviert sind.

Damit besteht der Ausgangspunkt bei der Entwicklung eines erfolgreichen Programms zuerst einmal aus einem multiplen Alignment der Ribonukleinsäuren. Ich habe mich dabei Augrund der Robustheit und der Effiziens für einen progressives Alignmentverfahren entschieden.

Multiples Alignment

Einer der erfolgreichsten Ansätze zu progressiven Alignments stammt von Notredame *et al.* [NHH00] mit ihren Programm T-Coffee (Tree-based Consistency Objective Function for alignment Evaluation). Herkömmliche progressive Ansätze leiden unter ihrer geizigen Natur. Fehler die in den ersten Alignments gemacht werden, können später beim Einfügen der restlichen Sequenzen nicht mehr verbessert werden ("Once a gap, always a gap!"). T-Coffee minimiert dieses Problem, indem das Programm beim alignieren zweier Sequenzen die Informationen aller restlichen Sequenzen mit einfliesen lässt und so Fehler am Anfang auf ein Minimum reduziert.

Ein weiterer Vorteil von T-Coffee liegt darin, dass die Grundlage für die Berechnung multipler Alignments aus einer Bibliothek mit den gewichteten Kanten aller paarweisen Alignments über den Eingabesequenzen besteht. Die Bibliothek wird normalerweise von T-Coffe selbst berechnet, kann aber auch als Eingabe vorgegeben werden. Damit ist es möglich, Positionsinformationen von verschiedenen Alignmentprogrammen oder aber auch strukturelle Einschränkungen einfliesen zu lassen.

Damit besteht das nächste Problem darin, für alle Paare der Eingabesequenzen ein Alignment zu berechnen. Idealer weise sollten die Alignments dabei nur

die konservierten Regionen repräsentieren und neben den sequenziellen auch die strukturellen Informationen der Regionen beinhalten. Ein multiples Alignment auf Grundlage dieser Alignmentkanten würde dann alle Anforderungen erfüllen, die für eine erfolgreiche Suche nach konservierten Motiven notwendig sind.

Paarweise Alignments

Einen der erfolgsversprechensten Ansätze zur Berechnung der paarweisen Alignments fand ich in *pmcomp* von Hofacker *et al.* [HBS04]. Der Vorteil dieses Programms liegt darin, dass es zwei Sequenzen aligniert und dabei gleichzeitig mit Hilfe der Basenpaarwahrscheinlichkeiten die wahrscheinlichste gemeinsamme Sekundärstruktur über den beiden Sequenzen bestimmt. Damit betrachtet der Algorithmus nicht nur eine oder mehrere fest vorgegebene Strukturen, sondern greift bei der Suche nach einer gemeinsammen Struktur auf alle theoretisch möglichen genesteten Strukturen der beiden Sequenzen zurück.

Allerdings berechnet pmcomp ein globales Alignment und ist damit in seiner eigentlichen Form ungeeignet. Aus diesem Grund habe ich das Programm dahingehend erweitern, dass es ein lokales Alignment berechnet. Als Vorlage diente mir dabei eine Arbeit von Backofen und Will [BW04], in welcher sie eine strukturelle Lokalität vorstellten und einen Algorithmus zur Berechnung eines paarweisen lokalen Alignments zweier Sequenzen mit gegebener Struktur präsentierten. Das Ergebniss dieser Erweiterung des Hofacker-Ansatzes ist ein Algorithmus, welcher das in Definition 14 vorgestellte paarweise lokale Alignment-Problem löst.

Jedoch können zwei Sequenzen durchaus mehr als nur eine gemeinsamme Teilstruktur aufweisen. In diesem Fall kann man nicht davon ausgehen, dass immer nur diejenige Teilstruktur mit dem höchsten Score in den anderen Sequenzen konserviert ist. Deshalb berechne ich neben dem besten lokalen Alignmnet zweier Sequenzen, also demjenige mit den höchsten Score, auch noch die nächst besten lokalen Alignments.

Dabei habe ich zwei verschiedene Methoden realisiert. Die erste berechnet in Analogie zu Programmen wie BLAST [AGM90] die k-besten, nicht überlappenden lokalen Alignments. Damit ist es möglich, alle gemeinsammen Teilstrukturen zweier Sequenzen gleichzeitig zu finden.

Die zweite Methode verzichtet hingegen auf die Einschränkung, nur nicht-überlappende Alignments zu betrachten. Der Vorteil dieses Ansatzes besteht darin, dass so alternative bzw. besonders stabile Strukturbereiche gefunden werden können. Abbildung 3.1 zeigt ein Beispiel für die beide Ansätze.

Mit diesem Algorithmus ist es nun möglich, Alignments zu berechnen, die sequenzielle und strukturelle Informationen über konservierte Teilstrukturen liefern. Damit bleibt nun noch das Problem, die Alignmentkanten zu gewichten.

Kantengewichte und Konserviertheit

Das Gewicht einer Kante sollte ihrer Zuverlässigkeit entsprechen. Je höher dieses ist, um so eher wird die Kante auch von T-Coffee im multiplen Alignment eingebaut.

Die Zuverlässigkeit einer Kante hängt dabei direkt von der Zuverlässigkeit des Alignments ab, in der sie sich befindet. Die Zuverlässigkeit des Alignments wiederrum hängt von der Ähnlichkeit der beiden alignierten Teilstrukturen ab. Je ähnlicher sich die beiden sind, um so unwahrscheinlicher ist es, dass sich die Bereiche zufällig gleichen. Da die Alignments auf der Grundlage von Ähnlichkeiten berechnet werden, ist der Score eines Alignments auch gleichzeitig ein Mass für die Ähnlichkeit der beiden Sequenzen und damit auch ein Mass für die Zuverlässigkeit des Alignments selbst.

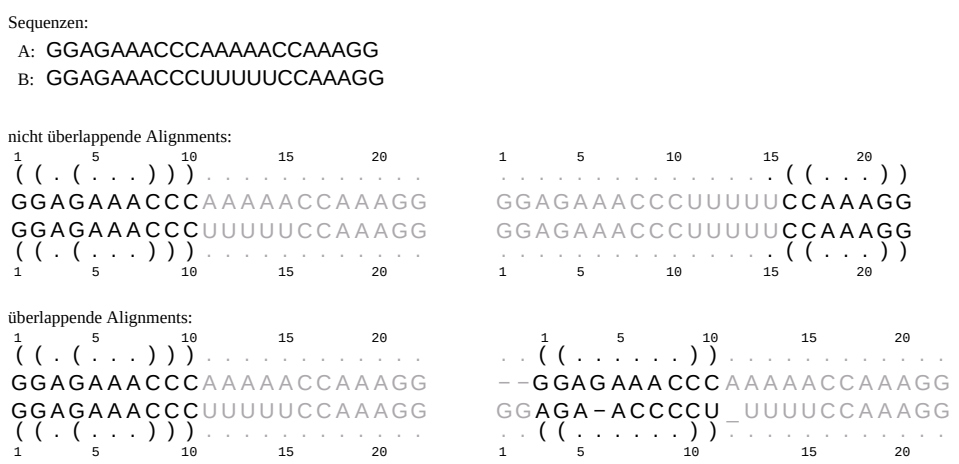

Abbildung 3.1: Beispiel für zwei überlappende und nicht überlappende Alignments

Aus diesem Grund verwende ich auch den Score eines Alignments für die Gewichtung der Kanten. Dazu teile ich einfach den Score zwischen allen Alignmentkanten zwischen zwei Basen auf. Falls eine Kante mehrfach vorkommen sollte – dies ist bei sich überlappenden Alignments möglich – werden die einzelnen Gewichte addiert. Alle restlichen Kanten, also diejenigen zwischen einer Base und einem Gap, erhalten ein Gewicht von Null, da sie keine Informationen für die Ableitung des multiplen Alignments enthalten.

Aus den gewichteten Alignmentkanten berechnet T-Coffee ein globales multiples Alignment. Da die Alignmentkanten jedoch aus lokalen Alignments stammen, werden die einzelnen Spalten des berechneten multiplen Alignments unterschiedlich stark von Kanten bestätigt.

Je stärker dabei die Bestätigung ist, um so größer ist auch der Beleg dafür, dass die in der Spalte alignierten Positionen in jeweils ähnlichen Motiven vorkommen. Damit ist die Anzahl der Kanten, die eine Spalte bestätigen, ein perfektes Maß für die Konserviertheit und damit auch für die Lokalität der Spalte.

Allerdings hängt die Anzahl der Kanten, die eine Spalte bestätigen können, von der Anzahl der Sequenzen im multiplen Alignment ab. Aus diesem Grund normiere ich die Anzahl mit der maximalen Kantenanzahl pro Spalte. Letztere beträgt bei einem Alignment von m Sequenzen $m(m-1)/2$ Kanten. Insgesamt wird so aus dem globalen multiplen Alignment ein lokales multiples Alignment, welches für jede Spalte die Konserviertheit der in ihr alignierten Positionen angibt. Abbildung 3.2 zeigt ein Beispiel für ein lokales multiples Alignment.

Die letzte Aufgabe für MuLoRA besteht nun darin, aus dem multiplen Alignment die Konsensussequenz und -Struktur abzuleiten.

Konsensussequenz und -Struktur

Die Konsensussequenz ergibt sich aus der Aneinanderreihung der Konsensusbuchstaben der einzelnen Spalten. Der Konsensusbuchstabe wiederum ergibt sich im Allgemeinen aus dem häufigsten Buchstaben der jeweiligen Spalte. Dabei unterscheide ich jedoch zusätzlich zwischen konservierten und unkonservierten Spalten.

Eine Spalte gilt dabei genau dann als unkonserviert, wenn der Kantenanteil der sie unterstützt unter einem Grenzwert dafür liegt. Indem unkonservierte Spalte ein '−' als Konsensusbuchstaben erhalten, können Spalten welche nicht von genügend Alignmentkanten bestätigt werden und deshalb von T-Coffee nur zufällig zusammengesetzt wurden, keinen Einfluß auf die Konsensussequenz nehmen.

Abbildung 3.2: Ein lokales multiples Alignment dreier Sequenzen. Die Werte unter den einzelnen Spalten geben dabei den Anteil der paarweisen Alignmentkanten in Prozenten an, welche diese Spalte unterstützen.

Konserviert Spalten hingegen erhalten als Konsensusbuchstaben entweder den häufigsten Buchstaben, sofern dessen Anteil in der Spalte über einem Grenzwert dafür liegt oder ein 'N' im anderen Fall. Diese Unterscheidung stellt sicher, dass der Anteil des Konsenusbuchstabens gegenüber den Anteilen der restlichen Buchstaben groß genug ist.

Für die Ableitung der Konsensusstruktur berechnet MuLoRA zuerst eine Konsensuspaarwahrscheinlichkeitmatrix, welche für alle Paare von Spalten die Wahrscheinlichkeit enthält, dass in der Konsensusstruktur zwischen den beiden Spalten ein Basenpaar existiert. Diese Wahrscheinlichkeiten ergeben sich dabei aus dem aritmetischen Mittel über die Basenpaarwahrscheinlichkeiten der einzelnen Sequenzen.

Anschließend werden die Wahrscheinlichkeiten in Basenpaargewichte transformiert, wobei Basenpaare über einer unkonservierten Spalte ein Gewicht von Null erhalten. Damit wird in Analogie zu der Konsensussequenz verhindert, dass Basenpaare in unkonservierten Spalten beginnen oder enden. Die Konsensusstruktur ergibt sich dann einfach aus derjenigen Struktur, welche die maximale Summe aller Basenpaargewichte der in ihr enthaltenen Basenpaare hat.

Zusammenfassend besteht der Algorithmus also aus folgenden Punkten:

1. Bestimmen der Basenpaarwahrscheinlichkeiten der Eingabesequenzen
2. Berechnen der paarweisen lokalen Sequenz-Struktur-Alignments für alle Paare der Eingabesequenzen
3. Bestimmen der Kantengewichte und Berechnung des multiplen Alignments anhand der gewichteten Alignmentkanten
4. Ableiten der Konsensussequenz und Konsensusstruktur des multiplen Alignments

Damit ist der allgemeine Überblick über die Funktionsweise von MuLoRA abgeschlossen.

3.2 Bestandteile

In diesem Abschnitt wende ich mich den einzelnen Bestanteilen des MuLoRA-Ansatzes in der Reihenfolge ihrer Verwendung zu. Somit erläutere ich zuerst die Berechnung der Basenpaarwahrscheinlichkeiten.

3.2.1 Basenpaarwahrscheinlichkeiten

Für die Berechnung der paarweisen Alignments benötigt der Algorithmus strukturelle Informationen in Form von Basenpaarwahrscheinlichkeiten. Diese werden dabei unabhängig vom späteren Alignmentprozess berechnet, wodurch verschiedene Berechnungsansätze oder Kombinationen von Ansätzen verwendet werden können. Ich habe mich für das Programm *RNAfold* des Vienna RNA secondary structure package von Hofacker [Hof03] entschieden.

RNAfold berechnet primär die MFE-Struktur – also die Struktur mit minimaler freier Energie – einer Sequenz auf Grundlage des Zuker-Algorithmus [ZS81]. Zusätzlich können dabei aber auch die Basenpaarwahrscheinlichkeiten unter Verwendung des McCaskill-Algorithmus [McC90] berechnet werden. Dabei verwende ich die standardmäßigen Energieparameter von Turner *et al.* [MSZ99].

Der McCaskill-Algorithmus

Allgemein wird die Struktur einer Ribonukleinsäure durch die Sequenz der Basen und spezifischen Interaktionen zwischen ihnen – wie beispielsweise Wasserstoffbrückenbindungen, Ringsystem-Konjugationen, hydrophobe Wechselwirkungen und elektrostatischen Interaktionen zwischen negativ geladenen Phosphatgruppen und gelösten Ionen – bestimmt. Allerdings haben RNA-Moleküle die „schlechte“ Eigenschaft, dass sie nicht unbedingt in der thermodynamisch stabilsten Form, also in der MFE-Struktur auftreten. Deshalb reicht auch im Allgemeinen ein einfacher Energieminimierungsansatz, wie beispielsweise der Zuker-Algorithmus, für eine zuverlässige Strukturvorhersage nicht aus.

McCaskill löst dieses Problem, indem er die freie Energie $E[P]$ einer Struktur P mit Hilfe der Gibbs-Bolzmann-Gleichung zu einer Strukturwahrscheinlichkeit konvertiert. Insgesamt ergibt sich so eine Wahrscheinlichkeitsverteilung über Strukturen, welche bei gegebener Durchschnittsenergie die Entropie maximiert.

$$p[P_i] = \frac{e^{-\frac{E[P_i]}{kT}}}{\sum_{P_j} e^{-\frac{E[P_j]}{kT}}}$$

Der Nenner dieser Gleichung $\sum_P e^{-(E[P]/kT)}$ wird auch als Partitionsfunktion Q der Bolzmann-Verteilung bezeichnet.

Um nun die freie Energie einer Struktur zu bestimmen, zerlegt McCaskill die Struktur in externe Basen, Hairpins, Stacks, interne Loops und multiple Loops. Abbildung 1.5 zeigt eine Übersicht über die verschiedenen Strukturbestantteile.

Damit ergibt sich die freie Energie einer Struktur als Summe über die freien Energien der einzelnen Bestandteile L. Deren Beiträge wurden dabei aus Modelmolekülen gewonnen.

$$E[P] = \sum_{L \in P} E_L$$

Allerdings müssen dabei aus Komplexitätsgründen die Berechnungen auf genestete Strukturen mit vereinfachten Annahmen für die Beiträge von großen und multiplen Loops beschränkt werden. Die Addition der freien Energie führt bei der Partitionsfunktion zu einer Multiplikation der Beiträge.

$$Q = \sum_P \prod_{L \in P} e^{-\frac{E[P_L]}{kT}}$$

Um nun diese Formel berechnen zu können, werden die Strukturen in unabhängige bzw. disjunkte Teilmengen zerlegt, wodurch es mit einem rekursiven Schema und einem dynamischen Programmieransatz möglich ist, die Partitionsfunktion in $O(n^3)$ Zeit und mit $O(n^2)$ Speicher in Abhängigkeit der Sequenzlänge n zu berechnen.

Damit ist es nun einfach, die Wahrscheinlichkeit einer Struktur P zu bestimmen, welche sich wie folgt berechnen lässt:

$$p[P] = \frac{e^{-\frac{E[P]}{kT}}}{Q}$$

Jedoch ist die Wahrscheinlichkeit einer einzelnen Sequenz biologisch nicht besonders interessant. Deshalb berechnet man die Wahrscheinlichkeiten von bestimmten Teilstrukturen. Dazu summiert man einfach alle Wahrscheinlichkeiten derjenigen Strukturen, welche diese Teilstruktur enthalten. Beschränkt man Teilstrukturen auf einzelne Basenpaare a, erhält man eine der wichtigsten Kenngrößen für die Beschreibung von Strukturen über einer Sequenz: die Basenpaarwahrscheinlichkeiten.

$$p[a] = \sum_{P \ni a} p[P] = \frac{\sum_{P \ni a} e^{-\frac{E[P]}{kT}}}{Q}$$

Da die Basenpaarwahrscheinlichkeiten alle Strukturen reflektieren, sind sie vollkommen unabhängig voneinander. Deshalb geben sie auch die Informationen über alle möglichen globale Strukturen wieder. Bei der Berechnung des Zählers kann man auf die Zwischenergebnisse der Partitionsfunktion Q zurückgreifen. Dabei muss man jedoch beachten, dass ein Basenpaar sowohl in externen bzw. nicht schließenden Positionen vorkommen kann, aber auch von anderen Basenpaaren umschlossen werden kann. Letzteres hat wiederum eine Zerlegung in disjunkte Mengen zur Folge. Insgesamt kann man so in $O(n^3)$ Zeit und mit $O(n^2)$ Speicher alle Basenpaarwahrscheinlichkeiten einer Sequenz der Länge n bestimmen.

Mit Hilfe dieser strukturellen Informationen ist es nun möglich, die paarweisen lokalen Alignments zu berechnen.

3.2.2 Paarweise lokale Seqenz-Struktur-Alignments

Bei den von mir entwickelten Algoritmus zur Lösung des in Definition 14 vorgestellten paarweisen lokalen Alignment-Problems handelt es sich um ein dynamisches Programmierverfahren. Dieses ermittelt unter Verwendung von Rekursionsgleichungen, deren Zwischenergebnisse für eine effiziente Berechnung in Tabellen gespeichert werden, den maximalen Score eines lokalen Alignments zweier Sequenzen. Die Lösung des paarweisen lokalen Alignment-Problems – also das optimale lokale Alignment – ergibt sich dann mittels Backtracking aus den Tabellen der Zwischenergebnisse.

Die Rekursionsgleichungen

Das Rekursionsschema des Algorithmus wird von zwei Arbeiten geprägt. Die Berechnung des Alignments bei gleichzeitiger Vorhersage einer gemeinsammen Struktur lehnt sich an einen Algorithmus von Hofacker *et al.* [HBS04] an. Die von mir verwendete Form der strukturellen Lokalität stammt hingegen aus einer Arbeit von Backofen und Will [BW04].

Der Hofacker-Algorithmus berechnet ein globales Alignment und findet dabei gleichzeitig die wahrscheinlichste gemeinsamme Sekundärstruktur zweier Sequenzen. Das paarweise lokale Alignment-Problems besteht hingegen daraus, das optimale lokales Sequenz-Struktur-Alignment über zwei Sequenzen zu finden. Da sowohl der Hofacker-Ansatz als auch das Alignment-Problem auf der selben Bewertungsfunktion (siehe Gleichung 2.1) beruhen, müssen die Hofacker-Rekursionsgleichungen für eine Lösung des Alignment-Problems nur dahingehend erweitert werden, dass sie ein nach Definition 12 lokales Alignment berechnen.

Der Hofacker-Algorithmus beruht dabei auf zwei Rekursionsgleichungen. Die erste Gleichung M wird dazu verwendet, um für jedes Paar von Teilsequenzen $S_{R_1}[\,i_1, j_1\,]$ und $S_{R_2}[\,i_2, j_2\,]$ über den beiden Eingabesequenzen S_{R_1} und S_{R_2} den Score $M(i_1, j_1, i_2, j_2)$ des optimalen globalen Alignments von $S_{R_1}[\,i_1, j_1\,]$ und $S_{R_2}[\,i_2, j_2\,]$ zu berechnen.

Dazu werden vier Rekusionsfälle unterschieden. Im ersten Fall wird ein Alignment um zwei alignierte Basen erweitert und dessen Score um die Ähnlichkeit σ der beiden Basen erhöht. Im zweiten und dritten Fall wird ein Alignment um eine Baseninsertion bzw. Basendeletion erweitert und zu dessen Score die Gap-Kosten addiert. Im vierten Fall wird schließlich ein Alignment, dessen Enden zwei gematchte Basenpaare bilden in ein anderes Alignment eingefügt und der Score der beiden Alignments kombiniert. Durch den letzten Rekusionsfall wird als Nebenprodukt zum eigentlichen Alignment auch eine gemeinsamme genestete Sekundärstruktur über den Sequenzen berechnet. Insgesamt ergibt sich also folgende Rekursionsgleichung:

$$M(i_1, j_1, i_2, j_2) = \max \begin{cases} M(i_1, j_1 - 1, i_2, j_2 - 1) + \sigma\big(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\big) \\ M(i_1, j_1 - 1, i_2, j_2) + \gamma \\ M(i_1, j_1, i_2, j_2 - 1) + \gamma \\ \max\limits_{\substack{i_1 < k_1 < j_1 \\ i_2 < k_2 < j_2}} \Big\{ M(i_1, k_1, i_2, k_2) + D\big((k_1 + 1, j_1), (k_2 + 1, j_2)\big) \Big\} \end{cases} \tag{3.1}$$

Dabei entspricht D der zweiten Rekursionsgleichung des Hofacker-Algorithmus. Diese wird dazu verwendet, um für jedes mögliche Paar von Basenpaaren a_1 über der Sequenz S_{R_1} und a_2 über der Sequenz S_{R_2} den Score $D(a_1, a_2)$ desjenigen optimalen Alignments der Teilsequenzen $S_{R_1}[\,a_1^l, a_1^r\,]$ und $S_{R_2}[\,a_2^l, a_2^r\,]$ zu berechnen, bei dem a_1 und a_2 gematcht sind. Dieser ergibt sich dabei aus der Addition des Scores des Alignments der Teilsequenzen $S_{R_1}[\,a_1^l + 1, a_1^r - 1\,]$ und $S_{R_2}[\,a_2^l + 1, a_2^r - 1\,]$ und den Beiträgen der Basenpaare.

$$\begin{aligned} D(a_1, a_2) \quad = \quad & M(i_1 + 1, j_1 - 1, i_2 + 1, j_2 - 1) + \rho_{S_{R_1}}(a_1) + \rho_{S_{R_2}}(a_2) \\ & + \tau\big(S_{R_1}[\,a_1^l\,], S_{R_1}[\,a_1^r\,], S_{R_2}[\,a_2^l\,], S_{R_2}[\,a_2^r\,]\big) \end{aligned} \tag{3.2}$$

Der Score eines optimalen globalen Alignments der Sequenzen S_{R_1} und S_{R_2} ergibt sich dann aus der Berechnung von $M(1, |S_{R_1}|, 1, |S_{R_2}|)$.

Um nun ein nach Definition 12 lokales Alignment zweier Sequenzen S_{R_1} und S_{R_2} berechnen zu können, müssen zusätzliche Fälle für das Entfernen von Anfangs- und Endbereichen der Sequenzen sowie das Durchführen von Exklusionen betrachtet werden. Dabei dürfen Exklusionen nur dann in einem Alignment durchgeführt werden, falls ein konserviertes Basenpaar existiert, welches dirketer Vorgänger der Exklusion ist und keine andere Exklusion im Alignment dieses Basenpaar als direkten Vorgänger hat.

Für die Erweiterung des Algorithmus von Hofacker folgt daraus, dass in Gleichung 3.2 bei der Berechnung des Alignments zwischen den gematchten und damit konservierten Basenpaaren Exklusionen zulässig sind. Allerdins darf es dabei nur eine Exklusion pro alignierter Teilsequenz geben, welche das konservierte Basenpaar als direkten Vorgänger hat.

Um dies sicher zu stellen, verwende ich in Analogie zu der Arbeit von Backofen und Will für jeden der vier erlaubten Fälle eine eigene Rekursionsgleichung $M_{a_2}^{a_1}(j_1, j_2)$. Diese enthalten dabei für alle j_1 und j_2 mit $a_1^l < j_1 < a_1^r$ und $a_2^l < j_2 < a_2^r$ den Score eines optimalen Alignments der Teilsequenzen $S_{R_1}[\,a_1^l + 1, j_1\,]$ und $S_{R_2}[\,a_2^l + 1, j_2\,]$, wobei bei der Berechnung von

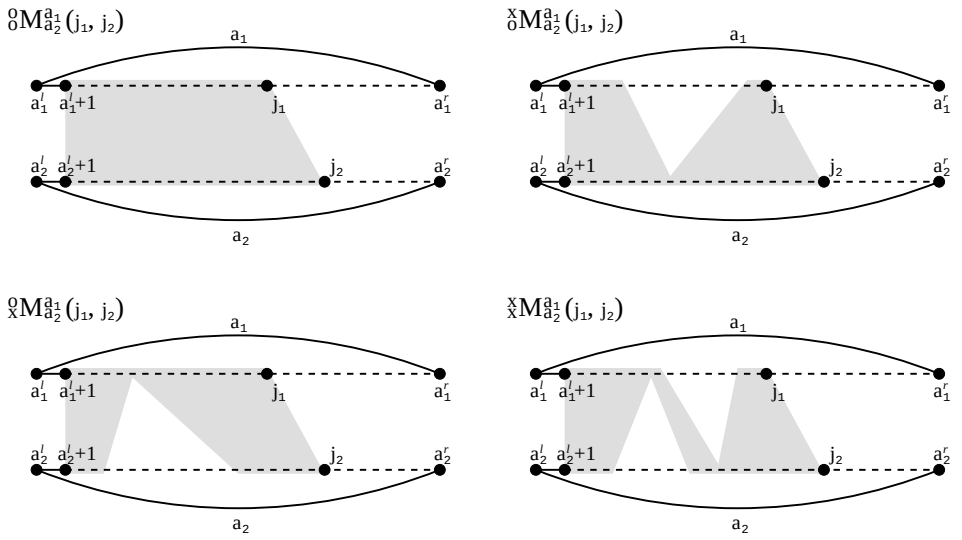

Abbildung 3.3: Beispiele für Alignments, welche von den einzelnen Rekursionsgleichungen $M_{a_2}^{a_1}(j_1, j_2)$ berechnet werden. Der grau hinterlegte Bereich entspricht dabei den Alignmentkanten.

- ${}_{x}^{x}M_{a_2}^{a_1}(j_1, j_2)$ maximal eine Exklusion mit a_1 und maximal eine Exklusion mit a_2 als direkten Vorfahren durchgeführt wurde,
- ${}_{o}^{x}M_{a_2}^{a_1}(j_1, j_2)$ maximal eine Exklusion mit a_1 als direkten Vorfahren durchgeführt wurde,
- ${}_{x}^{o}M_{a_2}^{a_1}(j_1, j_2)$ maximal eine Exklusion mit a_2 als direkten Vorfahren durchgeführt wurde und
- ${}_{o}^{o}M_{a_2}^{a_1}(j_1, j_2)$ keine Exklusion mit a_1 oder a_2 als direkten Vorfahren durchgeführt wurde.

Abbildung 3.3 zeigt für jede der vier Rekusionsgleichungen ein Beispiel für die von ihnen berechneten Alignments.

Damit ändert sich Gleichung 3.2 im paarweisen lokalen Alignmen-Algorithmus zu folgender Gleichung:

$$\begin{aligned} D(a_1, a_2) \quad = \quad & {}_{x}^{x}M_{a_2}^{a_1}(a_1^r - 1, a_2^r - 1) + \rho_{S_{R_1}}(a_1) + \rho_{S_{R_2}}(a_2) \\ & + \tau\big(S_{R_1}[\,a_1^l\,], S_{R_1}[\,a_1^r\,], S_{R_2}[\,a_2^l\,], S_{R_2}[\,a_2^r\,]\big) \end{aligned} \tag{3.3}$$

Bei den Definitionen der vier Rekursionsgleichungen ${}_{x}^{x}M_{a_2}^{a_1}$, ${}_{o}^{x}M_{a_2}^{a_1}$, ${}_{x}^{o}M_{a_2}^{a_1}$ und ${}_{o}^{o}M_{a_2}^{a_1}$ betrachte ich zuerst den Fall, dass keine Exklusion mit a_1 oder a_2 als direkten Vorgänger durchgeführt werden darf. Dieser entspricht damit vollkommen der in Gleichung 3.1 behandelten Situation, wodurch sich folgend Rekusionsgleichung ergibt:

$${}_{o}^{o}M_{a_2}^{a_1}(j_1, j_2) = \max \begin{cases} {}_{o}^{o}M_{a_2}^{a_1}(j_1 - 1, j_2 - 1) + \sigma\left(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\right) \\ {}_{o}^{o}M_{a_2}^{a_1}(j_1 - 1, j_2) + \gamma \\ {}_{o}^{o}M_{a_2}^{a_1}(j_1, j_2 - 1) + \gamma \\ \max\limits_{\substack{a_1^l < k_1 < j_1 \\ a_2^l < k_2 < j_2}} \left\{ {}_{o}^{o}M_{a_2}^{a_1}(k_1 - 1, k_2 - 1) + D\big((k_1, j_1), (k_2, j_2)\big) \right\} \end{cases} \tag{3.4}$$

Die Rekursion bricht ab, falls eine der Teilsequenzen die Länge Null hat, d.h. falls $j_1 = a_1^l$ oder $j_2 = a_2^l$. Diese Fälle sind wie folgt definiert.

$$\begin{aligned}
{}^{o}_{o}M^{a_1}_{a_2}(a_1^l, a_2^l) &= 0 \\
{}^{o}_{o}M^{a_1}_{a_2}(k_1, a_2^l) &= (k_1 - a_1^l) \cdot \gamma, \text{ für alle } k_1 \text{ mit } a_1^l < k_1 < a_1^r \\
{}^{o}_{o}M^{a_1}_{a_2}(a_1^l, k_2) &= (k_2 - a_2^l) \cdot \gamma, \text{ für alle } k_2 \text{ mit } a_2^l < k_2 < a_2^r
\end{aligned}$$

Die restlichen drei Rekusionsgleichungen beinhalten ebenfalls alle die vier Fälle für das Erweitern um zwei alignierte Basen, für das Erweitern um eine Baseninsertion bzw. Basendeletion und für das Einfügen eines Alignment, dessen Enden zwei gematchte Basenpaare bilden.

Zusätzlich kommt nun noch die Möglichkeit hinzu, dass eine Exklusion durchgeführt wurde. In dem Fall ergibt sich der Score aus dem Score an der Stelle des Exklusionsbegins. Dieser wird dabei aus einer derjenigen Rekursionsgleichungen ermittelt, die in der entsprechenden Sequenz noch keine Exklusion enthalten. Für ${}^{x}_{o}M^{a_1}_{a_2}$ bedeuted die Möglichkeit einer Exklusion in der ersten Teilsequenz folgendes:

$${}^{x}_{o}M^{a_1}_{a_2}(j_1, j_2) = \max \begin{cases}
{}^{x}_{o}M^{a_1}_{a_2}(j_1 - 1, j_2 - 1) + \sigma\left(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\right) \\
{}^{x}_{o}M^{a_1}_{a_2}(j_1 - 1, j_2) + \gamma \\
{}^{x}_{o}M^{a_1}_{a_2}(j_1, j_2 - 1) + \gamma \\
\max\limits_{\substack{a_1^l < k_1 < j_1 \\ a_2^l < k_2 < j_2}} \left\{ {}^{x}_{o}M^{a_1}_{a_2}(k_1 - 1, k_2 - 1) + D\big((k_1, j_1), (k_2, j_2)\big) \right\} \\
\max\limits_{a_1^l \le k_1 < j_1} \left\{ {}^{o}_{o}M^{a_1}_{a_2}(k_1, j_2) \right\}
\end{cases} \tag{3.5}$$

Bei den Rekursionsabbrüchen führt die Möglichkeit einer Exklusion zu Beginn der ersten Teilsequenz zu folgenden Bild:

$$\begin{aligned}
{}^{x}_{o}M^{a_1}_{a_2}(a_1^l, a_2^l) &= 0 \\
{}^{x}_{o}M^{a_1}_{a_2}(k_1, a_2^l) &= 0, \text{ für alle } k_1 \text{ mit } a_1^l < k_1 < a_1^r \\
{}^{x}_{o}M^{a_1}_{a_2}(a_1^l, k_2) &= (k_2 - a_2^l) \cdot \gamma, \text{ für alle } k_2 \text{ mit } a_2^l < k_2 < a_2^r
\end{aligned}$$

Bei ${}^{o}_{x}M^{a_1}_{a_2}$ besteht die Möglichkeit einer Exklusion in der zweiten Sequenz, wodurch sich folgende Rekursionsgleichung ergibt:

$${}^{o}_{x}M^{a_1}_{a_2}(j_1, j_2) = \max \begin{cases}
{}^{o}_{x}M^{a_1}_{a_2}(j_1 - 1, j_2 - 1) + \sigma\left(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\right) \\
{}^{o}_{x}M^{a_1}_{a_2}(j_1 - 1, j_2) + \gamma \\
{}^{o}_{x}M^{a_1}_{a_2}(j_1, j_2 - 1) + \gamma \\
\max\limits_{\substack{a_1^l < k_1 < j_1 \\ a_2^l < k_2 < j_2}} \left\{ {}^{o}_{x}M^{a_1}_{a_2}(k_1 - 1, k_2 - 1) + D\big((k_1, j_1), (k_2, j_2)\big) \right\} \\
\max\limits_{a_2^l \le k_2 < j_2} \left\{ {}^{o}_{o}M^{a_1}_{a_2}(j_1, k_2) \right\}
\end{cases} \tag{3.6}$$

In diesem Fall besteht bei den Rekursionsabbrüchen die Möglichkeit einer Exklusion zu Beginn der zweiten Teilsequenz.

$${}^{o}_{x}M^{a_1}_{a_2}(a_1^l, a_2^l) = 0$$

$$
\begin{aligned}
{}^{o}_{x}M^{a_1}_{a_2}(k_1, a_2^l) &= (k_1 - a_1^l) \cdot \gamma, \text{ für alle } k_1 \text{ mit } a_1^l < k_1 < a_1^r \\
{}^{o}_{x}M^{a_1}_{a_2}(a_1^l, k_2) &= 0, \text{ für alle } k_2 \text{ mit } a_2^l < k_2 < a_2^r
\end{aligned}
$$

Die Gleichung ${}^{x}_{x}M^{a_1}_{a_2}$ kann letztendlich eine Exklusion in jeder der beiden Teilsequenzen enthalten:

$$
{}^{x}_{x}M^{a_1}_{a_2}(j_1, j_2) = \max \begin{cases}
{}^{x}_{x}M^{a_1}_{a_2}(j_1 - 1, j_2 - 1) + \sigma\left(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\right) \\
{}^{x}_{x}M^{a_1}_{a_2}(j_1 - 1, j_2) + \gamma \\
{}^{x}_{x}M^{a_1}_{a_2}(j_1, j_2 - 1) + \gamma \\
\max\limits_{\substack{a_1^l < k_1 < j_1 \\ a_2^l < k_2 < j_2}} \left\{ {}^{x}_{x}M^{a_1}_{a_2}(k_1 - 1, k_2 - 1) + D\big((k_1, j_1), (k_2, j_2)\big) \right\} \\
\max\limits_{a_1^l \le k_1 < j_1} \left\{ {}^{o}_{x}M^{a_1}_{a_2}(k_1, j_2) \right\} \\
\max\limits_{a_2^l \le k_2 < j_2} \left\{ {}^{x}_{o}M^{a_1}_{a_2}(j_1, k_2) \right\}
\end{cases} \tag{3.7}
$$

Dadurch besteht bei den Rekursionsabbrüchen auch die Möglichkeit einer Exklusion zu Beginn beider Teilsequenzen.

$$
\begin{aligned}
{}^{x}_{x}M^{a_1}_{a_2}(a_1^l, a_2^l) &= 0 \\
{}^{x}_{x}M^{a_1}_{a_2}(k_1, a_2^l) &= 0, \text{ für alle } k_1 \text{ mit } a_1^l < k_1 < a_1^r \\
{}^{x}_{x}M^{a_1}_{a_2}(a_1^l, k_2) &= 0, \text{ für alle } k_2 \text{ mit } a_2^l < k_2 < a_2^r
\end{aligned}
$$

Mit Hilfe der Rekursionsgleichungen 3.3 bis 3.7 können nun alle lokalen Alignments berechnet werden, deren Enden aus zwei gematchten Basenpaaren bestehen. Damit fehlt für ein nach Definition 12 lokales Alignment zweier Sequenzen S_{R_1} und S_{R_2} noch die Möglichkeit, Anfangs- und Endbereiche der Sequenzen auszulassen.

Dies entspricht der Berechung des besten globalen Alignments über allen Teilsequenzen von S_{R_1} und S_{R_2} und damit einer sequenziellen Lokalität, wie sie vom Smith-Waterman-Algorithmus [SW81] verwendet wird. Deshalb erweitere ich in Analogie zu diesem Ansatz die Rekusionsgleichung 3.1 für die Berechnung eines globalen Alignments um die zusätzliche Möglichkeit, ein neues Alignment mit dem Score 0 zu beginnen, falls alle anderen Möglichkeiten einen negativen Score liefern. Damit lässt sich der maximale Score $T(j_1, j_2)$ der optimalen paarweisen globalen Alignments aller Teilsequenzen $S_{R_1}[\,i_1, j_1\,]$ und $S_{R_2}[\,i_2, j_2\,]$ mit $1 \le i_1 \le j_1$ und $1 \le i_2 \le j_2$ wie folgt berechnen:

$$
T(j_1, j_2) = \max \begin{cases}
0 \\
T(j_1 - 1, j_2 - 1) + \sigma\big(S_{R_1}[\,j_1\,], S_{R_2}[\,j_2\,]\big) \\
T(j_1 - 1, j_2) + \gamma \\
T(j_1, j_2 - 1) + \gamma \\
\max\limits_{\substack{0 < k_1 < j_1 \\ 0 < k_2 < j_2}} \left\{ T(k_1 - 1, k_2 - 1) + D\big((k_1, j_1), (k_2, j_2)\big) \right\}
\end{cases} \tag{3.8}
$$

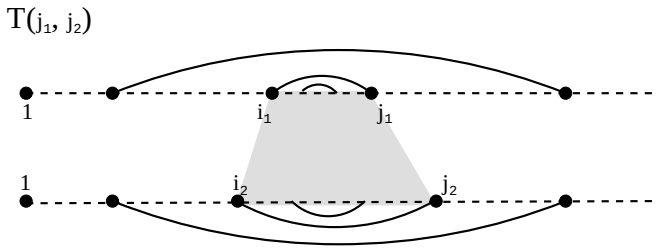

Abbildung 3.4: Beispiele für die Bedeutung der Rekusionsgleichung T. Der grau hinterlegte Bereich entspricht dabei dem optimalen lokalen Alignment mit Score $T(j_1, j_2)$.

Dabei gelten für T folgende Rekursionsabbrüche:

$$\begin{aligned} T(0,0) &= 0 \\ T(k_1,0) &= 0, \text{ für alle } k_1 \text{ mit } 0 < k_1 < j_1 \\ T(0,k_2) &= 0, \text{ für alle } k_2 \text{ mit } 0 < k_2 < j_2 \end{aligned}$$

Das optimale lokale Alignment zweier Sequenzen S_{R_1} und S_{R_2} ergibt sich dann aus der Berechnung von $T(|S_{R_1}|, |S_{R_2}|)$. Das Ende des Alignments wird dabei durch die Position mit dem maximalen Eintrag in T festgelgt und das Alignment selbst ergibt sich mittels Traceback bis ein Eintrag 0 erreicht ist. Abbildung 3.4 illustriert die Situation noch einmal.

Damit sind nun alle notwendigen Rekursionsgleichungen für die Lösung des in Definition 14 aufgestellten paarweisen lokalen Alignment-Problems für zwei RNA-Sequenzen S_{R_1} und S_{R_2} definiert. Als nächstes folgt der eigentliche Algorithmus.

Der Algorithmus

Die Eingabe für den paarweisen lokalen Alignmentalgorithmus besteht aus zwei Sequenzen S_{R_1} und S_{R_2}, sowie zwei Matrizen Ω_g mit $g \in \{1, 2\}$. Diese enthalten alle möglichen Basenpaare a_g, die sich nach RNAfold über S_{R_g} ausbilden könnten zusammen mit ihren Wahrscheinlichkeiten $p_{S_{R_g}}(a_g)$.

Der Score des optimalen lokalen Alignments von S_{R_1} und S_{R_2} (und damit das Alignment selbst) ergibt sich aus dem maximalen Eintrag in T. Die Einträge in T wiederrum hängen von den Alignmentscores aller Alignments, deren Enden zwei gematchte Basenpaare bilden, ab. Aus diesem Grund berechnet der Algorithmus zuerst für alle Paare von Basenpaaren $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ mit $p_{S_{R_1}}(a_1) \geq p_{min}$ und $p_{S_{R_2}}(a_2) \geq p_{min}$ den Score $D(a_1, a_2)$.

Dabei ist p_{min} eine Grenzwert, auf dessen Bedeutung ich bei der Komplexitätsanalyse im nächsten Abschnitt eingehe. Bis dahin gehe ich von $p_{min} = 0$ aus, wodurch $D(a_1, a_2)$ für alle Paare von Basenpaaren $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ bestimmt wird.

Da bei dieser Berechnung über die Gleichungen $M_{a_2}^{a_1}$ auf Einträge $D(b_1, b_2)$ mit $a_1^l < b_1^l < b_1^r < a_1^r$ und $a_2^l < b_2^l < b_2^r < a_2^r$ zurückgegriffen wird, sofern solche b_1 und b_2 in Ω_1 bzw. Ω_2 existieren, ist die Reihenfolge der Berechnungen entscheident.

Diese geschieht am effizientesten, indem man mit den an weitesten rechts beginnenden Basenpaaren anfängt. Da diese in genesteten Strukturen am weitesten innen liegen, lassen sich deren Scores für die Berechnung folgender Einträge verwenden.

Sollten dabei mehrere Basenpaare an der gleichen Position beginnen, braucht ${}_x^x M_{a_2}^{a_1}(a_1^r - 1, a_2^r - 1)$ nur für die beiden am weitesten rechts endenden Basenpaare

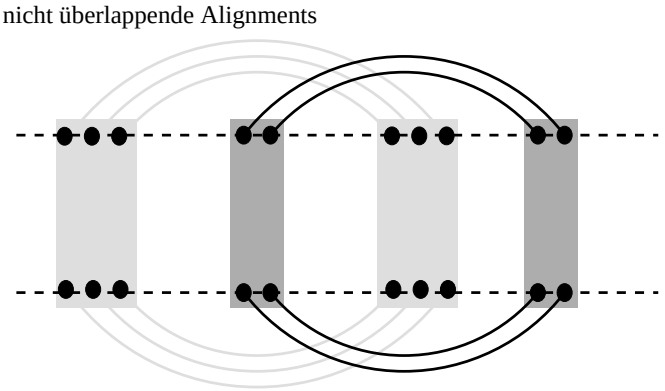

Abbildung 3.5: Beispiel für nicht überlappend Alignments. Der hellgraue Bereich stellt das optimale Alignment dar, wärend der dunkel bereich das nächst-beste Alignment zeigt.

a_1 und a_2 berechnet werden. Die Scores der restlichen Basenpaare mit rechten Enden b_1^r bzw. b_2^r sind dann einfach an den entsprechenden Positionen, also in ${}_x^x M_{a_2}^{a_1}(b_1^r - 1, b_2^r - 1)$ abzulesen. Da die bei der Berechnung von ${}_x^x M_{a_2}^{a_1}(a_1^r - 1, a_2^r - 1)$ erhaltenen Zwischenergebnisse nicht mehr für andere Berechnungen verwendet werden können, werden sie auch nicht gespeichert.

Nachdem die Einträge in D berechnet wurden, kann $T(|S_{R_1}|, |S_{R_2}|)$ bestimmt werden. Der Score s_{pla} des besten lokalen Alignments der Sequenzen S_{R_1} und S_{R_2} ergibt sich dann aus den maximalen Eintrag in T. Das Alignment und damit die Lösung des paarweisen lokalen Alignmentproblems (Definition 14) kann nun einfach mittels Backtracking bestimmt werden. Dazu beginnt man bei dem Eintrag in T welcher s_{pla} enthält und verfolgt die Operationen zurück, bis ein Eintrag 0 in T erreicht ist. Die Art und die Reihenfolge der Operationen bestimmen dann eindeutig das Alignment. Den kompletten Algorithmus habe ich noch einmal auf der folgenden Seite zusammengefasst.

Wie in der Beschreibung des Verfahrens begründet, berechne ich jedoch neben dem besten lokalen Alignment noch die nächstbesten Alignments, wobei ich mich für zwei Ansätze entschieden habe.

Die k-besten Alignments

Der erste Ansatz für die Bestimmung der k-besten Alignmnets berechnet die k lokalen paarweisen Alignments der Sequenzen S_{R_1} und S_{R_2} mit den höchsten Score, welche sich nicht überlappen. Dies entspricht damit dem Ansatz, wie er von Programmen wie BLAST [AGM90] bei sequenzieller Lokalität verwendet wird. Allerdings ist die Berechnung bei struktureller Lokalität wesentlich anspruchsvoller.

Während sich die besten nicht-überlappenden Alignments bei der Verwendung von sequenzieller Lokalität einfach mittels Traceback aus den einmal berechneten Rekusionsgleichungen ableiten lassen, ist das bei struktureller Lokalität nicht möglich. Die Ursache dafür liegt in den Exklusionen. Da diese selbst wieder konservierte Motive enthalten können, müssen sie bei der Suche nach weiteren lokalen Alignments auch mit betrachtet werden. Abbildung 3.5 zeigt ein Beispiel für diese Situation.

Da sich die Scores in diesen Bereichen aber aus den davorliegenden Bereichen ergeben, würden diese schon vorher alignierten Bereiche beim Traceback wieder verwendet werden. Deshalb ist für jedes neue Alignment über S_{R_1} und S_{R_2} auch eine neue Berechnung der Rekursionsgleichungen nötig, wobei dann das alignieren

Paarweiser lokaler Alignmentalgorithmus
Seien S_{R_1} und S_{R_2} zwei RNA-Sequenzen, Ω_1 und Ω_2 die Matrizen der Basenpaare über S_{R_1} bzw. S_{R_2} zusammen mit ihren Wahrscheinlichkeiten und p_{min} ein Grenzwert für die Basenpaarwahrscheinlichkeiten.
Weiterhin sei $\alpha_g(i)$ mit $g \in \{1,2\}$ und $i \in \{1,\ldots,|S_{R_g}|\}$ die Liste aller $a_g \in \Omega_g$ mit $a_g^l = i$ in der Reihenfolge der rechten Basenpaarenden a_g^r, $|\alpha_g[\,i\,]|$ die Anzahl der Basenpaare in $\alpha_g[\,i\,]$ und $\alpha_g[\,i\,][\,j\,]$ das j-te Basenpaar in $\alpha_g[\,i\,]$.
Dann ergibt sich das optimale paarweise lokale Alignment von S_{R_1} und S_{R_2} wie folgt:

for $i_1 := \big|S_{R_1}\big|$ *downto* 1 *do*
  if $\big|\alpha_1[\,i_1\,]\big| > 0$
    for $i_2 := \big|S_{R_2}\big|$ *downto* 1 *do*
      if $\big|\alpha_2[\,i_2\,]\big| > 0$ {
        $a_1 := \alpha_1\Big[\,i_1\,\Big]\Big[\,\Big|\alpha_1[\,i_1\,]\Big|\,\Big]$
        $a_2 := \alpha_2\Big[\,i_2\,\Big]\Big[\,\Big|\alpha_2[\,i_2\,]\Big|\,\Big]$
        if $p_{S_{R_1}}(a_1) \geq p_{min} \;\wedge\; p_{S_{R_2}}(a_2) \geq p_{min}$ {
          Berechne ${}_x^x M_{a_2}^{a_1}(a_1^r - 1, a_2^r - 1)$;
          for $j_1 := \big|\alpha_1[\,i_1\,]\big|$ *downto* 1 *do*
            for $j_2 := \big|\alpha_2[\,i_2\,]\big|$ *downto* 1 *do* {
              $b_1 := \alpha_1\Big[\,i_1\,\Big]\Big[\,\Big|\alpha_1[\,j_1\,]\Big|\,\Big]$
              $b_2 := \alpha_2\Big[\,i_2\,\Big]\Big[\,\Big|\alpha_2[\,j_2\,]\Big|\,\Big]$
              $D(b_1,b_2) := {}_x^x M_{a_2}^{a_1}(b_1^r - 1, b_2^r - 1) + \rho_{S_{R_1}}(b_1) + \rho_{S_{R_2}}(b_2)$
                $\quad + \tau\big(S_{R_1}[\,b_1^l\,], S_{R_1}[\,b_1^r\,], S_{R_2}[\,b_2^l\,], S_{R_2}[\,b_2^r\,]\big)$;
            }
        }
      }
Berechne $T(|S_{R_1}|, |S_{R_2}|)$;
$s_{pla} := 0$;
$l_1 := 0$;
$l_2 := 0$;
for $j_1 := |S_{R_1}|$ *downto* 1 *do*
  for $j_2 := |S_{R_2}|$ *downto* 1 *do*
    if $T(j_1, j_2) > s_{pla}$ {
      $s_{pla} := T(j_1, j_2)$;
      $l_1 := j_1$;
      $l_2 := j_2$;
    }
Berechne Traceback ab $T(l_1, l_2)$ *bis Eintrag mit* 0 *erreicht ist*;

schon einmal alignierter Bereiche verboten wird. Zu diesem Zweck speichere ich alle schon alignierten Positionen in einer Liste und überprüfe in jedem Alignmentschritt ob die gerade betrachteten Positionen darin enthalten sind.

Der zweite Ansatz verzichtet auf die Einschränkung auf nicht-überlappene Alignments und berechnet damit einfach nur die k besten Alignments. Um diese zu bestimmen ändere ich die Rekusionsfälle dahingehend ab, dass für jedes gematchte Positionspaar (i_1, i_2) zusätzlich eine Betrag $\psi \cdot \upsilon(i_1, i_2)$ addiert wird.

Dabei entspricht υ einer Funktion, welche für zwei Positionen die Anzahl angibt, wie oft diese beiden Positionen schon gematcht waren und ψ ein Strafscore für das wiederholte Verwenden gleicher Alignmentkanten zwischen zwei Basen. Da die Strafscores nur zu alternativen Motiven und nicht zu einem schlechteren Alignmentscore führen sollen, addiere ich alle angewanten Strafscores auf und korregiere nach der Berechnung des Alignments dessen Score mit Hilfe dieser Summe.

Auch bei diesem Ansatz muss der paarweise lokale Alignment-Algorithmus k-mal ausgeführt werden. Die so berechneten Alignment liefern dann entweder alternative konservierte Bereiche oder geben einen Hinweis auf besonders stabile Motive.

Als nächstes leitet MuLoRA mittels T-Coffee aus den paarweisen Alignments ein mutiples Alignment ab.

3.2.3 Multiples Alignment

Für die Berechnung des multiplen Alignments benötigt T-Coffee erst einmal die Bibliothek aller Alignmentkanten der paarweisen Alignments über den Sequenzen.

Kantengewichte

Wie im ersten Abschnitt dieses Kapitels schon erwähnt, sollen die Kantengewichte ihrer Zuverlässigkeit entsprechen, weshalb ich den Score der entsprechenden Alignments für die Gewichtung der Kanten verwende. Dieser wird dabei auf alle Alignmentkanten zwischen zwei Basen aufgeteilt. Kanten, die ein Gap enthalten bekommen hingegen ein Gewicht von 0, da sie keine Informationen für die Berechnung des multiplen Alignments liefern. Falls dabei eine Kante durch die Berechnung überlappender Alignments mehrfach vorkommen sollte, werden die einzelnen Gewichte addiert.

Damit hängt das Gewicht ζ einer Alignmentkante (i_1, i_2) in den k besten lokalen Alignments zweier Sequenzen $S_{R_{g_1}}$ und $S_{R_{g_2}}$ von dem Score $s(A_{g_1,g_2,k})$ des k-ten Alignments, und der Menge $B(A_{g_1,g_2,k})$ aller Kanten in $A_{g_1,g_2,k}$ zwischen zwei Basen ab. Insgesamt ergibt sich so folgende Funktion zur Berechnung der Gewichte:

$$\zeta_{g_1,g_2}(i_1, i_2) = \begin{cases} \sum\limits_k \left(\frac{s(A_{g_1,g_2,k})}{|B(A_{g_1,g_2,k})|} \right) & \text{falls } (i_1, i_2) \in B(A_{g_1,g_2,k}) \\ 0 & \text{sonst.} \end{cases} \tag{3.9}$$

T-Coffee

Progressive Alignmentansätze leiten unter dem Problem, dass Fehler in den ersten Alignmentsschritten beim Einfügen der restlichen Sequenzen nicht mehr verbessert werden können. Um dieses Problem zu minimieren, beginnt T-Coffee mit denjenigen Alignmentsschritten, welche von den meisten anderen paarweisen Alignments bestätigt werden.

Dazu erweitert das Programm in einem Vorverarbeitungsschritt die primäre Bibliothek. Das Ziel dabei ist, die einzelnen Kantengewichte so zu kombinieren, dass die neuen Gewichte die Informationen aus der gesamten Bibliothek reflektieren. Um dies zu erreichen, verwenden Notredame *et al.* einen Triplet-Ansatz.

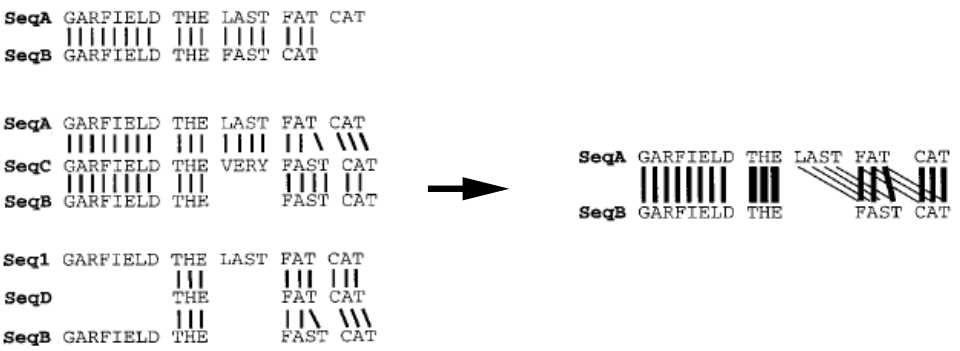

Abbildung 3.6: Beispiel für die Erweiterung der Eingabe-Bibliothek (aus [NHH00]).

Bei diesen wird für jedes mögliche Paar $S_{R_{g_1}}$ und $S_{R_{g_2}}$ mit $g_1 \neq g_2$ über den Eingabesequenzen S_{R_1} bis S_{R_m} das Alignment A_{g_1,g_2} zwischen $S_{R_{g_1}}$ und $S_{R_{g_2}}$ mit allen Alignments zwischen $S_{R_{g_1}}$ und $S_{R_{g_2}}$ über eine dritte Sequenz S_{R_h}, verglichen (siehe Abbildung 3.6).

Bei jedem Vergleich werden dabei die Kanten in A_{g_1,g_2} mit allen Kanten in $A_{g_1,h}$ und allen Kanten in A_{h,g_2} verglichen. Sollten dabei für eine Kante (i_1, i_2) in A_{g_1,g_2} zwei Kanten (i_1, j) und (j, i_2) in $A_{g_1,h}$ bzw. A_{h,g_2} existieren, wird (i_1, i_2) von den beiden anderen Kanten unterstützt und erhält deshalb zusätzlich den niedrigeren der beiden Scores von (i_1, j) bzw. (j, i_2).

Im Endeffekt erhält so jede Kante (i_1, i_2) in A_{g_1,g_2} ein Gewicht $\xi_{g_1,g_2}(i_1, i_2)$, welches wie folgt berechnet wird:

$$\xi_{g_1,g_2}(i_1, i_2) = \zeta_{g_1,g_2}(i_1, i_2) + \sum_{\substack{1 \leq h \leq m \\ h \neq g_1, h \neq g_2}} \min_j \left\{ \zeta_{g_1,h}(i_1, j),\ \zeta_{h,g_2}(j, i_2) \right\} \tag{3.10}$$

Auf Grundlage der so erweiterten Bibliothek berechnet T-Coffee eine Distanzmatrix und erstellt daraus mittels Neighbor-Joining [SN87] ein phylogenetischer Baum. Dieser bestimmt dann die Reihenfolge der einzelnen Alignmentschritte.

Die beiden Sequenzen mit dem geringsten Abstand werden zuerst aligniert. Das entstandene Alignment ist dann wie bei herkömmlichen progressiven Ansätzen fixiert und kann nicht mehr verändert werden. Danach werden in Abhängigkeit des phylogenetischen Baums entweder die beiden Sequenzen mit dem nächst niedrigsten Abstand aligniert oder eine Sequenz wird zum ersten Alignment hinzugefügt. Dies geht so weiter, bis alle Sequenzen in einem Alignment vereint sind, wobei ab den dritten Schritt auch zwei Alignments aligniert werden können.

Für die Berechnung der einzelnen Alignmentschritte nutzt T-Coffee eine effiziente Version des Needleman-Wunsch Algorithmus [NW70] von Gotoh [Got82]. Dieser verwendet dabei nur die Kantengewichte aus der erweiterten Bibliothek, wobei für die Berechnung eines Alignments, an dem ein bereits zuvor berechnetes Alignment beteilgt ist, die Durchschnittswerte der Spalten verwendet werden. Da in den Kantengewichten bereits die Substitutionsscores, Insertions- und Deletionskosten sowie die Strukturbeiträge enthalten sind, werden keine Parameter dieser Art benötigt.

Auf diese Weise berechnet T-Coffee schließlich ein globales Alignment A^m aller Eingabesequenzen S_{R_1} bis S_{R_m}. Um nun die Lokalität von A^m zu beschreiben, folgt als nächstes die Berechnung eines Konserviertheitsfaktors für jede Spalte von A^m.

Konserviertheitsfaktor

Die Konserviertheit einer Spalte hängt von der Ähnlichkeit der Motive ab, in welchen die in der Spalte alignierten Positionen enthalten sind. Je höher diese Ähnlich-

keit ist, um so konservierter ist auch die Spalte. Ähnliche Motive wiederrum werden von den paarweisen lokalen Alignments aligniert, wodurch für diese Bereiche viele Alignmentkanten existieren.

Aus diesem Grund verwende ich die Anzahl der Kanten, die eine Spalte bestätigen, als Maß für deren Konserviertheit. Da jedoch die maximale Anzahl der Kanten, die eine Spalte bestätigen können, von der Anzahl der Sequenzen im multiplen Alignment abhängt, normiere ich die Anzahl der Kanten mit der maximal möglichen Anzahl.

Für die Berechnung der Konserviertheitsfaktoren bestimme ich zuerst die Menge $K(i)$ der Kanten, welche die i-te Spalte bestätigen:

$$K(i) = \big\{(i_1, i_2) \big| i_1 < i_2 \wedge \exists g_1 \exists g_2 \; A^m[\,g_1, i\,] = i_1 \wedge A^m[\,g_2, i\,] = i_2 \wedge (i_1, i_2) \in A_{g_1,g_2}\big\}$$

Dann ergibt sich der Konserviertheitsfaktor $\kappa(i)$ der i-ten Spalte wie folgt:

$$\kappa(i) = \frac{\big|K(i)\big|}{m \cdot (m-1)/\,2} \tag{3.11}$$

Damit ist es nun möglich, durch die Angabe eines Grenzwertes κ_{min} für die Konserviertheit, lokale Bereiche in einem multiplen Alignment zu definieren. Die strukturelle Zusammenhänge dieser Bereiche werden aber erst durch die Berechnung der Konsensusstruktur erkennbar. Abbildung 3.7 zeigt beispielsweise ein multiples Alignment mit Konsensussequenz und -Struktur von sechs Hammerhead-Ribozymen des Typs III. Das gemeinsamme strukturelle Motiv besteht aus allen Spalten mit einen Konserviertheitsfaktor größer 0.6.

3.2.4 Konsensussequenz und -Struktur

Bei der Berechnung der Konsensussequenz und -Struktur teile ich die einzelnen Spalten des multiplen Alignments A^m zuerst bezüglich ihres Konserviertheitsfaktors in konservierte und unkonservierte Spalten ein. Dabei gilt eine Spalte i genau dann als konserviert, wenn ihr Konserviertheitsfaktor $\kappa(i)$ größer oder gleich eines Grenzwerts κ_{min} für die Konserviertheit ist.

Unkonservierte Spalten werden also von zu wenigen Kanten unterstützt und wurden daher unbegründet von T-Coffee gebildet. Damit die Buchstaben in diesen Spalten keinen Einfluß auf die Konsensussequenz nehmen können, erhalten unkonservierte Spalten ein Gap '$-$' als Konsensusbuchstaben.

Für alle konservierten Spalten i mit $1 \leq i \leq |A^m|$ und $\kappa(i) \geq \kappa_{min}$ berechne ich hingegen denjenigen Buchstaben $\omega_i \in \Sigma_C$ mit $\Sigma_C = \{\Sigma_N \cup \{-\}\}$, welcher das in Definition 15 aufgestellte Konsensussequenz-Problem löst. Dazu bestimme ich einfach für alle $\omega \in \Sigma_C$ die Ähnlichkeit $\nu\big(\omega, A^m[\,i\,]\big)$ von ω zu $A^m[\,i\,]$:

$$\nu\big(\omega, A^m[\,i\,]\big) = \sum_{1 \leq g \leq m} \delta(\omega, A^m[\,g\,][\,i\,]).$$

Dabei ist δ eine Substitutionsfunktion über $\Sigma_C \times \Sigma_C$ und wie folgt definiert:

$$\delta(\omega, A^m[\,g\,][\,i\,]) = \begin{cases} 1 & \text{falls} \quad \omega = A^m[\,g\,][\,i\,] = - \;\; \vee \;\; \omega = S_{R_g}[\,A^m[\,g\,][\,i\,]\,] \\ 0 & \text{sonst.} \end{cases}$$

Demnach entspricht $\nu\big(\omega, A^m[\,i\,]\big)$ der Häufigkeit von ω in der i-ten Spalte. Der häufigste Buchstabe entspricht dann dem Konsensusbuchstaben. Allerdings muss dazu dessen Häufigkeit einen Grenzwert ν_{min} erreichen. Sollte dem nicht so sein, ist die Häufigkeit nicht hoch genug, um ω_i deutlich von den restlichen Buchstaben in Σ_C abzusetzen und die Spalte erhält den Konsensusbuhstaben 'N'.

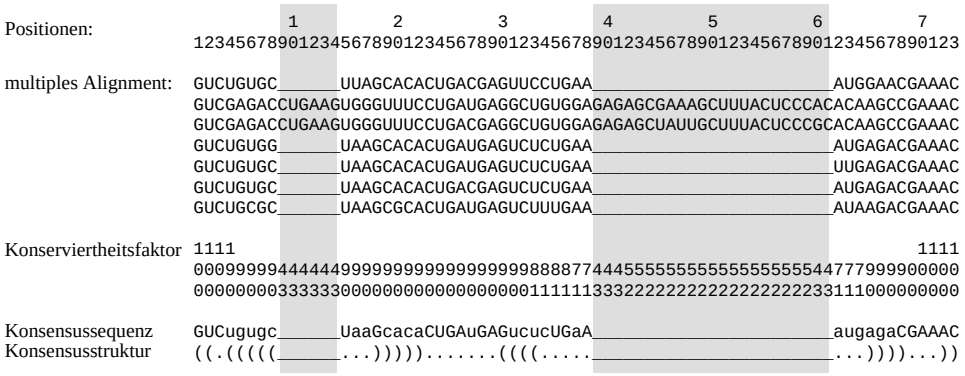

Abbildung 3.7: Beispiel für die Konsensussequenz und Struktur eines multiple Alignments von sechs Hammerhead-Ribozymen des Typs III. Die Konserviertheitsfaktoren sind dabei in Prozent angegeben. Die grau hinterlegten Bereiche kennzeichnen alle Kanten mit einen Konserviertheitsfaktor kleiner als 0.6.

Insgesamt folgt daraus folgende Gleichung zur Berechnung des i-ten Konsensusbuchstabens über einen multiplen Alignment A^m:

$$S_C[\,i\,] = \begin{cases} \arg\max\limits_{\omega \in \Sigma_C} \left\{ \nu\big(\omega, A^m[\,i\,]\big) \right\} & \text{falls} \quad \begin{array}{c} \kappa(i) \geq \kappa_{min} \ \wedge \\ \nu\big(\omega, A^m[\,i\,]\big) \geq \nu_{min} \end{array} \\ N & \text{falls} \quad \begin{array}{c} \kappa(i) \geq \kappa_{min} \ \wedge \\ \nu\big(\omega, A^m[\,i\,]\big) < \nu_{min} \end{array} \\ - & \text{sonst} \end{cases} \tag{3.12}$$

Die Konsensussequenz S_C folgt dann einfach aus den aneinandergereihten Konsensusbuchstaben.

Damit wende ich mich nun dem letzten Punkt des MuLoRA-Ansatzes zu, einer Lösung des in Definition 16 aufgestellten Konsensusstruktur-Problems. Da diese aus der genesteden Strukturmenge P_C über der Konsensussequenz S_C besteht, welche die maximale Summe aller Basenpaargewichte der in ihr enthaltenen Basenpaare besitzt, entspricht die Problemstellung einer Basenpaarmaximierung mit spezifischen Gewichten.

Das Gewicht $\vartheta_{S_C}\big((i,j)\big)$ eines Basenpaares (i,j) über S_C hängt dabei in Analogie zu den Basenpaargewichten der Eingabesequenzen von der Wahrscheinlichkeit $p_{S_C}\big((i,j)\big)$, mit der sich (i,j) über S_C ausbildet und der kleinsten noch signifikanten Wahrscheinlichkeit p_{sig} ab.

$$\vartheta_{S_C}\big((i,j)\big) = log \frac{p_{S_C}\big((i,j)\big)}{p_{sig}}$$

Die Basenpaarewahrscheinlichkeiten p_{S_C} der Konsensussequenz ergeben sich dabei aus dem arithmetischen Mittel aller Basenpaarwahrscheinlichkeiten der Eingabesequenze S_{R_1} bis S_{R_m}.

$$p_{S_C}\big((i,j)\big) = \frac{\sum\limits_{1 \leq g \leq m} \Big(p_{S_{R_g}}\big((A^m[\,g\,][\,i\,], A^m[\,g\,][\,j\,])\big)\Big)}{m}$$

Um nun die genestede Struktur über S_C mit der maximalen Summe aller Basenpaargewichte zu berechnen, verwende ich den Nussinov-Algorithmus [NPG78]. Allerdings verbiete ich dabei in Analogie zur Konsensussequenz das Ausbilden von

Basenpaaren über nicht konservierten Positionen. Insgesamt ergibt sich so folgende Rekursionsgleichung zur Lösung des Konsensusstruktur-Problems:

$$N(i,j) = \max \begin{cases} \begin{cases} N(i+1,j-1) + \vartheta_{S_C}\big((i,j)\big) & \text{falls } \begin{matrix}\kappa(i) \geq \kappa_{min} \wedge \\ \kappa(j) \geq \kappa_{min}\end{matrix} \\ -\infty & \text{sonst} \end{cases} \\ N(i+1,j) \\ N(i,j-1) \\ \max\limits_{i<k<j} \Big\{ N(i,k) + N(k+1,j) \Big\} \end{cases} \tag{3.13}$$

Dabei gelten für N folgende Rekursionsabbrüche:

$$\begin{aligned} N(i,i) &= 0, \text{ für alle } i \text{ mit } 1 \leq i \leq n \\ N(i,i-1) &= 0, \text{ für alle } i \text{ mit } 2 \leq i \leq n \end{aligned}$$

Aus den Zwischenergebnissen der Berechnung von $N(1,|A^m|)$ lässt sich dann mittels Backtracking die Konsensustruktur bestimmen. Abbildung 3.7 zeigt ein Beispiel für die Konsensussequenz und -Struktur eines multiplen Alignments.

Damit wären nun alle Bestandteil des MuLoRA-Ansatzes geklärt. Als nächstes folgen die Kosten.

3.3 Komplexitätsbetrachtungen

In diesem Abschnitt untersuche ich die Zeit- und Speicherkomplexität des MuLoRA-Ansatzes. Dabei gehe ich zuerst auf die von mir entwickelten Algorithmen ein und gebe anschließend einen Gesamtüberblick.

3.3.1 Paarweiser lokaler Alignmentalgorithmus

Der Algorithmus zur Berechnung eines lokalen paarweisen Alignments erhält als Eingabe zwei Sequenzen S_{R_1} und S_{R_2} und die Matrizen Ω_1 und Ω_2 aller möglichen Basenpaare mit ihren Wahrscheinlichkeiten, die sich nach RNAfold über S_{R_1} bzw. S_{R_2} ausbilden könnten.

Für die Komplexitätsanalyse gehe ich dabei von Sequenzlängen n_1 für S_{R_1} bzw. n_2 für S_{R_2} aus.

Analyse

Der Algorithmus berechnet zuerst für alle möglichen Paare von Basenpaaren $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ mit $p_{S_{R_1}}(a_1) \geq p_{min}$ und $p_{S_{R_2}}(a_2) \geq p_{min}$ den Score $D(a_1,a_2)$. Da sich die Kantengewichte ρ und der Substitutionsscore τ in konstanter Zeit bestimmen lassen, hängt der Zeitbedarf für die Berechnung eines einzelnen Eintrages $D(a_1,a_2)$ nur von dem Zeitbedarf für die Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r-1,a_2^r-1)$ ab.

Für die Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r-1,a_2^r-1)$ wiederrum müssen für alle j_1 mit $a_1^l \leq j_1 < a_1^r - 1$ und für alle j_2 mit $a_2^l \leq j_2 < a_2^r - 1$ die Werte in ${}^{o}_{o}M^{a_1}_{a_2}(j_1,j_2)$, ${}^{x}_{o}M^{a_1}_{a_2}(j_1,j_2)$, ${}^{o}_{x}M^{a_1}_{a_2}(j_1,j_2)$ und ${}^{x}_{x}M^{a_1}_{a_2}(j_1,j_2)$ berechnet werden.

Da bei jeder dieser Berechnungen auf bereits vorher berechnete Ergebnisse zurückgegriffen wird und sich der Substitutionscore σ in konstanter Zeit bestimmen läßt, hängt der Zeitbedarf für jede einzelne Berechnung nur von der Anzahl der

Möglichkeiten für das Einfügen eines zuvor berechneten Alignments und der Anzahl der Möglichkeiten für das Durchführen einer Exklusionen ab.

- Die Anzahl der Möglichkeiten für das Einfügen eines zuvor berechneten Alignments ergibt sich aus der Anzahl aller Paare von Basenpaaren $b_1 \in \Omega_1$ und $b_2 \in \Omega_2$, die in den Positionen j_1 bzw. j_2 enden. Da über einer Sequenz S mit einer Länge von n nur maximal n verschiedene Basenpaare existieren können, die an der gleichen Position enden, läßt sich diese Anzahl mit $O(n_1 \cdot n_2)$ abschätzen.

- Die Anzahl der Möglichkeiten für das Durchführen einer Exklusion in Sequenz S_{R_1} beträgt entweder $j_1 - a_1^l$ oder 0, falls in der Rekusionsgleichung keine Exklusionen in S_{R_1} erlaubt sind. Analog dazu beträgt die Anzahl der Möglichkeiten für das Durchführen einer Exklusion in Sequenz S_{R_2} entweder $j_2 - a_1^2$ oder 0. Damit exisitieren im Allgemeinen Fall $O(n_1+n_2)$ Möglichkeiten, eine Exklusion durchzuführen.

Damit liegt der Zeitbedarf für jede einzelne Berechnung in $O(n_1 \cdot n_2)$. Da die Anzahl aller Berechnungen $\left((a_1^r - 1 - a_1^l) \cdot (a_2^r - 1 - a_1^2)\right)$ beträgt und deshalb in $O(n_1 \cdot n_2)$ liegt, ergibt sich für die Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r - 1, a_2^r - 1)$ ein Gesamtaufwand von $O({n_1}^2 \cdot {n_2}^2)$ Operationen.

Der Speicheraufwand für die Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r - 1, a_2^r - 1)$ beträgt dabei für jede der vier Rekusionsgleichungen $\left((a_1^r - 1 - a_1^l) \cdot (a_2^r - 1 - a_1^2)\right)$ und lässt sich so mit $O(n_1 \cdot n_2)$ abschätzen. Da die erhaltenen Zwischenergebnisse nicht mehr für andere Berechnungen gebraucht und deswegen auch nicht gespeichert werden, kann der Speicher aber bei jeder neuen Berechnung wieder verwendet werden.

Bei der Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r - 1, a_2^r - 1)$ für zwei Basenpaare $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ erhält man als Nebenprodukt auch gleichzeitig für alle möglichen Paare von Basenpaaren $b_1 \in \Omega_1$ und $b_2 \in \Omega_2$ mit $b_1^l = a_1^l$ und $b_2^l = a_2^l$ das Ergebnis einer Berechnung von ${}^{x}_{x}M^{b_1}_{b_2}(b_1^r - 1, b_2^r - 1)$.

Deshalb sind für die Bestimmung aller Einträge in D nur so viele Berechnungen nötig, wie es verschiedene Paare von Basenpaaranfängen a_1^l und a_2^l mit $a_1 \in \Omega_1$ bzw. $a_2 \in \Omega_2$ gibt. Da über einer Sequenz S mit einer Länge von n auch nur maximal n verschiedene Basenpaare existieren können, die an unterschiedlichen Position beginnen, läßt sich diese Anzahl ebenfalls mit $O(n_1 \cdot n_2)$ abschätzen.

Insgesamt liegt der Zeitbedarf für die Bestimmung aller Einträge in D somit in $O({n_1}^3 \cdot {n_2}^3)$.

Die Anzahl der Einträge wird dabei von der Anzahl aller Paare von Basenpaaren $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ bestimmt. Da über einer Sequenz S mit einer Länge von n nur maximal n^2 verschiedene Basenpaare existieren, liegt der Speicherbedarf von D in $O({n_1}^2 \cdot {n_2}^2)$.

Als nächstes berechnet der Algorithmus $T(|S_{R_1}|, |S_{R_2}|)$. Da der Rekursionsfall für den Beginn eines neuen lokalen Alignments in konstanter Zeit ausgewertet wird und alle restlichen Rekusionsfälle mit denen von ${}^{o}_{o}M$ identisch sind, entsprechen die Kosten denjenigen einer Berechnung von ${}^{o}_{o}M^{a_1}_{a_2}(|S_{R_1}|, |S_{R_2}|)$ mit $a_1 = (0, |S_{R_1}| + 1)$ und $a_2 = (0, |S_{R_2}| + 1)$. Somit liegt der Zeitbedarf in $O({n_1}^2 \cdot {n_2}^2)$ und der Speicherbedarf in $O(n_1 \cdot n_2)$.

Die letzte Aufgabe des Algorithmus besteht darin, die Position mit dem maximalen Eintrag in T zu finden und das Alignment mittels Backtracking ab dieser Position zu bestimmen.

Bei der Suche nach den maximalen Eintrag wird jeder der $n_1 \cdot n_2$ Einträge untersucht, wodurch der Zeitaufwand in $O(n_1 \cdot n_2)$ liegt.

Für die Berechnung des Alignments sind $O(n_1 + n_2)$ Backtracking-Schritte notwendig. Da diese durch alle Rekursionsgleichungen führen können, ergibt sich der maximale Aufwand eines einzelnen Schrittes aus dem maximalen Aufwand über alle möglichen Rekursionsfälle. Dieser beträgt dabei $O(n_1 \cdot n_2)$ für die Suche nach einem eingefügten Alignment, wodurch sich ein Gesamtzeitbedarf von $O((n_1+n_2)\cdot n_1 \cdot n_2)$ ergibt.

Allerdings müssen für jedes eingefügte Alignment mit den gematchten Basenpaaren $a_1 \in \Omega_1$ und $a_2 \in \Omega_1$ noch einmal alle Zwischenergebnisse der Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r - 1, a_2^r - 1)$ bestimmt werden. Da dazu $O({n_1}^2 \cdot {n_2}^2)$ Operationen nötig sind, kommen im schlimmsten Fall noch einmal $O((n_1+n_2)\cdot {n_1}^2 \cdot {n_2}^2)$ Operationen für die Berechnung das multiplen Alignments hinzu.

Damit liegt der Zeitbedarf für die Suche des maximalen Eintrags in T und die Ableitung des Alignment insgesamt in $O((n_1+n_2)\cdot {n_1}^2\cdot {n_2}^2)$. Da die Berechnung von ${}^{x}_{x}M^{a_1}_{a_2}(a_1^r - 1, a_2^r - 1)$ keinen zusätzlichen Speicher benötigt, wird der Speicherbedarf nur von dem berechneten Alignment bestimmt und liegt deshalb in $O(n_1 + n_2)$.

Insgesamt setzt sich der Zeitaufwand für die Berechnung eines paarweisen lokalen Alignments also aus $O({n_1}^3 \cdot {n_2}^3)$ Operationen für die Berechnung aller Einträge in D, $O({n_1}^2 \cdot {n_2}^2)$ Operationen für die Berechnung der Einträge in T und $O((n_1+n_2)\cdot {n_1}^2 \cdot {n_2}^2)$ Operationen für die Bestimmung des Alignments zusammen. Zusammengefasst ergibt das einen Zeitbedarf von $O({n_1}^3 \cdot {n_2}^3)$.

Die Berechnung der k-besten Alignments erhöht diesen Bedarf noch einmal um den Faktor k. Da dieser aber konstant ist, bleibt die Gesamtkomplexität gleich.

Der Speicherbedarf des paarweisen lokalen Alignment-Algorithmus setzt sich aus $O(n_1 \cdot n_2)$ Speicher für die Einträge in den vier M-Tabellen, $O({n_1}^2 \cdot {n_2}^2)$ Speicher für die Einträge in D, $O(n_1 \cdot n_2)$ Speicher für die Einträge in T sowie $O(n_1 + n_2)$ Speicher für das berechnete Alignment zusammen und liegt damit insgesamt in $O({n_1}^2 \cdot {n_2}^2)$.

Basenpaarwahrscheinlichkeiten

Analysiert man die Komplexität in Abhängigkeit von o_1, der Anzahl der Basenpaare $a_1 \in \Omega_1$ und von o_2, der Anzahl der Basenpaare $a_2 \in \Omega_2$, erhält man eine obere Zeitschranke von $O({o_1}^2 \cdot {o_2}^2 \cdot n_1 \cdot n_2 \cdot (n_1 + n_2))$ und eine obere Speicherschranke von $O(o_1 \cdot o_2 + n_1 \cdot n_2)$.

Obwohl diese Abschätzung nur für Mengen Ω_1 und Ω_2 erreicht wird, die aus Basenpaaren mit unterschiedlichen Anfangs- und Endpositionen bestehen, zeigt sie dennoch, wie stark die Komplexität von der Anzahl der Basenpaare abhängt. Aus diesem Grund ist es sinnvoll, die Anzahl der Basenpaare zu begrenzen. Um dabei jedoch keine aussagekräftigen Basenpaare zu verlieren, sollte man nur diejenigen entfernen, welche über schlechte thermodynamische Eigenschaften verfügen.

Deshalb verwende ich einen Grenzwert p_{min} für die Basenpaarwahrscheinlichkeiten und betrachte im Algorithmus nur noch diejenigen Basenpaare $a_1 \in \Omega_1$ und $a_2 \in \Omega_2$ mit $p_{S_{R_1}}(a_1) \geq p_{min}$ und $p_{S_{R_2}}(a_2) \geq p_{min}$.

Durch die Verwendung eines konstanten p_{min} größer 0 sinkt auch die Gesamtkomplexität des Algorithmus. Die Ursache dafür liegt in der Tatsache, dass über einer Sequenz S der Länge n nur $1/p_{min}$ Basenpaare a mit $p_S(a) \geq p_{min}$ existieren können, welche in der selben Position Enden.

Damit muss die Suche nach eingefügten Alignments nur noch für eine konstante Anzahl von Basenpaaren durchgeführt werden, wodurch die Anzahl der notwendigen Operationen für die Auswertung eines Eintrages in den Gleichungen M nur noch von der Anzahl der möglichen Exklusionen abhängt und deshalb in $O(n_1+n_2)$ liegt.

Für die Berechnung eines Eintrages in D ist somit nur noch ein Zeitaufwand von $O((n_1 + n_2) \cdot n_1 \cdot n_2)$ nötig.

Da pro Position i in S nur noch eine konstante Anzahl von Basenpaaren existiert, welche in i Enden, liegt die Gesamtanzahl der möglichen Basenpaare über S in $O(n)$. Aus diesem Grund müssen auch nur noch $O(n_1 \cdot n_2)$ Einträge in D berechnet und gespeichert werden.

Insgesamt ergibt sich so für ein p_{min} größer 0 ein Zeitbedarf in $O((n_1 + n_2) \cdot {n_1}^2 \cdot {n_2}^2)$ und ein Speicherbedarf in $O(n_1 \cdot n_2)$.

Allerdings wird durch die Verwendung eines p_{min} größer 0 der Lösungsraum eingeschränkt, weshalb der Algorithmus nicht mehr garantiert eine optimale Lösung des in Definition 14 aufgestellten paarweisen lokalen Alignment-Problems berechnet.

Dabei zeigt sich in den Anwendungen des Algorithmus jedoch, dass die Einschränkung der Basenpaarwahrscheinlichkeiten kaum Auswirkungen aus das Ergebniss der Berechnung hat. Erst bei sehr hohen Werten für p_{min} kommte es zu einer sichtbaren Verschlechterung. Dies läßt vermuten, dass der Fehler gegenüber einer optimalen Berechnung gering ist.

Damit stellt das Einschränken der Basenpaarwahrscheinlichkeiten eine hervorragende Heuristik dar, um den Zeitaufwand des Algorithmus zu verringern. Ausführlicher gehe ich auf die verschiedenen Resultate im nächsten Kapitel ein.

3.3.2 Konsensus-Sequenz-Struktur-Algorithmus

Der Algorithmus zur Berechnung der Konsensussequenz und -Struktur erhält als Eingabe ein multiples Alignment A^m mit $|A^m| = n$, die Matrizen Ω_g mit $g \in \{1, \dots m\}$ aller möglichen Basenpaare mit ihren Wahrscheinlichkeiten, die sich nach RNAfold über der g-ten in A^m alignierten Sequenz ausbilden können und eine Liste κ mit den Konserviertheitsfaktoren aller Spalten in A^m.

Konsensussequenz

Als erstes bestimmt der Algorithmus für jede Spalte i mit $1 \leq i \leq n$ den Konsensusbuchstaben $S_C[\,i\,]$. Für unkonservierte Spalten mit $\kappa(i) < \kappa_{min}$ geschieht das in $O(1)$ Zeit.

Bei konservierten Spalten wird für jeden Buchstaben ω aus dem Konsensusalphabet Σ_C die Ähnlichkeit $\nu(\omega, A^m[\,i\,])$ von ω und der i-ten Spalte von A^m berechnet. Dazu wird für jede Zeile von A^m der Wert $\delta(\omega, A^m[\,g\,][\,i\,])$ bestimmt und diese Werte addiert. Da δ in konstanter Zeit berechnet werden kann und die Anzahl der $\omega \in \Sigma_C$ ebenfalls konstant ist, lassen sich die Ähnlichkeiten ν einer Spalte in $O(m)$ Zeit berechnen.

Die Bestimmung des Maximums und dessen Vergleich mit ν_{min} benötigen noch einmal konstante Zeit, wodurch sich insgesamt ein Zeitbedarf von $O(m \cdot n)$ für die Bestimmung aller Konsensusbuchstaben und damit auch für die Berechnung der Konsensussequenz ergibt. Der Speicherbedarf für die Sequenz beträgt $O(n)$.

Konsensusstruktur

Für die Berechnung der Konsensusstruktur bestimmt der Algorithmus zuerst die Konsensuspaarwahrscheinlichkeiten $p_{S_C}\big((i,j)\big)$ aller Paare von Positionen i und j mit $1 \leq i < j \leq n$. Diese ergeben sich dabei aus dem arithmetischen Mittel der Basenpaarwahrscheinlichkeiten über den einzelnen Zeilen.

Für die Berechnung einer einzelnen Wahrscheinlichkeit folgt daraus ein Aufwand von $O(m)$ Operationen. Da es insgesamt $O(n^2)$ verschieden Paare von Positionen gibt, werden für die Berechnungen aller Konsensuspaarwahrscheinlichkeiten $O(m \cdot n^2)$ Operationen und für deren Speicherung $O(n^2)$ Speichereinheiten benötig.

Aus den Wahrscheinlichkeiten berechnet der Algorithmus als nächstes die Basenpaargewichte. Da diese Berechnung eines Gewichts in konstanter Zeit erfolgt, liegt der Zeitbedarf dafür in $O(n^2)$, während der Speicherbedarf wieder $O(n^2)$ beträgt.

Für die Bestimmung der Konsensusstruktur verwendet der Algorithmus den Ansatz von Nussinov. Dieser wertet dabei für alle Paare von Positionen i und j mit $1 \leq i < j \leq n$ vier Rekusionsgleichungen aus, wobei der maximale Aufwand für die Suche nach einer Bifurkation besteht und in $O(n)$ liegt. Damit werden für die Bestimmung der Konsensusstruktur noch einmal $O(n^3)$ Zeit und $O(n^2)$ Speicher benötigt.

Insgesamt liegt der Resourcenbedarf für die Berechnung der Konsensusstruktur also in $O(n^3 + m \cdot n^2)$ Zeit und $O(n^2)$ Speicher. Diese Werte geben damit auch die Komplexität des ganzen Konsensus-Sequenz-Struktur-Algorithmus wieder.

3.3.3 Gesamtüberblick

Die Eingabe für den MuLoRA-Ansatz besteht aus den zu untersuchenden RNA-Sequenzen. Bei der Komplexitätsanalyse gehe ich dabei von m Sequenzen S_{R_1} bis S_{R_m} mit einer durchschnittlichen Länge von n Basen aus.

Basenpaarwahrscheinlichkeiten

Die Berechnung der Basenpaarwahrscheinlichkeitsmatrizen mit RNAfold benötigt pro Sequenz $O(n^3)$ Operationen und $O(n^2)$ Speicherplatz für die bis zu n^2 Basenpaarwahrscheinlichkeiten. Für alle m Sequenzen bedeutet dies einen Zeitbedarf von $O(m \cdot n^3)$ und einen Speicherbedarf von $O(m \cdot n^2)$.

Paarweise Alignments

Die Berechnung der paarweisen Alignments aller $m(m-1)/2$ Paare der m Eingabesequenzen benötigt einen Gesamtzeitbedarf von $O(m^2 \cdot n^5)$. Dieser würde sich ohne die Verwendung einer minimalen Basenpaarwahrscheinlichkeit p_{min} auf $O(m^2 \cdot n^6)$ erhöhen.

Der Speicherbedarf liegt in $O(m^2 \cdot n^2)$ und würde sich ohne p_{min} auf $O(m^2 \cdot n^4)$ erhöhen.

Multiple Alignments

Das Kantengewicht einer Alignmentkante lässt sich in konstanter Zeit berechnen. Für die Berechnung der Kanten aller paarweiser Alignments folgt daraus ein Ressourcenbedarf von $O(m^2 \cdot n)$ Zeit und Speicher.

Bei der Berechnung eines multiplen Alignments benötigt T-Coffee $O(m^3 \cdot n)$ Operationen für die Berechnung der erweiterten Bibliothek, $O(m^3)$ Operationen für die Berechnung des Neighbor-Joining-Baumes und $O(m \cdot n^2)$ operationen für die Berechnung des progressiven Alignments. Der Speicherbedarf für die Bibliotheken liegt in $O(m^2 \cdot n^2)$, die Berechnung der $m-1$ progressiven Alignments benötigt pro Schritt $O(n^2)$ Speicher, wobei dieser für jedes Alignment wieder verwendet wird und das multiple Alignment benötigt letztendlich noch einmal $O(m \cdot n)$ Speicher.

Die Kosten für die Berechnung der Konserviertheitsfaktoren der $O(m \cdot n)$ Spalten des multiplen Alignments setzen sich aus den Kosten für die Berechnungen der Kanten-Mengen $K(i)$ zusammen, welche die i-te Spalte bestätigen. Dabei muss für jedes Paar von Positionen in einer Spalte das entsprechende Alignment nach einer Kante mit diesem Paar durchsucht werden. Da die Kanten in quatratischen Tabellen bezüglich ihrer beiden Positionen gespeichert werden, benötigt die Suche nur eine konstante Zeit, wodurch sich ein Gesamtzeitbedarf von $O(m^3 \cdot n)$ für die Bestimmung aller Konserviertheitsfaktoren ergibt. Der Speicherbedarf setzt sich aus $O(m \cdot n^2)$

Einheiten für die Speicherung der Alignmentkanten der m Alignments und den $O(m \cdot n)$ Einheiten für die Speicherung der Konserviertheitsfaktoren zusammen.

Damit liegt der Gesamtaufwand für die Berechnung des multiplen Alignments bei $O(m^3 \cdot n + m \cdot n^2)$ Zeit sowie $O(m^2 \cdot n^2)$ Speicher.

Konsensussequenz und -Struktur

Da die Anzahl der Spalten des multiplen Alignments in $O(m \cdot n)$ liegt, sind für die Berechnung der Konsensussequenz und -struktur schließlich noch einmal $O(m^3 \cdot n^3)$ Operationen und $O(m^2 \cdot n^2)$ Speicher notwendig.

Zusammenfassung

Damit ergibt sich ein Gesamtzeitbedarf von $O(m \cdot n^3) + O(m^2 \cdot n^5) + O(m^3 \cdot n + m \cdot n^2) + O(m^3 \cdot n^3)$. Dies ergibt zusammengefasst ein Zeitbedarf von $O(m^2 \cdot n^5 + m^3 \cdot n^3)$. Ohne die Verwendung einer minimalen Basenpaarwahrscheinlichkeit p_{min} würde der Zeitbedarf in $O(m^2 \cdot n^6 + m^3 \cdot n^3)$ liegen. Da in den meisten Fällen $n \gg m$ gilt, wird der Zeitbedarf also hauptsächlich von der Berechnung der paarweisen Alignments bestimmt.

Der Gesamtspeicherbedarf beträgt $O(m \cdot n^2) + O(m^2 \cdot n^2) + O(m^2 \cdot n^2) + O(m^2 \cdot n^2)$, was zusammengefasst einen Speicherbedarf von $O(m^2 \cdot n^2)$ ergibt. Ohne die Verwendung einer minimalen Basenpaarwahrscheinlichkeit p_{min} würde dieser sich auf $O(m^2 \cdot n^4)$ erhöhen. Dabei wird auch der Speicherbedarf von der Berechnung der paarweisen Alignments bestimmt.

Damit ist die Analyse des Ansatzes beendet. Im nächsten Kapitel folgen die Ergebnisse.

Kapitel 4

Ergebnisse

Dieses Kapitel wendet sich den praktischen Seiten des Algorithmus zu. Im ersten Abschnitt beschreibe ich die Bestimmung der Parameter und begründe die Wahl einer Minimalwahrscheinlichkeit anhand einer Basenpaarwahrscheinlichkeitsanalyse. Im zweiten Abschnitt demonstriere ich zuerst die Leistungsfähigkeit des paarweisen lokalen Alignmentalgorithmus und wende mich dann dem kompletten MuLoRA-Ansatz zu.

Um dabei die Qualität der berechneten Ergebnisse einordnen zu können, verwende ich als Eingabe RNA-Sequenzen mit bekanntem multiplen Alignment und bekannter Konsensusstruktur. Damit ist es möglich, die Berechnungen mit den richtigen Werten zu vergleichen, wobei ich dabei zwei verschiedene Bewertungsfunktionen verwende:

Die erste Bewertungsfunktion s_{col} berechnet den Anteil der Basenpaare in der gegebenen Konsensusstrukturmenge P_{giv}, bei denen sowohl die Anfangs- und Endpositionen des Basenpaars als auch *alle* Positionen im multiplen Alignment A^m_{giv} der entsprechenden Spalten mit der Ergebnisstrukturmenge P_{res} bzw. dem berechnetem Alignment A^m_{res} übereinstimmen.

$$s_{col}(P_{res}, P_{giv}) = \frac{\left|\left\{a \in P_{giv} \mid \exists b \in P_{res}\Big(a_l = b_l \ \wedge \ a_r = b_r \ \wedge \ \forall g \in \{1,\ldots,m\} \ (A^m_{giv}[g][a_l] = A^m_{res}[g][a_l] \wedge A^m_{giv}[g][a_r] = A^m_{res}[g][a_r])\Big)\right\}\right|}{\left|P_{giv}\right|}$$

Die zweite Bewertungsfunktion s_{bp} ordnet zuerst jedes Basenpaar der gegebenen und berechneten Strukturmenge den Zeilen des entsprechenden multiplen Alignments zu und berechnet dann den Anteil der Basenpaare im gegeben Alignment, für die ein entsprechendes Basenpaar im berechnetem Alignment existiert.

$$s_{bp}(P_{res}, P_{giv}) = \frac{\left|\left\{(a,g) \mid a \in P_{giv}, g \in \{1,\ldots,m\}, \exists b \in P_{res}\Big(a_l = b_l \ \wedge \ a_r = b_r \wedge A^m_{giv}[g][a_l] = A^m_{res}[g][a_l] \wedge A^m_{giv}[g][a_r] = A^m_{res}[g][a_r]\Big)\right\}\right|}{m \cdot \left|P_{giv}\right|}$$

Von der Bedeutung entspricht s_{col} damit dem Anteil der komplett richtig erkannten Konsensusstruktur, während s_{bp} eher dem Anteil der Sequenzen angibt, bei dem die Konsensusstruktur richtig erkannt wurde. Damit ist die zweite Bewertungsfunktion s_{bp} nicht ganz so streng wie s_{col}.

Als Datenquelle für diese Analysen verwende ich die *Rfam* [GMM05]. Rfam ist eine Datenbank, in der über 500 verschiedene ncRNA-Familien in multiplen Alignments mit gegebener Konsensusstruktur gespeichert sind. Die Daten stammen

dabei aus über 200 kompletten Genomen aus allen drei Säulen des Lebens und liefern so über eine große taxonomische Bandbreite Informationen über konservierte funktionelle RNAs.

4.1 Parameterabschätzung

Der MuLoRA-Ansatz verwendet verschiedene Parameter für die Bestimmung der Basenpaarwahrscheinlichkeiten, für die Berechnung der paarweisen lokalen Alignments und für die Ableitung der Konsensussequenz und -Struktur. Während ich für den ersten Punkt, die Bestimmung der Basenpaarwahrscheinlichkeiten mit RNAfold, die standardmäßigen Energieparameter von Turner *et al.* [MSZ99] verwende, habe ich die Parameter für die restlichen Punkte des Ansatzes anhand eines Trainingssatzes aus der Rfam bestimmt.

4.1.1 Parametertraining

Bei der Wahl des Trainingssatzes spielen vor allem drei Punkte eine Rolle:

- Zum einen sollten die RNA-Familien eine möglichst große Bandbreite an verschiedenen Strukturen und Sequenzen abdecken, damit die gewonnenen Parameter eine hohe Allgemeingültigkeit haben.
- Andererseits hängen alle Basenpaarwahrscheinlichkeitsparameter, in MuLoRA betrifft das p_{sig}, von der Länge der Sequenzen ab. Die Ursache dafür liegt darin, dass über einer Sequenz S der Länge n bis zu $n-1$ verschiedene Basenpaare existieren können, die an der gleichen Position beginnen. Deshalb steigt mit n auch die Anzahl dieser möglichen Basenpaare. Da aber die Gesamtwahrscheinlichkeit aller möglichen Basenpaare, die an der selben Position beginnen, 1 ergeben muss, müssen mit steigenden n die Wahrscheinlichkeiten mit $1/n$ fallen.

  Damit ergibt sich für einen Wahrscheinlichkeitsparameter p_{par} eine Formel von $p_{par} = c_{par} \cdot 1/n$. Um nun den Faktor c_{par} bestimmen zu können, sollten die Sequenzlängen der gewählten Familien relativ gleich sein.
- Der letzte Punkt ist eher praktischer Natur. Da für das Training viele Durchläufe nötig sind, sollten nicht zu viele Familien verwendet werden und die Anzahl der Sequenzen und deren Längen nicht zu hoch sein.

Aufgrund dieser Anforderungen habe ich nach Familien mit drei bis vier Sequenzen, Durchschnittslängen zwischen 50 und 100 Nukleotiden und einer möglichst großen Bandbreite an Strukturen und Sequenzkonserviertheit gesucht.

Letztendlich habe ich mich für folgende zehn Familien mit einer durchschnittlichen Sequenzlänge von n, einer Sequenzanzahl von m und einer Anzahl von $|P|$ Basenpaaren in der Konsensusstruktur entschieden:

Nummer	n	m	$\|P\|$	Beschreibung
RF00112	86	3	15	RyeE RNA
RF00143	81	3	24	mir-6 microRNA precursor
RF00161	64	3	23	Nanos 3' UTR translation control element
RF00178	68	3	16	mir-24 microRNA precursor family
RF00245	80	3	22	mir-19 microRNA precursor family
RF00326	82	4	6	Small nucleolar RNA Z155
RF00338	95	3	17	Small nucleolar RNA snR53
RF00367	65	3	22	mir-BHRF1-3 microRNA precursor family
RF00371	92	3	19	sroE RNA
RF00502	53	4	12	Turnip crinkle virus core promoter hairpin

Bei den Trainingsdurchläufen habe ich neben den Kombinationen der im Ansatz vorgestellten Parameter zusätzlich getestet, ob es Sinn macht, eine Minimallänge λ_{min} für Exklusionen und einen Strafscore ϵ für das Einfügen einer Exklusion zu verwenden. Ersteres würde den Unterschied bei den biologischen Hindergründen von Gaps und Exklusionen stärker herausstellen. Während Gaps das Einfügen oder Entfernen von *einzelnen* Basen darstellen, beschreiben Exklusionen das Entfernen *ganzer* Sequenzbereiche. Da dies in der Evolution jedoch nicht einfach so geschieht, ist es auch gerechtferigt das Einfügen von Exklusionen zu bestrafen.

Die im Ansatz verwendeten Grenzwerte p_{min}, κ_{min} und ν_{min} habe ich für das Training auf 0 gesetzt, um die Lösungsmenge nicht einzuschränken. Für die Berechnung alternativer Motive verwende ich sowohl die Version der k-besten nicht überlappenden Alignments, als auch die Version ohne Einschränkungen, wobei ich k auf 3 gesetzt habe.

Insgesamt hat sich so bei Trainingsdurchläufen gezeigt, dass der Ansatz gegenüber Parameterschwangungen relativ robust ist, so lange die Parameter dabei noch ihrem „biologischen Sinn“ entsprechen. Deshalb ist die im Folgenden angegebene Parameterkombination nur eine von vielen, bei denen optimale Ergebnisse für den Trainingssatz erreicht wurden:

Substitutionsscores in σ: Gleiche Nukleotide erhalten eine Ähnlichkeitsbewertung von 4, ungleiche eine von -4.

Substitutionsscores in τ: Jedes gleiche Ende eines Basenpaars erhält einen Score von 1, jedes ungleiche einen von -1.

Gapkosten γ: Jedes Einfügen oder Entfernen einer Base erhält einen Strafscore von -10.

Signifikante Basenpaarwahrscheinlichkeit p_{sig}: Für den Längenbereich des Trainingssatzes sollte p_{sig} um die 10^{-3} sein. Da die Durchschnittslänge der enthaltenen Sequenzen $76,6$ Nukleotide beträgt, ergibt sich für p_{sig} insgesamt eine Formel von:

$$p_{sig} = \frac{1}{13 \cdot n}$$

Score für wiederholte Alignmentkanten ψ: Die Berechnung der k-besten überlappenden Alignments verstärkt entweder die Kantengewichte zwischen sehr stabilen Motiven oder führt zu alternativen Motiven. Für den Trainingssatz hat sich gezeigt, dass ein ausgewogenes Verhältniss zwischen diesen beiden Möglichkeiten mit ψ gleich -2 erreicht wird.

Minimale Exklusionslänge λ_{min}: Bei den einzelnen Durchläufen hat sich gezeigt, dass Exklusionen oft für Gaps eingebaut wurden. Dies hat zwar an den Ergebnissen nichts geändert, entspricht aber nicht dem Sinn der Exklusion. Deswegen verwende ich für λ_{min} einen Wert von 6.

Exklusionskosten ϵ: Auch das Einführen eines Strafscores für Exklusionen hatte keinen großen Einfluß auf die Ergebnisse, solange dieser nicht zu hoch war. Ich habe mich entschieden, das Durchführen einer Exklusion wie eine Basendeletion zu bestrafen und habe deshalb ϵ auf -10 gesetzt.

Mit diesen Parametern wurden folgende Ergebnisse für s_{bp} und s_{col} (in Prozent) erreicht:

Nummer	nicht-überlappend		beliebig	
	s_{bp}	s_{col}	s_{bp}	s_{col}
RF00112	100	100	100	100
RF00143	92	96	92	96
RF00161	91	97	91	97
RF00178	100	100	100	100
RF00245	86	95	86	86
RF00326	100	100	17	82
RF00338	88	90	88	90
RF00367	86	86	86	86
RF00371	100	100	100	100
RF00502	100	100	100	100
Ø	94	97	86	94

Der Grund für die schlechteren Ergebnisse der beliebigen k-besten Alignments liegt in der Zusammensetzung des Trainingssatzes. Bei einer anderen Zusammensetzung könnte es auch umgedreht aussehen. Der Grund dafür liegt darin, dass die RNA-Familien jeweils aus einem einzelnen Strukturmotiv bestehen. Da dieses schon bei dem ersten Alignment bestimmt wird, führen die restlichen Alignments zu Alternativen des Motivs. Damit erhöhen sich jedoch sowohl die Möglichkeiten für bessere als auch für schlechtere Alignments. Für die k-besten nicht überlappenden Alignments folgt daraus, dass deren Berechnung eigentlich nicht nötig ist.

Bei weiteren Analysen dazu hat sich dann auch bestätigt, dass die Ergebnisse bei der Bestimmung des besten Alignments, die Ergebnisse bei der Bestimmung der k-besten nicht überlappenden Alignments und die Ergebnisse bei der Bestimmung der k-besten beliebigen Alignments sich im Mittel kaum unterscheiden.

Somit hängt die Wahl der Version für die Berechnung alternativer Motive vom Vorwissen über die erwarteten Strukturen ab. Sollte dies nicht vorhanden sein, ist die Wahl reine Glückssache. Aus diesem Grund beziehen sich alle weiteren Betrachtungen in diesem Kapitel auch auf Berechnungen, bei denen nur die besten Alignments bestimmt wurden.

Weiterhin hat sich bei Vergleichen der berechneten Ergebnisse mit den gegebenen Alignments und Strukturen gezeigt, dass konservierte Motive im Allgemeinen einen Konserviertheisfaktor über $0,8$ haben. Damit biedet es sich an, κ_{min} auf $0,8$ zu setzten. Die Wahl des Grenzwertes ν_{min}, ab dem der häufigste Buchstabe als Konsensusbuchstabe betrachtet wird, ist dabei beliebig. Ich verwende einen Wert von $\nu_{min} = 0,5$.

Damit besteht die letzte Aufgabe bei der Abschätzung der Parameter darin, einen guten Grenzwert p_{min} für die minimale Basenpaarwahrscheinlichkeit zu bestimmen. Dieser sollte zum einen den Aufwand für die Berechnung so weit wie möglich senken, zum anderen aber nicht zu einer Verschlechterung der Ergebnisse führen.

Da sich optimale Strukturen aus Paarwahrscheinlichkeiten zusammensetzten, die größer oder gleich p_{sig} sind, wäre $p_{min} := p_{sig}$ ein erster Ansatz. Allerdings hängt p_{sig} von der Sequenzlänge ab, wodurch sich an der Gesamtkomplexität nichts ändern würde. Für die Wahl eines konstanten p_{min} ist jedoch ein genaueres Wissen über die für die Ausbildung von Strukturen verantwortlichen Basenpaarwahrscheinlichkeiten notwendig.

4.1.2 Basenpaarwahrscheinlichkeitsanalyse

Für eine Analyse der Basenpaarwahrscheinlichkeitsverteilung in natürlichen RNA-Strukturen habe ich für alle 13.040 Sequenzen in der Rfam die Paarwahrscheinlichkeiten mit RNAfold berechnet, die Wahrscheinlichkeiten der Basenpaare in der

Konsensusstruktur herausgefiltert und diese bezüglich der Sequenzlänge in verschiedene Gruppen eingeordnet.

Insgesamt erhielt ich so die Wahrscheinlichkeiten von 417.061 Basenpaaren, wobei die Sequenzlängen bis zu 850 Nukleotiden reichten. Für die Einteilung in die verschieden Gruppen verwendete ich Abstände von 50 Nukleotiden, wodurch sich 17 Gruppen ergaben.

Bei der Auswertung dieser Daten hat sich gezeigt, dass die Wahrscheinlichkeiten alles andere als gleichmäßig verteilt sind. Im Grunde genommen setzten sich die Strukturen nur aus Basenpaaren zusammen, deren Wahrscheinlichkeiten entweder bei 0 oder bei 1 liegen. Zwar sinkt der Anteil der hochwahrscheinlichen Basenpaare mit steigender Sequenzlänge n erwartungsgemäß mit $1/n$, jedoch ändert sich dabei nichts an dem Anteil der Wahrscheinlichkeiten in dem Bereich zwischen 0 und 1. Statt dessen steigt der Anteil der Basenpaare ohne Wahrscheinlichkeit an. Abbildung 4.1 verdeutlicht die Situation.

Die Konsequenz aus dieser Verteilung besteht darin, dass sich natürliche RNA-Strukturen hauptsächlich aus Basenpaaren zusammensetzen, die mit ihrer Wahrscheinlichkeit bei 0 liegen. Da diese aber zu schlechte thermodynamische Eigenschaften besitzen, um die Struktur zu erklären, wird diese von den wenigen restlichen Basenpaaren bestimmt, wobei deren Wahrscheinlichkeiten dann hauptschälich bei 1 liegen.

Aus diesem Grund ist es biologisch absolut gerechtferdigt, bei der Berechnung der paarweisen lokalen Alignments nur diejenigen Basenpaare für das Ableiten einer gemeinsammen Struktur zu betrachten, die mit ihrer Wahrscheinlichkeit über einem Grenzwert p_{min} liegen. Diese für das Strukturmotiv verantwortlichen Basenpaare enthalten dann alle notwendigen Information für die Ableitung des multiplen Alignments.

Sobald dieses bestimmt wurde, werden von MuLoRA wieder alle Basenpaare für das Berechnen der Konsensusstruktur berücksichtigt, wodurch diese dann auch die Basenpaare mit niedrigen Wahrscheinlichkeiten enthält und so die Realität wiedergibt.

Für die Wahl von p_{min} gilt dabei, dass es aufgrund der Paarwahrscheinlichkeitsverteilung auf einen Wert bis zu ungefähr 0.9 gesetzt werden kann, ohne dabei die Fehlerquote der paarweisen Alignments durch das mißachten vieler für die Struktur wichtiger Basenpaare zu stark zu erhöhen.

Der Berechnungsaufwand für die Ableitung einer gemeinsammen Struktur hängt mit $O(o^4)$ von der Anzahl o der von RNAfold angegebenen Basenpaare mit einer Wahrscheinlichkeit von mindestens p_{min} ab (siehe Abschnitt 3.3.1). Da diese Anzahl durch die deutlich gleichmäßigere Verteilung der von RNAfold bestimmten Wahrscheinlichkeiten höher als die Anzahl der für die Struktur wichtigen Basenpaare kleiner p_{min} ist, liegt die Zeitersparnis mit wachsendem $p_{min} \leq 0.9$ sehr deutlich über dem Zuwachs der Fehlerquote.

Soweit die Theorie. Im nächsten Abschnit wende ich mich den Anwendungen des Ansatzes zu.

4.2 Anwendungen

In diesem Abschnitt demonstriere ich die Leistungsfähigkeit des Ansatzes anhand verschiedener Aufgabenstellungen und vergleiche die Ergebnisse mit gebräuchlichen Alignmentprogrammen. Dabei habe ich mich für *ClustalW*, *Marna*, *pmmulti* und *RNAforester* entschieden.

Bei ClustalW [THG94] handelt es sich um ein Alignmentprogramm, welches nur auf Sequenzebene arbeitet. Für die Erstellung eines multiplen Alignments berechnet

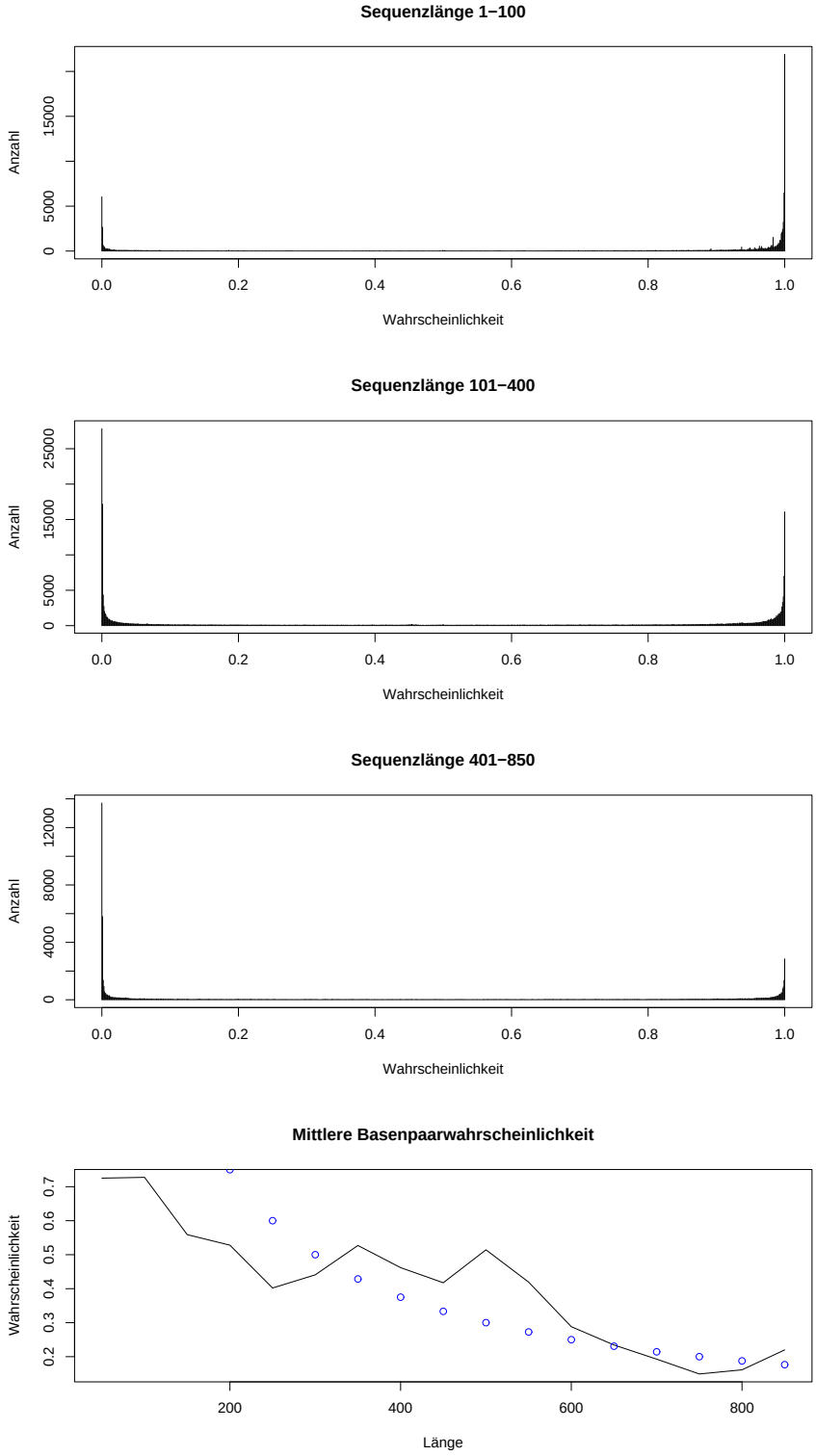

Abbildung 4.1: In den ersten drei Grafiken sind die Längenverteilungen zu drei repräsentativen Gruppen zusammengefasst. Die letzte Grafik zeigt den Rückgang der mittleren Wahrscheinlichkeit mit steigender Sequenzlänge n. Die Punkte geben dabei eine Vergleichskurve $f(n) = 150/n$ an.

ClustalW für alle Paare der Sequenzen ein paarweises lokales Alignment und leitet aus diesem mittels eines progressiven Ansatzes das Alignment ab.

Für die Berechnung der Konsensusstruktur eines ClustalW-Alignments verwende ich dabei das Programm RNAalifold des Vienna RNA secondary structure package [Hof03].

Dabei gilt für alle Berechnungen in diesem Kapitel, dass für MuLoRA die im letzten Abschnitt angegebenen Parameter und wenn nicht anderes angegeben $p_{min} = 0.01$ verwendet wurde, für alle anderen Programme ihre standartmäßigen Parameter verwendetet wurden und die Berechnungen auf einem Athlon XP 2000+ mit 1 GByte Speicher durchgeführt wurden.

Für eine Abschätzung von möglichen Anwendungen habe ich zuert die Laufzeit untersucht.

4.2.1 Laufzeitanalyse

Um die Laufzeit des Ansatztes in Abhängigkeit der Sequenzlängen n zu bestimmen, habe ich sechs Familien mit jeweils 4 Sequenzen in einem Bereich von 50 bis 400 Nukleotiden ausgewählt. Für diese Familien habe ich dann mit mehreren Werten für p_{min} ein multiples Alignment berechnet und dabei die durchschnittliche Anzahl $|\Omega|$ der von RNAfold angegebenen möglichen Basen mit einer Wahrscheinlichkeit von mindestens p_{min}, die Laufzeit t in Sekunden und die Ergebnisbewertungen s_{col} bzw. s_{bp} bestimmt.

Um die erhaltenen Werte besser einschätzen zu können, habe ich anschließend mit den Vergleichsprogrammen nochmals multiple Alignments für die gewählten Familien berechnet und dabei wieder t, s_{col} und s_{bp} bestimmt. Insgesamt ergaben sich so folgende Resultate:

RF00502: $n = 53$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	82	16	11	11	11	11	11
t	6	1	1	1	1	1	1
s_{col}	100	100	100	100	100	100	100
s_{bp}	100	100	100	100	100	100	100

	ClustalW	Marna	RNAforester	pmmulti
t	1	1	1	2
s_{col}	92	100	100	92
s_{bp}	92	100	100	92

RF00288: $n = 97$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	337	66	29	25	25	22	19
t	360	105	48	37	36	24	20
s_{col}	100	100	100	100	100	100	100
s_{bp}	100	100	100	100	100	100	100

	ClustalW	Marna	RNAforester	pmmulti
t	1	2	1	8
s_{col}	86	86	86	86
s_{bp}	86	86	86	86

RF00433: $n = 152$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	1242	167	80	60	33	23	15
t	9.560	936	308	229	109	43	28
s_{col}	86	86	86	86	86	86	86
s_{bp}	86	86	86	86	86	86	86

	ClustalW	Marna	RNAforester	pmmulti
t	1	9	2	202
s_{col}	86	92	73	88
s_{bp}	86	98	75	88

RF00457 $n = 204$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	1746	285	102	59	42	24	11
t	49.864	3.436	654	219	97	15	1
s_{col}	67	67	55	55	55	55	67
s_{bp}	67	67	61	61	61	61	67

	ClustalW	Marna	RNAforester	pmmulti
t	1	15	–	1.300
s_{col}	45	48	–	85
s_{bp}	45	62	–	85

RF00459: $n = 286$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	2860	352	141	98	68	36	6
t	$> 1.000.000$	8.237	2.207	1.391	1.058	519	58
s_{col}	–	0	0	0	0	0	0
s_{bp}	–	1	0	0	0	0	0

	ClustalW	Marna	RNAforester	pmmulti
t	1	27	–	–
s_{col}	5	12	–	–
s_{bp}	5	18	–	–

RF00222: $n = 370$

p_{min}	0,00	0,01	0,10	0,25	0,50	0,75	0,90
$\|\Omega\|$	3173	610	231	138	93	48	20
t	$> 1.000.000$	38.914	12.138	7.248	4.073	1.291	497
s_{col}	–	10	10	10	10	10	10
s_{bp}	–	10	10	10	10	10	10

	ClustalW	Marna	RNAforester	pmmulti
t	4	151	–	–
s_{col}	25	0	–	–
s_{bp}	25	0	–	–

Damit hat sich erst einmal die im letzten Abschnitt aufgestellte Vermutung bestätigt, dass die Wahl von hohen Wahrscheinlichkeiten für p_{min} sich nicht negativ auf die Ergebnisse auswirkt, sofern dabei die Grenze zu den hohen Basenpaaranteilen mit einer Wahrscheinlichkeit bei 1 nicht überschritten wird.

Einmal (RF00457) wurde sogar mit einem hohen Wert für p_{min} das selbe gute Ergebniss, wie für einen niedrigen Wert erziehlt, während alle dazwischenliegenden

Werte für p_{min} schlechtere Ergebnisse erreichten. Bei weiteren Berechnungen hat sich dieses Phänomen sehr oft gezeigt. Eine Erklärung könnte darin liegen, das mehrere mögliche Basenpaare mit mittleren Wahrscheinlichkeiten eine alternative Struktur bilden können. In der Natur wird jedoch durch das Zusammenspiel von niedrig- und hochwahrscheinlichen Basenpaaren eine künstigere Struktur erreicht.

Während bei der Verwendung niedriger p_{min} dieses Zusammenspiel nicht gestört wird, fallen bei mittleren Werten für p_{min} die Basenpaare mit niedrigen Wahrscheinlichkeiten aus der Berechnung raus und die alternative Struktur wird begünstigt. Erhöht man jedoch p_{min} noch mehr, fallen auch die mittleren Wahrscheinlichkeiten raus und die hochwahrscheinlichen Basenpaare bestimmen wieder die richtige Struktur.

Für genaue Aussagen ist jedoch eine aufwendigere Analyse der Bedeutung von Basenpaaren für die Struktur in Abhängigkeit ihrer Wahrscheinlichkeit notwendig.

Des weiteren zeigt sich, dass durch die Verwendung von $p_{min} > 0$ tatsächlich ein drastischer Rückgang der Laufzeit erreicht wird. Wie stark dieser Rückgang jedoch genau ist, hängt von der Anzahl der ausgeschlossenen Basenpaare und damit von der Zusammensetzung der von RNAfold angegebenen Basenpaare ab. Da jedoch die von RNAfold angegebenen Basenpaare nicht besonders gut mit der maximal möglichen Anzahl von Basenpaaren korrelieren und die Anzahl der Basenpaare mit einer Wahrscheinlichkeit größer p_{min} mit wachsenden p_{min} auch unterschiedlich stark fällt, ist eine genauere Analyse schwierig.

Das selbe gilt auch für eine generelle Abschätzung der Laufzeit. Da diese von der Länge der Sequenz, von der Anzahl und Zusammensetzung der betrachteten Basenpaare und von den Abständen zwischen den Basenpaaren abhängt, ist es umöglich, genauere Angaben zu machen.

Für $p_{min} = 0$ wurden bei keiner Familie bessere Ergebnisse erziehlt. Jedoch hat sich gezeigt, dass die Laufzeit für praktische Anwendungen unakzeptabel ist. So habe ich beispielsweise die Berechnungen für die beiden längsten Sequenzen nach über einer Woche abgebrochen. Aus diesem Grund gehe ich den weiteren Analysen auch nicht mehr auf $p_{min} = 0$ ein.

Im Vergleich mit den anderen Programmen zeigt sich, das die Laufzeiten von ClustalW, Marna und RNAforester denen von MuLoRA auch für große p_{min} überlegen sind. Da diese aber mit einer festen Anzahl von Strukturen rechnen, ist dies auch nicht verwunderlich. Das Laufzeitverhalten von pmmulti ist dabei dem von MuLoRA für mittlere p_{min} ähnlich. Allerdings brechen sowohl pmmulti als auch RNAforester die Berechnung von multiplen Alignments bei zu langen oder zu unterschiedlichen Sequenzen ab.

Der Vergleich der Ergebnisse lässt dabei auch noch keine festen Schlüsse zu. Jedes Programm zeigt hier je nach Sequenzbeschaffenheit Stärken und Schwächen. Aus diesem Grund habe ich für eine Analyse der Ergebnisse die RNA-Familien bezüglich ihrer Sequenzkonserviertheit in zwei Gruppen unterteilt. Während die Familien mit konservierten Sequenzen nur selten Differenzen innerhalb der Alignmentspalten aufweisen, enthalten die Familien mit unkonservierten Sequenzen viele komplimentäre Basenaustausche und oft größere Lücken.

4.2.2 Konservierte Sequenzen

Für die Gruppe mit den konservierten Sequenzen habe ich 97 Familien ausgewählt, die meist aus kurzen, gleich langen Sequenzen bestehen. Allerdings gibt es auch Fälle, bei denen Anfangs oder Endregionen in einigen Sequenzen fehlen oder leichte Sequenzunterschiede vorhanden sind.

Erwartungsgemäß sollte bei diesen Sequenzen das Betrachten von strukturellen Komponenten keine große Rolle für das Ergebniss spielen. Um dies zu testen, berechne ich die multiplen Alignments wieder mit mehreren Werten für p_{min}.

Bei den Berechnungen habe ich folgende Ergebnisse erhalten:

	0,01		0,25		0,50		0,75		1,00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
Ø	66	72	65	72	66	73	67	71	67	71
	MuLoRA		ClustalW		Marna		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
Ø	66	72	61	63	59	63	26	29	60	65

Die Tabellen mit den jeweiligen Werten für jede einzelne Familie befinden sich im Anhang (Tabellen A.1 bzw. A.2).

Bei den Ergebnissen bestätigte sich die Vermutung, dass das Betrachten von strukturellen Komponenten keinen großen Einfluß auf die Qualität des Ergebnisses hat. Selbst für $p_{min} = 1$ und der dementsprechend geringen Anzahl von betrachteten Basenpaaren zeigt sich kein schlechteres Ergebniss als in den anderen Spalten. Im Gegenteil waren diese sogar besser als diejenigen mit niedrigen Werten für p_{min}, was wohl daran liegt, das falsche Strukturen durch den Wegfall vieler Basenpaare ausgeschlossen werden und das richtige Alignment dann mit Hilfe der Sequenz berechnet werden kann.

Der Vergleich der einzelnen Programme untereinander zeigt auch keine Überraschungen. Die durchschnittlichen Ergebnisse liegen alle nah beieinander, was fast immer auch auf die einzelnen Ergebnisse zutrifft. Die schlechten Ergebnisse für RNAforester kommen dabei durch die häufigen Abbrüche bei der Berechnung multipler Alignments von unterschiedlich langen Sequenzen zu stande.

Da sich so keine Aussagen zu den Stärken und Schwächen von MuLoRA machen lassen, wende ich mich nun herausforderenden Aufgaben zu.

4.2.3 Unkonservierte Sequenzen

Für die Untersuchung der Ergebnisse bei Alignments von unkonservierten Sequenzen habe ich insgesamt 30 RNA-Familien mit einer hohen Anzahl von komplimentären Basenaustauschen und größeren Lücken im mutliplen Alignment herausgesucht. Deshalb sollte bei diesen Familien das Erkennen der konservierten Strukturbereiche eine bedeutente Rolle für das berechnen richtiger Alignments spielen.

Insgesamt erhielt ich dabei folgendes Resultat:

	0,01		0,25		0,50		0,75		1,00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
Ø	41	61	39	60	40	56	41	58	32	48
	MuLoRA		ClustalW		Marna		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
Ø	41	61	23	30	25	48	3	6	18	32

Die Tabellen mit den jeweiligen Werten für jede einzelne dieser Familien befinden sich im Anhang (Tabellen A.3 bzw. A.4).

Auch diesmal zeigte sich, das die Wahl von p_{min} keinen großen Einfluß auf die Qualität der Ergebnisse hat und damit ruhig hoch gewählt werden kann. Allerdings dürfen dabei, wie bei der Laufzeitanalyse schon erwähnt, die für die Struktur wichtigen Basenpaare mit einer Wahrscheinlichkeit bei 1 nicht ausgeschlossen werden.

RF00008 Hammerhead ribozyme (type III)

Rfam:

```
AF170503     GAAAGGUCUGUGCUU......AGCACACUGACGAGUUCCUGA.......................AAUGGAACGAAACCUUUU
Y14700       .UCCAGUCGAGACCUGAAGUGGGUUUCCUGAUGAGGCUGUGGAGAGAGCGAAAGCUUUACUCCCACACAAGCCGAAACUGGA.
AJ247122     .UCCAGUCGAGACCUGAAGUGGGUUUCCUGACGAGGCUGUGGAGAGAGCUAUUGCUUUACUCCCGCACAAGCCGAAACUGGA.
AJ005300     UAAGAGUCUGUGGUA......AGCACACUGAUGAGUCUCUGA.......................AAUGAGACGAAACUCUUG
AJ005302     AAAGAGUCUGUGCUA......AGCACACUGAUGAGUCUCUGA.......................AUUGAGACGAAACUCUUG
AJ550911     GAAGAGUCUGUGCUA......AGCACACUGACGAGUCUCUGA.......................AAUGAGACGAAACUCUUC
AJ550898     GAAGAGUCUGCGCUA......AGCGCACUGAUGAGUCUUUGA.......................AAUAAGACGAAACUCUUC
#CS          .((((((.(((((.........)))))).......((((.................................))))...)))))).
```

MuLoRA:

```
AF170503     GAAAGGUCUGUGC______UUAGCACACUGACGAGUUCCUGAA_____________________________AUGGAACGAAACCUUUU
Y14700       _UCCAGUCGAGACCUGAAGUGGGUUUCCUGAUGAGGCUGUGGAGAGAGCGAAAGCUUUACUCCCACACAAGCCGAAACUGGA_
AJ247122     _UCCAGUCGAGACCUGAAGUGGGUUUCCUGACGAGGCUGUGGAGAGAGCUAUUGCUUUACUCCCGCACAAGCCGAAACUGGA_
AJ005300     UAAGAGUCUGUGG______UAAGCACACUGAUGAGUCUCUGAA_____________________________AUGAGACGAAACUCUUG
AJ005302     AAAGAGUCUGUGC______UAAGCACACUGAUGAGUCUCUGAA_____________________________UUGAGACGAAACUCUUG
AJ550911     GAAGAGUCUGUGC______UAAGCACACUGACGAGUCUCUGAA_____________________________AUGAGACGAAACUCUUC
AJ550898     GAAGAGUCUGCGC______UAAGCGCACUGAUGAGUCUUUGAA_____________________________AUAAGACGAAACUCUUC
#CS          .((((((.(((((______...)))))).......((((.....______________________________...)))))...)))))).

              1111111                                                                    111111111
60000000999994444449999999999999999988887744455555555555555555544777999900000000005
20000000000003333330000000000000000011111133322222222222222222331110000000000007
```

ClustalW mit RNAalifold:

```
AF170503     -----GAAAGGUCUGUGCUUAGCACACUGACGAGUUCCUGAAAUGGAACGAAACCUUUU----------------------
AJ550911     -----GAAGAGUCUGUGCUAAGCACACUGACGAGUCUCUGAAAUGAGACGAAACUCUUC----------------------
AJ550898     -----GAAGAGUCUGCGCUAAGCGCACUGAUGAGUCUUUGAAAUAAGACGAAACUCUUC----------------------
AJ005300     -----UAAGAGUCUGUGGUAAGCACACUGAUGAGUCUCUGAAAUGAGACGAAACUCUUG----------------------
AJ005302     -----AAAGAGUCUGUGCUAAGCACACUGAUGAGUCUCUGAAUUGAGACGAAACUCUUG----------------------
Y14700       UCCAGUCGAGACCUGAAGUGGGUUUCCUGAUGAGGCUGUGGAGAGAGCGAAAGCUUUACUCCCACACAAGCCGAAACUGGA
AJ247122     UCCAGUCGAGACCUGAAGUGGGUUUCCUGACGAGGCUGUGGAGAGAGCUAUUGCUUUACUCCCGCACAAGCCGAAACUGGA
#CS          .....(((((((..(((.....)))).........(((((......)))))...)))))))......................
```

Marna:

```
AF170503     GAAAGGUC-UGUGC-UUA---GCACA-CUGACGA-G-UUC---------CUGAA---------AUGGA--A-C-GAAACCUUUU
Y14700       U-CCAGUCGAGACCUGAAGUGGGUUUCCUGAUGAGGCUGUGGAGAGAGCGAAA-GCUUUACUCCCACACAAGCCGAAACUGG-A
AJ247122     U-CCAGUCGAGACCUGAAGUGGGUUUCCUGACGAGGCUGUGGAGAGAG-CUAUUGCUUUACUCCCGCACAAGCCGAAACUGG-A
AJ005300     UAAGAGUC-UGUGG-UAA---GCACA-CUGAUGAGUCU-----------CUGA------A-----AU--GAGACGAAACUCUUG
AJ005302     AAAGAGUC-UGUGC-UAA---GCACA-CUGAUGAGUCU-----------CUGA------A-----UU--GAGACGAAACUCUUG
AJ550911     GAAGAGUC-UGUGC-UAA---GCACA-CUGACGAGUCU-----------CUGA------A-----AU--GAGACGAAACUCUUC
AJ550898     GAAGAGUC-UGCGC-UAA---GCGCA-CUGAUGAGUCU-----------UUGA------A-----AU--AAGACGAAACUCUUC
#CS          (((((((._(((((_...___)))))_.......((((.._________(...______.___.....)))))...)))))))
```

Abbildung 4.2: Alignments von sechs Hammerhead Ribozymen des Typ III. MuLoRA ist als einzigtes Programm in der Lage, die Konsensusstruktur richtig zu erkennen. RNAforester und pmmulti hingegen berechneten überhaupt kein multiples Alignment

Indem ich aber für $p_{min} = 1$ einen großen Teil davon doch ausgeschlossen habe, ist das schlechtere Ergebniss in der entsprechenden Spalte nicht verwunderlich.

Für $p_{min} = 0.75$ wurde wieder ein besseres Ergebnisse erziehlt, als für mittlere Werte für p_{min}. Damit wird die in der Laufzeitanalyse aufgestellte These, dass hohe Grenzwerte p_{min} besser als mittlere Grenzwerte sind, nochmals bestätigt.

Der Vergleich der einzelnen Programme geht diesmal eindeutig zu Gunsten von MuLoRA aus. Während RNAforester nur sehr selten und pmmulti nur in der Hälfte der Fälle überhaupt ein Alignment berechnete, hatten auch Marna und ClustalW Schwierigkeiten mit den evolutionären Sequenzlücken.

MuLoRA hingegen handhabt diese Lücken mit dem durchführen von Exklusionen. Dies stellt eindrucksvoll den Sinn einer strukturellen Lokalität bei der Berechnung multipler Alignments unter Beweis. Abbildung 4.2 zeigt als Beispiel das in der Rfam gegebene und die von den verschiedenen Programmen berechneten Alignments der RNA-Familie RF00008. Dabei handelt es sich um die schon erwähnten Hammerhead Ribozyme des Typ III, deren Struktur in Abbildung 1.1 gezeigt wird.

Als letztes habe ich ein Strukturmotiv der Rfam um genomische Bereiche in die 3'- und 5'-Richtung erweitert und getestet, ob MuLoRA in der Lages ist, das gemeinsamme Motiv zu finden.

4.2.4 Motivsuche

Für die Suche nach einen konservierten Motiv in genomischen Sequenzen habe ich das Motiv der RNA-Familie RF00173 verwendet. Dabei handelt es sich um Hairpin Ribozyme, die in folgenden Genen enthalten sind:

- M21212: Arabis mosaic virus small satellite RNA
  Das Gen besteht aus 300 Nuleotiden, wobei das Motiv 52 Nukleotide lang ist und an Position 144 beginnt.
- D00685: Chicory yellow mottle virus small satellite RNA
  Das Gen besteht aus 457 Nuleotiden, wobei das Motiv 52 Nukleotide lang ist und an Position 152 beginnt.
- M17439: Tobacco ringspot virus mRNA
  Das Gen besteht aus 360 Nuleotiden, wobei das Motiv 52 Nukleotide lang ist und an Position 135 beginnt.

Nach der Berechnung der multiple Alignments hat sich folgendes Bild ergeben:

	MuLoRA		ClustalW		Marna		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00173	62	88	100	100	0	58	—	—	—	—

Damit hat MuLoRA 62 Prozent aller Spalten richtig entdeckt und sogar 88 Prozent aller Basenpaare des Motivs richtig vorausgesagt. Diese Ergebnis wurde jedoch aufgrund der hohen Sequenzähnlichkeit der Gene von ClustalW in den Schatten gestellt.

Trotzdem zeigt dieses Resultat, dass MuLoRA sehr gut im Stande ist, konservierte Motive in Sequenzen zu erkennen. Die Ergebisse bei den unkonservierten Sequenzen verdeutlichen dabei, dass durch die Handhabung einer strukturellen Lokalität selbst Motive in weit entfernten Sequenzen gefunden werden, wodurch MuLoRA anderen Programmen diesbezüglich überlegen ist.

Dabei arbeitet MuLoRA durch eine biologisch gerechtfertigte Einschränkung der betrachteten Basenpaare bei der Berechnung einer gemeinsamen Struktur effizient, ohne das sich die Ergebnisse des Ansatzes verschlechtern.

Kapitel 5

Zusammenfassung und Ausblick

In dieser Arbeit habe ich erstmalig einen Ansatzes für die Berechnung mutlipler lokaler Sequenz-Struktur-Alignments mit einer auf Strukturen zugeschnittenen Form von Lokalität entwickelt.

Dazu habe ich eine Form von struktureller Lokalität vorgestellt und auf dieser Grundlage das paarweise lokale Alignmentproblem definiert. Anschließend habe ich einen Algorithmus entwickelt, der diese Problem löst. Dabei berücksichtige ich durch die Verwendung von thermodynamischen Informationen in Form von Basenpaarwahrscheinlichkeiten allen möglichen Strukturen über den Eingabesequenzen.

Dieser Algorithmus liefert so für meinen Ansatz Sequenz- und Strukturinformationen von lokalen Motiven. Aus diesen Informationen berechne ich dann mit Hilfe des fortschrittlichen multiplen Alignmentprogramms T-Coffee, ein multiples Alignment.

Um die Lokalität des multiplen Alignments auszudrücken, habe ich eine Bewertungsfunktion entwickelt, welche durch die Informationen der paarweisen Alignmentkanten die Konserviertheit der Spalten bestimmt. Für die anschließende Berechnung der Konsensussequenz- und Struktur habe ich einen Algorithmus entwickelt, welcher auf Grundlagen der Basenpaarwahrscheinlichkeiten die wahrscheinlichste Struktur über dem Alignment bestimmt.

Durch die Einschränkung der betrachteten Basenpaare bei der Berechnung der paarweisen Alignments, arbeitet mein Ansatz in Vergleich zu verwandten Strukturvorhersageansätzen effizienter, ohne das darunter die Qualität der Ergebnisse leidet. Um dies nachzuweisen, habe ich eine Analyse der Basenpaarwahrscheinlichkeitsverteilung in natürlichen RNA-Sequenzen durchgeführt und gezeigt, dass diese Einschrängung biologisch gerechtfertigt ist.

Anschließend habe ich durch Berechnungen von multiplen Alignments aus natürlichen Sequenzen und den Vergleich der Ergebnisse mit den korrekten Alignments aus der Rfam nachgewiesen, dass der Algorithmus gute Ergebnisse erziehlt, bei evolutionär entfernten Sequenzen anderen Programmen überlegen ist und effizient arbeitet.

Ziele für zukünftige Erweiterungen sehe ich vor allem in einer weiteren Verbesserung der Laufzeit. So könnte durch die Anwendung von subquadratischen Sequenze-Alignment-Techniken [CLZ03] eventuell eine geringere Laufzeit erreicht werden.

Eine weiterer Punkt währe die Ausdehnung des Lösungsraumes auf crossing-Strukturen. So könnte man eventuell durch die Berechnung der k-besten, nicht überlappenden Alignments und deren Zusammenführen wie es in Abbildung 3.5 gezeigt wird, hoch wahrscheinliche Pseudoknoten finden.

Anhang A

Ergebnistabellen

A.1 Konservierte Sequenzen

Vergleich von verschiedenen p_{min}:

	0,01		0,25		0,50		0,75		1,00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00014	77	77	77	77	77	77	77	77	77	77
RF00051	100	100	100	100	100	100	92	99	92	99
RF00071	100	100	0	50	100	100	100	100	100	100
RF00074	76	90	81	100	81	100	100	100	100	100
RF00076	88	97	100	100	75	88	100	100	100	100
RF00085	0	0	0	0	0	50	0	0	0	0
RF00087	0	50	0	50	0	50	0	50	0	50
RF00088	0	0	0	0	0	0	0	0	0	0
RF00089	30	78	30	78	30	78	30	78	30	78
RF00110	100	100	100	100	92	98	92	98	92	98
RF00116	83	83	83	83	83	83	83	83	83	83
RF00119	100	100	100	100	100	100	100	100	100	100
RF00120	50	50	50	50	50	50	50	50	50	50
RF00131	100	100	100	100	100	100	100	100	100	100
RF00136	100	100	100	100	100	100	100	100	100	100
RF00146	100	100	100	100	100	100	100	100	100	100
RF00151	100	100	100	100	100	100	100	100	100	100
RF00153	0	17	0	0	0	0	0	0	0	0
RF00154	100	100	100	100	100	100	100	100	100	100
RF00157	0	0	0	0	0	0	0	0	0	0
RF00158	40	40	0	33	0	33	0	33	0	33
RF00173	62	88	12	71	12	71	0	67	0	67
RF00184	100	100	100	100	100	100	100	100	100	100
RF00186	75	75	75	75	75	75	75	75	75	75
RF00187	100	100	100	100	100	100	100	100	100	100
RF00188	42	58	42	58	42	58	42	58	42	58
RF00196	100	100	100	100	100	100	100	100	100	100
RF00197	81	85	81	85	88	96	88	96	88	96
RF00207	100	100	100	100	100	100	100	100	100	100
RF00208	100	100	100	100	100	100	100	100	100	100
RF00217	100	100	100	100	100	100	100	100	100	100
RF00219	100	100	100	100	100	100	100	100	100	100

	0,01		0,25		0,50		0,75		1,00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00249	100	100	100	100	100	100	100	100	100	100
RF00253	87	87	87	89	87	89	87	89	87	89
RF00254	100	100	100	100	100	100	100	100	100	100
RF00255	100	100	100	100	100	100	57	83	57	83
RF00257	54	61	54	74	54	72	54	73	54	73
RF00258	81	94	81	94	81	94	81	94	81	94
RF00262	83	83	83	83	83	83	83	83	83	83
RF00266	100	100	100	100	100	100	100	100	100	100
RF00267	41	41	41	41	41	41	41	41	41	41
RF00268	100	100	100	100	100	100	100	100	100	100
RF00270	100	100	100	100	100	100	100	100	100	100
RF00271	0	30	44	44	44	44	44	44	44	44
RF00273	10	10	0	0	0	10	10	18	10	18
RF00274	29	33	29	39	0	28	43	33	43	33
RF00276	0	6	0	9	0	9	0	9	0	9
RF00277	0	0	67	89	89	97	0	0	0	0
RF00280	75	75	75	81	75	81	75	81	75	81
RF00281	50	50	50	45	50	45	50	45	50	45
RF00282	14	14	0	0	0	0	0	0	0	0
RF00283	0	52	0	49	0	22	0	20	0	20
RF00284	50	58	50	58	60	60	60	60	60	60
RF00285	38	54	62	62	62	62	62	62	62	62
RF00287	29	29	29	29	29	29	29	29	29	29
RF00288	100	100	100	100	100	100	100	100	100	100
RF00289	67	67	7	47	67	67	67	67	67	67
RF00292	0	0	0	0	0	0	0	0	0	0
RF00293	17	17	17	17	17	17	17	17	17	17
RF00297	45	45	45	45	45	45	45	45	45	45
RF00299	0	0	0	0	0	0	0	0	0	0
RF00301	38	38	38	38	38	38	38	38	38	38
RF00310	64	64	64	64	64	64	64	64	64	64
RF00317	100	100	100	100	100	100	100	100	100	100
RF00318	100	100	100	100	100	100	100	100	100	100
RF00320	73	91	73	91	73	91	73	91	73	91
RF00323	100	100	100	100	100	100	100	100	100	100
RF00325	100	100	100	100	100	100	100	100	100	100
RF00329	100	100	100	100	100	100	100	100	100	100
RF00331	100	100	100	100	100	100	100	100	100	100
RF00332	100	100	100	97	100	97	100	97	100	97
RF00339	40	40	40	40	40	40	40	40	40	40
RF00343	0	100	0	100	0	100	100	100	100	100
RF00346	75	75	75	75	75	75	75	75	75	75
RF00347	0	0	0	0	0	0	0	0	0	0
RF00351	62	79	62	79	62	79	88	88	88	88
RF00352	100	100	100	100	100	100	100	100	100	100
RF00355	18	18	18	18	18	18	18	18	18	18
RF00358	100	100	100	100	100	100	100	100	100	100
RF00359	44	92	44	91	44	74	56	97	56	97
RF00361	87	87	87	87	67	80	67	80	67	80
RF00363	96	96	96	97	96	97	96	97	96	97
RF00365	100	100	100	100	100	100	100	100	100	100
RF00366	100	100	100	100	100	100	100	100	100	100

	0,01		0,25		0,50		0,75		1,00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00370	79	79	79	79	79	79	79	79	79	79
RF00382	82	94	82	94	82	94	82	94	82	94
RF00390	100	100	100	100	100	100	100	100	100	100
RF00439	77	77	69	73	69	73	69	73	69	73
RF00440	100	100	100	100	100	100	100	100	100	100
RF00454	0	0	0	2	0	0	0	0	0	0
RF00455	100	100	100	100	100	100	96	99	96	99
RF00466	69	69	69	69	69	69	69	69	69	69
RF00493	100	100	100	100	100	100	100	100	100	100
RF00494	69	69	69	69	69	69	69	69	69	69
RF00498	100	100	100	100	100	100	100	100	100	100
RF00500	100	100	100	100	100	100	100	100	100	100
RF00505	0	0	0	0	0	0	0	0	0	0
Ø	66	72	65	72	66	73	67	71	67	71

Tabelle A.1: Vergleich der Alignmentergebnisse über konservierten Sequenzen für verschiedene Werte p_{min}

Vergleich der einzelnen Programme:

	MuLoRA		Marna		ClustalW		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00014	77	77	100	100	77	77	73	74	73	73
RF00051	100	100	100	100	100	100	100	100	88	88
RF00071	100	100	0	0	50	88	–	–	0	75
RF00074	76	90	76	95	100	100	10	90	81	98
RF00076	88	97	96	96	75	81	79	85	83	83
RF00085	0	0	0	0	0	50	–	–	100	100
RF00087	0	50	100	100	0	0	–	–	75	75
RF00088	0	0	0	0	0	0	–	–	0	0
RF00089	30	78	50	72	80	90	–	–	0	35
RF00110	100	100	42	85	75	90	83	90	92	98
RF00116	83	83	57	62	83	83	–	–	96	96
RF00119	100	100	45	45	100	100	–	–	100	100
RF00120	50	50	100	100	100	100	–	–	38	38
RF00131	100	100	93	93	100	100	89	89	85	85
RF00136	100	100	100	100	71	71	–	–	71	71
RF00146	100	100	100	100	67	67	–	–	100	100
RF00151	100	100	100	100	100	100	–	–	100	100
RF00153	0	17	0	0	0	0	–	–	0	0
RF00154	100	100	100	100	100	100	–	–	100	100
RF00157	0	0	0	0	0	0	–	–	75	75
RF00158	40	40	100	100	0	0	–	–	60	60
RF00173	62	88	100	100	62	88	–	–	62	79
RF00184	100	100	100	100	100	100	100	100	100	100
RF00186	75	75	58	58	58	58	58	58	100	100
RF00187	100	100	45	45	45	45	73	73	100	100
RF00188	42	58	67	67	67	89	42	53	50	50
RF00196	100	100	100	100	100	100	92	92	75	75
RF00197	81	85	88	96	88	96	88	96	88	96
RF00207	100	100	94	94	94	94	100	100	100	100

	MuLoRA		Marna		ClustalW		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00208	100	100	89	89	89	89	–	–	100	100
RF00217	100	100	100	100	100	100	100	100	100	100
RF00219	100	100	100	100	90	90	–	–	50	100
RF00249	100	100	95	99	77	94	50	90	100	100
RF00253	87	87	78	78	83	85	78	80	70	70
RF00254	100	100	100	100	100	100	100	100	100	100
RF00255	100	100	96	96	100	100	93	93	89	89
RF00257	54	61	82	86	57	62	–	–	82	92
RF00258	81	94	62	79	73	91	69	90	73	86
RF00262	83	83	75	75	75	75	75	75	83	83
RF00266	100	100	100	100	100	100	–	–	100	100
RF00267	41	41	65	69	65	69	–	–	71	71
RF00268	100	100	0	0	0	0	–	–	50	50
RF00270	100	100	100	100	100	100	–	–	100	100
RF00271	0	30	100	100	33	33	–	–	0	30
RF00273	10	10	0	0	0	0	–	–	10	10
RF00274	29	33	0	0	0	33	–	–	0	10
RF00276	0	6	0	0	0	0	–	–	0	0
RF00277	0	0	67	67	0	0	–	–	0	0
RF00280	75	75	83	83	75	75	–	–	50	50
RF00281	50	50	0	0	0	0	–	–	0	0
RF00282	14	14	0	0	0	0	–	–	14	14
RF00283	0	52	33	35	67	67	–	–	–	–
RF00284	50	58	50	50	50	50	–	–	70	70
RF00285	38	54	56	56	44	44	–	–	50	50
RF00287	29	29	0	0	0	0	–	–	14	14
RF00288	100	100	86	86	86	86	86	86	86	86
RF00289	67	67	67	67	0	0	–	–	60	60
RF00292	0	0	0	0	7	7	–	–	0	0
RF00293	17	17	0	0	0	0	–	–	17	17
RF00297	45	45	36	36	91	91	–	–	36	36
RF00299	0	0	0	0	0	0	0	0	36	36
RF00301	38	38	62	62	25	29	–	–	0	33
RF00310	64	64	45	45	55	61	–	–	45	48
RF00317	100	100	80	80	80	80	–	–	100	100
RF00318	100	100	100	100	100	100	100	100	100	100
RF00320	73	91	64	64	18	18	–	–	73	91
RF00323	100	100	100	100	40	40	–	–	60	60
RF00325	100	100	0	0	50	50	–	–	0	0
RF00329	100	100	100	100	100	100	–	–	100	100
RF00331	100	100	83	83	83	94	–	–	67	67
RF00332	100	100	100	100	70	90	–	–	0	70
RF00339	40	40	40	40	70	80	–	–	40	40
RF00343	0	100	86	90	71	86	–	–	0	81
RF00346	75	75	0	0	0	50	–	–	–	–
RF00347	0	0	0	0	0	0	–	–	0	0
RF00351	62	79	62	62	62	62	–	–	88	88
RF00352	100	100	83	83	100	100	–	–	83	83
RF00355	18	18	0	0	0	0	–	–	0	0
RF00358	100	100	43	43	43	43	–	–	100	100
RF00359	44	92	33	78	33	50	–	–	33	75
RF00361	87	87	93	93	53	58	27	58	87	87

	MuLoRA		Marna		ClustalW		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00363	96	96	79	79	88	92	79	81	92	92
RF00365	100	100	100	100	100	100	100	100	100	100
RF00366	100	100	100	100	100	100	100	100	84	84
RF00370	79	79	71	76	79	79	79	79	79	79
RF00382	82	94	73	91	73	91	–	–	73	73
RF00390	100	100	0	0	100	100	–	–	0	0
RF00439	77	77	77	77	46	46	–	–	77	77
RF00440	100	100	0	0	100	100	–	–	100	100
RF00454	0	0	0	0	0	0	–	–	0	0
RF00455	100	100	100	100	100	100	100	100	91	91
RF00466	69	69	59	60	69	69	62	66	69	69
RF00493	100	100	100	100	100	100	91	94	100	100
RF00494	69	69	62	62	62	62	–	–	62	62
RF00498	100	100	91	91	91	91	82	86	100	100
RF00500	100	100	92	92	100	100	92	92	92	92
RF00505	0	0	0	0	0	0	–	–	0	0
Ø	66	72	61	63	59	63	26	29	60	65

Tabelle A.2: Vergleich der Alignmentergebnisse über konservierten Sequenzen für die einzelnen Programme

A.2 Unkonservierte Sequenzen

Vergleich von verschiedenen p_{min}:

	0,01		0,25		0,50		0,75		1.00	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00005	19	60	19	65	19	44	19	43	10	37
RF00006	85	92	70	89	80	88	80	88	85	92
RF00008	100	100	100	100	100	100	100	100	13	71
RF00019	0	68	100	100	100	100	100	100	40	70
RF00027	87	97	68	84	68	81	87	97	68	80
RF00031	14	45	14	45	14	45	0	36	0	34
RF00033	88	92	88	92	88	92	88	92	88	92
RF00034	71	73	96	96	96	98	100	100	29	50
RF00049	0	65	0	15	0	25	0	60	0	25
RF00054	0	50	0	0	0	0	0	0	0	0
RF00055	0	75	0	75	0	75	0	75	0	50
RF00057	55	55	55	55	55	55	55	55	55	55
RF00069	67	67	67	67	67	67	67	67	0	50
RF00070	50	50	0	25	33	33	50	50	50	50
RF00073	96	96	96	96	92	94	92	94	92	94
RF00075	100	100	100	100	100	100	100	100	100	100
RF00093	0	0	0	6	0	0	0	50	0	0
RF00109	0	3	0	11	0	1	0	18	0	18
RF00163	0	43	0	57	43	54	43	57	0	21
RF00181	80	95	0	75	0	60	0	50	0	60
RF00204	100	100	100	100	100	100	100	100	100	100
RF00213	0	93	0	96	0	68	0	46	0	21
RF00218	0	17	0	21	0	17	0	21	0	0
RF00221	0	0	0	25	0	13	0	0	0	0
RF00227	0	28	0	28	0	28	0	28	0	28
RF00309	0	27	0	9	0	14	0	14	0	0
RF00335	0	11	43	57	0	14	0	11	43	43
RF00345	100	100	100	100	100	100	100	100	75	92
RF00353	54	54	0	33	0	36	0	36	54	53
RF00506	51	64	51	64	51	73	51	64	51	64
Ø	41	61	39	60	40	56	41	58	32	48

Tabelle A.3: Vergleich der Alignmentergebnisse über unkonservierten Sequenzen für verschiedene Werte p_{min}

Vergleich der einzelnen Programme:

	MuLoRA		Marna		ClustalW		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00005	19	60	57	69	14	48	–	–	71	71
RF00006	85	92	0	42	55	72	–	–	–	–
RF00008	100	100	0	62	33	96	–	–	–	–
RF00019	0	68	0	23	0	75	–	–	–	–
RF00027	87	97	87	97	68	84	0	82	68	77
RF00031	14	45	0	14	0	34	–	–	0	33
RF00033	88	92	88	88	88	92	–	–	81	86

	MuLoRA		Marna		ClustalW		RNAforester		pmmulti	
	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}	s_{col}	s_{bp}
RF00034	71	75	50	62	29	75	–	–	25	25
RF00049	0	65	0	0	0	0	–	–	0	60
RF00054	0	50	0	0	0	0	–	–	0	0
RF00055	0	75	0	0	0	50	–	–	0	60
RF00057	55	55	55	55	55	55	–	–	55	55
RF00069	67	67	67	67	0	0	–	–	67	67
RF00070	50	50	0	0	17	17	–	–	–	–
RF00073	96	96	96	96	83	94	96	96	42	90
RF00075	100	100	0	34	100	100	–	–	–	–
RF00093	0	0	0	0	0	0	–	–	–	–
RF00109	0	3	0	0	0	0	–	–	0	14
RF00163	43	43	0	0	0	29	–	–	–	–
RF00181	80	95	0	0	0	60	–	–	0	40
RF00204	100	100	100	100	40	80	–	–	100	100
RF00213	0	93	0	0	29	82	–	–	43	43
RF00218	0	17	0	0	0	17	–	–	–	–
RF00221	0	0	0	0	0	6	–	–	–	–
RF00227	0	28	0	0	0	39	–	–	0	22
RF00309	0	27	0	0	0	0	–	–	–	–
RF00335	43	11	0	0	0	43	–	–	0	32
RF00345	100	100	0	0	75	92	–	–	0	67
RF00353	54	54	54	54	0	38	–	–	0	31
RF00506	51	64	41	41	51	76	–	–	–	–
Ø	41	61	23	30	25	48	3	6	18	32

Tabelle A.4: Vergleich der Alignmentergebnisse über konservierten Sequenzen für die einzelnen Programme

Literaturverzeichnis

[AGM90] Altschul, S. F., Gish, W., Miller, W., Myers, E.W., Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.*, **215** (3), 403–410.

[BW04] Backofen, R., Will, S. (2004). Local Sequence Structure Motifs in RNA. *Journal of Bioinformatics and Computational Biology (JBCB)*, **2** (4), 681–698.

[Bar04] Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, **116**, 281–297.

[Bro00] Brown, M. (2000). Small subunit ribosomal RNA modeling using stochastic context-free grammars. *Proc. 8th Int. Conf. Intelligent Systems for Molecular Biology (ISMB'00)*, 57–66.

[Cou02] Couzin, J. (2002). Breakthrough of the year: Small RNAs make big splash. *Science*, **298**, 2296–2297.

[CLZ03] Crochemore, M., Landau, G.M., Ziv-Ukelson M. (2003). A Sub-quadratic Sequence Alignment Algorithm for Unrestricted Cost Matrices. *SIAM J. Comput*, **32** (5), 1654–1673.

[Edd01] Eddy, S. R. (2001). Non-Coding RNA Genes and the Modern RNA World. *Nature Reviews Genetics*, **2** (12), 919–929.

[FW05] Fedor, M. J., Williamson, J. R. (2005). The catalytic diversity of RNAs. *Nature Rev. Mol. Cell. Biol.*, **6**, 399–412.

[FD87] Feng, D.-F., Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, **25**, 351–360.

[Got82] Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

[GMM05] Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R., Bateman, A. (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, **33**, D121–D124.

[HK96] Hentze, M., Kuehn, L. (1996). Molecular control of vertebrate iron metabolism: mRNA-based regulatory circuits operated by iron, nitric oxide, and oxidative stress. *Proc. Natl. Acad. Sci. USA*, **93**, 8175–8182.

[HHS90] Hertz, G. Z., Hartzell III, G. W., Stormo, G. D. (1990). Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computer Applications in the Biosciences*, **8**, 189–191.

[HTG03] Höchsmann, M., Töller, T., Giegerich, R., Kurtz, S. (2003). Local Similarity in RNA Secondary Structures. *CSB*, **6**, 7695–7704.

[Hof03] Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Research*, **31** (13), 3429–3431.

[HBS04] Hofacker, I. L., Bernhart, S. H. F. Stadler, P. F. (2004). Alignment of RNA Base Pairing Probability Matrices. *Bioinformatics*, **20**, 2222–2227.

[JLM02] Jiang, T., Lin, G., Ma, B., Zhang, K. (2002). A general edit distance between RNA structures. *J. of Computational Biology*, **9** (2), 371–388.

[JWZ95] Jiang, T., Wang, J., Zhang, K. (1995). Alignment of trees – an alternative to tree edit. *Theoretical Computer Science*, **143** (1), 137–148.

[Jus01] Just, W (2001). Computational Complexity of Multiple Sequence Alignment with SP-Score. *Journal of Computational Biology*, **8** (6), 615–623.

[KH05] Komar, A. A., Hatzoglou, M. (2005). Internal Ribosome Entry Sites in Cellular mRNAs: Mystery of Their Existence. *J. Biol. Chem.*, **280** (25), 23425–23428.

[KBM94] Krogh, A., Brown, M., Mian, I. S., Sjölander, K. Haussler, D. (1994). Hidden Markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, **235**, 1501–1531.

[Lat94] Lathrop, R.H. (1994). The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.*, **7**, 1059–1068.

[MSZ99] Mathews, D., Sabina, J., Zucker, M., Turner, H. (1999). Expanded sequence dependence of thermodynamic parameters provides robust prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

[McC90] McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

[NW70] Needleman, S. B., Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**, 443–453.

[NHB01] Nocker, A., Hausherr, T., Balsiger, S., Krstulovic, N.P., Hennecke, H., Narberhaus, F. (2001). A mRNA-based thermosensor controls expression of rhizobial heat shock genes. *Nucleic Acids Res.*, **29**, 4800–4807.

[NHH00] Notredame, C., Higgins, D. G., Heringa, J. (2000). T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. *J. Mol. Biol.*, **302**, 205–217.

[NPG78] Nussinov, R., Pieczenik, G., Griggs, J. R., Kleitman, D. J. (1978). Algorithms for loop matchings. *SIAM Journal of Applied Mathematics*, **35**, 68–82.

[PFM94] Pley, H. W., Flaherty, K. M., McKay, D. B. (1994). Three-dimensional structure of a hammerhead ribozyme. *Nature*, **372**, 68–74.

[SN87] Saitou, N., Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–125.

[SBH94] Sakakibara, Y., Brown, M., Hughey, R., Mian, I., Der, K.S., Underwood, R., Haussler, D. (1994). Recent methods for RNA modeling using stochastic context-free grammars. Proc. 5th Ann. Symposium on Combinatorial Pattern. *Matching (CPM'94)*, **LNCS 807**, 289–306.

[San85] Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

[SZ90] Shapiro, B., Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparisons. *Computer Appl. Biosci.*, **6**, 309–318.

[SWZ94] Shasha, D., Wang, J., Zhang, K., Shih, F. (1994). Exact and approximate algorithms for unordered tree matching. *IEEE Trans. Systems, Man, and Cybernetics*, **24**, 668–678.

[SB03] Siebert, S., Backofen, R. (2003). MARNA: A server for multiple alignment of RNAs. *Proceedings of the German Conference on Bioinformatics (GCB2003)*, **1**, 135–140.

[SW81] Smith, T. F., Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195–197.

[Sto02] Storz, G. (2002). An Expanding Universe of Noncoding RNAs. *Science*, **296** (5571), 1260–1263.

[Sto98] Stoy, J. (1998). Multiple Sequence Alignment with the Divide-and-Conquer Method. *Gene*, **211**, GC45–GC56.

[THG94] Thompson, J. D., Higgs, D. G., Gibson, T. J. (1994). CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties, and weight matrix choice. *Nucl. Acids Res.*, **22**, 4673–4680.

[WSP97] Wilting, R., Schorling, S., Persson, B. C., Böck, A. (1997). Selenoprotein synthesis in archaea: Identification of an mRNA element of Methanococcus jannaschii probably directing selenocysteine insertion. *J. Mol. Biol.*, **266** (4), 637–641.

[WCB02] Winkler, W., Cohen-Chalamish, S., Breaker, R.R. (2002). An mRNA structure that controls gene expression by binding FMN. *Proc. Nat. Acad. Sci. U.S.A.*, **99**, 15908–15913.

[Zha96a] Zhang, K. (1996). A constrained edit distance between unordered labeled trees. *Algorithmica*, **15**, 205–222.

[Zha96b] Zhang, K. (1996). Efficient parallel algorithms for tree editing problems. *Proc. 7th Ann. Symposium on Combinatorial Pattern Matching (CPM'96)*, **LNCS 1075**, 361–372.

[ZWM99] Zhang, K., Wang, L., Ma, B. (1999). Computing similarity between RNA structures. *Proc. 10th Ann. Symposium on Combinatorial Pattern Matching (CPM'99)*, **LNCS 1645**, 281–293.

[ZS81] Zuker, M., Stiegler, P. (1981). Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig und ohne Verwendung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe.

Jena, den 13. Juni 2006

(Wolfgang Otto)