RNA-Alignments und RNA-Struktur *in silico*

I n a u g u r a l - D i s s e r t a t i o n
zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf
vorgelegt von

Andreas Wilm
aus Düsseldorf

Januar 2006

Aus dem Institut für Physikalische Biologie
der Heinrich-Heine Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: apl. Prof. Dr. G. Steger

Korreferent: Univ.-Prof. Dr. R. Wagner

Drittgutachter: Univ.-Prof. Dr. R. Giegerich

Tag der mündlichen Prüfung: 2. Mai 2006

Meinen Homies

Danksagung

Mein Dank gilt natürlich in erster Linie meinem Doktorvater Herrn apl. Prof. Dr. Gerhard Steger, der früh die Idee einer „BRAliBase“ interessant fand. Er hat mir jederzeit die wissenschaftliche Freiheit gegeben, die zur Vollendung dieser Arbeit nötig war und Rückschläge gelassen hingenommen.

Herrn Prof. Dr. Rolf Wagner danke ich dafür, dass er sich trotz des nicht gerade heiß-geliebten Themas bereit erklärte, diese Arbeit als Korreferent zu beurteilen.

Der Studienstiftung des deutschen Volkes bin ich für die großzügige Förderung zu ganz erheblichem Dank verpflichtet.

Ausdrücklicher Dank gilt selbstredend meinen Eltern für jede Art von Unterstützung die sie mir haben zukommen lassen.

Ich danke der kompletten Rechnergruppe, insbesondere Indra Mainz und Deniz Dalli, da sie die Endphase-Katalysatoren dieser Arbeit waren.

Ein besonderer Dank gilt Dr. Paul Gardner, dessen Geduld und Diskussionsbereitschaft zum Gelingen dieser Arbeit entscheidend beigetragen hat. Thank you very much, Paul!

Man verzeihe mir, dass ich auf eine namentliche Erwähnung (auch ehemaliger) Mitarbeiter des Instituts verzichte, die mir nicht nur fachlich, sondern auch persönlich zu Seite gestanden haben: ich danke Euch allen zutiefst und von Herzen. Es mag Wehmut sein, aber gerade das letzte Jahr mit Euch war super und ich danke Euch allen für die einmalige Atmosphäre innerhalb und außerhalb des Instituts.

Weiterhin danke ich dem China Restaurant Hongkong für die Verpflegung in der heißen Schreib-Phase und allen die ich vergessen habe.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1

Einleitung

Die Struktur und Funktion von Nukleinsäuren ist seit über 50 Jahren Gegenstand der Forschung. Nachdem Crick, Watson & Wilkins für die Aufklärung der Doppelhelix-Struktur der Desoxyribonukleinsäure („Deoxyribonucleic Acid“; DNA) den Nobelpreis erhielten, wurde von Crick (1958) das sogenannte zentrale Dogma der Molekularbiologie aufgestellt (siehe auch Crick, 1970). Dieses besagt, dass die in der DNA gespeicherte genetische Information mit Hilfe von Ribonukleinsäuren („Ribonucleic Acid“; RNA) weitergeleitet und schließlich in Proteine übersetzt wird. In diesem Bild übernimmt die DNA die Rolle des reinen Informationsspeichers, die RNA ist passiver Informationsvermittler und die Proteine sind alleiniger Funktionsträger und katalysieren die chemischen Reaktionen in der Zelle. Weiterhin erlaubt dieses Dogma nur einen unidirektionalen Informationsfluss.

Dass dies eine Vereinfachung ist, wurde bereits 1970 deutlich, als die Reverse Transkriptase (RNA-abhängige DNA Polymerase) entdeckt wurde (Nobelpreis 1975; Baltimore, 1970; Temin, 1970), die es Retroviren – wie beispielsweise HIV – ermöglicht, ihr in Form von RNA vorliegendes Genom in das wirtseigene (DNA-)Genom zu integrieren. Schließlich wurden durch Altman & Cech RNAs entdeckt, die intrinsisch katalytische Fähigkeiten besitzen (RNase P und selbstspleißende Introns; Nobelpreis 1989). Diese wurden in Anlehnung an das Wort Enzyme Ribozyme genannt. Die Tatsache, dass RNA damit nicht nur Informationsträger ist, sondern auch chemische Reaktionen katalysieren kann, führte zu der Idee der RNA-Welt (siehe beispielsweise Gilbert, 1986). Diese versucht mit der RNA, als dem ersten autark replizierenden Molekül mit katalytischen Fähigkeiten, den Ursprung des Lebens zu erklären. Schließlich konnte gezeigt werden, dass auch das Ribosom, also der Komplex aus Proteinen und rRNAs, welcher für die Proteinbiosynthese zuständig ist, selbst ein Ribozym ist, da die hier entscheidende Peptidyltransferase-Aktivität einer RNA zukommt (Nissen *et al.*, 2000).

Mittlerweile ist neben den an der Proteinbiosynthese maßgeblich beteiligten (klassischen) RNAs, der transfer-RNA (tRNA) und der ribosomalen RNA (rRNA), eine große Zahl weiterer sogenannter nicht-Protein-kodierender RNAs („non-protein-coding RNA“; ncRNA) gefunden worden, die autonom eine Vielzahl von Funktionen übernehmen (siehe beispielweise Vogel &

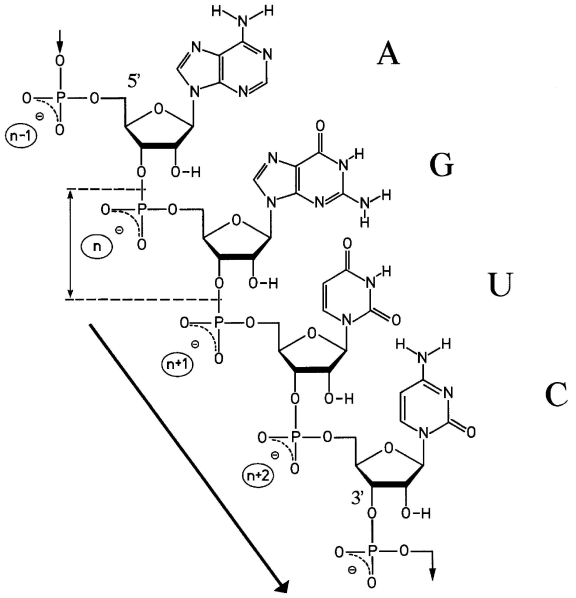


Abbildung 1.1: Bausteine der RNA. RNA hat die folgenden vier Nukleotide als Bausteine: Adenin (A), Guanin (G), Cytosin (C) und Uracil (U). Jedes der Nukleotide besteht aus einer (namensgebenden) Base, einer Ribose und einem Phosphat. Die Sequenz ist polar aufgebaut: sie hat ein 5'-Ende (Phosphat am C5 der Ribose) und ein 3'-Ende (Phosphat am C3 der Ribose). Die Sequenz (Primärstruktur) wird immer in 5'-3'-Richtung angegeben; die hier gezeigte Sequenz lautet AGUC. Nach Steger (2003).

Sharma, 2005, für einen Überblick über bakterielle ncRNAs). Die Entdeckung des Phänomens der RNA-Interferenz (RNAi), in dem sehr kleine RNAs posttranskriptional die Genexpression kontrollieren, wurde von der Zeitschrift Science zum Durchbruch des Jahres 2002 gewählt (Couzin, 2002). In den letzten Jahren wurden regelmäßig neue ncRNA-Klassen entdeckt, wie die sogenannten Riboswitches, strukturelle mRNA-Elemente, die Metaboliten binden können und so die Translation oder Transkription der eigenen mRNA kontrollieren (siehe beispielsweise Mandal & Breaker, 2004; Tucker & Breaker, 2005; Winkler *et al.*, 2004).

Die Funktion der ncRNAs ist in allen Fällen durch ihre dreidimensionale Struktur bestimmt (hier: „Function Follows Form"), welche wiederum in der Sequenz kodiert ist.

1.1 RNA-Struktur

Das Polymer RNA besteht aus einer kovalent-verknüpften Kette der Nukleotide Adenin (A), Guanin (G), Cytosin (C) und Uracil (U) (siehe Abbildung 1.1). Sie unterscheidet sich damit von der DNA zum einen durch das Nukleotid Uracil anstatt Thymin (T) und zum anderen durch die Ribose anstatt einer 2'-Desoxyribose. Die Sequenz, also die Abfolge von Nukleotiden in 5'-3'-Richtung wird auch Primärstruktur genannt.

Obwohl RNA im Gegensatz zur DNA meist einzelsträngig vorliegt, kann sie höhere Strukturen bilden (man spricht von Faltung). Die Grundlage hierfür bilden Basenpaare, die durch Wasserstoffbrücken komplementärer Basen und vor allem Stapelwechselwirkungen („Stacking"; Dipol-induzierte-Dipol-Wechselwirkung) benachbarter Basen energetisch favorisiert sind. Die Sekundärstruktur ist eine Liste von Basenpaaren, die durch Paarung von Nukleotiden mit ihrem jeweiligen Komplement entsteht. Dabei werden die Basenpaare A : U und G : C (*vice versa*) Watson-Crick- oder auch kanonische Basenpaare genannt. Bei G : U bzw. U : G spricht man vom Wobble-Basenpaar. Die einfachste Sekundärstruktur entsteht bei komplementären 5'- und 3'-Enden durch Rückfaltung der RNA auf sich selbst. Es bildet sich ein sogenannter Hairpin,

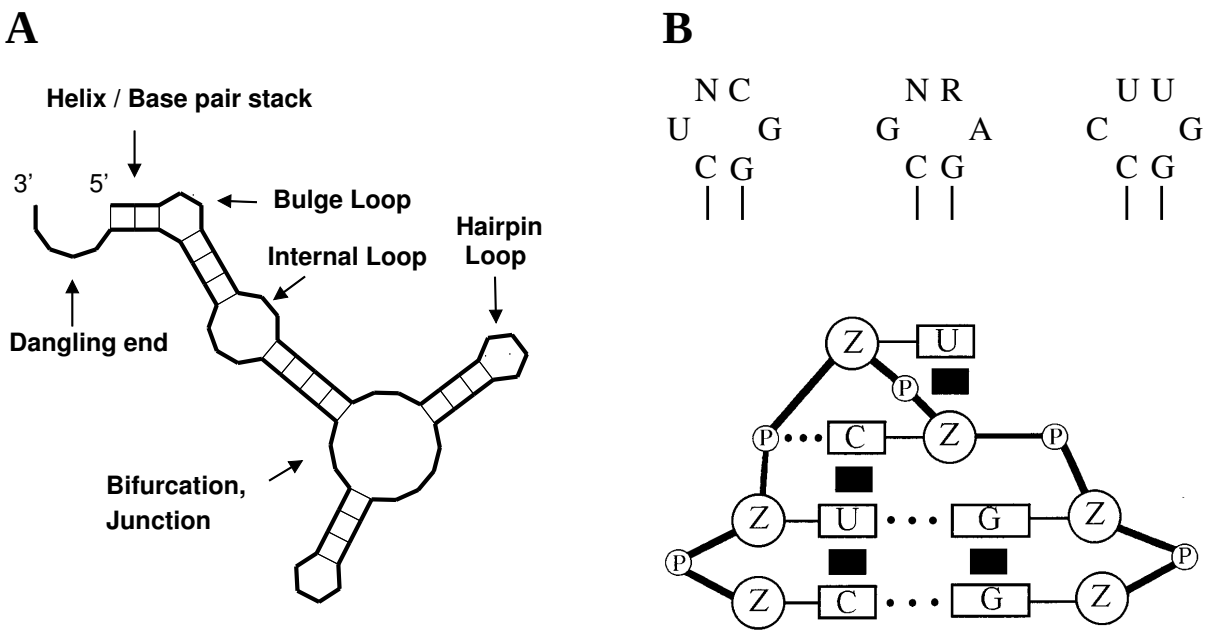


Abbildung 1.2: Sekundärstrukturelemente einer RNA. In **A** sind verschiedene Loop-Typen und weitere Sekundärstrukturelemente gezeigt. Normalerweise ist die Loopbildung thermodynamisch ungünstig. Die sogenannten extrastabilen Tetraloops (**B**) sind ein Sonderfall. Oben sind drei Typen dieser Loops gezeigt. Unten sind die Besonderheiten der Tetraloops beispielhaft an einem UNCG-Tetraloop dargestellt. Die Basen zeigen nicht nach „außen“ (ins Medium), sondern können weiterhin Stapelwechselwirkungen ausüben, im Falle vom Uracil (U) sogar mit einer Ribose (hier Z für Zucker; siehe Allain & Varani, 1995, für die exakte Struktur). Nach Steger (2003).

der durch einen gepaarten Bereich (Helix) und einen ungepaarten Bereich (Hairpin Loop) gebildet wird. In Abbildung 1.2 sind dieser und weitere Loop-Typen gezeigt. Die Bildung von Loops ist im allgemeinen thermodynamisch ungünstig ($\Delta G > 0$). Einen energetisch weniger ungünstigen Fall stellen die extrastabilen Tetraloops dar, da diese durch ihre besondere Konformation weiterhin Stapelwechselwirkungen ausbilden können (siehe B in Abbildung 1.2).

Hat sich die Sekundärstruktur einmal ausgebildet, so können sich tertiäre Kontakte bzw. die Tertiärstruktur bilden. Ein Beispiel ist in C und D der Abbildung 1.3 zu sehen. Einfache Tertiärstrukturelemente sind Loop-Loop-Wechselwirkungen (wie im gezeigten Beispiel), Basentripel, also die Wechselwirkung zwischen drei Basen, sowie Pseudoknoten. Die Tertiärstruktur beschreibt die räumliche Anordnung, sprich die 3D-Struktur des Moleküls.

1.2 Das Alignment-Problem

Das Alignment von zwei oder mehr Sequenzen (oder auch Zeichenketten) ist Gegenstand von 40 Jahren Forschung (Levenshtein, 1966) und so gibt es mittlerweile eine enorme Anzahl von entsprechenden Publikationen und Programmen, jedoch bleiben die zur Verfügung stehenden Lösungen (aus später genannten Gründen) suboptimal. Aufgrund der Fülle der zur Verfügung stehenden Ansätze und Techniken werden hier nur beispielhaft einige Überlegungen dargestellt.

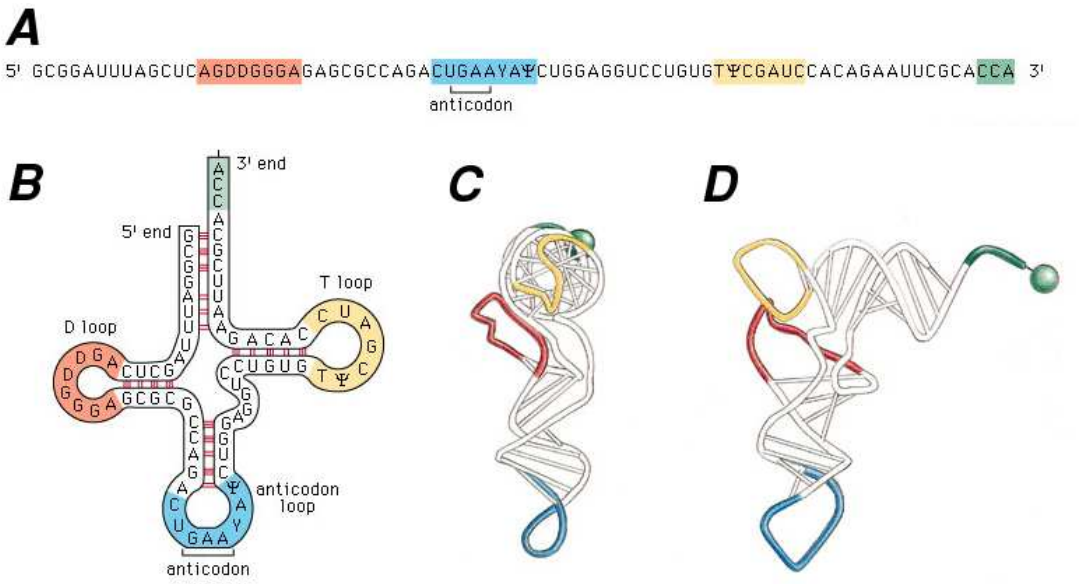


Abbildung 1.3: Struktur-Ordnungen am Beispiel einer tRNA. Gezeigt sind Primär- (**A**), Sekundär- (**B**) und Tertiärstruktur (**C** und **D**) am Beispiel einer $tRNA^{Phe}$. Einzelne Domänen sind in den Darstellungen hervorgehoben: D-Loop (rot), Anticodon (blau), T-Loop (gelb) und der Akzeptor-Arm (grün). **A** zeigt die Sequenz der tRNA. Ψ und D sind die durch posttranskriptionelle Modifikation von Uracil entstandenen Basen, Pseudouridin und Dihydrouridin, welche der Stabilisierung der Struktur dienen. Die Sekundärstruktur (**B**) von tRNAs wird auch als Kleeblattstruktur bezeichnet. Basenpaare sind dort als rote Verbindungslinien dargestellt. In den 3D-Darstellungen (**C** und **D**) sieht man, dass das Molekül eine L-Form besitzt und dass D- und T-Loop sich räumlich sehr nahe sind. Nach Alberts (1998).

Für detaillierte Ausführungen sei auf Gotoh (1999) oder Lehrbücher, wie Durbin *et al.* (1998); Gusfield (1999); Steger (2003) verwiesen.

„Alignment“ umschreibt eigentlich eine Gruppe von Problemen, deren exakte Definition nur je nach Fragestellung gegeben werden kann. Grundsätzlich handelt es sich bei einem Alignment um einen Sequenzvergleich. Im einfachen Fall des paarweisen Alignments werden sich zwei verwandte Sequenzen gegenübergestellt und durch das Einfügen von sogenannten Gap-Symbolen (üblicherweise „-“) in die Sequenzen versucht, die in der Evolution aufgetretenen Mutationen so auszugleichen, dass sich anschließend homologe Bereiche gegenüberstehen. Dabei wird davon ausgegangen, dass die Sequenzen entweder einen gemeinsamen Vorläufer hatten oder eine ähnliche Funktion haben. Idee dabei ist, dass für die biologische Funktion wichtige Regionen in ihrer Mutationsmöglichkeit zumindest teilweise eingeschränkt sind und damit konservierter als andere Regionen sind. Die Evolution der Sequenzen lässt sich dabei durch Operationen wie Substitutionen, Insertionen und Deletionen beschreiben. Bei großen Sequenzen kommt es zusätzlich beispielsweise zu Duplikationen.

Etwas formaler ausgedrückt ist ein paarweises Alignment eine Transformation einer Sequenz in eine andere, mit Hilfe einer Serie von Edit-Operationen, namentlich Match (Übereinstimmung), Substitution (Ersetzung), Deletion oder Insertion. In Abbildung 1.4 ist ein einfaches Beispiel gezeigt. Bei Deletion und Insertion spricht man auch oft von Indel, da diese in den wenigsten Fällen unterschieden werden.

```
Sequenz 1:     A C G C T G -
Sequenz 2:     - C A - T G T
Operationen:   D M S D M M I
```

Abbildung 1.4: Alignment-Operationen. Gezeigt ist ein einfaches Beispiel eines paarweisen Alignments zweier DNA-Sequenzen. Sind zwei ausgerichtete/alignierte Nukleotide identisch, handelt es sich um einen Match (**M**). Sind zwei ausgerichtete Nukleotide nicht identisch, spricht man von einer Substitution (**S**). Schließlich gibt es Lücken, die durch das Einfügen von Resten (Insertion; **I**) oder durch das Entfernen von Resten entstehen (Deletionen; **D**).

Da es offensichtlich sehr viele Möglichkeiten gibt, ein solches Alignment zu erstellen, muss für die Operationen ein sogenanntes Kosten-Modell („Scoring Model“) aufgestellt werden, welches beispielsweise übereinstimmende Reste („Matches“) begünstigt, Gaps und Substitutionen hingegen bestraft. Das einfachste Modell ist das sogenannte Einheitskostenmodell (auch Levenshtein-Distanz; Levenshtein, 1966), welches folgende Kosten/Gewichte w für zwei alignierte Reste a und b definiert:

$$
\begin{aligned}
w(a,b) &= 0 && \text{Match (a=b)} \\
w(a,b) &= 1 && \text{Substitution (a}\neq\text{b)} \\
w(a,-) &= 1 && \text{Deletion} \\
w(-,b) &= 1 && \text{Insertion}
\end{aligned}
$$

Somit ist nach einer optimalen (kostengünstigen) Anzahl Edit-Operationen gesucht, welche die Sequenzen aligniert. Auch wenn eine Lösung des in Abbildung 1.4 gezeigten Problems trivial erscheint, so ergibt sich eine mit der Sequenz-Länge exponentiell wachsende Anzahl von Lösungen (Durbin *et al.*, 1998). Die exakte Definition und Lösung des Problems hängt von der genauen Fragestellung ab. Ein formales Beispiel zur Lösung eines paarweisen Alignments wird im folgenden Abschnitt gegeben. Lösungsansätze für das weitaus komplexere multiple Alignment werden im Anschluss aufgezeigt.

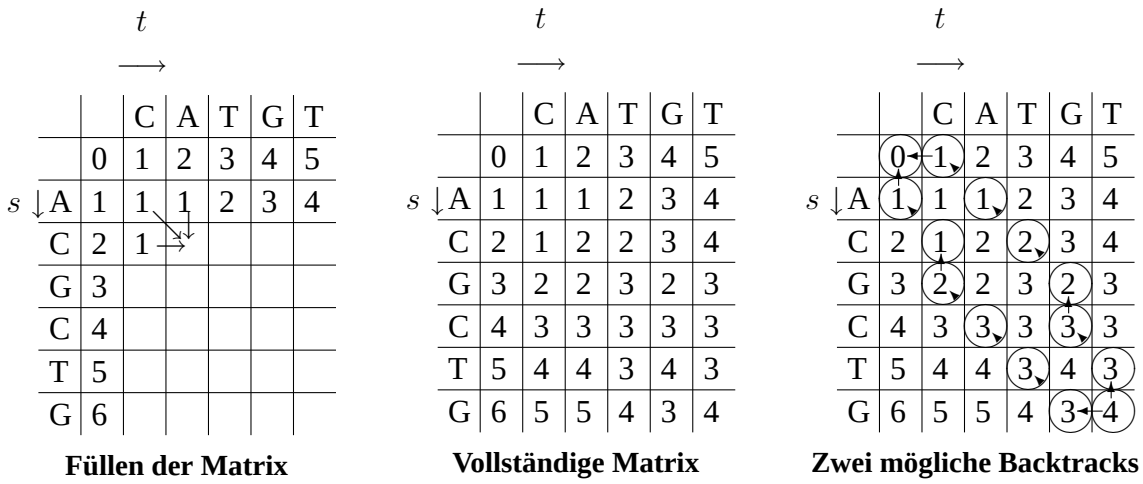
1.2.1 Paarweises Alignment

Für das paarweise Alignment, also das Alignment zweier Sequenzen, gibt es effiziente Algorithmen. Je nach Fragestellung unterscheidet man folgende Varianten: Will man lediglich ein Motiv (also eine kleine Sequenz oder eine Domäne) an eine größere Sequenz alignieren, spricht man von einem lokalen Alignment. Sollen zwei Sequenzen vergleichbarer Länge aligniert werden, spricht man von einem globalen Alignment. Weiterhin gibt es Varianten wie „lokale Ähnlichkeit“ oder „längste gemeinsame Subsequenz“.

Die Lösungsansätze bedienen sich alle der sogenannten „dynamischen Programmierung“ (siehe beispielsweise Eddy, 2004c, für eine schöne Beschreibung), welche eine Programmiertechnik ist, die eine Lösung für ein großes Problem bestimmt, indem sie zunächst die (gleichartigen) Teilprobleme löst. Die Gesamtlösung wird dann aus den tabellierten Teillösungen zusammengesetzt („Bottom-up“). Das Problem des globalen paarweisen Alignments wurde von Needleman

$$\text{Initialisierung:} \quad \begin{cases} d_{0,0} &= 0 \\ d_{i,0} &= d_{i-1,0} + w(i,-) \\ d_{0,j} &= d_{0,j-1} + w(-,j) \end{cases}$$

$$\text{Rekursion:} \quad d_{i,j} = \min \begin{cases} d_{i-1,j-1} + w(i,j), \\ d_{i-1,j} + w(i,-), \\ d_{i,j-1} + w(-,j) \end{cases}$$



„High Road“		„Low Road“	
Sequenz s:	- A C G C T G	Sequenz s:	A C G C T G -
Sequenz t:	C A T G - T -	Sequenz t:	- C - A T G T
Operationen:	I M S M D M D	Operationen:	D M D R M M I

Abbildung 1.5: Dynamische Programmierung und Backtrack beim globalen paarweisen Alignment. Das globale paarweise Alignment zweier Sequenzen s und t der Länge i bzw. j ist hier gezeigt. **Oben:** Initialisierung und Rekursionsformel für die dynamische Programmierung. **Mitte:** Mehrere Zustände der Distanz-Matrix. Die Bewertung wurde mit Hilfe des im Text erwähnten Einheitskostenmodells durchgeführt. **Unten:** zwei optimale Lösungen. Für Details siehe Text. Nach Steger (2003)

& Wunsch (1970) gelöst. Eine entsprechende Lösung für das lokale Alignment-Problem wurde durch Smith & Waterman (1981) beschrieben.

Hier sei beispielhaft die dynamische Programmierung für das globale Alignment zweier Sequenzen s und t mit Längen i und j unter Verwendung des erwähnten Einheitskostenmodells (Levenshtein-Distanz) beschrieben (siehe Abbildung 1.5). In diesem Fall stellt die Triviallösung der dynamischen Programmierung das Alignment zweier Zeichenketten der Länge 0 dar. Der entsprechende Wert $d_{0,0}$ wird in der Distanz-Matrix auf 0 initialisiert. Weiterhin wird beim globalen Alignment die erste Spalte und erste Reihe so vorbelegt, dass sich der Wert jeder Zelle aus dem Vorgängerwert zuzüglich der Kosten für eine Insertion bzw. Deletion ergibt. Wie anhand

der Rekursionsformel zu sehen, ergeben sich die restlichen Werte immer in Abhängigkeit von drei Nachbarn, wobei $d_{i-1,j-1} + w(i,j)$ einem Match oder einer Substitution entspricht. Die beiden anderen Alternativen entsprechen einer Deletion oder Insertion. Nachdem die Distanz-Matrix gefüllt ist, steht die optimale/minimale Edit-Distanz rechts unten in der Matrix ($d_{i,j}$). Um hieraus ein Alignment zu erstellen, muss ein entsprechender „Backtrack“ (auch „Traceback“) durchgeführt werden. Wie im Beispiel gezeigt, ergeben sich mehrere alternative, aber gleich gute Lösungen, je nachdem welchen Weg der Backtrack wählt.

Hier bleibt festzuhalten, dass die erwähnten Algorithmen (Smith & Waterman und Needleman & Wunsch) immer eine (mathematisch bzw. formal) optimale Lösung garantieren. Allerdings kann dies eine von vielen optimalen Lösungen sein. Zudem hängt die Lösung von den gewählten Bedingungen (Kosten-Modell) ab.

1.2.2 Multiples Alignment

Das Alignment mehrerer Sequenzen ist ungleich schwieriger, da sich hier in Abhängigkeit von der Sequenzzahl eine exponentielle Laufzeit ergibt. Formal ist das Problem NP-vollständig. So muss die dynamische Programmierung für das Alignment von drei Sequenzen den optimalen Weg in einem Kubus, statt einer 2D-Matrix (wie in Abbildung 1.5 gezeigt) finden. Bei k Sequenzen handelt es sich dann um einen k-dimensionalen Hyperkubus. Die Ausführung eines solchen Algorithmus ist schon bei wenigen Sequenzen nicht mehr praktikabel. Einige Ansätze versuchen durch Beschränkung des Suchraumes noch fast-optimale multiple Alignments zu berechnen, so beispielsweise DCA („Divide-and-Conquer Multiple Sequence Alignment “; Stoye, 1998). Jedoch ist in den meisten Fällen ein Einsatz von vereinfachenderen Heuristiken zwingend notwendig. Hier ist dann selbst unter Einsatz eines korrekten Kosten-Modells keine optimale Lösung mehr garantiert. Die bekannteste Heuristik ist das progressive Alignment, dessen Idee im folgenden Abschnitt beschrieben wird.

Progressives und iteratives Alignment

Die Idee des progressiven Alignments wurde mehrfach unabhängig entwickelt. Das häufigste Zitat ist jedoch Feng & Doolittle (1987), weshalb auch von der Feng & Doolittle-Methode gesprochen wird. CLUSTALW (siehe Abschnitt 3.1.2) ist das bekannteste Programm, welches diese Methodik implementiert. Die Idee beim progressiven Alignment ist, das multiple Alignment wieder in paarweise Alignments zu zerlegen. Dafür wird nach folgendem Schema vorgegangen.

1. Für jedes mögliche Sequenzpaar wird durch paarweises Alignment eine approximative evolutionäre Distanz bestimmt.
2. Mit Hilfe dieser Distanzen wird per per UPGMA, Neighbour-Joining (NJ) oder ähnlichen Cluster-Analyse-Methoden ein phylogenetischer Baum erstellt. Man erhält den sogenannten „Guide Tree“ (siehe A in Abbildung 1.6).
3. Die Sequenzen werden nun sukzessiv anhand der im Baum vorgegebenen Ordnung aligniert, wobei in jedem Schritt ein sogenanntes Profil entsteht (siehe B in Abbildung 1.6).

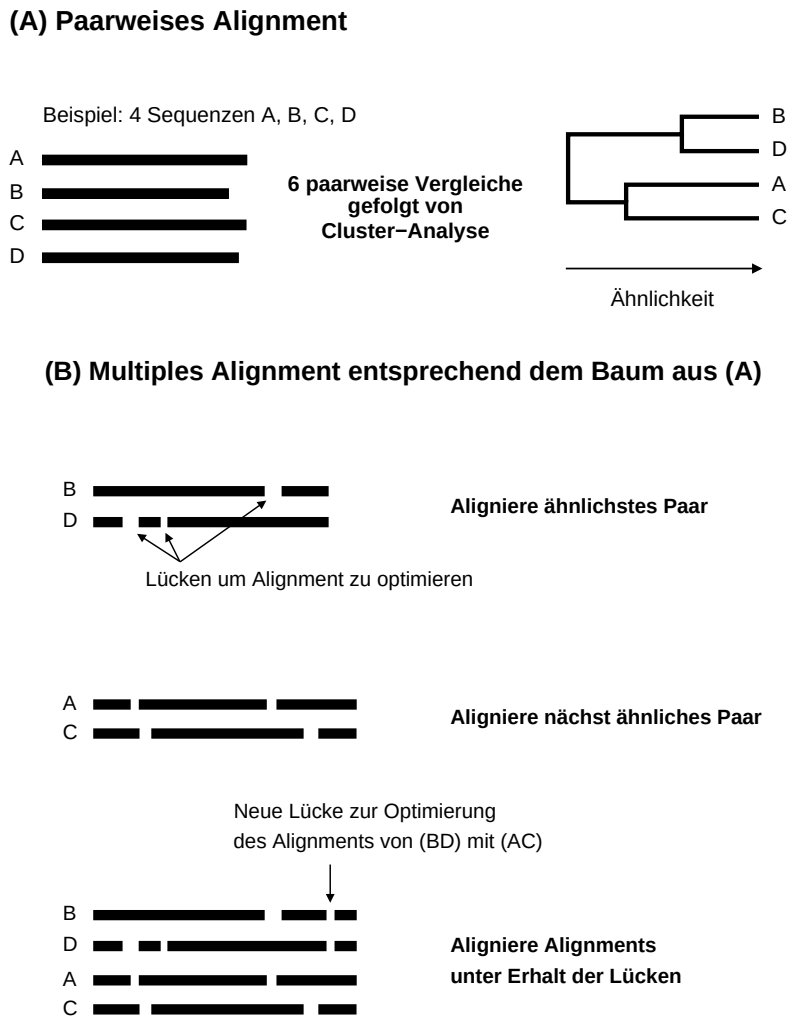


Abbildung 1.6: Vorgehen beim progressiven Alignment. A: Anhand aller paarweisen Distanzen der gezeigten vier Sequenzen wird ein Guide Tree erstellt. **B:** Den Verzweigungsordnungen des Baumes folgend wird je ein paarweises Alignment von Sequenz-an-Sequenz, Sequenz-an-Profil oder Profil-an-Profil erstellt, wobei jeweils ein neues Profil entsteht. Dies geschieht unter Beibehaltung der bereits eingefügten Gaps. Siehe auch Beschreibung im Text. Nach Steger (2003).

Das Alignment wird hier auf gierige („greedy“) Art und Weise durch paarweise Alignments von Sequenzen oder präalignierten Gruppen (Profilen) entlang der Verzweigungsordnung des Guide Trees erstellt. Gaps, die in frühen Phasen eingeführt wurden, müssen in den sukzessive folgenden Alignment-Schritten beibehalten werden (siehe A in Abbildung 1.6). Das stellt unter Umständen ein Problem dar, da frühe Misalignments (Gaps, die sich erst später als falsch eingefügt herausstellen) sich nicht mehr entfernen lassen („Once a gap always a gap“; siehe Abbildung 1.7).

Iterative Alignment-Methoden (siehe beispielsweise MUSCLE in Abschnitt 3.1.9) können Fehler korrigieren, indem sie, nachdem auf beliebige Art und Weise ein initiales Alignment erstellt wurde, dieses Alignment mit Hilfe verschiedener Techniken in zwei Subalignments (entsprechend zwei Profilen) aufteilen und hiermit ein erneutes Alignment durchführen. Sollte dieses Alignment je nach eingesetzter Bewertungsfunktion besser bewertet werden, so wird es

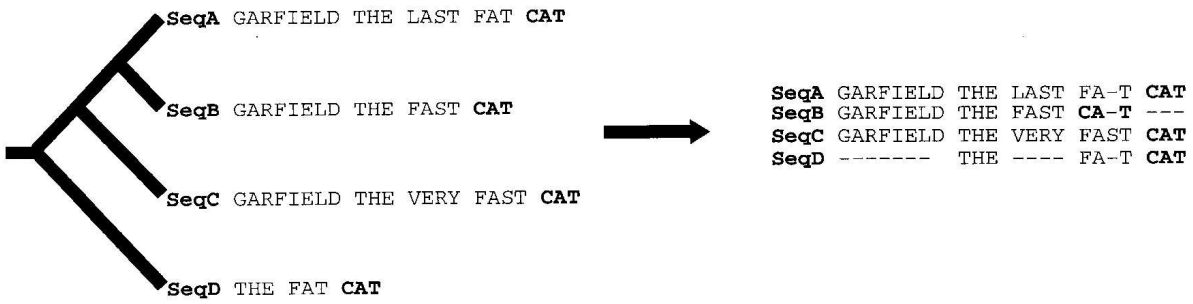


Abbildung 1.7: Fehler im progressiven Alignment. Gemäß dem gezeigten Guide-Tree (links) werden zunächst die Sequenzen SeqA und SeqB miteinander aligniert. Hier besteht die Möglichkeit (richtigerweise) CAT mit CAT zu alignieren und interne Gaps einzufügen, oder ein Mismatch zwischen C und F inkl. terminalen Gaps zu bilden. Da terminale Gaps in nahezu allen Bewertungsfunktionen weniger hart bestraft werden als interne, wird letztgenannte Variante bevorzugt (siehe rechts oben). Im nächsten Schritt wird durch Hinzufügen einer weiteren Sequenz (SeqC) klar, dass das Alignment der beiden CAT-Zeichenketten im vorherigen Schritt zu einer besseren Bewertung geführt hätte. Der Fehler pflanzt sich hier fort, da keine Korrektur wie in iterativen Methoden vorgenommen werden kann. Entnommen aus Notredame *et al.* (2000).

beibehalten und womöglich weiter verbessert; anderenfalls wird es verworfen. Für eine Diskussion verschiedener iterativer Techniken sei auf Wallace *et al.* (2005) verwiesen.

1.2.3 Bewertungsfunktionen

Grundsätzlich alignieren alle Programme Sequenzen so, dass die durch die jeweilige interne Bewertungsfunktion („Scoring Function“ oder „Objective Function“) resultierende Bewertung optimiert wird. Die Wahl dieser Bewertungsfunktion hat entscheidenden Einfluss auf die Qualität des Alignments. Für das paarweise Alignment ist die „Score“ dabei generell die Summe aller Edit-Kosten, die zwei Sequenzen ineinander überführen.

Im Falle von multiplen Alignments lassen sich verschiedene Bewertungsfunktionen definieren. Die am häufigsten eingesetzte Bewertungsfunktion ist die „Sum-of-Pairs“ (SOP; nicht zu verwechseln mit dem später vorgestellten Bewertungsmaß SPS), die alle paarweisen Alignments pro Spalte (u. U. gewichtet) bewertet. Bei einem Alignment mit N Sequenzen ergeben sich die Kosten S_c der Spalte c zu

$$S_c = \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_c(i,j) \; . \tag{1.1}$$

Hier ist $w_c(i,j)$ das anhand des Kostenmodells definierte Gewicht für die Reste in den entsprechenden Reihen i und j.

Eine alternative Funktion stellt COFFEE dar („**C**onsistency Based **O**bjective **F**unction **f**or Alignm**e**nt **E**valuation“; Notredame *et al.*, 1998). Diese Funktion versucht die Übereinstimmung mit einem Set paarweiser Alignments (auch Bibliothek genannt; siehe Abschnitt 3.1.18) zu maximieren. Es wird also versucht ein multiples Alignment zu erstellen, welches möglichst konsistent mit allen paarweisen Alignments einer vorher erstellten Bibliothek ist. Man spricht deshalb auch vom Konsistenz-basierten Alignment.

1.2.4 Gapkosten

Im erwähnten Einheitskostenmodell wurden Gaps einfach bestraft. Es ist jedoch meist sinnvoll zu unterscheiden, ob ein Gap neu eingefügt wurde oder benachbart zu einem bereits bestehenden Gap eingefügt wird. Das einfache Modell der linearen Gapkosten wurde deshalb erweitert durch sogenannte affine Gapkosten (ein effizienter Algorithmus findet sich in Gotoh, 1982). Hier wird zwischen Gap-Open (Einfügen eines neuen Gaps) und Gap-Extension-Kosten (Verlängerung eines Gaps) unterschieden. Die Kosten $\gamma(g)$ für das Einfügen von Gaps der Länge g ergeben sich dann folgendermaßen:

$$\gamma(g) = o + e \cdot (g - 1) \tag{1.2}$$

Hier ist o die sogenannte Gap-Open-Penalty und e die Gap-Extension-Penalty, wobei üblicherweise $o > e$ gewählt wird, so dass längere Insertionen und Deletionen weniger hart bestraft werden, als dies bei linearen Gapkosten der Fall wäre. Die entsprechenden Standard-Parameter der meisten Programme sind empirisch gesetzt.

1.2.5 Substitutionsmatrizen

Um zu entscheiden, wie die Substitution zweier Reste zu werten ist, werden sogenannte Substitutionsmatrizen eingesetzt. Die bekanntesten sind neben der Gonnet- die BLOSUM- und PAM-Matrizen.

Die BLOSUM-Matrizen („Blocks Substitution Matrix“ Eddy, 2004d; Henikoff & Henikoff, 1992) beruhen auf aus lokalen multiplen Alignments der BLOCKS-Datenbank extrahierten Werten, wohingegen die PAM-Matrizen („Percent Accepted Mutation“; Dayhoff *et al.*, 1978) auf globalen Alignments nah verwandter Proteine basieren. In beiden Fällen handelt es sich um Protein-spezifische Substitutionsmatrizen.

Für Nukleinsäuren werden meist einfache Werte verwendet. So enthält die sogenannte „DNA Identity Matrix (Unitary Matrix)“ Werte von 1 für einen Match und -10000 für einen Mismatch bzw. eine Substitution. Eine Alternative stellen die RIBOSUM-Matrizen (Klein & Eddy, 2003) dar, welche analog zu den BLOSUM-Matrizen anhand ribosomaler RNA-Alignments erstellt wurden. Diese und eine weitere Nukleinsäure-spezifische Matrix nach Gotoh (1999) werden in Abschnitt 3.7.7 eingeführt und verwendet.

1.2.6 Spezialfall RNA-Alignment

Das ncRNA-Alignment stellt in einiger Hinsicht eine besondere Herausforderung dar. ncRNAs sind in ihrer Struktur konservierter, als in ihrer Sequenz. So besitzen beispielsweise extrastabile Tetraloops (siehe Abbildung 1.2) keinerlei Sequenz-Ähnlichkeiten, sind aber homologe Elemente, die es zu alignieren gilt. Weiterhin evolvieren (nicht kodierende) RNA-Sequenzen in gepaarten Bereichen über sogenannte Struktur-neutrale Mutationen. Mutiert in einem gepaarten Bereich ein Nukleotid, so besteht aufgrund des wahrscheinlich drohenden Funktionsverlustes ein evolutionärer Druck diese Mutation auszugleichen, sprich die verlorene Basenpaarung zu

```
AACCAAAAAAGAGAA        AACCA-AAAAAGAGAA
..((.....).)..         ..((.-....).)..
AACUUAAAAAGAGAA        AA-CUUAAAAAGAGAA
..(.(...)).....        ..-(.(...)).....
```

Abbildung 1.8: Alignment der Punkt-Klammer-Notation als Sekundärstruktur-Darstellung. Links: Beispiel für ein Alignment, welches durch die Alignierung der Punkt-Klammer-Notation inkonsistent geworden ist. Sind zwei gepaarte Nukleotide aligniert, so müssen auch die entsprechenden Basenpaarungspartner aligniert sein, was hier nicht der Fall ist. Die korrekte Zuordnung der Klammern/Basenpaare geht bei einem „Sequenz-Alignment" der Punkt-Klammer-Notation verloren. **Rechts** ist eine mögliche Korrektur gezeigt. Nach Gardner & Giegerich (2004).

kompensieren. Dies kann durch eine Rück-Mutation des betroffenen Nukleotids geschehen, aber auch durch eine Mutation des Basenpaarungspartners. Diese sogenannten kompensatorischen Basenpaaraustausche haben zur Folge, dass basengepaarte Bereiche oft geringe Sequenz-Homologien aufweisen. Ein Alignment mit Hilfe traditioneller Sequenz-Alignment-Programme, die keinerlei Struktur-Informationen berücksichtigen, ist deshalb oft problematisch. Mit Sankoff (1985) existiert zwar ein Algorithmus für die simultane Lösung von Strukturvorhersage und Alignment, jedoch ist dessen Laufzeit und Speicherbedarf exponentiell von der Anzahl der Sequenzen abhängig.

Eine denkbare Möglichkeit wäre es, zunächst die Struktur jeder Sequenz vorherzusagen (siehe Abschnitt 1.4) und dann die entstehende Sekundärstruktur-Repräsentationen mit Hilfe von Sequenz-Alignment-Programmen zu alignieren. Als Repräsentationen bietet sich beispielsweise die Punkt-Klammer-Notation an, in der ungepaarte Bereiche mit Punkten und gepaarte Bereiche mit Klammern beschrieben werden, wobei jeder öffnenden Klammer eine korrespondierende schließende Klammer zugeordnet ist. Jedoch wird ein entsprechend modifiziertes Sequenz-Alignment schnell inkonsistent, da die Zuordnung von (alignierten) Klammern zu ihrem Gegenstück auch für die entsprechend alignierten Basenpaare gelten muss (siehe Abbildung 1.8 für ein Beispiel und Gardner & Giegerich, 2004, für eine formale Beschreibung). Diese „Fernbeziehung" wird aber in einem Sequenz-Alignment nicht berücksichtigt.

Echte Struktur-Alignment-Programme basieren auf Vereinfachungen das Sankoff-Algorithmus, so beispielsweise DYNALIGN, FOLDALIGN und PMCOMP (siehe Abschnitte 3.1.5, 3.1.6 respektive 3.1.11). Einen Mittelweg wählt STRAL (siehe Abschnitt 3.1.17).

1.3 Nutzen und Anwendung von RNA-Alignments

RNA-Alignments sind Grundlage für eine Vielzahl von Anwendungen. So werden seit Jahrzehnten Phylogenievorhersagen mit Hilfe von rRNA-Alignments durchgeführt (Olsen & Woese, 1993). Der ribosomalen RNA kommt insofern eine besondere Rolle als phylogenetischer Marker zu, da sie zentrale Funktionen in jeder Zelle übernimmt und in allen bekannten Organismen vorhanden ist. RNA-basierte Phylogenievorhersagen sind immer noch Gegenstand aktueller Forschung (siehe beispielsweise Hudelot *et al.*, 2003; Wolf *et al.*, 2005b).

Mit der stetig wachsenden Anzahl von sequenzierten Genomen steigt auch das Interesse an vergleichenden Sequenz-Analysen und der Suche nach noch unentdeckten ncRNAs. Für die Suche wurde eine Vielzahl von Strategien entwickelt, die zum Teil direkt auf multiplen RNA-Alignments basieren, so beispielsweise RNAZ (Washietl *et al.*, 2005) oder INFERNAL (Eddy, 2002). Gleichzeitig lassen sich anhand multipler RNA-Alignments Muster erstellen, die wiederum in Homologiesuchen eingesetzt werden können (siehe beispielsweise Gräf *et al.*, 2005, 2006).

Weiterhin sind RNA-Alignments essentielle Grundlage für nahezu alle Methoden zur Vorhersage von RNA-Konsensusstrukturen, wie CONSTRUCT, ILM, PFOLD, RNAALIFOLD etc. (hier in Abschnitt 1.5 besprochen).

Der großen Bedeutung von RNA-Alignments wurde mit Erstellung der Rfam („RNA family Database“) Rechnung getragen (Griffiths-Jones *et al.*, 2003, 2005), womit eine zentrale Quelle von RNA-Alignments, entsprechenden Kovarianz-Modellen zur Homologiesuche, sowie Konsensus-Strukturen zur Verfügung steht.

1.4 Einzelstruktur-Vorhersagen für RNA

Struktur-Vorhersagen für einzelne RNAs beschränken sich in nahezu allen Fällen auf die Sekundärstruktur. Hierfür sind effiziente Algorithmen nötig, da die Anzahl möglicher Sekundärstrukturen exponentiell mit der Länge der Sequenz wächst.

Formal wird eine Sekundärstruktur als eine Liste von Basenpaaren beschrieben, die folgende Bedingungen erfüllen: Eine Base kann maximal eine Basenpaarung eingehen und Basenpaare dürfen sich nicht überkreuzen, d. h. zwei Paare (i,j) und (k, l) dürfen keinen Pseudoknoten bilden: $i < k < j < l$.

Eine Methode der Sekundärstruktur-Vorhersage (neben der graphischen Methode nach Tinoco *et al.*, 1971) wird im Nussinov-Algorithmus (Basenpaar-Maximierung) beschrieben (Nussinov *et al.*, 1978). Dieser sagt per dynamischer Programmierung die Sekundärstruktur mit maximaler Anzahl Basenpaare vorher (siehe Eddy, 2004a, für eine schöne Beschreibung).

Der Nussinov-Algorithmus bedient sich lediglich einfacher Basenpaarregeln. Thermodynamische Methoden (siehe beispielsweise Zuker, 2000) funktionieren grundsätzlich ähnlich, jedoch nutzen sie verfügbare thermodynamische Parameter (Mathews *et al.*, 1999) für Basenpaare, Loop-Energien etc. und setzen statt der Basenpaar-Maximierung eine Energie-Minimierung ein. Die zur Verfügung stehenden Parameter werden auch „Nearest-Neighbour Rules“ genannt, da in Helices die Energien abhängig von benachbarten Basenpaaren sind. Entsprechende Implementationen sind LINALL (Schmitz & Steger, 1992), RNAFOLD (Hofacker, 2003) und MFOLD (Zuker, 2003). Diese Programme können mit Hilfe des Algorithmus nach Zuker & Stiegler (1981) eine optimale Sekundärstruktur mit minimaler freier Energie vorhersagen. Diese Struktur wird entsprechend MFE-Struktur („Minimum Free Energy“) genannt. Weiterhin ist es möglich suboptimale Strukturen nach Steger *et al.* (1984); Zuker (1989) zu bestimmen.

Da eine RNA in Lösung niemals eine fixe Struktur einnimmt, sondern ein Struktur-Ensemble vorliegt, ist es u. a. von Interesse die Strukturverteilung – also die Wahrscheinlichkeiten für

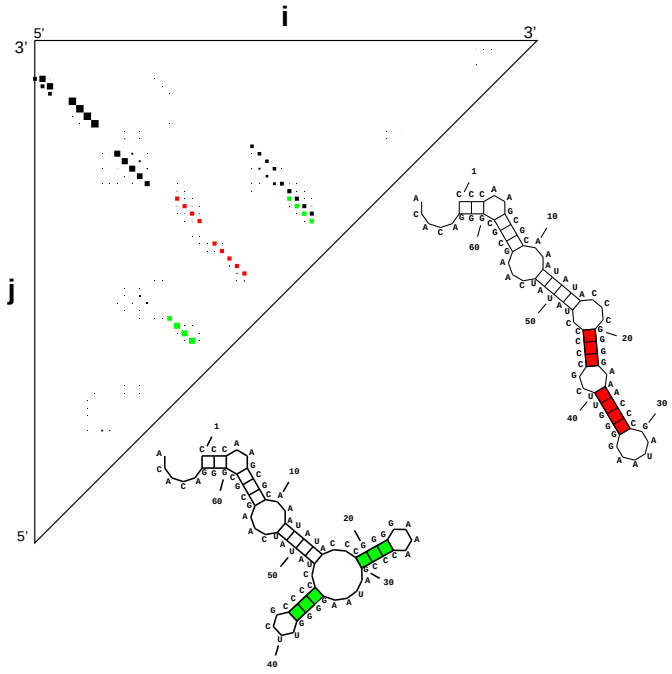


Abbildung 1.9: Thermodynamische Strukturverteilungen. Im Dotplot ist beispielhaft eine Strukturverteilung gezeigt. Die Sequenz ist horizontal (i) in 5'-3' und vertikal (j) in 3'-5'-Richtung aufgetragen. Basenpaare sind als Quadrate eingezeichnet, deren Fläche proportional zur entsprechenden Wahrscheinlichkeit ist. Man kann hier auch von einer Superposition/Überlagerung aller möglichen Strukturen/Faltungen sprechen. Beispielhaft sind zwei in dieser Strukturverteilung enthaltenen Strukturen gezeigt. Die entsprechenden Basepaare sind farbig markiert. Nach Steger (2003).

jedes mögliche Basenpaar – mit Hilfe des McCaskill-Algorithmus (McCaskill, 1990) zu berechnen. Dieser erlaubt die optimale Berechnung der Zustandssumme („Partition Function"), welche eine statistische Beschreibung des thermodynamischen Gleichgewichts darstellt. Diese Wahrscheinlichkeiten lassen sich in Form eines Dotplots darstellen, in dem die Sequenz horizontal in 5'-3' und vertikal in 3'-5'-Richtung aufgetragen wird (siehe Abbildung 1.9). Mögliche Basenpaare werden in diesem spiegelsymmetrischen Plot als Quadrate eingetragen, wobei die Fläche der Quadrate proportional zur entsprechenden thermodynamischen Wahrscheinlichkeit ist. Helices sind im Plot als Diagonalen zu erkennen, da sie durch konsekutive Abfolgen von Basenpaaren gebildet werden.

Die Sekundärstruktur lässt sich nur deshalb gut und ohne Berücksichtigung der Tertiärstruktur vorhersagen, da sie sich größtenteils unabhängig von der Tertiärstruktur bildet (Tinoco & Bustamante, 1999). Die Vorhersage ist allerdings nur so gut, wie die eingesetzten Parameter sind. Zudem werden kinetische Effekte während der Faltung außer Acht gelassen. Zur Qualität der Vorhersage per Energieminimierungsmethoden gibt es unterschiedliche Untersuchungen: Von Mathews *et al.* (1999) werden durchschnittlich 73% korrekt vorhergesagte Basenpaare für Sequenzen kleiner 800 Nukleotiden angegeben. Doshi *et al.* (2004) geben 71% für die 5S rRNA und 69% für tRNAs an, weisen aber auch auf schlechtere Werte für längere ribosomale Sequenzen hin (bis 20%); Dowell & Eddy (2004) bestimmten 56% korrekt vorhergesagte Basenpaare.

1.5 Konsensusstruktur-Vorhersagen für RNA

Die Konsensusstruktur-Vorhersage für RNAs basiert nahezu immer auf einem (multiplen) RNA-Alignment. Als goldener Standard für die Konsensusstruktur-Vorhersage gilt die sogenannte vergleichende Sequenz-Analyse („Comparative Sequence Analysis“; Pace *et al.*, 1999). Hier werden in einem Alignment paarweise über alle Spalten kompensatorische Basenpaaraustausche und Kovarianzen gesucht. Diese gelten als Hinweis darauf, dass die entsprechenden Stellen unter Erhalt der Basenpaarung/Struktur mutierten (siehe auch Anmerkung in Abschnitt 1.2.6) und somit ein Konsensus-Basenpaar bilden. Hiermit konnten schon früh sehr gute Modelle der ribosomalen RNAs vorhergesagt werden (ausführlich untersucht in Gutell *et al.*, 2002). Allerdings wird für diese statistische Methode ein sehr genaues Alignment von sehr vielen Sequenzen benötigt. Beide Bedingungen lassen sich in den wenigsten Fällen gleichzeitig erfüllen.

Weitere statistische Ansätze bedienen sich sogenannter stochastischer kontextfreier Grammatiken („Stochastic Context Free Grammars“; SCFG; siehe Dowell & Eddy, 2004). Eine Implementation ist PFOLD (Knudsen & Hein, 2003). Für eine Einleitung und ausführliche Diskussion der SCFGs sei auf Durbin *et al.* (1998) verwiesen.

Grundsätzlich lassen sich auch vereinfachte Implementationen des Sankoff-Algorithmus zur Konsensusstruktur-Vorhersage nutzen, da hier Konsensusstruktur und Struktur-Alignment simultan berechnet werden. Jedoch sind diese Programme aufgrund der hohen Komplexität nur für zwei Sequenzen einsetzbar (siehe Abschnitt 3.1.5, 3.1.6 bzw. 3.1.11).

Sowohl thermodynamische als auch statistische Ansätze haben typische Einschränkungen. So ist in Sequenz-konservierten Bereichen keine Struktur-Vorhersage per Kovarianz möglich, wohingegen thermodynamische Methoden aufgrund fehlender Parameter, beispielsweise für ungewöhnliche Basenpaare, scheitern können (Beispiel Loop E der 5S rRNA). Deshalb basieren die meisten Methoden auf einer Kombination aus Thermodynamik und Statistik, so beispielsweise RNAALIFOLD (Hofacker *et al.*, 2002) und ILM (Ruan *et al.*, 2004). Hierdurch ist eine Konsensusstruktur-Vorhersage mit relativ wenigen Sequenzen möglich und es werden die Eigenarten der jeweiligen Methode ausgeglichen. In

Zwei in diesem Zusammenhang hervorzuhebende Programme sind CONSTRUCT (siehe Abschnitt 3.3 für eine ausführliche Erläuterung) und X2S (Juan & Wilson, 1999). Beides sind semi-automatische Programme, die ebenfalls auf einer Kombination von Thermodynamik und Statistik basieren, und es dem Benutzer erlauben, dem jeweiligen Problem angemessen eine Gewichtung und Filterung der Daten vorzunehmen. Das Besondere an ihnen ist, dass sie es dem Benutzer ermöglichen das Alignment mit Unterstützung einer ausgefeilten graphischen Benutzeroberfläche zu korrigieren, wohingegen andere Methoden auf einem statischen Alignment basieren. Dies ist insofern entscheidend, als dass die Qualität der Konsensusstruktur-Vorhersage immer direkt von der Qualität des Alignments abhängt und gleichzeitig die Erstellung/Berechnung eines korrekten RNA-Alignments wie beschrieben sehr schwierig ist.

In Gardner & Giegerich (2004) findet sich ein ausführlicher Vergleich der meisten hier genannten Programme.

1.6 Benchmarks von Alignments

Da das Alignment-Problem immer nur annähernd gelöst werden kann und gleichzeitig eine sehr große Zahl von Programmen existieren, stellt sich die Frage, welcher der verfügbaren Ansätze unter welchen Bedingungen die besten Ergebnisse liefert. Gleiches gilt für eingesetzte Heuristiken, Bewertungsfunktionen, Substitutionsmatrizen und so weiter. So wurden im Laufe der Zeit einige Leistungsvergleichstests („Benchmarks“) durchgeführt, wie beispielsweise durch Thompson *et al.* (1999a) und Lassmann & Sonnhammer (2002). Im Zuge dessen sind mehrere Benchmark-Datenbanken konstruiert worden, die aus speziell zusammengestellten Referenz-Alignments bestehen, wie HOMSTRAD (Mizuguchi *et al.*, 1998), OXBench (Raghava *et al.*, 2003), PREFAB (Edgar, 2004b), SABmark (Van Walle *et al.*, 2005) und SMART (Letunic *et al.*, 2004). Ein Teil der für die entsprechenden Benchmarks eingesetzten Bewertungsmaße wird in Abschnitt 3.5 besprochen.

Im Zuge des bis *dato* ausführlichsten Benchmarks von Thompson *et al.* (1999a) wurde die bekannte BAliBASE („Benchmark Alignment Database“) (Thompson *et al.*, 1999b) erstellt, welche mehrfach erweitert wurde (Bahr *et al.*, 2001; Thompson *et al.*, 2005). Wie die anderen Datenbanken dient sie der systematischen Evaluierung von Protein-Alignment-Methoden, wobei sie anhand aufgelöster 3D-Strukturen verifizierte und manuell geprüfte Alignments enthält. Weiterhin sind kategorisierte Test-Sets enthalten, die sich hinsichtlich ihrer Sequenz-Ähnlichkeit, -Anzahl und -Länge, so wie der Anzahl nicht-zugehöriger Sequenzen („Orphans“) unterscheiden. Mit Hilfe dieser Test-Sets ist eine Quantifizierung des Einflusses der genannten Faktoren, sowie von Programm-Optionen auf die Alignment-Methoden möglich.

Jedoch sind die genannten Benchmarks inkl. der aufgeführten Benchmark-Datenbanken Protein-spezifisch. Erst kürzlich wurde ein, wenn auch sehr spezieller Benchmark für das paarweise, genomische Alignment nicht kodierender DNA durchgeführt (Pollard *et al.*, 2004). Ein Benchmark speziell für das Problem des Alignments von RNA bzw. ncRNA existierte bis zu Beginn dieser Arbeit nicht. So wurden die wenigen echten RNA-Alignment-Programme beispielsweise bisher zumeist über die Qualität der (simultan berechneten) Struktur-Vorhersage getestet.

1.7 Aufgabenstellung und Ziel dieser Arbeit

In der vorliegenden Arbeit sollte eine Benchmark-Datenbank für RNA-Alignments entwickelt werden, die als Datensätze möglichst perfekte Alignments von nicht-kodierenden RNAs (ncRNAs) enthält. Diese Datenbank sollte damit ein Pendant zu der Protein-spezifischen BAliBASE (Bahr *et al.*, 2001; Thompson *et al.*, 1999b, 2005) werden und entsprechend BRAliBase („Benchmark RNA Alignment Database“) [1] genannt werden.

[1] Der Begriff BRAliBase wurde zwischenzeitlich von Paul Gardner (Department of Evolutionary Biology, University of Copenhagen) adaptiert.

Für diese Datenbank werden eine große Anzahl qualitativ hochwertiger Referenz-Alignments (als richtige Lösung) benötigt, welche idealerweise auf einer 3D-Struktur-Superposition basieren sollten oder deren korrekte Struktur-Homologie auf andere Art sichergestellt sein sollte. Gleichzeitig sollten die Alignments in ihrer Zusammensetzung bzw. ihren Eigenschaften, wie Sequenz-Homologie und Sequenz-Zahl, gezielt variieren, um den Einfluss dieser Eigenschaften auf die verschiedenen Programme/Methoden analysieren zu können. Um dies sicherstellen zu können, müssen spezielle Verfahren zur Kompilation der Referenz-Alignments entwickelt werden.

Um die Leistung der Alignment-Programme im Vergleich zum Referenz-Alignment quantitativ bestimmen zu können, müssen zudem adäquate Güte-Maße entwickelt werden, welche die Besonderheiten des RNA-Alignments abbilden können. Die bis *dato* verfügbaren Maße sind speziell für den Test von Protein-Alignments entwickelt worden und erlauben es nicht, die strukturelle Konservierung in einem RNA-Alignment zu beschreiben.

Schließlich sollen mit Hilfe dieses Benchmarks existierende Alignment-Programme und -Methoden systematisch auf ihre Eignung für das RNA-Alignment evaluiert werden. So soll die Frage untersucht werden, unter welchen Bedingungen der Einsatz echter Struktur-Alignment-Programme nötig ist, bzw. unter welchen Bedingungen die weitaus schnelleren Sequenz-Alignment-Programme ähnlich gute Lösungen liefern. Zudem lassen sich mit der Verfügbarkeit einer solchen Benchmark-Datenbank Programmfehler systematisch untersuchen und Programmparameter optimieren.

Weiterhin soll festgestellt werden, welche der Alignment-Methoden bzw. welches Alignment-Programm im Allgemeinen am besten für das RNA-Alignment geeignet ist.

2

Material und Methoden

2.1 Entwicklungsumgebung

Bei dem eingesetzten Betriebssystem handelte es sich um GNU/Linux in Form der Debian[1] geht meist schief wegen sonderzeichen -Distributionen Version 3.0 und 3.1 mit den Linux-Kerneln[2] geht meist schief wegen sonderzeichen 2.4 sowie 2.6. Die meisten Rechnungen wurden auf Pentium III-Doppelprozessor-Systemen mit je 800 MHz Taktrate und 512 MB RAM bzw. einem 64Bit-AMD-Opteron Doppelprozessor-System (1800 MHz) mit 4 GB RAM ausgeführt. Einige Struktur-Alignments des Abschnitts 3.6 erfolgten zusätzlich auf einer Sun V20z 244, ausgestattet mit zwei 64Bit-AMD-Opteron Prozessoren (1800 MHz) und 8 GB RAM, auf der Red Hat Fedora Linux[3] geht meist schief wegen sonderzeichen Core 3 mit Linux-Kernel 2.6 installiert war.

2.2 Alignment-Programme und Optionen

In den folgenden Tabellen sind alle in Abschnitt 3.6 und Abschnitt 3.7 eingesetzten Alignment-Programme aufgeführt. Zusätzlich zu der Version sind die entscheidenden Kommandozeilenparameter und ein entsprechendes Optionskürzel angegeben. Variable Parameter sind rekursiv gedruckt; Zeilenumbrüche sind durch einen umgekehrten Schrägstrich („Backslash“) gekennzeichnet.

[1] http://www.debian.org/

[2] http://kernel.org/

[3] http://fedora.redhat.com/

2.2.1 Benchmark I (BRAliBase II)

Tabelle 2.1: Versionen und Kommandozeilenparameter der in Abschnitt 3.6 eingesetzten Alignment-Programme.

Sequenzalignment-Programme	
ALIGN-M	**Version 2.1 (Van Walle *et al.*, 2004)**
ALIGN-M (1)	`align_m -m RNA2`
ALIGN-M (2)	`align_m -m RNA2 -p2m_Fmin 0.7 -p2m_nseq_min 5`
ALIGN-M (3)	`align_m -m RNA2 -s2p_go 10 -s2p_ge 1`
ALIGN-M (4)	`align_m -m RNA2 -s2p_go 10 -s2p_ge 1 -p2m_Fmin 0.7 -p2m_nseq_min 5`
ALIGN-M (5)	`align_m -m RNA2 -s2p_w 3`
CLUSTALW	**Version 1.82 (Thompson *et al.*, 1994)**
CLUSTALW	`clustalw -type=dna -align`
CLUSTALW (qt)	`clustalw -type=dna -align -quicktree`
DIALIGN	**Version 2.2 (Morgenstern, 1999, 2004)**
DIALIGN	`dialign2-2 -n`
DIALIGN (it)	`dialign2-2 -n -it`
DIALIGN (o)	`dialign2-2 -n -o`
DIALIGN (it,o)	`dialign2-2 -n -it -o`
HANDEL	**Version 0.1 (Programmpaket dart) (Holmes, 2003)**
HANDEL	`handalign.pl`
MAFFT	**Version 4.22 (hier Katoh *et al.*, 2002)**
MAFFT (fftnsi)	`fftnsi`
MAFFT (fftns)	`fftns`
MAFFT (nwnsi)	`nwnsi`
MAFFT (nwns)	`nwns`
MUSCLE	**Version 3.51 (Edgar, 2004a,b)**
MUSCLE	`muscle`
MUSCLE (nj)	`muscle -cluster1 neighborjoining -cluster2 neighborjoining`
MUSCLE (mi32)	`muscle -maxiters 32`
MUSCLE (nj,mi32)	`muscle -maxiters 32 -cluster1 neighborjoining -cluster2 neighborjoining`
MUSCLE (m6)	`muscle -maxtrees 6`
MUSCLE (nj,mt6)	`muscle -maxtrees 6 -cluster1 neighborjoining -cluster2 neighborjoining`
MUSCLE (mi32,mt6)	`muscle -maxiters 32 -maxtrees 6`
MUSCLE (nj,mi32,mt6)	`muscle -maxiters 32 -maxtrees 6 -cluster1 neighborjoining \ -cluster2 neighborjoining`
PCMA	**Version 2.0 (Pei *et al.*, 2003)**
PCMA	`pcma`
PCMA (agi20)	`pcma -ave_grp_id=20`
PCMA (agi60)	`pcma -ave_grp_id=60`
POA	**Version 2 (Lee *et al.*, 2002)**
POA	`poa -v blosum80.mat`
POA (g)	`poa -do_global blosum80.mat`
POA (p)	`poa -do_progressive blosum80.mat`
POA (g,p)	`poa -do_global -do_progressive blosum80.mat` (die `blosum80.mat` von POA enthält auch Werte für Nukleotid-Substitutionen)
PROALIGN	**Version 0.5 (Löytynoja & Milinkovitch, 2003)**
PROALIGN (bw400)	`java -Xmx256m -jar ProAlign_0.5a0.jar -bwidth=400`

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

PRRN	**Programmpaket scc Version 3.0 (Gotoh, 1996)**
PRRN	`prrn`
PRRN (S10)	`prrn -S10`
T-COFFEE	**Version 1.37 (Notredame *et al.*, 2000)**
T-COFFEE	`t_coffee`
T-COFFEE (c)	`t_coffee -in=Mlalign_id_pair,Mclustalw_pair`
T-COFFEE (f)	`t_coffee -in=Mlalign_id_pair,Mfast_pair`
T-COFFEE (s)	`t_coffee -in=Mlalign_id_pair,Mslow_pair`
Strukturalignment-Programme	
DYNALIGN	**2. Edition (Mathews & Turner, 2002; Mathews, 2005)**
DYNALIGN	`dynalign len2-len1+5 0.4 5 20 2 1 0` (len1 bezeichnet die Länge der kürzeren, len2 der längeren Sequenz)
FOLDALIGN	**Version 2.0.0 (Havgaard *et al.*, 2005b)**
FOLDALIGN	`foldalign -global -max_diff 25 -score_matrix global.fmat`
PMCOMP	**Programmpaket Vienna RNA 1.5 alpha (Hofacker *et al.*, 2004)**
PMCOMP	`pmcomp.pl`
PMCOMP (fast)	`pmcomp.pl -fast`
STEMLOC	**Version 0.19b (Holmes, 2004, 2005)**
STEMLOC (slow)	`stemloc -global -multiple -verbose -nfold 1000 -norndfold`
STEMLOC (fast)	`stemloc -global -multiple -verbose -nfold 110 -norndfold`

2.2.2 Benchmark II (BRAliBase IV)

Tabelle 2.2: Versionen und Kommandozeilenparameter der in Abschnitt 3.7 eingesetzten Alignment-Programme.

ALIGN-M	**Version 2.3 (Van Walle *et al.*, 2004)**
ALIGN-M	`align_m -m` *`MATRIX`*
ALIGN-M (s2p)	`align_m -m` *`MATRIX`* `-s2p_w 23 -s2p_go 8 -s2p_ge 0.5`
CLUSTALW	**Version 1.83 (Thompson *et al.*, 1994)**
CLUSTALW	`clustalw -type=dna -align -dnamatrix=`*`MATRIX`* `-pwdnamatrix=`*`MATRIX`*
CLUSTALW	`clustalw -type=dna -align -pwgapopen=`*`GO`* `-gapopen=`*`GO`* `-pwgapext=`*`GE`* `-gapext=`*`GE`*
DIALIGN	**Version 2.2.1 (Morgenstern, 1999, 2004)**
Siehe Abschnitt 2.2.1	
DIALIGN-T	**Version 0.2.1 (Subramanian *et al.*, 2005)**
DIALIGN-T	`dialign-t -D`
HANDEL	**Programmpaket dart Version 0.2 (Holmes, 2003)**
Siehe Abschnitt 2.2.1	
MAFFT	**Version 5.667 (hier Katoh *et al.*, 2005)**
MAFFT (einsi)	`einsi`
MAFFT (fftns)	`fftns`
MAFFT (fftnsi)	`fftnsi`
MAFFT (ginsi)	`ginsi`
MAFFT (linsi)	`linsi`
MAFFT (nwns)	`nwns`
MAFFT (nwnsi)	`nwnsi`
MAFFT (fftns,alt)	`fftns -op 0.51 -ep 0.041`

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

MAFFT (ginsi,alt)	`ginsi -op 0.51 -ep 0.041`
MAFFT (linsi,alt)	`linsi -op 0.51 -ep 0.041`
MUSCLE	**Version 3.6 (Edgar, 2004a,b)**
MUSCLE	`-seqtype rna`
MUSCLE	`-seqtype rna -gapopen` *`GO`*
PCMA	**Version 2.0 (Pei *et al.*, 2003)**
Siehe Abschnitt 2.2.1	
POA	**Version 2 (Lee *et al.*, 2002)**
POA	`poa -do_global` *`MATRIX`*
POA (p)	`poa -do_global -do_progressive` *`MATRIX`*
PRANK	**Version 270705b – 1508b (Löytynoja & Goldman, 2005)**
PRANK	`prank -gaprate=`*`GR`* `-gapext=`*`GE`*
PROALIGN	**Version 0.5a2 und 0.5a3 (Löytynoja & Milinkovitch, 2003)**
Siehe Abschnitt 2.2.1	
PRRN	**Version 3.0 (Programmpaket scc) (Gotoh, 1996)**
PRRN	`prrn`
PRRN (S10)	`prrn -S10`
PRRN (J2)	`prrn -J2`
PRRN (J2,S10)	`prrn -J2 -S10`
STRAL	**Version 0.4.0 (Dalli, 2006)**
STRAL	`stral`
T-COFFEE	**Version 3.03 (Notredame *et al.*, 2000)**
T-COFFEE (lp,sp)	`t_coffee -in=Mlalign_id_pair4dna,Mslow_pair4dnalib`
T-COFFEE (lp,cp)	`t_coffee -in=Mlalign_id_pair4dna,Mclustalw_pair`
T-COFFEE (lp,cm)	`t_coffee -in=Mlalign_id_pair4dna,Mclustalw_msa`
T-COFFEE (lp,sp,cp,cm)	`t_coffee -in=Mlalign_id_pair4dna,Mslow_pair4dnalib,Mclustalw_pair,Mclustalw_msa`

2.3 Programme zur Bewertung von Alignments

Zur Bewertung der Alignments kamen – wenn nicht anders vermerkt – folgende nicht selbst entwickelte Programme zum Einsatz: RNAZ (Version 0.1.1; Washietl *et al.*, 2005) zur Berechnung des SCI, BAliScore (Thompson *et al.*, 1999a) zur Berechnung der SPS und das im Programmpaket SQUID (Version 1.9g; Eddy, 2005) enthaltene ALISTAT um die Sequenz-Identität als APSI („Average Pairwise Sequence Identity“; siehe Abschnitt 3.5.3) zu bestimmen.

2.4 Sonstige Programme und Bibliotheken

Als Compiler wurden ausschließlich solche der GCC[4] geht meist schief wegen sonderzeichen in den Versionen 2.95 bis 4.0 verwendet. An verschieden Stellen wurden die Bibliotheken RNALIB Version 1.5a (Hofacker *et al.*, 1994; Hofacker, 2003) und SQUID Version 1.9g (Eddy, 2005) genutzt. Zur Erstellung von Diagrammen und zur Berechnung der statistischen Test kam das Softwarepaket R[5] geht meist schief wegen sonderzeichen ab Version 2 zum Einsatz.

[4] http://gcc.gnu.org/

[5] http://www.r-project.org/

2.5 Lowess-Funktion

Die Werte der in Abschnitt 3.6 und Abschnitt 3.7 gezeigten Plots streuen stark. Um trotzdem einen klaren visuellen Eindruck zu vermitteln, wurden die Werte mit Hilfe der Lowess-Funktion (Cleveland, 1979, 1981) geglättet. Diese Funktion wird oft zur Normalisierung/Glättung von Scatter-Plots in der Microarray-Analyse genutzt. Der Name Lowess leitet sich von "**Lo**cally **We**ighted **S**catter Plot **S**moothing" ab. Es handelt sich dabei um eine lokal gewichtete Regressionsfunktion, d. h. jeder geglättete Wert ergibt sich aus den Werten der Nachbarn, wobei nähere Datenpunkte stärker gewichtet werden. Man kann auch von einer lokal linearen Approximation sprechen.

Ein Vorteil dieser Funktion ist, dass sie keine Annahme über die Verteilung der Werte voraussetzt. Ein Nachteil stellt die Wahl des Glättungsparameters dar. Dieser gibt die Fensterbreite für die zu berücksichtigenden Datenpunkte an. Große Werte führen also zu stark geglätteten Kurven. Der Parameter muss dem jeweiligen Datensatz angepasst werden, wobei er in dieser Arbeit immer möglichst niedrig gewählt wurde, um eine zu starke Glättung zu vermeiden.

Es sei darauf hingewiesen, dass durch diese Glättung bei einer lokal sehr geringen Anzahl von Datenpunkten etwas artifizielle Kurvenverläufe an den Kurvenenden entstehen können. In den entsprechenden Plots (Abschnitt 3.6 und Abschnitt 3.7) werden deshalb die Kurvenverläufe erst ab APSI-Werten größer 0,2 bzw. größer 0,4 gezeigt.

2.6 Statistische Rangtests

Ähnlich zum Vorgehen in Thompson *et al.* (1999a) wurden in Abschnitt 3.7 Friedman-Tests und Wilcoxon-Rangsummentests durchgeführt. Beide Test sind nicht-parametrische (verteilungsfreie, parameterfreie) Test-Verfahren, d. h. sie setzen keine Annahme über die Verteilung der Werte voraus. Für eine genauere Beschreibung wird in beiden Fällen auf Lehrbücher, wie beispielsweise Precht & Kraft (1993) oder Sachs (2004) verwiesen.

Mit Hilfe des Friedman-Tests wurden Rangfolgen der Programme bestimmt. Anschließend wurden unabhängig davon Wilcoxon-Rangsummentests für jedes mögliche Programm-Paar durchgeführt. Beide Tests wurden von Indra Mainz in Form von R-Skripten implementiert und jeweils ein Signifikanzniveau von 5% verwendet.

2.6.1 Friedman-Test

Der Friedman-Test wurde eingesetzt, um eine Rangfolge der Programme zu bestimmen. Dabei wird die Nullhypothese des Tests „Der Median aller Behandlungen ist gleich" gegen die Alternative „Mindestens ein Median unterscheidet von den anderen" getestet. Die Analyse geschieht in Blöcken, wobei ein Block hier einer Alignment-Bewertung entspricht, und es werden den Werten innerhalb eines Blockes Ränge zugeteilt. Anschließend erfolgt die Bildung der Rangsummen für jede „Behandlung" (hier jedem Programm). Sollten sich die Rangsummen nicht

zufällig unterscheiden, so wird die Nullhypothese verworfen. Zur Überprüfung dient die sog. Friedman-Prüfgröße.

2.6.2 Wilcoxon-Rangsummentest

In Anschluss an jeden Friedman-Test wurde für jedes Werte-Paar der Wilcoxon-Rangsummentest durchgeführt, der hier darüber Aufschluss gibt, ob sich zwei Programme bzw. deren Leistung signifikant unterscheidet. Der Wilcoxon-Rangsummentest dient dem Vergleich zweier unabhängiger Stichproben. Diese werden zunächst gemeinsam nach ihrer Größe sortiert und jedem Platz in der entstehenden Folge wird eine Rangzahl zugeteilt. Anschließend wird für jede der Stichproben die Rangsumme gebildet. Diese werden mit einer Prüfgröße U (daher der Alternativname U-Test) verglichen, anhand dessen die Nullhypothese „Die Verteilungsfunktionen der beiden Grundgesamtheiten sind identisch“ entschieden wird.

2.7 Sequenzen und Alignments

Die in Abschnitt 3.7, 3.6 und 3.3 verwendeten Alignments und Sequenzen wurden den dort angegebenen Datenbanken oder Veröffentlichungen entnommen.

Eine zentrale Rolle nahmen die Alignments der Rfam ein. In Abschnitt 3.6 wurden Alignments der Version 5 (Griffiths-Jones *et al.*, 2003), in Abschnitt 3.7 Alignments der Version 7 verwendet (Griffiths-Jones *et al.*, 2005). Die Alignments der Rfam sind in den meisten Fällen der Literatur entnommen und wurden u. U. manuell korrigiert. Mit Hilfe dieser sogenannten „Seed“-Alignments wird eine Datenbank-Suche durchgeführt und gefundene homologe Sequenzen dem Alignment hinzugefügt, wodurch die „Full“-Alignments entstehen.

3

Ergebnisse

In diesem Kapitel wird zunächst grob die Vorgehensweise der eingesetzten Alignment-Programme beschrieben (siehe Abschnitt 3.1), um später eine Diskussion der beim Benchmark festgestellten Eigenschaften zu ermöglichen. Für detaillierte Darstellungen wird auf die jeweils genannten Publikationen verwiesen. Da bei der Nutzung der Programme zum Teil massive Probleme auftraten, musste für nahezu jedes Programm ein Helfer-Skript entwickelt werden.

Bei der Entwicklung einer Benchmark-Datenbank für RNA-Alignments war es unabdingbar das Werkzeug CONSTRUCT einzusetzen, welches hier u. a. eine Evaluation der unten erwähnten Bewertungsfunktionen und eine Visualisierung und qualitative Begutachtung von RNA-Alignments ermöglichte. Da CONSTRUCT um grundlegende Funktionen erweitert wurde, werden die Funktionsweise und die neuen Eigenschaften dieses Werkzeuges in Abschnitt 3.3 besprochen. Zunächst sollte eine Referenz-Alignment-Datenbank ausschließlich mit Hilfe von CONSTRUCT-verifizierten Alignments aufgebaut werden (siehe Abschnitt 3.4). Dieser Plan wurde später allerdings verworfen, da nicht zuletzt zur statistischen Auswertung eine sehr hohe Zahl von Referenz-Alignments mit gezielt variierenden Eigenschaften benötigt wurde. Die Erstellung mit Hilfe von CONSTRUCT hätte beträchtliche Zeit in Anspruch genommen. Stattdessen wurden anhand von Alignments aus der Rfam auf zwei verschiedene Arten Referenz-Alignments gewünschter Eigenschaften erstellt (siehe Abschnitt 3.6.2 und Abschnitt 3.7.2).

Um die Güte von RNA-Alignments, sowie die Leistung von Alignment-Programmen quantitativ beschreiben zu können, bedarf es entsprechender Bewertungsfunktionen, die im Abschnitt 3.5 besprochen und mit Hilfe von CONSTRUCT und den zuvor erstellten Alignments evaluiert wurden (Daten nicht gezeigt).

In Kooperation mit Paul Gardner[1] und Stefan Washietl[2] entstand schließlich der erste bis dahin publizierte Benchmark von Alignment-Programmen angewendet auf RNAs (siehe Abschnitt 3.6 und Gardner *et al.*, 2005). Diese Arbeit wurde anschließend durch neue Referenz-Alignments, sowie ausführliche statistische Auswertungen wesentlich fortgeführt. Die Resultate sind in Abschnitt 3.7 vorgestellt.

[1] Department of Evolutionary Biology, University of Copenhagen

[2] Institut für Theoretische Chemie und Molekulare Strukturbiologie, Universität Wien

3.1 Beschreibung der eingesetzten Alignment-Programme

Aufgrund der großen Zahl an vorhandenen Alignment-Programmen konnte in dieser Arbeit nur eine Auswahl an Programmen verwendet werden. Dabei wurden nur solche Programme eingesetzt, die lokal installierbar und nicht ausschließlich als Webservice zur Verfügung stehen (wie beispielsweise MA-RNA), da nur so sinnvoll eine Erstellung hunderter Alignments im Batch-Verfahren möglich war. Weiterhin konnten Programme, die keinen IUPAC-Mehrdeutigkeitscode bzw. keine Eingabe von Ns („a**N**y nucleotide“) als Nukleotid-Zeichen zulassen (beispielsweise RNAFORESTER), nicht sinnvoll eingesetzt werden. Zudem musste in allen Fällen garantiert sein, dass durch die Programme ein komplettes „globales“ Alignment erstellt wird, d. h. nicht nur alignierte Fragmente ausgegeben werden, die mit den hier eingesetzten Methoden nicht bewertet werden können. Eine Einteilung des Großteils der hier aufgeführten Programme in entsprechende Kategorien findet sich in Abbildung 3.13. Die Vorgehensweise der einzelnen Programme wird im Folgenden grob erklärt. Detaillierte Beschreibungen würden den Rahmen dieser Arbeit sprengen, weshalb auf die jeweils genannten Referenzen verwiesen sei.

3.1.1 ALIGN-M

ALIGN-M (Van Walle *et al.*, 2004) ist ein multiples Alignment-Programm, welches sich vor allem für das Alignment von hoch divergenten Sequenzen eignen soll. Das Programm besteht aus drei separat, sequentiell und auch iterativ nutzbaren Modulen namens S2P, P2P und P2M. Das Programm berechnet normalerweise in einem dreistufigen Prozess ein multiples Alignment. Im ersten Schritt (Modul S2P) wird ein Set aus hoch-bewerteten („high-scoring“) lokalen Alignments berechnet. Die Scores der erstellten lokalen Alignments werden über einen Bereich vorgegebener Länge, der keine Gaps enthält, gemittelt und mit Hilfe der Sum-of-Pairs bewertet bzw. über den FASTER-Algorithmus approximiert (für Details siehe Desmet *et al.*, 2002; Van Walle *et al.*, 2004). Im zweiten Schritt (Modul P2P) werden, ähnlich der Bibliothekserweiterung bei T-COFFEE, die zuvor erstellten Alignments eingesetzt, um daraus Scores zu berechnen, die das Vorgehen bei den folgenden Alignments sinnvoll leiten/führen („guide“) sollen. Hierfür werden per dynamischer Programmierung mehrere paarweise Alignments pro Sequenz-Paar erzeugt. Die im ersten Schritt berechneten Scores fließen in diesen Prozess ein, indem zunächst die Matrix für die dynamische Programmierung mit den Werten („Similarity Scores“) aus der vorgegebenen Substitutionstabelle gefüllt wird. Anschließend werden Scores von Resten, welche sich in den zuvor erzeugten lokalen Alignments finden, durch sogenannte Waypoint-Scores ersetzt, die sich aus den zuvor berechneten lokalen paarweisen Alignments ergeben. Schließlich werden im dritten Schritt (Modul P2M) die noch vorliegenden paarweisen Alignments auf ihre Konsistenz hin überprüft, was durch Umwandlung in sogenannte Konsistenz-Matrizen geschieht (für Details siehe Van Walle *et al.*, 2003, 2004). Für jede dieser Konsistenz-Matrizen wird ein finales konsistentes paarweises Alignment berechnet. Die damit bei N Sequenzen entstehenden $N(N-1)/2$ Alignments können dann letztendlich zu einem multiplen Alignment kombiniert werden. ALIGN-M ist damit eine Art lokales, Konsistenz-basiertes Alignment-Programm.

3.1.2 CLUSTALW

CLUSTALW (Thompson *et al.*, 1994) ist das Standard-Beispiel für ein progressives Alignment-Programm (siehe auch Chenna *et al.*, 2003; Thompson *et al.*, 1997). CLUSTALW bzw. die entsprechende Version inkl. graphischer Benutzeroberfläche (GUI) namens CLUSTALX kann zudem als das Standard-Alignment-Programm bezeichnet werden.

Die Vorgehensweise lässt sich in die folgenden drei Schritte unterteilen: Zunächst wird ein globales Alignment aller (bei N Sequenzen) $N(N-1)/2$ möglichen Paare durchgeführt, um anhand der hierbei ermittelten Distanzen im anschließenden Schritt per Neighbour-Joining einen „Guide-Tree“ zu erstellen. Anhand dessen wird im finalen Schritt ein multiples Alignment erstellt, indem die Sequenzen (und später Profile) immer den Verzweigungen des Baumes folgend mit der nächst ähnlichen Sequenz oder dem nächst ähnlichen Profil aligniert werden.

Im Gegensatz zur Vorgänger-Version CLUSTALV wurden in CLUSTALW sehr spezielle Heuristiken eingeführt, wie eine Sequenz-Gewichtung und positionsspezifische Gapkosten. So werden aus dem Guide-Tree anhand der Sequenz-Ähnlichkeiten – also der Distanz in Abhängigkeit zum letzten gemeinsamen Verzweigungspunkt – Gewichte (CLUSTAL**W**=Weights) extrahiert, die in die Parameter des progressiven Alignments einfließen. Ein weiterer Unterschied zum klassischen progressiven Alignment ist, dass die initial vorgegebenen Gap-Kosten anhand der Sequenz-Ähnlichkeiten, den Sequenz-Längen und in Abhängigkeit von bereits (auch entfernt) vorhandenen Gaps dynamisch variiert werden, womit die Sensitivität und Effizienz erhöht werden soll. Gleichzeitig werden terminale Gaps nicht bewertet.

Der zeitintensive Teil der Methode ist das Alignment zweier Gruppen von Sequenzen im finalen progressiven Alignment. Um hier ein Alignment mit großen Sequenz-Längen mit einem sinnvollem Zeitaufwand zu finden, wird eine Version des speichereffizienten Algorithmus nach Miller & Myers (1988) verwendet, welcher so verändert wurde, dass er eine Variierung der Gap-Kosten zulässt.

3.1.3 DIALIGN

DIALIGN (Morgenstern, 1999, 2004) aligniert Gap-freie Segmente als ganzes, ohne dass Gaps eingefügt werden müssen, bzw. ohne dass diese explizit bestraft werden. Man spricht auch von einem Segment-basiertem Ansatz, da ein Vergleich von Fragmenten statt einzelner Reste stattfindet. DIALIGN vermeidet es dabei, nicht-ähnliche Bereiche zu alignieren, und verwendet grundsätzlich nur solche Fragmente, die in etwa gleich lang sind und statistisch signifikante Ähnlichkeit aufweisen. Hierfür werden zunächst homologe Segment-Paare identifiziert. Diese sind in einem Dotplot als Diagonalen zu erkennen (daher auch der Name; siehe auch **a** in Abbildung 3.1). Diese Segmente werden anhand der P-Values, ähnlich BLAST, gewichtet. Da diese gewichteten Diagonalen-Sets pro paarweisem Vergleich untereinander nicht kompatibel sind, wird hieraus ein konsistentes Subset extrahiert. Hierbei werden die Gewichte der Diagonalen durch sogenannte „Overlap Weights“ justiert und entsprechend sortiert (**b** und **c** in Abbildung 3.1). Anschließend werden die so gefundenen Diagonalen mit einem „greedy“ Algorithmus entsprechend ihrer Gewichte zu einem multiplen Alignment zusammengesetzt.

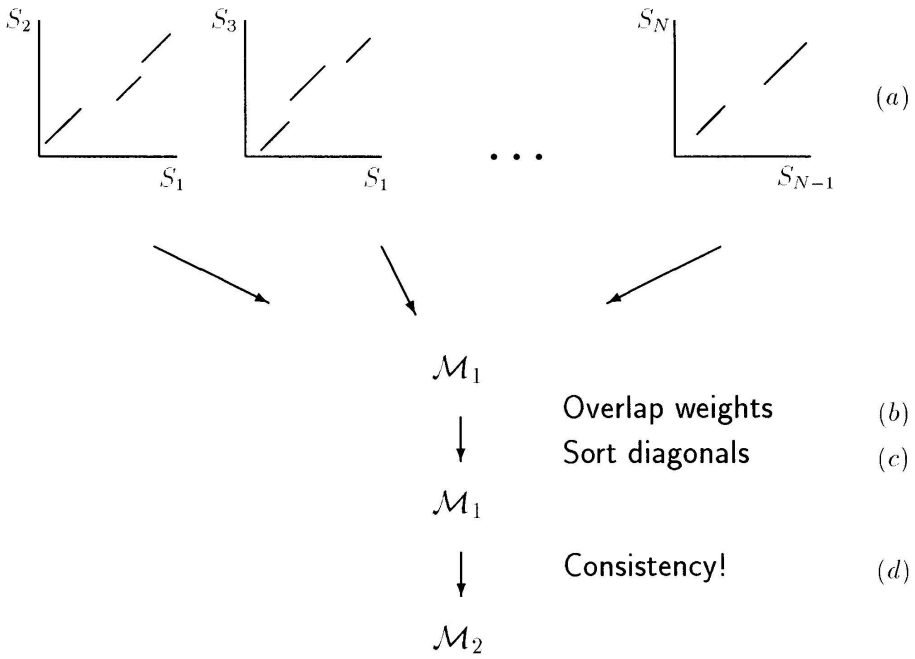


Abbildung 3.1: Vorgehensweise von DIALIGN. Eine Beschreibung befindet sich im Text. Entnommen aus Morgenstern (1999).

3.1.4 DIALIGN-T

DIALIGN-T (Subramanian *et al.*, 2005) ist eine Reimplementation bzw. Erweiterung des Segment-basierten Ansatzes von DIALIGN (siehe vorheriger Abschnitt). In Subramanian *et al.* (2005) wird gezeigt, dass die Bewertungsfunktion („Objective Function“) von DIALIGN systematisch isolierte, hoch-bewertete („high-scoring“) Fragmente überbewertet. Um dies zu umgehen, wurden in DIALIGN-T neue Heuristiken im paarweisen und multiplen Alignment implementiert, die dafür sorgen, dass eine Folge/Kette von schlecht-bewerteten („low-scoring“) Fragmenten einem isolierten, lokalen Fragment hoher Ähnlichkeit vorgezogen werden. Weiterhin kann das Programm im Gegensatz zu DIALIGN begrenzt mit sogenannten inkonsistenten Fragmenten umgehen, indem diese Fragmente soweit verkürzt werden, bis sie wieder konsistent sind. Zudem wurde die Möglichkeit hinzugefügt andere Substitutionsmatrizen als BLOSUM 62 zu verwenden, was in DIALIGN nicht möglich ist (dort hart einkodiert).

3.1.5 DYNALIGN

DYNALIGN (Mathews & Turner, 2002; Mathews, 2005) implementiert eine Vereinfachung des Sankoff-Algorithmus (Sankoff, 1985) und erstellt somit ein echtes RNA-Struktur-Alignment. Dabei wird simultan ein Alignment erstellt und die gemeinsame MFE-Struktur der Sequenzen vorhergesagt, wobei folgende Vereinfachungen eingeführt werden müssen, um die exponentielle Komplexität des Sankoff-Algorithmus zu meiden: zum einen ist nur ein paarweises Alignment möglich und zum anderen wird eine maximale Distanz M zwischen zwei zu alignierenden Sequenzen definiert bzw. vom Benutzer vorgegeben. Durch letztere Maßnahme reduziert sich die Komplexität des Algorithmus auf $\mathcal{O}(M^3N^3)$, wobei N die Länge der kürzeren Sequenz ist.

Weiterhin kann DYNALIGN keine Pseudoknoten vorhersagen. Für die Struktur-Vorhersage werden thermodynamische („Nearest-Neighbour“) Regeln verwendet, wobei sich die freie Energie der gemeinsamen Struktur durch folgende Formel ergibt:

$$\Delta G^0_{\text{total}} = \Delta G^0_{\text{Sequenz 1}} + \Delta G^0_{\text{Sequenz 2}} + (\Delta G^0_{\text{Gap}}) \cdot (\#\,\text{Gaps})\ . \tag{3.1}$$

Hier ist ΔG^0_{Gap} ein empirischer Faktor, der jedes Gap in dem Alignment bestraft. Vorausgesetzt, dieser Faktor und M sind optimal gewählt, so ist garantiert, dass DYNALIGN eine optimale Lösung findet, da keinerlei Heuristiken eingesetzt werden. DYNALIGN setzt keinerlei Sequenz-Informationen ein, um so auch ohne jegliche vorhandene Sequenz-Ähnlichkeit ein genaues Struktur-Alignment berechnen zu können.

3.1.6 FOLDALIGN

Version 2 von FOLDALIGN (Havgaard *et al.*, 2005a,b) basiert wie DYNALIGN und PMCOMP (siehe Abschnitte 3.1.5 und 3.1.11) ebenfalls auf dem Sankoff-Algorithmus (Sankoff, 1985), wobei hier als Vereinfachung die maximale Länge eines gesuchten Struktur-Motivs, sowie der maximale Längenunterschied zwischen zwei zu vergleichenden Zeichenketten limitiert werden. FOLDALIGN bedient sich vereinfachter Regeln aus Energieminimierungsmethoden (Stacking-Boni etc.) in Verbindung mit RIBOSUM-Matrizen (Klein & Eddy, 2003), womit im Gegensatz zu DYNALIGN auch Sequenz-Informationen genutzt werden. Wie bei DYNALIGN ist lediglich ein Alignment von zwei Sequenzen möglich. In der Standard-Einstellung von FOLDALIGN wird ein lokales Alignment berechnet. In dieser Arbeit hingegen wurde jeweils per Kommandozeilen-Parameter ein globales Alignment erzwungen (siehe Abschnitt 2.2.1).

3.1.7 HANDEL

HANDEL (Holmes, 2003; Holmes & Bruno, 2001) implementiert die Idee des „statistischen Alignments“. Das zugrundeliegende Modell ist das sogenannte Thorne-Kishino-Felsenstein-Modell, welches auch TKF91 genannt wird. HANDEL nutzt paarweise Hidden-Markov-Modelle (HMM), um hieraus evolutionäre HMMs zu generieren. Paarweise HMMs ähneln Standard-HMMs mit der Ausnahme, dass paarweise statt einfacher Emissionen stattfinden (eine ausführliche Erklärung findet sich in Durbin *et al.*, 1998). Evolutionäre HMMs stellen einen Sonderfall multipler HMMs dar (Holmes & Bruno, 2001) und entstehen durch Assoziierung der TKF91 paarweisen HMMs an einen phylogenetischen Guide-Tree („Branch-HMM“). Die evolutionären HMMs werden genutzt, um per dynamischer Programmierung ein „wahrscheinliches“ multiples Alignments zu erstellen. Für eine formale Beschreibung der neuartigen HMM-Typen und der komplexen Vorgehensweise sei auf Holmes (2003), sowie die dort genannten Referenzen verwiesen.

3.1.8 MAFFT

MAFFT (Katoh *et al.*, 2005, 2002) unterscheidet sich von anderen Sequenz-Alignment-Programmen insbesondere durch die Anwendung der schnellen Fourier-Transformation (FFT), womit hier eine sehr schnelle Identifikation homologer Bereiche möglich ist. Da in der aktuellen Version 5 insgesamt zehn verschiedene Ansätze implementiert sind, wird für eine detaillierte Beschreibung auf die beiden genannten Publikationen und die MAFFT-Homepage[3] geht meist schief wegen sonderzeichen verwiesen.

Das Vorgehen ist wie folgt: Zunächst wird für alle möglichen Sequenzpaare eine approximative Distanzmatrix bestimmt. Dies geschieht mit Hilfe des „k-mer-Countings“: hier dient die Anzahl gleicher 6-Tupel als Näherung der Sequenz-Ähnlichkeit (bei den Optionen einsi, ginsi und linsi werden paarweise Alignments verwendet; s.u.). Anhand dieser Distanzmatrix wird über eine modifizierte UPGMA-Variante ein Guide-Tree erstellt, der als Grundlage für ein progressives Alignment dient. Hiermit endet das Programm bei Wahl der Option FFT-NS-1. In einem optional folgenden, zweiten progressiven Schritt wird aus dem berechneten Alignment erneut ein Guide-Tree erstellt, welcher für ein Re-Alignment genutzt wird (Option FFT-NS-2; hier mit Kürzel ffnts geführt). Hiermit sollen die durch den zunächst nur approximativ bestimmten Guide-Tree induzierten Fehler behoben werden.

Weiterhin sind in MAFFT eine Reihe iterativer Ansätze implementiert (die entsprechenden Optionskürzel enden auf „i“; hier beispielsweise fftnsi). Hierbei wird ähnlich PRRN die gewichtete Sum-of-Pairs („Weighted Sum-of-Pairs“, auch WSP) als Bewertungsfunktion eingesetzt. Bei der iterativen Verbesserung wird, ähnlich zu MUSCLE, der Guide-Tree neu aufgespalten und die zu den entstehenden Sub-Bäumen gehörigen Profile neu aligniert („Tree-Dependent Restricted Partitioning“; siehe auch Abschnitt 3.1.9).

In den progressiven Alignment-Schritten werden die Gruppen mit Hilfe der FFT aligniert. Dafür wird im Falle von Proteinen ein Vektor der physikochemischen Eigenschaften (Polarität und Volumen) und im Falle von Nukleotiden ein Vektor der Häufigkeiten der einzelnen Nukleotide verwendet. Die FFT-Analyse ergibt Peaks, welche die Verschiebung zwischen homologen Blöcken repräsentiert. Diese homologen Blöcke oder Segmente werden in eine Homologie-Matrix eingetragen, aus welcher per dynamischer Programmierung eine optimale Anordnung der homologen Segmente extrahiert wird. Durch die Präprozessierung ist der Suchraum in der Matrix drastisch reduziert.

Wie bei CLUSTALW wird eine Gewichtung der Sequenzen vorgenommen. Allerdings findet im Gegensatz zu CLUSTALW eine Veränderung der Bewertungsfunktionen bzw. der Substitutionsmatrix (in Katoh *et al.*, 2002, „Similarity Matrix“ genannt) und der Gap-Kosten statt. So enthält die Substitutionsmatrix nicht nur positive Werte, obwohl dies als optimal für den Needleman-Wunsch-Algorithmus gilt (Needleman & Wunsch, 1970; Vogt *et al.*, 1995) und deshalb beispielsweise in CLUSTALW auch so eingesetzt wird. Stattdessen wird eine anhand der Häufigkeiten der Reste normalisierte Matrix verwendet. In die Formel für diese Normalisierung (siehe Katoh *et al.*, 2002) geht ein weiterer Faktor ein, der als eine Art Gap-Extension-Faktor dient (Kommandozeilen-Parameter `ep`). Die Werte dieser Matrix beruhen dabei auf PAM-Matrizen,

[3] http://www.biophys.kyoto-u.ac.jp/ katoh/programs/align/mafft/

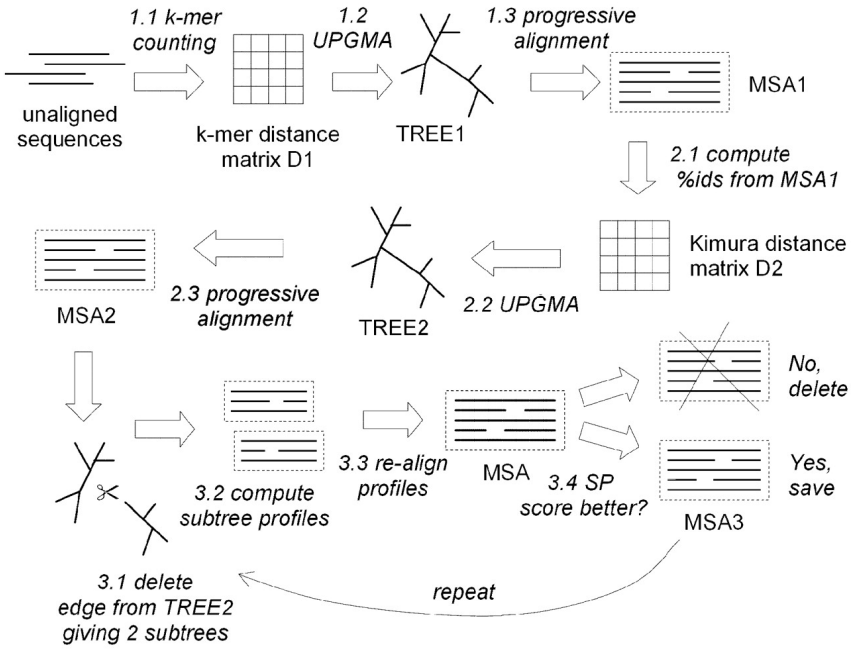


Abbildung 3.2: Vorgehensweise von MUSCLE. Eine Beschreibung befindet sich im Text. Entnommen aus Edgar (2004b).

wobei diese für Nukleinsäuren anhand des Zwei-Parameter-Modells nach Kimura (1980) berechnet wurden.

Die Gap-Parameter und Substitutionsmatrix wurden in Version 5 des Programms (Katoh *et al.*, 2005) verbessert. Weiterhin kamen Konsistenz-basierte Varianten des Programms hinzu, so beispielsweise G-INS-i (hier mit dem Kürzel ginsi geführt) und L-INS-i (hier linsi). Diese erstellen die initiale Distanz-Matrix durch ein paarweises Alignment (statt dem oben erwähnten k-mer-Counting), welches auf lokale (linsi) oder globale (ginsi) Art berechnet wird. Während der Konstruktion des multiplen Alignments und der anschließenden Iteration werden die Informationen des paarweisen Alignments berücksichtigt (ähnlich T-COFFEE), indem eine Konsistenz-Wertung in die gewichtete Sum-of-Pairs-Bewertungsfunktion einfliesst.

3.1.9 MUSCLE

MUSCLE (Edgar, 2004a,b) aligniert ähnlich wie MAFFT die Sequenzen im ersten Schritt über einen groben, aber schnellen progressiven Schritt. Hier wird dieses Alignment dann durch einen erneuten progressiven Schritt und anschließende Iteration verfeinert. Die Strategie ist in Abbildung 3.2 dargestellt und im Folgenden kurz beschrieben: Im ersten Schritt, dem groben progressiven Alignment („Draft Progressive Alignment“; Punkte 1.1 – 1.3 in vorgenannter Abbildung) wird mit Hilfe des sogenannten „k-mer-Countings“, welche vereinfacht gesagt identische Substrings (k-Tupel) zwischen den Sequenz-Paaren bestimmt, eine Distanz-Matrix berechnet (ähnlich zu MAFFT). Anhand dieser wird per UPGMA-Clustering (alternativ auch per Neighbour-Joining) ein Guide-Tree (TREE1) erstellt, der wiederum für ein progressives Alignment (MSA1) genutzt wird.

Im zweiten Schritt (Punkte 2.1 – 2.3 in der Abbildung) findet ein verbessertes progressives Alignment statt, um die durch den (höchstwahrscheinlich suboptimalen) Baum aus Schritt 1 induzierten Fehler zu beheben. Hierzu wird der Baum über die Kimura-Distanz neu berechnet (TREE2), wobei aus Laufzeitgründen nur solche Sub-Bäume neu berechnet werden, die sich relativ zum ersten Baum geändert haben.

Schließlich wird eine iterative Verbesserung durchgeführt (Punkte 3.1 – 3.4 in der Abbildung). Hierfür wird aus dem zuvor erstellten Guide-Tree (TREE2) eine Kante ausgesucht und durch Löschen derselben zwei Sub-Bäume erstellt, deren zugehörige Profile re-aligniert werden (Variante des sogenannten „Tree-Dependent Restricted Partitioning“). Sollte das neue Alignment eine höhere Bewertung erhalten, wird es beibehalten und nun der daraus generierte Baum in einer erneuten Iteration wie beschrieben partitioniert. Ansonsten wird das Alignment verworfen und der zuvor verwendete Baum weiter genutzt. Dieser Vorgang wird wiederholt bis ein vorgegebener Schwellenwert oder Konvergenz erreicht wird, also keine weitere Verbesserung möglich ist.

Ein weiterer neuer Ansatz von MUSCLE ist die „Log-Expectation Score“, welche zur Bewertung der Profil-Alignments herangezogen wird, bei Nukleotid-Alignments jedoch laut MUSCLE-Handbuch[4] geht meist schief wegen sonderzeichen keine Anwendung findet.

3.1.10 PCMA

PCMA (Pei *et al.*, 2003) ist ein Akronym für „**P**rofile **C**onsistency **M**ultiple Sequence **A**lignment“. Da die Alignment-Qualität grundsätzlich von der Diversität der zu alignierenden Sequenzen abhängt (Thompson *et al.*, 1999a), werden in PCMA während des progressiven Alignments je nach Diversität des gerade zu alignierenden Profils unterschiedliche Strategien angewendet und so ein multiples Sequenz-Alignment erzeugt.

Dabei werden im ersten Schritt sehr ähnliche Sequenzen analog zur Strategie von CLUSTALW global zu Gruppen/Profilen aligniert. Der Benutzer gibt dabei an, bis zu welchem Schwellenwert die prä-alignierten Gruppen mit Hilfe von CLUSTALW aligniert werden sollen (Parameter `ave_grp_id`, siehe auch Abschnitt 2.2.1). So entstehen schließlich mehr oder weniger divergente (prä-alignierte) Profile.

Diese werden im zweiten Schritt mit Hilfe einer Strategie ähnlich der von T-COFFEE (siehe Abschnitt 3.1.18) auf Konsistenz getestet. Zudem wird ebenfalls analog zu T-COFFEE eine paarweise globale und lokale Alignment-Bibliothek aufgebaut und wie dort erweitert. Die lokalen Profil-Alignments werden anhand einer neuen Bewertungsfunktion namens COMPASS bewertet, die ähnlich dem PSI-BLAST-Ansatz funktioniert (siehe Pei *et al.*, 2003, für eine genauere Erklärung).

[4] http://www.drive5.com/muscle/muscle.html

3.1.11 PMCOMP und PMMULTI

Ähnlich DYNALIGN und FOLDALIGN (siehe Abschnitt 3.1.5 bzw. 3.1.6) ist PMCOMP (Hofacker *et al.*, 2004) ein echtes Struktur-Alignment-Programm und basiert auf einer Variante des Sankoff-Algorithmus (Sankoff, 1985). Um zu vermeiden, dass Alignment und RNA-Struktur simultan berechnet werden müssen, aligniert PMCOMP stattdessen Basenpaarungsmatrizen, die zuvor mit Hilfe des McCaskill-Algorithmus (McCaskill, 1990) bzw. RNAFOLD (Hofacker *et al.*, 1994; Hofacker, 2003) berechnet werden.

Neben der Einschränkung auf das paarweise Alignment besteht eine weitere Vereinfachung zum Sankoff-Algorithmus darin, dass statt der dort genannten thermodynamischen Modelle auf das Pendant des Nussinov-Algorithmus (Nussinov *et al.*, 1978) zurückgegriffen wird, wobei allerdings die thermodynamischen Parameter der Basenpaarungsmatrizen verwendet werden. Für die Rekursionsformel der dynamischen Programmierung sei auf Hofacker *et al.* (2004) verwiesen. In die Bewertungsfunktion von PMCOMP fließen gewichtete Substitutionswerte für ungepaarte Basen, Basenpaare und lineare Gapkosten ein. Bei der in dieser Arbeit verwendeten Version wird der Sequenz-Anteil jedoch (noch) ignoriert. Damit tragen ungepaarte Basen nicht zur Bewertung bei, was u. a. zur Folge hat, dass Gaps innerhalb von ungepaarten Bereichen willkürlich angeordnet sind. PMCOMP benötigt $\mathcal{O}(l^4)$ Speicher und $\mathcal{O}(l^6)$ Operationen bei Sequenzen der Länge l.

Da sich entsprechend dem Profil eines progressiven Alignments eine Konsensus-Basenpaarungsmatrix definieren lässt, können die Ideen von PMCOMP auf ein progressives, multiples Alignment übertragen werden. Hierbei müssen dann zunächst alle (bei N Sequenzen) $N(N-1)/2$ möglichen Paare aligniert werden, um daraus einen Guide-Tree zu erstellen. Dies ist aufgrund der hohen Komplexität von PMCOMP sehr zeitaufwendig. Stattdessen wurde eine schnellere Variante entwickelt [hier PMCOMP (fast) genannt], welche die Basenpaarungsmatrizen – ähnlich zu Bonhoeffer *et al.* (1993) und STRAL (siehe Abschnitt 3.1.17) – zu einem Vektor kondensiert. Dieser Vektor enthält für jedes Nukleotid die aufsummierten Wahrscheinlichkeiten mit einem Nukleotid „upstream“ (Richtung 5’-Ende der Sequenz) zu paaren ($p^<$), mit einem Nukleotid „downstream“ (Richtung 3’) zu paaren ($p^>$), oder ungepaart zu sein (p^0). Die entstehenden Profile (im folgenden Beispiel A und B genannt) lassen sich mit folgender Bewertungsfunktion ähnlich zum üblichen Sequenz-Alignment mit quadratischer Komplexität alignieren:

$$\rho = \sqrt{p_A^> p_B^>} + \sqrt{p_A^< p_B^<} + \sqrt{p_A^0 p_B^0} \ . \tag{3.2}$$

Sind alle paarweisen Vergleiche auf diese Weise berechnet, muss die abgewandelte Version des Sankoff-Algorithmus nur noch auf alle $N-1$ Profile angewendet werden. Diese Idee des multiplen Alignments ist in PMMULTI implementiert. Das Programm wurde jedoch in dieser Arbeit nicht getestet.

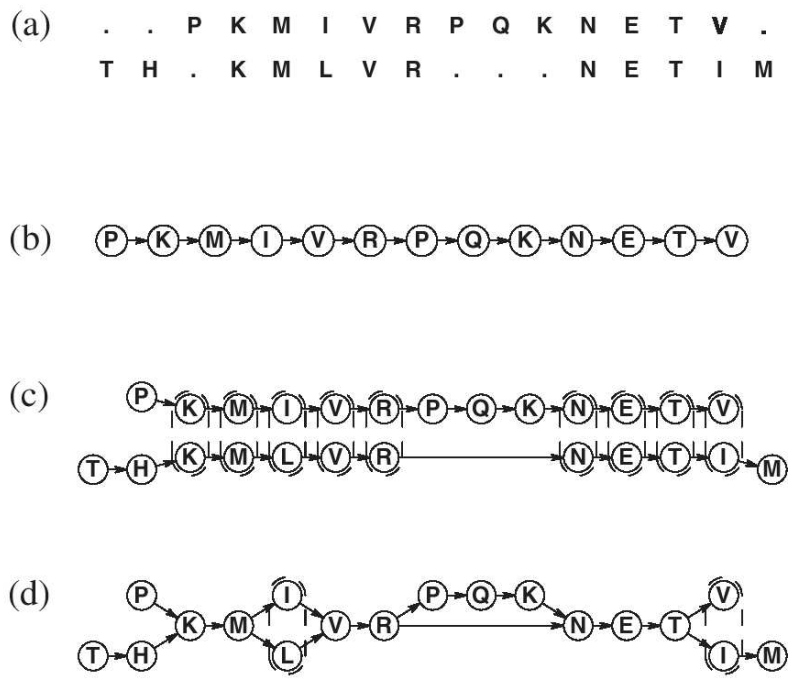


Abbildung 3.3: PO-MSA Datentyp verwendet in POA. A: Typische (degenerative) Darstellung eines (paarweisen) Alignments in Reihen und Spalten. **B:** Eine einzelne Sequenz im PO-MSA Format. **C:** Zwei alignierte Sequenzen im PO-MSA-Format. **D:** PO-MSA-Darstellung des paarweisen Alignments. Identisch alignierte Zeichen werden zu einem Knoten zusammengefügt. Entnommen aus Lee *et al.* (2002).

3.1.12 POA

POA (Lee *et al.*, 2002) verwendet im Gegensatz zu den üblichen progressiven Alignment-Methoden keine Profile. Laut den Autoren stellen diese Profile, welche im progressiven Alignment für die dynamische Programmierung nötig sind, ein Problem dar, da die Erstellung derselben zu einem Informationsverlust führt. So lassen sich mehrere unterschiedliche Alignments (Beispiel gemischte Spalten) in einem Profil zusammenfassen (weshalb sich auch ein Alignment anhand seines Profils nicht eindeutig rekonstruieren lässt). Zwar sind in einem solchen Profil die Häufigkeiten aller Reste bekannt, es lässt sich aber beispielsweise nicht mehr feststellen, von welcher Sequenz ein bestimmter Rest kommt, wodurch wiederum die Bewertung von Gaps schwierig wird. Die typische, „degenerative“ Darstellung eines Alignments als Buchstaben in Reihen und Spalten und die damit verbundenen Probleme sollen in POA durch Verwendung eines neuen Datentyps vermieden werden. Dieser Datentyp wird PO-MSA genannt („Partial Order Multiple Sequence Alignment“). Ein einfaches Beispiel ist in Abbildung 3.3 gezeigt. In diesem Datentyp lässt sich die Information eines Alignments ohne Verlust speichern, es lässt sich ein eindeutiges Alignment aus ihm extrahieren (*vice versa*) und diese Datentypen lassen sich direkt, ohne Verwendung von Profilen, alignieren. Dazu wurden die Standard-Alignment-Algorithmen nach Needleman & Wunsch (1970) und Smith & Waterman (1981) in Lee *et al.* (2002) entsprechend erweitert. Eine weitere Besonderheit ist, dass in POA zusätzlich zu den üblichen Edit-Operationen Insertion, Deletion und Substitution die „Homologe Rekombination“ implementiert ist.

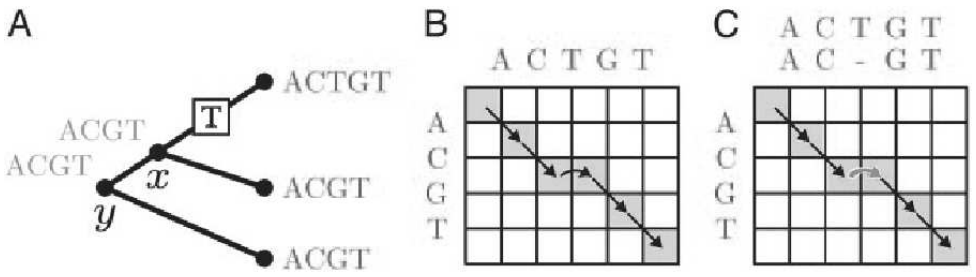


Abbildung 3.4: Vermeidung des „Über-Alignments“ (Insertions-Korrektur) durch PRANK. A: Guide-Tree für das progressive Alignment. Das eingerahmte T markiert eine Insertion. **B** und **C** zeigen die Schritte der dynamischen Programmierung, welche an x und y (siehe **A**) geschehen. Der graue Pfeil (**C**) markiert die Stelle, an der die Insertion bereits bestraft wurde und in PRANK im Gegensatz zu anderen Methoden nicht nochmals gezählt wird. Entnommen aus Higgins *et al.* (2005).

3.1.13 PRANK

PRANK (Löytynoja & Goldman, 2005) ist ein Akronym für „**Pr**obabilistic **A**lignment **K**it“. Es nutzt ein paarweises Hidden-Markov-Modell (HMM; siehe Durbin *et al.*, 1998, und Anmerkung in Abschnitt 3.1.7) mit einer probabilistischen/evolutionären Bewertungsfunktion zur Erstellung des multiplen Alignments.

Das Besondere an PRANK ist, dass es während des progressiven, multiplen Alignment-Schrittes versucht, zwischen Insertionen und Deletionen zu unterscheiden, was andere progressive Alignment-Methoden nicht tun (siehe Abbildung 3.4 für ein Beispiel). Einzelne Insertions-Ereignisse, die in frühen Phasen des progressiven Alignments stattfinden, werden von anderen progressiven Methoden, beispielsweise CLUSTALW, zu einem späteren Zeitpunkt erneut bestraft, da hier die Insertionen in alle (Sub-)Alignments oder Profile eingefügt werden müssen, die während des progressiven Alignments vereint werden. So kommt es zu einer Mehrfachbestrafung von Insertionen (siehe auch Higgins *et al.*, 2005). Die Autoren sprechen von einem „Über-Alignment“ der anderen progressiven Ansätze, was zu kompakten, ansehnlichen – da in Blöcken strukturierten – Alignments führt und zeigen, dass ihr Programm bei Anwendung auf genomische Regionen mit vielen Insertionen „bessere“ Alignments erzeugt, die phylogenetisch konsistent, aber weniger „dicht“ sind, heißt mehr Gaps enthalten.

3.1.14 PROALIGN

PROALIGN (Löytynoja & Milinkovitch, 2003) ist ein probabilistisches Alignment-Programm, das ein paarweises Hidden-Markov-Modell (siehe Anmerkung in Abschnitt 3.1.7) mit einem progressiven Algorithmus und einem evolutionären Modell verbindet, welches den Substitutionsprozess der Reste beschreibt. Dabei wurden die Ideen zum HMM-Alignment aus Durbin *et al.* (1998) aufgegriffen. Programm-intern werden Sequenzen als Vektoren von Übergangszuständen zwischen Resten dargestellt und jedes paarweise Alignment rekonstruiert die Vorgängersequenz anhand des gegebenen evolutionären Modells. Hierdurch kann progressiv ein multiples Alignment erstellt werden.

Eine Besonderheit an PROALIGN ist, dass die Parameter des Programms anhand einer hohen Zahl mit Hilfe von ROSE (Stoye *et al.*, 1998) erzeugter Alignments trainiert wurden.

3.1.15 PRRN

PRRN (Gotoh, 1996, 1999) kann (wie das ehemalige Protein-spezifische Pendant PRRP) als das Standard-Programm für iteratives Alignment bezeichnet werden. Das Programm besteht seit über zehn Jahren und wurde wie CLUSTAL mehrfach verbessert (siehe auch Referenzen in den genannten Publikationen).

Um ein initiales (progressives) Alignment zu verbessern, nutzt PRRN dabei wiederholt paarweise Gruppen-Alignments, um die gewichtete Sum-of-Pairs („Weighted Sum-of-Pairs“, auch WSP) zu optimieren. Dabei wird eine sogenannte doppelt verschachtelte, randomisierte iterative Methode genutzt („DNR“ nach Gotoh, 1996). Die innere Iteration optimiert die Sum-of-Pairs (SOP), während die äußere Iteration die Gewichte optimiert, die aus einem phylogenetischen Baum bestimmt werden, der anhand des bereits bestehenden Alignments konstruiert wurde.

3.1.16 STEMLOC

STEMLOC (Holmes, 2004, 2005) ist ein weiteres echtes RNA-Struktur-Alignment-Programm. Es basiert auf paarweisen SCFGs („Stochastic Context-Free Grammars“; siehe beispielsweise Durbin *et al.*, 1998). Wie andere Struktur-Alignment-Programme vereinfacht es den Sankoff-Algorithmus (Sankoff, 1985) zum einen durch Reduktion auf das paarweise Alignment und hier zudem durch Anwendung von Heuristiken, wie den sogenannten „Envelopes“ (auch „Go-Faster Stripes“) die im Wesentlichen den Suchraum des Algorithmus einschränken (siehe angegebene Publikationen und die STEMLOC-Homepage[5] geht meist schief wegen sonderzeichen für Details).

3.1.17 STRAL

STRAL (Dalli, 2006) verbindet Struktur- und Sequenz-Alignment und erstellt auf progressive Art und Weise ein multiples RNA-Alignment. Dabei werden die Ideen aus Bonhoeffer *et al.* (1993) und Yang & Blanchette (2004) aufgegriffen, in denen (ähnlich zu der „schnellen“ paarweisen Alignment-Variante von PMCOMP/PMMULTI; siehe Abschnitt 3.1.11) Basenpaarungsmatrizen zu Vektoren kondensiert werden, die für jedes Nukleotid die Wahrscheinlichkeit enthalten „downstream“ gepaart vorzuliegen (p^1), „upstream“ gepaart vorzuliegen (p^2) und nicht gepaart zu sein (p^0). Diese Information wird zusammen mit einem Sequenz-Anteil in der Bewertungsfunktion von STRAL verwendet, um das Alignment zweier Basen i und k aus den Sequenzen A und B zu bewerten:

$$s_{i,k} = \alpha \left(\sqrt{p^1_{A_i} p^1_{B_k}} + \sqrt{p^2_{A_i} p^2_{B_k}} \right) + \sqrt{p^0_{A_i} p^0_{B_k}} \cdot d(A_i, B_k) \tag{3.3}$$

[5] http://biowiki.org/StemLoc

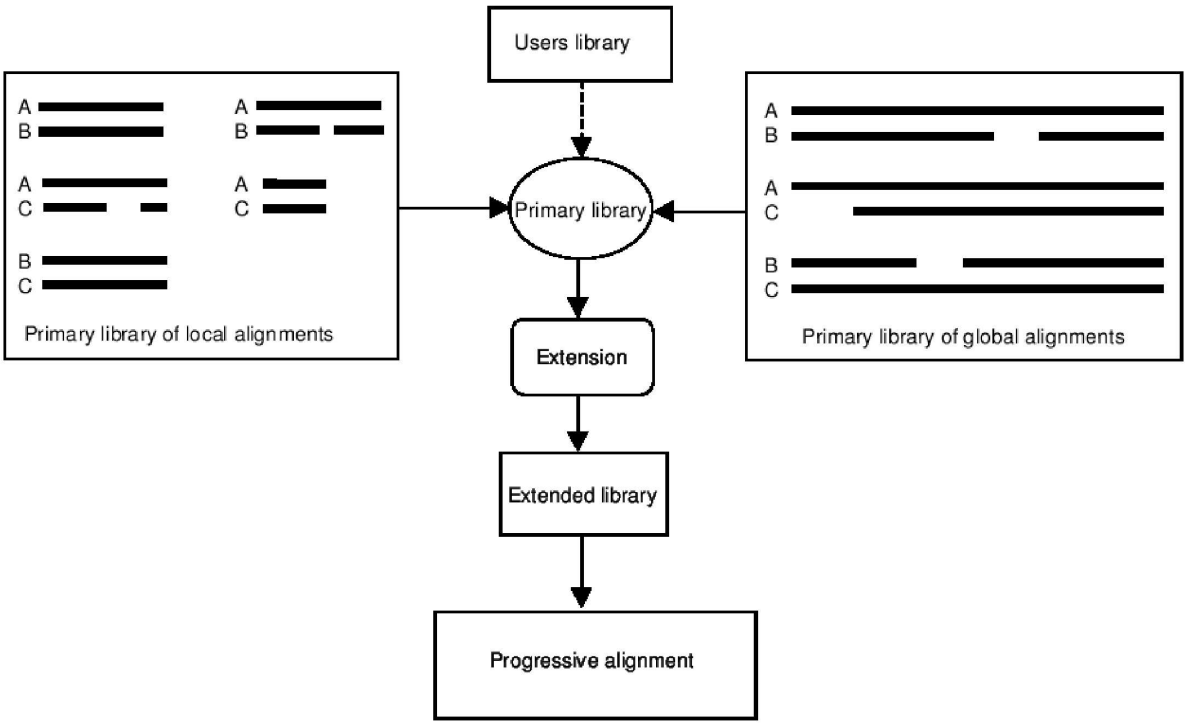


Abbildung 3.5: Vorgehensweise von T-COFFEE. Zunächst werden paarweise Alignments aus unterschiedlichen Quellen (hier paarweise globale und lokale Alignments) in einer primären Bibliothek vereint. Diese wird erweitert (siehe Text), um in dem sich anschließenden progressiven Alignment genutzt zu werden. Entnommen aus Notredame (2002).

Der Faktor α bestimmt hierbei die Gewichtung des Struktur-Anteils über den Sequenz-Teil, der durch die Substitutionsmatrix d gegeben ist. Als Standard werden die RIBOSUM-Matrizen verwendet. Das multiple Alignment wird, wie in anderen progressiven Alignment-Methoden, durch Vergleich aller Paare, Konstruktion eines Guide-Trees anhand der während des paarweisen Vergleichs ermittelten Distanzen und schließlich einem Profil-Alignment entlang der Verzweigungen des Guide-Trees erstellt.

3.1.18 T-COFFEE

T-COFFEE („**T**ree-based **C**onsistency **O**bjective **F**unction **f**or Alignm**e**nt **E**valuation“; Notredame *et al.*, 2000) versucht die in Abbildung 1.7 gezeigten typischen Fehler des multiplen Alignments ohne eine nachträgliche iterative Verfeinerung zu umgehen. Hierzu wird eine paarweise Bibliothek aus Alignments erstellt und im späteren Alignment-Prozess nach maximaler Konsistenz mit den Alignments in der Bibliothek gesucht (siehe Abbildung 3.5).

Die Bibliothek kann grundsätzlich aus jeder Art von paarweisen Alignments erstellt werden. In der Standard-Einstellung werden globale Alignments mit Hilfe von CLUSTALW und lokale Alignments mit Hilfe von LALIGN erzeugt. Diese Bibliothek enthält damit paarweise Reste-Übereinstimmungen, die in einem Folgeschritt per Sequenz-Identität gewichtet und vereint werden. Die jetzt entstandene primäre Bibliothek wird dann erweitert, indem die Konsistenz eines jeden Reste-Paares mit denen der anderen Alignments überprüft und wiederum entsprechend

gewichtet wird. Die Idee dabei ist, dass das endgültige Gewicht eines Paares auch Informationen aus der gesamten Bibliothek enthalten soll. Die Alignments basieren damit nicht nur darauf, wie zwei Sequenzen zueinander alignierbar sind, sondern auch inwiefern dieses Paar kompatibel mit dem Rest der Bibliothek ist. Die Bewertung der Alignments aus Sequenzen oder Sequenz-Gruppen im sich anschließenden progressiven Alignment findet anhand dieser erweiterten Bibliothek und der sogenannten COFFEE-Bewertungsfunktion (Notredame *et al.*, 1998) statt. T-COFFEE ist damit ein progressives, Konsistenz-basiertes Alignment-Programm.

3.2 Programmfehler und zu berücksichtigende Eigenarten

Eine unerwartete Schwierigkeit stellte die Benutzung eines Großteils der in Abschnitt 3.6 und 3.7 verwendeten Programme dar. Zum Teil ließen sich die Programme nur durch Anpassungen des Quellcodes installieren. Weiterhin war die Auswahl der Parameter und korrekte Benutzung der Programme meist erst nach Rücksprache mit den Autoren sinnvoll möglich, selbst dann wenn Handbücher oder Anleitungen vorhanden waren. Schließlich verändern einige Programme die Sequenz-IDs oder sogar die Sequenzen selbst, was die anschließende Bewertung ohne Gegenmaßnahmen unmöglich machte. Manche Programme stürzen bei Anwendung auf einige der im Folgenden verwendeten Alignments reproduzierbar ab. Aus den genannten Gründen war es nötig, für jedes Alignment-Programm ein eigenes Helfer-Skript („Wrapper“) zu schreiben, welches die (zum Teil sehr spezielle) Aufarbeitung der Eingabe-Daten, den korrekten Aufruf des Alignment-Programms, sowie die Reformatierung der Ausgabe sicherstellte.

Im Zuge dessen entstand unter anderem die Tcl-Bibliothek SQUICL. SQUICL ist eine in Tcl und C programmierte Bibliothek, die u. a. als Frontend zu der SQUID-Bibliothek (Eddy, 2005) und der RNA-Bibliothek (auch RNALIB genannt) des Vienna-RNA-Packets (Hofacker *et al.*, 1994; Hofacker, 2003) dient. Eine komplette Kommandoreferenz der Version 0.3.0 befindet sich auf Seite 111 im Anhang.

Im Folgenden werden einige der Eigenarten und Fehler der Alignment-Programme aufgelistet, wobei aus Platzgründen nicht auf Besonderheiten bei der Eingabe- und Ausgabe-Formatierung eingegangen wird.

- ALIGN-M: Für ALIGN-M stand zu Beginn dieser Arbeit keine Substitutionsmatrix für RNA zur Verfügung (obwohl zwingend notwendig). Nach Rücksprache mit dem Autor wurde eine solche erstellt und ist im Programmpaket (nur als binäre Distribution erhältlich) nun als `RNA2` enthalten. Sind in den Sequenzen Zeichen vorhanden, die nicht in der vorgegebenen Substitutionstabelle enthalten sind, so kommt es zum Programmabbruch. Der IUPAC-Mehrdeutigkeitscode wird nicht vollständig unterstützt. Sind Leerzeichen oder Bindestriche in Sequenz-Namen enthalten, bricht das Programm die Ausführung ab.
- DIALIGN: Hier musste die Option `-n` verwendet werden, um eine Umwandlung in Proteine zu verhindern.

- DYNALIGN: Für DYNALIGN musste der „Maximum Separation Parameter“ M (siehe Mathews & Turner, 2002) dynamisch berechnet werden, da er der Längen-Differenz der zu alignierenden Sequenzen angepasst werden muss und gleichzeitig wegen enormen Einfluss auf den Speicherverbrauch nicht zu groß gewählt werden darf.
- HANDEL: Das Programm wandelt im Sequenznamen Unterstriche in Bindestriche um.
- POA: Das Programm verändert den IUPAC-Mehrdeutigkeitscode. Beispielsweise findet ein Umwandlung von M nach A und von R nach A statt.
- PROALIGN: PROALIGN bedarf einer Anpassung der „Suchbandbreite“, da es ansonsten zu einem Programmabbruch kommt. Für die in dieser Arbeit verwendeten Alignments genügte die Einstellung `-bwidth=400`. Sind Bindestriche in Sequenznnamen enthalten, werden diese gelöscht.
- PRRN: Das erste Zeichen eines Sequenz-Dateinamens darf keine Zahl sein.
- STEMLOC: Bei der Verwendung von STEMLOC ist nicht garantiert, dass ein Alignment erzeugt wird. In den meisten Fällen hilft ein Anheben des Parameters `nfold`. So konnten alle Sequenzen aus dem Struktur-Alignment-Datensatz erst mit einem Wert von 110, statt dem vom Autor ursprünglich in der Option `-fast` vorgegebenen 100 erreicht werden.
- T-COFFEE: Vor der Version 2 war eine Umwandlung der Sequenzen in DNA zwingend nötig, da das Programm sonst abstürzte.

3.3 CONSTRUCT

3.3.1 Idee

CONSTRUCT wurde ursprünglich zur Thermodynamik-basierten Vorhersage konservierter Sekundärstrukturen entwickelt (Lück, 1997; Lück *et al.*, 1999, 1996). Daher erklärt sich auch der Name, welcher ein Akronym für „**Construct**ion of **Con**sensus **Struct**ures“ ist.

Das Programm (oder besser Programmpaket) beruht auf einer Kombination aus Thermodynamik (Basenpaarungswahrscheinlichkeiten), Sequenz-Alignment, gegenseitigem Informationsgehalt (als Maß für kompensatorische Basenpaaraustausche), sowie der Intelligenz des Benutzers. Die grundlegende Idee ist folgende: Zunächst wird ein Sequenz-Alignment homologer RNAs erstellt (beispielsweise mit Hilfe von CLUSTALX). Dann wird für jede einzelne Sequenz die Sekundärstrukturverteilung (basierend auf dem Algorithmus von McCaskill, 1990, implementiert beispielsweise in RNAFOLD) bestimmt, welche sich in Form eines Dotplots (Tinoco *et al.*, 1971) visualisieren lässt (siehe auch Abbildung 1.9). Fügt man nun die Gaps aus dem Alignment in die Basenpaarungsmatrizen ein, so erhalten sie alle die identische Dimension und lassen sich übereinanderlagern. Konservierte Sekundärstrukturelemente sollten nun übereinander zu liegen kommen, wenn sie zuvor korrekt aligniert waren. Summiert man also die einzelnen Matrizen, so ergibt sich ein thermodynamischer Konsensus-Dotplot, aus welchem sich per dynamischer Programmierung (ähnlich dem Nussinov-Algorithmus zur Basenpaarmaximierung, siehe Nussinov *et al.*, 1978) eine optimale Konsensus-Sekundärstruktur vorhersagen lässt. In

vielen Fällen sind jedoch gerade konservierte Sekundärstrukturelemente im initialen Sequenz-Alignment nicht korrekt aligniert worden. Ein einfaches Beispiel sind extrastabile Tetraloops, die aufgrund ihrer Strukturhomologie aligniert werden sollten, jedoch aufgrund ihrer möglicherweise divergenten Sequenz (UNCG, GNRA, ...) nicht von einem Sequenz-Alignment als homolog identifiziert werden können. Da diese Alignment-Fehler in dem Konsensus-Dotplot von CONSTRUCT sehr leicht identifizierbar sind, wurde dem Programm ein Alignment-Editor hinzugefügt. Dieser erlaubt eine einfache Alignment-Korrektur durch den Benutzer, da die Auswirkung jeder Änderung im Alignment direkt in der Dotplot-Darstellung sichtbar wird.

3.3.2 Vorgehensweise

CONSTRUCT wurde u. a. im Rahmen dieser Arbeit stark ausgebaut, weshalb im Folgenden der Programm-Ablauf der aktuellen Version beschrieben wird (siehe hierfür Abbildung 3.6). Auch wenn sich dieser von der aktuellsten Veröffentlichung (Lück *et al.*, 1999) unterscheidet, sei hiermit auch ausdrücklich auf die detaillierte Beschreibung dort und in Steger (2003) verwiesen.

1. Zunächst wird vom Benutzer ein initiales Sequenz-Alignment (mit beispielsweise CLUSTALX) berechnet.
2. Dann wird für jede einzelne Sequenz eine thermodynamische Basenpaarungsmatrix erstellt. Dies geschieht durch Verwendung des Programms CS_FOLD (ehemals CS_MAKE), welches als Frontend für RNAFOLD (Hofacker *et al.*, 1994; Hofacker, 2003) dient.
3. Die Gaps des in Schritt 1 erstellten Alignments werden in die Basenpaarungsmatrizen eingefügt, womit gleich große Matrizen entstehen, die im GUI („Graphical User Interface“; graphische Benutzeroberfläche) übereinandergelagert dargestellt werden. Durch Addition der Matrizen entsteht ein Konsensus-Dotplot. Die Wahrscheinlichkeiten der Konsensus-Basenpaare werden dabei so berechnet, dass „Hintergrundrauschen“ durch vereinzelte Basenpaare verhindert wird (siehe Abschnitt 3.3.3 und Lück *et al.*, 1999). Weiterhin können den Sequenzen Gewichte zugeordnet werden, um eine Überrepräsentation einzelner Familien zu vermeiden.
4. Der gegenseitige Informationsgehalt (siehe Abschnitt 3.3.4), der als Maß für kompensatorische Basenpaaraustausche dient, wird in der linken unteren Hälfte des GUI dargestellt. Grenzwerte zur Unterdrückung des typischen statistischen Rauschens können vom Nutzer gewählt werden (beides implementiert durch Riks, 2001).
5. Das GUI besteht aus minimal zwei Fenstern: dem Konsensus-Dotplot, sowie dem zugehörigen Alignment-Editor. Im Konsensus-Dotplot sind strukturell misalignierte Positionen leicht erkennbar und im Alignment-Editor lassen sich eben diese korrigieren, wobei die Veränderungen sofort im Konsensus-Dotplot-Fenster dargestellt werden. Die Optimierung eines Strukturelementes ist in Steger (2003, Kapitel 5) beispielhaft gezeigt.

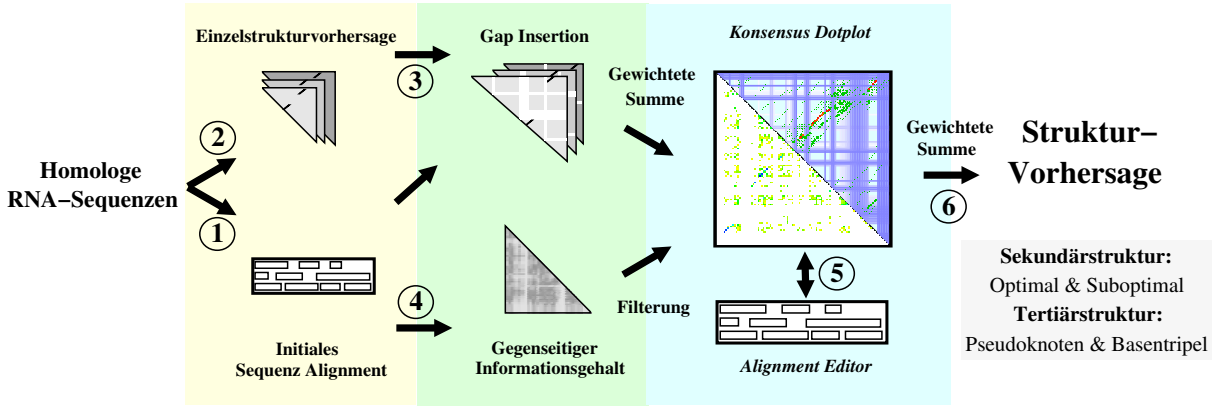


Abbildung 3.6: Ablaufschema des Programmpaketes CONSTRUCT. Für Details siehe Text. Der nur einmal zu Beginn des Ablaufes ausgeführte zeitaufwendige Teil, welcher die Berechnung der Basenpaarungsmatrizen (Schritt 2) sowie die Erstellung des Sequenz-Alignments (Schritt 1) betrifft, ist gelblich hinterlegt. Die Insertion der Gaps in die Basenpaarungsmatrizen sowie die Berechnung des gegenseitigen Informationsgehalts erfolgt für den Benutzer transparent (Schritt 3 und 4; grünlich hinterlegt). Das eigentliche GUI ist im bläulich hinterlegten Teil zu erkennen. Spätestens durch wiederholte Korrektur der misalignierten Bereiche im Alignment-Editor (Schritt 5) entsteht im Konsensus-Dotplot eine prominente Konsensus-Struktur. Die verschiedenen Varianten der Strukturvorhersage (Schritt 6) erfolgen wie im Text beschrieben.

---

6. Die Struktur-Vorhersage erfolgt auf Basis einer vom Benutzer bestimmten Linearkombination aus Thermodynamik und gegenseitigem Informationsgehalt, wobei Gewichtungsfaktoren sowie weitere Grenzwerte und Filter benutzt werden können. Die Vorhersage beinhaltet optimale Konsensus-Sekundärstrukturen (nach Nussinov *et al.*, 1978), suboptimale Konsensus-Sekundärstrukturen nach Steger *et al.* (1984) bzw. Zuker (1989) (implementiert durch Riks, 2001), sowie die Vorhersage tertiärer Wechselwirkungen in Form von Pseudoknoten und Basentripeln durch sogenannte maximal gewichtete Zuordnungen („Maximum Weighted Matching“; Tabaska *et al.*, 1998), welche ebenfalls durch Riks (2001) implementiert wurden. Die Strukturen lassen sich in einer Reihe von Formaten anzeigen. Im Falle der „Struktur-Alignment“-Anzeige wird eine ausführliche Statistik mit ausgegeben, die u. a. eine Analyse der Vorhersage per Chi-Quadrat-Test (angewendet auf den gegenseiten Informationsgehalt) beinhaltet.

3.3.3 Thermodynamischer Konsensus-Dotplot

Im thermodynamischen Konsensus-Dotplot berechnet sich die Wahrscheinlichkeit p_c eines Konsensus-Basenpaares an der Position i, j nach:

$$p_c(i,j) = \left(\frac{\sum_{s=1}^{N} w_s \cdot p_s(i,j)^{1/a}}{\sum_{s=1}^{N} w_s} \right)^b \tag{3.4}$$

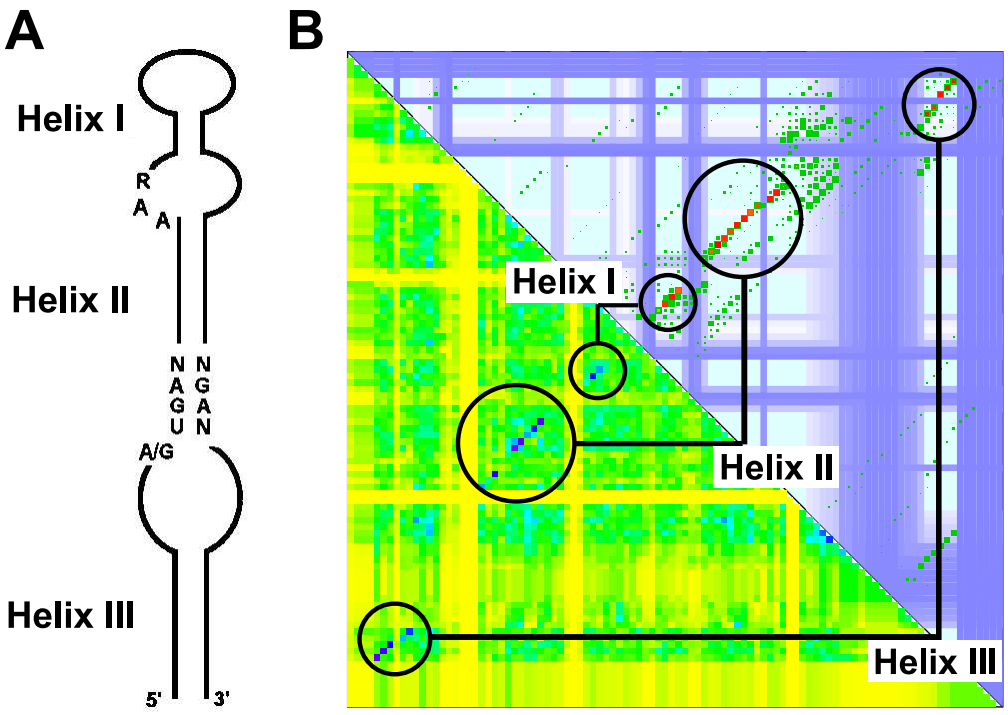


Abbildung 3.7: Sekundärstrukturen im CONSTRUCT-Dotplot. A: Konsensus-Struktur eines SECIS-Elementes (der Form 2) nach Lescure *et al.* (2000). Die unverzweigte Struktur besteht im Wesentlichen aus drei Helices, wobei Helix II durch ein Nicht-Watson-Crick-Quartett und das Triplet AAR begrenzt wird. **B**: Der CONSTRUCT-Dotplot zeigt ein Fagegaltier *et al.* (2000) entnommenes und mit CONSTRUCT korrigiertes Alignment. Alle Helices sind sowohl im thermodynamischen Konsensus-Dotplot (rechte obere Hälfte) als auch in der linken unteren Hälfte mit Darstellung des gegenseitigen Informationsgehalts zu erkennen. Die Farbkodierung ist im Text erklärt (siehe Abschnitt 3.3.3 und Abschnitt 3.3.4).

Dabei ist w_s der benutzerdefinierte Gewichtungsfaktor für Sequenz s. Hiermit lässt sich verhindern, dass beispielsweise Sequenzen einer im Alignment besonders häufig auftretenden Art den Konsensus dominieren. Die Wahrscheinlichkeit $p_s(i,j)$, dass die Nukleotide i und j in Sequenz s gepaart sind, ergibt sich direkt aus den via RNAFOLD berechneten Strukturverteilungen. Die beiden Exponenten $1/a$ und b dienen dazu, den Einfluss einzelner, aber in den anderen Strukturen nicht konservierten Basenpaarungen zu minimieren. Beide Faktoren wurden mittlerweile (empirisch) auf den Wert 3 festgelegt (fest einkodiert).

Die Konsensus-Basenpaarwahrscheinlichkeiten werden farbkodiert im thermodynamischen Konsensus-Dotplot von CONSTRUCT dargestellt (siehe Abbildung 3.7). Basenpaare einzelner Sequenzen erscheinen als grüne Quadrate, deren Fläche proportional zur Wahrscheinlichkeit $p_s(i,j)$ ist. Konsensus-Basenpaare sind je nach Anzahl beteiligter Sequenzen gelb bis rot gefärbte Quadrate, deren Fläche proportional zur Wahrscheinlichkeit $p_c(i,j)$ ist. Die beim Alignment eingefügten Gaps erscheinen als Streifen, die je Anzahl der Gaps in der betreffenden Alignment-Spalte weiß bis violett gefärbt sind.

3.3.4 Gegenseitiger Informationsgehalt

Der gegenseitige Informationsgehalt (Chiu & Kolodziejczak, 1991) wird auch „Mutual Information Content“, „Mutual Information Score“ oder „Mutual Information Statistics“ genannt und zumeist mit MI abgekürzt. Der gegenseitige Informationsgehalt $\mathrm{MI}(i,j)$ für ein Nukleotidpaar an der Position i,j ergibt sich nach:

$$\mathrm{MI}(i,j) = \sum_{\mathrm{X,Y}} f_{ij}(\mathrm{XY}) \log \frac{f_{ij}(\mathrm{XY})}{f_i(\mathrm{X}) f_j(\mathrm{Y})} \; . \tag{3.5}$$

Mit f ist die Häufigkeit des Auftretens eines Nukleotids X oder Y an der Stelle i bzw. j bezeichnet; $f_{ij}(\mathrm{XY})$ ist die Wahrscheinlichkeit des gemeinsamen Auftretens der Nukleotide X und Y an den Positionen i und j (siehe beispielsweise Kapitel 5 in Steger, 2003, für eine genaue Erklärung und Herleitung).

Die Werte werden farbkodiert von gelb über grün und blau nach rot in der linken unteren Hälfte des CONSTRUCT-Dotplots dargestellt (siehe Abbildung 3.7). Im Gegensatz zum thermodynamischen Dotplot wird der gegenseitigen Informationsgehalt aufgrund des erhöhten Rechenaufwandes nicht nach jeder Alignment-Modifikation neu berechnet.

Der Werte des gegenseitigen Informationsgehalts können durch eine graphische Methode („Color-Mapping“) gefiltert werden, um so das typische statistische Rauschen zu unterdrücken. Weiterhin wurde die Paar-Entropie-Normierung implementiert. Bei dieser wird der gegenseitige Informationsgehalt normiert, indem der Wert durch die Verbundentropie H(X,Y) geteilt wird (siehe Martin *et al.*, 2005, für eine genaue Erklärung).

3.3.5 Erweiterungen an CONSTRUCT

In meiner Diplomarbeit (Wilm, 2002) wurde der Quelltext des Programmpaketes in großen Teilen komplett überholt. So wurde die sehr langsame Vorgehensweise für den Aufbau und die Aktualisierung des GUI nach Änderung des Alignments überarbeitet, was zu einem drastischen Geschwindigkeitsgewinn führte. Für weitere Geschwindigkeitsoptimierungen wurden große Bereiche als kompilierte Erweiterungen in den (im CONSTRUCT-Paket enthaltenen) Tcl-Interpreter integriert. Jedoch konnte zum damaligen Zeitpunkt nur der Kern des Programms neu implementiert werden. Die Komplettierung der neuen Version [nun Versionsnummer 3.1, im Gegensatz zu 2.0 nach Lück *et al.* (1999) und 2.1 nach Riks (2001)] war u. a. Gegenstand dieser Arbeit.

Wie bekannte Struktur-Einschränkungen in das Programm eingebunden werden können, wird in Abschnitt 3.3.6 beschrieben. Eine kurze Liste weiterer Änderungen ist im Folgenden aufgeführt:

- **Formate:**
  Die Basenpaarungswahrscheinlichkeiten werden nun direkt aus den von RNAFOLD erzeugten PostScript-Dateien gelesen. Die ehemals mit CONSTRUCT ausgelieferte, spezielle Version von RNAFOLD namens CS_RNAFOLD, welche die Matrizen in einem speziellen Binärformat speicherte, ist somit überflüssig geworden. Zur namentlichen Abgrenzung wurde das entsprechende Frontend CS_MAKE in CS_FOLD umbenannt. Die

Konsensusstruktur-Vorhersage kann nun zusätzlich im RNAML-Format (Waugh *et al.*, 2002) und im (vor allen in der Rfam genutzten) Stockholm-Format gespeichert werden.

- **Neue Skripte:**
  Im sogenannten Connect-Format gespeicherte Strukturen lassen sich durch das Skript CS_STRUCT_DISPL, welches auf die CONSTRUCT eigenen Struktur-Visualisierungsroutinen Drawstruct und Circles zugreift, wieder anzeigen. Um zu vielen Alignments zugehörige, sogenannte Project-Dateien ohne den wiederholten Aufruf von CS_FOLD erstellen zu können, wurde das Skript CSFOLDBATCH entwickelt. Auf ähnliche Weise erlaubt das ebenfalls neue Skript CSDPBATCH das Speichern von Dotplots im Batch-Verfahren.

- **Installation:**
  Um die Installation zu vereinfachen, wurde diese über die in der Unix-Welt verbreiteten GNU-Autotools realisiert.

- **Dokumentation:**
  Dem Paket wurde eine man-page und ein (in Teilen unvollständiges) Handbuch hinzugefügt.

- **Neue Funktionen im Dotplot-Fenster:**
  Durch Klicken auf ein (Konsensus-)Basenpaar wird das Alignment-Fenster zu dem entsprechenden 5'- (linke Maustaste) oder 3'-Nukleotid (rechte Maustaste) gescrollt.
  Durch einen Klick mit der mittleren Maustaste auf ein Konsensus-Basenpaar werden Informationen zu diesem in der Konsole ausgegeben.

- **Neue Funktionen im Alignment-Fenster:**
  Die Suche nach Sequenz-Elementen per regulären Ausdrücken wurde über das Menü „Alignment / Seq Search“ realisiert. Die Suche ignoriert dabei Gaps und die gefundenen Treffer werden farblich markiert. Die Markierung bleibt auch nach Verschieben von Sequenz-Abschnitten erhalten. Da oft bestimmte Bereiche innerhalb eines Alignments über mehrere Sequenzen korrigiert werden müssen, wurde die Möglichkeit mehrere Sequenzenabschnitte zu bewegen (unter zu Hilfenahme der Strg-Taste) implementiert. Die Nukleotide im Alignment-Fenster lassen sich in ihr Helix-Äquivalent in der optimalen Sekundärstruktur umwandeln (Menü „Alignment / Map Nt to Helix“), d. h. Nukleotide der ersten Helix werden in `a`'s umbenannt, Nukleotide der zweiten Helix in `b`'s usw. Somit lassen sich u. U. bereits im Alignment-Fenster strukturelle Gruppen erkennen.

3.3.6 Berücksichtigung bekannter Struktur-Informationen

CONSTRUCT beruht auf einer Kombination mehrerer Methoden, um die Vorhersage-Qualität zu erhöhen. Um einen engeren Bezug zum Experiment herzustellen, wurde es ermöglicht, bereits bekannte Struktur-Informationen – aus beispielsweise chemischem/enzymatischem Mapping, In-Line-Probing (Soukup & Breaker, 1999), oder 3D-Strukturaufklärungen, wie NMR und Röntgenkristallographie – zu berücksichtigen.

```
begin entry
    id:      h_SelY
    weight:  0.12
    seqlen:  65
    bpmat:   h_SelY_dp.ps.gz
    foldcmd: cs_rnafold -T 37 -p -d 3
    comment: h_SelY / 65nt / weight 0.12
    mapinfo: 4-5:p 7-12:u 24-27:u 33-35:u
end entry
```

Abbildung 3.8: Beispiel eines Sequenz-Eintrages aus einer CONSTRUCT-Project-Datei. Die in einer CONSTRUCT-Project-Datei enthaltenen Sequenz-Einträge können die hier beispielhaft gezeigten Attribute aufweisen. Unter `mapinfo` lassen sich, wie im Text beschrieben, Informationen zu Basenpaarungen speichern. Im vorliegenden Beispiel sind die Nukleotide an den Positionen 4 und 5 gepaart und die Nukleotide 7–12, 24–27 und 33–35 ungepaart. Diese Information kann sowohl im CONSTRUCT-Dotplot, als auch in der Struktur-Alignment-Anzeige verwendet werden.

Hierzu wurde den Einträgen in der sogenannten Project-Datei von CONSTRUCT (welche der Kommunikation von CS_FOLD und dem Hauptprogramm CS_DP dient) ein weiteres Feld namens „mapinfo“ hinzugefügt (siehe Abbildung 3.8 für ein Beispiel). Hier lassen sich – durch ein Leerzeichen separiert – exakte Basenpaarungen oder einfache Informationen über Paarungszustände eintragen. Ist beispielsweise eine Basenpaarung zwischen Nukleotid 13 und 23 bekannt, so wird dies durch ein `13:23` repräsentiert. Ist bekannt, dass die Nukleotide 30-35 ge**p**aart und die Nukleotide 40-50 **u**ngepaart sind, so sieht der entsprechende Eintrag wie folgt aus: `30-35:p 40-50:u`.

Diese Information kann u. a. genutzt werden, um falsch vorhergesagte Basenpaarungen aus den Basenpaarungsmatrizen bzw. im Dotplot zu streichen, wodurch idealerweise eine Reduktion des „Basenpaarschmiers“ erreicht werden sollte, der u. U. die Alignment-Korrektur erschwert. Ist beispielsweise bekannt, dass ein Nukleotid ungepaart vorliegt, so werden alle vorhergesagten Basenpaare, an denen dieses Nukleotid beteiligt ist, aus dem Dotplot graphisch entfernt. Diese Vorgehensweise wurde beispielhaft mit Hilfe der Purin-Riboswitch- und SECIS-Form2-Alignments (siehe auch Tabelle 3.2) untersucht (Daten nicht gezeigt). Hier brachte sie nicht den gewünschten Erfolg; der visuelle Eindruck der im Dotplot dargestellten Strukturverteilungen änderte sich nahezu gar nicht, obwohl im Falle des SECIS-Alignment Struktur-Mapping-Informationen für vier aus 21 Sequenzen vorlagen.

Gleichzeitig werden in der „Struktur-Alignment“-Anzeige vorhergesagte Basenpaare markiert, die den eingetragenen Informationen widersprechen (siehe Abbildung 3.9 für ein Beispiel). Ziel wäre es dann, die Zahl dieser Ausreißer durch entsprechende Alignment-Modifikationen zu minimieren.

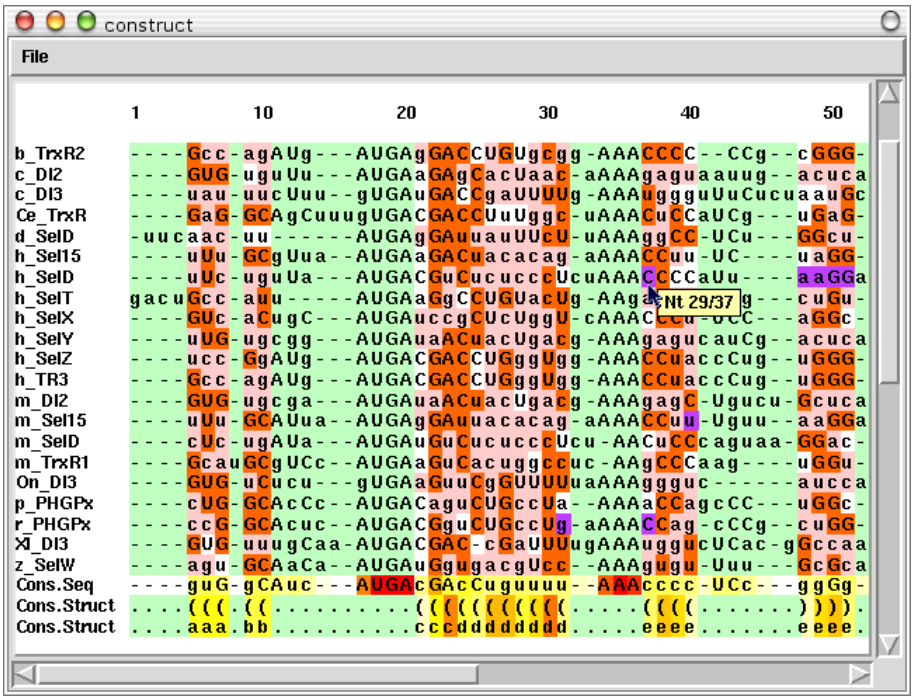


Abbildung 3.9: Berücksichtigung bekannter Basenpaare in der Struktur-Alignment-Ansicht. Für mehrere Sequenzen wurden Fagegaltier *et al.* (2000) Daten aus Struktur-Mapping-Experimenten entnommen und die entsprechenden Basenpaarungsinformationen, wie in Abbildung 3.8 beispielhaft gezeigt, in CONSTRUCT eingebunden. Eine mit CONSTRUCT vorhergesagte Konsensus-Struktur ist in der hier gezeigten Struktur-Alignment-Ansicht dargestellt. Rötliche Bereiche liegen laut Vorhersage basengepaart, grüne ungepaart vor. Den Mapping-Informationen widersprechende Positionen sind violett hinterlegt. Im Beispiel ist das Nukleotid C29 (alignierte Position 37) laut Struktur-Mapping ungepaart und laut Konsensusstruktur-Vorhersage gepaart.

Extraktion von Sekundärstrukturen aus PDB-Dateien

Um die Informationen aufgeklärter 3D-Strukturen nutzen zu können, wurde eine Methode entwickelt, die es ermöglicht aus PDB-Dateien (Berman *et al.*, 2000) Basenpaarungen zu extrahieren (nur solche können wie im vorangegangen Abschnitt beschrieben in CONSTRUCT einbezogen werden). Die Sekundärstruktur ist so gut wie nie in den zur Veröffentlichung von Strukturen genutzten PDB-Dateien eingetragen, jedoch ist die Information hierzu implizit enthalten. Im Folgenden wird diese Vorgehensweise nur kurz beschrieben, da sie im Laufe der Arbeit keine größere Anwendung mehr fand.

Das entscheidende Programm ist MC-ANNOTATE (Gendrona *et al.*, 2001), welches eine quantitative geometrische Analyse von RNA/DNA-3D-Strukturen erlaubt. Es annotiert Nukleotid-Konformationen, -Interaktionen, Pucker-Modus, Stacking etc. und bestimmt für jede Konformationen einen sogenannten „Peculiarity“-Grad. Dieser ist ein statistisches Maß für die Abweichung einer Konformation von der Norm der jeweiligen Klasse, wobei diese Norm durch eine vorherige Analyse aller bekannten Strukturen definiert wurde. Somit lassen sich mit diesem Wert u. a. mögliche stereochemische Fehler in der Struktur aufklären (für eine genauere Beschreibung siehe die MC-ANNOTATE-Webseite[6] geht meist schief wegen sonderzeichen und Gendrona *et al.*, 2001).

Tabelle 3.1: Vergleich der bestimmten 5S rRNA Sekundärstrukturen. Die in der Tabelle aufgeführten Basenpaardistanzen bzw. Ähnlichkeiten der Strukturen für die 5S rRNA wurden mit Hilfe von RNADISTANCE (Hofacker *et al.*, 1994) bestimmt. Als Referenz diente die von Ban *et al.* (2000) beschriebene Sekundärstruktur. Die durch die im Text geschilderte Vorgehensweise inkl. Verwendung der Peculiarity-Werte bestimmte Sekundärstruktur (4) ist der Referenz am ähnlichsten.

	(1)	(2)	(3)	(4)	(5)
RNAFOLD Version 1.4 (1)	-	4	50	56	66
MFOLD Version 3.1 (2)		-	52	58	68
MC-ANNOTATE und Nussinov (3)			-	18	20
Wie 3, gewichtet mit Peculiarity-Werten (4)				-	18
Ban *et al.* (2000) (5)					-

Lässt man eine Struktur mit Hilfe von MC-ANNOTATE analysieren (was durch das Hochladen einer PDB-Datei auf den Webserver geschieht), so werden unter anderem die extrahierten Basenpaarungen inkl. der zugehörigen Peculiarity-Werte ausgegeben. Diese Liste stellt allerdings keine eindeutige Sekundärstruktur dar, sondern enthält auch tertiäre und falsch vorhergesagte Wechselwirkungen. Um hieraus nun eine eindeutige Sekundärstruktur zu gewinnen, wurde gemäß dem Nussinov-Algorithmus zur Basenpaarmaximierung (Nussinov *et al.*, 1978) vorgegangen. Dabei werden optional die (negierten) Peculiarity-Werte in die hierfür benötigte Paarungsmatrix (Dotplot) eingetragen, um über diese Gewichtung möglicherweise falsche Basenpaarungen zu bestrafen.

Als Beispiel soll hier die 5S rRNA von *Haloarcula marismortui* dienen. Ban *et al.* (2000) haben die 50S rRNA (große ribosomalen Untereinheit; PDB-ID 1FFK) von *Haloarcula marismortui* mittels Röntgenkristallographie bei einer Auflösung von $2,4$ Å bestimmt. Um aus der PDB-Datei allein die Sekundärstruktur der 5S rRNA zu bestimmen, wurde zunächst per RASMOL (Sayle & Milner-White, 1995) nur diese selektiert und in einer neuen PDB-Datei gespeichert. Diese wurde wiederum auf dem MC-ANNOTATE-Webserver[7] geht meist schief wegen sonderzeichen analysiert. Die Basenpaarungen inkl. zugehöriger Peculiarity-Werte wurden per Skript aus den resultierenden HTML-Seiten (siehe dort: „Pairing / Non-Adjacent relations“) extrahiert. Anschließend wurde unter Verwendung des Nussinov-Algorithmus eine eindeutige Sekundärstruktur bestimmt, wobei dies einmal mit und einmal ohne Gewichtung durch Verwendung der Peculiarity-Werte geschah. Tabelle 3.1 zeigt einen Vergleich der so bestimmten Struktur(en) mit der von Ban *et al.* manuell bestimmten und der mittels RNAFOLD bzw. MFOLD vorhergesagten Strukturen. Die per Thermodynamik vorhergesagten Strukturen (MFOLD und RNAFOLD) unterscheiden sich untereinander kaum, sehr wohl aber von der Referenz aus Ban *et al.* (2000). Die mit Hilfe des zuvor beschriebenen Ansatzes bestimmten Strukturen sind der Referenz deutlich ähnlicher. Durch Verwendung der Peculiarity-Werte (Bestrafung von laut MC-ANNOTATE ungewöhnlichen Basenpaaren) lässt sich die Ähnlichkeit weiter steigern, womit die prinzipielle Tauglichkeit des Ansatzes gezeigt ist. Die somit bestimmten Strukturen lassen sich wie zuvor beschrieben in CONSTRUCT verwenden.

[6] http://www-lbit.iro.umontreal.ca/mcannotate/

[7] http://www-lbit.iro.umontreal.ca/mcannotate/

Tabelle 3.2: Übersicht der mit CONSTRUCT erstellten/verifizierten Referenz-Alignments. Die Tabelle führt Eigenschaften der mit CONSTRUCT erstellten/verifizierten Referenz-Alignments in Form der Anzahl der Sequenz, durchschnittliche Sequenzlänge, Sequenz-Homologie (in Prozent APSI; siehe Abschnitt 3.5.3) und Struktur-Konservierung (SCI; siehe Abschnitt 3.5.4) auf. Die geklammerten SCI-Werte sind die des Quell-Alignments. Die unter „Mit CONSTRUCT verifiziert" aufgeführten Alignments ließen sich aufgrund der im Text genannten Limitierungen nicht korrigieren.

Mit CONSTRUCT erstellt/korrigiert					
RNA	Quelle	# Seq.	⌀ Länge	APSI	SCI
Archaea 5S rRNA	5S rRNA DB (Szymanski *et al.*, 2002)	50	124,0	63	0,66 (0,58)
Archaea 7S RNA	SRP-DB (Rosenblad *et al.*, 2003)	22	310,7	52	0,83 (0,80)
Mamit-tRNA Alanine	Mamit-tRNA DB (Helm *et al.*, 2000)	31	68,9	80	1,05 (0,92)
Mamit-tRNA Arginin	Mamit-tRNA DB (Helm *et al.*, 2000)	31	67,9	81	0,69 (0,70)
Mamit-tRNA Asparagin	Mamit-tRNA DB (Helm *et al.*, 2000)	31	73,0	80	0,69 (0,68)
Mamit-tRNA Aspartat	Mamit-tRNA DB (Helm *et al.*, 2000)	30	68,0	79	0,89 (0,87)
Purin-Riboswitch	Mandal *et al.* (2003)	31	78,4	60	0,99 (0,93)
SECIS Form2	Fagegaltier *et al.* (2000)	21	66,4	39	0,78 (0,30)
SECIS Methanococcus	Kryukov & Gladyshev (2004)	14	35,9	50	1,03 (0,96)

Mit CONSTRUCT verifiziert					
RNA	Quelle	# Seq.	⌀ Länge	APSI	SCI
Eukaryotische 5S rRNA	5S rRNA DB (Szymanski *et al.*, 2002)	302	119,5	65	0,45
Eukaryotische 7S RNA	SRP-DB (Rosenblad *et al.*, 2003)	73	278,1	48	0,17
HIV-1 5'-Region	Knudsen *et al.* (2004)	20	650,8	87	0,67
tRNA	tRNA DB (Sprinzl & Vassilenko, 2005)	552	76,4	49	1,23

3.4 Referenz-Alignments erstellt mit CONSTRUCT

Da in CONSTRUCT die maximal verfügbare Information in die Erstellung eines Alignments fließen kann, ist es hiermit möglich Referenz-Alignments sehr hoher Qualität zu erzeugen. So wurde zunächst mit dem Aufbau einer Referenz-Alignment-Datenbank mit Hilfe der in Tabelle 3.2 aufgeführten Alignments begonnen. Einige der dort angegebenen Quell-Alignments sind allerdings zu groß, um sie sinnvoll mit CONSTRUCT bearbeiten zu können. So sind beispielsweise die Sequenzen des Alignments der HIV-1 5'-Region aus Knudsen *et al.* (2004) zu lang und die Zahl der Sequenzen im tRNA-Alignment aus tRNA DB (Sprinzl & Vassilenko, 2005) zu hoch. Als Faustregel lässt sich sagen, dass die Erstellung bzw. Korrektur eines Alignments mit CONSTRUCT nur bis zu einer Alignment-Länge von 400 Nukleotiden und einer Zahl von 100 Sequenzen sinnvoll zu bewerkstelligen ist. Die hier aufgeführten Alignments dienten im Laufe der Arbeit lediglich als Sets zum initialen Test von Alignment-Programmen, als Härtetest wie im Falle des hoch-divergenten und damit schwer zu alignierenden SECIS-Sequenzen und zur Evaluation der in Abschnitt 3.5 beschriebenen Bewertungsmaße.

Die ursprüngliche Idee, eine Datenbank aus ausschließlich mit CONSTRUCT als *ultima ratio* erstellten/verifizierten Referenz-Alignments zu erstellen, wurde schließlich verworfen. Grund dafür ist, dass die Erstellung einer sehr hohen Anzahl von Alignments (um eine sinnvolle statistische Analyse zu ermöglichen), die auch noch gezielt in ihren Eigenschaften (Anzahl Sequenzen, Sequenzhomologie) variieren sollten, aus Zeitgründen nicht mit CONSTRUCT erfolgen konnte. Die anstatt dessen gewählten Vorgehensweisen werden in den Abschnitten 3.6 und 3.7 erläutert.

3.5 Bewertungsmaße für (RNA-)Alignments

Seit Veröffentlichung des Standard-Benchmarks für Protein-Alignments (Thompson *et al.*, 1999a) findet die dort verwendete Sum-of-Pairs-Score (SPS; auch BAliScore genannt; siehe Abschnitt 3.5.1) weite Verbreitung. Sie basiert auf einem Vergleich zwischen einem Referenz- und einem Test-Alignment, wobei im Wesentlichen die Anzahl „korrekt“ alignierter Rest-Paare, heißt solcher Paare, die in Referenz- und Test-Alignment identisch aligniert sind, bestimmt wird. Da sie einen Quasi-Standard darstellt, wurde sie auch in dieser Arbeit verwendet.

Die Bewertung von Alignments struktureller bzw. nicht-kodierender RNAs stellt einen Sonderfall dar, da hier auch besonders das Alignment struktureller Elemente, also von gepaarten und ungepaarten Bereichen von Interesse ist. Dabei ist es beispielsweise entscheidender, dass Helices miteinander aligniert sind, und weniger, wie die Reste innerhalb dieser Helices aligniert sind. Aus diesem Grund wurde zur Bewertung der Leistung von echten RNA-Alignment-Programmen, wie DYNALIGN, FOLDALIGN oder PMMULTI, in der Literatur oft der Umweg eingeschlagen, die Leistung über die Struktur-Vorhersage-Qualität und weniger über die resultierenden Alignments an sich zu bestimmen.

Eine Bewertung des strukturellen Anteils eines Alignments ist nicht trivial. Die Verwendung einer Konsensus-Struktur zur Bewertung für solche Alignments verbietet sich, da diese ebenfalls nur durch eine Vorhersage bestimmt werden kann, womit sich dann wieder die Frage nach der Qualität der Vorhersage stellt. Zunächst habe ich versucht, neue Maße zu entwickeln, die Abstraktionen des visuellen Eindrucks des CONSTRUCT-Konsensus-Dotplots darstellen. So bietet es sich beispielsweise an, alle Konsensusbasenpaar-Wahrscheinlichkeiten des Konsensus-Dotplots aufzusummieren, um einen Wert zu erhalten, der die strukturelle Konservierung in einem Alignment beschreibt. Auf ähnliche Weise lassen sich die Werte des gegenseitigen Informationsgehalts verwenden. Aufgrund des hier immantenten statistischen Rauschens musste eine automatische Filterung erfolgen. So wurden nur solche Werte verwendet, die größer als der Mittelwert zuzüglich zweimal der entsprechenden Standardabweichung sind. Weiterhin wurde auch die von RNAALIFOLD berechnete Konsensus-MFE eines Alignments als Maß in Erwägung gezogen. Um die Nähe struktureller Elemente oder Basenpaare im Dotplot ausdrücken zu können, wurde weiterhin ein Maß namens BpCluster entwickelt. Dieses erlaubt durch Anwendung eines Distanzmaßes und einer Sprungfunktion eine Aussage darüber, wie dicht Basenpaare im CONSTRUCT-Konsensus-Dotplot beieinander liegen, wohingegen bei vorgenannten Maßen nur die Aussage möglich ist, „korrekt aligniert“ oder „nicht korrekt aligniert“. In allen Fällen erhält man Werte, die in etwa das Maß an struktureller Konservierung in einem Alignment beschreiben.

Allerdings ist allen diesen Werten gemeinsam, dass sie nicht absolut interpretierbar sind. Das heißt eine Normierung, welche einen Vergleich zwischen Alignments unterschiedlicher Sequenzzusammensetzung erlaubt, ist schwer oder gar nicht möglich (siehe auch die entsprechende Anmerkung in Abschnitt 3.5.4). Diese Eigenschaft ist deshalb so wichtig, da es auch möglich sein sollte, beispielsweise die Güte eines Alignments von 20 tRNA-Sequenzen mit der eines Alignments von 50 5S rRNA-Sequenzen vergleichen zu können. Die genannten Maße hingegen erlauben nur einen Vergleich von Alignments genau einer Sequenz-Zusammensetzung.

Damit ist eine Bestimmung der Alignment-Güte in Abhängigkeit von beispielsweise der Sequenz-Homologie nicht möglich. Aufgrund dessen werden die Eigenschaften dieser Maße im Folgenden nicht weiter diskutiert.

Theoretisch bietet ein in Washietl & Hofacker (2004) beschriebener Ansatz, welcher ursprünglich der Vorhersage nicht-kodierender RNAs diente, eine Alternative. Hier wurde zunächst mit Hilfe von RNAALIFOLD eine Konsensus-MFE aus einem Alignment berechnet. Dann wurde das Alignment auf die dort beschriebene, spezielle Art und Weise mehrfach randomisiert und jeweils erneut die RNAALIFOLD-MFE bestimmt. Um die Signifikanz des „echten“ im Gegensatz zu den zufälligen Energie-Werten zu bestimmen, wurden aus den entsprechenden Werten sogenannte „Z-Scores“ bestimmt. Die Anwendung dieser Methode (implementiert in ALIFOLDZ) verbietet sich hier jedoch schon alleine aufgrund der hohen Laufzeit pro Alignment. Zudem werden aufgrund des Randomisierungsschrittes bei aufeinanderfolgenden, aber identischen Programmläufen unterschiedliche Werte ausgegeben.

Im Folgenden werden die in dieser Arbeit verwendeten Maße (siehe Abschnitte 3.6 und 3.7) im Detail erklärt. Die Wahl fiel auf SPS (sowie SPS$'$) als Sequenz-Maß und den SCI als Struktur-Maß. Zudem wurde zur Bestimmung der Sequenz-Homologie eines Alignments die APSI verwendet. Diese Maße erfüllen die Anforderung, dass die Werte normiert und in allen Fällen gleich interpretierbar sind, d. h. ein Vergleich zwischen Alignments unterschiedlicher Sequenz-Zusammensetzung ist möglich.

3.5.1 Sum-of-Pairs Score (SPS)

Die Sum-of-Pairs-Score (im Folgenden SPS genannt) ist ein sehr weitverbreitetes Maß zur Bewertung von Alignments (siehe u. a. Karplus & Hu, 2001; Lassmann & Sonnhammer, 2002; Pollard *et al.*, 2004). Sie wurde von Thompson *et al.* (1999a) eingeführt und im Programm BALI_SCORE implementiert (unglücklicherweise ist dieses Bewertungsmaß genauso benannt, wie die in den meisten Programmen genutzte Bewertungsfunktion, mit der versucht wird, die Ähnlichkeiten in einer Spalte zu maximieren). Die SPS bestimmt den Anteil der zwischen Referenz- und Test-Alignment identisch alignierten Reste und ist wie folgt definiert:
Gegeben sei ein Alignment mit N Sequenzen und der Länge M (Anzahl Spalten im Alignment). Wenn in Spalte i zwei Reste der Sequenzen j und k genauso aligniert sind, wie im Referenz-Alignment, dann ist p_{ijk} definiert als 1, anderenfalls als 0.

Die „Score“ für Spalte i ergibt sich zu:

$$S_i = \sum_{j=1}^{N} \sum_{k \neq j}^{N} p_{ijk} \ . \tag{3.6}$$

Die SPS wird dann berechnet als:

$$SPS = \sum_{i=1}^{M} S_i \, / \sum_{i=1}^{M_r} S_{ri} \ . \tag{3.7}$$

M_r ist hier die Anzahl Spalten des Referenz-Alignments, also die Länge des Referenz-Alignments, und S_{ri} ist die „Score“ S_i der i-ten Spalte im Referenz-Alignment.

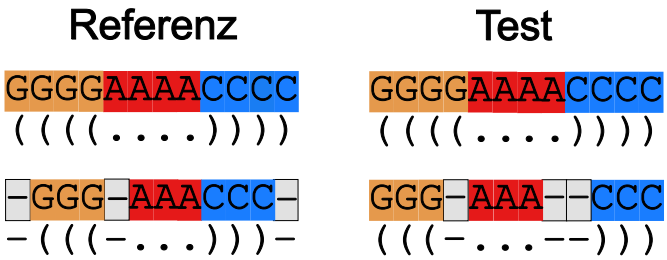


Abbildung 3.10: Beispiel einer Fehlbewertung durch die Sum-of-Pairs-Score. Da bei der SPS alle Reste identisch aligniert sein müssen und der Struktur-Kontext nicht betrachtet wird, kann es im Extremfall dazu kommen, dass die SPS wie im gezeigten Beispiel einen Wert von Null annimmt. Hier sind Referenz- und Test-Alignment strukturell gleich aligniert. Die unter den Sequenzen angegebene Struktur liegt in der sogenannten Punkt-Klammer-Notation vor, in der Punkte nicht-gepaarten Bereichen entsprechen und jedes korrespondierende Klammerpaar einem Basenpaar entspricht. In Referenz- und Test-Alignment sind gepaarte und ungepaarte Bereiche korrekt aligniert. Da aber keiner der Reste identisch aligniert ist, ergibt sich eine SPS von Null.

Die SPS kann Werte zwischen 0 und 1 annehmen, je nachdem ob Referenz- und Test-Alignment keinerlei oder volle Übereinstimmung zeigen, womit sie als Maß der Sensitivität auf Sequenz-Ebene gelten kann. Für einen maximalen SPS-Wert müssen alle Reste zwischen Referenz- und Test-Alignment identisch aligniert sein. Schon ein Versatz von einer Position genügt, um einen Wert von Null zu erreichen, auch wenn durch diesen Versatz das Alignment struktureller Elemente nicht beeinträchtigt werden sollte (siehe Abbildung 3.10 für ein Beispiel).

Diese Art der Bewertung, als auch die Implementation als BALI_SCORE besitzen weitere spezielle Eigenarten. So werden Alignment-Spalten, die im Test-Alignment mehr als 20% Gap-Symbole enthalten, nicht in die Berechnung mit aufgenommen, was unter Umständen dazu führt, dass selbst der Vergleich von einem identischen Referenz- und Test-Alignment zu Werten kleiner 1 führen kann. Weiterhin wird das Alignment eines Restes mit einem Gap genauso gewichtet, wie ein Misalignment des Restes. Deshalb erfuhr die SPS in verschiedenen Benchmarks immer wieder individuelle Änderungen (siehe beispielsweise Karplus & Hu, 2001; Lassmann & Sonnhammer, 2002).

3.5.2 COMPALIGN (SPS′)

Das Programm COMPALIGN ist im Programm-Paket SQUID (Eddy, 2005) enthalten. Es berechnet die Übereinstimmung (auch Identität: „Identity“) auf Sequenz-Ebene zwischen einem Referenz- und Test-Alignment und ähnelt insofern der SPS. Die folgende Beschreibung ist dem Quell-Code des Programms entnommen: Die Identität zweier Alignments mit N Sequenzen ist definiert als durchschnittliche Identität über alle möglichen $N(N-1)/2$ paarweisen Alignments. Diese wiederum ist folgendermaßen definiert:
Gegeben seien zwei in der Referenz alignierte Sequenzen k_1 und k_2, sowie die beiden entsprechenden Sequenzen t_1 und t_2 im Test-Alignment. TC („Total Columns“) sei die Anzahl solcher Spalten zwischen k_1 und k_2, die mindestens ein Nicht-Gap-Symbol enthalten. MC („Matched Columns“) ist die Anzahl Spalten, auf die eine der folgenden Aussagen zutrifft:

- Zwei Nicht-Gap-Symbole in k_1 und k_2 sind in t_1 and t_2 genauso aligniert.
- Der Rest in k_1 ist genauso wie das entsprechende Symbol in t_1 mit einem Gap aligniert.
- Der Rest in k_2 ist genauso wie das entsprechende Symbol in t_2 mit einem Gap aligniert.

Die durchschnittliche Identität zwischen zwei paarweisen Alignments ist dann definiert als MC/TC.

COMPALIGN wird in Abschnitt 3.7.1 statt BALI_SCORE (SPS) zur Berechnung der Sequenz-Übereinstimmung genutzt. Es lässt sich zeigen, dass COMPALIGN und BALI_SCORE ähnliche Werte erzeugen. Die Kurvenverläufe bei Bewertung der Alignments sind ähnlich, mit dem Unterschied, dass die mit COMPALIGN berechneten Werte in den meisten Fällen geringer sind, als die mit BALI_SCORE berechneten (Daten nicht gezeigt; siehe Dalli, 2006). Aufgrund der Ähnlichkeit der beiden Maße wird im Falle von den mit Hilfe von COMPALIGN berechneten Werten im Folgenden auch vom SPS-Äquivalent SPS$'$ die Rede sein.

3.5.3 Average Pairwise Sequence Identity (APSI)

Die „Average Pairwise Sequence Identity“ (durchschnittliche paarweise Sequenzidentität; im Folgenden APSI genannt) ist ein Maß für die Sequenz-Konservierung oder -Homologie innerhalb eines Alignments bzw. innerhalb der Alignment-Spalten und wurde hier mit Hilfe des Programms ALISTAT berechnet, welches im Paket SQUID (Eddy, 2005) enthalten ist. Wie der Name besagt, ergibt sich die APSI für ein Alignment mit N Sequenzen aus dem Durchschnitt aller $N(N-1)/2$ paarweisen Sequenz-Identitäten. Diese sind definiert als Quotient aus der Anzahl der übereinstimmenden Positionen (inkl. Gap-Symbolen) und der kleineren Länge der beiden nicht alignierten Sequenzen. Der Autor weist im Quell-Code des Programms darauf hin, dass es theoretisch eine Vielzahl an möglichen Nennern zur Berechnung der paarweisen Identität gibt. Allerdings führt eine andere als die hier getroffene Wahl entweder dazu, dass Alignments, die artifiziell viele Gaps enthalten, sehr hohe Werte erhalten oder dass lokale Alignments, d. h. solche von Fragmenten an lange Sequenzen, niedrige Werte erhalten.

3.5.4 Structure Conservation Index (SCI)

Der „Structure Conservation Index“ (im Folgenden SCI genannt) dient u. a. dazu, nicht-kodierende RNAs in genomischen Alignments vorherzusagen (Washietl *et al.*, 2005). Er ist ein Maß für die enthaltene oder erhaltene Sekundärstruktur-Information in einem Alignment und beschreibt, wie gut sich aus einem Alignment eine Konsensus-Struktur im Vergleich zu den Einzelstrukturen vorhersagen lässt.

Der SCI basiert auf RNAALIFOLD (Hofacker *et al.*, 2002), mit dessen Hilfe die Konsensus-MFE eines Alignments berechnet wird. Diese Pseudoenergie wird aus einer Kombination von Thermodynamik und Kovarianz berechnet, wobei letztere dazu dient, kompensatorische und konsistente Basenaustausche zu bewerten. Mit diesem Wert an sich lassen sich schon Alignments mit gleicher Sequenzzusammensetzung bewerten und untereinander vergleichen. Ein

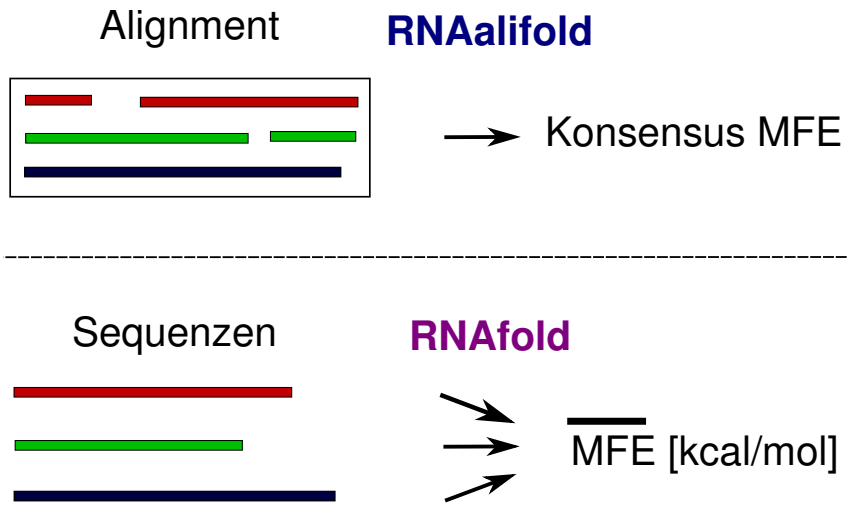


Abbildung 3.11: Illustration zur Berechnung des SCI. Mit Hilfe von RNAALIFOLD wird die Konsensus-MFE des Alignments berechnet. Für jede im Alignment enthaltene Sequenz wird separat mittels RNAFOLD (oder der RNALIB) die MFE bestimmt. Der Quotient aus Konsensus-MFE und den gemittelten Einzel-MFEs ergibt den SCI. Für Details siehe Text.

Vergleich von Alignments unterschiedlicher Sequenzen ist allerdings nicht möglich. Deshalb werden zudem die MFEs jeder im Alignment enthaltenen Sequenz mit Hilfe von RNAFOLD bzw. der RNALIB (Hofacker *et al.*, 1994; Hofacker, 2003) berechnet und gemittelt (siehe Abbildung 3.11). Der Quotient aus Konsensus-Energie E_{A} und gemittelten Einzel-Energien $\bar{E}_{\mathrm{S}}$ ergibt dann den SCI:

$$\mathrm{SCI} = E_{\mathrm{A}} / \bar{E}_{\mathrm{S}} \tag{3.8}$$

Der SCI kann Werte zwischen 0 und etwas größer 1 annehmen. Ein Wert nahe 0 sagt aus, dass anhand des Alignments keine Konsensus-Struktur von RNAALIFOLD vorhergesagt werden konnte. Ein Wert nahe 1 deutet hingegen auf eine konservierte Sekundärstruktur. Sollte der Wert größer als 1 werden, so ist die Konsensusstruktur-Vorhersage zudem durch konsistente, kompensatorische Basenpaaraustausche unterstützt.

Der SCI ist damit ein Maß, das ausschließlich auf Sekundärstrukturinformation beruht und keinerlei Aussage über das korrekte Alignment von nicht gepaarten Regionen macht. Weiterhin benötigt dieses Maß kein Referenz-Alignment. Der SCI ist durch die gemittelten Einzel-MFEs derart genormt, dass auch ein Vergleich von Alignments unterschiedlicher Sequenzzusammensetzung möglich ist.

3.6 Benchmark I (BRAliBase II)

Wie in Abschnitt 3.4 beschrieben, lässt sich eine Datenbank bestehend aus vielen Referenz-Alignments, die in ihren Eigenschaften variieren sollen, nicht ohne weiteres mit Hilfe von CONSTRUCT erstellen. Ein Ausweg stellt die Verwendung bereits publizierter, vertrauenswürdiger Alignments mit sehr vielen Sequenzen dar, aus denen dann kleinere Sub-Alignments gewünschter Eigenschaften generiert werden müssen. Hierfür eignen sich theoretisch Alignments aus Spezialdatenbanken, wie beispielsweise der 5S rRNA-Datenbank (Szymanski *et al.*, 2002), SRP-Datenbank (Rosenblad *et al.*, 2003), tRNA-Datenbank (Sprinzl & Vassilenko, 2005) oder natürlich der Rfam (Griffiths-Jones *et al.*, 2003, 2005). Auch wenn die Qualität einiger der enthaltenen Alignments zumindest anzweifelbar ist, haben sie den hier ausschlaggebenden Vorteil, sehr viele Sequenzen zu enthalten, was eine Kompilation von Sub-Alignments variierender Eigenschaften erst ermöglicht.

In Kooperation mit Paul Gardner[8] und Stefan Washietl[9] entstand die Publikation Gardner *et al.* (2005), in der bereits kompilierte Test-Sets aus den voran genannten Datenbanken zum Benchmark verwendet wurden. Im Anschluss an die Publikation durchgeführte weitergehende Analysen deckten jedoch Fehler in den verwendeten Programmen/Skripten auf, womit die in der Publikation aufgeführten Daten und Diagramme zum Teil nicht korrekt sind. Die Schlussfolgerungen bleiben nur teilweise gültig. Im Folgenden werden Idee, Vorgehensweise und Ergebnisse dargestellt, allerdings werden korrigierte und zum Teil erweiterte Daten verwendet.

3.6.1 Idee und Zielsetzung

Idee dieser Zusammenarbeit mit Paul Gardner und Stefan Washietl war es zu testen, bis zu welchem Sequenz-Homologiegrad der Einsatz von reinen Sequenz-Alignment-Programmen für das RNA-Alignment-Problem noch sinnvoll ist bzw. ab welchem Sequenz-Homologiegrad es nötig ist, weitaus komplexere Struktur-Alignment-Programme zu nutzen, die auch die RNA-Sekundärstruktur berücksichtigen. Weiterhin sollte der Einfluss verschiedener Optionen auf die Programme getestet werden.

Hierzu wurden aus bereits publizierten Alignments, die möglichst viele Sequenzen enthielten, kleinere Referenz-Alignments generiert und zwar so, dass diese einen möglichst weiten Sequenz-Homologie-Bereich (gemessen in Form des APSI; siehe Abschnitt 3.5.3) abdecken. Diese Alignments bzw. die zugehörigen dealignierten Sequenzen wurden dann mit Hilfe der zu testenden Programme wiederum aligniert. Die Güte der berechneten Test-Alignments wurde dann anhand der SPS (siehe Abschnitt 3.5.1) und des SCI (siehe Abschnitt 3.5.4) in Abhängigkeit vom Sequenz-Homologiegrad (APSI bzw. Referenz-APSI) des jeweiligen Referenz-Alignments (als „richtige Lösung“) bewertet.

[8] Department of Evolutionary Biology, University of Copenhagen

[9] Institut für Theoretische Chemie und Molekulare Strukturbiologie, Universität Wien

3.6.2 Referenz-Alignments

Als Referenz-Alignments konnte zum Teil auf bereits für Washietl & Hofacker (2004) erstellte Test-Sets zurückgegriffen werden. Dort wurden zur Erzeugung einer recht hohen Zahl von Referenz-Alignments der Rfam-Datenbank Version 5.0 (hier Griffiths-Jones *et al.*, 2003) folgende Datensätze entnommen:

- 5S rRNA („Seed“-Alignment)
- tRNA („Seed“-Alignment)
- U5 spliceosomale RNA („Full“-Alignment)
- Group II Intron („Full“-Alignment)

Das zunächst ebenfalls verwendete und der SRP-Datenbank (Rosenblad *et al.*, 2003) entnommene eukaryotische SRP-RNA-Set wurde (fälschlicherweise) zu einem relativ frühen Zeitpunkt aufgrund scheinbar mangelnder Alignment-Qualität verworfen. Da dieser Fehleindruck durch die in Abschnitt 3.6.4 beschriebenen Programm-Fehler begründet war, wurde das SRP-RNA-Set in die hier vorgestellten Ergebnisse wieder mit einbezogen.

Aus den genannten Datensätzen wurden auf folgende, in Washietl & Hofacker (2004) genauer beschriebene Art und Weise Referenz-Alignments mit fixer Anzahl Sequenzen $k = 5$ konstruiert: Im ersten Schritt wurden je Set mit Hilfe von BLASTCLUST Cluster mit definierter Sequenz-Identität erstellt. Der Sequenz-Identitätsbereich wurde möglichst breit gewählt, um Referenz-Alignments für jeden Sequenz-Homologie-Grad zu erhalten. Innerhalb dieser Cluster wurden alle möglichen Kombinationen (mit gegebenem k) berechnet, wovon dann zufällig eine in Abhängigkeit von der Größe des Clusters gewählte Anzahl extrahiert wurde. Hiermit sollte eine möglichst gleichmäßige Verteilung der Sequenzen aus den initialen Sets in den entstehenden Alignments gewährleistet werden.

Hierdurch konnten pro RNA-Familie für $k = 5$ ca. 100 Referenz-Alignments erstellt werden, die einen durchschnittlichen SCI von 0,87 aufwiesen (siehe Tabelle 3.3). Die Zahl der Alignments ist wie zu erwarten nicht sehr gleichmäßig über den Sequenzidentitätsbereich verteilt (siehe Abbildung 3.12). So ergibt sich zwischen 50% und 70% APSI ein leichtes Maximum und die Zahl der Alignments fällt unterhalb 50% stark ab. Dieser erste Datensatz (im Folgenden auch Sequenz-Alignment-Datensatz genannt) wurde für den Benchmark von reinen Sequenz-Alignment-Programmen eingesetzt.

Für den Benchmark der Struktur-Alignment-Programme musste ein reduzierter Datensatz erstellt werden, da diese Programme naturgemäß äußerst rechen- und speicherintensiv sind oder ohnehin nur ein paarweises Alignment erlauben. Dieser zweite Datensatz bestand aus 118 paarweisen ($k = 2$) tRNA-Referenz-Alignments mit einem durchschnittlichen SCI von 1,05 (siehe ebenfalls Tabelle 3.3). Dadurch, dass hier nur je zwei Sequenzen miteinander kombiniert werden mussten, ergab sich eine höhere Kombinationsvielfalt, was dazu führte, dass Alignments mit, im Vergleich zum vorgenannten Set, geringerer Sequenz-Identität ($<$20% APSI) erstellt werden konnten. Dieser Datensatz wird im Folgenden auch Struktur-Alignment-Datensatz genannt.

Tabelle 3.3: Anzahl Referenz-Alignments und durchschnittlicher SCI der Datensätze. Für das Set, welches dem Benchmark der Sequenz-Alignment-Programme diente (in der Publikation „data-set-1“ genannt), wurden je RNA-Familie ca. 100 Alignments mit je fünf Sequenzen erstellt. Das entsprechende Set zum Test der Struktur-Alignment-Programme („data-set-2“) bestand aus paarweisen tRNA-Alignments.

Sequenz-Alignment-Datensatz	(„data-set-1“)	
RNA Familie	Anzahl	⌀ SCI
Group II Intron	92	0,71
5S rRNA	89	0,91
SRP RNA	93	0,81
tRNA	98	1,15
U5 RNA	109	0,77
$\sum$	481	0,87

Struktur-Alignment-Datensatz	(„data-set-2“)	
RNA Familie	Anzahl	⌀ SCI
tRNA	118	1,05

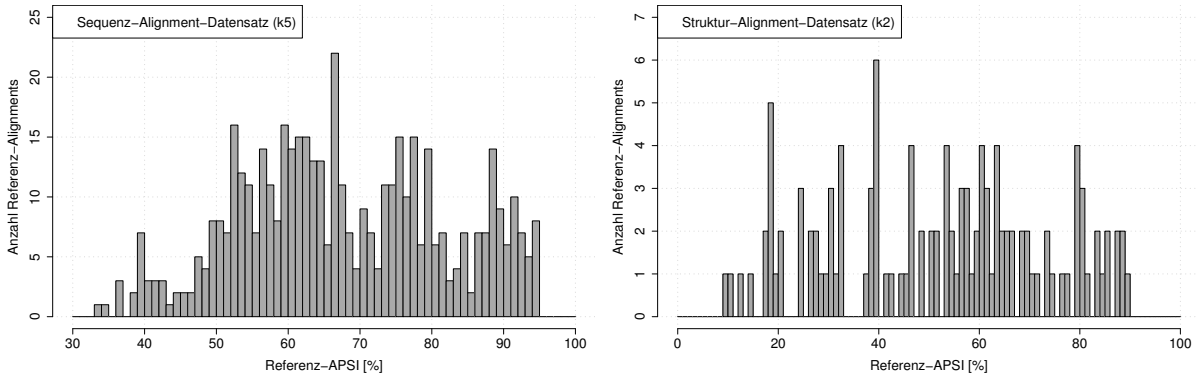


Abbildung 3.12: Histogramm der Alignment-Anzahl über den Sequenz-Homologie-Bereich. Links: Die Anzahl Alignments ist nur leicht ungleichmäßig über die Sequenz-Identität oder -Homologie (gemessen als APSI) verteilt. Unterhalb 50% APSI ist die Erstellung entsprechender Subalignments mit der im Text genannten Methode kaum noch möglich. **Rechts:** Mit den paarweisen tRNA-Alignments lässt sich eine Sequenzidentität von fast 15% erreichen. (Man beachte, dass die Achsen der beiden Plots nicht identisch skaliert sind.)

3.6.3 Eingesetzte Alignment-Programme

Eine detaillierte Auflistung aller eingesetzten Programme, der entsprechenden Versionen inkl. verwendeter Optionen und Kommandozeilenparameter findet sich in Abschnitt 2.2.1. Abbildung 3.13 zeigt eine Einteilung der Programme in Kategorien in Form eines Venn-Diagramms. Im Gegensatz zu der üblichen Vorgehensweise wurden nicht nur die Standard-Einstellungen der Programme verwendet, sondern die Optionen soweit es sinnvoll erschien, variiert. Auf eine gezielte Variation der Gap-Kosten wurde allerdings verzichtet.

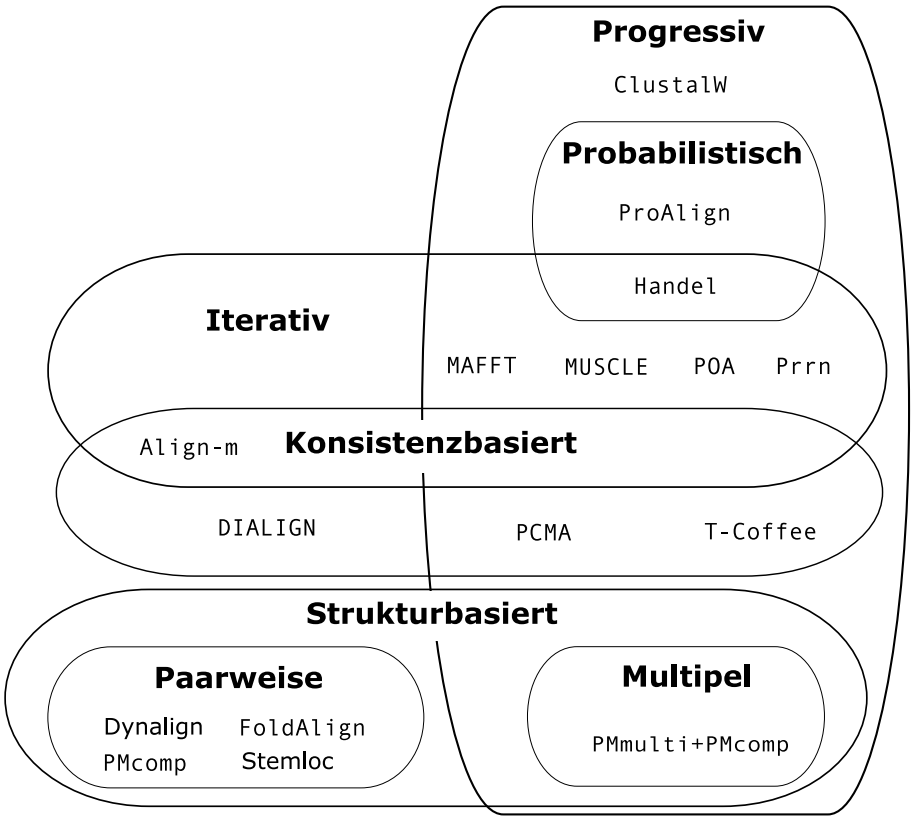


Abbildung 3.13: Venn-Diagramm der verwendeten Alignment-Programme. Ähnlich Thompson *et al.* (1999a) sind hier die für den Benchmark eingesetzten Sequenz- und Struktur-Alignment-Programme in die üblichen Kategorien eingeteilt. Nach Gardner *et al.* (2005).

3.6.4 Eingesetzte Bewertungsmaße

In der Publikation Gardner *et al.* (2005) kamen zur Bewertung der Alignments die SPS (siehe Abschnitt 3.5.1) berechnet mit Hilfe von BALI_SCORE (Thompson *et al.*, 1999a), sowie der SCI (siehe Abschnitt 3.5.4) berechnet mit Hilfe von RNAZ (Washietl *et al.*, 2005) zum Einsatz. Die Sequenz-Identität in Form des APSI (siehe Abschnitt 3.5.3) wurde mit Hilfe von ALISTAT (Eddy, 2005) bestimmt.

Schon während der Arbeit an der Publikation waren mehrere Fehler in den Programmen bekannt. So werden beispielsweise Sequenz-Namen durch SQUID's SREFORMAT, welches zur Formatierung der Alignments genutzt wurde, rechtsbündig geschrieben, was nicht der Norm entspricht. Dies ist allerdings erst in Kombination mit den Routinen zum Lesen von Alignments im Clustal-Format von RNAZ (gleiches gilt für RNAALIFOLD) fatal, da diese die so formatierten Dateien anstandslos lesen, jedoch falsche Werte (ohne einen Programmabbruch) ausgeben. So mussten nicht nur für die Ausführung der Alignment-Programme Skripte entworfen werden (siehe Abschnitt 3.2), sondern auch für die Programme, welche der Bewertung der Alignments dienten. Eines dieser Skripte, welches der Berechnung der Sequenz-Identitäten diente, war jedoch teilweise inkompatibel zu SQUID's ALISTAT. Dies führte dazu, dass die Sequenz-Identitäten (APSI) in der genannten Publikation nicht korrekt sind, womit sich auch

die dort gezeigten, sehr niedrigen APSI-Werte erklären lassen. Der Fehler wirkte sich allerdings nur bei dem Sequenz-Alignment-Datenset aus. Da die Sequenz-Identitäten der Referenz-Alignments betroffen sind, wirkte sich der Fehler auf alle Programme gleichermaßen aus.

Ein weiterer Fehler in RNAZ (bzw. der RNALIB) führt dazu, dass die Energie von Sequenzen, welche in Kleinbuchstaben vorliegen, falsch berechnet werden, da Tetra-Loop-Energien im Programm (bzw. der RNALIB) durch Groß-/Kleinschreibung-sensitiven Vergleich mit einer Liste bekannter Tetra-Loops berechnet werden[10]. Damit würde auch der SCI u.U. inkorrekt berechnet. Im vorliegenden Fall wirkte sich zumindest dieser Fehler nicht aus, da alle Sequenzen in Großbuchstaben vorlagen.

Im Folgenden werden nur Daten gezeigt, welche nach der Publikation neu berechnet wurden. Für die Berechnung des SPS wurde hier weiterhin BAliScore benutzt, für die Berechnung des SCI jedoch das in Abschnitt 3.7.4 vorgestellte Programm SCIF.

3.6.5 Benchmark der Sequenz-Alignment-Programme

Es bietet sich an, das Abschneiden der Programme (gemessen als SCI oder SPS) gegen den Sequenz-Homologiegrad (gemessen als APSI) des Referenz-Alignments aufzutragen. Die so erzeugten Plots zeigen allerdings eine starke Streuung (siehe Abbildung 3.14), da die Qualität der erzeugten Alignments nur bedingt vom Sequenz-Homologiegrad abzuhängen scheint. So gibt es beispielsweise recht Sequenz-homologe Sets, bei denen PROALIGN (wie später gezeigt, als „gutes“ Programm) schlecht-bewertete Alignments erzeugt, bzw. auch sehr divergente Sets, die PROALIGN unerwartet gut aligniert. Beides ist in Abbildung 3.14 zu erkennen. Aufgrund der recht starken Streuung war es nötig, die Plots mit Hilfe der Lowess-Funktion (siehe Abschnitt 2.5) zu glätten. Der in dieser Abbildung gezeigte Kurvenverlauf ist typisch, wenn der SCI als Bewertungsmaß verwendet wird, und zeigt sich bei anderen Programmen, wenn auch verschoben, ebenfalls (siehe Abbildung 3.15 oben).

Dort sind aus Gründen der Übersicht nur die Standard-Optionen der Programme aufgeführt, es sei denn, eine andere Optionswahl stellte sich laut des später gezeigten Rankings (siehe Tabelle 3.4) als eindeutig besser heraus, wie im Falle von POA (g,p), T-COFFEE (c) und PCMA (agi20). Der Kurvenverlauf bei Verwendung des SPS ist ähnlich, wenn auch um einiges gleichmäßiger (siehe Abbildung 3.15 unten). In beiden Fällen lässt sich erkennen, dass die Programme ab einer Sequenz-Homologie größer 75% nahezu gleich gut abschneiden, wenn man von MAFFT, dessen Alignments deutlich schlechter bewertet werden, absieht. Bei der SCI-Bewertung ergibt sich an dieser Stelle sogar ein leichtes, unerwartetes Maximum, welches kein Artefakt der Lowess-Glättung ist, wie in Abbildung 3.14 beispielhaft zu erkennen ist. Fällt der Referenz-APSI unter 75%, so fällt auch die Leistung der Programme deutlich ab. Ab einem APSI von 70% liegen die SCI-Werte der Programme im Vergleich zum Referenz-SCI klar niedriger. Im Falle der SCI-Bewertung ergibt sich ein Plateau zwischen etwa 50% bis 65% APSI. Bei einem Homologiegrad von etwa 50% zeigt sich ein erneuter Abfall der Alignment-Güte, wobei dieser bei Anwendung des SPS nicht so deutlich ausfällt. Im Falle des Referenz-Alignment-SCI zeigt sich nun auch die Auswirkung einer aufgrund der Sequenz-Divergenz erhöhten

[10] pers. Komm. Stefan Washietl

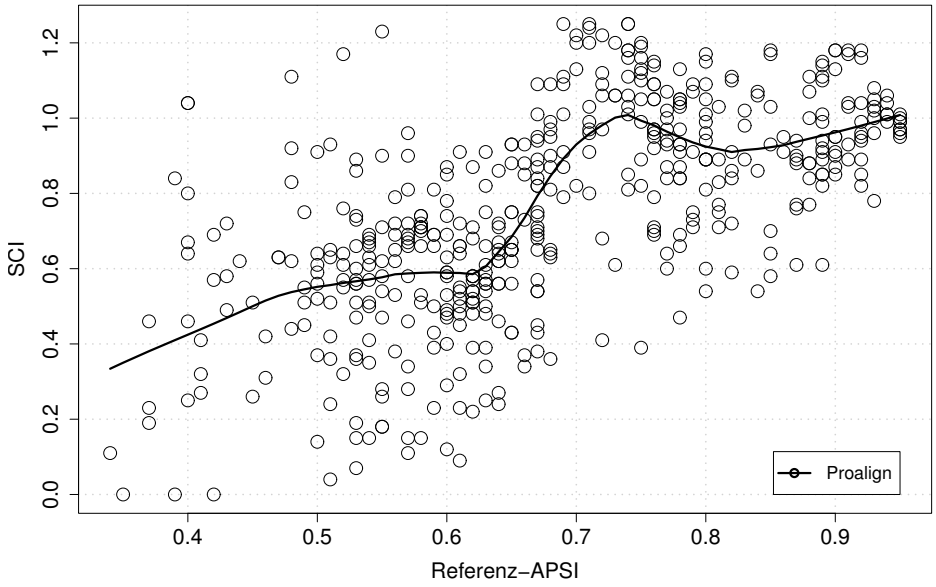


Abbildung 3.14: Streuung der Datenpunkte und Lowess-Glättung. Um die Streuung der Datenpunkte zu demonstrieren, ist hier beispielhaft die Leistung von PROALIGN gezeigt (angewendet auf das Sequenz-Alignment-Datenset; aufgetragen als SCI gegen die Sequenz-Identität des Referenz-Alignments). Jeder Punkt entspricht einem Alignment. Der Kurvenzug entspricht der Lowess-Funktion, berechnet mit einem Glättungsfaktor von 0,3.

Zahl von kompensatorischen Basenpaaraustauschen, welche einen Anstieg statt eines weiteren Abfalls bewirkt.

Wie die Plots zeigen, schneidet MAFFT deutlich schlechter ab, als alle anderen Programme. Gefolgt wird es von DIALIGN und ALIGN-M, deren Alignments ebenfalls durchgehend schlecht bewertet werden. Die Leistung von PCMA fällt mit sinkender Sequenz-Homologie (APSI<0,55) drastisch ab. Alignments erstellt mit HANDEL zeigen für divergente Sequenzen eine überdurchschnittliche SPS, wohingegen der SCI-Kurvenverlauf eher unterdurchschnittlich erscheint. POA, PROALIGN und PRRN hingegen gehören zu den am besten bewerteten Programmen.

Um eine quantitativere Analyse zu ermöglichen, wurde eine Rangfolge erstellt, die auf dem Produkt von SPS und SCI basiert (siehe Tabelle 3.4). Um die Leistungsunterschiede bei verschiedenen Homologie-Graden abbilden zu können, wurden hierfür anhand der in Abbildung 3.15 gezeigten Kurvenverläufe (bzw. Abbildung 2 in Gardner *et al.*, 2005) drei Homologie-Gruppen abgegrenzt:

- Hohe Sequenzhomologie: Referenz-APSI >= 75%
- Mittlere Sequenzhomologie: 55% <= Referenz-APSI <75%
- Geringe Sequenzhomologie: Referenz-APSI < 55%

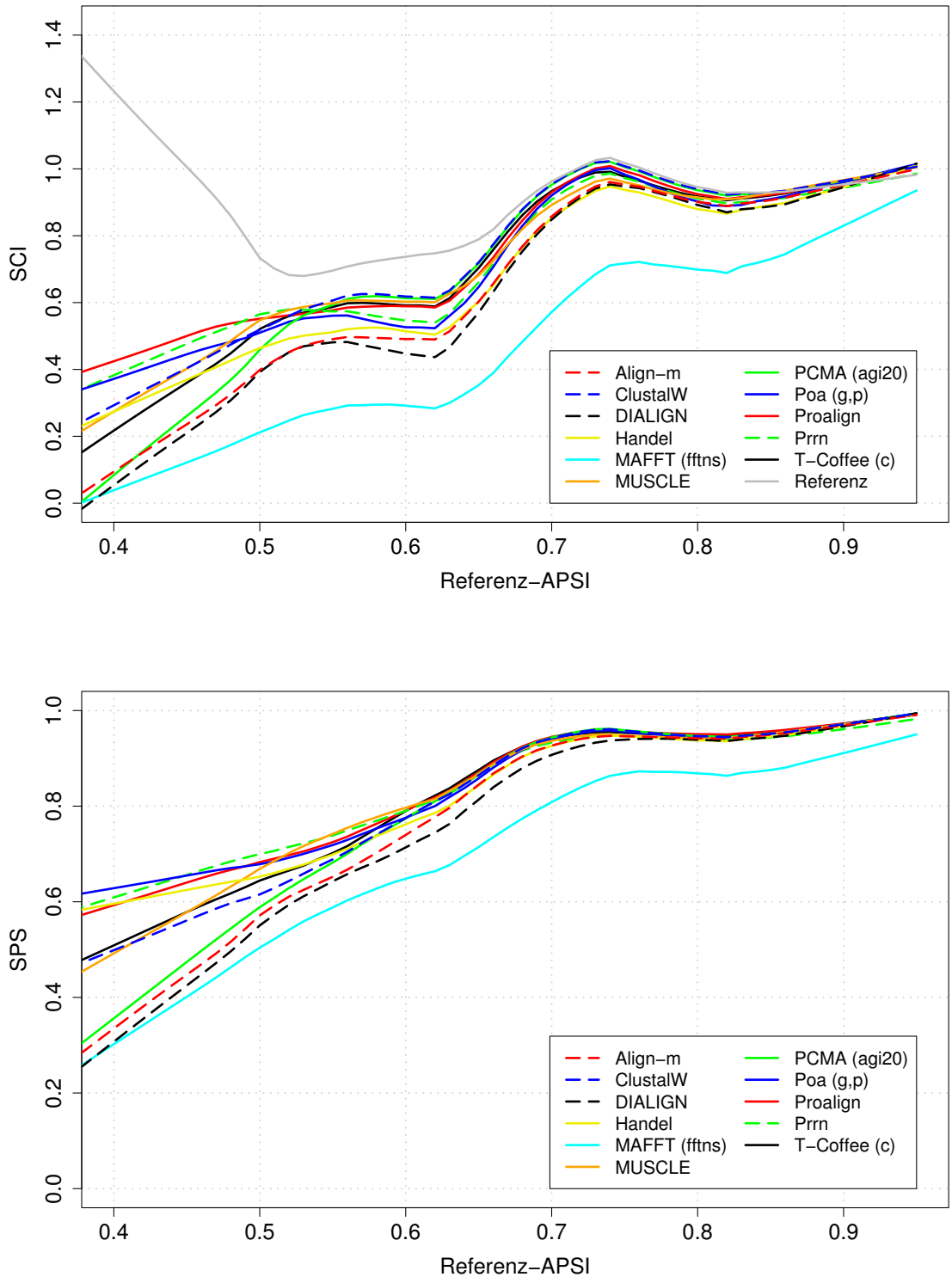


Abbildung 3.15: Leistung der Sequenz-Alignment-Programme in Abhängigkeit von der Sequenz-Homologie der Referenz-Alignments. Die beiden Abbildungen zeigen die Leistung einer Auswahl der Sequenz-Alignment-Programme (siehe Text) bei Anwendung auf das Sequenz-Alignment-Datenset (mit je fünf Sequenzen). Aufgetragen ist als Bewertungsmaß der SCI (**oben**) bzw. SPS (**unten**) gegen die Sequenz-Identität (APSI) des jeweiligen Referenz-Alignments. Beide Plots wurden mit Hilfe der Lowess-Funktion und einem Faktor von 0,3 geglättet. Die negativen Werte im SCI-Plot unter eines APSI-Wertes von 0,4 sind ein Artefakt der Lowess-Glättung. Siehe Abschnitt 2.2.1 für eine Erklärung der in Klammern aufgeführten Optionskürzel.

Tabelle 3.4: Durchschnittlicher SCI und SPS aller mit Hilfe des Sequenz-Alignment-Datensatzes getesteten Programme. Die Ränge wurden anhand des Produkts von SCI und SPS innerhalb jeder der drei Homologie-Gruppen bestimmt. Die ersten zehn Ränge sind jeweils fett markiert. Für eine Erklärung der in Klammern aufgeführten Optionskürzel siehe Abschnitt 2.2.1.

Programm/Option	Hoch-homolog (Ref.-APSI >= 75%)			Medium-homolog (55% <= Ref.-APSI < 75%)			Niedrig-homolog (Ref.-APSI < 55%)		
	SCI	SPS	Rang	SCI	SPS	Rang	SCI	SPS	Rang
Referenz	0,9518	1,0	N/A	0,8150	1,0	N/A	0,8504	1,0	N/A
ALIGN-M (1)	0,9532	0,9239	18	0,8033	0,6053	24	0,5236	0,3351	26
ALIGN-M (2)	0,9532	0,9239	17	0,8033	0,6053	23	0,5236	0,3351	25
ALIGN-M (3)	0,9522	0,9166	22	0,7828	0,5536	31	0,5090	0,3207	29
ALIGN-M (4)	0,9522	0,9166	21	0,7828	0,5536	30	0,5090	0,3207	28
ALIGN-M (5)	0,9388	0,8698	33	0,7770	0,5592	29	0,4828	0,3393	27
CLUSTALW	0,9561	0,9410	**5**	0,8345	0,7112	**1**	0,6179	0,4811	13
CLUSTALW (qt)	0,9592	0,9466	**1**	0,8338	0,7012	**5**	0,5919	0,4743	16
DIALIGN	0,9507	0,9185	20	0,7864	0,5777	27	0,5091	0,3194	30
DIALIGN (it)	0,9452	0,8875	31	0,7734	0,5263	32	0,4867	0,2929	31
DIALIGN (it,o)	0,9402	0,8765	32	0,7567	0,4888	34	0,4804	0,2624	35
DIALIGN (o)	0,9460	0,9141	24	0,7883	0,6035	26	0,5256	0,3782	21
HANDEL	0,9510	0,9114	23	0,8194	0,6253	20	0,6401	0,4342	17
MAFFT (fftnsi)	0,9187	0,8180	37	0,7277	0,4385	37	0,4909	0,2242	37
MAFFT (fftns)	0,8823	0,7471	38	0,7017	0,3956	38	0,4692	0,1928	38
MAFFT (nwnsi)	0,9337	0,8657	35	0,7502	0,4724	35	0,5149	0,2470	34
MAFFT (nwns)	0,9312	0,8631	36	0,7427	0,4538	36	0,4996	0,2242	36
MUSCLE	0,9536	0,9325	15	0,8416	0,6831	**8**	0,6621	0,5144	**5**
MUSCLE (mi32)	0,9535	0,9328	13	0,8417	0,6819	**9**	0,6606	0,5160	**4**
MUSCLE (mi32,mt6)	0,9535	0,9328	12	0,8417	0,6818	**10**	0,6604	0,5157	**6**
MUSCLE (mt6)	0,9536	0,9326	14	0,8416	0,6831	**7**	0,6616	0,5144	**7**
MUSCLE (nj)	0,9533	0,9372	**10**	0,8237	0,6528	16	0,6479	0,5106	**10**
MUSCLE (nj,mi32)	0,9534	0,9372	**8**	0,8239	0,6525	18	0,6470	0,5100	12
MUSCLE (nj,mi32,mt6)	0,9534	0,9372	**7**	0,8239	0,6526	17	0,6471	0,5100	11
MUSCLE (njmt6)	0,9533	0,9372	**9**	0,8237	0,6528	15	0,6479	0,5106	**9**
PCMA	0,9561	0,9410	**4**	0,8306	0,7000	**6**	0,5263	0,3432	24
PCMA (agi20)	0,9561	0,9410	**3**	0,8344	0,7110	**2**	0,5628	0,3997	18
PCMA (agi60)	0,9560	0,9410	**2**	0,8144	0,6674	14	0,5269	0,3502	23
POA	0,9471	0,8859	30	0,7899	0,5621	28	0,4887	0,2853	32
POA (g)	0,9563	0,9202	19	0,8215	0,6420	19	0,6412	0,4434	15
POA (g,p)	0,9566	0,9284	16	0,8397	0,6617	13	0,6734	0,4934	**8**
POA (p)	0,9482	0,8951	29	0,8042	0,5953	25	0,5302	0,3502	22
PROALIGN	0,9583	0,9354	**6**	0,8438	0,6946	**3**	0,6693	0,5311	**3**
PRRN (S10)	0,9461	0,9118	26	0,8492	0,6660	11	0,7038	0,5513	**1**
PRRN	0,9455	0,9129	25	0,8467	0,6637	12	0,6933	0,5328	**2**
T-COFFEE	0,9519	0,9062	28	0,8240	0,6073	22	0,5817	0,3651	20
T-COFFEE (c)	0,9571	0,9319	11	0,8457	0,6923	**4**	0,6271	0,4662	14
T-COFFEE (f)	0,9519	0,9062	27	0,8240	0,6073	21	0,5817	0,3651	19
T-COFFEE (s)	0,9409	0,8647	34	0,7865	0,5069	33	0,5241	0,2502	33

In der Tabelle sind im Gegensatz zu den vorgenannten Plots alle verwendeten Programm-Optionen aufgeführt. In den meisten Fällen hat ein Optionswechsel allerdings nur geringfügige Auswirkung auf die Leistung des entsprechenden Programms. Eine der Ausnahmen stellt T-COFFEE dar, welches deutlich besser abschneidet, wenn die T-COFFEE-Bibliothek mit paarweisen CLUSTALW-Alignments gefüllt wird [T-COFFEE (c) statt (f) oder (s); siehe Abschnitt 2.2.1]. Weiterhin werden mit POA erstellte Alignments besser bewertet, wenn diese global und progressiv erstellt werden.

Im interessanten niedrig-homologen Bereiche zeigt sich, dass PRRN, PROALIGN, MUSCLE und POA am besten abschneiden. Dabei ist auffällig, dass dies für MUSCLE nur dann gilt, wenn es als Clustering-Methode UPGMA (Standard) statt Neighbour-Joining benutzt. PRRN scheint erst im niedrig-homologen Bereich im Vergleich zu den anderen Programmen gute Alignments zu erzeugen. Die Leistung von CLUSTALW ist überdurchschnittlich, lässt im niedrig-homologen Bereich aber vergleichsweise nach. Einzig PROALIGN rangiert in allen drei Homologie-Gruppen konsistent unter den ersten zehn Rängen.

3.6.6 Benchmark der Struktur-Alignment-Programme

Da echte Struktur-Alignment-Programme in den meisten Fällen *per se* nur ein paarweises Alignment erlauben oder diese Vereinfachung des Alignment-Problems aufgrund des extremen Ressourcen-Verbrauchs geboten ist, konnten die eingesetzten Programme nur auf das paarweise Datenset (Struktur-Alignment-Datenset) angewendet werden. Die Qualität der erzeugten Alignments wurde wiederum mit der SPS und dem SCI in Abhängigkeit von der Sequenz-Homologie des Referenz-Alignments in Form des APSI gemessen (siehe Abbildung 3.16).

Um einen direkten Vergleich mit den zuvor getesteten Sequenz-Alignment-Programmen zu gewährleisten, wurden die Werte der als „gut“ identifizierten Programme CLUSTALW und PROALIGN in die Plots mit aufgenommen.

Sowohl bei den mit Hilfe des SCI, als auch mit Hilfe der SPS bewerteten Alignments zeigt sich ein anderer Kurvenverlauf, als bei dem zuvor verwendeten Sequenz-Alignment-Datenset. Hier scheint der Sequenz-Homologie-Grad von ca. 55% APSI in beiden Fällen ein erster Schwellenwert zu sein. Im Falle der SPS ist diese Grenze besonders deutlich ausgeprägt. Bis zu diesem Wert sind auch die Sequenz-Alignment-Programme noch sehr hochbewertet, fallen danach aber deutlich ab, wobei CLUSTALW stärker als PROALIGN betroffen ist, dessen Leistung bei 40% APSI einen weiteren Schwellenwert aufweist.

FOLDALIGN, PMCOMP und DYNALIGN zeigen ab einem Sequenz-Homologie-Grad von ca. 60% APSI einen steigenden SCI und setzten sich damit hier von den anderen Programmen ab. DYNALIGN erzeugt ab ca. 50% APSI sogar Alignments, deren SCI durchschnittlich höher als der der Referenz liegt. Dies geschieht jedoch auf Kosten der Sequenz-Komponente, denn betrachtet man die SPS, so schneidet DYNALIGN von Beginn an sehr schlecht ab. FOLDALIGN hingegen zeigt auch hier eine gleichmäßig hoch bleibende Leistung, wohingegen PMCOMP schon leicht abfällt. Wie zu erwarten ist die schnellere Variante von PMCOMP (hier als `fast` gekennzeichnet), welche die Basenpaarungsmatrizen als kondensierte Vektoren ähnlich dem Sequenz-Alignment aligniert, schlechter als die Standard-Variante. Die Leis-

tung der „schnellen“ Variante ist bei Anwendung der SPS jedoch immer noch vergleichbar mit DYNALIGN. Überraschend ist die schlechte Leistung von STEMLOC. Dessen Kurvenverläufe ähneln stark denen der Sequenz-Alignment-Programme. Die Optionen `slow` und `fast` unterscheiden sich *de facto* nicht.

Es lässt sich festhalten, dass 60% APSI zumindest für das paarweise Alignment ein klarer Schwellenwert zu sein scheint. Ab diesem Wert erzeugen auch die besseren Sequenz-Alignment-Programme deutlich schlechtere Alignments, wohingegen der SCI der mit Struktur-Alignment-Programmen erzeugten Alignments steigt. FOLDALIGN scheint als einziges dieser Programme die richtige Balance zwischen Sequenz- und Struktur-Bewertung zu finden.

3.6.7 Anwendungen

Da mit der vorgenannten Veröffentlichung der erste systematische Benchmark von Alignment-Programmen angewendet auf RNAs publiziert war, stand somit auch erstmals die Möglichkeit zur gezielten Optimierung von Programmen und deren Parametern für das RNA-Alignment zur Verfügung. Gleichzeitig konnte die Leistung neu entwickelter Programme mit den hier getesteten verglichen werden, wie beispielsweise für TLARA in Bauer *et al.* (2005) geschehen. Weiterhin lassen sich systematische Fehler verfolgen.

So war es beispielsweise unter Verwendung dieses Benchmarks möglich, mehrere Fehler und Eigenarten in frühen STRAL-Versionen (Dalli, 2006) aufzudecken. Weiterhin wurden hiermit die Programm-Parameter optimiert (siehe ebenfalls Dalli, 2006).

Die außerordentlich schlechte Leistung von MAFFT bewegte den Autor des Programms dazu, die publizierten Daten als Ausgangspunkt zur Parameter-Optimierung zu nutzen. Es stellte sich dabei heraus, dass die Standard-Einstellungen der beiden Gapkosten-Parameter `op` und `ep` für MAFFT erheblich zu niedrig lagen (siehe auch Anmerkung auf der MAFFT-Homepage[11] geht meist schief wegen sonderzeichen). Die neuen, optimierten Parameter haben als Standard-Einstellung neben weiteren Verbesserungen Einzug in die Version 5 von MAFFT (Katoh *et al.*, 2005) erhalten. Dass diese einen erheblichen Einfluss auf die Leistung von MAFFT haben, wird in Abschnitt 3.7.8 gezeigt. Der Autor konnte zudem zeigen, dass sich die Gap-Parameter von CLUSTALW und PRRN ebenfalls verbessern lassen[12]. Diese Idee wird in Abschnitt 3.7.9 aufgegriffen.

[11] http://www.biophys.kyoto-u.ac.jp/ katoh/programs/align/mafft/

[12] pers. Komm.

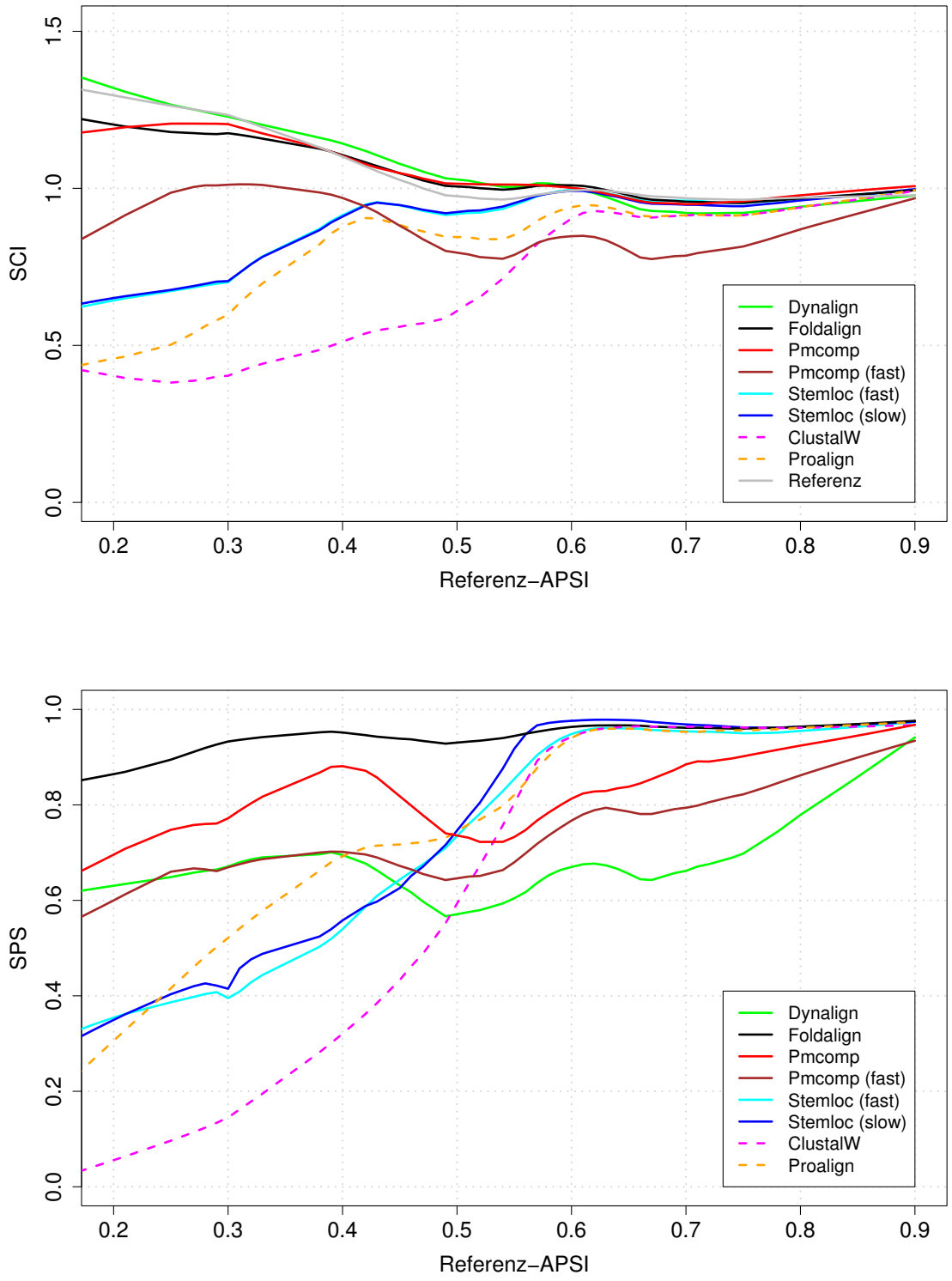


Abbildung 3.16: Leistung der Struktur-Alignment-Programme in Abhängigkeit von der Sequenz-Homologie der Referenz-Alignments. Die beiden Abbildungen zeigen die Leistung der Struktur-Alignment-Programme bei Anwendung auf das Struktur-Alignment-Datenset (mit je zwei Sequenzen). Zum Vergleich wurden die Leistungen von CLUSTALW und PROALIGN, die zuvor als gute Sequenz-Alignment-Programme identifiziert wurden (siehe Tabelle 3.4) hinzugefügt. Aufgetragen ist als Bewertungsmaß der SCI (**oben**) bzw. SPS (**unten**) gegen die Sequenz-Identität (APSI) des jeweiligen Referenz-Alignments. Beide Plots wurden mit Hilfe der Lowess-Funktion und einem Faktor von 0,3 geglättet. Siehe Abschnitt 2.2.1 für eine Erklärung der in Klammern aufgeführten Optionskürzel.

3.7 Benchmark II (BRAliBase IV)

3.7.1 Idee und Zielsetzung

Der in Abschnitt 3.6 und in Gardner *et al.* (2005) beschriebene Benchmark ist der erste systematische Benchmark von Alignment-Programmen angewendet auf RNA-Sequenzen. Einige Fragen konnten dort jedoch nicht untersucht werden. So ließ sich der Einfluss der Sequenzzahl auf die Leistung der Programme nicht messen, da nur Referenz-Alignments mit jeweils fünf Sequenzen erzeugt wurden (bzw. zwei für das strukturelle tRNA-Set). Weiterhin wurde dort nur eine beschränkte Auswahl an RNA-Familien verwendet. Damit besteht theoretisch die Möglichkeit, dass sich die Leistung der Programme bei Anwendung auf Sequenzen anderer RNA-Familien (mit entsprechend anderer Basenzusammensetzung, anderer durchschnittlicher Sequenz-Länge etc.) ändern könnte. Zudem waren zwei der zur Erstellung der Referenz-Alignments erstellten Sets (U5 spliceosomale RNA und Group II Intron) „Full"-Alignments der Rfam, die nicht manuell gewartet, sondern durch das Programm INFERNAL erstellt wurden und damit sogar potentiell nicht zugehörige Sequenzen enthalten könnten.

Aus diesem Grund wurde eine Fortsetzung der Arbeit angestrebt mit dem Ziel, garantiert hoch qualitative Referenz-Alignments vieler RNA-Familien mit unterschiedlicher Sequenz-Zahl zu erstellen. Diese wurde eingesetzt, um die vorigen Ergebnisse zu validieren, den Einfluss der Sequenzzahl zu bestimmen (siehe Abschnitt 3.7.6) und statistische Rangtests durchzuführen. Weiterhin wurde eine Parameter-Optimierung einiger Programme durchgeführt (siehe Abschnitt 3.7.9) und der Einfluss von verschiedenen Substitutionsmatrizen (siehe Abschnitt 3.7.7) untersucht.

Die grundlegende Bewertung der Alignments erfolgte wie in Abschnitt 3.6.1 beschrieben. Anhand der erstellten Referenz-Alignments wurde die (Referenz-)Sequenzhomologie (als APSI) bestimmt. Die Sequenzen der Referenz-Alignments wurden dann durch die zu testenden Programme aligniert und das resultierende Alignment mit Hilfe der in Abschnitt 3.7.4 genannten Bewertungsmaße bewertet.

3.7.2 Referenz-Alignments

Zur Erstellung der Referenz-Alignments wurden ausschließlich „Seed"-Alignments der Rfam-Datenbank (hier Version 7.0; Griffiths-Jones *et al.*, 2005) verwendet, da diese der Literatur entnommen und manuell gepflegt werden, womit sie im Vergleich zu den mit INFERNAL erstellten „Full"-Alignments eine recht hohe Qualität aufweisen sollten.

Da zur Konstruktion der Referenz-Alignment-Sets ein ausreichend hoher Pool an Sequenzen in den „Seed"-Alignments enthalten sein muss, wurden von vorneherein nur solche Alignments berücksichtigt, die mehr als 50 Sequenzen enthielten.

Mit wachsender Sequenz-Länge leidet die Qualität thermodynamischer Strukturvorhersagen (siehe beispielweise Doshi *et al.*, 2004; Mathews *et al.*, 1999), da beispielsweise kinetische Effekte bei der RNA-Faltung eine größer werdende Rolle spielen. Da erneut der SCI zur Bewertung der Alignments herangezogen werden sollte, musste gewährleistet sein, dass die Qualität

Tabelle 3.5: Auflistung der verwendeten „Seed“-Alignments aus der Rfam Version 7.0. Aufgeführt sind das Kürzel der RNA-Familie, die Accession-Nummer, die Anzahl der enthaltenen Sequenzen, die durchschnittliche Sequenzlänge, der APSI, sowie eine Kurzbeschreibung der RNA-Familie.

Name	Accession	# Seq.	⌀ Länge	APSI [%]	Beschreibung
5S_rRNA	RF00001	602	116	59	5 S ribosomale RNA
5_8S_rRNA	RF00002	63	142	76	5.8 S ribosomale RNA
Cobalamin	RF00174	171	204	46	Cobalamin Riboswitch
Entero_5_CRE	RF00386	65	88	84	Enterovirus 5’ Kleeblatt CRE
Entero_CRE	RF00048	56	61	81	Enterovirus CRE
Entero_OriR	RF00041	60	118	87	Enteroviral 3’ UTR Element
gcvT	RF00504	117	101	51	GcvT Element
Hammerhead_1	RF00163	75	81	61	Hammerhead Ribozym (type I)
Hammerhead_3	RF00008	84	55	72	Hammerhead Ribozym (type III)
HCV_SLIV	RF00469	110	36	88	HCV Stem-loop IV
HCV_SLVII	RF00468	63	64	78	HCV Stem-loop VII
HepC_CRE	RF00260	52	51	84	HCV CRE
Histone3	RF00032	64	26	78	Histone 3’ UTR Stem-loop
HIV_FE	RF00480	853	51	91	HIV Ribosomales Frameshift Signal
HIV_GSL3	RF00376	1404	79	88	HIV gag Stem-loop 3
HIV_PBS	RF00375	388	91	84	HIV Primer Bindestelle
Intron_gpII	RF00029	116	76	55	Group II katalytisches Intron
IRES_HCV	RF00061	823	201	91	HCV IRES
IRES_Picorna	RF00229	208	213	82	Picornavirus IRES
K_chan_RES	RF00485	85	113	64	Kalium-Kanal RNA Editing Signal
Lysin	RF00168	60	182	49	Lysin Riboswitch
Retroviral_psi	RF00175	173	104	87	Retrovirales Psi Packaging Element
SECIS	RF00031	64	64	44	SECIS
sno_14q_I_II	RF00181	59	75	66	C/D box snoRNA 14q(I)/14q(II)
SRP_bact	RF00169	70	98	52	Bakterielle SRP RNA
SRP_euk_arch	RF00017	71	297	49	Eukaryotische SRP RNA
S_box	RF00162	71	110	67	SAM Riboswitch (S-box Leader)
T-box	RF00230	67	244	44	T-box Leader
TAR	RF00250	426	56	91	Trans-Activation Response Element (TAR)
THI	RF00059	237	110	52	TPP Riboswitch (THI Element)
tRNA	RF00005	1114	71	43	tRNA
U1	RF00003	54	155	59	U1 spliceosomale RNA
U2	RF00004	77	173	60	U2 spliceosomale RNA
U6	RF00026	53	106	79	U6 spliceosomale RNA
UnaL2	RF00436	144	54	76	UnaL2 line 3’-Element
yybP-ykoY	RF00080	74	128	45	yybP-ykoY Element

dieses auf Thermodynamik beruhenden Bewertungsmaßes nicht durch Verwendung zu großer Sequenzlängen negativ beeinflusst wird. Deshalb wurden nur solche Alignments verwendet, deren durchschnittliche Sequenz-Länge nicht über 300 Nukleotiden lag. Somit kamen ausschließlich die in Tabelle 3.5 aufgeführten „Seed“-Alignments der Rfam zum Einsatz.

Diese Alignments besitzen (im Gegensatz zu den „Full“-Alignments) nur in den wenigsten Fällen eine ausreichend hohe Anzahl von Sequenzen, um wie in Abschnitt 3.6 beschrieben Referenz-Alignments mit vorgegebener Anzahl von Sequenzen mit Hilfe von BLASTCLUST erstellen zu können. Stattdessen wurde ein „naiver“ kombinatorischer Ansatz verfolgt, der im Folgenden beschrieben wird (siehe auch Abbildung 3.17 und Abbildung 3.18).

```
   Input  : Alignment: SeedAln
   Output : Gewünschte Subalignments
   Global : Gewünschte Sequenzzahl k*
            Lösungskandidat Kandidat* (Menge aus Sequenzen)
            SCI-Schwellenwert SciThresh*
            APSI-Bereich ApsiRange*
 1 SCI-Schwellenwert SciThresh* ← 0.6
 2 foreach k* ∈ {2, 3, 5, 7, 10, 15} do
 3     for MaxApsi ← 95 to 20 step −10 do
 4         MinApsi ← MaxApsi − 10
 5         APSI-Bereich ApsiRange* ← (MinApsi, MaxApsi)
 6         Lösungskandidat Kandidat* ← ∅
 7         SeqPool ← Alle SequenzPaare SP ∈ SeedAln mit MinApsi ≤ Apsi(SP) ± 10% < MaxApsi
 8         Entferne doppelte Einträge aus SeqPool
 9         SeqIdListe ← GreedyRecRandComb(SeqPool )
10         if SeqIdListe ≠ ∅ then
11             Sichere Alignment aus SeqIdListe
12             SeqPool ← SeqPool\SeqIdListe
```

Abbildung 3.17: Algorithmus zur Kompilation der Referenz-Alignments. Mit Hilfe dieses im Pseudo-Code angegebenen Algorithmus wurden die Referenz-Alignments durch Neukombination der Rfam-„Seed"-Alignments erstellt. Alignments für einen bestimmten Sequenz-Homologie-Bereich (ApsiRange in Zeile 5) wurden dabei aus solchen Sequenz-Paaren bzw. Sequenzen neu kombiniert, die untereinander selber in etwa die gewünschte Sequenz-Homologie besaßen. Allerdings wurde für die Paare eine Abweichung von ±10% APSI zugelassen (Zeile 7), um eine höhere Kombinationsvielfalt zu erreichen. Globale Variablen sind mit einem hochgestellten Stern versehen. Die Prozedur `GreedyRecRandComb` (Zeile 9) ist in Abbildung 3.18 erläutert.

Ziel war es Sub-Alignments aus den in Tabelle 3.5 aufgeführten Alignments für jede Sequenzzahl $k \in \{2, 3, 5, 7, 10, 15\}$ zu erzeugen, die in ihrer Gesamtheit einen möglichst weiten Sequenzhomologie-Bereich abdecken. Beginnend bei 95% wurden in Intervallen von je 10% alle Sequenzpaare bestimmt, deren APSI mit ±10% Abweichung innerhalb dieser Intervallgrenze lag (siehe Abbildung 3.17). Aus der entstehenden Liste wurden alle doppelten Einträge entfernt. Der resultierende Sequenzpool[13] wurde in einem rekursiven Ansatz (siehe Abbildung 3.18) verwendet, um die gewünschte Anzahl Sequenzen so zu kombinieren, dass ihr APSI wiederum im gewünschten Intervall lag. Um eine gewisse strukturelle Konservierung sicherzustellen, mussten die entstehenden Sub-Alignments einen SCI $\geq$ 0,6 aufweisen, ansonsten wurden sie verworfen. Um zu vermeiden, dass bestimmte Sequenzen gehäuft vertreten sind, wurden Sequenzen, die einmal in einem solchen APSI-Intervall ($\pm 10\%$) verwendet wurden, aus der für eine weitere Lösung zur Verfügung stehenden Menge an Sequenzen entfernt. Anhand der generierten Sequenz-ID-Listen wurden die entsprechenden Sequenzen inklusive Gaps zu einem neuen Alignment zusammengefasst und alle Spalten, die nur aus Gaps bestanden, entfernt.

[13] Die Pool-Größe musste auf 900 Sequenzen beschränkt werden, da die Rekursion aufgrund des hohen Speicherbedarfs bei mehr als 900 Elementen abbrach (Tcl-Fehlermeldung: „too many nested evaluations (infinite loop?)").

```
Input  : Pool zur Verfügung stehender Sequenzen SeqPool
Output : Gültige Lösungsmenge aus Sequenzen oder leere Menge
Global : Gewünschte Sequenzzahl k*
         Lösungskandidat Kandidat* (Menge aus Sequenzen)
         SCI-Schwellenwert SciThresh*
         APSI-Bereich ApsiRange*
1  while |SeqPool| ≥ k* do
2      if |Kandidat*| == k* then
3          if Apsi(Kandidat*) in ApsiRange* and Sci(Kandidat*) ≥ SciThresh* then
4              return Kandidat*
5          else
6              return ∅
7      else
8          Ziehe und entferne zufällig Sequenz S aus SeqPool
9          Kandidat* ← Kandidat* ∪ S
10         SeqIdListe ← GreedyRecRandComb(SeqPool)
11         if SeqIdListe ≠ ∅ then
12             return SeqIdListe
13         else
14             Kandidat* ← Kandidat*\S
15 return ∅
```

Abbildung 3.18: Rekursiver Teil des Algorithmus zur Kompilation der Referenz-Alignments (`GreedyRecRandComb`). Die Prozedur wird initial vom Algorithmus zur Kompilation der Referenz-Alignments (siehe Abbildung 3.17) aufgerufen. Sollte die als Lösungskandidat zur Verfügung stehende Menge an Sequenzen, die nötige Anzahl an Sequenzen enthalten (Zeile 2), so wird getestet, ob sie eine korrekte Lösung darstellt (Bedingung in Zeile 3) und entsprechend die Lösung oder die leere Menge zurückgegeben. Anderenfalls wird der genannten Menge eine zufällige Sequenz aus dem zur Verfügung stehenden Pool an Sequenzen hinzugefügt. Um zu testen, ob hierdurch eine gültige Lösung erstellt wurde, ruft sich die Prozedur anschließend selbst auf (Zeile 10). Falls ja, wird die entsprechende Liste aus Sequenzen zurückgegeben, anderenfalls wird die eben hinzugefügte Sequenz verworfen.

Mit dieser Vorgehensweise konnten 18990 Alignments generiert werden. Idealerweise würden aus jeder RNA-Familie in etwa gleich viele Alignments erzeugt. Dies ist aufgrund der ungleichmäßigen Verteilung an Sequenzen in den „Seed“-Alignments der Rfam nicht möglich, wie in Tabelle 3.6 ersichtlich wird (vgl. auch Tabelle 3.5 auf Seite 64).

Die strukturelle Konservierung in Form des SCI ist ebenfalls in Tabelle 3.6 dargestellt. Der durchschnittliche SCI von 0,93 ist im Vergleich zu der während der Kompilation der Sets verwendeten unteren Grenze von 0,6 recht hoch. In den meisten Fällen nimmt der SCI mit der Anzahl der Sequenzen in den Alignments ab, obwohl hier theoretisch kompensatorische Basenpaaraustausche zu höheren Werten führen könnten. Einzige Ausnahme stellen die Histon3- und tRNA-Alignments dar.

Es war nicht möglich, eine gleichmäßige Anzahl Alignments über den gesamten Sequenz-Identitätsbereich zu erzeugen, da die meisten „Seed“-Alignments aus zu wenigen und/oder zu homologen Sequenzen bestehen. So lässt sich beispielsweise aus dem mit 53 Sequenzen recht kleinen, und mit 79% APSI recht Sequenz-homologen U6-Alignment kein einziges Referenz-Alignment

Tabelle 3.6: Anzahl Referenz-Alignments und durchschnittlicher SCI pro RNA-Familie. Aufgeführt sind die Anzahl an generierten Referenz-Alignments sowie der durchschnittliche SCI je RNA-Familie und Anzahl enthaltener Sequenzen ($k2$–$k15$) in dem Alignment. Fälle, in denen keine Referenz-Alignments generiert werden konnten, sind mit N/A gekennzeichnet. Wie zu erwarten, korreliert die Anzahl der Alignments pro Familie sowohl mit der gewünschten Sequenzzahl k, als auch mit der Anzahl Sequenzen im verwendeten Ausgangsalignment (vgl. Tabelle 3.5 auf Seite 64).

RNA-Familie	$k2$		$k3$		$k5$		$k7$		$k10$		$k15$		$\sum$	
	#Seq.	∅ SCI	#Seq.	∅ SCI	#Seq.	∅ SCI	#Seq.	∅ SCI	#Seq.	∅ SCI	#Seq.	∅ SCI	#Seq.	∅ SCI
5S_rRNA	1162	0,95	568	0,89	288	0,87	150	0,83	90	0,79	50	0,74	2308	0,91
5_8S_rRNA	76	0,83	45	0,75	17	0,70	5	0,70	3	0,67	NA	NA	146	0,78
Cobalamin	188	0,77	61	0,71	15	0,69	4	0,66	NA	NA	NA	NA	268	0,75
Entero_5_CRE	48	1,03	32	1,03	19	1,04	10	1,03	8	1,03	5	1,03	122	1,03
Entero_CRE	65	0,84	38	0,80	20	0,80	13	0,75	8	0,69	4	0,70	148	0,81
Entero_OriR	49	0,94	31	0,92	17	0,88	11	0,84	8	0,84	4	0,84	120	0,91
gcvT	167	0,79	67	0,72	22	0,69	12	0,68	3	0,68	1	0,66	272	0,76
Hammerhead_1	53	0,79	32	0,76	9	0,70	1	0,71	NA	NA	NA	NA	95	0,77
Hammerhead_3	126	1,01	99	1,00	52	1,02	32	1,06	17	1,05	12	1,01	338	1,02
HCV_SLIV	98	0,98	63	0,97	36	0,97	26	0,97	16	0,96	10	0,96	249	0,97
HCV_SLVII	51	0,89	33	0,85	19	0,84	13	0,81	10	0,77	7	0,74	133	0,85
HepC_CRE	45	1,01	29	0,99	18	0,98	11	0,97	7	0,95	3	0,93	113	0,99
Histone3	84	1,03	59	1,04	27	1,04	11	1,05	7	1,05	6	1,05	194	1,03
HIV_FE	733	0,97	408	0,96	227	0,95	147	0,95	98	0,94	56	0,93	1669	0,96
HIV_GSL3	786	0,85	464	0,79	246	0,75	151	0,73	95	0,72	61	0,71	1803	0,80
HIV_PBS	188	0,92	124	0,88	76	0,87	55	0,86	38	0,85	25	0,83	506	0,88
Intron_gpII	181	0,79	82	0,72	35	0,66	22	0,65	11	0,64	4	0,63	335	0,74
IRES_HCV	764	0,83	403	0,78	205	0,74	146	0,71	83	0,70	47	0,68	1648	0,78
IRES_Picorna	181	0,96	117	0,93	75	0,91	53	0,90	35	0,87	25	0,83	486	0,92
K_chan_RES	124	0,75	40	0,70	2	0,69	NA	NA	NA	NA	NA	NA	166	0,74
Lysin	80	0,95	48	0,88	30	0,85	17	0,80	7	0,76	3	0,75	185	0,89
Retroviral_psi	89	0,89	57	0,85	34	0,81	24	0,80	17	0,76	11	0,74	232	0,84
SECIS	114	0,88	67	0,86	33	0,83	16	0,80	11	0,77	6	0,77	247	0,86
sno_14q_I_II	44	0,77	14	0,72	1	0,70	NA	NA	NA	NA	NA	NA	59	0,75
SRP_bact	114	0,97	76	0,96	39	0,98	19	0,94	12	0,94	7	0,91	267	0,96
SRP_euk_arch	122	0,92	94	0,85	42	0,82	21	0,74	12	0,72	6	0,69	297	0,86
S_box	91	0,86	51	0,80	25	0,73	12	0,71	7	0,68	2	0,67	188	0,81
T-box	18	0,74	8	0,74	NA	NA	NA	NA	NA	NA	NA	NA	26	0,74
TAR	286	0,98	165	0,98	92	0,99	62	0,99	42	0,99	28	0,95	675	0,98
THI	321	0,85	144	0,79	69	0,75	32	0,75	17	0,69	5	0,71	588	0,81
tRNA	2039	1,12	1012	1,17	461	1,22	267	1,22	143	1,24	100	1,24	4022	1,16
U1	82	0,86	65	0,80	26	0,75	16	0,71	6	0,68	NA	NA	195	0,81
U2	112	0,88	83	0,83	38	0,75	22	0,72	14	0,69	7	0,69	276	0,82
U6	30	0,78	21	0,73	14	0,67	7	0,66	1	0,62	NA	NA	73	0,73
UnaL2	138	0,80	71	0,76	43	0,73	20	0,71	7	0,68	NA	NA	279	0,77
yybP-ykoY	127	0,89	64	0,81	33	0,78	18	0,72	12	0,70	8	0,68	262	0,83
$\sum$	8976	0,95	4835	0,92	2405	0,91	1426	0,90	845	0,89	503	0,89	18990	0,93

mit 15 Sequenzen ($k15$) erstellen. Die Verteilung der Alignments ist in Abbildung 3.19 dargestellt. Es zeigt sich, dass mit der vorgenannten Vorgehensweise eine für alle Sequenz-Zahlen typisch ungleichmäßige Verteilung der Alignments über den Sequenzhomologie-Bereich erzeugt wird. Über 80% APSI lässt sich eine sehr hohe Zahl Alignments generieren. Zwischen 60% und 80% APSI ergibt sich ein Minimum und unterhalb von 60% steigt die Anzahl dann nochmals leicht an. Unter einer Sequenzidentität von 40% APSI lassen sich jedoch nur noch Alignments mit 2 und 3 Sequenzen erstellen, da hier die Kombinationsmöglichkeiten höher sind, als für die Fälle mit mehr Sequenzen.

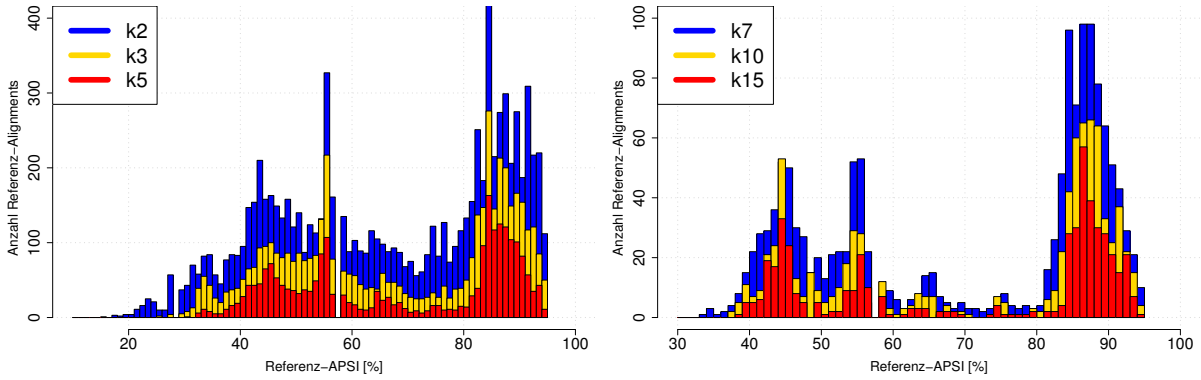


Abbildung 3.19: Histogramm der Alignment-Anzahl verteilt über den Sequenz-Identitätsbereich (APSI). Links: Verteilung der Alignments mit 2, 3 und 5 Sequenzen. **Rechts:** Verteilung der Alignments mit 7, 10 und 15 Sequenzen. Die Werte sind nicht kumulativ aufgeführt; die Verteilungen für kleinere ks sind jeweils im Hintergrund dargestellt. Man beachte, dass die Achsen der beiden Plots nicht identisch skaliert sind.

3.7.3 Eingesetzte Alignment-Programme

Im Gegensatz zu dem in Abschnitt 3.6 geschilderten Benchmark wurden hier (mit der Ausnahme von STRAL) ausschließlich Sequenz-Alignment-Programme getestet. Dabei kamen in den meisten Fällen neue Versionen der Programme zum Einsatz und es wurden zusätzliche Optionen getestet. In Abschnitt 2.2.1 sind alle Versionen und Kommandozeilenparameter sowie hier verwendete Kürzel detailliert aufgelistet. Mit DIALIGN-T (Subramanian *et al.*, 2005), PRANK (Löytynoja & Goldman, 2005) sowie STRAL (Dalli, 2006) kamen zudem drei neue Programme hinzu. PRANK wäre in Abbildung 3.13 als probabilistisches, progressives Programm ähnlich PROALIGN einzuordnen (siehe Abschnitt 3.1.13). DIALIGN-T verfolgt – als Weiterentwicklung von DIALIGN – einen Konsistenz-basierten, iterativen Ansatz (siehe Abschnitt 3.1.4). STRAL ist ein neuartiges Programm, welches progressiv mit Hilfe kondensierter Basenpaarungswahrscheinlichkeitsmatrizen eine Mischung aus Sequenz- und Struktur-Alignment erstellt (siehe Abschnitt 3.1.17).

3.7.4 Eingesetzte Bewertungsmaße

Ähnlich wie im Abschnitt 3.6 werden auch im Folgenden zwei Maße eingesetzt, um die Leistung der Programme zu beschreiben. Als Maß der Struktur-Konservierung eines Alignments wurde wiederum der SCI (siehe Abschnitt 3.5.4) verwendet. Da das zuvor hierfür genutzte Programm RNAZ aber, wie bereits in Abschnitt 3.6.4 erwähnt, einige fehlerhafte Routinen beinhaltet und zudem mehr Information berechnet als notwendig, wurde hier ein eigenes Programm namens SCIF entwickelt, welches keine spezielle Formatierung benötigt und ausschließlich den SCI berechnet. Damit konnten Fehler vermieden und die Berechnung erheblich beschleunigt werden.

Auch das Programm BALI_SCORE, welches zur Berechnung der SPS eingesetzt wurde, besitzt einige unerwünschte Eigenheiten. Neben den in Abschnitt 3.5.1 erwähnten Eigenarten müssen die Alignments ein spezielles Format aufweisen und das Programm bricht mit einem Fehler ab, falls die Sequenzen in Referenz- und Test-Alignment nicht gleich sortiert sein sollten. Hier wurde mit dem eigens entwickelten Programm COMPALIGNP Abhilfe geschaffen. COMPALIGNP ist eine modifizierte Variante von COMPALIGN, welches im Programmpaket SQUID (Eddy, 2005) enthalten ist. Das berechnete Maß wird im Folgenden SPS' genannt (siehe Abschnitt 3.5.2 für eine genauere Beschreibung). Es lässt sich zeigen, dass die Kurvenverläufe bei der Bewertung von Alignments mit BAliScore bzw. COMPALIGN oder COMPALIGNP lediglich einen leichten Versatz zeigen. Die beiden Maße liefern also nahezu identische Ergebnisse (Daten nicht gezeigt; siehe dazu Dalli, 2006).

Weiterhin lässt sich zeigen, dass die beiden Maße SPS und SCI miteinander korrelieren (Mainz, 2006). Zur Vereinfachung der statistischen und graphischen Auswertung wurden deshalb die beiden Maße – ähnlich zur Vorgehensweise, die bei der Rangfolgen-Bestimmung in Abschnitt 3.6 (siehe Tabelle 3.4) verwendet wurde – durch Multiplikation zu einem neuen Maß kombiniert. Dieses Maß wird im Folgenden Braliscore genannt und bezeichnet somit das Produkt berechnet aus den Werten von SPS' (Äquivalenzmaß zur SPS) und des mit Hilfe von SCIF berechneten SCI.

3.7.5 Statistische Methoden

Um eine statistische Aussage über die Leistung der verschiedenen Programme treffen zu können, wurden zwei verschiedene statistische Methoden angewendet: Friedman-Tests (siehe Abschnitt 2.6.1), die es erlauben eine Rangfolge der Programme anhand der Leistung zu erstellen, und Wilcoxon-Rangsummentests (siehe Abschnitt 2.6.2) um entscheiden zu können, ob signifikante Leistungsunterschiede zwischen zwei Programmen vorliegen.

Bei ersten Tests, die als Datengrundlage alle Ergebnisse verwendeten, zeigte sich eine unerwartet hohe Anzahl nicht signifikanter Unterschiede zwischen den Leistungen der einzelnen Programme. Ursache hierfür war die hohe Anzahl von Test-Alignments mit einer Referenz-Sequenz-Identität größer 80% APSI (siehe Abbildung 3.19). Ab einem solch hohen Sequenzhomologie-Grad sind die Leistungsunterschiede zwischen den Programmen, wie bereits in Abschnitt 3.6 gezeigt, nur noch marginal, da das Alignment-Problem nahezu trivial wird. Aus diesem Grund werden die Rangsummen asymmetrisch verzerrt. Um dem entgegenzuwirken, wurden nur Alignments mit einer Referenz-Sequenz-Identität kleiner gleich 80% APSI in den folgend gezeigten statistischen Auswertungen eingesetzt.

Eine Alternative wäre, wie in Abschnitt 3.6.5 geschehen, verschiedene Homologie-Gruppen zu unterscheiden (siehe Tabelle 3.4). Da hier aber jeweils eine „globale“ Rangfolge erstellt werden sollte und die Grenzen der Homologie-Gruppen mit variierenden Sequenz-Zahlen nicht identisch bleiben, wurde auf eine solche Einteilung verzichtet.

3.7.6 Einfluss der Sequenz-Anzahl

Der unterschiedliche Kurvenverlauf der Leistung von PROALIGN und CLUSTALW für fünf Sequenzen bzw. für zwei Sequenzen (siehe Abbildung 3.15 und Abbildung 3.16) gibt einen Hinweis darauf, dass die Anzahl der Sequenzen einen Einfluss auf die Leistung hat. In Abbildung 3.20 ist exemplarisch der Einfluss der Sequenz-Anzahl auf die Leistung von CLUSTALW und PRRN gezeigt. CLUSTALW dient hier als repräsentatives Beispiel für einen nicht-iterativen Ansatz, PRRN für einen iterativen Ansatz.

Bei CLUSTALW sind die Leistungsunterschiede in Abhängigkeit von der Sequenz-Zahl stärker ausgeprägt, wenn als Bewertungsmaß der SCI statt des SPS-Äquivalents SPS' genutzt wird. Über einer Referenz-Sequenzidentität von 55% APSI fällt die Leistung mit steigender Sequenzzahl in beiden Fällen; unterhalb der 55% verhält sich dies eher umgekehrt. Im Falle von PRRN ist der Kurvenverlauf für die mit SPS' bestimmten Daten ähnlich dem vom CLUSTALW. Bei Anwendung des SCI zeigt sich jedoch im Sequenz-divergenten Bereich kleiner 55% APSI deutlich, dass mit steigender Sequenz-Zahl auch bessere Alignments erzeugt werden.

Um die Leistung direkt vergleichen zu können, wurden die Werte in Abbildung 3.20 C voneinander subtrahiert. Steigt die Referenz-Sequenzidentität über 70% APSI, wird die Leistung beider Programm nahezu gleich gut bewertet. Je höher allerdings die Anzahl der Sequenzen ist, umso mehr steigt die Leistung PRRN im Vergleich zu CLUSTALW, sowohl bei der Bewertung mit SPS', als auch bei Bewertung mit dem SCI. Zudem steigt die Leistung ebenfalls in Abhängigkeit von der Sequenz-Divergenz der Alignments.

Dies lässt sich ebenso für andere Kombinationen von iterativen/nicht-iterativen Programmen zeigen (beispielsweise mit PROALIGN und POA statt CLUSTAL, sowie mit MAFFT oder MUSCLE statt PRRN). Die mit der Sequenz-Anzahl und der Sequenz-Divergenz stetig steigende relative Leistung der iterativ arbeitenden Programme ist in allen Beispielen ausgeprägt (Daten nicht gezeigt), im gezeigten Beispiels allerdings am deutlichsten zu erkennen.

3.7.7 Einfluss von Substitutionsmatrizen

Die große Mehrzahl der Alignment-Programme benötigt eine Substitutionsmatrix, welche die Kosten für Substitutionen und Indels (Insertionen/Deletionen) beinhaltet. Einige Programme erlauben es, eine andere als die jeweilige Standard-Substitutionsmatrize zu verwenden. In diesem Abschnitt wird untersucht, inwiefern die Verwendung anderer Substitutionsmatrizen die Leistung einer Auswahl von Programmen beeinflusst.

Dazu kamen eine in Gotoh (1999) veröffentlichte Substitutionsmatrix (im Folgenden Gotoh-Matrix genannt), sowie eine der sogenannten RIBOSUM-Matrizen (Klein & Eddy, 2003) zum Einsatz. Die Werte der Substitutionsmatrix aus Gotoh (1999) spiegeln die Tatsache wieder, dass Transitionen (Substitution von Purin/Purin oder Pyrimidin/Pyrimidin) häufiger auftreten, als Transversionen (Substitution zwischen Purin und Pyrimidin; siehe Li *et al.*, 1985) und beinhaltet bereits den IUPAC-Mehrdeutigkeitscode (Cornish-Bowden, 1985). Der Name der RIBOSUM-Matrizen leitet sich von Ribosomal RNA Substitution Matrix ab. Die Werte hierfür wurden von Klein & Eddy (2003) anhand ribosomaler RNA-Alignments (SSU-Alignments)

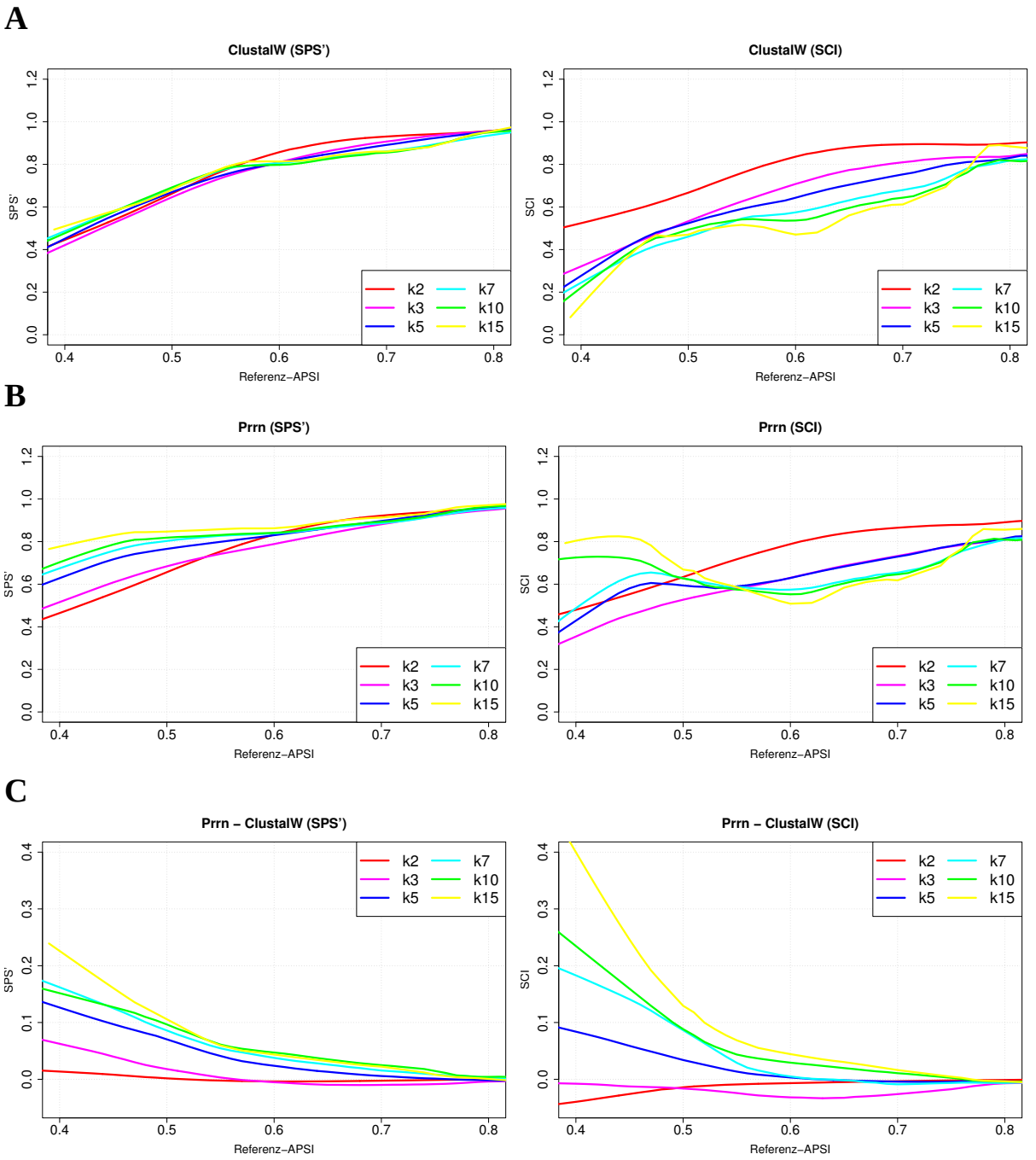


Abbildung 3.20: Einfluss der Sequenz-Anzahl auf die Leistung von iterativ und nicht-iterativ arbeitenden Alignment-Programmen. Die Leistung von CLUSTALW (**A**) als nicht-iterativ arbeitendem Programm und von PRRN (**B**) als einem iterativen Programm ist in Abhängigkeit von der Sequenz-Anzahl k gezeigt. In **C** ist die Differenz der Leistung beider Programme dargestellt. Die Leistung ist jeweils als SPS-Äquivalent SPS$'$ (links) und als SCI (rechts) gegen die Sequenzidentität (APSI) des jeweiligen Referenz-Alignments aufgetragen.

aus der European Ribosomal RNA Database (Van de Peer *et al.*, 1994) erstellt. Im Folgenden wurde ausschließlich die 4x4 RIBOSUM85-60-Matrix verwendet, wobei die entsprechenden Werte für den IUPAC-Mehrdeutigkeitscode hinzugefügt wurden. Da die Höhe der Gap-Kosten und die Werte der Substitutionsmatrizen voneinander abhängen, mussten die Werte der Matri-

zen mit Hilfe einer linearen Funktion auf die Werte der Original-Matrizen skaliert und wenn nötig auf Ganzzahlen gerundet werden.

Die Wahl der Alignment-Programme wurde auf ALIGN-M, CLUSTALW und POA begrenzt, da die Verwendung alternativer Matrizen sich hier relativ problemlos gestaltete. Um eine Rangfolge aufstellen und statistische Unterschiede zwischen der Verwendung unterschiedlicher Substitutionsmatrizen feststellen zu können, wurden wie in Abschnitt 3.7.5 beschrieben Friedman-Tests und Wilcoxon-Rangsummentests angewendet. Bewertungsgrundlage war jeweils die Braliscore und es wurden nur Daten für Alignments mit einer entsprechenden Referenz-Sequenz-Identität $\leq$ 80% APSI berücksichtigt. Die gewonnenen Ergebnisse sind in Tabelle 3.7 in Abhängigkeit von der Sequenz-Zahl zusammengefasst. Hier sei angemerkt, dass die Unterschiede zwischen den Ergebnissen mit den verschiedenen Matrizen bei steigender Sequenz-Zahl k weniger signifikant werden, da in diesen Fällen auch die Zahl der Test-Alignments und damit auch die Zahl der betrachteten Stichproben geringer ist.

Im Fall von CLUSTALW zeigt sich, dass die Verwendung der Standard-Substitutionsmatrize durchweg statistisch signifikant bessere Alignments erzeugt, als die Verwendung der beiden Alternativen. Im Falle von ALIGN-M und POA ist die jeweilige Standard-Substitutionsmatrix immer die schlechtere Wahl, auch wenn die Unterschiede nicht immer statistisch signifikant sind. Die Verwendung der Gotoh-Matrix für POA erzeugt in den meisten Fällen (Ausname $k5$) die besten Alignments (nur für zwei und drei Sequenzen statistisch signifikant). Bei ALIGN-M ist nicht eindeutig zu erkennen, welche der beiden Alternativ-Matrizen zu besseren Ergebnissen führt.

3.7.8 Gapkosten-Optimierung von MAFFT

Die Gapkosten von MAFFT wurden durch den Autor K. Katoh anhand der in Gardner *et al.* (2005) publizierten Daten verbessert. Die in Version 4 des Programms (Katoh *et al.*, 2002) verwendeten Standard-Parameter ($op = 0,51$ und $ep = 0,041$) waren zu gering. Die in der neueren Version 5 (Katoh *et al.*, 2005) verwendeten Parameter wurden entsprechend angehoben ($op = 1,531$ und $ep = 0,123$). Es sei daraufhingewiesen, dass laut Autor bzw. der Homepage von MAFFT [14] geht meist schief wegen sonderzeichen die Parameter nicht gänzlich optimal sind, um so eine Überoptimierung auf das verwendete Datenset zu verhindern.

In Abbildung 3.21 ist der resultierende Leistungsanstieg exemplarisch für fünf Sequenzen ($k5$) dargestellt. Hierzu wurde die Version 5 des Programms mit den neuen, optimierten sowie den alten Parametern aus Version 4 und der Standard-Option `fftns` ausgeführt. Ein direkter Vergleich von Version 4 und Version 5 verbietet sich, da weitere Verbesserungen in die neue Version eingeflossen sind (siehe Katoh *et al.*, 2005).

Die optimierten Parameter führen zu einer messbaren Leistungssteigerung bis zu 80% APSI. Wie in Abschnitt 3.6 beschrieben, war die Leistung von MAFFT zuvor im Vergleich zu anderen Programmen deutlich schlechter. Im Sequenz-divergenten Bereich ($<$50% APSI) schneidet MAFFT mit optimierten Parametern sogar besser ab, als CLUSTALW und PROALIGN, was

[14] http://www.biophys.kyoto-u.ac.jp/ katoh/programs/align/mafft/

Tabelle 3.7: Einfluss der Verwendung verschiedener Substitutionsmatrizen auf die Leistung von ALIGN-M, CLUSTALW und POA. In der Tabelle sind die mit Hilfe des Friedman-Tests erstellten Ränge für jedes Programm einzeln aufgeführt. Konnten mit dem Wilcoxon-Rangsummentests statistisch signifikante Unterschiede zwischen den Rängen festgestellt werden, so sind die entsprechenden Ränge hochgestellt aufgeführt. Unterscheidet sich beispielsweise Rang drei von den Rängen eins und zwei signifikant, so ist dies in der Tabelle als $3^{1,2}$ aufgeführt. Als Bewertungsgrundlage diente die Braliscore, angewendet auf Alignments mit einer entsprechenden Referenz-Sequenz-Identität $\leq$ 80% APSI. Im Falle von ALIGN-M fehlen die Tests für zwei Sequenzen ($k2$), da die Werte hier zu ähnlich sind, womit der Friedman-Test kein (signifikantes) Ranking erstellen konnte. Es sei angemerkt, dass die `blosum80.mat`-Matrix von POA auch Werte für Nukleotid-Substitutionen enthält.

Programm	**Matrize**	$k2$	$k3$	$k5$	$k7$	$k10$	$k15$
ALIGN-M	Standard (`RNA2`)	N/A	$3^{1,2}$	$3^{1,2}$	3^{1}	3	3
ALIGN-M	Gotoh	N/A	2^{3}	2^{3}	2	1	1
ALIGN-M	RIBOSUM	N/A	1^{3}	1^{3}	1^{3}	2	2
CLUSTALW	Standard (einkompiliert)	$1^{2,3}$	$1^{2,3}$	$1^{2,3}$	$1^{2,3}$	$1^{2,3}$	$1^{2,3}$
CLUSTALW	Gotoh	$2^{1,3}$	$2^{1,3}$	$2^{1,3}$	$2^{1,3}$	$2^{1,3}$	2^{1}
CLUSTALW	RIBOSUM	$3^{1,2}$	$3^{1,2}$	$3^{1,2}$	$3^{1,2}$	$3^{1,2}$	3^{1}
POA (p)	Standard (`blosum80.mat`)	$3^{1,2}$	$3^{1,2}$	3^{1}	3	3	3
POA (p)	Gotoh	1^{3}	1^{3}	2	1	1	1
POA (p)	RIBOSUM	2^{3}	2^{3}	1^{3}	2	2	2

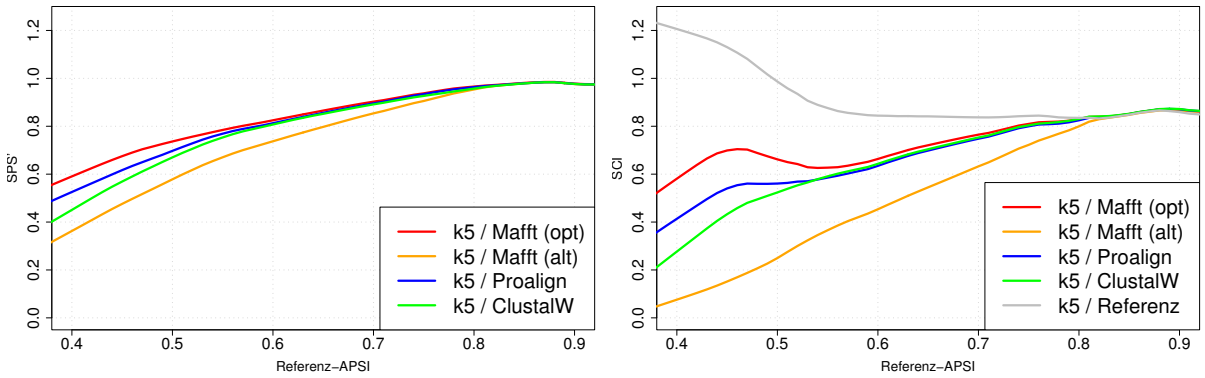


Abbildung 3.21: Leistungsanstieg von MAFFT durch Parameter-Optimierung. Gezeigt ist die Leistung von MAFFT Version 5 (in der Standard-Einstellung `fftns`) mit alten (alt) und neuen, optimierten (opt) Gap-Parametern exemplarisch für fünf Sequenzen ($k5$). Links wurde die Leistung als SPS$'$ (SPS-Äquivalent), rechts als SCI gemessen. Die Werte für CLUSTALW und PROALIGN sind zum Vergleich ebenfalls gezeigt.

einen enormen Leistungsanstieg bedeutet. Es sei daraufhingewiesen, dass die Leistung von MAFFT als iterative Methode bei Verwendung einer höheren Anzahl von Sequenzen noch weiter steigt (siehe Abschnitt 3.7.6).

3.7.9 Gapkosten-Optimierung von CLUSTALW, MUSCLE, PRANK und STRAL

Um die Gap-Kosten-Parameter (üblicherweise Gap-Open und Gap-Extension) der Programme zu optimieren, wurden die jeweiligen Standard-Werte zunächst mit Faktoren zwischen 0,5 und 1,5 (mit einer Schrittweite von 0,25) multipliziert. Die damit erzeugten Alignments wurden mit Hilfe der Braliscore bewertet. Um zu bestimmen, welche Gapkosten-Kombination die besten Alignments erzeugt, wurden Friedman-Tests angewendet, wofür wie in Abschnitt 3.7.5 beschrieben nur Daten solcher Alignments verwendet wurden, deren entsprechende Referenz-Sequenz-Identität kleiner gleich 80% APSI ist. Damit ergaben sich pro Parameter-Kombination sechs Werte – für Alignments mit 2, 3, 5, 7, 10 und 15 Sequenzen. Um eine übersichtliche Darstellung zu ermöglichen, wurden vereinfachend die berechneten Ränge über die sechs verschiedenen Test-Sets gemittelt. Wurde durch diese Variation der Parameter kein eindeutiges Maximum erreicht, wurden die Parameter weiter variiert, womit sich eine zum Teil uneinheitliche Schrittweite in der Parameter-Variation ergab.

Für CLUSTALW liegen die Standard-Werte für die Gap-Open- und Gap-Extension-Kosten bei 15,0 bzw. 6,66. Diese Kosten werden vom Programm automatisch gewählt, sobald DNA/RNA-Sequenzen geladen werden und gelten sowohl für das paarweise, als auch das multiple Alignment. Entsprechend wurde hier ebenfalls nicht zwischen den paarweisen und multiplen Parametern unterschieden (siehe auch die Optionswahl in Abschnitt 2.2.2). Die durch Variation der Parameter ermittelten Ergebnisse sind in Tabelle 3.8 zusammengefasst. Im Mittel wurden Alignments, die mit einer Parameter-Kombination aus Gap-Open-Kosten von 22,5 und Gap-Extension-Kosten von 0,83 erstellt wurden, am besten bewertet. Da die Friedman-Tests lediglich eine qualitative Aussage darüber erlauben, wie sich die Leistung eines Programms im Vergleich zu anderen verhält, ist in Abbildung 3.22 die Leistung von CLUSTALW mit optimierten und Standard-Parametern für alle Test-Sets gezeigt, um eine quantitative Aussage zu ermöglichen. Die hohe Platzierung in den auf der Braliscore basierenden Rangtests (siehe Tabelle 3.8) ist, wie hier zu erkennen, in erster Linie durch einen Anstieg des SCI begründet, wohingegen sich die Bewertung durch das SPS-Äquivalent-Maß SPS' nur wenig verändert. Wie bereits in Abschnitt 3.7.6 gesehen, fällt die Leistung (gemessen als SCI) des Programms mit steigender Sequenz-Zahl deutlich. Im Vergleich zum Leistungsanstieg von MAFFT nach der Optimierung (siehe Abschnitt 3.7.8) fallen die Unterschiede zwischen den optimierten und den Standard-Einstellungen hier eher gering aus.

Ähnlich wie bei CLUSTALW wurde bei der Parameter-Optimierung von PRANK vorgegangen, wobei hier allerdings das Problem entstand, dass mit bestimmten Parameter-Kombinationen aufgrund fehlerhafter Programmabbrüche keine Alignments mit nur zwei oder drei Sequenzen erstellt werden konnten. Aus diesem Grunde sind die Ränge nur über die Test-Sets mit 5, 7, 10 und 15 Sequenzen gemittelt. Wie in Tabelle 3.9 zu sehen, sind die Parameter für Gap-Open- (`gaprate`) und Gap-Extension-Kosten (`gapext`) in der Standard-Einstellung viel zu hoch gewählt. Aus Zeitgründen ließ sich die optimale Kombination nicht mehr ermitteln. Sie wird aber bei Werten liegen, die mindestens zehnfach bzw. vierfach niedriger als die Standard-Werte liegen.

Tabelle 3.8: CLUSTALW-Parameter-Optimierung: Durchschnittliche Rangplatzierung der einzelnen Gap-Parameter-Kombinationen. In der Tabelle sind die durchschnittlichen Ränge über alle Test-Sets aufgeführt. Gap-Extension ist mit ge, Gap-Open mit go abgekürzt. Die gemittelten Ränge für das optimale und das Standard-Werte-Paar sind fett dargestellt. Für Details siehe Text.

	ge 0,42	ge 0,83	ge 1,67	ge 3,33	ge 4,99	ge 6,66	ge 8,32	ge 9,99
go 7,5	56,0	55,0	54,0	53,0	51,2	50,0	47,0	42,8
go 11,25	47,5	44,0	41,5	37,2	34,5	27,3	28,2	31,5
go 15,0	20,8	24,0	20,0	14,5	13,5	**15,5**	22,3	29,3
go 18,75	10,8	8,3	8,2	7,5	11,3	20,8	27,5	35,8
go 22,5	4,7	**2,8**	3,7	8,8	17,7	27,0	34,5	39,2
go 26,25	5,8	5,5	8,8	17,5	31,2	36,7	42,3	46,2
go 30,0	15,2	17,2	22,8	32,8	39,3	45,0	49,0	51,5

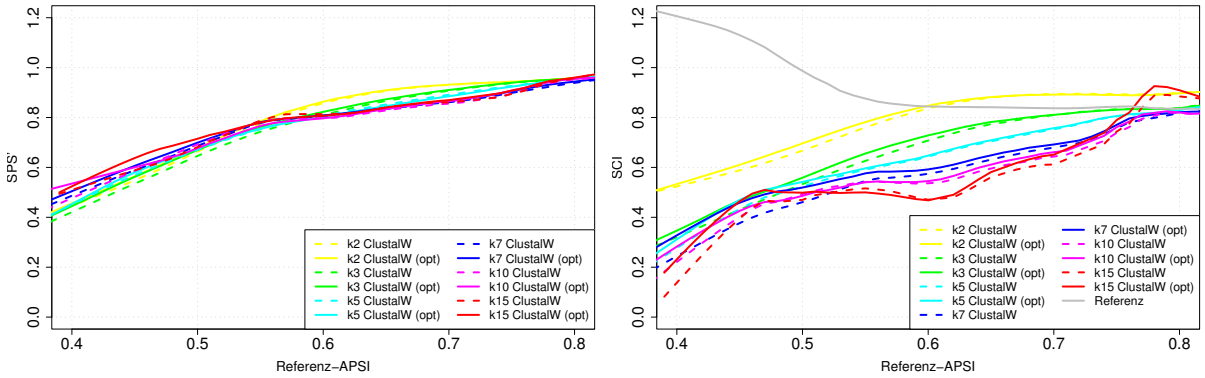


Abbildung 3.22: Leistungsanstieg von CLUSTALW nach Parameter-Optimierung. Gezeigt ist die Leistung von CLUSTALW mit Standard-Parametern, sowie mit optimierten (opt) Gap-Parametern in Abhängigkeit von der Sequenzzahl k (siehe Text). Links wurde die Leistung als SPS$'$ (SPS-Äquivalent), rechts als SCI gemessen.

Weiterhin wurde auf die erwähnte Art versucht, den Gap-Open-Parameter von MUSCLE 3.6 zu optimieren. Der Standard-Wert hängt hier von der „Profile Scoring“-Funktion ab. Diese wird durch das Programm automatisch auf die „Sum-of-Pairs Nucleotide Profile Score“ (SPN) gesetzt, womit sich ein Standard-Gap-Open-Parameter von -400 ergibt. Die Leistung von MUSCLE wurde mit entsprechenden Werten von -200, -300, -500 und -600 getestet. Es ergab sich jedoch in keinem Fall ein höherer Rang (Daten nicht gezeigt). Somit ist dieser Parameter zumindest in Version 3.6 optimal.

Hier sei ebenfalls nur erwähnt, dass die Parameter (Gap-Kosten und Struktur- vs. Sequenz-Gewichtung) des in Entwicklung befindlichen Programms STRAL auch mit Hilfe der hier vorgestellten Vorgehensweise optimiert wurden. Für entsprechende Daten und eine Diskussion wird allerdings auf Dalli (2006) verwiesen.

Tabelle 3.9: PRANK-Parameter-Optimierung: Durchschnittliche Rangplatzierung der einzelnen Gap-Parameter-Kombinationen. In der Tabelle sind die durchschnittlichen Ränge über alle Test-Sets mit 5, 7, 10 und 15 Sequenzen aufgeführt. Der Parameter `gaprate` ist mit gr, `gapext` mit ge abgekürzt. Die gemittelten Ränge für das optimale und das Standard-Werte-Paar sind fett dargestellt. Für Details siehe Text.

	ge 0,05	ge 0,125	ge 0,1875	ge 0,25	ge 0,375	ge 0,5	ge 0,625	ge 0,75
gr 0,0025	3,5	**2,0**	4,8	N/A	N/A	N/A	N/A	N/A
gr 0,00625	6,8	3,5	3,2	N/A	N/A	N/A	N/A	N/A
gr 0,00938	8,8	6,5	8,0	N/A	N/A	N/A	N/A	N/A
gr 0,0125	N/A	N/A	N/A	8,2	11,0	13,5	18,2	24,0
gr 0,01875	N/A	N/A	N/A	12,8	12,5	15,8	21,2	29,8
gr 0,025	N/A	N/A	N/A	15,8	17,2	**19,0**	25,8	31,5
gr 0,03125	N/A	N/A	N/A	20,0	22,0	23,8	28,0	32,8
gr 0,0375	N/A	N/A	N/A	25,0	27,0	27,8	31,5	34,0

3.7.10 Benchmark aller Programme

Ähnlich zu Abschnitt 3.6.5 wird in diesem Abschnitt ein Vergleich aller eingesetzten Programme durchgeführt, um so eine Aussage darüber treffen zu können, welches Programm sich im Allgemeinen am ehesten für das RNA-Alignment Problem eignet.

Für den Test wurde zunächst für jedes der in Abschnitt 2.2.2 genannten Programme durch Friedman- und Wilcoxon-Rangtests (siehe Abschnitt 3.7.5) bestimmt, ob es eine Parameter-Kombination – worunter auch die in Abschnitt 3.7.7 und 3.7.9 optimierten Parameter fallen – gibt, die besser als die Standard-Einstellung abschneidet. Falls ja, wurde diese in den folgend aufgeführten Tests neben der Standard-Einstellung mit einbezogen.

Da aufgrund von unerwarteten Programmabbrüchen für HANDEL keine Alignments mit zwei Sequenzen und für PRANK mit optimierten Parametern keine Alignments mit zwei und drei Sequenzen erzeugt werden können, fehlen die entsprechenden Daten. Da wie bereits im vorherigen Benchmark (siehe Abschnitt 3.6) gesehen, POA wesentlich bessere Alignments mit der globalen Alignment-Option `do_global` erstellt, wurde diese als „Standard“ verwendet. Die Ergebnisse der wie in Abschnitt 2.2.2 beschrieben durchgeführten Friedman-Tests sind in Tabelle 3.10 zusammengefasst und nach durchschnittlicher Braliscore sortiert.

Die Leistung von DIALIGN-T ist im Vergleich zu den anderen Programmen durchgehend am schlechtesten bewertet und unterscheidet sich auch in allen Fällen signifikant vom Rest (entsprechende Wilcoxon-Rangtests nicht gezeigt). In der Rangfolge folgen (nach HANDEL) die ebenfalls auf lokalen Alignment-Strategien bzw. auf Konsistenz-Kriterien basierenden Alignment-Programme DIALIGN und ALIGN-M, die auch mit optimierten Parametern im Vergleich zu anderen Programmen nicht besser abschneiden. Bei PRANK lässt die Leistung mit steigender Sequenzzahl im Vergleich zu anderen Programmen nach. Ähnliches gilt für PROALIGN. Die Leistung von PRRN hingegen wird mit steigender Sequenz-Zahl stetig besser bewertet. Die optimierten Varianten von POA (siehe Abschnitt 3.7.7) und CLUSTALW (siehe Abschnitt 3.7.9) schneiden wie erwartet im Gesamtvergleich deutlich besser ab, als die entsprechenden Standard-Varianten. Sie gehören bereits zu den durchgängig hochbewerteten Program-

men. MAFFT (ginsi), MUSCLE, STRAL sowie MAFFT (ffnts) werden im Vergleich zu den anderen Programmen in allen Test-Sets deutlich besser bewertet. In entsprechenden Wilcoxon-Rangtests ist zu erkennen, dass die Leistung dieser Gruppe in nahezu allen Fällen signifikant besser ist, als die der anderen Programme (Daten nicht gezeigt), wobei sich innerhalb dieser Gruppe wiederum meist die Leistung von MAFFT (ginsi) positiv und signifikant von der Leistung der anderen Programme unterscheidet.

Die Friedman-Tests erlauben zwar eine Aussage darüber, wie konstant die Leistungsunterschiede ausgeprägt sind, nicht jedoch darüber, wie sehr sich die Leistungen voneinander unterscheiden. Um einen visuellen Eindruck der qualitativen Unterschiede zu geben, ist die Leistung der genannten Spitzengruppe in Abbildung 3.23 als Braliscore gegen die Referenz-Alignment-Homologie für jedes Alignment-Set gezeigt. Um eine Abgrenzung gegen die restlichen Programme zu ermöglichen, wurden in die Diagramme die Daten für POA mit aufgenommen, welches gerade eben nicht mehr zu der signifikant besten Programm-Gruppe gehört. Wie zu erkennen sind die Unterschiede mit steigender Sequenzzahl ausgeprägter. Bei Alignments mit zwei Sequenzen ist die Qualität der produzierten Alignments nahezu gleich. Ab sieben Sequenzen und mit fallender Sequenz-Homologie werden die Leistungsunterschiede immer deutlicher. Alignments erzeugt mit MAFFT und der globalen, Konsistenz-basierten Option ginsi werden deutlich besser bewertet, als die der anderen Programme. Eine interessante Ausnahme stellen die mit STRAL erzeugten Alignments dar, welche in einem Homologiebereich von 55%–70% zwar nur leicht, aber konstant besser bewertet werden.

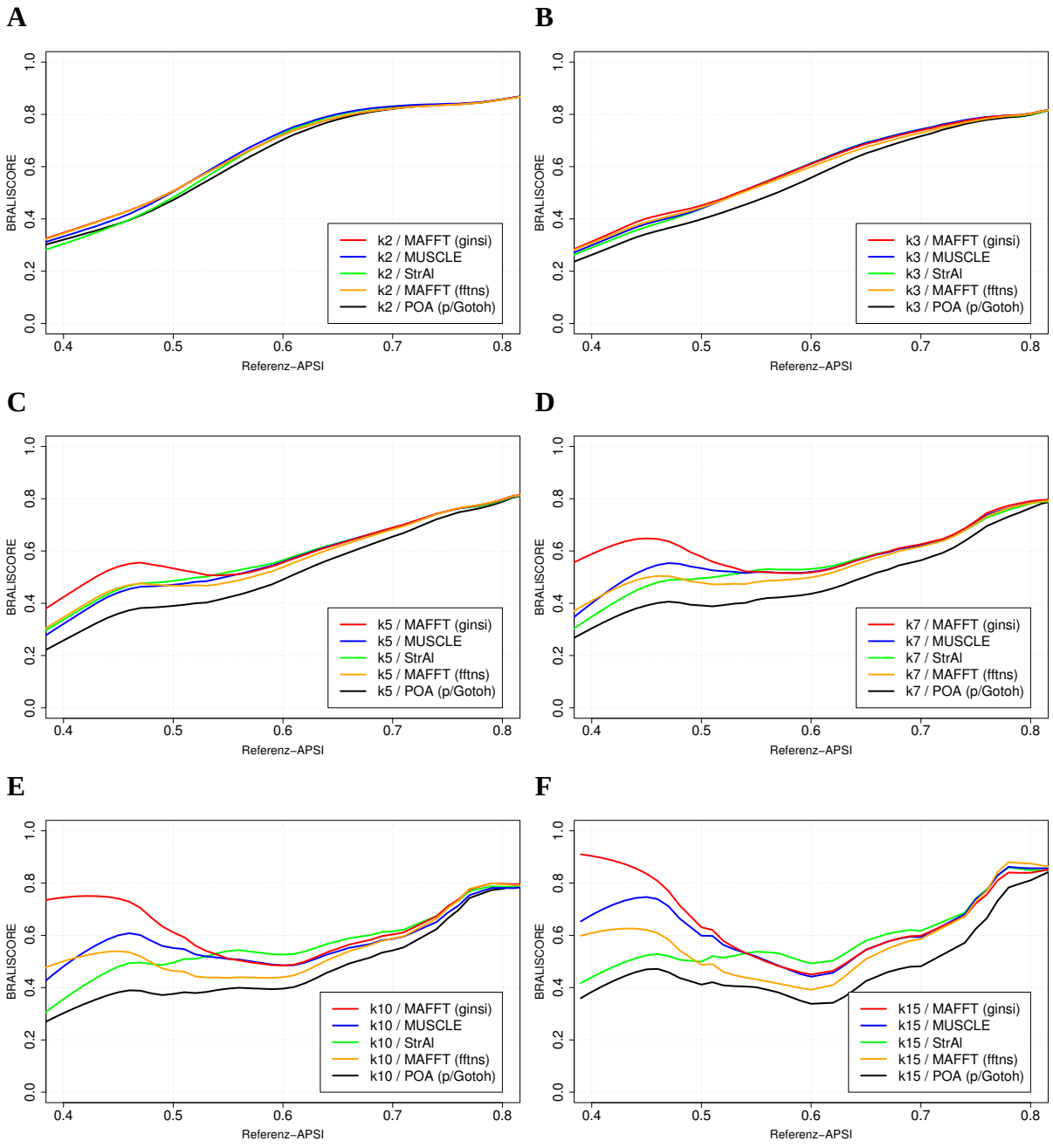


Abbildung 3.23: Leistung der besten Programme. Die Leistung der besten Programme bzw. Programm-Optionen (siehe Tabelle 3.10) ist hier für Alignments mit 2, 3, 5, 7, 10 und 15 Sequenzen gezeigt (A–F). Die Leistung ist als Braliscore gegen die Sequenzidentität (APSI) des jeweiligen Referenz-Alignments aufgetragen.

Tabelle 3.10: Friedman-Test aller eingesetzten Programme. Die Tabelle führt für jedes Test-Set mit k Sequenzen und für jedes Bewertungsmaß separat die per Friedman-Tests bestimmte Rangfolge auf. Die anhand der Braliscore (abgekürzt mit „BS“) bestimmten Ränge sind fett gedruckt. SPS′ bezeichnet das mit COMPALIGN berechnete SPS-Äquivalent-Maß. Das hinter CLUSTALW und PRANK angegebene „opt“ steht für die in Abschnitt 3.7.9 genannten optimierten Parameter. Konnten aufgrund fehlender Alignments keine Ränge bestimmt werden, ist dies mit N/A vermerkt. Die Sortierung erfolgte anhand der über alle Test-Sets gemittelten Braliscore-Bewertung.

	k2			k3			k5			k7			k10			k15		
	BS	SPS′	SCI	BS	SPS′	SCI	BS	SPS′	SCI	BS	SPS′	SCI	BS	SPS′	SCI	BS	SPS′	SCI
MAFFT (ginsi)	**2**	1	3	**2**	1	3	**1**	1	3	**1**	1	2	**1**	1	2	**1**	1	2
MUSCLE	**1**	3	1	**3**	2	2	**2**	2	2	**2**	2	1	**2**	2	1	**2**	3	1
STRAL	**5**	7	2	**1**	5	1	**3**	5	1	**3**	6	3	**5**	6	3	**5**	6	5
MAFFT (fftns)	**3**	2	4	**4**	3	4	**4**	3	4	**4**	3	4	**4**	4	4	**4**	4	3
POA (p/Gotoh)	**6**	4	8	**6**	4	9	**9**	6	11	**6**	5	11	**7**	5	9	**6**	5	7
CLUSTALW (opt)	**8**	9	6	**8**	9	5	**8**	11	5	**7**	11	5	**6**	10	6	**7**	9	6
PRRN	**14**	12	14	**13**	12	14	**7**	4	10	**5**	4	6	**3**	3	5	**3**	2	4
PRANK (opt)	**N/A**	N/A	N/A	**N/A**	N/A	N/A	**5**	7	6	**8**	7	8	**10**	8	10	**10**	8	10
PROALIGN	**7**	6	7	**7**	7	6	**6**	9	7	**10**	9	9	**11**	9	11	**11**	10	11
PRANK	**4**	5	5	**5**	6	7	**11**	10	12	**12**	10	12	**12**	11	13	**13**	12	13
T-COFFEE (lp,sp,cp,cm)	**11**	10	12	**10**	8	10	**10**	8	9	**9**	8	10	**8**	7	8	**9**	7	9
CLUSTALW	**10**	11	9	**9**	11	8	**12**	12	8	**11**	12	7	**9**	12	7	**8**	11	8
PCMA (agi20)	**12**	13	10	**11**	13	11	**13**	14	13	**13**	15	13	**13**	15	12	**12**	15	12
POA	**9**	8	11	**12**	10	12	**14**	13	14	**15**	13	15	**16**	14	16	**15**	13	15
PCMA	**13**	14	13	**14**	15	13	**15**	16	15	**16**	18	16	**15**	16	15	**16**	16	16
ALIGN-M (RIBOSUM)	**18**	19	18	**17**	18	17	**16**	18	16	**14**	14	14	**14**	13	14	**14**	14	14
DIALIGN (o)	**15**	15	15	**16**	16	16	**17**	17	17	**17**	17	17	**17**	17	17	**18**	19	17
ALIGN-M (RNA2)	**18**	19	18	**19**	19	18	**20**	20	19	**18**	20	18	**18**	18	18	**17**	18	18
HANDEL	**N/A**	N/A	N/A	**15**	14	15	**18**	15	18	**19**	16	20	**20**	19	20	**20**	17	20
DIALIGN	**16**	16	16	**18**	17	19	**19**	19	20	**20**	19	19	**19**	20	19	**19**	20	19
DIALIGN-T	**19**	17	19	**20**	20	20	**21**	21	21	**21**	21	21	**21**	21	21	**21**	21	21

4

Diskussion

In diesem Kapitel werden die in den vorangegangenen Abschnitten vorgestellten Ergebnisse diskutiert. Zunächst wird dabei nochmals auf das Programm CONSTRUCT, seine besonderen Fähigkeiten und Eigenschaften, sowie Limitierungen eingegangen (siehe Abschnitt 4.1).

Um die Leistung von Alignment-Programmen angewendet auf RNA-Sequenzen bestimmen zu können, mussten zunächst entsprechende Maße gefunden werden, welche fähig sind, die Eigenarten des RNA-Alignments abzubilden. Eine Diskussion der in dieser Arbeit eingesetzten Maße findet sich in Abschnitt 4.3.

Schließlich wird der Einfluss der Sequenzzahl und der verwendeten Substitutionsmatrix auf die Leistung der Programme (Abschnitt 4.4 und 4.5), sowie die hier durchgeführte Parameter-Optimierung besprochen (Abschnitt 4.6).

Anschließend werden die Ergebnisse der beiden vorgestellten Benchmarks diskutiert (siehe Abschnitt 4.7) und soweit möglich, mit denen anderer Benchmarks verglichen und ein Resumee gezogen.

4.1 CONSTRUCT

Die in CONSTRUCT implementierte semiautomatische Kombination aus Thermodynamik, Sequenz-Alignment, Statistik (gegenseitiger Informationsgehalt) und Benutzer-Intelligenz erlaubt es, die Eigenarten und Limitierungen der einzelnen Methoden zu umgehen. Der Ansatz wurde von Zuker als äußerst elegant („most elegant“; Zuker, 2000) beschrieben und ist mittlerweile einzigartig, da das einzige ähnliche Programm X2S (Juan & Wilson, 1999) nicht mehr weiterentwickelt wurde und der Quellcode nicht mehr erhältlich ist.

Seit der ersten Beschreibung (Lück *et al.*, 1996) wurden die Funktionen des Programms mehrfach erweitert (Lück *et al.*, 1999; Riks, 2001; Wilm, 2002). So ist CONSTRUCT mittlerweile ein flexibles Mehrzweckwerkzeug geworden, das zur Erstellung und Korrektur von RNA-

Alignments, zur Konsensusstruktur-Vorhersage und zur Erstellung von Mustern für Homologie-Suchen eingesetzt wird (siehe beispielsweise Antal *et al.*, 2000; Gräf *et al.*, 2001; Owens *et al.*, 2003). Dabei spielt die Benutzerinteraktion eine entscheidende Rolle, nicht zuletzt da die Leistung von „Mensch und Maschine“ generell denen vollautomatischer Methoden überlegen ist („human plus machine performance“; Fischer *et al.*, 1999).

In den folgenden Abschnitten werden die Eignung von CONSTRUCT als Alignment-Editor (Abschnitt 4.1.1) und zur Konsensusstruktur-Vorhersage (Abschnitt 4.1.2) diskutiert. Neben den kurz in Abschnitt 3.3.5 erwähnten Neuerungen zur Erhöhung der Benutzerfreundlichkeit wird in Abschnitt 4.1.3 die Möglichkeit besprochen, bekannte Strukturen zu berücksichtigen. Schließlich werden auch die Limitierungen des Programms erwähnt (Abschnitt 4.1.4).

4.1.1 CONSTRUCT als Alignment-Editor

Die Qualität von Konsensusstruktur-Vorhersagen (und Phylogenie-Vorhersagen etc.) hängt immer von der Qualität des hierfür eingesetzten RNA-Alignments ab. Da aber gleichzeitig die Erstellung/Berechnung eines RNA-Alignments notorisch schwierig ist (siehe Abschnitt 1.2.6), sollten zur Korrektur dieser Alignments entsprechende Werkzeuge eingesetzt werden, nicht zuletzt da eine ausschließlich manuelle Korrektur sehr fehleranfällig ist.

Die Zahl der speziell auf das RNA-Alignment zugeschnittenen Editoren ist äußerst gering. Das zur Pflege der Rfam „Seed“-Alignments eingesetzte RALLEE („**R**NA **Al**ignment **E**ditor in **E**macs“; Griffiths-Jones, 2005) ist kein eigenständiges Programm, sondern lediglich ein spezieller Modus für den Text-Editor Emacs[1] geht meist schief wegen sonderzeichen . JPHYDIT (Jeon *et al.*, 2005) erlaubt das semiautomatische paarweise Alignment einer neuen Sequenz an eine Template-Sequenz aus einem bereits bestehenden Alignment und zeigt dabei Sekundärstruktur-Informationen an. Keiner dieser Editoren hat eine ähnlich hohe Funktionsvielfalt und eine ähnlich wohl durchdachte graphische Benutzeroberfläche („elaborate GUI“; Zuker, 2000) wie CONSTRUCT. Die Kopplung von thermodynamischem Konsensus-Dotplot und Alignment führt den Benutzer in der Korrektur des Sequenz-Alignments (siehe Punkt 5 in Abbildung 3.6). Falsch alignierte Strukturen sind im Konsensus-Dotplot schnell zu detektieren. So sind beispielsweise misalignierte Helices als diffuse Ansammlung von Basenpaaren zu erkennen. Die graphische Benutzeroberfläche (GUI) erlaubt dabei eine schnelle Zuordnung der Basenpaare des Dotplots zu den entsprechenden 5’- und 3’-Nukleotiden im Alignment-Fenster, deren Position dann entsprechend dem strukturellen Alignment korrigiert werden kann. Die Visualisierung des Struktur-Alignments im Konsensus-Dotplot, welche den Benutzer bei der Korrektur des Alignments leitet, erlaubt es ein RNA-Alignment zu erstellen das hinsichtlich Struktur und Sequenz korrekt aligniert ist, selbst wenn es sich um sehr divergente Sequenzen handelt. Ein extremes Beispiel ist das SECIS-Form2-Alignment aus Fagegaltier *et al.* (2000), welches in wenigen Schritten so weit verbessert ist, dass der SCI von 0,30 auf 0,78 (siehe auch Tabelle 3.2) steigt, ohne dass die Sequenz-Homologie darunter leidet.

[1] http://www.gnu.org/software/emacs/emacs.html

Nach der Korrektur eines initialen Sequenz-Alignments mit Hilfe von CONSTRUCT lassen sich optimale und suboptimale Konsensus-Strukturen, sowie tertiäre Wechselwirkungen vorhersagen.

4.1.2 CONSTRUCT zur Konsensusstruktur-Vorhersage

Nahezu alle Programme benötigen zur Konsensusstruktur-Vorhersage ein zuvor erstelltes, möglichst korrektes RNA-Alignment. Lediglich COMRNA (Ji *et al.*, 2004) und CARNAC (Touzet & Perriquet, 2004) bilden hier eine Ausnahme.

Die Qualität der Vorhersage hängt direkt von der Qualität des Alignments ab. Bekannte Programme wie ILM (Ruan *et al.*, 2004), PFOLD (Knudsen & Hein, 2003) und RNAALIFOLD (Hofacker *et al.*, 2002) benötigen ein fixes Alignment als Eingabe (für einen Vergleich siehe Gardner & Giegerich, 2004). PFOLD versucht etwaigen Alignment-Fehlern zu begegnen, indem es mit einer gewissen Wahrscheinlichkeit berücksichtigt, dass ein Nukleotid falsch aligniert sein könnte. CONSTRUCT hingegen erlaubt die Korrektur des Alignments mit Hilfe seiner graphischen Oberfläche.

Von den drei genannten Programmen sind nur ILM und CONSTRUCT fähig Pseudoknoten und tertiäre Wechselwirkungen vorherzusagen, da in beiden Programmen der MWM-Algorithmus („Maximum Weighted Matching“; nach Tabaska *et al.*, 1998) implementiert ist und als (neben der Thermodynamik optionale) Datengrundlage der gegenseitigen Informationsgehalt („Mutual Information Content“; siehe Chiu & Kolodziejczak, 1991, und Abschnitt 3.3.4) genutzt werden kann.

RNAALIFOLD beruht ebenfalls auf einer Kombination von Thermodynamik und Statistik (Kovarianz), jedoch werden hier nur konsistente Basenpaaraustausche berücksichtigt, also solche Austausche in denen ein Basenpaar durch ein anderes Basenpaar ausgetauscht wurde. Damit soll zum einen das für den gegenseitigen Informationsgehalt typische „statistische Rauschen“ unterdrückt und zum anderen auch Austausche von $\mathrm{G:C}$ nach $\mathrm{G:U}$ *vice versa* berücksichtigt werden. Der gegenseitige Informationsgehalt hingegen nutzt explizit keine Basenpaarungsregeln, womit beispielsweise die Vorhersage von ungewöhnlichen (nicht-kanonischen oder Wobble) Basenpaaren in CONSTRUCT erst möglich wird.

Um dem statistischen Rauschen des gegenseitigen Informationsgehalts entgegenzuwirken, sind in CONSTRUCT eine Reihe von Filtern implementiert, deren Anwendung graphisch unterstützt wird. So lässt sich ein Schwellenwert definieren, unterhalb dessen Daten ignoriert werden. Ein solcher Schwellenwert lässt sich ebenso auf die thermodynamischen Daten anwenden. Diese Werte sind vom Benutzer frei wählbar und lassen sich somit dem jeweiligen Problem (dem jeweiligen Sequenz-Set) anpassen. In keinem der anderen genannten Programme ist eine Filterung dieser Art möglich. ILM, RNAALIFOLD und CONSTRUCT erlauben zudem eine Gewichtung zwischen Statistik- (Kovarianz bzw. gegenseitiger Informationsgehalt) und Thermodynamik-Term.

4.1.3 Berücksichtigung bekannter Struktur-Informationen

Eine der Neuerungen von CONSTRUCT ist die in Abschnitt 3.3.6 beschriebene Methode, bereits bekannte Sekundärstrukturen in CONSTRUCT einzubinden bzw. diese zunächst aus 3D-Strukturen zu extrahieren. Zur Bestimmung einer Sekundärstruktur aus einer 3D-Struktur wurde folgendermaßen vorgegangen:

1. Extraktion der gewünschten RNA aus einer PDB-Datei mit Hilfe von RASMOL (Sayle & Milner-White, 1995)
2. Annotation der Nukleotid-Konformationen dieser RNA durch MC-ANNOTATE (Gendrona *et al.*, 2001)
3. Bestimmung der Sekundärstruktur mit niedrigster Peculiarity-Summe (siehe Abschnitt 3.3.6) per Nussinov-Algorithmus (Nussinov *et al.*, 1978)

Am Beispiel der 5S rRNA von *Haloarcula marismortui* (Ban *et al.*, 2000) wurde die Tauglichkeit dieser Vorgehensweise gezeigt.

Eine alternative Möglichkeit wäre die Verwendung der Programme RNAVIEW und RNAMLVIEW (Yang *et al.*, 2003). Mit diesen Programmen wurden mittlerweile alle Strukturen in der NDB (Berman *et al.*, 1992) annotiert. Damit sind zwar die entsprechenden Basenpaare nun also schon Teil der Datenbank-Einträge (siehe dort: „Derivative Data / Base Pair Parameters“), jedoch lassen sich diese nur eingeschränkt nutzen. So ist z. B. für den Fall, dass in einer PDB-Datei mehrere RNAs enthalten sind, eine Zuordnung der Basenpaare zu einer bestimmten RNA kaum möglich, da nicht zwischen den verschiedenen RNAs unterschieden wurde und stattdessen die Basenpaare aller enthaltenen RNAs durchgehend nummeriert vorliegen.

Das Wissen um bekannte oder – wie gerade beschrieben – extrahierte Sekundärstrukturen kann in CONSTRUCT durch entsprechende Einträge in die sogenannten Project-Dateien genutzt werden (siehe Abbildung 3.8). Im Programm werden die Informationen an zwei Stellen eingesetzt: im Dotplot und in der sogenannten Struktur-Alignment-Ansicht. Im Dotplot werden alle falsch vorhergesagten Basenpaare gelöscht, mit dem Ziel die Darstellung übersichtlicher zu machen. Diese Vorgehensweise brachte jedoch nicht den gewünschten Erfolg. Der visuelle Eindruck der im Dotplot dargestellten Strukturverteilungen änderte sich nahezu gar nicht. Möglicherweise waren die Strukturvorhersagen per RNAFOLD schon gut genug, oder die berücksichtigten Struktur-Mapping-Information reichten nicht aus. In der Struktur-Alignment-Ansicht hingegen lassen sich Verstöße der vorhergesagten Konsensus-Struktur gegen die vorgegebenen Struktur-Informationen gut visualisieren (siehe Abbildung 3.9). Somit wird dem Benutzer ein Hinweis gegeben, inwiefern das Alignment modifiziert werden muss (oder auch die Parameter zur Konsensusstruktur-Vorhersage verändert werden sollten).

Bei dieser Vorgehensweise wurde nur die graphische Darstellung der Basenpaare verändert, wohingegen die Werte der entsprechenden Basenpaarungsmatrix unverändert blieben. Eine weitere Möglichkeit, bekannte Sekundärstrukturen oder Basenpaarungen zu berücksichtigen, besteht darin beim Programmaufruf von RNAFOLD bestimmte Basenpaarungen vorzugeben oder zu verbieten (Kommandozeilen-Parameter `-C`). Hierfür müsste RNAFOLD manuell aufgerufen werden, und die entstehenden PostScript-Dateien, in denen die Werte für die Strukturverteilungen gespeichert sind, ausgetauscht werden.

4.1.4 Limitierungen

Grundsätzlich stellt sich – wie bei anderen Programmen auch – das Problem der Parameter-Wahl. Diese sollten im Idealfall durch den Benutzer immer an das jeweilige Datenset angepasst werden. Hier besteht die Gefahr, dass die Parameter so getrimmt werden, dass eine gewünschte Lösung wahrscheinlicher wird. Dieses Problem besteht jedoch bei nahezu allen Programmen, die dem Benutzer die Freiheit der Parameter-Wahl geben. Die Berücksichtigung einfacher Regeln, wie beispielsweise erhöhte Gewichtung des Informationsgehalts bei erwarteten tertiären Wechselwirkungen oder ungewöhnlichen Basenpaaren, ist jedoch in jedem Fall sinnvoll.

Auch wenn der Kern von CONSTRUCT komplett ersetzt wurde, laufzeitintensive Teile des Programms kompiliert vorliegen und der Aufbau der graphischen Elemente optimiert wurden (Wilm, 2002), gibt es nicht zuletzt aufgrund der Komplexität der graphischen Darstellung klare Limitierungen. So lässt sich festhalten, dass der Aufbau und die Aktualisierung der graphischen Oberfläche nach einer Alignment-Modifikation ab einer ungefähren Sequenz-Länge von 400 Nukleotiden auch auf modernen Computern sehr langsam wird. Die Zahl der Sequenzen geht weniger deutlich in das Verhalten von CONSTRUCT ein, sollte jedoch ca. 500 Sequenzen nicht überschreiten.

Da CONSTRUCT kein automatisches Alignment-Programm ist und die Benutzerinteraktion sogar ein wesentlicher Bestandteil der Programm-Strategie ist, kann die Alignment-Korrektur je nach Qualität des initialen Sequenz-Alignments viel Zeit in Anspruch nehmen.

4.2 Eignung der Bewertungsmaße

Um die Leistung der Alignment-Programme zu messen, wurden in den beiden hier vorgestellten Benchmarks (siehe Abschnitt 3.6 und Abschnitt 3.7) aus bereits publizierten Alignments kleinere Referenz-Alignments generiert. Hierfür wurden die enthaltenen Sequenzen zu neuen Alignments kombiniert, wobei anschließend Spalten, die nur Gaps enthielten, gelöscht wurden.

Da es sich anbietet, die Leistung der Programme in Abhängigkeit von der Sequenz-Homologie der Referenz-Alignments zu testen, wurde bei der Kompilation jeweils so vorgegangen, dass die entstehenden Sub-Alignments über einen möglichst breiten Sequenz-Homologie-Bereich variieren. Da hier zwei verschiedene Methoden zur Erstellung der Tests zum Einsatz kamen, unterscheiden sich die so erstellten Alignment-Sets in ihren Eigenschaften (siehe Abschnitt 4.3).

Als Maß für die Sequenz-Homologie der Referenz-Alignments wurde die sogenannte durchschnittliche paarweise Sequenz-Identität (APSI; siehe Abschnitt 3.5.3) eingesetzt. Grundsätzlich wäre es auch möglich gewesen, eine Variante der SOP („Sum-of-Pairs“; siehe Abschnitt 1.2.3; nicht zu verwechseln mit der SPS) einzusetzen.

Die Referenz-Alignments bzw. die entsprechenden dealignierten Sequenzen wurden dann mit Hilfe der zu testenden Programme (re-)aligniert. Die Güte der so berechneten Test-Alignments wurde anschließend auf Sequenz-Ebene anhand der Übereinstimmung mit dem Referenz-Alignment und auf Struktur-Ebene anhand der im Alignment enthaltenen Sekundärstruktur-Information bzw. -Konservierung bestimmt.

Als Maß der Sequenz-Übereinstimmung wurde die SPS („Sum-of-Pairs-Score“; auch BAliScore) eingesetzt, da sie einen Standard für Alignment-Benchmarks darstellt (siehe Abschnitt 3.5.1 und Thompson *et al.*, 1999a). Es existiert eine Reihe ähnlicher oder daraus abgeleiteter Maße, wie beispielsweise die „Total Column Score“ (Thompson *et al.*, 1999a), die „Modeller/Developer Score“ (Sauder *et al.*, 2000), die „*Q* Score“ (Edgar, 2004b) oder die „Overlap Score“ (Lassmann & Sonnhammer, 2002). In dieser Arbeit wurde neben BALI_SCORE für die Berechnung der SPS auch das Programm COMPALIGN verwendet, dessen Maß vereinfacht gesagt eine Mischung aus SPS und APSI darstellt und hier als SPS-Äquivalent SPS' bezeichnet wird (siehe Abschnitt 3.5.2). Allen diesen Maßen ist gemeinsam, dass sie die Sequenz-Übereinstimmung durch einen Vergleich zwischen Referenz- und Test-Alignment bestimmen.

Die SPS kann als Maß der Sensitivität für ein Sequenz-Alignment bezeichnet werden. Sie bestimmt das Verhältnis der Anzahl korrekt alignierter Paare zur Anzahl aller Paare, d. h. es wird pro Restepaar entschieden, ob es zwischen Test und Referenz-Alignment identisch aligniert ist. Deshalb ist das Maß allerdings auch für bestimmte Fehler anfällig. Sind die Sequenzen beispielsweise nur eine Position gegeneinander versetzt, nimmt die SPS einen Wert von Null an. Ein extremes Beispiel ist in Abbildung 3.10 gezeigt. Weitere Beispiele lassen sich mit Abfolgen von gleichen Nukleotiden konstruieren (ähnlich der Abfolge von As in der genannten Abbildung), bei denen es grundsätzlich egal sein sollte, ob sie links oder rechts ausgerichtet aligniert werden.

Ein alternatives Maß, welches diese Positionsunterschiede berücksichtigt, ist die „Shift Score“ (Cline *et al.*, 2002). Sie bestimmt die Positionsunterschiede von (nicht korrekt) alignierten Resten in allen paarweisen Alignments und nutzt dabei eine durchdachte Normierung. Damit inkorporiert sie mehr Informationen als die vorgenannten Maße, welche lediglich eine (binäre) Aussage darüber treffen, ob zwei Reste identisch aligniert sind, oder nicht. Gleichzeitig ähnelt sie damit entfernt dem sogenannten „cs_shift“-Maß (Wilm, 2002), das eine ungefähre Aussage über die Anzahl an Modifikationsschritten trifft, welche benötigt werden, um ein Test- in ein Referenz-Alignment mit Hilfe von CONSTRUCT zu überführen.

Theoretisch hätte auch die APSI als Bewertungsmaß für die Test-Alignments eingesetzt werden können. Dies erwies sich jedoch als wenig aussagekräftig (Daten nicht gezeigt). Die Kurvenverläufe der Leistung der Programme waren nahezu identisch und entsprachen in etwa der jeweiligen Sequenz-Homologie der Referenz (die ja ebenfalls in Form der APSI gemessen wurde).

Ein geeignetes (genormtes) Maß für die Struktur-Homologie eines RNA-Alignments zu finden, erwies sich u. a. deshalb als schwierig (siehe auch Diskussion in Abschnitt 3.5), da auf keine bereits bekannten Strukturen zurückgegriffen werden konnte und auf Konsensusstruktur-Vorhersagen, aufgrund möglicher Fehler in dieser Vorhersage, weitestgehend verzichtet werden sollte.

In dieser Arbeit wurde der sogenannte SCI („Structure Conservation Index“; siehe Abschnitt 3.5.4 und Washietl *et al.*, 2005) eingesetzt. Der SCI wurde ursprünglich entwickelt, um in genomischen Alignments neue ncRNAs vorherzusagen. Das Maß eignet sich hervorragend zur Bewertung von RNA-Alignments, da es die in einem RNA-Alignment enthaltene Sekundärstruktur-Konservierung bestimmt. Dies geschieht, indem das Verhältnis aus den MFE-Energien

der im Alignment enthaltenen Sequenzen und der Konsensus-MFE des Alignments gebildet wird. Sind Sekundärstruktur-Elemente korrekt aligniert, so sollte die Konsensus-MFE in etwa dem Durchschnitt aller einzelnen Energien entsprechen.

Der SCI ist relativ robust gegenüber Fehlern in der Strukturvorhersage: sollten die Einzelstrukturvorhersagen aufgrund fehlender Parameter, ungewöhnlicher Basenpaare etc. scheitern, so wird mit hoher Wahrscheinlichkeit auch die Konsensusstruktur-Vorhersage scheitern *vice versa* und der SCI bleibt als Quotient aus beiden Werten relativ unbeeinträchtigt. Außerdem ist diese Bestimmung der Sekundärstruktur-Konservierung vollkommen unabhängig von einem Referenz-Alignment. Dies ist insofern wichtig, als dass aufgrund der hohen Zahl von automatisch generierten Referenz-Alignments die Korrektheit derselben nicht immer garantiert werden kann. Bei Anwendung eines Maßes wie der SPS wird das Referenz-Alignment aber als richtige Lösung definiert. Mit Hilfe des SCI hingegen lassen sich bei Vergleich von (definiertem) Referenz- und Test-Alignment u. U. sogar Fehler in der „Referenz“ aufdecken. Ein ähnliches strukturbasiertes Maß existiert für Protein-Alignments nicht. Hier gibt es jedoch eine Reihe von Programmen, die die Qualität eines Alignments in Abhängigkeit zu einer bereits aufgelösten Struktur beschreiben (beispielsweise ADPB; siehe O'Sullivan *et al.*, 2003).

Dadurch, dass der SCI Konsensus-MFE und durchschnittliche Einzelstruktur-MFEs in Relation setzt, ist der Wert „normiert“ und ein Vergleich von Alignments unterschiedlicher Sequenzzusammensetzung wird möglich. Allerdings sind diesem Vergleich theoretisch Grenzen gesetzt, da mit steigender Sequenz-Zahl die u. U. auch steigende Anzahl Basenpaaraustausche zu leicht erhöhten Werten der Konsensus-MFE und damit des SCI selbst führen könnte. Dieser Effekt ließ sich hier jedoch nicht beobachten.

SPS und SCI bilden ein sich komplementierendes Gespann zur Bewertung von RNA-Alignments, da das eine Maß die Sequenz-Übereinstimmung mit einer Referenz und das andere Maß die Sekundärstruktur-Konservierung in einem Alignment misst. Zudem kann gezeigt werden, dass sie miteinander korrelieren (Mainz, 2006). Insofern war es auch folgerichtig das Produkt aus beiden Maßen zu nutzen (hier Braliscore genannt), um eine Rangfolge der getesteten Programme zu erstellen (siehe Tabelle 3.4 bzw. insbesondere Tabelle 3.10).

Die Leistung der Programme wurde hier immer nur in Abhängigkeit von der Sequenz-Homologie in Form der APSI bestimmt. Grundsätzlich wäre auch eine Untersuchung der Leistung in Abhängigkeit von dem Referenz-SCI oder dem Produkt aus Referenz-SCI und Referenz-APSI interessant.

4.3 Qualität und Eigenschaften der Test-Sets

Da wie eingangs erwähnt eine manuelle bzw. durch CONSTRUCT unterstützte Erstellung einer großen Zahl von Referenz-Alignments mit variierenden Sequenz-Eigenschaften aus Zeitgründen nicht möglich war, wurden in dieser Arbeit Referenz-Alignments durch Neukombination von Sequenzen aus großen Quell-Alignments erstellt.

Im ersten Benchmark (Abschnitt 3.6) wurden hierfür sowohl „Seed“- als auch „Full“-Alignments der Rfam (hier Griffiths-Jones *et al.*, 2003) sowie ein Alignment eukaryotischer SRP-

RNAs der SRP-Datenbank (Rosenblad *et al.*, 2003) verwendet. Die in Gardner *et al.* (2005) genannten Probleme mit dem SRP-RNA-Alignment bzw. -Datenset erwiesen sich aufgrund von Programmfehlern als falsch (siehe Anmerkung in Abschnitt 3.6.2). Die genannten Quell-Alignments wurden verwendet, um Referenz-Alignments zu je fünf Sequenzen (bzw. zwei Sequenzen) zu erstellen (siehe ebenfalls Abschnitt 3.6.2). Hierdurch konnten pro RNA-Familie bzw. Quell-Alignment je 100 Alignments konstruiert werden, deren Verteilung über den APSI-Bereich nur leicht ungleichmäßig ist (siehe Abbildung 3.12). Bei 55–70% APSI zeigt sich ein leichtes Maximum und unterhalb von 50% fällt die Zahl der Alignments drastisch ab. Für das paarweise tRNA-Set lassen sich aufgrund der höheren Kombinationsmöglichkeiten noch Referenz-Alignments bis zu einer Sequenz-Identität von 15% APSI erstellen. Gleichzeitig zeigen die Alignments eine gute Struktur-Konservierung mit einem durchschnittlichen SCI von 0,87 bzw. 1,05 (siehe Tabelle 3.3).

Im zweiten Benchmark (Abschnitt 3.7) wurden neue Referenz-Alignments erstellt. Da im ersten Benchmark als Datenquelle lediglich fünf RNA-Familien eingesetzt wurden, war es theoretisch möglich, dass die gemessene Leistung der Programme von den Eigenschaften der (wenigen) RNA-Familien abhing. Zudem kamen dort zwei „Full"-Alignments der Rfam zum Einsatz, die durch Erweiterung der „Seed"-Alignments über eine automatische Datenbanksuche (mittels INFERNAL Eddy, 2002) entstehen und damit möglicherweise nicht homologe Sequenzen enthalten. Im zweiten Benchmark wurden aufgrund dessen nur die (im Vergleich zu den „Full"-Alignments) qualitativ hochwertigeren „Seed"-Alignments der Rfam (hier Version 7.0; Griffiths-Jones *et al.*, 2005) als Datenquelle eingesetzt. Um einen Einfluss einer dominierenden RNA-Familie möglichst auszuschließen, wurden der Rfam insgesamt 36 „Seed"-Alignments entnommen (siehe Tabelle 3.5). Weiterhin wurde in der zur Kompilation eingesetzten Methode (siehe Abschnitt 3.7.2) sichergestellt, dass nur Referenz-Alignments mit einem SCI größer 0,6 konstruiert wurden. Insgesamt sollte so eine hohe Qualität der Referenz-Alignments sichergestellt werden.

Wie in Tabelle 3.6 gezeigt, liegt der SCI der entstandenen Referenz-Alignments erstaunlich hoch, bedenkt man, dass bei der Kompilation lediglich ein SCI von mindestens 0,6 vorgegeben war. In den meisten Fällen nimmt der SCI mit der Anzahl der Sequenzen in den Alignments ab, obwohl hier theoretisch kompensatorische Basenpaaraustausche zu höheren Werten führen könnten. Einzige Ausnahme stellen die Histon3- und tRNA-Alignments dar. In der genannten Tabelle ist ebenfalls zu erkennen, dass die Alignments von 5S rRNA, HIV, HCV IRES, tRNA und TAR in ihrer Anzahl deutlich dominieren. Mit steigender Sequenzzahl k der Referenz-Alignments ist dieser Effekt immer ausgeprägter. Bei 15 Sequenzen bestehen die Referenz-Alignments fast ausschließlich aus Alignments dieser RNA-Familien. Eine ausgeglichenere Verteilung zwischen den RNAs, ähnlich zu dem Datenset aus dem ersten Benchmark, wäre wünschenswerter gewesen.

Die Verteilung der Anzahl Alignments über den Sequenz-Homologie-Bereich ist ungleichmäßiger als im Datenset des ersten Benchmarks (siehe Abbildung 3.19), wobei dies auch durch die wesentlich höhere Anzahl erstellter Alignments begründet ist. Über 80% APSI lässt sich eine sehr hohe Zahl Alignments generieren. Zwischen 60% und 80% APSI ergibt sich ein drastisches Minimum und unterhalb von 60% steigt die Anzahl dann wieder leicht an (warum sich bei 58% kein Alignment erstellen ließ bleibt unklar). Die ungleichmäßige Verteilung zeigt sich bei allen

Sequenzzahlen ($k2$–$k15$). Hierbei ist zu beachten, dass die Sequenzen eines Referenz-Alignments bei einem paarweisen Vergleich in etwa die gleiche Sequenz-Identität wie das Alignment selber aufweisen und somit nahezu äquidistant sind. Dies ist so, da bei der Kompilation der Test-Sets im ersten Schritt immer nur solche Sequenzen verwendet wurden, deren paarweise Sequenz-Identität bereits in etwa der gewünschten Sequenz-Identität des Ausgangsalignments entspricht (siehe Abbildung 3.17). Einfach gesagt besteht ein Alignment mit beispielsweise 60% APSI aus Sequenzen, die untereinander ebenfalls in etwa eine solche Sequenz-Identität aufweisen. Übertragen auf die eben erwähnte ungleichmäßige Verteilung heißt das, dass auch in den Ausgangsalignments bereits eine solche ungleiche Verteilung vorlag, diese also zum Großteil aus paarweise hochhomologen und paarweise divergenten Sequenzen bestehen.

Vergleicht man die Datensets des ersten und des zweiten Benchmarks, so lässt sich in aller Kürze festhalten, dass eine hohe Qualität der Alignments des ersten Benchmarks nicht garantiert ist, im Gegensatz zu denen des zweiten Benchmarks. Dafür ist bei den Referenz-Alignments des ersten Benchmarks eine gleichmäßige Verteilung der Anzahl der Alignments über die fünf RNA-Familien gegeben. Die erwähnten Schwächen der Alignments des zweiten Benchmarks (ungleichmäßige Verteilung über den Sequenz-Homologie-Bereich und über die RNA-Familien) werden allerdings durch die hohe Anzahl Alignments und durch die variierende Sequenz-Zahl ausgeglichen. Hierdurch war eine Messung der Leistung in Abhängigkeit von der Sequenz-Zahl überhaupt erst möglich.

Eine intelligente Reduktion des Datensets wäre wünschenswert. So könnte beispielsweise die Anzahl Alignments pro 1%igem-Identitätsintervall auf etwa 10 reduziert werden, wobei gleichzeitig eine Überrepräsentation von Alignments bestimmter Familien verhindert werden könnte. Ein alternativer Weg zur Erstellung von Referenz-Alignments wird in Abschnitt 4.8 kurz diskutiert.

4.4 Einfluss der Sequenzzahl

Im ersten Benchmark wurden Referenz-Alignments mit fünf Sequenzen (dort Sequenz-Alignment-Datenset genannt) und mit zwei Sequenzen (Struktur-Alignment-Datenset genannt) eingesetzt. Der unterschiedliche Kurvenverlauf der Leistung von PROALIGN und CLUSTALW (siehe Abbildung 3.15 und Abbildung 3.16) gibt einen ersten deutlichen Hinweis darauf, dass die Leistung der Programme von der Sequenzzahl abhängt (für Proteine erstmals durch McClure *et al.*, 1994, gezeigt). In diesem Fall könnte allerdings auch eine Abhängigkeit von den Eigenschaften der tRNA-Sequenzen vorliegen, welche alleinige Datengrundlage für das Struktur-Alignment-Datenset waren. In Abbildung 3.16 sind zwei Grenzwerte bei 40% bzw. 60% APSI zu erkennen, bei denen die Leistung der beiden Sequenz-Alignment-Programme jeweils deutlich fällt. Diese markanten Punkte sind bei Anwendung auf das Sequenz-Alignment-Datenset mit fünf Sequenzen (siehe Abbildung 3.15) nicht ausgeprägt.

Der Einfluss der Sequenzzahl wurde in Abschnitt 3.7.6 genauer untersucht. Dabei wurden in Abbildung 3.20 exemplarisch ein iterativ-arbeitendes Programm (PRRN) und ein nicht-iterativ arbeitendes Programm (CLUSTALW) gegenübergestellt. Wird die Leistung von CLUSTALW als

SPS′ gemessen, so sind die Unterschiede in der Leistung bei steigender Sequenzzahl nicht allzu ausgeprägt. Im homologen Bereich ($>$ 60% APSI) fällt die Leistung mit steigender Sequenz-Zahl leicht; unterhalb 60% APSI gilt der Umkehrfall. Wird die Leistung allerdings als SCI gemessen, so fällt sie deutlich mit steigender Sequenzzahl.

Die Leistung von PRRN – als iterativ arbeitendem Programm – verhält sich bei Betrachtung der SPS′ ähnlich wie bei CLUSTALW, auch wenn die Unterschiede ausgeprägter sind. Je divergenter die Sequenzen und je höher die Sequenzzahl umso besser ist die Leistung. Das lässt sich hier aber im Gegensatz zu CLUSTALW auch bei Betrachtung des SCI feststellen, vorausgesetzt, die Sequenzen sind divergent genug ($<$ 55% APSI).

Stellt man beide Programme gegenüber (C in Abbildung 3.20) zeigt sich klar, dass PRRN als iteratives Programm im Vergleich zu CLUSTALW mit steigender Sequenz-Zahl und zudem fallender Sequenz-Homologie immer besser abschneidet. Dies lässt sich ebenso für andere Kombinationen von iterativ/nicht-iterativ arbeitenden Programmen zeigen (beispielsweise mit PROALIGN und POA statt CLUSTAL, sowie mit MAFFT oder MUSCLE statt PRRN). Sowohl mit steigender Sequenz-Zahl, als auch mit fallender Sequenz-Homologie scheinen die Programme generell mehr Fehler in ein RNA-Alignment einzufügen. Programme, die ein initiales Alignment iterativ verfeinern, haben hier einen klaren Vorteil, da sie Alignment-Fehler im iterativen Zyklus korrigieren können.

Nach Katoh *et al.* (2005) ist diese Fähigkeit in erster Linie von der Anzahl enthaltener homologer Sequenzen abhängig. Dort wurde eine Strategie entwickelt (MAFFTE.RB) mit deren Hilfe dieser Umstand ausgenutzt wird, um die Alignment-Qualität zu erhöhen. Dafür werden zu einem zu alignierenden Sequenz-Set automatisch homologe Sequenzen hinzugefügt, dann alle Sequenzen aligniert und schließlich die zuvor hinzugefügten Sequenzen wieder entfernt. Dadurch ließ sich die Alignment-Qualität im Vergleich zu der Variante, in der keine homologen Sequenzen hinzugefügt werden, deutlich verbessern.

4.5 Einfluss von Substitutionsmatrizen

In Abschnitt 3.7.7 wurde untersucht, inwiefern die Verwendung unterschiedlicher Substitutionsmatrizen die Leistung der Programme beeinflusst. Dafür wurden die Standard-Matrizen von ALIGN-M, CLUSTALW und POA durch die Gotoh-Matrix (Gotoh, 1999) und eine RIBOSUM-Matrix ersetzt (Klein & Eddy, 2003). Die Werte dieser Matrizen wurden so skaliert, dass sie dem Werte-Bereich der Standard-Matrizen entsprachen, um so eine Wechselwirkung mit den Gapkosten auszuschließen.

In allen drei Fällen wurden Friedman-Tests und Wilcoxon-Rangtests zur statistischen Validierung der Ergebnisse eingesetzt. Interessanterweise sind die Ergebnisse vollkommen widersprüchlich (siehe Tabelle 3.7): Für CLUSTALW ist die Standard-Matrix immer die signifikant beste Wahl. Für die Standard-Matrizen von POA und ALIGN-M gilt das Gegenteil. Für POA eignet sich die Gotoh-Matrix am ehesten. Für ALIGN-M lässt sich keine eindeutige Tendenz feststellen. Die Gründe hierfür sind unklar. Aufgrund der Wilcoxon-Rangtests ist ausgeschlossen, dass es sich um zufällige Ereignisse handelt.

Erst kürzlich wurde eine weitere Substitutionsmatrix für Nukleinsäuren veröffentlicht (Wolf *et al.*, 2005a). Die „ITS2 Score“ genannte Matrix wurde anhand einer Homologie-Modellierung von 20000 „rRNA **I**nternal **T**ranscribed **S**pacer 2“-Strukturen (ITS2) konstruiert. Ein Vergleich dieser Matrix bzw. ihr Einfluss auf die Leistung der Programme mit den beiden hier genannten Matrizen steht noch aus.

Es bleibt festzuhalten, dass aufgrund der Abhängigkeit von Gapkosten und Substitutionsmatrix hier neben der Veränderung der Gapkosten theoretisch eine weitere Optimierungsmöglichkeit besteht. So wurde im Falle von POA zwischenzeitlich versehentlich eine nicht an die Standard-Werte angepasste Gotoh-Matrix verwendet, die zu deutlich besseren Ergebnissen, als bei Verwendung aller anderen Matrizen führte (Daten nicht gezeigt). Da es bei der Verwendung von POA nicht die Möglichkeit gibt, Gapkosten anzugeben, ist die Veränderung der Substitutionsmatrix ein alternativer Weg der Parameter-Optimierung.

4.6 Gapkosten-Optimierung

Die vom Autor von MAFFT erfolgreich durchgeführte Optimierung der Gapkosten führte zu einer dramatischen Leistungssteigerung des Programms (siehe Abbildung 3.21). Die Optimierung fand mit Hilfe der in Gardner *et al.* (2005) publizierten Daten statt und die entsprechenden Parameter wurden in der neuen Version des Programms (Version 5; Katoh *et al.*, 2005) als Standard verwendet (siehe auch Abschnitt 3.7.8). Diese Optimierung führte dazu, dass das im ersten Benchmark deutlich am schlechtesten abschneidende Programme im zweiten Benchmark zu den besten Programmen zählt. Der Autor konnte zudem zeigen, dass sich die Gapkosten von CLUSTALW und PRRN verbessern ließen (Gap-Open- und Gap-Extension-Kosten: 20/0,5 statt 15/5 für CLUSTALW bzw. 10/3 statt 9/2 für PRRN)[2].

Die Idee der Gapkosten-Optimierung wurde hier aufgegriffen und ist in Abschnitt 3.7.9 besprochen. Hier wurde vereinfachend die Braliscore als Bewertungsmaß genutzt, damit eine einfache Bestimmung der Rangfolge mit nur einem Maß möglich war. Zudem wurden, wie in Abschnitt 3.7.5 begründet, nur Referenz-Alignments mit einer Sequenz-Homologie $\leq$ 80% APSI eingesetzt und die Ränge über alle Sequenz-Sets (also $k2$–$k15$) gemittelt. Dies ist zwar eine grobe Vereinfachung – u. a. da sich die veränderten Parameter auch unterschiedlich auf die SPS$'$ und SCI-Bewertung auswirken – jedoch war die Gapkosten-Optimierung so einfach durchführbar.

Obwohl CLUSTALW altbewährt ist und bei Verwendung von Nukleinsäuren (vom Benutzer unbemerkt) angepasste Parameter lädt, war es hier möglich, die Gapkosten zu optimieren (siehe Tabelle 3.8), auch wenn der Leistungsanstieg eher gering ausfällt und sich in erster Linie bei der SCI-Bewertung auswirkt (siehe Abbildung 3.22; ähnliches ließ sich auch bei der Optimierung von MAFFT beobachten, siehe Abbildung 3.21).

[2] pers. Komm.

PRANK (siehe Abschnitt 3.1.13 und Löytynoja & Goldman, 2005) hingegen ist ein relativ junges Programm. Die Gapkosten-Optimierung zeigte hier, dass die Standard-Werte sehr weit vom Optimum entfernt liegen (siehe Tabelle 3.9).

Nach Veröffentlichung von Gardner *et al.* (2005) kündigte der Autor von MUSCLE (siehe Abschnitt 3.1.9 und Edgar, 2004b) an, die Gapkosten des Programms ebenfalls zu optimieren[3]. Mit Hilfe des in dieser Arbeit durchgeführten zweiten Benchmarks (siehe Abschnitt 3.7) konnte gezeigt werden, dass die Parameter in der neuen Version optimal sind (siehe Anmerkung in Abschnitt 3.7.9).

Grundsätzlich scheinen die Parameter der Programme (auch der schon länger bestehenden) anhand von Protein-Alignments optimiert worden zu sein. Bei allen hier getesteten Programmen, die nicht schon aufgrund der Veröffentlichung des ersten Benchmarks (Gardner *et al.*, 2005) optimiert wurden, ließen sich bessere Parameter finden. Die Parameter des noch in Entwicklung befindlichen Programms STRAL (siehe Abschnitt 3.1.17 und Dalli, 2006) wurden von Beginn an anhand der hier vorgestellten Test-Sets optimiert.

4.7 Vergleich der Leistung aller Programme

Da sich die Alignment-Programm-Versionen und -Optionen sowie die Daten-Sets zwischen dem ersten und dem zweiten hier durchgeführten Benchmark unterscheiden, werden die Ergebnisse im Folgenden separat diskutiert.

4.7.1 Benchmark I (BRAliBase II)

Ein erstes überraschendes Ergebnis des in Abschnitt 3.6 geschilderten Benchmarks war, dass die Leistung aller Programme unabhängig vom eingesetzten Bewertungsmaß stark variiert (siehe Abbildung 3.14 für ein Beispiel). Es gibt also Referenz-Sets mit hoher Sequenz-Homologie, bei denen die Alignment-Programme schlecht abschneiden und umgekehrt. Aufgrund der starken Streuung der Datenpunkte wurde bei den dort folgend aufgeführten Plots die Lowess-Glättung eingesetzt.

Sequenz-Alignment-Datenset

Die in Abbildung 3.15 gezeigten Kurvenverläufe der Leistung der einzelnen Sequenzalignment-Programme sind alle recht ähnlich, wenn auch verschoben. Grundsätzlich gilt: je geringer die Sequenz-Homologie, umso geringer ist die Leistung der Programme. Oberhalb 75% APSI schneiden alle Programme in etwa gleich gut ab, sieht man von MAFFT (hier die alte, nicht optimierte Version) ab. Bei diesem Grad an Sequenz-Homologie ist das Alignment-Problem meist trivial und durch Einfügen weniger Gaps gelöst. Der leichte SCI-Anstieg bei 75% APSI ist, wie in Abbildung 3.15 zu sehen, kein Artefakt der Lowess-Glättung, sondern scheint eher eine

[3] pers. Komm.

Eigenschaft der Datensets widerzuspiegeln. Unterhalb von 70% APSI kommt es zu einem deutlichen Leistungsabfall der Programme, welcher sich ab 55% noch steigert. Dieser Leistungsabfall ist bei Bewertung mit dem SCI deutlicher. Hier zeigt sich auch jeweils bei den beiden genannten Grenzwerten eine Veränderung des Referenz-SCI-Verlaufs. An beiden Grenzwerten wird der Leistungsunterschied der einzelnen Programme deutlich. Grundsätzlich lässt sich hier sagen, dass die sogenannte „Twilight Zone“ (nach Doolittle, 1981, siehe auch Anmerkung in Abschnitt 4.8), also der Homologiebereich, ab dem das Alignment mit Sequenz-Alignment-Programmen sehr schwierig wird, bei ca. 55% APSI liegt. Im Allgemeinen sind nur relativ geringe Unterschiede in der Bewertung der Leistung eines Programms zwischen SCI und APSI zu bemerken.

Anhand der genannten Grenzwerte (55% und 75% APSI) wurden Homologiegruppen definiert und durch das Produkt von SCI und SPS eine Rangfolge erstellt (siehe Tabelle 3.4). In allen drei Homologie-Gruppen schnitt PROALIGN gleichmäßig gut ab, wahrscheinlich eine Folge davon, dass die Parameter des Programms anhand von einer hohen Zahl mit ROSE (Stoye *et al.*, 1998) generierter Alignments optimiert wurden. Im hohen und mittleren Homologie-Bereich gehörten MUSCLE und PCMA zu den besten Programmen. Im niedrig homologen Bereich wurden neben PROALIGN und POA (g,p) die beiden iterativen Methoden PRRN und MUSCLE gut bewertet.

Die Leistung von POA ist am besten, wenn es zu einem globalen, progressiven Alignment (g,p) gezwungen wird. Die Leistung von T-COFFEE steigt, wenn die vom Programm benötigte paarweise Alignment-Bibliothek mit CLUSTALW erstellt wird. Dies verwundert nicht, da die CLUSTALW-Alignments selber gut bewertet werden. Ähnliches gilt für PCMA: Hier wird durch den Parameter `ave_grp_id` (agi) angegeben, bis zu welchem Schwellenwert CLUSTALW eingesetzt werden soll, bevor zu einer T-COFFEE-ähnlichen Strategie gewechselt wird. Je länger (niedriger Wert) CLUSTALW eingesetzt wird, umso besser werden die Alignments bewertet. Die lokalen Konsistenz-basierten Methoden ALIGN-M und DIALIGN schneiden – neben MAFFT – sehr schlecht ab.

Struktur-Alignment-Datenset

Der Benchmark der Struktur-Alignment-Programme mit Hilfe des paarweisen tRNA-Datensets (Struktur-Alignment-Datenset) ist in Abschnitt 3.6.6 geschildert. Die Leistung der Programme ist in Abbildung 3.16 gezeigt. Auffällig ist hier, dass die Leistung von STEMLOC (gleichgültig mit welcher Option) sich kaum von der eines Sequenzalignment-Programms (siehe CLUSTALW und PROALIGN dort) unterscheidet: bei 60% APSI kommt es zu einem ersten deutlichen Leistungseinbruch, bei 40% APSI fällt die Leistung nochmals. Die schnelle Variante von PMCOMP, welche kein Alignment der Basenpaarungsmatrizen, sondern der daraus extrahierten Paarungsvektoren (siehe Abschnitt 3.1.11) durchführt, schneidet ähnlich schlecht ab, da sie weder Sequenz-Informationen, noch die kompletten Informationen der Basenpaarungsmatrizen für ein Alignment nutzt (im Gegensatz zur normalen PMCOMP-Variante).

Bei den anderen Struktur-Alignment-Programmen ist die Leistung relativ unabhängig von der Sequenz-Homologie. Die mit DYNALIGN, FOLDALIGN und PMCOMP erzeugten Alignments

werden beim Einsatz des SCI als Bewertungsmaß genauso gut wie die Referenz-Alignments bewertet. Da DYNALIGN nur die Struktur aligniert und keinerlei Sequenz-Information verwendet, ist die Bewertung durch die SPS selbst bei hoch-homologen Sequenzen schlecht. Erstaunlich ist, dass PMCOMP hier besser abschneidet als DYNALIGN, obwohl es in der hier verwendeten Version ebenfalls keine Sequenz-Informationen nutzt. Beim Alignment der Basenpaarungsmatrizen scheint genug „Sequenz-Information“ erhalten zu bleiben.

FOLDALIGN nutzt als einziges der Struktur-Alignment-Programme auch Sequenz-Informationen in Form der RIBOSUM-Matrizen. Die Leistung von FOLDALIGN ist hier sowohl auf Sequenz-, als auch auf Struktur-Ebene exzellent.

4.7.2 Benchmark II (BRAliBase IV)

Die Ergebnisse des zweiten Benchmarks sind in Tabelle 3.10 zusammengefasst. Im Gegensatz zu Tabelle 3.4 sind die Ränge hier mit Hilfe von Friedman-Tests berechnet worden. Hier wurden keine Homologie-Gruppen unterschieden, u. a. da die Grenzen der Homologie-Gruppen mit variierenden Sequenz-Zahlen nicht identisch bleiben. Zudem wurden die Daten für sehr homologe Referenz-Alignments (APSI>80%) nicht mit einbezogen, da diese in ihrer Anzahl überwogen und so einen verzerrenden Einfluss auf die Rangtests gehabt hätten.

Die Leistung der lokalen, Konsistenz-basierten Methoden ALIGN-M, DIALIGN und DIALIGN-T wurde in diesem Benchmark erneut in allen Fällen am schlechtesten bewertet. Das ebenfalls Konsistenz-basierte T-COFFEE rangierte in dem hier durchgeführten Test vor CLUSTALW. Allerdings musste die Bibliothek des Programms (siehe Abschnitt 3.1.18) mit Hilfe von vier verschiedenen und teils nicht dokumentierten, vom Autor empfohlenen[4], paarweisen Alignment-Quellen aufgebaut werden (siehe auch Abschnitt 2.2.2).

Die Leistung des relativ neuen Programms PRANK ist mit der von PROALIGN vergleichbar bzw. sogar leicht besser, wenn die in Abschnitt 3.7.9 erarbeiteten, optimierten Parameter eingesetzt werden. Mit steigender Sequenz-Zahl fällt die Leistung von PRANK, obwohl sie sich aufgrund der implementierten Insertions-Korrektur, welche die bei progressiven Methoden auftretende „Überbestrafung“ von Insertionen (siehe Abbildung 3.4) relativ zu anderen (nicht-iterativen) Methoden bessern sollte. Hier ist eher das Gegenteil der Fall. Allerdings lässt sich anhand von Wilcoxon-Rangtests zeigen, dass mit Hilfe der erwähnten Insertions-Korrektur signifikant bessere Alignments erzeugt werden, als ohne diese Option (Daten nicht gezeigt).

Unter den nicht optimierten Methoden schneiden PRRN und PROALIGN durchschnittlich am besten ab, wobei die Leistung von PRRN als iterativ arbeitendem Programm mit steigender Sequenz-Zahl im Gegensatz zu PROALIGN steigt.

Die drei durchgehend best-bewertetsten Programme sind MAFFT, MUSCLE und STRAL. Anhand von Wilcoxon-Rangtests lässt sich zeigen, dass die Leistung dieser Gruppe in nahezu allen Fällen signifikant besser ist, als die der anderen Programme. Bemerkenswert ist hierbei, dass STRAL trotz seines frühen Entwicklungsstadiums und vor allen Dingen trotz fehlender Iteration zu dieser Gruppe gehört. Zudem scheint die Leistung von STRAL – wie sonst bei

[4] pers. Komm.

anderen nicht-iterativ arbeitenden Programmen zu beobachten – kaum von der Sequenzzahl abzuhängen und im mittleren Sequenz-Homologiebereich zeigt es sogar die knapp beste Leistung (siehe Abbildung 3.23). Gleichzeitig sind MAFFT, MUSCLE und STRAL (und auch POA) sehr schnelle Programme (Daten nicht gezeigt). Innerhalb der genannten Dreiergruppe ist die Leistung von MAFFT (ginsi) die signifikant beste, womit es als das Alignment-Programm der Wahl bezeichnet werden kann.

4.8 Vergleich mit den Ergebnissen anderer Benchmarks

Ein Vergleich mit den Ergebnissen anderer Alignment-Benchmarks ist nur begrenzt möglich, da mit Ausnahme vom Pollard *et al.* (2004) lediglich Protein-Alignment-Benchmarks vorliegen (wie beispielsweise Lassmann & Sonnhammer, 2002; McClure *et al.*, 1994; Thompson *et al.*, 1999a) und diese mittlerweile veraltete Programmversionen benutzten.

Der Benchmark von Pollard *et al.* ist sehr speziell, da er das paarweise (genomische) Alignment von zehn Kilobasen großen Sequenzen testet, die mit Hilfe von ROSE (Stoye *et al.*, 1998) erstellt wurden. ROSE generiert unter Vorgabe eines Substitutionsmodells, einer Ur-Sequenz, entsprechenden Mutationswahrscheinlichkeiten und einer durchschnittlichen Sequenz-Länge Familien von DNA-, RNA- und Protein-Sequenzen inkl. einem zugehörigen, unter der Annahme des Modells „wahren“ Alignments. Die Parameter des probabilistischen Alignment-Programms PROALIGN (siehe Abschnitt 3.1.14) wurden anhand von ROSE-Alignments trainiert, womit u. a. die gute Leistung von PROALIGN in den hier durchgeführten Tests erklärt werden kann. Grundsätzlich ließen sich mit Hilfe von ROSE auch Test-Sets für einen RNA-Alignment-Benchmark generieren. Die hier verwendeten „echten“ Sequenzen haben im Zweifel allerdings den Vorteil, dass sie natürliche Mutationsraten und Nukleotidgehalte widerspiegeln.

Weiterhin finden sich eine große Zahl weiterer Benchmarks in den Publikationen der jeweiligen Alignment-Programme (Edgar, 2004b; Notredame *et al.*, 2000; Van Walle *et al.*, 2004), die in den meisten Fällen auf Daten entsprechender Datenbanken wie OXBench (Raghava *et al.*, 2003), PREFAB (Edgar, 2004b), SABmark (Van Walle *et al.*, 2005) und SMART (Letunic *et al.*, 2004) zurückgreifen. Je nach Verwendung der Daten werden zumeist die Vorteile der eigenen Programme herausgestellt, wobei es zum Teil zu widersprüchlichen Ergebnissen kommt. Beispielsweise wurden ALIGN-M (und DIALIGN) in Van Walle *et al.* (2004) insbesondere bei divergenten Test-Sets besser als beispielsweise CLUSTALW bewertet. Für ALIGN-M heißt es dort, dass es konsistent weniger Reste falsch aligniert als andere Programme. Tatsächlich ist eher das Gegenteil der Fall, wie beispielsweise in Edgar (2004b) und auch in den hier durchgeführten Tests gezeigt wurde.

Thompson *et al.* (1999a) bedienten sich Test-Sets der ersten BAliBASE-Version (hier Thompson *et al.*, 1999b). Die Autoren fanden, dass die sogenannte „Twilight Zone“ (Doolittle, 1981) – der Homologie-Bereich unterhalb dessen die Qualität des Sequenz-Alignments dramatisch fällt – bei ca. 20% Sequenz-Übereinstimmung liegt. Oberhalb dieser Grenze wurden im Mittel 80% der Reste von den Programmen korrekt aligniert. Hier konnte gezeigt werden, dass dieser Bereich für RNA-Alignments mit ca. 55% wesentlich höher liegt, insbesondere wenn

man die strukturelle Konservierung in Form des SCI mit berücksichtigt (siehe beispielsweise Abbildung 3.15 und Abbildung 3.16).

Weiterhin konnten die Autoren zeigen, dass iterativ arbeitende Programme in den meisten Fällen bessere Alignments erzeugen, als andere Methoden. Dies konnte auch hier für RNA-Alignments und insbesondere für divergente Sequenzen und steigende Anzahl an Sequenzen gezeigt werden (siehe Abschnitt 3.7.6 sowie Abbildung 3.10). Thompson *et al.* untersuchten auch den Einfluss sogenannter „Orphans“, also nicht-zugehöriger bzw. sehr divergenter Sequenzen. Sie zeigten, dass die Qualität der mit Hilfe iterativ arbeitender Programme erzeugten Alignments mit steigender Zahl „Orphans“ fällt. Die Leistung progressiv vorgehender Programme wie CLUSTALW war hiervon unbeeinträchtigt. Dieses Phänomen konnte hier für RNA-Alignments nicht untersucht werden, da keine entsprechenden Test-Sets konstruiert wurden, was aber ohne weiteres möglich wäre.

Ein grundsätzlicher Unterschied zur BAliBASE ist natürlich der Aufbau der Test-Sets. In der BAliBASE wurden die Alignments manuell und anhand einer 3D-Superposition (mit Hilfe von LSQMAN; siehe Kleywegt & Jones, 1995) korrigiert. Dies war hier aufgrund der hohen Zahl an Alignments (und auch der allgemein geringen Zahl an aufgelösten RNA-Strukturen) nicht möglich. Zudem wurden die Alignments der BAliBASE so annotiert, dass eine Unterscheidung zwischen korrekt alignierbaren Regionen („Core Blocks“) und nicht alignierbaren Regionen innerhalb eines Alignments unterschieden werden kann. Verwendet man ausschließlich die korrekt alignierbaren Regionen zur Bewertung der Protein-Alignments, so lässt sich zeigen, dass die Leistung der Alignment-Programme mit steigender Sequenz-Länge ebenfalls steigt. Die ansonsten generell schlechtere Leistung lokaler Alignment-Programme gleicht sich mit steigender Sequenz-Länge denen der globalen Alignment-Programme an. In den hier durchgeführten Tests war es schon allein aufgrund der hohen Zahl an Alignments nicht möglich die erwähnten „Core Blocks“ zu definieren. Eine Untersuchung des Einflusses der Sequenz-Länge fand nicht statt.

Lassmann & Sonnhammer (2002) führten einen Test der Programme CLUSTALW, DIALIGN, POA und T-COFFEE anhand von BAliBASE Test-Sets sowie eigens mit Hilfe von ROSE konstruierten Alignments durch. Auch hier wurde festgestellt, dass die Sequenz-Länge einen positiven Einfluss auf die Leistung der Programme hat. T-COFFEE zeigte dort von allen Programmen bei Alignments mit (nach ROSE) geringer evolutionärer Distanz die beste Leistung (siehe auch Notredame *et al.*, 2000), wohingegen DIALIGN bei hoher evolutionärer Distanz die besten Alignments erzeugte. Beides lässt sich nicht auf RNA-Alignments übertragen. Hier ist es vielmehr so, dass die Leistungsunterschiede der Programme ab einer Sequenz-Identität von 70% APSI extrem gering sind und die Leistung von DIALIGN genauso wie die von ALIGN-M, als weiterem lokal und Konsistenz-basiert arbeitendem Programm, vergleichsweise schlecht sind.

4.9 Schlussfolgerungen

Die Leistung aller Alignment-Programme angewendet auf ncRNA-Sequenzen hängt genauso wie bei Proteinen grundsätzlich von der Anzahl der Sequenzen und der Sequenz-Homologie ab. Über einer Sequenz-Homologie von etwa 75% APSI ist die Leistung der Programme nahezu identisch. Erst darunter werden die Unterschiede der verschiedenen Methoden deutlich. Die sogenannte „Twilight Zone“, also der Homologie-Bereich unterhalb dessen die Qualität der Alignments dramatisch fällt, liegt für RNAs bei 55% Sequenz-Homologie, und damit deutlich höher als bei Proteinen (ca. 20%).

Grundsätzlich zeigen Sequenz-Alignment-Programme, die Segment-basierte Ansätze (beispielsweise DIALIGN) oder lokale Alignment-Methoden (beispielsweise ALIGN-M) als Grundlage haben, eine sehr schlechte Leistung. Grund hierfür könnten die durch die Basenpaarungen induzierten Fernbeziehungen in ncRNA-Sequenzen sein. Hoch-homologe Bereiche finden sich hier zumeist nur in ungepaarten Bereichen. Diese können zwar theoretisch durch die genannten Ansätze aligniert werden, jedoch scheinen die benachbarten gepaarten Bereiche hierbei misaligniert zu werden.

Echte Struktur-Alignment-Programme lassen sich aufgrund der enormen Komplexität zur Zeit nur für das paarweise Alignment verwenden. Unter Ihnen ist FOLDALIGN das Mittel der Wahl. Jedoch ist auch hier der Einsatz erst unterhalb einer Sequenz-Homologie von ca. 55% APSI sinnvoll.

Durch Veröffentlichung des ersten Alignment-Benchmarks für ncRNAs (siehe Gardner *et al.*, 2005, und Abschnitt 3.6) war es erstmals möglich, systematisch Programm-Parameter an das RNA-Alignment-Problem anzupassen. Alle hiermit optimierten Programme (MAFFT, MUSCLE und STRAL) zeigten in dem zweiten Benchmark (siehe Abschnitt 3.7) die signifikant beste Leistung. Das Programm STRAL, welches einen Hybrid-Ansatz aus Struktur- und Sequenz-Alignment implementiert und trotzdem äußerst schnell arbeitet, stellt unter diesen Programmen eine Ausnahme dar, da es als einziges keinen iterativen Ansatz verfolgt. Trotzdem erzeugt es im Bereich mittlerer Sequenz-Homologie (55%–75% APSI) bereits die besten Alignments, auch wenn der Leistungsunterschied zu den anderen Programmen nur sehr gering ist. Je divergenter die Sequenzen und je höher die Sequenzzahl, umso deutlicher werden generell die Leistungsvorteile der iterativ arbeitenden Programme. Dabei spielt es keine Rolle, ob diese Programme das Initial-Alignment mit schnellen, approximativen Methoden erstellen, wie beispielsweise MAFFT und MUSCLE. Allgemein zeigt MAFFT mit der Option ginsi die signifikant beste Leistung und kann als ein für das RNA-Alignment universell geeignetes Programm bezeichnet werden.

5

Zusammenfassung

Alignments nicht-Protein-kodierender RNAs (ncRNAs) haben ein weites Spektrum von Anwendungen: sie werden für Phylogenie-Vorhersagen (z. B. Olsen & Woese, 1993), Konsensusstruktur-Vorhersagen (z. B. Knudsen & Hein, 2003), sowie für Homologiesuchen in Datenbanken und zur Suche nach neuen ncRNAs eingesetzt (z. B. Eddy, 2002). Dabei hat die Qualität des eingesetzten Alignments entscheidenden Einfluss auf den Erfolg dieser Methoden. Gleichzeitig ist das korrekte Alignment von ncRNAs u. a. deshalb besonders schwierig, da sie in basengepaarten Bereichen durch kompensatorische Basenpaaraustausche – wechselseitige Mutationen, welche die Basenpaarung erhalten, aber die Sequenz-Homologie zerstören – evolvieren. Zwar existiert ein Algorithmus für die simultane Lösung von Strukturvorhersage und Alignment (Sankoff, 1985), jedoch ist dieser praktisch nicht einsetzbar, da seine Laufzeit und sein Speicherbedarf exponentiell von der Anzahl der Sequenzen abhängig sind. Selbst vereinfachende Implementationen dieses Algorithmus sind aufgrund ihrer Komplexität auf das paarweise Alignment beschränkt, so dass auch für das Alignment von ncRNAs Sequenz-Alignment-Programme eingesetzt werden.

In dieser Arbeit sollte ein Benchmark von Alignment-Programmen angewendet auf ncRNAs durchgeführt werden. Dieser Benchmark inkl. der zugehörigen Datenbank können als RNA-Pendant der Protein-spezifischen BAliBASE (Thompson *et al.*, 2005) verstanden werden. Um einen solchen Benchmark zu ermöglichen, mussten zunächst entsprechende Bewertungsmaße entwickelt werden, welche die Eigenschaften eines RNA-Alignments auf Sequenz- und Struktur-Ebene abbilden können. Hier wurden die sich ideal ergänzenden Maße SCI („Structure Conservation Index“; Washietl *et al.*, 2005) und SPS („Sum-of-Pairs-Score“; Thompson *et al.*, 1999a) eingesetzt. Weiterhin mussten Test-Sets mit jeweils vorhandener „richtiger“ Lösung konstruiert werden, die in ihren Eigenschaften (Sequenz-Anzahl und Sequenz-Homologie) gezielt variieren, um so den Einfluss der Eigenschaften auf die Programme quantitativ bestimmen zu können. Die zunächst angedachte Vorgehensweise, diese mit Hilfe des Programms CONSTRUCT (Lück *et al.*, 1999) zu erstellen, musste aus Zeitgründen verworfen werden. Stattdessen wurden auf zwei verschiedene Arten Test-Sets aus großen, vertrauenswürdigen Alignments der Rfam-Datenbank („RNA family Database“; Griffiths-Jones *et al.*, 2005) konstru-

iert. In Kooperation entstand der erste systematische Benchmark von Alignment-Programmen angewendet auf ncRNA-Sequenzen (Gardner *et al.*, 2005). Anhand dessen wurde es erstmals möglich, Programm-Parameter für das RNA-Alignment-Problem zu optimieren, wie beispielsweise für die Programme MAFFT, MUSCLE und STRAL geschehen. Dieser Benchmark wurde durch einen zweiten Test komplementiert, der aktuelle Programmversionen, verbesserte Test-Sets und statistische Rangtests beinhaltete. Mit den beiden Daten-Sätzen und dem zur Verfügung stehenden Bewertungssystem war ein objektiver Vergleich und eine Evaluation von Alignment-Programmen möglich.

Es konnte u. a. gezeigt werden, dass die sogenannte „Twilight Zone“, der Homologie-Bereich unterhalb dessen die Qualität der Alignments dramatisch fällt, für RNAs bei 55% Sequenz-Homologie, statt wie bei Proteinen bei 20% liegt und oberhalb von etwa 75% Sequenz-Homologie die Leistung aller Programme nahezu gleich gut ist. Weiterhin ergab sich, dass iterative Alignment-Methoden insbesondere bei divergenten Sequenzen und bei steigender Sequenz-Zahl im Vergleich zu nicht-iterativen Methoden deutlich die besten Alignments erzeugen. Das Programm MAFFT (Katoh *et al.*, 2005) zeigt mit der Option „ginsi“ statistisch signifikant die beste Leistung von allen hier getesteten Programmen.

Literaturverzeichnis

Alberts, Bruce (1998). *Essential Cell Biology.* Garland Publishing, Inc., New York.

Allain, F.H. & Varani, G. (1995). Structure of the P1 helix from group I self-splicing introns. *J. Mol. Biol.*, **250**(3), 333–353.

Antal, M., Mougin, A., Kis, M., Boros, E., Steger, G., Jakab, G., Solymosy, F. & Branlant, C. (2000). Molecular characterization at the RNA and gene levels of U3 snoRNA from a unicellular green alga, Chlamydomonas reinhardtii. *Nucl. Acids Res.*, **28**(15), 2959–2968.

Bahr, Anne, Thompson, Julie D., Thierry, J.-C. & Poch, Olivier (2001). BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucl. Acids Res.*, **29**(1), 323–326.

Baltimore, D. (1970). Viral RNA-dependent DNA polymerase. *Nature*, **226**, 1209–1211.

Ban, N., Nissen, P., Hansen, J., Moore, P.B. & Steitz, T.A. (2000). The Complete Atomic Structure of the Large Ribosomal Subunit at 2.4 A Resolution. *Science*, **289**(5481), 905–920.

Bauer, Markus, Klau, Gunnar W. & Reinert, Knut (2005). Fast and Accurate Structural RNA Alignment by Progressive Lagrangian Relaxation. In *Lecture Notes in Bioinformatics* (Berthold, M. R. et al., Hrsg.), volume **3695** of *In Proc. of CompLife 2005, First International Symposium on Computational Life Science, Konstanz, Germany.* Springer, Berlin, S. 217–228.

Berman, H. M., Olson, W. K., Beveridge, D. L., Westbrook, J., Gelbin, A., Demeny, T., Hsieh, S.-H., Srinivasan, A. R. & Schneider, B. (1992). The Nucleic Acid Database: A Comprehensive Relational Database of Three-Dimensional Structures of Nucleic Acids. *Biophys. J.*, **63**, 751–759.

Berman, Helen M., Westbrook, John, Feng, Zukang, Gilliland, Gary, Bhat, T. N., Weissig, Helge, Shindyalov, Ilya N. & Bourne, Philip E. (2000). The Protein Data Bank. *Nucl. Acids Res.*, **28**(1), 235–242.

Bonhoeffer, S., McCaskill, J.S., Stadler, P.F. & Schuster, P. (1993). RNA multi-structure landscapes. A study based on temperature dependent partition functions. *Eur. Biophys. J.*, **22**, 13–24.

Chenna, Ramu, Sugawara, Hideaki, Koike, Tadashi, Lopez, Rodrigo, Gibson, Toby J., Higgins, Desmond G. & Thompson, Julie D. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucl. Acids Res.*, **31**(13), 3497–3500.

Chiu, D.K. & Kolodziejczak, T. (1991). Inferring consensus structure from nucleic acid sequences. *Comp. Appl. Biosci.*, **7**, 347–352.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.*, **74**, 829–836.

Cleveland, W. S. (1981). LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician*, **35**, 54.

Cline, Melissa, Hughey, Richard & Karplus, Kevin (2002). Predicting reliable regions in protein sequence alignments. *Bioinformatics*, **18**(2), 306–314.

Cornish-Bowden, A. (1985). Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucl. Acids Res.*, **13**, 3021–30.

Couzin, Jennifer (2002). Breakthrough Of The Year: Small RNAs Make Big Splash. *Science*, **298**(5602), 2296–2297.

Crick, F. (1958). On protein synthesis. *Symp. Soc. Exp. Biol.*, **12**, 138–163.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, **227**, 561–563.

Dalli, Deniz (2006). Multiples RNA-Sequenz-Struktur-Alignment. Diplomarbeit, Heinrich Heine-Universität Düsseldorf.

Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., Hrsg.), volume **5**. Natl. Biomed. Res Found., Washington, DC., S. 345–352.

Desmet, Johan, Spriet, Jan & Lasters, Ignace (2002). Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins: Structure, Function, and Genetics*, **48**(1), 31–43.

Doolittle, R.F. (1981). 5 S ribosomal RNA genes and the AluI family: Evolutionary and functional significance of a region of strong homology. *FEBS Letters*, **126**(2), 147–149.

Doshi, Kishore, Cannone, Jamie, Cobaugh, Christian & Gutell, Robin (2004). Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, **5**(1), 105.

Dowell, Robin & Eddy, Sean R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, **5**(1), 71.

Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (1998). *Biological sequence analysis.* Cambridge University Press, Cambridge.

Eddy, Sean R. (2002). A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, **3**(1), 18.

Eddy, Sean R. (2004). How do RNA folding algorithms work? *Nature Biotechnology*, **22**(11), 1457–1458.

Eddy, Sean R. (2004). What is a hidden Markov model? *Nature Biotechnology*, **22**, 1315 – 1316.

Eddy, Sean R. (2004). What is dynamic programming? *Nature Biotechnology*, **22**, 909–910.

Eddy, Sean R. (2004). Where did the BLOSUM62 alignment score matrix come from? *Nature Biotechnology*, **22**(8), 1035–1036.

Eddy, Sean R. (2005). SQUID - C function library for sequence analysis.

Edgar, Robert (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**(1), 113.

Edgar, Robert C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.*, **32**(5), 1792–1797.

Fagegaltier, D., Lescure, A., Walczak, R., Carbon, P. & Krol, A. (2000). Structural analysis of new local features in SECIS RNA hairpins. *Nucl. Acids Res.*, **28**, 2679–2689.

Feng, D.F. & Doolittle, R.F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.

Fischer, D., Barret, C., Bryson, K., Elofsson, A., Godzik, A., Jones, D., Karplus, K.J., Kelley, L.A., MacCallum, R.M., Pawowski, K., Rost, B., Rychlewski, L. & Sternberg, M. (1999). CAFASP-1: critical assessment of fully automated structure prediction methods. *Proteins*, **Suppl 3**, 209–217.

Gardner, Paul & Giegerich, Robert (2004). A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, **5**(1), 140.

Gardner, Paul P., Wilm, Andreas & Washietl, Stefan (2005). A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucl. Acids Res.*, **33**(8), 2433–2439.

Gendrona, P., Lemieuxa, S. & Major, F. (2001). Quantitative analysis of nucleic acid three-dimensional structures. *Journal of Molecular Biology*, **308**(5), 919–936.

Gilbert, Walter (1986). Origin of life: The RNA world. *Nature*, **319**, 618.

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

Gotoh, Osamu (1996). Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinement as Assessed by Reference to Structural Alignments. *J. Mol. Biol.*, **264**, 823–838.

Gotoh, O. (1999). Multiple sequence alignment: algorithms and applications. *Adv. Biophys.*, **36**, 159–206.

Gräf, S., Przybilski, R., Steger, G. & Hammann, C. (2005). A database search for hammerhead ribozyme motifs. *Biochem. Soc. Trans.*, **33**(Pt 3), 477–478.

Gräf, Stefan, Strothmann, Dirk, Kurtz, Stefan & Steger, Gerhard (2001). HyPaLib: a database of RNAs and RNA structural elements defined by hybrid patterns. *Nucl. Acids Res.*, **29**(1), 196–198.

Gräf, S., Teune, J.-H., Strothmann, D., Kurtz, S. & Steger, G. (2006). A computational approach to search for non-coding RNAs in large genomic data. In *Small RNAs: Analysis and Regulatory Functions.* (Nellen, W. & Hammann, C., Hrsg.), volume **17** of *Nucleic Acids and Molecular Biology*, S. 57–74. Springer Verlag.

Griffiths-Jones, Sam (2005). RALEE–RNA ALignment Editor in Emacs. *Bioinformatics*, **21**(2), 257–259.

Griffiths-Jones, Sam, Bateman, Alex, Marshall, Mhairi, Khanna, Ajay & Eddy, Sean R. (2003). Rfam: an RNA family database. *Nucl. Acids Res.*, **31**(1), 439–441.

Griffiths-Jones, Sam, Moxon, Simon, Marshall, Mhairi, Khanna, Ajay, Eddy, Sean R. & Bateman, Alex (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucl. Acids Res.*, **33**(suppl_1), D121–124.

Gusfield, D. (1999). *Algorithms on strings, trees, and sequences. Computer science and computational biology.* Cambridge University Press, Cambridge.

Gutell, R.R., Lee, J.C. & Cannone, J.J. (2002). The accuracy of ribosomal RNA comparative structure models. *Curr. Opin. Struct. Biol.*, **12**(3), 301–310.

Havgaard, Jakob H., Lyngso, Rune B. & Gorodkin, Jan (2005). The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search. *Nucl. Acids Res.*, **33**(suppl_2), W650–653.

Havgaard, Jakob Hull, Lyngso, Rune B., Stormo, Gary D. & Gorodkin, Jan (2005). Pairwise local structural alignment of RNA sequences with sequence similarity less than 40 %. *Bioinformatics*, **21**(9), 1815–1824.

Helm, M., Brule, H., Friede, D., Giege, R., Putz, D. & Florentz, C. (2000). Search for characteristic structural features of mammalian mitochondrial tRNAs. *RNA*, **6**(10), 1356–1379.

Henikoff, Jorja G. & Henikoff, Steven (1992). Amino acid substitution matrices from protein blocks. *Proc. Nat. Acad. Sci. U.S.A.*, **89**(22), 10915–10919.

Higgins, D. G., Blackshields, G. & Wallace, I. M. (2005). Mind the gaps: Progress in progressive alignment. *PNAS*, **102**(30), 10411–10412.

Hofacker, I.L., Fekete, M. & Stadler, P.F. (2002). Secondary Structure Prediction for Aligned RNA Sequences. *Journal of Molecular Biology*, **319**(5), 1059–1066.

Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, S., Tacker, M & Schuster, P. (1994). Fast folding and comparsion of RNA structures. *Monatsh. Chem.*, **125**, 167–188.

Hofacker, Ivo L. (2003). Vienna RNA secondary structure server. *Nucl. Acids Res.*, **31**(13), 3429–3431.

Hofacker, Ivo L., Bernhart, Stephan H. F. & Stadler, Peter F. (2004). Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**(14), 2222–2227.

Holmes, Ian (2003). Using guide trees to construct multiple-sequence evolutionary HMMs. *Bioinformatics*, **19**(suppl_1), 147i–157.

Holmes, I (2004). A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*, **5**(166).

Holmes, I (2005). Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, **6**(1), 73.

Holmes, Ian & Bruno, William J. (2001). Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, **17**(9), 803–820.

Hudelot, C., Gowri-Shankar, V., Jow, H., Rattray, M. & Higgs, PG. (2003). RNA-based phylogenetic methods: application to mammalian mitochondrial RNA sequences. *Mol. Phyl. Evol.*, **28**(2), 241–252.

Jeon, Yoon-Seong, Chung, Hwanwon, Park, Sunyoung, Hur, Inae, Lee, Jae-Hak & Chun, Jongsik (2005). jPHYDIT: a JAVA-based integrated environment for molecular phylogeny of ribosomal RNA sequences. *Bioinformatics*, **21**(14), 3171–3173.

Ji, Yongmei, Xu, Xing & Stormo, Gary D. (2004). A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, **20**(10), 1591–1602.

Juan, V. & Wilson, C. (1999). RNA secondary structure prediction based on free energy and phylogenetic analysis. *J. Mol. Biol.*, **289**(4), 935–947.

Karplus, Kevin & Hu, Birong (2001). Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics*, **17**(8), 713–720.

Katoh, Kazutaka, Kuma, Kei-ichi, Toh, Hiroyuki & Miyata, Takashi (2005). MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucl. Acids Res.*, **33**(2), 511–518.

Katoh, Kazutaka, Misawa, Kazuharu, Kuma, Kei-ichi & Miyata, Takashi (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucl. Acids Res.*, **30**(14), 3059–3066.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**(11), 111–120.

Klein, Robert & Eddy, Sean R. (2003). RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinformatics*, **4**(1), 44.

Kleywegt, G.J. & Jones, T.A. (1995). Where freedom is given, liberties are taken. *Structure*, **3**(6), 535–540.

Knudsen, B., Andersen, E.S., Damgaard, C., Kjems, J. & Gorodkin, J. (2004). Evolutionary rate variation and RNA secondary structure prediction. *Computational Biology and Chemistry*, **28**(3), 219–226.

Knudsen, Bjarne & Hein, Jotun (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucl. Acids Res.*, **31**(13), 3423–3428.

Kryukov, Gregory V. & Gladyshev, Vadim N. (2004). The prokaryotic selenoproteome. *EMBO reports*, **5**(5), 538–543.

Lassmann, Timo & Sonnhammer, Erik L.L. (2002). Quality assessment of multiple alignment programs. *FEBS Letters*, **529**(1), 126–130.

Lee, Christopher, Grasso, Catherine & Sharlow, Mark F. (2002). Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**(3), 452–464.

Lescure, A., Gautheret, D., Fagegaltier, D., Carbon, P. & Krol, A. (2000). From RNA Structure to the Identification of New Genes: The Example of Selenoproteins. *Journal of Health Science*, **46**, 405–408.

Letunic, Ivica, Copley, Richard R., Schmidt, Steffen, Ciccarelli, Francesca D., Doerks, Tobias, Schultz, Jorg, Ponting, Chris P. & Bork, Peer (2004). SMART 4.0: towards genomic data integration. *Nucl. Acids Res.*, **32**(suppl_1), D142–144.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, **10**(8), 707–710, Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848 (1965).

Li, W.H., Wu, C.I. & Luo, C.C. (1985). A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. *Mol Biol Evol*, **2**(2), 150–174.

Löytynoja, Ari & Goldman, Nick (2005). From The Cover: An algorithm for progressive multiple alignment of sequences with insertions. *PNAS*, **102**(30), 10557–10562.

Löytynoja, Ari & Milinkovitch, Michel C. (2003). A hidden Markov model for progressive multiple alignment. *Bioinformatics*, **19**(12), 1505–1513.

Lück, R.H. (1997). Thermodynamische Vorhersage konservierter Strukturelemente in einzelsträngiger RNA. Doktorarbeit, Heinrich Heine-Universität Düsseldorf.

Lück, R., Gräf, S. & Steger, G. (1999). ConStruct: a tool for thermodynamic controlled prediction of conserved secondary structure. *Nucl. Acids Res.*, **27**(21), 4208–4217.

Lück, R., Steger, G. & Riesner, D. (1996). Thermodynamic prediction of conserved secondary structure: Application to RRE-element of HIV, tRNA-like element of CMV, and mRNA of prion protein. *J. Mol. Biol.*, **258**, 813–826.

Mainz, Indra (2006). Statistik von RNA-Struktur-Alignments. Diplomarbeit, Heinrich-Heine-Universität Düsseldorf.

Mandal, M., Boese, B., Barrick, J.E., Winkler, W.C. & Breaker, R.R. (2003). Riboswitches control fundamental biochemical pathways in Bacillus subtilis and other bacteria. *Cell*, **113**(5), 577–586.

Mandal, M. & Breaker, R.R. (2004). Gene regulation by riboswitches. *Nature Reviews Molecular Cell Biology*, **5**(6), 451–463.

Martin, L. C., Gloor, G. B., Dunn, S. D. & Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, **21**(22), 4116–4124.

Mathews, D.H., Sabina, J., Zuker, M. & Turner, D.H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

Mathews, D.H. & Turner, D.H. (2002). Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, **317**(2), 191–203.

Mathews, David H. (2005). Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics*, **21**(10), 2246–2253.

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

McClure, MA, Vasi, TK & Fitch, WM (1994). Comparative analysis of multiple protein-sequence alignment methods [published erratum appears in Mol Biol Evol 1994 Sep;11(5):811]. *Mol Biol Evol*, **11**.

Miller, W. & Myers, E. (1988). Optimal alignments in linear space. *Comput. Applic. Biosci.*, **4**, 11–17.

Mizuguchi, K., Deane, C. M., Blundell, T. L. & Overington, J. P. (1998). HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Sci*, **7**(11), 2469–2471.

Morgenstern, B (1999). DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**(3), 211–218.

Morgenstern, Burkhard (2004). DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. *Nucl. Acids Res.*, **32**(suppl_2), W33–36.

Needleman, S.B. & Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Nissen, Poul, Hansen, Jeffrey, Ban, Nenad, Moore, Peter B. & Steitz, Thomas A. (2000). The Structural Basis of Ribosome Activity in Peptide Bond Synthesis. *Science*, **289**(5481), 920–930.

Notredame, C. (2002). Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, **3**, 131–44.

Notredame, C., Higgins, D.G. & J., Heringa (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.

Notredame, C., Holm, L. & Higgins, D.G. (1998). COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*, **14**(5), 407–422.

Nussinov, R., Pieczenik, G., Griggs, J.R. & Kleitman, D.J. (1978). Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82.

Olsen, GJ & Woese, CR (1993). Ribosomal RNA: a key to phylogeny. *FASEB J.*, **7**(1), 113–123.

O'Sullivan, Orla, Zehnder, Mark, Higgins, Des, Bucher, Philipp, Grosdidier, Aurelien & Notredame, Cedric (2003). APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, **19**(suppl_1), i215–221.

Owens, R. A., Thompson, S. M. & Kramer, M. (2003). Identification of neutral mutants surrounding two naturally occurring variants of Potato spindle tuber viroid. *J Gen Virol*, **84**(3), 751–756.

Pace, N.R., Thomas, B.C. & Woese, C.R. (1999). Probing rna structure, function, and history by comparative analysis. In *The RNA World* (Gesteland, R.F., Cech, T.R. & Atkins, J.F., Hrsg.), S. 113–141. Cold Spring Harbor Laboratory Press, New York.

Pei, Jimin, Sadreyev, Ruslan & Grishin, Nick V. (2003). PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, **19**(3), 427–428.

Pollard, DA, Bergman, CM, Stoye, J, Celniker, SE & Eisen, MB (2004). Benchmarking tools for the alignment of functional noncoding DNA. *BMC Bioinformatics*, **5**, 6.

Precht, Manfred & Kraft, Roland (1993). *Bio-Statistik 2*. R. Oldenbourg Verlag, Wien.

Raghava, GPS, Searle, Stephen, Audley, Patrick, Barber, Jonathan & Barton, Geoffrey (2003). OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, **4**(1), 47.

Riks, Jochen (2001). Vorhersage konservierter Strukturelemente in einzelsträngiger Ribonukleinsäure. Diplomarbeit, Heinrich-Heine-Universität Düsseldorf.

Rosenblad, Magnus Alm, Gorodkin, Jan, Knudsen, Bjarne, Zwieb, Christian & Samuelsson, Tore (2003). SRPDB: Signal Recognition Particle Database. *Nucl. Acids Res.*, **31**(1), 363–364.

Ruan, Jianhua, Stormo, Gary D. & Zhang, Weixiong (2004). An Iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics*, **20**(1), 58–66.

Sachs, Lothar (2004). *Angewandte Statistik*. Springer, Berlin.

Saitou, N. & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Sauder, J. Michael, W. Arthur, Jonathan & Dunbrack, Roland L. Jr. (2000). Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins: Structure, Function, and Genetics*, **40**(1), 6–22.

Sayle, Roger A. & Milner-White, E. James (1995). RASMOL: biomolecular graphics for all. *Trends in Biochemical Sciences*, **20**(9), 374–376.

Schmitz, M. & Steger, G. (1992). Base-pair probability profiles of RNA secondary structures. *Comp. Appl. Biosci.*, **8**, 389–399.

Smith, T.F. & Waterman, M.S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Soukup, G.A. & Breaker, R, R. (1999). Relationship between internucleotide linkage geometry and the stability of RNA. *RNA*, **5**(10), 1308–1325.

Sprinzl, Mathias & Vassilenko, Konstantin S. (2005). Compilation of tRNA sequences and sequences of tRNA genes. *Nucl. Acids Res.*, **33**(suppl_1), D139–140.

Steger, Gerhard (2003). *Bioinformatik: Methoden zur Vorhersage von RNA- und Proteinstruktur.* Birkhäuser Verlag, Basel.

Steger, G., Hofmann, H., Förtsch, J., Gross, H.J., Randles, J.W., Sänger, H.L. & Riesner, D. (1984). Conformational transitions in viroids and virusoids: Comparison of results from energy minimization algorithm and from experimental data. *J. Biomol. Struct. Dyn.*, **2**(3), 543–571.

Stoye, J. (1998). Multiple sequence alignment with the Divide-and-Conquer method. *Gene*, **211**(2), GC45–45.

Stoye, J, Evers, D & Meyer, F (1998). Rose: generating sequence families. *Bioinformatics*, **14**(2), 157–163.

Subramanian, Amarendran, Weyer-Menkhoff, Jan, Kaufmann, Michael & Morgenstern, Burkhard (2005). DIALIGN-T: An improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, **6**(1), 66.

Szymanski, Maciej, Barciszewska, Miroslawa Z., Erdmann, Volker A. & Barciszewski, Jan (2002). 5S Ribosomal RNA Database. *Nucl. Acids Res.*, **30**(1), 176–178.

Tabaska, J.E., Cary, R.B., Gabow, H.N. & Stormo, G.D. (1998). An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, **14**, 691–699.

Temin, H. M. (1970). RNA-dependent DNA polymerase in virions of Rous sarcoma virus. *Nature*, **226**, 1211–1213.

Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F. & Higgins, D.G. (1997). The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucl. Acids. Res.*, **25**(24), 4876–4882.

Thompson, J.D., Higgins, D.G. & Gibson, T.J. (1994). CLUSTAL W: 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.

Thompson, J.D., Plewniak, F. & Poch, O. (1999). A comprehensive comparison of multiple sequence alignment programs. *Nucl. Acids Res.*, **27**(13), 2682–2690.

Thompson, J.D., Plewniak, F. & Poch, O. (1999). BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**(1), 87–88.

Thompson, Julie D., Koehl, Patrice, Ripp, Raymond & Poch, Olivier (2005). BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, **61**(1), 127–136.

Tinoco, Jr, I. & Bustamante, C. (1999). How RNA folds. *J. Mol. Biol.*, **293**, 271–281.

Tinoco, Jr, I., Uhlenbeck, O.C. & Levine, M.D. (1971). Estimation of secondary structure in ribonucleic acids. *Nature*, **230**, 362–367.

Touzet, Helene & Perriquet, Olivier (2004). CARNAC: folding families of related RNAs. *Nucl. Acids Res.*, **32**(suppl_2), W142–145.

Tucker, B.J. & Breaker, R.R. (2005). Riboswitches as versatile gene control elements. *Curr. Opin. Struct. Biol.*, **15**(3), 342–348.

Van de Peer, Y., Van den Broeck, I., De Rijk, P. & De Wachter, R. (1994). Database on the structure of small ribosomal subunit RNA. *Nucl. Acids Res.*, **22**(17), 3488–3494.

Van Walle, Ivo, Laster, Ignace & Wyns, Lode (2003). Consistency matrices: Quantified structure alignments for sets of related proteins. *Proteins: Structure, Function, and Genetics*, **51**(1), 1–9.

Van Walle, Ivo, Lasters, Ignace & Wyns, Lode (2004). Align-m–a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, **20**(9), 1428–1435.

Van Walle, Ivo, Lasters, Ignace & Wyns, Lode (2005). SABmark–a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, **21**(7), 1267–1268.

Vogel, Jørg & Sharma, Cynthia Mira (2005). How to find small non-coding RNAs in bacteria. *Biological Chemistry*, **386**(12), 1219–1238.

Vogt, Gerhard, Etzold, Thure & Argos, Patrick (1995). An Assessment of Amino Acid Exchange Matrices in Aligning Protein Sequences: The Twilight Zone Revisited. *J. Mol. Biol.*, **249**(4), 816–831.

Wallace, Iain M., Orla, O'Sullivan & Higgins, Desmond G. (2005). Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics*, **21**(8), 1408–1414.

Washietl, Stefan & Hofacker, Ivo L. (2004). Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, **342**(1), 19–30.

Washietl, Stefan, Hofacker, Ivo L. & Stadler, Peter F. (2005). Fast and reliable prediction of noncoding RNAs. *Proc. Nat. Acad. Sci. U.S.A.*, **102**(7), 2454–2459.

Waugh, A., Gendron, P., Altman, R., Brown, J. W., Case, D., Gautheret, D., Harvey, S. C., Leontis, N., Westbrook, J., Westhof, E., Zuker, M. & Major, F. (2002). RNAML: a standard syntax for exchanging RNA information. *RNA*, **8**(6), 707–717.

Wilm, Andreas (2002). Optimierung von Alignments und Konsensus-Struktur-Vorhersagen für RNA. Diplomarbeit, Heinrich Heine-Universität Düsseldorf.

Winkler, W.C, Nahvi, A., Roth, A., Collins, J.A. & Breaker, R.R. (2004). Control of gene expression by a natural metabolite-responsive ribozyme. *Nature*, **18**(428), 281–286.

Wolf, Matthias, Achtziger, Marco, Schultz, Jörg, Dandekar, Thomas & Müller, Tobias (2005). Homology modeling revealed more than 20,000 rRNA internal transcribed spacer 2 (ITS2) secondary structures. *RNA*, **11**(11), 1616–1623.

Wolf, M., Friedrich, J., Dandekar, T. & Müller, T. (2005). CBCAnalyzer: inferring phylogenies based on compensatory base changes in RNA secondary structures. *In Silico Biology*, **5**(3), 291–294.

Yang, Huanwang, Jossinet, Fabrice, Leontis, Neocles, Chen, Li, Westbrook, John, Berman, Helen & Westhof, Eric (2003). Tools for the automatic identification and classification of RNA base pairs. *Nucl. Acids Res.*, **31**(13), 3450–3460.

Yang, Q. & Blanchette, M. (2004). StructMiner: A tool for alignment and detection of conserved secondary structure. *Genome Informatics*, (15), 102–111.

Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science*, **244**(244), 48–52.

Zuker, M. (2000). Calculating nucleic acid secondary structure. *Curr. Opin. Struct. Biol.*, **10**, 303–310.

Zuker, Michael (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucl. Acids Res.*, **31**(13), 3406–3415.

Zuker, M. & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucl. Acids Res.*, **9**, 133–148.

Appendix

A SQUICL Kommandoreferenz

Im Folgenden ist eine kurze Kommandoreferenz für die während dieser Arbeit entstandene Tcl-Bibliothek SQUICL (Version 0.3.0) aufgeführt (siehe auch Abschnitt 3.2).

Das meistgenutzte Element dieser Bibliothek, im Folgenden *seq-handle* oder Sequenz-Handle genannt, ist eine Art Zeiger auf einen Datentyp, der sowohl Sequenzdateien als auch Alignments in Form einer „Union“ speichert.

Tabelle 5.1: Kommando-Referenz SQUICL 0.3.0.

Namespace `squicl`

`squicl::Dealign` *`seq`*
Entfernt alle Gaps aus der übergebenen Sequenz *seq* und gibt den entstehenden String zurück.

`squicl::DumpVienna` *`seqhandle ?fname?`*
Schreibt die Inhalt des Sequenz-Handles *seqhandle* in der sogenannten Vienna-Notation in eine Datei (*fname*) oder auf die Standard-Ausgabe.

`squicl::FileIsMsa` *`filename`*
Bestimmt, ob die Datei *filename* ein multiples Sequenz-Alignment ist.

`squicl::IsGap` *`nt`*
Bestimmt, ob das übergebene Nukleotid *nt* ein Gap ist.

`squicl::PairwiseIdentity` *`seq1 seq2`*
Bestimmt die paarweise Identität zwischen den zwei übergebenen Sequenzen *seq1* und *seq2*. Diese ist gegeben als der Bruch aus Anzahl der Übereinstimmungen und Länge der kürzeren Sequenz. Siehe auch Abschnitt 3.5.3.

`squicl::ToDna` *`seq`*
Konvertiert die Sequenz *seq* in DNA.

`squicl::ToIupac` *`seq`*
Konvertiert die Sequenz *seq* IUPAC-konform.

`squicl::ToRna` *`seq`*
Konvertiert die Sequenz *seq* in RNA.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Namespace `squicl::rnaalifold`

`squicl::rnaalifold::Init` *?rnafold_override_path?*
Initialisiert den Namespace vor dem ersten Gebrauch.

`squicl::rnaalifold::ExecFold` *seqhandle* ?*indir*? ?*temperature*? ?*rnaalifold_extra_args*?
Führt RNAALIFOLD mit dem Alignment *seqhandle* als Input aus und gibt die Konsensus-Struktur, -Sequenz, -Energie sowie weitere Parameter als Array zurück. Für Details siehe squicl_rnaalifold.tcl.

`squicl::rnaalifold::Mfe` *seqhandle*
Berechnet die mittels RNAALIFOLD berechnete Konsensus-MFE und Konsensus-Struktur des Alignments *seqhandle* und gibt diese als Liste zurück.

Namespace `squicl::utils`

`squicl::utils::IdToFilename` *id*
Konvertiert die Sequenz-Id *id* in einen Dateinamen, der keinerlei Sonder- oder Steuer-Zeichen enthält.

`squicl::utils::MkTemp` *file_or_dir prefix* ?*in_dir*?
Erzeugt ein/e neue/s temporäre/s Datei/Verzeichnis (*file_or_dir*) mit Prefix *prefix*.

Namespace `squicl::msa`

`squicl::msa::Comp` *seqhandle1 seqhandle2*
Berechnet die gemittelte Identität über alle möglichen Sequenz-Paare der Alignments (*seqhandle1* und *seqhandle2*). Siehe auch `squicl::PairwiseIdentity`.

`squicl::msa::IupacConsensus` *seqhandle*
Bestimmt den Konsensus-String des Nukleinsäure-Alignments *seqhandle* und beherrscht im Gegensatz zu `squicl::msa::MajorityRuleConsensus` die Deutung von IUPAC-Code und den Umgang mit Gaps.

`squicl::msa::MajorityRuleConsensus` *seqhandle*
Berechnet die Konsensus-Sequenz des Alignments *seqhandle* durch simple Majoritätsregel, wobei im Gegensatz zu `squicl::msa::IupacConsensus` Spalten mit weniger als 50% Nukleotiden bzw. Aminosäuren ignoriert werden. Für Details siehe squicl_rnafold.tcl.

`squicl::msa::MinGap` *seqhandle*
Entfernt alle Spalten, welche nur aus Gaps bestehen, aus dem Alignment *seqhandle*.

`squicl::msa::MutualInfo ?-log_e? ?-unbiased?` *seqhandle*
Berechnet den Gegenseitigen Informationsgehalt des Alignments *seqhandle*.

`squicl::msa::Sci` *seqhandle*
Berechnet den SCI. Siehe auch Abschnitt 3.5.3.

`squicl::msa::SumOfPairs ?-incl_colcost?` *seqhandle*
Berechnet die Sum-of-Pairs-Cost des Alignments *seqhandle*.

`squicl::msa::PwIdent` *seqhandle*
Bestimmt die gemittelte paarweise Identität aller Sequenz-Paare des Alignments *seqhandle*. Siehe auch Abschnitt 3.5.3.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Namespace `squicl::seq`

`squicl::seq::Free` *`seqhandle`*
Gibt den von einem Sequenz-Handle belegten Speicher frei.

`squicl::seq::GetSeq` *`seqhandle seqnumber`*
Gibt Sequenz-Name/Id und Sequenz-String als Liste zurück.

`squicl::seq::HandleIsAln` *`seqhandle`*
Bestimmt, ob der Sequenz-Handle *seqhandle* ein Alignment ist.

`squicl::seq::Length` *`seqhandle ?seqnumber?`*
Gibt die Sequenzlänge der Sequenz mit Nummer *seqnumber* des Sequenz-Handles *seqhandle* bzw. die Alignment-Länge zurück.

`squicl::seq::ListFormats ?aligned?`
Gibt eine Liste unterstützter Sequenz-Datei-Formate aus.

`squicl::seq::NumSeq` *`seqhandle`*
Gibt die Anzahl Sequenzen des Sequenz-Handles *seqhandle* zurück.

`squicl::seq::Read ?-force_aln? ?force_nal?` *`seqfilename`*
Liest eine Sequenz-Datei (*seqfilename*) ein und gibt einen entsprechenden Sequenz-Handle zurück.

`squicl::seq::SetSeqName` *`seqhandle seqnumber newseqname`*
Überschreibt den Sequenzennamen der Sequenz mit Nummer *seqnumber* in *seqhandle* mit dem neuen Wert *newseqname*.

`squicl::seq::Sort` *`seqhandle`*
Sortiert die Sequenzen in *seqhandle* anhand der lexikographischen Ordnung über die enthaltenen (im Falle eines Alignments, dealignierten) Sequenz-Strings.

`squicl::seq::ToDna` *`seqhandle`*
Konvertiert alle Sequenzen in *seqhandle* in DNA.

`squicl::seq::ToCase` *`seqhandle case`*
Konvertiert alle Sequenzen in *seqhandle* in Groß- (*case*='u') bzw. Kleinschrift (*case*='l').

`squicl::seq::ToIupac` *`seqhandle`*
Konvertiert alle Sequenzen des Sequenz-Handles *seqhandle* IUPAC konform.

`squicl::seq::ToRna` *`seqhandle`*
Konvertiert alle Sequenzen in *seqhandle* in RNA.

`squicl::seq::ToStrippedDownIupac` *`seqhandle`*
Wandelt jeden Rest, der nicht im Alphabet ACGTUN enthalten ist, in N um.

`squicl::seq::Write` *`seqhandle format ?filename?`*
Schreibt die Sequenzen von *seqhandle* formatiert (*format*) in eine Datei (*filename*) oder auf die Standard-Ausgabe.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Namespace `squicl::rnafold`

`squicl::rnafold::Init` *?rnafold_override_path?*

Initialisiert den Namespace vor dem ersten Gebrauch.

`squicl::rnafold::EnergyOfStruct` *seq struc*

Berechnet die Energie der Sequenz *seq* und vorgegebener Struktur *struc*.

`squicl::rnafold::ExecFold` *rnaname rnaseq ?indir? ?temperature? ?rnafold_extra_args?*

Führt RNAFOLD mit der Sequenz namens *rnaname* und der Nukleotidabfolge *rnaseq* aus und gibt die MFE-Struktur und Pfade zu den entstehenden PostScript-Dateien als Array zurück. Für Details siehe aligneval.c (SQUID).

`squicl::rnafold::Mfe` *seq*

Berechnet die Struktur der Sequenz *seq* und gibt die MFE und die zugehörige Struktur als Punkt-Klammer-Notation in Form einer Liste zurück.

Glossar

APSI *Average Pairwise Sequence Identity* Die „durchschnittliche paarweise Sequenzidentität“ ist ein Maß der Sequenz-Homologie/-Konservierung innerhalb eines Alignments. Siehe Abschnitt 3.5.3. *Seite 20*

BAliBASE *Benchmark Alignment Database* Eine Protein-Alignment-Benchmark-Datenbank zur Evaluation von multiplen Protein-Alignment-Programmen. Siehe Bahr *et al.* (2001); Thompson *et al.* (1999b, 2005). *Seite 15*

Batch *Stapelverarbeitung* Sequentielle Abarbeitung mehrerer Einzeloperationen. *Seite 24*

Benchmark *Leistungsvergleichstest* Ein Benchmark ist ein Testverfahren, welches eine objektive Leistungsmessung erlaubt. *Seite 15*

BLOSUM *Blocks Substitution Matrix* Protein-spezifische Substitutionsmatrix, deren Werte anhand von lokalen multiplen Alignments der BLOCKS-Datenbank berechnet wurden. Siehe Eddy (2004d); Henikoff & Henikoff (1992). *Seite 10*

BRAliBase *Benchmark RNA Alignment Database* Das in dieser Arbeit vorgestellte RNA-Pendant zur BAliBASE. Der Begriff wurde mittlerweile von Paul Gardner adaptiert (siehe auch die BRAliBase-Homepage[1] geht meist schief wegen sonderzeichen). *Seite 15*

SPS′ Sequenz-Maß zur Berechnung der durchschnittlichen paarweisen Sequenz-Identität eines Alignments. Kann als SPS-Äquivalent bezeichnet werden. Siehe Abschnitt 3.5.2. *Seite 48*

CRE *Cis-Acting Replication Element* *Seite 64*

DNA *Deoxyribonucleic Acid* Auch deutsch: Desoxyribonukleinsäure (DNS). Makromolekül, das als Träger der genetischen Information dient. Siehe auch RNA. *Seite 1*

Frontend Ein Frontend ist die meist graphische Schnittstelle, die die benutzerfreundliche Bedienung eines Programms ermöglicht. *Seite 36*

GCC *GNU Compiler Collection* *Seite 20*

[1] http://www.binf.ku.dk/users/pgardner/bralibase/

GNU *GNU's Not UNIX* Das GNU Projekt[2] geht meist schief wegen sonderzeichen versucht ein freies UNIX-ähnliches Betriebssystem inkl. Betriebssystemkern (Kernel) und zugehöriger Programme zu entwickeln. *Seite 42*

GUI *Graphical User Interface* Grafische Benutzeroberfläche. *Seite 25*

HCV *Hepatitis C virus* HCV gehört zur Gruppe der *Flaviviridae* und hat ein einzelsträngiges RNA-Genom. *Seite 64*

HIV *Human Immunodeficiency Virus* HIV gehört zu den Retroviren und besitzt ein in doppelter Kopie vorliegendes einzelsträngiges RNA-Genom. *Seite 1*

HMM *Hidden-Markov-Model* Ein Hidden-Markov-Modell ist ein statistisches Modell, welches Zustände („states") und an sie gebundene Emissionswahrscheinlichkeiten und Übergangswahrscheinlichkeiten definiert. Bei Besuch eines Zustandes wird eine Sequenz in Abhängigkeit von der Emissionswahrscheinlichkeit generiert und in Abhängigkeit von der Übergangswahrscheinlichkeit in den nächsten Zustand gewechselt. Der Viterbi-Algorithmus findet hier Anwendung, um die Abfolge von Zuständen mit größter Wahrscheinlichkeit zu finden. Siehe Durbin *et al.* (1998) und Eddy (2004b). *Seite 27*

IRES *Internal Ribosome Entry Site* Interne Ribosomenbindungsstelle, die eine 5'-Cap unabhängige Proteinsynthese im eukaryotischen System ermöglicht. *Seite 64*

IUPAC *International Union of Pure and Applied Chemistry* Die International Union of Pure and Applied Chemistry legt u. a. einen Einbuchstaben-Code für Nukleinsäuren fest, der auch eventuelle Unvollständigkeiten ausdrücken kann. Siehe auch Cornish-Bowden (1985). *Seite 24*

man-page *Manual Page* Bezeichnet die Hilfe- und Dokumentationsseiten unter UNIX-artigen Betriebssystemen. *Seite 42*

MFE *Minimum Free Energy* *Seite 12*

mRNA *messenger-RNA* Auch Boten-RNA. Während der Transkription entstehende RNA-Kopie eines DNA-Abschnittes, anhand der am Ribosom ein Protein translatiert wird. *Seite 2*

MWM *Maximum Weighted Matching* Algorithmus aus der Klasse der Zuordnungsprobleme. Angewendet auf RNA stellt der MWM-Algorithmus eine RNA-Sequenz als Liste möglicher Basenpaare dar, die in einem Graph modelliert werden. Durch Anwendung des Algorithmus ist die Vorhersage von Sekundär- und Tertiär-Strukturen mit einem Aufwand von nur $\mathcal{O}(N^3)$ bei Sequenzlänge N möglich. Siehe Tabaska *et al.* (1998). *Seite 83*

N/A *Not Available/Applicable* Nicht verfügbar. *Seite 59*

ncRNA *non-protein-coding RNA* Nicht-Protein-kodierende RNAs, auch strukturelle RNAs. Hierzu werden üblicherweise auch Motive in untranslatierten mRNA-Regionen (wie SECIS, Riboswitches etc.) gezählt. *Seite 1*

[2] http://www.gnu.org/

NDB *Nucleic Acid Database* Datenbank mit aufgelösten 3D-Strukturen von Nukleinsäuren. Siehe Berman *et al.* (1992). *Seite 84*

NJ *Neighbour-Joining* Neighbour-Joining ist eine Methode, die es ermöglicht phylogenetische Bäume aus evolutionären Distanz-Matrizen zu erstellen. Hierbei wird zunächst von einem sternförmigen Baum ausgegangen und Paare gesucht, die die Verzweigungslänge minimieren. Eine genaue Beschreibung findet sich in Saitou & Nei (1987). Siehe auch UPGMA. *Seite 7*

NMR *Nuklearmagnetische Resonanz* Eine Methode zur 3D-Strukturaufklärung, die auf Kernspinresonanz basiert. *Seite 42*

PAM *Percent Accepted Mutation* Protein-spezifische Substitutionsmatrix, deren Werte auf globalen Alignments nah verwandter Proteine basieren. Siehe Dayhoff *et al.* (1978). *Seite 10*

PDB *Protein Data Bank* Datenbank aufgelöster 3D-Strukturen. Siehe Berman *et al.* (2000). *Seite 44*

Rfam *RNA family Database* Datenbank von ncRNA-Alignments und Kovarianz-Modellen. Pendant der Pfam (Protein families database). Siehe Griffiths-Jones *et al.* (2003, 2005). *Seite 12*

RIBOSUM *Ribosomal RNA Substitution Matrix* Sammelbegriff für die von Klein & Eddy (2003) anhand ribosomaler RNA-Alignments (SSU-Alignments der European Ribosomal RNA Database; Van de Peer *et al.*, 1994) erstellten RNA-Substitutionsmatrizen. Dabei wurde ähnlich wie bei der Erstellung der BLOSUM-Matrizen vorgegangen (Henikoff & Henikoff, 1992). *Seite 10*

RNA *Ribonucleic Acid* Ribonukleinsäure (RNS). Siehe auch DNA *Seite 1*

rRNA *ribosomale RNA* Die ncRNA, welche Bestandteil der Ribosomen ist. *Seite 1*

SCFG *Stochastic Context Free Grammars* Stochastische kontextfreie Grammatiken sind eine Spezialform von formalen Grammatiken. Basenpaarungen lassen sich durch entsprechende Produktionen einfach in kontextfreien Grammatiken formulieren. In den stochastisch kontextfreien Grammatiken ist jede Produktion mit einer Wahrscheinlichkeit belegt. Siehe Durbin *et al.* (1998). *Seite 14*

SCI *Structure Conservation Index* Maß der Sekundärstrukturkonservierung in einem RNA-Alignment. Siehe Abschnitt 3.5.4. *Seite 20*

SECIS *Selenocysteine insertion sequence* Strukturelles RNA-Element, welches dafür verantwortlich ist, dass an einem Stopp-Kodon (UGA) die 21. Aminosäure Selenocystein eingebaut wird. *Seite 40*

snoRNA *Small Nucleolar RNA* Klasse von ncRNAs, die rRNAs modifizieren. *Seite 64*

SOP *Sum-of-Pairs* Interne Bewertungsfunktion vieler Sequenz-Alignment-Programme. Siehe Abschnitt 1.2.3. *Seite 9*

SPS *Sum-of-Pairs-Score* Maß für die Übereinstimmung zweier Alignments auf Sequenz-Ebene. Siehe Abschnitt 3.5.1. *Seite 9*

SRP *Signal Recognition Particle* Der SRP vermittelt kotranslational die Translokation von sekretorischen und Membran-Proteinen. Die 7 S-RNA ist die RNA-Komponente des SRP. *Seite 53*

Tcl *Tool Command Language* Eine einfache, interpretierte Programmiersprache. Siehe Tcl Developer Xchange Homepage[3] geht meist schief wegen sonderzeichen . *Seite 36*

tRNA *transfer-RNA* Kleine RNAs, die Aminosäuren zum Ribosom transportieren, damit diese dort während Proteinbiosynthese (Translation) eingebaut werden. *Seite 1*

UPGMA *Unweighted Pair Group Method with Arithmetic Mean* Gilt als die einfachste Methode einen Stammbaum aus einer Distanzmatrix zu erstellen. Siehe auch NJ. *Seite 7*

UTR *Untranslated Regions* UTR sind die Regionen einer mRNA, die nicht in ein Protein translatiert werden und sich an das 5'- bzw. 3'-Ende der kodierenden Sequenz anschließen. *Seite 64*

Venn-Diagramm Venn-Diagramme (auch Mengendiagramme genannt) veranschaulichen grafisch Mengenbeziehungen. *Seite 54*

[3] http://www.tcl.tk/

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

Düsseldorf, den 20. Januar 2006

(Andreas Wilm)