Computer Aided DNA Sequence Design

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik
von

Udo Feldkamp

Dortmund

2005

Tag der mündlichen Prüfung: 6. 10. 2005

Dekan: Prof. Dr. Bernhard Steffen

Gutachter: Prof. Dr. Wolfgang Banzhaf
Prof. Dr. Petra Mutzel
Prof. Dr. Christof M. Niemeyer

Danksagung

Ich danke Herrn Prof. Wolfgang Banzhaf für die Begleitung meiner Forschung und die Betreuung bei der Erstellung dieser Arbeit. Meinen Kollegen vom Lehrstuhl für Systemanalyse des Fachbereichs Informatik an der Universität Dortmund danke ich für eine angenehme und kreative Arbeitsatmosphäre.

Für die gute Kooperation, die u. a. die Ergebnisse von Abschnitt 7.5 ermöglicht haben, danke ich Herrn Prof. Christof M. Niemeyer vom Fachbereich Chemie der Universität Dortmund, sowie Dr. Ron Wacker und Dr. Hendrik Schröder der Chimera Biotec GmbH. Das in Abschnitt 7.5 beschriebene Experiment wurde von der Universität Dortmund im Rahmen des Forschungsbandes „Molekulare Aspekte der Biowissenschaften“ finanziell unterstützt.

Hilmar Rauhe der Informium AG danke ich für Anregungen und Diskussionen.

Schließlich danke ich meiner Mutter, die mir das Studium ermöglicht hat, Ruth Gehrmann für ihre ausdauernde Unterstützung und Alisa, die zeitweise wenig von ihrem Papa hatte, für ihre ansteckende Fröhlichkeit.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Im zwanzigsten Jahrhundert gab es eine Vielzahl wissenschaftlicher Entwicklungen in den verschiedensten Bereichen und Disziplinen. Zwei wichtige Bereiche sind die Molekularbiologie und die Mikrotechnologie. Ein Höhepunkt für die Biologie war die Entdeckung der DNA-Doppelhelix, die 1953 von James Watson und Francis Crick veröffentlicht wurde [173]. Die Kenntnis der molekularen Struktur des Stoffes, der die Gene, also die Erbanlagen, eines jeden Lebewesen trägt, ermöglichte die Entwicklung einer Vielzahl von Techniken zur Untersuchung von Lebewesen, ihrer Gene, der Entwicklung eines Organismus aus einer Zelle, und der Vererbung und Evolution auf molekular- statt nur populationsgenetischer Ebene. Hierauf begründen sich auch die gegen Ende des Jahrhunderts entstandenen und auch heute noch wachsenden Teildisziplinen der *Genomics* und der *Proteomics*, also der Untersuchung der Gesamtheit aller Gene bzw. aller durch sie kodierten Proteine eines Organismus. Ein weiterer Höhepunkt war in diesem Bereich sicherlich die Veröffentlichung des nahezu kompletten menschlichen Genoms durch zwei unabhängige Gruppen im Jahr 2001 [79, 168].

In der Technologie fand eine fortschreitende Miniaturisierung von Werkzeugen und Produkten statt, bis in den Mikrometerbereich hinein. Diese Mikrotechnologie erlaubte nicht nur die Konstruktion moderner Computer, was nicht zuletzt ein wachsende Bedeutung der Informatik und letztendlich auch diese Arbeit zur Folge hat, sondern auch die Herstellung winziger Geräte für den Einsatz in der Medizin, der Chemie, und wiederum auch der Molekularbiologie. Die weitere Miniaturisierung in den Nanometerbereich hinein bietet jedoch einige Schwierigkeiten. Für die Herstellung von Mikrochips gängige Verfahren wie die Photolithographie ließen sich nur mit erheblichem Aufwand auf die Herstellung von Nanostrukturen übertragen. Bei der Erstellung von Konstrukten im Nanometerbereich durch Manipulation einzelner Moleküle oder gar Atome stellen sich die Probleme der *dicken Finger* (die Manipulationswerkzeuge sind größer und gröber als die Werkstoffe und Werkstücke) und der *klebrigen Finger* (van-der-Waals-Kräfte zwischen Werkzeug und Werkstück machen sich in diesem Maßstab bemerkbar).

Als ein Ausweg für die Nanotechnologie gelten daher selbstorganisierende Konstruktionsmethoden [22], insbesondere das *molekulare Self-Assembly*, also das selbsttätige Zusammenfügen von Molekülen zu größeren Strukturen. Weitere Aspekte der Selbstorganisation, wie Selbsterhaltung und Entwicklung, werden hierbei nicht betrachtet [23]. Wünschenswert sind hierbei natürlich Moleküle, bei denen einfach vorzunehmende Änderungen ihrer Eigenschaften zur Änderung der Strukturen, die bei dem durch die Moleküleigenschaften definierten Self-Assembly eingenommen werden, führen, die also eine *Programmierbarkeit* des Self-Assembly erlauben. Die DNA ist ein solches Molekül. Zwei DNA-Stränge haben die Fähigkeit, sich spontan zusammenzufügen, wenn die Abfolgen der in ihnen enthaltenen Basen zueinander kom-

plementär sind. Durch die Wahl der Basensequenzen kann man also vorherbestimmen, welche Moleküle sich wie miteinander verbinden. Da DNA zudem leicht synthetisierbar ist und es aus der Molekularbiologie viele Techniken zur Behandlung von DNA gibt, stellt sie sich unabhängig von ihrer biologischen Bedeutung als idealer Werkstoff für das programmierbare Self-Assembly dar.

Leider gibt es in der Anwendung auch hier praktische Probleme, da die Hybridisierung, also die Verbindung zweier komplementärer DNA-Moleküle, ein stochastischer Prozeß ist. Das bedeutet, daß nicht alle Moleküle einen Bindungspartner finden, daß bereits eingegangene Bindungen sich wieder lösen können, und daß auch Moleküle, die nur teilweise komplementär sind, Bindungen eingehen können. Um Wahrscheinlichkeiten für das Auftreten solcher Fehler zu minimieren, also ein *effizientes* und *spezifisches* Self-Assembly zu ermöglichen und damit die Determiniertheit der Programmierung zu erhöhen, werden Verfahren zur Auswahl bzw. zum Entwurf von geeigneten Basensequenzen benötigt. Aufgrund der hohen Anzahl möglicher Sequenzen und der oft aufwendigen Berechnung ihrer Eigenschaften ist die Inanspruchnahme computergestützter Verfahren unverzichtbar.

In der vorliegenden Arbeit soll genau hierzu ein Beitrag geliefert werden, indem ein Verfahren zum computergestützten DNA-Sequenz-Design vorgestellt und untersucht wird. Die Neigung zweier Moleküle zur Hybridisierung wird hierbei durch das Vorhandensein komplementärer Subsequenzen fixer Länge modelliert. Die Anordnung solcher Subsequenzen in einem Graphen erlaubt die Abbildung der Suche nach spezifisch hybridisierenden Sequenzen auf die Suche nach knotendisjunkten Pfaden. Ein greedy Algorithmus realisiert nicht nur diese Suche, sondern läßt sich auch auf die Berücksichtigung einer Vielzahl von Moleküleigenschaften erweitern.

In Kapitel 2 werden zunächst die chemischen Grundlagen der DNA erläutert und einige wichtige Techniken zu ihrer Behandlung vorgestellt. Besonders detailliert wird hierbei der Hybridisierungsvorgang betrachtet. Die gängige Theorie zu dessen Ablauf sowie zur Stabilität der Verbindungen wird zusammengefaßt.

Kapitel 3 definiert das zu lösende Problem des DNA-Sequenz-Design. Hierzu wird zunächst das Ziel, das programmierbare Self-Assembly, beschrieben. Die wichtigsten Anwendungsgebiete, also das DNA-Computing, die DNA-Nanotechnologie und die Suche nach PCR-Primern und Microarray-Sonden werden vorgestellt und mit einer Auswahl von Beispielen illustriert. Schließlich wird das DNA-Sequenz-Design-Problem selbst sowohl in allgemeiner als auch in einer spezielleren Formulierung definiert.

Dem ersten Teilproblem des DNA-Sequenz-Design, der Modellierung der Hybridisierungswahrscheinlichkeit, widmet sich Kapitel 4. Ein verbreitetes Modell bildet die Ähnlichkeit einer Sequenz zum Komplement einer zweiten Sequenz auf die Neigung der beiden zur Hybridisierung ab. Gemessen wird diese Ähnlichkeit mit Hilfe von Distanzmaßen für Zeichenketten. Eine ganze Reihe dieser Maße werden vorgestellt und sowohl theoretisch als auch in einem Experiment auf ihre Eignung zur Schätzung der Hybridisierungsneigung überprüft. Außerdem werden auch einige weitere Modellierungen erläutert, insbesondere die im in dieser Arbeit vorgestellten Verfahren verwendete Einmaligkeit von Subsequenzen fixer Länge.

Das zweite Teilproblem des DNA-Sequenz-Design ist die Wahl eines Algorithmus, der Sequenzen sucht oder konstruiert, die gemäß dem im ersten Teilproblem gewählten Modell effizient und spezifisch hybridisieren. Kapitel 5 gibt zunächst einen Überblick über veröffentlichte Verfahren anderer Gruppen und diskutiert diese kurz.

Anschließend wird in Kapitel 6 der graphbasierte Algorithmus ausführlich dargestellt und seine Vor- und Nachteile diskutiert. Einige Programme, die auf diesem Algorithmus beruhen, werden kurz vorgestellt.

In Kapitel 7 wird eine Reihe von Experimenten beschrieben, in denen verschiedene Eigenschaften von Sequenzen, die mit dem hier vorgestellten Algorithmus erzeugt wurden, untersucht werden. Die Größen generierter Sequenzmengen für verschiedene Parameterwahlen werden mit den theoretischen Maximalwerten verglichen und erzeugte Sequenzen werden auf ihre Neigung zur Bildung von einzelsträngigen Sekundärstrukturen und Fehlhybridisierungen geprüft. Veröffentlichte Sequenzmengen anderer Gruppen werden mit generierten Sequenzmengen bzgl. mehrerer Gütekriterien verglichen. Während alle diese Experimente *in silico*, also im Rechner durchgeführt wurden, konnte in einem letzten Experiment die Güte von erzeugten Sequenzen auch *in vitro*, also in der biochemischen Anwendung, überprüft und mit der anderer, veröffentlichter Sequenzen verglichen werden.

Anhand von vier Beispielen wird das Design von komplexeren DNA-Strukturen mit einem der in dieser Arbeit vorgestellten Programme in Kapitel 8 demonstriert.

Abschließend werden die wichtigsten Ergebnisse in Kapitel 9 noch einmal zusammengefaßt und einige offene Fragen und Anknüpfungspunkte für zukünftige Forschung angesprochen.

Kapitel 2

Chemische Grundlagen zu Nukleinsäuren

2.1 Eigenschaften von DNA

Ein Desoxyribonucleinsäuremolekül (deoxyribonucleic acid, DNA) ist eine Kette sogenannter Nukleotide. Jedes Nukleotid besteht aus einem Fünf-Kohlenstoff-Zucker (Desoxyribose), einer Purin- oder Pyrimidinbase, und einem Phosphatrest am 5′-Kohlenstoffatom des Zuckers. Dieser Phosphatrest ist mit dem 3′-Kohlenstoffatom des nächsten Nukleotids in der Kette verbunden. Die beiden Enden eines DNA-Strangs werden üblicherweise nach den freien Kohlenstoffatomen des ersten bzw. letzten Nukleotids als 5′-Ende und 3′-Ende bezeichnet. Die Base eines Nukleotids ist entweder eine der beiden Purinbasen Adenin und Guanin oder eine der beiden Pyrimidinbasen Cytosin und Thymin. Diese Basen werden üblicherweise mit ihren Anfangsbuchstaben abgekürzt, so daß man einen DNA-Strang als Zeichenkette aus `A`, `C`, `G` und `T` schreiben kann. Oft werden 5′- und 3′-Ende markiert (z. B. 5′-`ACGTGACAG`-3′, 3′-`GCTACAT`-5′). Stränge ohne eine solche Markierung sind per Konvention von links nach rechts in 5′-3′-Richtung zu lesen. Wiederholungen von Basen oder Basenmustern werden oft durch Angabe des Musters und der Anzahl der Wiederholungen abgekürzt. So steht z. B. $\texttt{A}_5$ für `AAAAA` und $(\texttt{GCT})_3$ für `GCTGCTGCT`. Kurze DNA-Moleküle (< 100 Nukleotide (nt)) werden als Oligonukleotide oder Oligomere bezeichnet, längere Moleküle als Polymere. Die ebenfalls in der Natur vorkommende (und auch bei einigen der in dieser Arbeit vorgestellten Anwendungen verwendete) Ribonucleinsäure (RNA) unterscheidet sich von der DNA hauptsächlich durch ein anderes Zuckermolekül, sowie durch die Verwendung der Base Uracil (Abk. `U`) anstelle von Thymin.

Während die Zuckermoleküle einer Kette kovalent miteinander verbunden sind und so das sogenannte Rückgrad (*backbone*) des DNA-Strangs bilden, können zwei Basen eine nichtkovalente Wasserstoffbrückenbindung miteinander eingehen und ein Basenpaar bilden. Eine solche Bindung findet normalerweise nur zwischen einer Adenin- und einer Thyminbase (über zwei Wasserstoffbrücken) bzw. zwischen einer Cytosin- und einer Guaninbase (über drei Wasserstoffbrücken) statt. Daher werden Adenin und Thymin (bzw. Cytosin und Guanin) auch als komplementär zueinander bezeichnet. Zwei DNA-Stränge heißen zueinander komplementär, wenn die i-te Base des einen Strangs, in 5′-3′-Richtung gelesen, komplementär zur i-ten Base des anderen Strangs, in 3′-5′-Richtung gelesen, ist (z. B. 5′-`AGCG`-3′ und 5′-`CGCT`-3′). Ein Strang, der zu sich selbst komplementär ist, heißt selbstkomplementär (z. B. 5′-`AAGCTT`-3′). Der Bezeichner einer DNA-Sequenz, die zu einer gegebenen Sequenz komplementär ist, wird

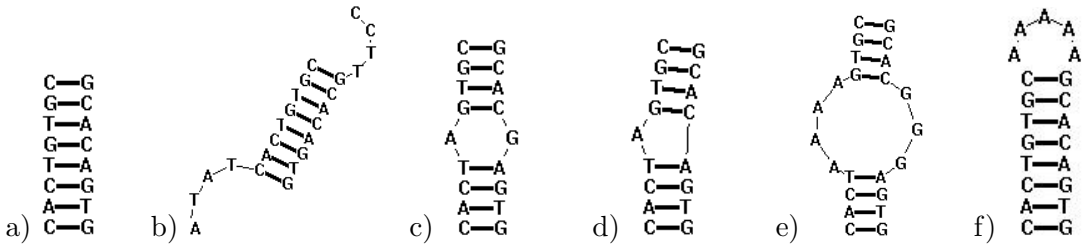


Abbildung 2.1: Verschiedene Formen der Hybridisierung. a) perfekter Duplex, b) Duplex mit dangling Ends (auch sticky Ends genannt), c) Single Base Pair Mismatch, d) Single Base Bulge Loop, e) Internal Loop, f) Hairpin Loop. Ein breiter Strich repräsentiert alle Wasserstoffbrücken je eines Basenpaars, dünne Striche deuten den Verlauf des Backbone an.

in dieser Arbeit (wie häufig in der Literatur) durch einen Querstrich markiert, so ist z. B. die Sequenz $\overline{X}$ komplementär zur Sequenz X. Die Wasserstoffbrücken können leicht gelöst werden, z. B. durch Erhöhung der Temperatur. Dadurch schmilzt ein doppelsträngiges DNA-Molekül (auch Duplex genannt) zu zwei einzelsträngigen. Der umgekehrte Prozeß wird Hybridisierung genannt . Zwei DNA-Stränge müssen nicht perfekt komplementär sein, um einen Duplex bilden zu können. Es ist auch möglich, daß nur Teilbereiche komplementär sind und hybridisieren, während andere Teile derselben Stränge einzelsträngig bleiben. Dies führt zu verschiedenen möglichen Formen von Duplexen, die Fehlpaarungen (Mismatches), Ausbuchtungen (Bulges) und Schleifen (Loops) enthalten können. Zusätzlich zu diesen intermolekularen Hybridisierungen können auch intramolekulare Basenpaarungen auftreten, bei denen verschiedene Bereiche ein und desselben Strangs, wenn sie zueinander komplementär sind, hybridisieren und sogenannte Haarnadelschleifen (Hairpin Loops) und andere einzelsträngige Sekundärstrukturen bilden (Abb. 2.1). I. a. liegen Kombinationen dieser Hybridisierungsformen in Duplexen und Sekundärstrukturen eines Strangs vor. Einzelsträngige DNA wird oft mit ssDNA (*single stranded DNA*) bezeichnet, doppelsträngige DNA mit dsDNA (*double stranded DNA*).

Es existieren viele Labortechniken zur Verarbeitung von DNA. Bestimmte Enzyme, Exonukleasen, verdauen DNA-Moleküle, indem sie Nukleotid für Nukleotid von der Kette entfernen. Eine bestimmte Exonuklease arbeitet normalerweise in eine bestimmte Richtung ($5'$-$3'$ oder umgekehrt) und verdaut spezifisch nur einzel- oder nur doppelsträngige DNA. Endonukleasen zerschneiden einen DNA-Strang in zwei Moleküle durch Lösen kovalenter Bindungen im Rückgrad der DNA. Bei Restriktionsnukleasen geschieht dies in oder nahe einer enzymspezifischen Erkennungssequenz, der Restriktionsschnittstelle. Ligase, ein weiteres Enzym, verbindet die Rückgrade zweier DNA-Stränge kovalent zu einem Strang (Abb. 2.2). DNA-Moleküle können weiterhin per Gelelektrophorese nach ihrer Länge sortiert, oder ihre Basensequenz kann mittels Sequenzierungstechniken ausgelesen werden. Zur Herstellung von DNA-Molekülen mit einer bestimmten Basensequenz existieren Syntheseverfahren, die Oligomere bis zu einer Länge von etwa 100 Basen in ausreichenden Mengen produzieren können.

DNA-Stränge können mit einem bestimmten Enzym, der Polymerase, kopiert und mit der sogenannten Polymerase-Kettenreaktion (Polymerase Chain Reaction, PCR) exponentiell vervielfältigt werden. So können auch Moleküle in ursprünglich niedriger Konzentration durch *Amplifizierung* in ausreichender Menge z. B. für weitere Analysetechniken zur Verfügung gestellt werden [63] (Abb. 2.3). Soll eine Sequenz X, die Subsequenz einer längeren Sequenz wie einem Gen oder einer Gruppe von Genen sein kann, sowie deren Komplementärsequenz $\overline{X}$ (die beiden sogenannten *Template-Sequenzen*) amplifiziert werden, so benötigt man zwei

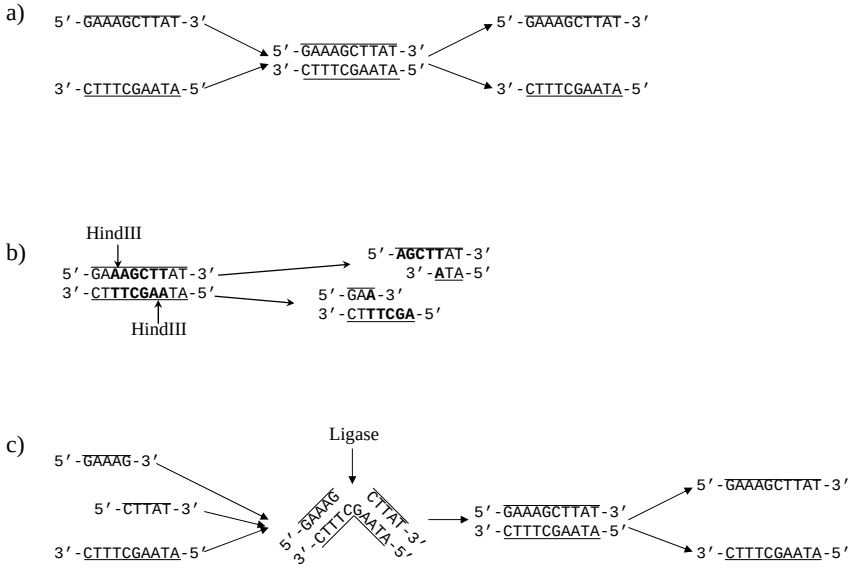


Abbildung 2.2: DNA-Operationen. Die Disoxyribose-Backbones sind in dieser Skizze durch Linien angedeutet. a) Hybridisierung und Schmelzen. Zwei einzelsträngige Moleküle mit komplementärer Basensequenz verbinden sich durch Bildung von Wasserstoffbrücken zu einem Duplex. Durch Erhöhung der Temperatur lassen sich diese Verbindungen wieder lösen, der Duplex „schmilzt“ in zwei Einzelstränge. b) Restriktion. Die Nuklease HindIII bindet an die selbstkomplementäre Basensequenz `AAGCTT` eines DNA-Doppelstrangs und durchtrennt die kovalente Verbindung in beiden Backbones jeweils zwischen den Adenin-Basen. c) Ligation. Ligase stellt eine kovalente Bindung zwischen einem 3′- und einem 5′-Ende zweier DNA-Stränge her. Um die Effizienz dieser Operation zu erhöhen, sorgt man oft durch Hybridisierung auf einen dritten Strang dafür, daß die beiden zu verbindenden Enden nahe beieinander sind. Häufig schließt der Begriff Ligation die vorangehende Hybridisierung mit ein.

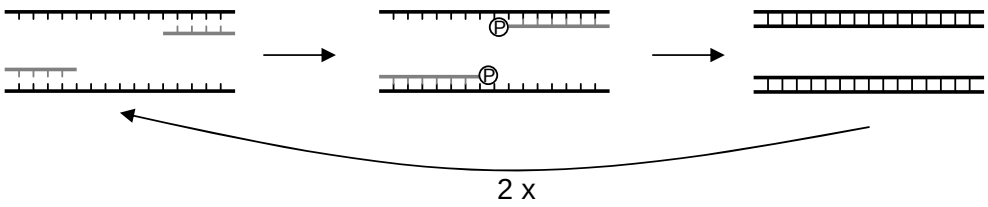


Abbildung 2.3: Schema einer Polymerase-Kettenreaktion (PCR). Der zu amplifizierende Duplex wird in seine Einzelstränge aufgeschmolzen, die als Templates dienen. An deren 3′-Enden werden Primer hybridisiert (links). Das Polymerase-Enzym verlängert die Primer basenweise in 3′-Richtung komplementär zum Template (mitte). Die Verlängerung endet, wenn der verlängerte Strang genauso lang ist wie das Template (links). Nun liegen zwei identische Duplexe vor, die wieder aufgeschmolzen werden, es werden Primer hybridisiert usw.

kurze Oligomere (*Primer*), die jeweils komplementär zu den Subsequenzen am 3′-Ende von X und $\overline{X}$ sind, und daher *in vitro* mit diesen Enden hybridisieren. Das Polymerase-Enzym verlängert dann die Primer an ihrem jeweiligen 3′-Ende, indem sie für jede Base des Templates die komplementäre Base an den Primer anfügt. So wird Base für Base die Kopie ergänzt. Ist die Verlängerung abgeschlossen, so werden die neuentstandenen Duplexe wieder aufgeschmolzen, X und $\overline{X}$ liegen nun in verdoppelter Anzahl vor. Nun werden die Primer durch Abkühlen wieder an die Templates hybridisiert und der Kopierzyklus beginnt von Neuem. Da jeder Zyklus die Anzahl der Stränge verdoppelt, wächst diese exponentiell mit der Anzahl der Zyklen, zumindest in den ersten Zyklen (bei üblichen Protokollen gilt dies für 20 bis 30 Zyklen [63]).

DNA-Microarrays (oder DNA-Chips) sind Glasträger, auf denen in regelmäßigen Abständen Vertiefungen, sog. *Spots* aufgebracht sind. In jedem dieser Spots sind viele Kopien einer DNA-Sequenz an einem ihrer beiden Enden über ein Linkermolekül mit der Glasoberfläche verbunden. Diese Sequenzen heißen *Sonden* und werden komplementär zu nachzuweisenden Sequenzen, den *Zielsequenzen*, gewählt. Wird nun eine Lösung mit DNA-Strängen auf den Chip gegeben, hybridisieren die Zielsequenzen, sofern in der Lösung vorhanden, mit den zu ihnen komplementären Sonden. Wurden die in Lösung befindlichen Moleküle vorher mit einem Fluoreszenz-Farbstoff markiert, so leuchten unter einer UV-Lampe die Spots auf, an denen sich viele Moleküle der Zielsequenz angelagert haben. Sonden, deren komplementären Zielsequenzen nicht in der Lösung vorhanden waren, bleiben einzelsträngig, die entsprechenden Spots sind dunkel. Über die Signalintensität des Lichts lassen sich sogar die Konzentrationen, mit denen verschiedene DNA-Moleküle in der Lösung vorhanden sind, zumindest relativ zueinander bestimmen. Neben dem Einsatz in bestimmten DNA-Computing-Modellen und in der Nanotechnologie (siehe nächstes Kapitel) werden Microarrays sehr stark in der Molekulargenetik verwendet, z. B. für die Genexpressionsanalyse, also die Messung, wie stark ein bestimmtes Gen in der Zelle abgelesen und sein Produkt (RNA, Protein) konzentriert sein wird.

Im Rechner werden meist Nukleotide zu Buchstaben und Oligo- und Polymere zu Zeichenketten, genauer zu Wörtern über dem Alphabet $\Sigma = \{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$ abstrahiert. Eigenschaften z. B. der räumlichen Struktur oder der lokalen Ladungsverteilung werden ignoriert bzw. auf ihre Auswirkung auf die Stabilität eines Duplex reduziert und dazu implizit in den thermodynamischen Parametern eines Hybridisierungsmodells erfaßt (siehe nächsten Abschnitt).

2.2 Hybridisierung genauer betrachtet

Zu sagen, daß sich zwei komplementäre DNA-Stränge treffen, hybridisieren und einen Duplex bilden, ist eine starke Vereinfachung der Realität, die einige Fehlerquellen bei Hybridisierungsprozessen ignoriert. Diese müssen aber beim DNA-Sequenz-Design berücksichtigt werden, daher ist eine genauere Betrachtung des Prozesses notwendig.

Zunächst ist die Hybridisierung von DNA-Molekülen ein stochastischer Prozeß, d. h. man hat normalerweise nicht nur je ein Exemplar der beteiligten Sequenzen, sondern sehr viele Kopien in Lösung. Da diese z. B. in einem Reagenzglas nicht räumlich sortiert sind, gibt es einen gewissen Anteil von Molekülen, die keinen geeigneten Partner in ihrer Nähe finden, mit dem sie hybridisieren können, insbesondere wenn die meisten potentiellen Partner bereits in Duplexen gebunden sind. Desweiteren ist die Hybridisierung eine Reaktion, die prinzipiell in beide Richtungen abläuft, d. h. ein gewisser Anteil der Duplexe schmilzt auch wieder in zwei Einzelstränge. Abhängig von Reaktionsbedingungen wie Temperatur, pH-Wert oder Konzentration von Salzen pendelt sich die Reaktion auf einen Gleichgewichtszustand ein, bei dem ein Anteil der DNA einsträngig, der andere Anteil doppelsträngig vorliegt, und in beide Richtungen gleicher Reaktionsumsatz erreicht ist, d. h. es werden ebensoviele Duplexe durch Hybridisierung erzeugt wie durch Schmelzen zerstört. Um eine möglichst hohe *Hybridisierungseffizienz* zu erreichen, also einen möglichst hohen Anteil an gepaarter DNA, muß man das Gleichgewicht soweit wie möglich zum gewünschten Zustand verlagern. Dies kann durch geeignete Wahl der Reaktionsbedingungen erfolgen, z. B. wird durch Abkühlen der Lösung die Stabilität der Duplexe erhöht, so daß nur noch sehr wenige Duplexe wieder schmelzen, der überwiegende Anteil der DNA also nach Erreichen des Gleichgewichts doppelsträngig vorliegt.

Es befinden sich i. a. bei Anwendungen in interessanten Größenordnungen nicht nur eine Sequenz und ihr Komplement in Lösung, sondern viele verschiedene Sequenzen. Da auch Moleküle hybridisieren können, die nur teilweise komplementär sind, konkurrieren diese Sequenzen mit dem jeweils beabsichtigten, perfekt komplementären Partner einer Sequenz um Bindung. Solche unbeabsichtigten Paarungen werden als *Fehl-* oder *Kreuzhybridisierungen* bezeichnet. Starkes Abkühlen zur Verlagerung des Reaktionsgleichgewichts zum hybridisierten Zustand erhöht auch die Stabilität und damit die Auftrittswahrscheinlichkeit von Kreuzhybridisierungen. Das Gleichgewicht kann also nicht beliebig weit verschoben werden. Die Vermeidung von Kreuzhybridisierungen unter Maximierung der Effizienz ist eine Hauptaufgabe des Designs von DNA-Sequenzen (siehe nächstes Kapitel).

Die Hybridisierung zweier DNA-Stränge ist weiterhin natürlich kein instantaner Prozeß, bei dem im einen Moment noch zwei Einzelstränge, im nächsten Moment schon ein Doppelstrang vorliegt. Der tatsächliche zeitliche Ablauf wird in zwei Phasen unterteilt, die Nukleations- und die Wachstumsphase (Abb. 2.4). Zunächst bilden zwei komplementäre Nukleotide ein Basenpaar (Nukleation), anschließend schließen sich die Nachbarn dieser Nukleotide zu einem Basenpaar zusammen, dann deren Nachbarn usw. (Wachstumsphase). Im gängigen Energiemodell dieses Vorgangs ist für die Stabilität des ersten Basenpaars (neben Reaktionsbedingungen wie Temperatur und Salzkonzentration) die Anzahl der Wasserstoffbrücken entscheidend. Bei den weiteren, in der Wachstumsphase hinzugefügten Basenpaaren dagegen dominiert das sogenannte *stacking*, d. h. das räumliche Aufeinanderstapeln von benachbarten Basenpaare, so daß diese zusammen die bekannte Doppelhelix bilden [34]. Die Bildung des ersten Basenpaars ist energetisch eher ungünstig, das erste Basenpaar also instabil. Erst das Stacking von zwei weiteren Basenpaaren stabilisiert diesen Nukleus ausreichend, damit das weitere Wachstum des Doppelstrangs spontan ablaufen kann [142]. Daher ist das Auftreten mehrerer Nukleationen mit anschließenden parallel ablaufenden Wachstumsphasen bei kurzen DNA-Molekülen eher

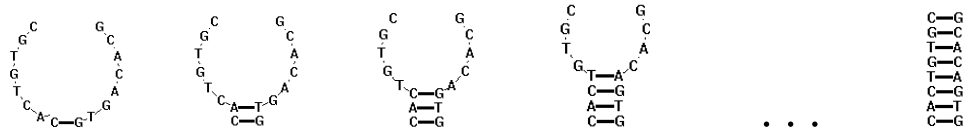


Abbildung 2.4: Zeitlicher Ablauf der Hybridisierung. Zunächst bildet sich nur ein Basenpaar, vorzugsweise ein G-C-Basenpaar (Nukleation). Anschließend bilden sich weitere Paare, vom Nukleus ausgehend, wie ein sich schließender Reißverschluß (Wachstumsphase). Der Nukleus muß sich nicht, wie hier dargestellt, am Ende des werdenden Duplex bilden.

unwahrscheinlich, bei längeren Sequenzen dagegen durchaus möglich.

Zur vereinfachten Betrachtung von Hybridisierungen gibt es verschiedene Modelle [34] (Abb. 2.5):

- Das *aligned model* betrachtet nur perfekte Duplexe. D. h. die Hybridisierung erfolgt kontinuierlich und beide Stränge sind in Phase, also nicht gegeneinander verschoben. Dieses Modell ist nur für sehr kurze Oligonukleotide realistisch, bei denen Verschiebungen mit hoher Wahrscheinlichkeit instabil sind.
- Bei *staggering zipper model* sind Verschiebungen der beiden Stränge gegeneinander enthalten, es werden aber auch hier nur kontinuierliche Duplexbildungen betrachtet.
- Im *allgemeinen Modell* sind neben Verschiebungen auch nichtkontinuierliche Hybridisierungen erlaubt.
- Das *all-or-none model* abstrahiert den zeitlichen Ablauf der Hybridisierung so weit, daß nur zwei Zustände betrachtet werden: Alle Basenpaare sind gebildet oder beide Moleküle liegen komplett einsträngig vor. Auch dieses Modell ist nur für kurze Oligonukleotide gültig, bei denen die Zwischenzustände als energetisch ungünstig angesehen werden können, die schnell wieder durch stacking des nächsten Basenpaars verlassen werden. Bei längeren Sequenzen kann es durchaus dazu kommen, daß stabilere Teilsequenzen (reich an G-C-Basenpaaren) hybridisieren, weniger stabile Bereiche aber einzelsträngig bleiben.

Entsprechende Überlegungen und Modelle gelten natürlich nicht nur für intermolekulare Hybridisierung zweier DNA-Stränge, sondern auch für die intramolekulare Duplexbildung zwischen Teilbereichen desselben Moleküls bei der Bildung von einzelsträngigen Sekundärstrukturen.

2.3 Duplexstabilität

Die Stabilität eines Hybridisierungszustandes und davon abhängig die Wahrscheinlichkeit, daß eine Menge von Molekülen diesen Zustand einnimmt, wird in der Literatur über die freie Enthalpie sowie über die Schmelztemperatur gemessen. Die freie Enthalpie (auch Gibbs-Energie genannt), oder genauer die Differenz der freien Enthalpien ΔG beider Zustände einer Reaktion (z. B. einsträngig und doppelsträngig), ist definiert als die Differenz aus Enthalpiedifferenz und dem Produkt aus Temperatur und Entropiedifferenz: $\Delta G = \Delta H - T\Delta S$. Damit ist sie ein Maß dafür, wie spontan eine Reaktion abläuft. Es läßt sich aus ihr auch berechnen, bei welchem Konzentrationsverhältnis von Produkten zu Edukten sich das Reaktionsgleichgewicht einstellen wird, und damit eine Aussage treffen, wie stabil die Produkte sind. Für spontane

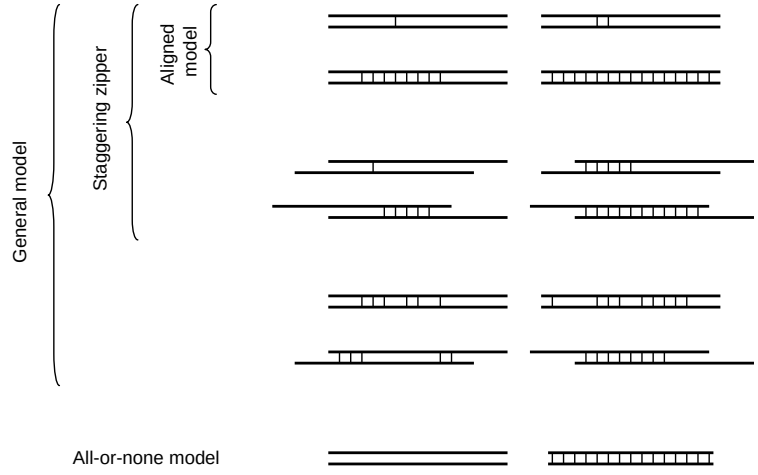


Abbildung 2.5: Schematische Darstellung der Hybridisierungsmodelle (nach [34]). Gezeigt sind die vier im Text beschriebenen Modelle *aligned model*, *staggering zipper model*, *general model* und *all-or-none model*. In der Darstellung stehen die stärkeren Linien für die Backbones, die dünneren Linien für die Wasserstoffbrücken der Watson-Crick-Basenpaare.

Reaktionen ist ΔG negativ, je größer der Betrag, desto stärker ist das Reaktionsgleichgewicht in Richtung Produkte verschoben, als desto stabiler können die Produkte also angesehen werden. Da auch die Entropiedifferenz negativ ist, erkennt man an der Definition, daß höhere Temperaturen ΔG erhöhen, die Stabilität eines Duplex also verringern.

Die Schmelztemperatur T_m einer Menge von DNA-Molekülen ist definiert als die Temperatur, bei der 50 % der DNA einzelsträngig vorliegt [92] (Abb. 2.6). Damit ist eine hohe Schmelztemperatur ein Indiz für stabile Duplexe.

Sowohl ΔG als auch T_m hängen nicht nur von der Reaktionsumgebung ab, sondern auch stark von der Basenzusammensetzung und -abfolge der DNA-Sequenz. Es existieren verschiedene Methoden, um die Schmelztemperatur von DNA zu berechnen, eine davon läßt sich auch für die Berechnung der freien Enthalpie verwenden.

Wallace-Regel Unter der Modellannahme, daß die Anzahl der Wasserstoffbrücken die Schmelztemperatur bestimmt, ergibt sich aus Empirie für eine DNA-Sequenz die Abschätzung

$$T_m = N_{AT} \cdot 2°C + N_{GC} \cdot 4°C, \tag{2.1}$$

wobei N_{AT} die Anzahl der `A-T`-Basenpaare und N_{GC} die Anzahl der `G-C`-Basenpaare in der Sequenz ist [141]. Diese Formel ist nur für sehr kurze Sequenzen ($\leq$ 20 nt) wenigstens annähernd gültig. Sowohl die sehr grob geschätzten Parameter 2 und 4 als auch das Ignorieren des Stackings im Modell führen bei längeren Sequenzen zu sehr großen Fehlern.

GC-Prozent-Formel Dieses Modell beruht ebenfalls auf einem linearen Zusammenhang zwischen dem Anteil an `G-C`-Basenpaaren in der Sequenz und der Schmelztemperatur [44, 34].

$$T_m = 81.5 + 41 \cdot R_{GC} - \frac{600}{l} \tag{2.2}$$

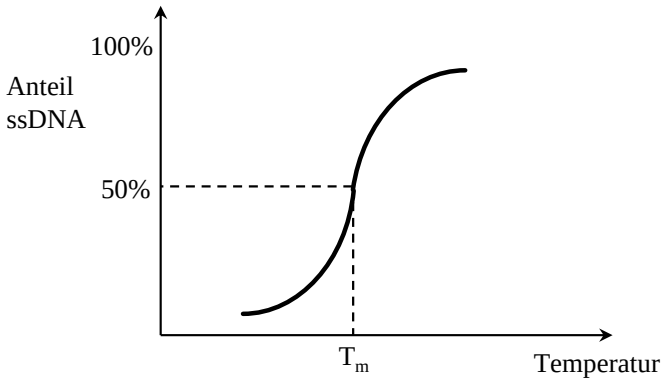


Abbildung 2.6: Skizze einer Schmelzkurve. Mit steigender Temperatur erhöht sich der Anteil einzelsträngiger DNA (ssDNA), da immer mehr Duplexe schmelzen. Die Schmelztemperatur T_m ist die Temperatur, bei der 50 % der DNA als ssDNA vorliegt.

wobei l die Sequenzlänge und $R_{GC} = N_{GC}/l$ der Anteil der `G-C`-Basenpaare an der gesamten Sequenz ist. Diese Formel wird hauptsächlich für längere Sequenzen (> 50 nt) verwendet. Im World Wide Web verfügbare T_m-Rechner verwenden auch häufig $\frac{500}{l}$ als letzten Summanden (z. B. [64]), was auch in [174] angegeben wird.

nearest-Neighbor-Modell Bei diesem Modell werden zunächst Enthalpie und Entropie berechnet, und aus diesen anschließend die Schmelztemperatur [33, 141]. Hierzu wird jedem Paar von benachbarten Basenpaaren der Sequenz ein Einzelbeitrag zu Enthalpie und Entropie zugeordnet, abhängig von den Basen in diesem Nachbarpaar. Die Summe der Einzelbeiträge aller (überlappenden) Nachbarpaare in der Sequenz ergeben die Gesamtenthalpie bzw. -entropie, ggf. ergänzt durch weitere Summanden für die Initiierung der Reaktion (Nukleation) und für die Symmetrie (bei selbstkomplementären Sequenzen). In den verschiedenen veröffentlichten Instanzen dieses Modells finden sich nicht nur unterschiedliche Werte für die Einzelbeiträge, sondern evtl. auch weitere Summanden, z. B. für An- oder Abwesenheit von `G-C`-Basenpaaren [33, 145]. Ein Versuch, möglichst viele dieser Instanzen anzunähern, ist das Unified Parameter Set von SantaLucia et al. [7, 144], das als Beispiel in Tabelle 2.1 angegeben ist. Enthalpie und Entropie der Hybridisierung z. B. der Sequenz `GGATCC` berechnet man mit diesem Parametersatz wie folgt:

$$\begin{aligned}\Delta H &= 2 \cdot \Delta H_{init_{GC}} + \Delta H_{GG/CC} + \Delta H_{GA/CT} + \Delta H_{AT/TA} + \Delta H_{GA/CT} + \Delta H_{GG/CC} \\ &= (2 \cdot 0.1 - 8.0 - 8.2 - 7.2 - 8.2 - 8.0)\ \text{kcal/mol} \\ &= -39.4\ \text{kcal/mol}\end{aligned}$$

$$\begin{aligned}\Delta S &= 2 \cdot \Delta S_{init_{GC}} + \Delta S_{symm} + \Delta S_{GG/CC} + \Delta S_{GA/CT} + \Delta S_{AT/TA} + \Delta S_{GA/CT} + \Delta S_{GG/CC} \\ &= (2 \cdot (-2.8) - 1.4 - 19.9 - 22.2 - 20.4 - 22.2 - 19.9)\ \text{cal/K·mol} \\ &= -111.6\ \text{cal/K·mol}\end{aligned}$$

Da z. B. $\begin{smallmatrix}5'-GA-3'\\3'-CT-5'\end{smallmatrix}$ und $\begin{smallmatrix}5'-TC-3'\\3'-AG-5'\end{smallmatrix}$ ununterscheidbar sind, tragen sie auch gleiche Beiträge zu den jeweiligen Summen bei. Durch diese Ununterscheidbarkeit reduzieren sich die $4^2 = 16$

	ΔH° (kcal/mol)	ΔS° (cal/K·mol)	ΔG° (kcal/mol)
AA/TT	-7.9	-22.2	-1.00
AT/TA	-7.2	-20.4	-0.88
TA/AT	-7.2	-21.3	-0.58
CA/GT	-8.5	-22.7	-1.45
GT/CA	-8.4	-22.4	-1.44
CT/GA	-7.8	-21.0	-1.28
GA/CT	-8.2	-22.2	-1.30
CG/GC	-10.6	-27.2	-2.17
GC/CG	-9.8	-24.4	-2.24
GG/CC	-8.0	-19.9	-1.84
init G-C	0.1	-2.8	0.98
init A-T	2.3	4.1	1.03
symm	0.0	-1.4	0.40

Tabelle 2.1: Summanden für das nearest-Neighbor-Modell aus dem Unified Parameter Set [6, 144]. Angegeben sind die Parameter für die 10 unterscheidbaren Nachbarpaare (siehe Text), für die Initiierung (Nukleation) der Hybridisierung bei terminalen `G-C`- bzw. `A-T`-Basenpaaren, sowie für die Symmetrie von selbstkomplementären Sequenzen. Das Symbol ° kennzeichnet die Standardbedingungen (Konzentrationen der Edukte und Produkte = 1 M), die als Modellannahme in die Messung der Summanden geflossen sind.

möglichen Nachbarpaare auf 10 unterscheidbare Summanden im Parametersatz (Tab. 2.1). Der Summand für die Initiierung mit `G-C`-Basenpaaren an den Enden wird zweimal angewendet, einmal für jedes Duplexende. Die Symmetrie (Selbstkomplementarität) der Sequenz trägt nichts zur Enthalpie bei [34], daher fehlt dort ein entsprechender Summand.

Aus Enthalpie und Entropie läßt sich nun die Schmelztemperatur berechnen.

$$T_m = \frac{\Delta H}{\Delta S + R \cdot \ln(c/f)} - 273.15°\mathrm{C} \tag{2.3}$$

mit $R = 1.987$ cal/(K·mol) die Gaskonstante, c die Konzentration der DNA-Moleküle, $f = 1$ [145] oder $f = 2$ [34] für selbstkomplementäre Sequenzen, und $f = 4$ sonst.

Diese Berechnungsmethode wird für Sequenzen bis zu einer Länge von 70 nt als zuverlässig angesehen [159]. Das zugrunde liegende Modell orientiert sich stark am z. Z. als realistisch betrachteten Hybridisierungsmodell. So stehen die Summanden für die Nachbarpaare für die Enthalpie bzw. Entropie des jeweiligen Stacking-Vorgangs, d. h. für die räumliche Anordnung der beiden Basenpaare zu einem Helixabschnitt, während die Nukleation mit einem eigenen Summanden berücksichtigt wird.

Ähnlich wie die Enthalpie und die Entropie läßt sich in diesem Modell auch die Gibbs-Energie ΔG berechnen (s. Tab. 2.1, letzte Spalte), die als ein zuverlässigeres Maß für die Duplexstabilität als die Schmelztemperatur angesehen wird [34]. Ein weiterer Vorteil dieses Modells ist die recht einfache Erweiterung z. B. auf Mismatches [6, 9, 10, 8, 128], Bulges [161], dangling Ends [30] und internal Loops [186], die jeweils weitere Summanden beitragen. Dadurch lassen sich z. B. auch Stabilitäten von einzelsträngigen Sekundärstrukturen berechnen [74, 186].

Abhängigkeit von Salzkonzentrationen Sowohl T_m als auch ΔG sind abhängig von den Konzentrationen, in denen verschiedene Salze in der Lösung vorliegen, und auch vom pH-Wert der Lösung. Die angegebenen Parametersätze sind für bestimmte Bedingungen experimentell bestimmt worden, meist sind dies die Standardbedingungen $[Na^+] = 1$ M und pH 7. Um T_m bzw. ΔG für andere Bedingungen zu berechen, müssen Korrekturterme verwendet werden. Z. B. werden zur Korrektur der Schmelztemperatur für andere Natriumkonzentrationen $+16.6 \log[Na^+]$ [34] für die GC-Prozent-Formel bzw. $+12.5 \log[Na^+]$ [145] für das nearest-Neighbor-Modell vorgeschlagen.

Gleichgewichtskonstanten, Partition Function In welche Richtung eine Reaktion bevorzugt verläuft, läßt sich durch die *Gleichgewichtskonstante* K angeben, die das Verhältnis von Produktkonzentrationen zu Eduktkonzentrationen im Reaktionsgleichgewicht angibt (hier also das Verhältnis von doppelsträngiger zu einzelsträngiger DNA). Für die freie Enthalpieänderung der Hybridisierung von DNA gilt $\Delta G = -RT \ln K$, wobei R die Gaskonstante (s. o.) und T die absolute Temperatur ist [34]. Umformung ergibt

$$K = e^{-\Delta G/RT}. \tag{2.4}$$

Berücksichtigt man nicht nur die Konfigurationen „komplett einzelsträngig“ und „komplett doppelsträngig“, sondern auch andere mögliche Hybridisierungsformen, so lassen sich die Gleichgewichtskonstanten aller Hybridisierungsreaktionen zur sog. *Partition Function* q_c zusammenfassen:

$$q_c = \sum_i K_i = \sum_i e^{-\Delta G_i/RT} \tag{2.5}$$

wobei K_i die Gleichgewichtskonstante und ΔG_i die freie Enthalpie der i-ten Konfiguration (Hybridisierungsform) ist. Je nach gewähltem Hybridisierungsmodell (s. o.) ändert sich auch die Zahl der Summanden, die bei der Partition Function berücksichtigt werden müssen.

Kapitel 3

Das DNA-Sequenz-Design-Problem

3.1 Das Ziel: Programmable Self-Assembly

Es gibt in der Natur viele Systeme, die zur Selbstorganisation fähig sind[22], insbesondere auch biologische und biomolekulare Systeme. So haben auch DNA-Moleküle die Fähigkeit, wenn sie in Lösung aufeinandertreffen, sich durch den Prozeß der Hybridisierung zu einem größeren Supramolekül zu verbinden. Dieser Vorgang des spontanen Aufbaus wird als *Self-Assembly* bezeichnet, und ist zunehmend für bottom-up-Konstruktionsprozesse wichtig.

Bei der DNA hängt die Neigung zur Hybridisierung weitgehend von der Basensequenz ab, eine Eigenschaft, die sich leicht bei der Synthese von DNA-Strängen vorgeben läßt. Somit läßt sich durch die Sequenzwahl in gewissem Maße vorherbestimmen, welche Moleküle sich verbinden und welche nicht, das Self-Assembly wird dadurch *programmierbar*.

Die Einschränkung „in gewissem Maße“ begründet das Betätigungsfeld des DNA-Sequenz-Designs. Der Prozeß der Hybridisierung ist nicht perfekt vorhersagbar, geschweige denn vollkommen deterministisch programmierbar, sowohl weil heutige Modelle der Hybridisierung noch unvollkommen sind, als auch weil die Hybridisierungsreaktion ein naturgemäß stochastischer Vorgang ist. Ziel des Sequenz-Designs muß es also sein, das Self-Assembly *möglichst* genau zu programmieren, d. h. die Wahrscheinlichkeit für ein unbeabsichtigtes Verhalten der Moleküle zu minimieren. Das DNA-Sequenz-Design besteht daher aus zwei Teilen:

1. die Wahl eines möglichst realistischen Modells der Hybridisierung, das trotzdem eine effiziente Berechnung / Abschätzung / Einschränkung der Hybridisierungswahrscheinlichkeit für beliebige DNA-Moleküle erlaubt, und

2. ein Algorithmus, der möglichst effizient DNA-Sequenzen findet, die im gewählten Modell mit hoher Wahrscheinlichkeit das gewünschte, und mit sehr kleiner Wahrscheinlichkeit ein anderes Hybridisierungsverhalten zeigen.

Ggf. muß der Algorithmus auch weitere Anforderungen an die Sequenzen berücksichtigen, wenn die konkrete Anwendung, in der das Self-Assembly eingesetzt wird, dies erfordert (s. u.).

Solche Anwendungen für das DNA-Self-Assembly finden sich hauptsächlich in den Bereichen des DNA-Computing und der Nanotechnologie, aber auch in einfacher Form bei Labortechniken wie der Polymerase-Kettenreaktion (PCR) und DNA-Microarrays.

3.2 Anwendungen

3.2.1 DNA-Computing

Richard Feynman stellte bereits 1959 in seiner berühmten Rede „There's plenty of room at the bottom" die Vision vor, Rechnerelemente wie logische Gatter auf die Größe von Molekülen zu beschränken, und erkannte auch bereits die hohe Informationsdichte von DNA und die vielfältigen Funktionen von biologischen Molekülen. 1994 hat Leonard Adleman diese Aspekte verbunden [2]. Er löste ein Hamilton-Pfad-Problem, also die Suche nach einem Pfad durch einen Graphen, der jeden Knoten genau einmal besucht, indem er Knoten und Kanten so in DNA-Stränge kodierte, daß diese per Self-Assembly alle möglichen Pfade durch den Graphen bildeten, und sortierte anschließend mit normalen Labortechniken alle Pfade aus, die keine Hamiltonpfade waren (s. u.). Die Veröffentlichung dieses Experiments legte den Grundstein für das rasch wachsende Forschungsgebiet des *DNA-Computing*. Grund für die Begeisterung für dieses Thema war vor allem die hohe Parallelität der Berechnung, insbesondere beim Self-Assembly-Schritt, die sich bei Adleman durch das gleichzeitige Hybridisieren und Ligieren von etwa 10^{14} Molekülen ergab. Als weiteren Vorteil nennt Adleman die Energieeffizienz der Berechnung. Er hat ausgerechnet, daß für rund $2 \cdot 10^{19}$ Operationen 1 Joule Energie umgesetzt wird, während das theoretische Maximum bei etwa $34 \cdot 10^{19}$ Operationen/Joule liegt. Außerdem erlaubt die Kodierung von Informationen in DNA-Moleküle eine Speicherdichte von ca. 1 Bit/nm^3 [2].

Leider bleiben NP-vollständige Probleme auch bei Verwendung von DNA NP-vollständig. Die Parallelisierung der Pfadkonstruktion erlaubt zwar einen Algorithmus, dessen Anzahl von Rechenschritten nur linear mit der Eingabegröße wächst, dafür wächst der Bedarf an Speicherplatz und Parallelprozessoren, hier also an DNA, exponentiell mit der Anzahl von Knoten im Eingabegraph. Nach verschiedenen Schätzungen wurde man zur Lösung von Hamilton-Pfad-Problemen mit dem Adleman-Protokoll für 23 Knoten bereits einige Kilogramm DNA benötigen, bei 70 Knoten (alternative Schätzung: 200 Knoten) übersteigt die Masse benötigter DNA die Erdmasse [107, 101, 106]. Weitere Kritik am DNA-Computing bezieht sich auf die tatsächliche Rechenzeit. Während der Ligationsschritt in Sekundenschnelle ablief, stand Adleman insgesamt eine Woche im Labor, um das komplette Protokoll durchzuführen, da die nachfolgenden Arbeitsschritte sehr aufwendig von Hand durchgeführt werden mußten. Eine ähnliche Argumentation bezieht sich auf die Energieeffizienz. Arbeitsschritte wie PCR oder Gelelektrophorese benötigen sehr viel Energie [107], was die Sparsamkeit der Hybridisierung und Ligation zumindest zum Teil kompensiert. Außerdem sind der stochastische Charakter vieler Techniken und die damit verbundenen Fehlerraten ein Problem. Dies betrifft nicht nur Fehlhybridisierungen, sondern auch Mutationen beim Kopieren von DNA-Strängen mit Polymerase, eine nicht-hundertprozentige Effizienz von Ligase-Enzymen usw. Zusammengefaßt gesagt: „Reliability, efficiency, and scalability are perhaps the three most burning issues for molecular computing." (Max H. Garzon und Russell J. Deaton in [62])

Als Supercomputer zur Lösung NP-vollständiger Problem ist also auch ein DNA-Computer ungeeignet, zumindest mit dem Ansatz „Generiere zuerst alle Lösungskandidaten, filtere dann die Ungültigen heraus". Da aber einige auf DNA basierende Rechnermodelle zumindest theoretisch berechnungs-universell sind (z. B. [177]), bleibt die Hoffnung, sinnvolle Berechnungen mit DNA durchführen zu können, wenn vielleicht auch nur für bestimmte Anwendungsnischen [4, 3, 40]. Außerdem verspricht sich die wissenschaftliche Gemeinschaft durch den neuen Blickwinkel und die Abbildung von Wechselwirkungen zwischen Molekülen auf Informationsverarbeitung neue Erkenntnisse sowohl zur Natur von Berechnungen als auch zur Modellierung

molekularbiologischer Systeme [3, 21, 24]. Ein theoretisches Rechenmodell, das sich aus Überlegungen zum DNA-Computing ergeben hat, ist das Membrane-Computing, bei dem durch Membranen abgetrennte Zellbereiche Moleküle verarbeiten bzw. mit benachbarten Bereichen austauschen [126]. Andere Erkenntnisse, insbesondere auch zum Design von geeigneten DNA-Sequenzen, sind in die DNA-basierte Nanotechnologie eingeflossen (siehe weiter unten).

Im Folgenden sollen einige DNA-Computing-Modelle und -Experimente verschiedener Forschungsgruppen vorgestellt werden. Diese Aufzählung ist bei weitem nicht vollständig, sondern soll einen Überblick über das Spektrum dieses Forschungsgebiets geben und einige, besonders wichtige oder interessante Ergebnisse vorstellen. Nicht zuletzt sollen auch Beispiele für Anforderungen an das DNA-Sequenz-Design gegeben werden. Für alle Modelle und Anwendugen sind Spezifität sowie hohe und gleichmäßige Effizienz der Hybridisierungen wichtig, die meisten Kodierungen von Bitvektoren erfordern außerdem die Berücksichtigung von Konkatenationen einzelner Sequenzen für die Sequenzunähnlichkeit, bei manchen ergeben sich aber noch zusätzliche Anforderungen.

Hamilton-Pfad Adleman suchte einen Hamilton-Pfad durch einen Graphen mit 7 Knoten und 14 Kanten, in dem genau ein solcher Pfad existiert (Abb. 3.1a) [2]. Dazu kodierte er Knoten und Kanten in 20-mere. Nennt man die 10-mere, aus denen die Sequenzen für zwei Knoten v und w bestehen, v_l und v_r bzw. w_l und w_r, so besteht die Sequenz für die Kante $v \rightarrow w$ aus den Komplementärsequenzen zu w_l und v_r (Abb. 3.1b). Somit hybridisiert eine Kantensequenz überlappend mit den Sequenzen der beiden Knoten, die durch die Kante verbunden werden, über ein Länge von jeweils 10 Nukleotiden. Das Hamilton-Pfad-Problem wurde nun mit folgendem Protokoll gelöst:

- Herstellung aller Pfade: Alle Knoten- und Kantensequenzen wurden zusammengegossen, durch Hybridisierung der überlappenden Hälften bildeten sich Supramoleküle, die Abfolgen von Knoten und der dazwischenliegenden Kanten enthalten, also Pfade abbilden. Durch Ligation wurden die Lücken im Zucker-Backbone zwischen benachbarten Knoten- bzw. Kantensequenzen geschlossen. Durch eine genügend hohe Anzahl verwendeter Moleküle (hier ca. 10^{14}) konnte angenommen werden, daß jeder mögliche Pfad mindestens einmal im Reagenzglas vertreten war.

- Verwerfen aller Pfade, die nicht mit v_0 beginnen und bei v_6 enden: Mit PCR unter Verwendung der Sequenzen für v_0 und v_6 als Primer wurden Pfade zwischen diesen beiden Knoten amplifiziert.

- Verwerfen aller Pfade, die nicht genau sieben Knoten enthalten: Durch Gelelektrophorese konnte Adleman die Sequenzen isolieren, die genau 140 Nukleotide lang waren (7 Knotensequenzen jeweils der Länge 20).

- Verwerfen aller Pfade, die nicht alle Knoten besuchen: Für jeden Knoten v_i gab Adleman mikroskopisch kleine magnetische Kügelchen in die Lösung, an der die Komplementärsequenz des Knotens v_i befestigt war. Nach Hybridisierung konnte er so mit einem Magneten die Sequenzen extrahieren, die v_i enthielten. Nach siebenfacher Iteration für alle Knoten, wobei das Extrakt eines Schritts die Anfangsmenge für den nächsten Schritt bildete, blieben im letzten Extrakt genau die Sequenzen übrig, die alle Knoten enthalten.

- Auslesen des Ergebnisses: Eine erneute PCR mit anschließender Gelelektrophorese zeigte, ob noch Moleküle nach den Extraktionsschritten vorhanden waren. Falls ja, so erfüllen

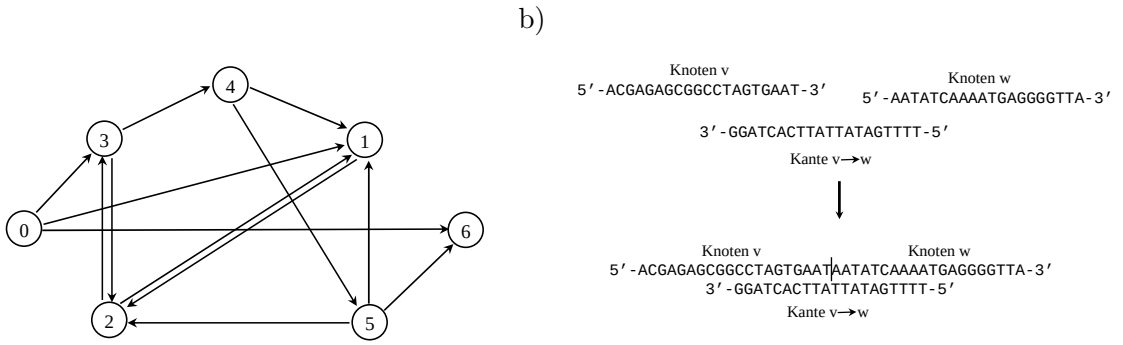


Abbildung 3.1: Probleminstanz und Pfadkodierung aus [2]. a) Von Adleman untersuchter Graph aus 7 Knoten und 14 Kanten. Der einzige Hamiltonpfad von v_0 nach v_6 folgt der Knotennumerierung. b) Kodierung von Knoten und Kanten in 20-mere. Jeweils 10 Basen einer Kantensequenz sind komplementär zu 10 Basen jeder der beiden inzidenten Knotensequenzen, so daß sich durch Hybridisierung der überlappenden Moleküle Ketten adjazenter Knoten bilden.

> diese die in den drei vorhergehenden Schritten überprüften Eigenschaften, kodieren also Hamiltonpfade.

Tatsächlich konnte Adleman DNA-Stränge der richtigen Länge in der Ergebnislösung nachweisen, verzichtete aber auf eine Überprüfung (z. B. durch Sequenzierung), ob diese den korrekten Pfad repräsentierten.

SAT Beim Erfüllungsproblem SAT ist ein Boolescher Term über Variablen x_0 bis x_{n-1} in Form einer Konjunktion von Klauseln gegeben. Jede Klausel ist wiederum eine Disjunktion von Literalen, also von Variablen und Negationen von Variablen. Bei der Problemvariante k-SAT besteht jede Klausel aus höchstens k Literalen. Ein Beispiel für einen solchen Term wäre also $(x_0 \vee \neg x_1) \wedge (x_1 \vee x_3 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$. Gefragt ist, ob es eine Belegung der Variablen x_0 bis x_{n-1} mit Werten aus {0,1} gibt, so daß der gegebene Term 1 ergibt.

Lipton wählte zur Lösung dieses Problems eine Kodierung für Bitvektoren, die später auch in Versuchen anderer Gruppen häufig zur Anwendung kam [102]. Hierzu konstruierte er einen gerichteten Graph, der abwechselnd Knoten für Bitpositionsinformationen und Knoten für die Bitbelegung (0 oder 1) enthielt (Abb. 3.2). Mit einer Abbildung von Pfaden auf DNA-Sequenzen, wie sie von Adleman vorgestellt wurde, können so Bitvektoren einer bestimmten Länge in DNA kodiert werden. Durch Hybridisierung und Ligation stellt man Bitvektoren mit zufälliger Verteilung von Nullen und Einsen her, eine Längensortierung per Gelelektrophorese erlaubt die Extraktion von Bitvektoren der Länge n. Auch hier sorgt eine genügend große Anzahl von Molekülen dafür, daß nahezu sicher alle 2^n möglichen Belegungen der Variablen x_0 bis x_{n-1} anschließend vorhanden sind.

Um zu überprüfen, ob eine dieser Belegungen den Term erfüllt, schlug Lipton ein Protokoll aus Extraktions- und Vereinigungsoperationen vor, das er allerdings nicht *in vitro* implementierte. Pro Klausel werden für jedes in dieser Klausel vorkommende Literal die Sequenzen extrahiert, deren zugehöriger Bitvektor dieses Literal erfüllt. Dies kann durch die von Adleman verwendeten magnetischen Kugeln realisiert werden, an die jeweils das Komplement der Sequenzen befestigt sind, die das entsprechende Literal kodieren. Für ein k-SAT-Problem ergeben sich so maximal k Reagenzgläser mit verschiedenen Extrakten. Durch Zusammengießen

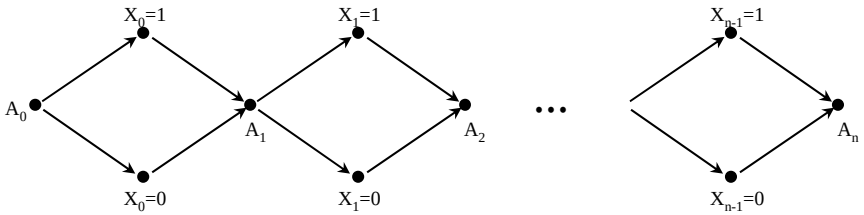


Abbildung 3.2: Bitvektorkodierung von Lipton [102]. Die mit A_i markierten Knoten können als Bitpositionen verwendet werden, die mit $x_i = b, b \in \{0, 1\}$ markierten Knoten stellen die Belegung der Variablen dar. Mit der Adlemanschen Abbildung von gerichteten Pfaden auf DNA-Stränge lassen sich also Bitvektoren in DNA kodieren.

der Inhalte dieser Gläser wird die Disjunktion innerhalb der Klausel implementiert. Iteration von Extraktion und Vereinigung über alle Klauseln, wobei jeweils aus der vereinigten Lösung des vorangegangenen Schritts extrahiert wird, stellt die Konjunktion der Klauseln dar. Ein anschließender Test auf vorhandene Moleküle beantwortet die Frage auf die Erfüllbarkeit des gegebenen Terms.

Sticker-Modell Eine andere Kodierung von Bitvektoren wird im Sticker-Modell verwendet [139]. Hier ist ein sogenannter Speicherstrang in n Bereiche aufgeteilt, wobei jeder Bereich ein Bit einer n-Bit-Zahl darstellt. Ist dieser Bereich einzelsträngig, gilt dieses Bit als auf 0 gesetzt, ist er doppelsträngig, hat das Bit den Wert 1. Während das Setzen eines Bits auf 1 sich einfach durch Hinzugabe und Hybridisierung des Komplements der entsprechenden Bitsubsequenz (des sog. Stickers) realisieren läßt, ist das Zurücksetzen auf 0 schwieriger. Schmelzen durch Temperaturerhöhung würde alle Sticker ablösen und somit alle Bits auf 0 setzen. Roweis et al. schlagen die Verwendung von *PNA strand invasion* vor. PNA ist eine Nukleinsäure mit einem Peptid- statt eines Zucker-Rückgrads. Ein zu einem DNA-Strang komplementäres PNA-Molekül bindet stabiler an dieses als komplementäre DNA und kann sogar einen bereits hybridisierten DNA-Strang verdrängen. Zu einem Sticker komplementäre PNA kann also diesen Sticker spezifisch vom Speicherstrang lösen.

Weitere SAT-Lösungen Trotz der von Anfang an (also seit 1994) bestehenden Skepsis gegenüber der Anwendung von DNA-Computing zur Lösung NP-vollständiger Probleme wurden eine ganze Reihe von Rechnermodellen und Protokollen genau hierfür entwickelt. Insbesondere das SAT-Problem erweist sich als besonders attraktive Benchmark-Anwendung.

Braich et al. schlagen eine Rechnerarchitektur vor, die dem Sticker-Modell ähnelt, verwenden aber die Lipton-Kodierung für Bitvektoren [31]. Damit konnten sie ein 3-SAT-Problem mit 20 Variablen und 24 Klauseln lösen. Dies ist die bisher größte *in vitro* gelöste Instanz eines NP-vollständigen Problems.

Oberflächenbasiertes DNA-Computing wurde von Liu et al. zur Lösung einer 3-SAT-Probleminstanz mit vier Variablen und vier Klauseln eingesetzt [104]. Sie kodierten die 2^4 verschiedenen Variablenbelegungen in 16 verschiedene DNA-Sequenzen, die als Sonden auf einem DNA-Chip befestigt wurden. Für jede Klausel wurden die Komplementärsequenzen der Belegungen auf den Chip gegeben und mit den entsprechenden Sonden hybridisiert, die die jeweilige Klausel erfüllen. Dann wurden einzelsträngige Sonden durch Endonukleasenverdau zerstört, anschließend wurden die Duplexe wieder geschmolzen und die Komplementärsequenzen vom Chip gewaschen. Iteration der Operationen Hybridisierung, Verdau, Schmelzen und

Waschen über alle Klauseln realisierten die Konjunktion. Auch hier stellen Moleküle, die nach allen Iterationen noch auf dem Chip vorhanden sind, die Lösungen, also erfüllende Variablenbelegungen, dar, und können z. B. mit PCR nachgewiesen und ausgelesen werden. Das Sequenzdesign würde sich eigentlich auf die Suche nach einzelnen, spezifisch bindenden Sequenzen beschränken, in [104] wurden die Sequenzen, die die Variablenbelegungen kodieren, aber beidseitig von fixen Sequenzen flankiert, die als Priming Sites für die PCR dienten. Damit muß hier auch die Konkatenation mit diesen fixen Teilsequenzen beachtet werden.

Zwar hängt die Anzahl der benötigten Rechenschritte bei diesem Protokoll linear von der Anzahl der Klauseln und Variablen pro Klausel ab, diese Rechnung ist jedoch leicht geschummelt, da der Hybridisierungsschritt tatsächlich exponentielle Bearbeitungszeit benötigt. Sollen für eine Klausel alle Belegungen auf den Chip gegeben werden, die eine in dieser Klausel vorkommende Variable mit einem bestimmten Wert belegen, so sind alle anderen Variablen frei belegbar, es müssen für dieses eine Literal also 2^{n-1} verschiedene Belegungssequenzen ausgewählt werden. Für alle Literale einer Klausel vervielfacht sich diese Anzahl (z. B. auf $1.75 \cdot 2^{n-1}$ für 3-SAT [179]). Außerdem beinhaltet der Vorgang dieser Auswahl, die der Anwender von Hand vornehmen muß, bereits einen Großteil der auszuführenden Rechnung, der DNA-Chip dient weniger als Computer sondern eher als molekularer Notizzettel.

Abgesehen von dieser Schwäche des von Liu et al. vorgeschlagenen Protokolls ist der oberflächenbasierte Ansatz an sich aber interessant, da er ein höheres Maß an Kontrolle bietet als der Ansatz im Reagenzglas. Die Kandidaten für eine Antwort auf das gestellte Problem befinden sich immer auf dem Chip und sind daher leichter von anderen, während der Rechnung temporär benötigten Molekülen leichter zu unterscheiden als in Lösung in einem Reagenzglas.

Wu verbesserte das oberflächenbasierte Protokoll, indem er eine Lipton-ähnliche Kodierung der Bitvektoren ohne die Bitpositionen vorschlug [179]. Statt der Zerstörung nicht-erfüllender Belegungen durch Verdau sollen die Komplementärsequenzen, die mit den erfüllenden Belegungen hybridisieren, mit Fluoreszenzfarbstoff markiert und die Fluoreszenz doppelsträngiger Moleküle für jede Klausel auf einen Film aufgenommen werden. Übereinanderlegen aller belichteten Filme ergäbe so für den Spot mit den Belegungen, die alle Klausel erfüllen, einen transparenten Fleck, da alle Filme dort transparent seien. Für ein 3-SAT-Problem müßten so pro Klausel nur drei Sequenzen ausgewählt werden. Allerdings beschränkt sich der Autor auf den Vorschlag, über eine Realisierung wird nicht berichtet.

Sakamoto et al. schlagen in [143] ebenfalls ein Protokoll zur Lösung von SAT-Problemen vor. Allerdings wird hierbei zunächst der gegebene Term ausmultipliziert, so daß eine Disjunktion von Monomen (Konjunktionen von Literalen) vorliegt. Das Ausmultiplizieren wird bereits *in vitro* durch Ligation der entsprechenden Literalsequenzen durchgeführt. Für jedes Monom wird hierbei durch Ligation eine Sequenz gebildet, die eine Konkatenation der im Monom enthaltenen Literale darstellt. Hierbei ist die Sequenz für ein Literal komplementär zur Sequenz für die Negation dieses Literals. Der gegebene Term ist genau dann erfüllt, wenn ein Monom existiert, in dem kein Literal zusammmen mit seiner Negation vorkommt. Sequenzen, die Monome darstellen, in denen Literale und deren Negation vorkommen, enthalten also zueinander komplementäre Teilsequenzen, die eine Hairpin-Loop bilden. Sortiert man alle Hairpin-Loops aus, so bleiben die Moleküle zurück, die Variablenbelegungen darstellen, die alle Klauseln erfüllen.

SATte Springer Das Springerproblem besteht darin, möglichst viele Springer so auf einem Schachbrett anzuordnen, daß sich keine zwei Springer gegenseitig bedrohen. Faulhammer et al. formulierten die sich ausschließenden Belegungen von Feldern mit Springern als Boolesche Formel und bildeten so daß Springerproblem auf das SAT-Problem ab [52]. Sie beschränkten

die Probleminstanz auf ein Brett mit 3×3 Feldern und kodierten die Springerverteilung mit 9-Bit-Vektoren in Lipton-Kodierung, wobei eine 1 im i-ten Bit für einen Springer auf Feld i steht. Auch hier wurden Lösungskandidaten durch iteriertes Markieren und Verdauen von Belegungen, die Klauseln nicht erfüllen, nach und nach eliminiert. Es blieben 31 Stellungen mit einem bis fünf Springern übrig, wovon eine Stellung illegal war. Diese war durch Punktmutation und Auslassen einer Base beim Kopieren durch PCR entstanden.

Travelling Salesman Die Lösung von Minimierungsproblemen gestaltet sich mit den bisher vorgeschlagenen Rechnermodellen deutlich schwieriger als die Lösung von Entscheidungsproblemen wie SAT. Zu beobachten ist dies z. B. bei der Erweiterung der Adleman-Kodierung für Graphen mit gewichteten Kanten, wie sie von Shin et al. zur Lösung des Travelling-Salesman-Problems vorgeschlagen wird [154]. Hierbei wird ein Pfad durch einen gerichteten Graphen gesucht, der jeden Knoten genau einmal besucht und eine minimale Summe von Kantengewichten hat. Die Kantensequenzen sind hier länger als bei Adleman. Zwischen die Teilsequenzen, die zu den Sequenzen der inzidenten Knoten komplementär sind, wird eine Gewichtsequenz eingefügt. Je kleiner das Gewicht der Kante ist, desto höher ist dafür der GC-Gehalt der Gewichtssequenz. Dies führt dazu, daß leichte Kanten eher und stabiler hybridisieren und somit eher Pfade mit kleinen Gesamtgewichten ligiert werden. Das Funktionieren dieser Idee wurde nicht *in vitro*, sondern durch eine Computersimulation gezeigt. Eine Schwäche dieses Protokolls ist die sehr grobe Schätzung der Hybridisierungsneigung durch den GC-Gehalt. Dies ließe sich aber im Sequenzdesign problemlos durch ein realistischeres Maß wie die freie Enthalpie ersetzen. Ein größeres Problem stellt die endliche Auflösung der darstellbaren Gewichte dar. Die Anzahl verschiedener GC-Gehalte ist durch die Sequenzlänge beschränkt. Aber auch die Anzahl unterschiedlicher (erst recht die deutlich unterschiedlicher) freier Enthalpien ist nach dem nearest-Neighbor-Modell stark beschränkt. Schließlich bleibt noch das Problem, daß suboptimale Lösungen nicht aussortiert werden, sondern sich nur die Wahrscheinlichkeit zu ihrer Erzeugung im Hybridisierungs-/Ligationsschritt verringert. Dadurch dürften in der resultierenden Lösung viele suboptimale Kandidaten vorliegen und ein gezieltes Auslesen des optimalen Pfads erschweren.

Boolesche Algebra Für universelle Anwendungen eines DNA-Computers möchte man nicht nur Entscheidungs- und Optimierungsprobleme lösen, sondern auch einfach nur rechnen, und zwar sowohl mit der Booleschen als auch mit der Arithmetischen Algebra.

Su und Smith entwickelten eine DNA-Implementierung der Booleschen Funktion NOR, die allein eine Basis der Booleschen Algebra bildet, d. h. auch alle anderen Booleschen Funktionen sind nur durch Verwendung von NOR realisierbar [158]. Dafür wurde ein oberflächenbasierter Ansatz mit einer Bitvektor-Kodierung, wie sie von Wu et al. (s. o.) verwendet wurde, eingesetzt. Das Protokoll besteht aus der Markierung von Kandidatensequenzen durch Hybridisierung mit Komplementen der vom auszuwertenden Term abhängigen Literalsequenzen, Ausweiten der Doppelsträngigkeit durch Polymerase bis zum freiliegenden (nicht an der Oberfläche befestigten) Ende, und Ligation eines doppelsträngigen Ergebnismoleküls an die markierten Kandidaten.

Eine Berechnung durch das Self-Assembly von DNA-„Kacheln“ schlagen Mao et al. aus der Gruppe von Nadrian Seeman vor [109, 110]. Diese Kacheln, sogenannte Triple-Crossover-Moleküle (TX-Kacheln) sind selbst bereits supramolekulare Konstrukte aus 3 DNA-Helices, die durch mehrere Crossover-Punkte, an denen sie Einzelstränge austauschen, verbunden sind (Abb. 3.4). Desweiteren sind sie mit vier sticky Ends versehen, die das Self-Assembly ermögli-

chen. Die Helixlängen der TX-Kacheln sind so gewählt, daß sich über die Länge einer ganzen Kachel eine ganzzahlige Anzahl von Windungen ergibt, so daß die Kacheln nach dem Self-Assembly in einer Ebene liegen. TX-Kacheln bieten viele Anwendungsmöglichkeiten, auch in der Nanotechnologie (s. u.), hier kodieren sie Bits eines Bitvektoren so, daß durch das Self-Assembly das kumulative XOR des Vektors berechnet wird. Aus derselben Gruppe stammen TX-Kacheln, die sich zu einem Äquivalent von NAND-Gatter-Schaltkreisen zusammenfügen [35]. Zusätzliche sticky Ends, die aus der Ebene der TX-Kacheln herausragen, erlauben die Anlagerung von weiteren Kacheln auf einer zweiten Ebene darüber, die jeweils Ein- bzw. Ausgaben der NAND-Gatter repräsentieren.

Eine andere Simulation von NAND-Gattern beruht auf DNA-Molekülen, die sich frei in Lösung befinden [12]. Dabei benötigt man für die Berechnung eines NAND-Gatters aber neben der Hybridisierung noch das Schneiden mit Restriktionsenzymen und die Sortierung der Moleküle nach ihrer Länge durch Gelelektrophorese.

Arithmetische Berechnungen Eine Addition von Bitvektoren modellierten Guarnieri et al. mit DNA [67]. Leider ist hierbei die Kodierung der Bits in Moleküle sehr aufwendig, zudem haben die Ergebnismoleküle eine völlig andere Form als die Eingabemoleküle, was die Hintereinanderausführung mehrerer Additionen verhindert.

Oliver schlägt in [124] eine Abbildung der Multiplikation Boolescher Matrizen auf Graphen vor. Eine erste Schicht von Knoten repräsentiert dabei die Zeilenindizes der ersten Matrix, eine zweite Schicht die Spaltenindizes der ersten und die Zeilenindizes der zweiten Matrix, eine dritte Schicht schließlich die Spaltenindizes der zweiten Matrix. Gerichtete Kanten verlaufen nur von Knoten einer Schicht zu Knoten der nächsten Schicht. Eine Kante führt von Knoten i der ersten Schicht zu Knoten j der zweiten Schicht genau dann, wenn in der ersten Matrix in der i-ten Zeile und j-ten Spalte eine Eins steht. Entsprechendes gilt für Kanten von der zweiten zur dritten Schicht und Einsen in der zweiten Matrix. In der Ergebnismatrix steht damit genau dann eine Eins an Position (i, j), wenn es einen Pfad von Knoten i in der ersten zu Knoten j in der dritten Schicht gibt. Die DNA-Moleküle repräsentieren hier die Kanten, die zu den Eingabematrizen gehören. Durch Self-Assembly verbinden sich diese zu Pfaden, die die Ergebnismatrix definieren.

Steganographie DNA eignet sich alleine schon wegen seiner geringen Größe ideal für die Steganographie, also die heimliche Übermittlung von Botschaften, die man in einer größeren Menge irrelevanter Informationen versteckt. Eine einfache Kodierung von Texten in DNA weist jedem Zeichen einer Menge von bis zu 64 Buchstaben, Ziffern und Satzzeichen ein drei Basen langes Codon[1] zu [38, 134]. Ein Text wird so zeichenweise in eine Aneinanderreihung entsprechender Codons übertragen. An beide Enden der so entstandenen Sequenz werden Primer-Bindungsstellen angefügt. Die DNA-Moleküle wurden synthetisiert und unter genomischer DNA von Ratten oder Menschen versteckt. Nur ein Empfänger, der die Primer-Sequenzen, die hier als Schlüssel dienen, kennt, kann den versteckten Text per PCR vom Hintergrund der genomischen DNA unterscheiden und auslesen. In einer zweiten Stufe der Steganographie wurde die DNA auf einen Punkt am Ende eines Satzes auf Papier aufgebracht, dieser Brief mit der Post verschickt, die DNA wieder extrahiert und die geheime Nachricht (Ort und Zeitpunkt des Angriffs auf die Normandie) ausgelesen.

[1] in Anlehnung an die dreibasigen Codons in Genen, die jeweils eine Aminosäure des durch das Gen kodierten Proteins definieren

Da die Drei-Basen-Kodierung sehr anfällig für Mutationen während der PCR-Amplifizierung ist [134], und die Nachrichtmoleküle sich vielleicht doch durch strukturelle Merkmale von genomischer DNA unterscheiden und sich so durch statistische Verfahren erkennen lassen, wählten Leier et al. eine Kodierung der Zeichen als 8-Bit-Binärvektoren, mit je einer Sequenz für 0 und 1, die aneinandergefügt wurden [96] (für Details siehe Abschnitt 8.2). Auch hier ist der Schlüssel die Kenntnis der Priming Sites an den Enden der Sequenz, die unter vielen zufälligen 8-Bit-Sequenzen mit anderen Primerbindungsstellen versteckt werden. Nachrichtsequenz und Hintergrund sind somit strukturell ähnlicher, die Verwendung von 20 Basen pro Bit bot genügend Redundanz, um weniger anfällig gegenüber Mutationen zu sein.

Evolutionärer Algorithmus Bäck et al. schlagen die Nutzung gerade solcher Mutationen bei der Vervielfältigung von Sequenzen durch PCR als Variationsoperator für einen *in vitro* ablaufenden Evolutionären Algorithmus vor [19] (für eine kurze Beschreibung von Evolutionären Algorithmen siehe auch Abschnitt 5.1). Die Selektion soll über ähnliche Filteroperationen wie von Adleman verwendet realisiert werden, die Fitness-Bewertung kann z. B. längenabhängig mit Gelelektrophorese durchgeführt werden.

Automaten Einen kompletten Spielautomaten für das Spiel Tic-Tac-Toe konstruierten Stojanovic und Stefanovic [157]. Dazu formulierten sie die perfekte Strategie, mit der man nicht verlieren kann, sondern immer mindestens ein Unentschieden erreicht, in Boolesche Terme. Jeder Term gibt für eines der neun Felder an, ob dieses in Abhängigkeit von den bisherigen Spielzügen des Gegners besetzt werden soll. Für die Auswertung dieser Terme wurden Gatter-Moleküle, die aus der Kombination mehrerer Hairpin-Loops bestehen, entworfen. Andere DNA-Moleküle, die eine Kombination bisheriger Eingaben repräsentieren, die einen solchen Term erfüllt, öffnen durch Hybridisierung die Hairpin Loops, wodurch in Kombination mit einem weiteren Reporter-Molekül ein Leuchtsignal ausgelöst wird. Physikalisch besteht der Automat aus neun Vertiefungen (die Felder), in denen die Gatter-Moleküle des zu dem jeweiligen Feld gehörigen Booleschen Terms in Lösung enthalten sind. Der menschliche Gegner gibt sein in DNA-Stränge kodiertes zu markierendes Feld in alle neuen Vertiefungen, eine davon leuchtet als Antwort auf und markiert so den Gegenzug des Automaten.

Eine Konstruktion beliebiger endlicher Automaten mit zwei Zuständen und für ein Eingabealphabet mit zwei Buchstaben stellten Benenson et al. vor [29, 27]. Dabei ist der Eingabestring als doppelsträngiges Molekül kodiert, in dem die Sequenzen für die Zeichen des Strings aneinander gereiht sind. Ein sticky End kodiert sowohl das nächste Zeichen als auch den aktuellen Zustand des Automaten. Ein entsprechendes doppelsträngiges Zustandsübergangsmolekül kann an diesem sticky End hybridisieren. Es enthält die Erkennungssequenz für das Restriktionsenzym FokI sowie eine Sequenz von Basen, die als Abstandhalter zwischen der Erkennungssequenz und dem sticky End dienen, und deren Anzahl vom jeweiligen Zustandsübergang abhängt. FokI schneidet doppelsträngige DNA nicht in der Erkennungssequenz, sondern dazu versetzt, so daß das Eingabemolekül abhängig von der Anzahl der Abstandhalterbasen an einer neuen Stelle geschnitten wird und das neue sticky End den neuen Zustand kodiert. Die Autoren haben als Anwendung die Beeinflussung von Genexpression vorgeschlagen [28]

Neuronales Netz Mills hat eine DNA-Simulation eines Perzeptrons, also eines einfachen Künstlichen Neuronalen Netzes mit zwei Schichten, vorgestellt [113]. Die Kodierung von Neuronen und Axonen entspricht in etwa der von Knoten und Kanten eines Graphen bei Adleman. Hybridisierung eines Neuronstrangs und eines Axonstrangs führt zu Verlängerung durch Po-

lymerase und anschließende Freisetzung des Oligomers, das das Ausgabeneuron repräsentiert. Aktivierungsstärken und Axongewichte werden auf die Konzentrationen der entsprechenden DNA-Moleküle abgebildet. Als Anwendung wird die Diagnose aufgrund von Genexpression vorgeschlagen, leider gibt der Autor keine Methode zum Training des Netzes an.

3.2.2 Nanotechnologie

Die Nanotechnologie beschäftigt sich mit dem Entwurf von Konstruktionen im Nanometerbereich. Viele top-down-Konstruktionsmethoden, bei denen Werkzeuge verwendet werden, die größer sind als die herzustellenden Strukturen, sind in diesem Bereich unbrauchbar. Probleme sind z. B. *thick fingers* (die Werkzeuge sind zu groß, um Nanostrukturen präzise herstellen zu können; die Herstellung von Nanowerkzeugen ist selbst ein nanotechnologisches Problem) und *sticky fingers* (bei der Manipulation einzelner Atome oder Moleküle machen sich Van-der-Waals-Kräfte zwischen Werkzeug und Material bemerkbar, die im makroskopischen Bereich irrelevant sind).

Eine vielversprechende Alternative bieten daher bottom-up-Ansätze, bei denen sich die Bauteile selbsttätig oder mit Hilfe anderer nanoskopischer Moleküle zusammensetzen. Dank der guten Erkennungseigenschaften ist DNA für dieses Self-Assembly sehr gut geeignet [175, 150, 114, 120].

DNA-Biotin-Streptavidin-Konjugate Außer mit dem jeweiligen Watson-Crick-Komplement, also mit einem anderen Nukleinsäuremolekül, kann DNA auch mit anderen Molekülen wie z. B. Biotin verbunden werden. Biotin-modifizierte DNA wiederum kann ein Konjugat mit Streptavidin bilden, das Bindungsstellen für drei weitere Biotin-Moleküle anbietet, die mit einer Vielzahl von Molekülen funktionalisiert werden können. Solche Funktionalisierungen und deren Anwendungen werden in der Gruppe von Christof Niemeyer untersucht [123, 122]. Das DNA-Self-Assembly kann dann genutzt werden, um z. B. die Biotin-Streptavidin-Konjugate auf einem langen DNA-Strang aufzureihen, der aus Subsequenzen besteht, die jeweils komplementär zu den an den Konjugaten befestigten DNA-Strängen sind [121].

DDI Eine weitere Anwendung des DNA-Self-Assembly ist die ebenfalls von der Niemeyer-Gruppe entwickelte *DNA-Directed Immobilization* (DDI), bei der auf einem Microarray die Oligomere befestigt sind, die zu denen der DNA-Biotin-Streptavidin-Konjugate komplementär sind, so daß die Streptavidin-Moleküle sowie ggf. daran befestigte andere Moleküle auf dem Träger positioniert werden (Abb. 3.3). Vorteile dieser Methode gegenüber anderen Immobilisierungsverfahren sind hohe Effizienz, milde Reaktionsbedingungen, die eine hohe Aktivität der befestigten Proteine erlauben, Reversibilität der Immobilisierung sowie ein relativ geringer experimenteller Aufwand [172]. An das Streptavidin lassen sich z. B. Antikörper binden, mit denen sich dann die entsprechenden Antigene in einem sog. Immunoassay nachweisen lassen [171].

Protein-Nachweis Eine andere Form des Protein-Arrays beruht auf dem Self-Assembly von vierarmigen DNA-Junctions (Abb. 5.2) [182]. Die hier verwendeten Junctions haben im Unterschied zu den abgebildeten dort, wo ein Strang von einem Arm auf den nächsten wechselt, noch eine Sequenz aus vier Thymin-Basen, die so einen nicht-hybridisierenden Bereich in der Mitte der Junction bilden. In diesem Bereich lassen sich ebenfalls Biotin-Streptavidin-Konjugate befestigen. Je nach Entwurf der sticky Ends der Junction-Moleküle lassen sich lineare Bänder konstanter Breite oder flächige Gitter per Self-Assembly konstruieren.

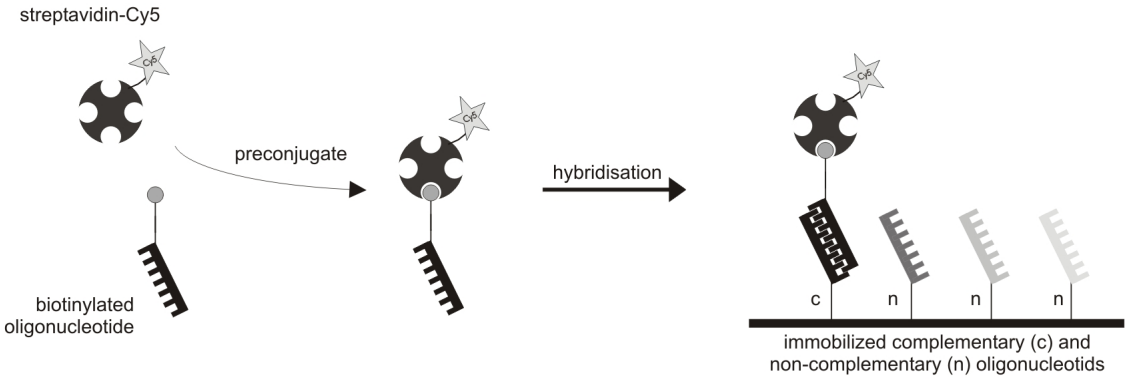


Abbildung 3.3: Schema der DNA-Directed Immobilization (DDI) (aus [55]). Mit dem Fluoreszenzfarbstoff Cy5 versehene Streptavidin-Moleküle und Biotin-modifizierte DNA-Oligomere bilden ein Konjugat, das sich durch Hybridisierung mit auf einem Träger befestigten Komplementär-Strängen auf dem Träger positionieren und durch Fluoreszenz nachweisen läßt.

Nam et al. setzen zum Nachweis von Proteinen Magnetische Mikropartikelsonden (MMP) und Nanopartikelsonden (NP) ein [117]. Die MMP sind mit Antikörpern bestückt, die NP sowohl mit Antikörpern als auch mit doppelsträngigen DNA-Oligomeren, die als „Bar-Code“ für das nachzuweisende Protein dienen. Liegt das Protein in der untersuchten Flüssigkeit, z. B. Blut, vor, so bindet es an den Antikörpern eines MMP. Die an den NP befestigten Antikörper binden ihrerseits an das Protein. Mit einem Magneten läßt sich dann der MMP-Protein-NP-Komplex extrahieren, Schmelzen der Oligomere setzt dann die Stränge der Bar-Codes frei.

Strukturen Die Arbeitsgruppe von Nadrian Seeman beschäftigt sich mit der Konstruktion verschiedenster DNA-Strukturen. Darunter fallen die bereits erwähnten Junctions (Abb. 5.2), aus Junctions zusammengefügte Gitter, ein Würfel mit Kanten aus DNA-Strängen, ein angeschnittenes Oktaeder, Double- und Triple-Crossover-Kacheln (DX- und TX-Kacheln, s. Abb. 3.4), Knoten und Borromäische Ringe [149, 94]. Andere Gruppen berichten z. B. über die Konstruktion eines Oktaeders [152], eines Tetraeders [66] und von Dendrimeren, in denen Y-förmige Junctions zu regelmäßigen Formen verbunden sind [100]. Aus den Double- und Triple-Crossover-Kacheln lassen sich DNA-Gerüste bilden, die z. B. über Biotin-Streptavidin-Konjugate Gold-Nanopartikel regelmäßig anordnen [95, 99]. Die Gruppe von Günter von Kiedrowski entwickelte Supramoleküle aus drei DNA-Oligomeren und einem chemischen Linkermolekül, das diese Oligomere zu einem ψ-förmigen Komplex (Trisoligos) verbindet [146]. Es lassen sich Kopien dieser Trisoligos mit gleicher Konnektivität herstellen [50] und Tetraeder durch Self-Assembly entsprechend entworfener Trisoligos konstruieren [170].

DNA läßt sich auch als Klebstoff verwenden, um Nanopartikel zu Aggregaten zu verbinden. Dazu befestigt man an der einen Hälfte der Nanopartikel bestimmte DNA-Oligomere, an der anderen Hälfte deren Komplemente. Self-Assembly führt dann zur Bildung der Aggregate [115]. Hazarika et al. entwickelten diesen Ansatz zu einer reversiblen Aggregatbildung unter Ausnutzung des *strand replacement* weiter [72]. Hierzu werden an den Nanopartikeln zwei Sorten von

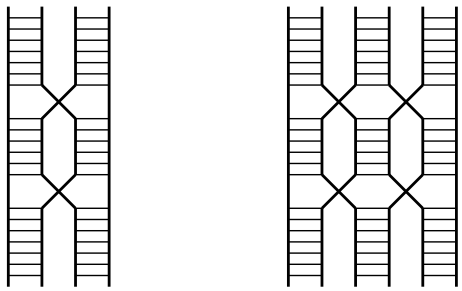


Abbildung 3.4: Schematische Darstellung einer Double-Crossover- (DX, links, [148]) und einer Triple-Crossover-Kachel (TX, rechts, [94]). Zwei bzw. drei Doppelstränge sind durch Crossover-Stellen, an denen sie Einzelstränge austauschen, miteinander verbunden. In der Abbildung stellen die stärkeren Linien die Backbones und die dünneren Linien die Wasserstoffbrücken der Basenpaare dar. Eine kompliziertere und realistischere Darstellung, die auch die Helizes zeigt, findet sich in den zitierten Werken. Es gibt auch komplexer aufgebaute DX- und TX-Kacheln, bei denen nicht alle Einzelstränge wie hier von oben nach unten durch laufen, sondern z. B. am Crossover-Punkt eine 180°-Wende machen.

DNA-Oligomeren A und B befestigt. Eine dritte Sorte von DNA-Strängen C besteht aus der Konkatenation der Komplemente von A und B, sowie einer weiteren kurzen Subsequenz D. Der $\overline{AB}$-Teil von C dient als Klebestreifen, der die Nanopartikel durch Hybridisierung mit A und B verbindet. Gibt man nun $\overline{C}$, den Komplementärstrang zu C, hinzu, so kann dieser zunächst an D hybridisieren. Da ein Duplex aus C und $\overline{C}$ wegen der zusätzlichen Basenpaare energetisch günstiger ist als eine Verbindung nur von A, B und C, und das Branching, also die gleichzeitige Bindung zweier Stränge an C, zusätzlich destabilisierend wirkt, verdrängt $\overline{C}$ A und B und bildet den Duplex mit C. Dadurch werden die Nanopartikel wieder voneinander getrennt.

Nanowires Eine Anwendungsidee ist die Verwendung von DNA als elektrischer Leiter im Nanometerbereich (sog. *Nanowires*). Über die Leitfähigkeit herrscht in der Literatur allerdings große Uneinigkeit. Dekker und Ratner geben in [46] einen Überblick über Experimente verschiedener Forschungsgruppen zu diesem Thema. Diesen Experimenten zufolge ist DNA sowohl Isolator als auch Halbleiter, Leiter und sogar Supraleiter. Diese erheblichen Unterschiede in den experimentellen Resultaten liegen vermutlich in verschiedenen Einflüssen durch Länge und Zusammensetzung der DNA-Moleküle, Eigenschaften der Pufferlösung oder sonstiger Versuchsumgebung oder die Befestigungsart an den Elektroden begründet. Nach Meinung der Autoren sind DNA-Moleküle nur im Bereich von wenigen Nanometern als Leiter zu verwenden. Über so kurze Distanzen können das Tunneln von Elektronen und das sog. *thermal hopping* für Ladungstransport sorgen. Längere DNA-Stränge sind aber als Isolatoren zu betrachten.

Vielversprechender im Hinblick auf die Herstellung von Nanowires ist das Bestücken der DNA-Moleküle mit Metall. Braun et al. befestigten zwei Oligomere an zwei Elektroden [32]. Anschließend ließen sie einen 16 μm langen DNA-Strang, dessen sticky Ends zu diesen Oligomeren komplemenär waren, mit diesen hybridisieren. Schließlich wurden Silberionen an der DNA angelagert, bis ein Nanodraht von 30-50 nm Durchmesser entstanden war, der ein nichtlineares Leitverhalten zeigte. Auch die aus vierarmigen Junctions konstruierten DNA-Bänder (s. o.) können mit Silber metallisiert werden und ergeben so Leiter mit Ohmschem Verhalten [182]. An mit Silberionen bestückte DNA läßt sich auch Gold anlagern. Keren et al. erzeugten

so Ohmsche Nanodrähte aus Gold [89]. Durch Einlagerung von RecA-Proteinen in Abschnitte der DNA ließen sich diese Bereiche vor der Silberanlagerung schützen, wodurch sie nicht-leitend blieben. Einen Kupferdraht basierend auf einem DNA-Strang aus künstlichen Basen stellten Tanaka et al. her [163]. Liu et al. haben ein Band aus zwei Sorten von Triple-Crossover-Kacheln hergestellt, das sich zu einer Nanoröhre zusammenrollte [103]. Durch Metallisierung mit Silber wurde auch aus ihr ein Ohmscher Leiter gemacht, mit einer höheren Leitfähigkeit als die des Silberdrahts von Braun et al.

Carbon-Nanotubes Carbon-Nanotubes (CNT) sind Röhren mit Durchmessern im Nanometerbereich, die aus einem zusammengerollten Gitter aus Kohlenstoffatomen bestehen. Je nach Art des Zusammenrollens haben CNT Leiter- oder Halbleitereigenschaften, und sind daher ein interessanter Baustoff für die Nanoelektronik. Keren et al. verwendeten das oben erwähnte DNA-RecA-Filament, um einen Feldeffekt-Transistor herzustellen [88]. Hierzu wurde eine CNT mit einem Anti-RecA-Streptavidin-Komplex am Filament befestigt, parallel zum DNA-Strang verlaufend. Nach der Metallisierung dienten die mit Gold versehenen DNA-Abschnitte als Source und Drain, das Silizium-Substrat, auf das das Konstrukt aufgebracht wurde, diente als Gate. Eine alternative Befestigungsmethode von CNT an DNA mit Hilfe von PNA, einer Nukleinsäure mit Peptid-Backbone, demonstrierten Williams et al. [176]. Zheng et al. zeigten, daß sich DNA ohne Hilfsmoleküle zur Verbindung mit CNT spiralförmig um eine CNT winden kann [185].

DNA-Maschinen Durch Änderung ihrer Konformation kann DNA auch mechanische Arbeit verrichten, erlaubt also die Konstruktion von Nanomaschinen. Ein einfaches Beispiel für eine solche Konformationsänderung ist der B-Z-Übergang. Bei der normalerweise vorkommenden B-Form der DNA ist die Doppelhelix rechtsherum gewunden. Unter hohen Ionenkonzentrationen kann ein DNA-Strang, der abwechselnd aus Purin- und Pyrimidinbasen besteht, die linksherum gewundene Z-Form einnehmen. Mao et al. haben mit einer solchen DNA ($\mathtt{CG}_{10}$) zwei DX-Kacheln verbunden und konnten diese durch den Übergang von der B- zur Z-Form um dreieinhalb Umdrehungen gegeneinander drehen [111]. Eine andere Form des reversiblen Übergangs zwischen zwei Molekülkonformationen demonstrierten Yan et al. mit Paranemic-Crossover- und Juxtapose-Crossover-Kacheln (PX und JX) [183]. Diese bestehen ähnlich wie DX-Kacheln aus zwei Strängen, die über drei (PX) bzw. zwei (JX) Crossover-Stellen miteinander verbunden sind. Durch Hinzugeben von DNA-Molekülen, die komplementär zu den Strängen sind, die im PX-Molekül das mittlere Crossover bildet, werden diese Stränge per Strand-Replacement (s. o.) entfernt. Ersetzt man sie durch Stränge, die an den einzelsträngig gewordenen Bereichen hybridisieren, ohne einen Crossover-Punkt zu bilden, so erhält man eine JX-Kachel. Durch das Fehlen dieses Crossovers ist die untere Hälfte der Kachel gegenüber der oberen Hälfte um 180° verdreht, um eine Achse parallel zu den Helixachsen. Auch diese neuen Stränge lassen sich durch Strand-Replacement wieder entfernen, wodurch die Reversibilität gegeben ist.

Ebenfalls mit Hilfe von Strand-Replacement betrieben Yurke et al. eine DNA-Pinzette [184]. Die eigentliche Pinzette ist ein doppelsträngiges DNA-Molekül mit sticky Ends an beiden Enden (Abb. 3.5). Ein Strang, der sowohl zu den sticky-Ends komplementäre Bereiche als auch einen weiteren Überhang hat, schließt die Pinzette durch Hybridisierung. Strand-Replacement mit dem Komplement dieses Strangs, das am Überhang angreifen kann, löst den Strang von den sticky Ends und öffnet die Pinzette wieder. Mit einer ähnlichen Technik, aber einer leicht abgewandelten Architektur konstruierte dieselbe Gruppe auch ein scherenförmiges Molekül

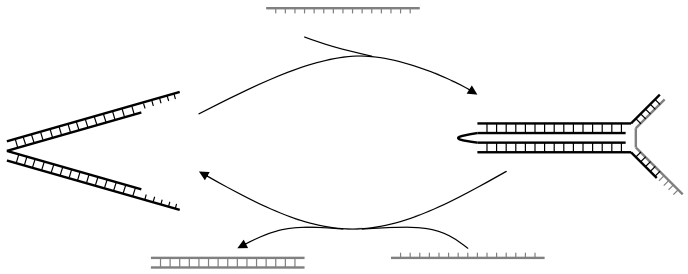


Abbildung 3.5: Schematische Darstellung der DNA-Pinzette aus [184]. In der Abbildung stellen die stärkeren Linien die Backbones und die dünneren Linien die Wasserstoffbrücken der Basenpaare dar. Links ist die geöffnete Konformation dargestellt. Durch Hybridisierung der sticky Ends mit einem komplementären Einzelstrang wird die Pinzette geschlossen (rechts). Ein Überhang dieses Stranges erlaubt Strand-Replacement durch Hybridisierung mit seinem vollständigen Komplement, wodurch die Pinzette wieder geöffnet wird.

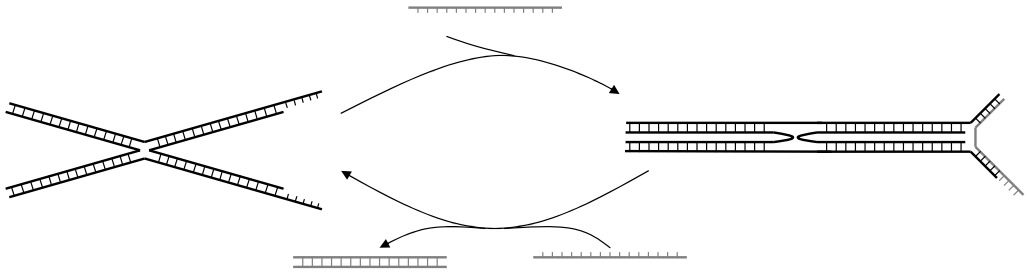


Abbildung 3.6: Schematische Darstellung der DNA-Schere aus [116]. Die Darstellung entspricht der der DNA-Pinzette in Abbildung 3.5. Hier wurde an die „Scharnierstelle“ der Pinzette ein nahezu spiegelgleiches Molekül angehängt, allerdings ohne sticky Ends. Das Schließen bzw. Öffnen der rechten Pinzette führt durch mechanische Starrheit der Helices ebenfalls zu einem Schließen bzw. Öffnen der linken Pinzette.

[116] (Abb. 3.6).

Reif stellt in [133] wandernde und rollende DNA-Moleküle vor, die eine bidirektionale Bewegung auf einem bestimmten DNA-„Untergrund“ ausführen können, leider ist aber bisher nur eine zufällige und keine orientierte Bewegung möglich.

Guanin-reiche DNA-Stränge können unter bestimmten Bedingung anstatt einer Duplex- eine Quadruplex-Struktur bilden, also eine Verbindung aus vier Strängen eingehen. Alberti et al. nutzten dies zum Bau einer Verkürzungs-Ausdehnungs-Maschine [5]. Sie verwendeten ein Guanin-reiches Molekül, das sich als Einzelstrang zu einer kompakten Quadruplex-Struktur faltet. Hybridisierung mit dem Komplement ergibt einen gestreckten Duplex. Durch Strand-Replacement ist auch dieser Vorgang reversibel. Eine medizinische Anwendung dieser Konstruktion als Protein-Shuttle schlagen Dittmer, Reuter und Simmel vor [48]. Sie konnten in der Quadruplex-Struktur ein Thrombin-Molekül binden, das beim Übergang in die Duplex-Form freigesetzt wurde.

Markierung Als Informationsträger in Nanometer-Größe eignet sich DNA auch zur Markierung verschiedener Materialien, z. B. zur Produktverfolgung oder zur Fälschungssicherung. Rauhe et al. demonstrierten mit einer 8-Bit-Kodierung (s. o. unter Steganographie, und Ab-

schnitt 8.2) die Markierung von Motoröl, Farbe und Papier sowie die erfolgreiche Extraktion der Moleküle aus diesen Stoffen mit anschließendem Auslesen der gespeicherten Information [132]. Dieses Verfahren läßt sich sinnvoll mit oben genannter Steganographie-Methode ergänzen.

3.2.3 PCR und Microarrays

Um einen effizienten und fehlerfreien Ablauf der PCR (s. Abschnitt 2.1) zu ermöglichen, müssen die Primer sorgfältig gewählt werden. Da die Verlängerung bei einer Temperatur ablaufen muß, die für die Polymerase günstig ist, muß die Schmelztemperatur der Primer deutlich über dieser liegen, um eine stabile Hybridisierung zu gewährleisten. Häufig wird die PCR eingesetzt, um eine spezifische Sequenz aus einer sehr großen Sequenzmenge zu amplifizieren. Dann müssen die Primer sehr spezifisch hybridisieren und dürfen nicht auch an anderen Sequenzen der gegebenen Menge binden, so daß diese verlängert würden. Weiterhin dürfen die beiden Primer keinen Duplex bilden und auch jeweils für sich keine Sekundärstrukturen ausbilden.

Auch bei DNA-Microarrays sind die Vermeidung sowohl von Sekundärstrukturen der Sonden als auch von Kreuzhybridisierungen zwischen den Zielsequenzen und zwischen Sonden und falschen Zielsequenzen wichtig, ebenso eine möglichst hohe und gleichmäßige Hybridisierungseffizienz, damit die Signalstärke tatsächlich mit der Konzentration der Zielsequenzen korreliert.

3.3 Problemdefinition

Ausgehend von den Anforderungen des Self-Assembly und dessen Anwendungsgebiete an die DNA-Moleküle kann das DNA-Sequenz-Design-Problem zunächst sehr allgemein formuliert werden:

Definition 1 *Gegeben sei eine Liste* $L = (l_1, l_2, \ldots, l_n)$ *von Sequenzlängen, eine Menge* $P = \{\{(i_s, i_p), (j_s, j_p)\}, i_s, j_s \leq n, i_p \leq l_{i_s}, j_p \leq l_{j_s}\}$ *von Zweiermengen, deren Elemente wiederum Paare aus Sequenz- und Positionsindices sind, sowie ein Hybridisierungsmodell, daß eine Vorhersage von Hybridisierungswahrscheinlichkeiten basierend auf den Basenabfolgen der Sequenzen erlaubt. Dann lautet das* DNA-Sequenz-Design-Problem (DSD)*: Finde n Sequenzen* $S_i \in \{A, C, G, T\}^{l_i}, i = 1, \ldots, n$*, so daß die dadurch definierten DNA-Moleküle* in vitro *mit möglichst hoher Wahrscheinlichkeit gemäß dem Hybridisierungsmodell so hybridisieren, daß anschließend für jedes Element* $\{(i_s, i_p), (j_s, j_p)\} \in P$ *die* i_p*-te Base der* i_s*-ten Sequenz mit der* j_p*-ten Base der* j_s*-ten Sequenz ein Basenpaar bildet, und es mit möglichst kleiner Wahrscheinlichkeit weitere Basenpaare gibt, die nicht in P enthalten sind.*

Diese Definition ist tatsächlich nicht die allgemeinst mögliche, man kann auch noch auf die Festlegung der Sequenzlängen verzichten, um z. B. bei der Suche nach Sequenzen einer bestimmten Schmelztemperatur eine größere Auswahl zu haben. Dieses ist bisher jedoch nur sehr selten der Fall, in der überwiegenden Anzahl der Fälle ist die Definition anwendbar.

Diese Definition erlaubt beliebige Strukturen, die die DNA-Moleküle annehmen sollen. Ist tatsächlich eine komplexe (d. h. über den linearen Fall hinausgehende) Struktur beabsichtigt, so heißt das Problem auch *Struktur-Design-Problem*. Ist die gewünschte Struktur die Sekundärstruktur eines einzelnen Stranges, wird das DSD-Problem für diese Fälle auch als *Inverse Secondary Structure Problem* bezeichnet[2]. Für viele DNA-Computing-Protokolle sowie für Microarray-Anwendungen reduziert sich das DSD-Problem allerdings auf die Suche nach einer Menge von Oligomeren, die perfekte Duplexe bilden sollen.

[2]Das „originale“, also nicht-inverse Secondary Structure Problem stellt die Suche nach einer Struktur dar, die ein DNA- oder RNA-Molekül mit vorgegebener Sequenz *in vitro* einnimmt.

Definition 2 *Gegeben sei eine Anzahl n von zu findenden Sequenzen der Länge l. Dann ist das* DNA-Word-Design-Problem (DWD)*: Finde n Wörter* $S_i \in \{A, C, G, T\}^l$*, so daß die dadurch definierten DNA-Moleküle* in vitro *mit möglichst hoher Wahrscheinlichkeit gemäß dem Hybridisierungsmodell zu perfekten Duplexen mit ihren Komplementärsequenzen hybridisieren, und es nur mit möglichst kleiner Wahrscheinlichkeit andere Verbindungen gibt.*

Konkrete Probleminstanzen definieren i. a. auch die Anforderungen an die Spezifität der Hybridisierung genauer. Weiterhin ergeben sich für konkrete Anwendungen weitere Anforderungen an die zu suchenden Sequenzen. Folgende Liste enthält eine Aufstellung der üblichen Anforderungen. Sie ist vermutlich nicht vollständig, da neue Anwendungen auch neue Bedingungen erfordern können. Ebenfalls wird eine konkrete Anwendung i. a. nur eine Teilmenge dieser Anforderungen an die Sequenzen stellen.

- Die Moleküle sollen *spezifisch* hybridisieren. Dies ist die wichtigste Anforderung. Sie kann in verschiedenen Formen konkretisiert werden:
  - Kein DNA-Wort soll mit einem anderen Wort hybridisieren.
  - Kein DNA-Wort soll mit dem Komplement eines anderen Wortes hybridisieren.
  - Kein DNA-Wort soll eine Sekundärstruktur wie Hairpin Loops, Bulges usw. bilden (oder zumindest keine andere als die beabsichtigte)
  - Kein DNA-Wort soll mit der Konkatenation zweier Wörter hybridisieren.
  - Kein DNA-Wort soll mit der Konkatenation der Komplemente zweier Wörter hybridisieren.
  - Keine Konkatenation zweier DNA-Wörter soll eine Sekundärstruktur bilden (oder keine andere als die beabsichtigte)
- Die Moleküle sollen mit möglichst *hoher Effizienz* hybridisieren. D. h. möglichst viele der Moleküle sollen die geplanten Hybridisierungen auch eingehen, das Reaktionsgleichgewicht soll also möglichst weit in Richtung des hybridisierten Zustands verschoben sein.
- Beim Einsatz mehrerer Sequenzen sollen alle mit *gleicher Effizienz* hybridisieren. Dies kann z. B. erforderlich sein, um eine Gleichverteilung der Wahrscheinlichkeiten bei einer zufälligen Erzeugung von Lösungskandidaten im DNA-Computing zu erzielen, um beim Self-Assembly von größeren Strukturen ein gleichmäßiges Wachstum zu erreichen, oder um bei DNA-Microarrays zur Genexpressionsanalyse sicherzustellen, daß die gemessene Signalstärke tatsächlich von der Konzentration der zu untersuchenden RNA in der Zelle abhängt, und nicht von der Basenkomposition der Sonden.
- Alle Sequenzen sollen eine möglichst ähnliche *Schmelztemperatur* haben (oder eine, die einer vorgegebenen Schmelztemperatur möglichst nahe kommt).
- Alle Sequenzen sollen eine möglichst ähnliche Stabilität, gemessen durch die *freie Enthalpie*, besitzen (oder eine, die einer vorgegebenen freien Enthalpie möglichst nahe kommt).
- Alle Sequenzen sollen einen möglichst ähnlichen *GC-Gehalt* haben (oder einen, die einem vorgegebenen GC-Gehalt möglichst nahe kommt).
- Zwei oder mehrere Mengen von Hybridisierungen von (Teil-)Sequenzen sollen *unterschiedlich stabil* sein (gemessen durch Schmelztemperatur oder freie Enthalpie), mit

bestimmten Mindestabständen der Stabilitäten. Dies kann für ein stufenweises Self-Assembly sinnvoll sein, bei dem in der ersten Stufe eingegangene Bindungen sich in der zweiten Stufe nicht wieder lösen sollen.

- Die Sequenzen sollen bestimmte Subsequenzen (z. B. Restriktionsschnittstellen, Bindungsstellen für Proteine, eine `GC`-Wiederholung zur Ausbildung von Z-DNA) enthalten. Dies legt i. a. auch die Positionierung dieser Subsequenzen fest, dementsprechend dürfen diese nirgendwo anders auftauchen.

- Bestimmte Sequenzen dürfen keinesfalls als Subsequenzen auftauchen. Dies können ebenfalls Restriktionsschnittstellen oder Proteinbindungsstellen sein, aber auch Einschränkungen wie maximal zwei aufeinanderfolgende Guaninbasen zur Vermeidung von Vierfachsträngen, `G-G`-Basenpaaren [160, 151], und starken Abweichungen von Stabilitäts- und Schmelztemperaturvorhersagen mit dem nearest-Neighbor-Modell [33, 145].

- Die Auswahl der Basen kann an bestimmten Positionen oder global für alle Sequenzen eingeschränkt werden. So wird gelegentlich auf dem Alphabet {`A,C,T`} gearbeitet, um die Wahrscheinlichkeit von Hairpin Loops zu verringern [52, 127]. Ebenso kann das *Ausfransen* (*fraying*) von Sequenzen, also das Öffnen eines Duplex an den Enden verhindert werden, indem als erstes und letztes Basenpaar je ein `G-C`-Basenpaar verwendet wird, das stabiler ist als ein `A-T`-Basenpaar.

- Die zu findenden Sequenzen dürfen keine Hybridisierungsneigung zu vorgegebenen Sequenzen aufzeigen, z. B. wenn eine bestehende Bibliothek erweitert werden soll.

Die Vielzahl unterschiedlicher Probleminstanzen läßt vermuten, daß es gemäß dem *No-Free-Lunch*-Theorem [178] keinen allgemeinen Sequenz-Design-Algorithmus gibt, der für alle Anwendungen gleich gut (oder überhaupt) geeignet ist. Aber zumindest tauchen einige der wichtigen Anforderungen immer wieder auf, diesen sollte ein halbwegs flexibel einsetzbares Design-Tool also mindestens gerecht werden können.

Kapitel 4

Modellierung der Hybridisierungswahrscheinlichkeit

Zur Modellierung der Hybridisierungswahrscheinlichkeiten gibt es verschiedene Ansätze. Viele beruhen nur auf der reinen Basensequenz als Zeichenkette, ohne auf biochemische Details zu achten. Andere schließen mehr thermodynamische Details der Hybridisierung ein. I. a. sind Erstere leichter zu berechnen als Letztere. Eine andere Unterscheidung der Modelle ist die in paarweise und mengenweise Maße. Während Erstere explizit die Hybridisierung zwischen zwei Molekülen modellieren, dienen Letztere eher als Gütemaß für eine Sequenzmenge, wobei die Kardinalität dieser Menge natürlich auch 2 oder sogar nur 1 betragen kann.

4.1 Distanzmaße als Maße für die Hybridisierungswahrscheinlichkeit

4.1.1 Motivation

Um eines der wichtigsten Teilziele des DSD-Problems, die Vermeidung von Kreuzhybridisierungen, zu realisieren, sollte ein Entwurfsalgorithmus die Wahrscheinlichkeit für eine solche Kreuzhybridisierung in Abhängigkeit der beteiligten Nukleotidsequenzen berechnen oder zumindest gut schätzen können. Nicht nur, weil sich DNA-Moleküle im Rechner einfach als Zeichenketten darstellen lassen, bieten sich Distanzmaße für die Unähnlichkeit von Zeichenketten für diese Schätzung an. Es ist auch biochemisch durchaus motiviert, da eine DNA-Sequenz X eine tendentiell höhere Neigung zur Hybridisierung mit einer anderen Sequenz Y zeigt, wenn X dem Watson-Crick-Komplement von Y ähnlich ist. Sämtliche solcher Distanzmaße abstrahieren aber verschieden stark von den tatsächlichen chemischen Vorgängen in vitro, und bieten damit unterschiedlich realistische Schätzungen für die Hybridisierungswahrscheinlichkeit.

In diesem Kapitel werden zunächst einige gebräuchliche Distanzmaße vorgestellt und Probleme bei ihrer Anwendung als Hybridisierungswahrscheinlichkeitsmaß diskutiert. Im zweiten Teil wird in einem *in silico* Experiment verglichen, wie sehr die Abschätzungen untereinander und gegenüber dem (vermutlich) realistischsten Maß korrelieren. Folgende Schreibweisen werden verwendet: Wenn $X = x_1\ x_2 \ldots x_{l_X}$ eine Zeichenkette über einem Alphabet Σ ist, so ist x_i das Zeichen, das in X an i- ter Position steht, und l_X die Länge von X. Die Watson-Crick-komplementäre Sequenz zu einer DNA-Sequenz X wird mit $\overline{X}$ bezeichnet. Die Funktion $\sigma^k(X)$ liefert die Verschiebung von X um k Stellen: $(\sigma^k(X))_i = x_{i+k}, k \in Z$. Die Verschiebung kann nach rechts oder links erfolgen, die Indizes können also auch negativ werden. Mit Lücken

werden sowohl Bulges als auch internal Loops bezeichnet, bei denen die beiden einzelsträngigen Abschnitte verschieden lang sind.

4.1.2 Distanzmaße

Hamming-Distanz Die Hamming-Distanz $H(X,Y)$ zwischen zwei Zeichenketten X und Y ist die Anzahl der Zeichenpositionen i, an denen $x_i \neq y_i$ gilt. Ursprünglich wurde die aus der Kodierungstheorie stammende Hamming-Distanz für gleich lange Ketten definiert, sie läßt sich aber für verschieden lange Ketten verallgemeinern, indem man z. B. die überzähligen Zeichen der längeren Kette als Unterschiede wertet und ihre Anzahl zur Hamming-Distanz addiert [61].

H-Maß, H-Distanz Garzon et al. haben H-Maß (H-measure) und H-Distanz (H-distance) als DNA-Computing-geeignete Erweiterung der Hamming-Distanz entwickelt [61, 129]. Das H-Maß $h(X,Y)$ ist definiert durch

$$h(X,Y) = \min_{-n<k<n}\{|k| + H(X, \sigma^k(\overline{Y}))\}, \tag{4.1}$$

wobei H die Hamming-Distanz des doppelsträngigen Teils der Verschiebung ist, also über die Indizes i mit $1 \leq i \leq l_X$ und $k \leq i \leq l_Y + k$. Die Sequenz X wird also immer mit dem Komplement von Y verglichen, dafür wird Selbstähnlichkeit nicht berücksichtigt, d.h. insbesondere hat eine Sequenz einen Abstand größer 0 zu sich selbst, außer sie ist selbstkomplementär.

Da das H-Maß somit keine Metrik darstellt, wurde es von Garzon et al. zur H-Distanz erweitert[1]. Diese wird nicht zwischen einzelnen Sequenzen gemessen, sondern zwischen Sequenzklassen (sog. projective oligos, oder Poligos), die jeweils eine Sequenz und ihr Watson-Crick-Komplement umfassen, also die beiden Sequenzen, die genau das H-Maß 0 zueinander haben. Selbstkomplementäre Sequenzen bilden Klassen der Kardinalität 1. Die H-Distanz zwischen zwei Poligos PX und PY ist dann definiert durch

$$HD(PX, PY) = \min_{\substack{X \in PX,\\ Y \in PY}}\{h(X,Y)\}. \tag{4.2}$$

Die H-Distanz zwischen zwei Sequenzen läßt sich leicht durch die H-Distanz zwischen den entsprechenden Poligos darstellen.

$\mathbf{H_M}$ Eine weitere Variante der Hamming-Distanz wird in [93] beschrieben. Dieses sehr spezielle Maß dient in erster Linie dem Vergleich von DNA-Codewörtern mit Konkatenationen aus Wörtern. Es ist für zwei Sequenzen X und Y definiert durch:

$$H_M(X,Y) = \begin{cases} \min\{H(X,Z) | Z \text{ Teilwort von } Y \text{ der Länge } l_X\}, & \text{falls } l_X \leq l_Y, \\ l_X, & \text{falls } l_X > l_Y. \end{cases} \tag{4.3}$$

Homologie Um eines der Probleme der n_b-Uniqueness (s. Abschnitt 4.3.1) bei schlechtem Verhältnis von Sequenz- zu Basisstranglänge kompensieren zu können, wurde ein Distanzmaß namens Homologie entworfen und implementiert [53]. Es ist definiert als das Verhältnis

$$Hom(X,Y) = \max\{match(X, \sigma^k(Y))\}/l, \tag{4.4}$$

[1] In [129] wird verwirrenderweise H-measure mit h-distance bezeichnet.

```
aacgctt--gact
-acg-ttaaggc-
```

Abbildung 4.1: Beispiel für ein Alignment zweier DNA-Sequenzen. Lücken sind mit einem - markiert. An drittletzter Position befindet sich ein Mismatch.

wobei *match* die Anzahl der gleichen Basen $x_i = y_{i+k}$ zählt, und $l = max(l_X, l_Y)$ die Länge der längeren Sequenz ist. Eine Homologie von 1 bedeutet also Identität der Sequenzen, zwei Sequenzen mit Homologie 0 haben keine Base gemeinsam. Eine Distanz sollte umso größer sein, je unähnlicher die Sequenzen sind, also bietet sich als Distanzmaß eher $1 - Hom(X, Y)$ an.

Edit-Distanz Die Edit-Distanz $E(X, Y)$, auch als Levenshtein-Distanz bekannt, gibt den Aufwand wieder, den es kostet, eine Zeichenkette in eine andere zu überführen [69]. Gegeben sind die Operationen *Insert*, mit der genau ein Zeichen in X eingefügt wird, *Delete*, die genau ein Zeichen aus X entfernt, und *Replace*, die ein Zeichen in X durch ein Zeichen in Y ersetzt. Die Edit-Distanz zwischen beiden Zeichenketten ist dann die minimale Anzahl dieser Operationen, die nötig ist, um X in Y zu transformieren. Z. B. ist E(wespe, wiese) = 2, da man mindestens ein i nach dem w einfügen und das p entfernen muß.

Globales Alignment Beim Alignment (dt.: Anordnung, Ausrichtung) zweier (oder mehrerer) Sequenzen versucht man, diese so parallel anzuordnen, daß ähnliche Subsequenzen nebeneinander zu liegen kommen. Hierbei sind neben Verschiebungen auch Lücken und Fehlpaarungen erlaubt (Abb. 4.1). Der Zusatz „global“ betont, dass die kompletten Sequenzen verglichen werden, und nicht nur Subsequenzen, wie es beim in der Bioinformatik ebenfalls gebräuchlichen lokalen Alignment der Fall ist.

Ein verbreiteter Algorithmus zur Berechnung globaler Alignments wurde von Needleman und Wunsch entwickelt [118] (übersichtlicher beschrieben in [49]). Es sei ein Bewertungssystem gegeben, das z. B. jede Gegenüberstellung von gleichen oder ungleichen Basen oder auch das Einfügen von Lücken auf bestimmte Werte abbildet. Der Algorithmus zur Berechnung eines bezüglich dieses Bewertungssystems optimalen (maximal bewerteten) Alignments zweier Sequenzen X und Y berechnet eine Matrix F, wobei $F(i, j)$ den Wert des optimalen Alignments der Teilsequenzen $x_1 \dots x_i$ und $y_1 \dots y_j$ ist. $F(i, j)$ läßt sich rekursiv definieren. Es sei $F(0, 0) = 0$. Für beliebige Indizes i und j gibt es drei Möglichkeiten, kürzere optimale Alignments um eine Position zu verlängern:

- x_i wird y_i gegenübergestellt, dann ist $F(i, j) = F(i - 1, j - 1) + s(x_i, y_i)$,
- x_i steht einer Lücke gegenüber, dann ist $F(i, j) = F(i - 1, j) - d$,
- y_j steht einer Lücke gegenüber, dann ist $F(i, j) = F(i, j - 1) - d$,

wobei s die Bewertungsfunktion für die Gegenüberstellung von Basen ist und d ein Strafterm für Lücken. Für ein optimales Alignment wird der größte dieser drei Werte gewählt. Nachdem iterativ alle $F(i, j)$ berechnet wurden, liefert $F(l_X, l_Y)$ den Wert des kompletten Alignments.

Die Bewertung eines globalen Alignments ist nicht direkt als Distanzmaß geeignet, da i. a. ähnliche Sequenzen (also mit geringer Distanz) eine höhere Bewertung bekommen, und je nach Bewertungssystem auch negative Bewertungen möglich sind. Sie läßt sich aber leicht in ein Distanzmaß wandeln, z. B. indem man sie von einer maximal möglichen Bewertung (für identische Sequenzen) subtrahiert.

Smith und Waterman entwickelten aus dem globalen Alignment das lokale Alignment, das zur Suche nach ähnlichen Subsequenzen angewandt wird [155]. Da hier nur komplette Sequenzen verglichen werden sollen, wird dieses Maß nicht näher betrachtet.

L-Tupel-basierte Distanzmaße Eine Reihe von Distanzmaßen verwenden kein Alignment der Sequenzen, sondern beruhen auf der Häufigkeit, mit der Subsequenzen einer festen Länge L (L-Tupel) in den zu vergleichenden Sequenzen vorkommen [169]. Hierzu wird für jede Sequenz X ein Vektor $c_L^X = (c_{L,1}^X, ..., c_{L,K}^X)$ berechnet, wobei $K = |\Sigma|^L$ die Anzahl aller möglichen L-Tupel ist, und $c_{L,i}^X$ angibt, wie oft der i-te aller möglichen L-Tupel in X vorkommt. Für einige Distanzmaße wird auch der Vektor der Frequenzen f_L^X verwendet, wobei $f_{L,i}^X = c_{L,i}^X/(l_X - L + 1)$.

Ein naheliegendes Maß für die Distanz zwischen zwei Sequenzen X und Y ist dann der Euklidische Abstand zwischen den entsprechenden Häufigkeitsvektoren

$$d_L^E(X,Y) = (c_L^X - c_L^Y)^T \cdot (c_L^X - c_L^Y) = \sum_{i=1}^{K} (c_{L,i}^X - c_{L,i}^Y)^2 \tag{4.5}$$

Um die Abhängigkeit der Distanz von der Tupellänge L abzuschwächen, kann man die Distanzen über verschiedene Werte von L kumulieren. Außerdem können die verschiedenen L-Tupel je nach Ziel der Anwendung mit Gewichten ρ_i gewichtet werden.

$$d^2(X,Y) = \sum_{L=l}^{u} \sum_{i=1}^{K} \rho_i (c_{L,i}^X - c_{L,i}^Y)^2 \tag{4.6}$$

Der Euklidische Abstand läßt sich standardisieren, indem man die Differenz in jeder Dimension (also für jedes L-Tupel) durch die Varianz s_{ii} der Frequenz des i-ten L-Tupels in einer Zufallssequenz der Länge l_X teilt [181][2].

$$\begin{aligned} &d_L^{SE}(X,Y) = \sum_{i=1}^{K} \frac{(c_{L,i}^X - c_{L,i}^Y)^2}{s_{ii}}, \text{ wobei} \\ &s_{ii} = l \cdot 4^{-L} - 4^{-2L}(l^2 - (l-L+1)(l-L)) + \sum_{k=1}^{L-1} (l-k) 4^{-k} Q_{L-k}, \text{mit} \\ &l = l_X - L + 1 \text{ und } Q_j = \begin{cases} 1 & \text{falls} (z_1, \ldots, z_j) = (z_{L-j+1}, \ldots, z_L), \\ 0 & \text{sonst} \end{cases} \\ &\text{und } Z = (z_1, \ldots, z_L) \text{ ist das } i\text{-te } L\text{-Tupel.} \end{aligned} \tag{4.7}$$

Auch dieses Maß läßt sich natürlich kumulieren.

$$d^{SE*}(X,Y) = \sum_{L=l}^{u} \sum_{i=1}^{K} d_L^{SE}(X,Y) \tag{4.8}$$

Einige vielleicht nicht ganz so naheliegende Distanzmaße sind der lineare Korrelationkoeffizient der Frequenzvektoren

$$d_L^{LCC}(X,Y) = \frac{K \sum_{i=1}^{K} f_{L,i}^X \cdot f_{L,i}^Y - \sum_{i=1}^{K} f_{L,i}^X \cdot \sum_{i=1}^{K} f_{L,i}^Y}{\left[K \sum_{i=1}^{K} (f_{L,i}^X)^2 - (\sum_{i=1}^{K} f_{L,i}^X)^2\right]^{1/2} \cdot \left[K \sum_{i=1}^{K} (f_{L,i}^Y)^2 - (\sum_{i=1}^{K} f_{L,i}^Y)^2\right]^{1/2}}, \tag{4.9}$$

[2] Vinga und Almeida zitieren in [169] die Gleichung für d^{SE} fehlerhaft aus [181]

`ABBABAABBAABABBA` → `A` · `B` · `BA` · `BAA` · `BBAA` · `BABB` · `A`

Abbildung 4.2: Beispiel für eine minimale Zerlegung einer Sequenz zur Bestimmung der Lempel-Ziv-Komplexität. Sowohl `A` als auch `B` sind zunächst neue Subsequenzen. Das zweite `B` kann durch Kopieren entstehen, `BA` ist wieder eine neue Subsequenz, usw. Die gezeigte Sequenz hat also eine Lempel-Ziv-Komplexität von 7.

der auch erst durch Umrechnung z. B. durch $1 - d^{LCC}$ ein wirkliches Distanzmaß wird, sowie zwei Metriken, die auf dem Winkel zwischen den Häufigkeitsvektoren beruhen:

$$d_L^{cos}(X,Y) = \theta_{XY}, \text{mit} \cos(\theta_{XY}) = \frac{\sum_{i=1}^{K} c_{L,i}^X \cdot c_{L,i}^Y}{\sqrt{\sum_{i=1}^{K}(c_{L,i}^X)^2} \cdot \sqrt{\sum_{i=1}^{K}(c_{L,i}^Y)^2}} \tag{4.10}$$

$$d_L^{Evol}(X,Y) = -\ln[(1+\cos\theta_{XY})/2] \tag{4.11}$$

Vinga und Almeida stellen noch weitere L-Tupel-basierte Distanzmaße vor, die hier nicht näher betrachtet werden, da sie auch nicht im experimentellen Vergleich untersucht wurden. So erfordert die sog. Mahalanobis-Distanz die Berechnung der Inversen der Kovarianz-Matrix der beiden Sequenzen, was recht schwierig ist, da die Kovarianz-Matrix nahezu singulär ist [169]. Bei der Berechnung der Kullback-Leiber-Diskrepanz wird durch die Frequenzen geteilt, zur Vermeidung einer Division durch Null müssen die Frequenzen also um einen nicht näher bestimmten Betrag verschoben werden. Ein auf der Kolmogorov-Komplexität der Sequenzen basierendes Maß ist vom theoretischen Ansatz her interessant und vielversprechend, allerdings ist es nicht trivial, die Kolmogorov-Komplexität zu berechnen bzw. anzunähern. Ein Versuch in dieser Richtung wird im nächsten Abschnitt beschrieben.

Komplexitätsbasierte Distanzmaße Die Kolmogorov-Komplexität einer Sequenz ist die Länge des kürzesten Programms, das diese Sequenz erzeugt und terminiert. Leider ist diese Länge prinzipiell nicht berechenbar (da sonst das Halteproblem lösbar wäre), man kann aber recht gute obere Schranken abschätzen, indem man die Sequenz komprimiert und die Länge der komprimierten Sequenz plus die Länge des Dekomprimierers angibt.

Eine verbreitete Methode zur Komprimierung ist der Lempel-Ziv-Algorithmus, der z. B. auch im Programm gzip verwendet wird. Lempel und Ziv entwickelten auch ein auf diesem Algorithmus basierendes Komplexitätsmaß [97]. Als einfache Operationen zur Erzeugung einer Sequenz erlauben sie das (bei Bedarf überlappende) Anhängen von bereits verwendeten Subsequenzen sowie das Verlängern einer solchen Subsequenz um ein Zeichen zu einer bisher noch nicht verwendeten Subsequenz. Die Lempel-Ziv-Komplexität gibt im Wesentlichen die minimale Anzahl der Operationen letzteren Typs an, die sich mit einem recht einfachen Algorithmus berechnen läßt (für ein Beispiel s. Abb. 4.2). Die genaue Definition ist recht umfangreich, daher verweise ich interessierte Leser auf [97].

Otu und Sayood haben mehrere Distanzmaße vorgeschlagen, die auf der Lempel-Ziv-Komplexität von Sequenzen beruhen. Motiviert sind diese durch folgende Überlegung: Wenn zwei Sequenzen X und Y sehr ähnlich sind, so werden sie viele Subsequenzen gemeinsam haben. Die Komplexität der Konkatenation XY (oder YX) beider Sequenzen dürfte also nicht viel höher sein als die der einzelnen Sequenzen, da die zweite Hälfte von XY weitgehend durch Kopieren der Subsequenzen aus X herstellbar ist und nur wenige neue, in X nicht vorhandene

Subsequenzen verwendet werden müssen. Die hier betrachteten Maße sind

$$d(X,Y) = max\{c(XY) - c(X), c(YX) - c(Y)\} \tag{4.12}$$

$$d^*(X,Y) = \frac{d(X,Y)}{max\{c(X), c(Y)\}} \tag{4.13}$$

$$d_1(X,Y) = c(XY) - c(X) + c(YX) - c(Y) \tag{4.14}$$

$$d_1^*(X,Y) = \frac{d_1(X,Y)}{c(XY)} \tag{4.15}$$

$$d_1^{**} = \frac{d_1(X,Y)}{\frac{1}{2}(c(XY) + c(YX))} \tag{4.16}$$

wobei $c(X)$ die Lempel-Ziv-Komplexität der Sequenz X, und XY die Konkatenation der Sequenzen X und Y ist. Alle fünf Maße sind Metriken. Da d und d_1 stark längenabhängig sind und d_1^* nicht symmetrisch ist, werden diese Maße im experimentellen Vergleich (s. u.) nicht berücksichtigt.

Freie Enthalpie Ein thermodynamisch detailliertes Maß ist die freien Enthalpie eines Duplex (s. Abschnitt 2.3). Bei einem allgemeinen Hybridisierungsmodell, das auch Mismatches, Bulges und Loops zuläßt, kann Dynamic Programming ähnlich wie beim globalen Alignment, aber unter Verwendung thermodynamischer Bewertungen gemäß eines um Bulges usw. erweiterten nearest-Neighbor-Modells, zur Berechnung des stabilsten Duplex, also dem mit der niedrigsten freien Enthalpie, eingesetzt werden. Ein solcher DP-Ansatz wird bisher hauptsächlich zur Vorhersage von einzelsträngigen Sekundärstrukturen verwendet [75, 186], seit kurzer Zeit gibt es auch eine Erweiterung dieses Verfahrens auf Duplexe [112]. Wie man die Sekundärstrukturvorhersage zur Schätzung der Duplexstabilität einsetzen kann, wird im nächsten Abschnitt beim experimentellen Vergleich beschrieben.

4.1.3 Theoretische Diskussion

Da die meisten Distanzmaße ursprünglich die Ähnlichkeit zweier Strings messen sollen, berücksichtigen sie (mit Ausnahme der speziell für das DNA-Computing entwickelten Maße H-Maß und H-Distanz) keine Watson-Crick-Komplementarität. Zur Messung der Hybridisierungswahrscheinlichkeit zweier Sequenzen X und Y muß man also bei diesen Maßen X mit $\overline{Y}$, also dem Komplement von Y, vergleichen.

Die Sequenzvergleichsmaße ignorieren verschiedene Aspekte der Hybridisierung. So erlaubt die Hamming-Distanz weder Verschiebungen der beiden Sequenzen gegeneinander noch Lücken, wie sie z. B. durch Bulge Loops entstehen. H-Maß und H-Distanz berücksichtigen zumindest Verschiebungen, aber keine Lücken. Diese beiden Maße sind auch nur für den relativen Vergleich von Sequenzen gleicher Länge sinnvoll, da die Länge von dangling Ends zum Abstand beiträgt. D. h., daß ein perfekter Duplex der Länge n mit dem H-Maß die Entfernung 0 bekommt, dieser Duplex mit zwei dangling Ends jeweils der Länge m aber die Entfernung m, obwohl beide Duplexe ähnlich stabil sein sollten. [3]

Nur eine sehr eingeschränkte Art von Verschiebungen fließt in H_M ein, und zwar nur solche, bei denen die kürzere Sequenz nicht über die längere hinaus ragt. Dieses Maß, das auch keine Lücken modelliert, scheint also auch nur für den sehr speziellen Zweck brauchbar zu sein, für den

[3] Tatsächlich ist hier der Duplex mit dem größeren Abstand wahrscheinlich sogar stabiler, da dangling Ends eher stabilisierend wirken [30]

es entwickelt wurde, nämlich den Vergleich eines Wortes mit der Konkatenation zweier gleich langer Wörter. Insbesondere ist das Maß auch nicht symmetrisch, wenn die beiden Sequenzen verschieden lang sind.

Die Homologie bzw. $1-Hom(X,Y)$ ist ein normalisiertes Maß, d. h. zwei beliebig unähnliche Sequenzen haben immer einen Abstand von höchstens 1, unabhängig von der Länge. Dies ist thermodynamisch nicht sehr realistisch, die Homologie ist also auch eher für den Vergleich von Sequenzpaaren gleicher Länge geeignet.

Die Edit-Distanz beinhaltet sowohl Verschiebungen als auch Lücken, jedoch spiegeln die Kosten für die drei Operationen nur mäßig die Thermodynamik der Hybridisierung wieder. So würden z. B. dangling Ends oder Bulge Loops mit Kosten bewertet, die der Anzahl der Basen im dangling End oder in der Loop entspricht, da jede dieser Basen eingefügt bzw. entfernt werden muß. Thermodynamisch exaktere Modelle gehen aber davon aus, daß der Einfluß von Schlaufen etwa logarithmisch von der Länge abhängt [34]. Außerdem hat auch hier, wie beim H-Maß, die Vergrößerung des Abstandes durch dangling Ends zur Folge, daß der Vergleich verschieden langer Sequenzen zu unrealistischen Ergebnissen führt.

Ein ähnliches Problem stellt sich auch beim globalen Alignment, für das ein geeignetes Bewertungssystem vorgegeben werden muß. Qualitativ ist klar, dass gleiche Basen x_i und y_j belohnt, unterschiedliche Basen sowie Lücken dagegen bestraft werden sollten, eine quantitative Festlegung ist aber nicht trivial. Ebenfalls gelten die für die Edit-Distanz gemachten Aussagen bzgl. der Schleifenlängen und dangling Ends auch für das globale Alignment.

Alle diese Maße zählen nur unterschiedliche bzw. gleiche Zeichen, bewerten aber nicht, ob gleiche (bzw. komplementäre) Zeichen zusammenhängen. So haben zwei Sequenzen $X =$ `GGGGGGGG` und $Y =$ `GGGGAAAA` die Hamming-Distanz $H(X,Y) = 4$, und X zu einer weiteren Sequenz $Z =$ `GAGAGAGA` die gleiche Distanz $H(X,Z) = 4 = H(X,Y)$. Während aber in einem Duplex aus X und $\overline{Y} =$ `TTTTCCCC` die vier aufeinanderfolgenden `G-C`-Basenpaare recht stabil hybridisieren können, wäre ein Duplex aus X und $\overline{Z} =$ `TCTCTCTC` äußert instabil, da die vereinzelten `G-C`-Basenpaare nur Nukleationen, aber kein Stacking benachbarter Basenpaare ermöglichen.

Die verschiedenen L-Tupel-basierten Distanzmaße sind nicht für die Erzeugung oder Analyse von spezifisch hybridisierenden Oligos entwickelt worden, sondern für verschiedene Anwendungen der Bioinformatik wie z. B. die Klassifikation von Proteinen anhand ihrer Aminosäuresequenz oder die Erstellung von phylogenetischen Bäumen [169]. Dementsprechend sollen sie auch keine Hybridisierungswahrscheinlichkeiten oder thermodynamische Stabilitäten modellieren, sondern z. B. bei den phylogenetischen Bäumen evolutionäre Veränderungen im Genom. Da in diesen Anwendungen zudem eher lange Sequenzen, also DNA-Polymere, untersucht werden, sind diese Maße vermutlich nur bedingt für das DNA-Sequenz-Design geeignet. Weiterhin stellt die Beschränkung auf Tupel einer bestimmten Länge L gerade bei den nichtkumulativen Maßen eine künstliche Rasterung dar, die i. a. nicht durch die zu modellierenden natürlichen Vorgänge motiviert ist. Außerdem steht auch hier der Anwender vor der Qual der Parameterwahl. Es ist *a priori* nicht ersichtlich, welches L für eine bestimmte Anwendung am besten geeignet ist. Auch die auf der Lempel-Ziv-Komplexität basierenden Maße sind vermutlich eher für längere Sequenzen geeignet. Zur Anwendung kommen sie z. B. in BLAST-Algorithmen[4], um heuristisch sehr unähnliche Sequenzpaare auszuschließen und sich rechenzeitintensive Alignments zu sparen. Allerdings beruht auch die Wahrscheinlichkeit für die Hybridisierung zweier DNA-Stränge X und Y auf gemeinsamen Subsequenzen zwischen X

[4]BLAST-Algorithmen umfassen eine Familie von Sequenz-Alignment-Algorithmen, die zur Suche ähnlicher Sequenzen in genomischen und proteomischen Datenbanken verwendet werden

und $\overline{Y}$, daher erscheint die Verwendung von Tupeln durchaus realistisch. Ein solches gemeinsames Tupel stellt eine Nukleation plus mehrere Stackings von Basenpaaren dar, bildet also den Hybridisierungsprozeß ab. Gerade für kurze Tupel bleibt aber auch genügend Flexibilität bezüglich der Berücksichtigung von Lücken.

Alle diese Distanzmaße lassen sich in Entwurfsalgorithmen zur Bewertung vorliegender Sequenzen verwenden, z. B. als Zielfunktion eines Optimierungsverfahrens oder als Filter in einem Suchprozeß. Es läßt sich aus ihnen jedoch nur schwer ein Sequenzentwurfsalgorithmus entwickeln, der bereits bei der Erzeugung der Sequenzen ein Mindestmaß an Unähnlichkeit garantiert.

4.2 Experimenteller Vergleich

4.2.1 Versuchsüberblick

Die verschiedenen Distanzmaße wurden auf ihre Eignung als Wahrscheinlichkeitsmaß für die Hybridisierung der verglichenen Sequenzen untersucht. Als Referenz wurde die minimale freie Enthalpie verwendet, der damit unterstellt wird, der chemischen Realität am nächsten zu kommen. Zufallssequenzen wurden mit einer Reihe von Distanzmaßen verglichen und die berechneten Distanzen auf Korrelation mit der minimalen freien Enthalpie geprüft. Um die Längenabhängigkeit einer solchen Korrelation zu überprüfen, wurde der Versuch für Sequenzmengen mit unterschiedlichen Sequenzlängen durchgeführt. Einige Maße setzen voraus, daß die zu vergleichenden Sequenzen gleich lang sind. Um die Auswirkungen der Nichtbeachtung dieser Voraussetzung zu untersuchen, wurden auch Sequenzen mit paarweise unterschiedlicher Länge verglichen.

Zu erwarten wäre im Idealfall ein Verhältnis wie in Abbildung 4.3 angedeutet. Hat eine Sequenz nur eine geringe Distanz zum Watson-Crick-Komplement einer anderen Sequenz, so sollten sie sehr stabil, also mit niedriger freier Enthalpie hybridisieren. Je unähnlicher die Sequenzen werden, desto instabiler sollte ein Duplex aus ihnen und desto höher sollte die freie Enthalpie des Duplex sein. Man kann natürlich nur eine tendentielle Ähnlichkeit zu Abbildung 4.3 erwarten, und nicht eine Linie, die genau gleich aussieht.

Desweiteren wurde die Korrelation der verschiedenen Distanzmaße untereinander betrachtet. Dies liefert zwar keine direkten Erkenntnisse zur Modellierung der Hybridisierungswahrscheinlichkeit, kann aber die Analyse der Art, wie konkrete Distanzen zustande kommen, weiter erhellen.

4.2.2 Material und Methoden

Zufallssequenzen Es wurden 8 Mengen von Zufallssequenzen jeweils der Länge 8, 10, 15, 20, 25, 30, 50 und 10–30 Nukleotide (nt) erzeugt. In der letzten Menge wurde die Länge jeder Sequenz gleichverteilt zufällig zwischen 10 und 30 nt gewählt. Jede Menge enthält 1000 Sequenzen, die in 500 Paare eingeteilt wurden, so daß pro Menge und Distanzmaß 500 Messungen vorliegen.

Distanzmaße Untersucht wurden die Distanzmaße Hamming, H-Maß, H-Distanz, Homologie, Edit-Distanz, globales Alignment, die Lempel-Ziv-Komplexitätsbasierten Maße d^* und d_1^{**}, sowie die L-Tupel-basierten Maße $d_L^E, d^2, d_L^{SE}, d^{SE*}, d_L^{LCC}, d_L^{Cos}$ und d_L^{Evol}. Letztere wurden für die Tupellängen $L = 4$, 5 und 6 gemessen, die kumulativen Maße d^2 und d^{SE*} wurden über diese drei Längen summiert. Bei der Berechnung von d^2 waren alle Gewichte $\rho_i = 1$.

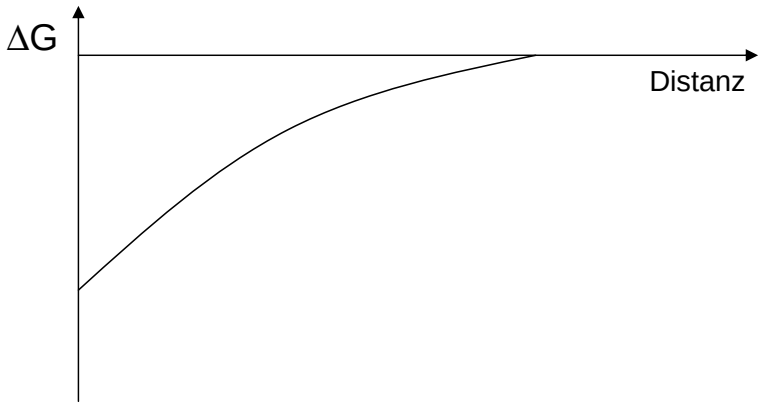


Abbildung 4.3: Erwarteter Zusammenhang von Distanz und freier Enthalpie. Hat eine Sequenz eine Distanz von 0 zum Komplement einer anderen Sequenz, so sollten die beiden einen perfekten Duplex bilden, der die höchste Stabilität und damit die kleinste freie Enthalpie aufweist. Mit zunehmender Distanz sollte die Duplexstabilität abnehmen, bis ab einer kritischen Unähnlichkeit keine Hybridisierung mehr stattfindet.

Die globalen Alignments wurden mit drei verschiedenen Parameterisierungen (s, d) berechnet, wobei s die Bewertung für komplementäre Basenpaare und d die Bewertung für Mismatches und Lücken ist. Die gewählten Parameterisierungen waren $(1, -2), (1, -1)$ und $(2, -1)$. Um die Distanz als Hybridisierungswahrscheinlichkeit interpretieren zu können, wurden für ein Paar von Sequenzen X und Y die Distanzen zwischen X und $\overline{Y}$ gemessen, nur H-Maß und H-Distanz wurden zwischen X und Y gemessen, da das Watson-Crick-Komplement in diesen beiden Maßen bereits berücksichtigt wird.

freie Enthalpie Die Stabilität einer Hybridisierung zweier DNA-Sequenzen wird durch ihre freie Enthalpie ΔG gemessen. Berechnet wurde diese mit dem Program RNAfold aus dem Vienna RNA Package von Hofacker et al. [75]. RNAfold dient eigentlich zur Vorhersage von einzelsträngigen Sekundärstrukturen für RNA-Sequenzen, läßt sich aber zur Berechnung der Duplexstabilität zweier Sequenzen zweckentfremden. Hierzu verbindet man die beiden Sequenzen über eine Linkersequenz aus Nukleotiden, die keine Paarungen eingehen können. Dies signalisiert man RNAfold durch die Verwendung „ungültiger“ Buchstaben für diese Basen, z. B. `N`. Eine Hairpin-Loop mit minimaler freien Enthalpie, deren Stamm der Duplex aus den beiden Sequenzen ist, kann dann als Annäherung des wahrscheinlichsten Duplex verwendet werden (Abb. 4.4).

Da die beiden Basen vor und nach der Linkersequenz nicht unbedingt ein komplementäres Basenpaar bilden, es also im Duplex terminal Mismatches oder dangling Ends gäbe, kann die eigentliche Schleife der Hairpin Loop unterschiedlich lang sein. Dies beeinflußt auch die Stabilität der Sekundärstruktur. Um den Einfluß einzelner zusätzlicher Basen in der Schleife gering zu halten, muß die Linkersequenz genügend lang sein. Da die Längenunterschied logarithmisch in die Energiedifferenz eingeht (Gl. 4.17), kann man diesen Einfluß bei Linkern der Länge 16 oder länger vernachlässigen [1], da er kleiner ist als der vermutliche Fehler der ΔG-Berechnung.

$$\Delta G_{loop}(l + \Delta l) - \Delta G_{loop}(l) = \frac{3}{2} RT \ln(1 + \Delta l / l) \tag{4.17}$$

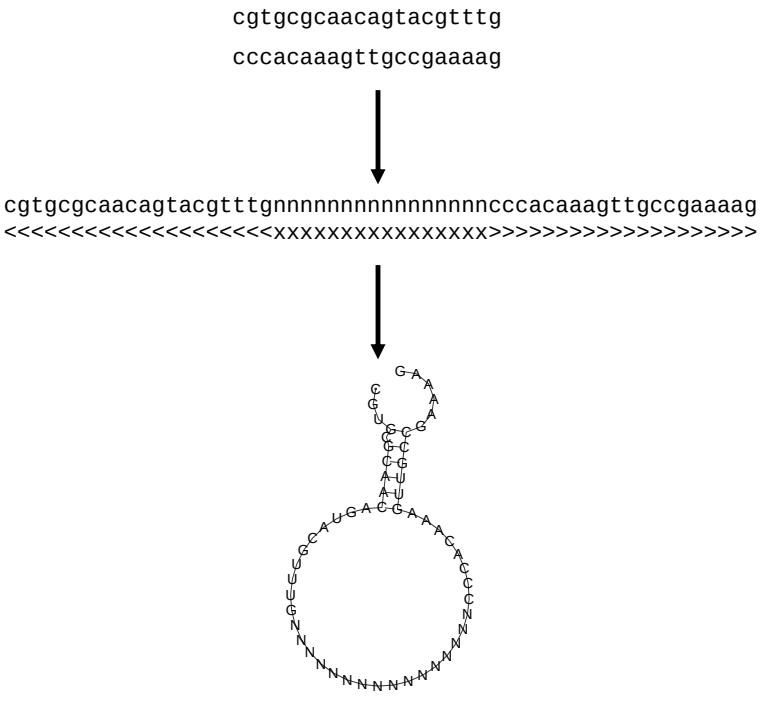


Abbildung 4.4: Berechnung des stabilsten Duplex mit RNAfold. Die beiden Sequenzen (oben) werden, durch eine Linkersequenz verbunden, zu einer Sequenz zusammengesetzt. Außerdem wird RNAfold als Randbedingung übergeben, daß die Einzelsequenzen keine Sekundärstrukturen bilden, sondern nur Basenpaarungen zwischen Basen beider Sequenzen betrachtet werden sollen (Mitte). RNAfold berechnet dann die Hybridisierung minimaler freier Enthalpie (unten). Im Bild ist T(hymin) durch U(racil) ersetzt, da RNAfold die Sequenzen als RNA betrachtet, und nicht als DNA.

Die ΔG-Werte sind also gegenüber den „echten“ Werten für eine reine Duplex-Hybridisierung verfälscht, aber alle um etwa den gleichen Betrag, sie sind also untereinander und mit den Distanzmaßen vergleichbar.

Die Distanzmaße können nur Unterschiede bzw. Ähnlichkeiten zwischen den beiden Sequenzen erfassen, nicht aber Ähnlichkeiten zwischen verschiedenen Bereichen ein und derselben Sequenz. Daher durfte RNAfold keine Sekundärstrukturen innerhalb einer der beiden Sequenzen berechnen, Basenpaare mußten also je eine Base aus jeder Sequenz beinhalten. Dies wurde durch die Angabe entsprechender Randbedingungen in der Eingabedatei für RNAfold erreicht (Abb. 4.4).

Leider stehen für die herunterladbare Version von RNAfold nur thermodynamische Parameter für RNA zur Verfügung. Ein Parametersatz für DNA ist nur mit dem Webserver verwendbar [74], was aber einen wesentlich höheren zeitlichen Aufwand für den Versuch bedeutet hätte. Außerdem erlaubt der Webserver nicht die Eingabe der o. g. Randbedingungen. Da zudem die Distanzmaße von den thermodynamischen Details von RNA genauso abstrahieren wie von denen von DNA, sollten die Ergebnisse dieses Experiments übertragbar sein.

Die freie Enthalpie einer Hybridisierungsreaktion ist temperaturabhängig. Hier wurde eine Temperatur von 25 °C vorgegeben, da dies zum Einen eine realistische Raumtemperatur ist, zum Anderen bei der Standardeinstellung von 37 °C viele der kurzen Sequenzpaare gar nicht hybridisierten und somit nicht vergleichbar waren (Vorversuche, Daten hier nicht enthalten). RNAfold läßt normalerweise die bei RNA möglichen `GU`-Basenpaare zu. Um die Vergleichbarkeit mit DNA zu ermöglichen, bei der keine `GT`-Basenpaare vorkommen, muß RNAfold diese Paarung verboten werden.

Insgesamt ergibt sich der folgende Aufruf von RNAfold:

```
RNAfold -noGU -T 25 -C < sequences.txt > out.txt
```

Korrelationskoeffizient Über jede der acht Sequenzmengen wurde für jedes Paar aus Distanzmaß und freier Enthalpie sowie auch für die Distanzmaße untereinander der Korrelationskoeffizient berechnet. Der Korrelationskoeffizient einer Meßreihe mit den Wertepaaren $(x_1, y_1), \ldots, (x_n, y_n)$ ist definiert durch

$$r = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{4.18}$$

Dabei sind s_x, s_y die Standardabweichungen und $\overline{x}, \overline{y}$ die Mittelwerte der beiden Meßreihen $(x_1, \ldots, x_n)$ und $(y_1, \ldots, y_n)$, sowie s_{xy} die Kovarianz der Meßreihe der Wertepaare.

Der Korrelationskoeffizient r ist ein Maß für die Stärke und Richtung des linearen Zusammenhangs der Werte zweier Meßreihen. Es gilt $-1 \leq r \leq 1$. Ein Korrelationskoeffizient nahe 1 oder -1 bedeutet einen starken linearen Zusammenhang zwischen den beiden Reihen, ist er dagegen nahe 0, so besteht kein Zusammenhang. Nach den Vorüberlegungen ist nicht unbedingt ein linearer Zusammenhang zwischen Distanzmaßen und freier Enthalpie zu erwarten, da aber viele Maße eine konstante Abnahme der Hybridisierungswahrscheinlichkeit pro Unterschied in den Sequenzen postulieren, ist die Verwendung des linearen Korrelationskoeffizienten durchaus motiviert.

4.2.3 Ergebnisse

Tabellen mit sämtlichen Korrelationskoeffizienten zwischen den Meßreihen sind im Anhang zu finden. Hier sollen nur interessante Teilbereiche betrachtet werden.

Länge	8	10	15	20	25	30	50	Mittel	StdAbw	10 – 30
Hamming	0.10	0.24	0.23	0.06	0.06	0.18	0.15	0.15	0.08	-0.29
H-Maß	0.43	0.44	0.43	0.36	0.31	0.29	0.27	0.36	0.07	-0.25
H-Distanz	0.33	0.27	0.23	0.22	0.21	0.14	0.19	0.23	0.06	-0.27
Homologie	-0.43	-0.44	-0.43	-0.36	-0.31	-0.29	-0.27	-0.36	0.07	-0.24
Edit-Distanz	0.32	0.34	0.40	0.32	0.28	0.36	0.47	0.36	0.06	-0.19
NW (1, -2)	-0.35	-0.35	-0.42	-0.35	-0.31	-0.39	-0.49	-0.38	0.06	-0.09
NW (1, -1)	-0.35	-0.36	-0.43	-0.36	-0.33	-0.41	-0.50	-0.39	0.06	-0.35
NW (2, -1)	-0.36	-0.36	-0.43	-0.36	-0.34	-0.42	-0.50	-0.40	0.06	-0.57
d^*	0.29	0.21	0.43	0.35	0.33	0.20	0.28	0.30	0.08	0.20
d_1^{**}	0.27	0.25	0.49	0.34	0.36	0.23	0.30	0.32	0.09	0.21
d_4^E	0.42	0.42	0.38	0.37	0.39	0.41	0.35	0.39	0.03	-0.37
d_5^E	0.34	0.28	0.33	0.26	0.33	0.35	0.25	0.31	0.04	-0.47
d_6^E		0.15	0.27	0.20	0.27	0.26	0.17	0.22	0.05	-0.52
$d^2(4,6)$	0.42	0.39	0.45	0.44	0.40	0.42	0.38	0.41	0.03	-0.27
d_4^{SE}	0.44	0.47	0.41	0.38	0.40	0.40	0.34	0.41	0.04	0.13
d_5^{SE}	0.33	0.31	0.36	0.26	0.35	0.33	0.24	0.31	0.05	0.08
d_6^{SE}		0.16	0.29	0.20	0.27	0.25	0.17	0.22	0.05	0.05
$d^{SE*}(4,6)$	0.47	0.34	0.38	0.31	0.37	0.35	0.26	0.35	0.07	0.06
d_4^{LCC}	-0.50	-0.57	-0.57	-0.52	-0.52	-0.46	-0.46	-0.51	0.05	-0.39
d_5^{LCC}	-0.38	-0.38	-0.50	-0.44	-0.44	-0.38	-0.34	-0.41	0.05	-0.32
d_6^{LCC}		-0.21	-0.41	-0.33	-0.36	-0.23	-0.28	-0.30	0.08	-0.19
d_4^{cos}	0.50	0.57	0.57	0.52	0.52	0.47	0.45	0.51	0.05	0.50
d_5^{cos}	0.38	0.38	0.50	0.44	0.44	0.38	0.34	0.41	0.05	0.38
d_6^{cos}		0.21	0.41	0.33	0.36	0.23	0.28	0.30	0.08	0.22
d_4^{Evol}	0.5	0.57	0.56	0.52	0.52	0.46	0.46	0.51	0.04	0.51
d_5^{Evol}	0.38	0.38	0.49	0.43	0.44	0.38	0.34	0.41	0.05	0.39
d_6^{Evol}		0.21	0.41	0.33	0.36	0.23	0.28	0.30	0.08	0.22

Tabelle 4.1: Korrelationskoeffizienten der Distanzmaße mit der freien Enthalpie ΔG_{25} über die acht Sequenzmengen sowie Mittelwert und Standardabweichung über die sieben Mengen mit konstanter Sequenzlänge. Bei den freien Feldern in der Spalte zu Sequenzlänge 8 konnte kein Korrelationskoeffizient berechnet werden, weil die Varianz der Distanzmessungen gleich Null war. Bei den mit dem Algorithmus von Needleman-Wunsch (NW) berechneten Alignments sind in Klammern die Belohnung für Watson-Crick-Basenpaare sowie die Strafe für Mismatches und Lücken angegeben.

Distanzmaße vs. freie Enthalpie Tabelle 4.1 zeigt die Korrelation der Distanzmaße mit der freien Enthalpie ΔG über alle 8 Mengen. Sämtliche Maße korrelieren nur schwach mit der freien Enthalpie. Der maximale Korrelationskoeffizient ist 0.57 (erreicht für d_4^{Cos} und d_4^{Evol} bei den 10-meren sowie für d_4^{Cos} bei den 15-meren), der minimale -0.57 (erreicht für d_4^{LCC} für 10- und 15-mere). Die Korrelationskoeffizienten für Homologie, die globalen Alignments und die d^{LCC}-Varianten sind negativ, weil sie nicht wie in Abschnitt 4.1 beschrieben zu wirklichen Distanzmaßen umgerechnet wurden, und daher größere Ähnlichkeit mit höheren Werten bemessen. Hier wären also Korrelationskoeffizienten nahe -1 ideal.

Die Hamming-Distanz zeigt eine nur sehr geringe Korrelation (durchschnittlicher Korrelationskoeffizient über die ersten sieben Sequenzmengen von 0.15 $\pm$ einer Standardabweichung von 0.08), H-Maß und Homologie eine deutlich höhere (0.36 bzw. -0.36 ± 0.07), während die H-Distanz dazwischen liegt (0.23 ± 0.06). Die Edit-Distanz und die drei Varianten der globalen Alignments sind vergleichbar mit H-Maß und Homologie (0.36 ± 0.06, -0.38 ± 0.06, -0.39 ± 0.06, -0.40 ± 0.06). Die komplexitätsbasierten Maße schneiden etwas schlechter ab ($0.30 \pm$

0.08 und 0.32 ± 0.09). Die Beträge der Korrelationskoeffizienten der L-Tupel-basierten Maße decken im Mittel den Bereich von 0.22 ± 0.05 bis 0.51 ± 0.05 ab. Bei allen fünf Maßen für ein fixes L sank die Korrelation mit wachsender Tupellänge. Die euklidischen bzw. standardisierten euklidischen Maße korrelieren schwächer als die anderen L-Tupel-Maße. Die beiden kumulierten Maße d^2 und d^{SE*} sind vergleichbar mit den besten Einzelmaßen, die sie kumulieren ($d^E(4)$ und $d^{SE}(4)$).

Für die 6-Tupel-Maße angewandt auf 8-mere (die freien Felder in Tabelle 4.1) ließen sich keine Korrelationkoeffizienten berechnen, da die Varianzen dieser Distanzmaße gleich Null waren, d. h. alle 500 Messungen einer Distanz ergaben denselben Wert. Tabelle 4.2, die auflistet, wieviele verschiedene Distanzen pro Maß und Sequenzmenge gemessen wurden, zeigt, daß die meisten L-Tupel-Maße (außer den standardisierten euklidischen) für kurze Sequenzen nur wenig verschiedene Distanzen messen. Bei Hamming, H-Maß, H-Distanz, Homology und Edit-Distanz gilt dies nicht nur für kurze sondern auch für längere Sequenzen.

Dies ist ein Grund für die insgesamt schwache Korrelation. Wie Tabelle 4.2 und Abbildung 4.5 zeigen, korrespondiert jeder Wert eines dieser Distanzmaße mit vielen freien Enthalpien. Allerdings kann dies nicht der alleinige Grund sein, da die Anzahl unterschiedlicher Distanzen bei den meisten Maßen mit der Sequenzlänge zunimmt, die Korrelation aber nicht stärker wird. Zudem wurde die stärkste Korrelation (z. B. d_4^{cos} angewandt auf 10-mere) mit gerade mal 7 verschiedenen Werten erlangt, die schwächste (Hamming angewandt auf 25-mere) mit 13. Zumindest sind die Korrelationskoeffizienten für diese Fälle als nicht sehr aussagekräftig zu betrachten.

H-Maß und H-Distanz Tabelle 4.3 zeigt die Korrelationskoeffizienten zwischen der Hamming-Distanz, dem H-Maß, der H-Distanz und der Homologie untereinander. Der Korrelationskoeffizient zwischen der H-Distanz und dem H-Maß ist deutlich kleiner als 1 (0.55 – 0.65). Erst beim Vergleich von Sequenzen verschiedener Länge (Spalte 10–30-mere) ist die Korrelation stärker. Eine ähnlich starke Korrelation besteht zwischen Hamming-Distanz und H-Maß für die 10- bis 30-mere. H-Maß und Homologie korrelieren für den Vergleich gleichlanger Sequenzen mit r = −1, nur beim Vergleich von Sequenzen verschiedener Länge wird die Korrelation schwächer (r = −0.62).

Edit-Distanz und globales Alignment Alle drei Varianten des globalen Alignments und die Edit-Distanz (die eigentlich auch ein Spezialfall des globalen Alignments ist, s. o.) korrelieren untereinander für die Vergleiche gleichlanger Sequenzen sehr stark ($0.87 \leq |r| \leq 0.99$, s. Tab. 4.4). Beim Vergleich unterschiedlich langer Sequenzen zeigt sich sowohl weiterhin starke (Edit-Distanz gegen NW(1,-2), $r = -0.90$) als auch deutlich schwächere Korrelation (Edit-Distanz gegen NW(2,-1), $r = -0.19$). Die Korrelation des H-Maßes mit den vier Alignment-Varianten sinkt mit zunehmender Sequenzlänge von $r \approx 0.7$ für 8-mere auf $r \approx 0.5$ für 50-mere. Beim Vergleich von Sequenzen unterschiedlicher Länge zeigt sich auch hier eine breite Spanne von Korrelationskoeffizienten (von −0.04 gegen NW(2,-1) bis 0.92 gegen Edit-Distanz).

Komplexitätsbasierte Maße Die beiden auf der Lempel-Ziv-Komplexität beruhenden Maße d^* und d_1^{**} korrelieren auf allen acht Mengen recht stark mit im Mittel $r = 0.86 \pm 0.02$ (Tab. 4.5).

L-Tupel-basierte Maße Die L-Tupel-basierten Distanzmaße korrelieren untereinander recht gut, in fast allen Fällen gilt $|r| \geq 0.5$ (Daten aus Platzgründen hier nicht wiederholt,

Länge	8	10	15	20	25	30	50	10–30
Hamming	7	8	10	11	13	14	20	23
H-Maß	6	6	7	7	8	8	12	22
H-Distanz	5	5	8	8	8	10	11	22
Homologie	6	6	7	7	8	8	12	86
Edit-Distanz	7	9	9	9	11	10	12	18
NW (1, -2)	16	20	23	25	28	29	35	35
NW (1, -1)	11	14	17	18	19	19	24	23
NW (2, -1)	15	18	22	25	27	28	33	31
d^*	8	11	16	15	17	18	22	31
d_1^{**}	24	25	35	35	44	43	50	70
d_4^E	7	9	12	13	17	21	32	48
d_5^E	3	6	8	8	11	14	18	43
d_6^E	1	3	6	6	7	9	10	42
$d^2(4,6)$	28	34	50	64	75	95	135	195
d_4^{SE}	68	118	251	344	405	454	488	491
d_5^{SE}	19	100	225	299	400	440	469	492
d_6^{SE}	1	26	154	233	306	369	454	498
$d^{SE*}(4,6)$	474	414	455	471	485	487	488	499
d_4^{LCC}	9	12	30	46	91	137	340	454
d_5^{LCC}	3	6	10	16	25	35	97	330
d_6^{LCC}	1	3	6	8	9	14	24	257
d_4^{cos}	6	7	20	36	75	117	338	215
d_5^{cos}	2	3	5	11	17	24	83	80
d_6^{cos}	1	2	3	6	6	9	17	25
d_4^{Evol}	6	7	20	36	75	117	339	215
d_5^{Evol}	2	3	5	11	17	24	83	80
d_6^{Evol}	1	2	3	6	6	9	17	25
ΔG_{25}	172	239	364	408	431	425	459	403

Tabelle 4.2: Anzahl verschiedener Meßergebnisse bei jeweils 500 Messungen.

Länge	8	10	15	20	25	30	50	10–30
Hamming vs. H-Maß	0.42	0.40	0.26	0.25	0.24	0.35	0.20	0.91
Hamming vs. H-Distanz	0.20	0.22	0.15	0.07	0.14	0.23	0.10	0.91
Hamming vs. Homologie	-0.42	-0.40	-0.26	-0.25	-0.24	-0.35	-0.20	-0.44
H-Maß vs. H-Distanz	0.55	0.59	0.60	0.55	0.60	0.63	0.65	0.98
H-Maß vs. Homologie	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-0.62
H-Distanz vs. Homologie	-0.55	-0.59	-0.60	-0.55	-0.60	-0.63	-0.65	-0.56

Tabelle 4.3: Korrelationskoeffizienten der Maße Hamming-Distanz, H-Maß, H-Distanz und Homologie untereinander.

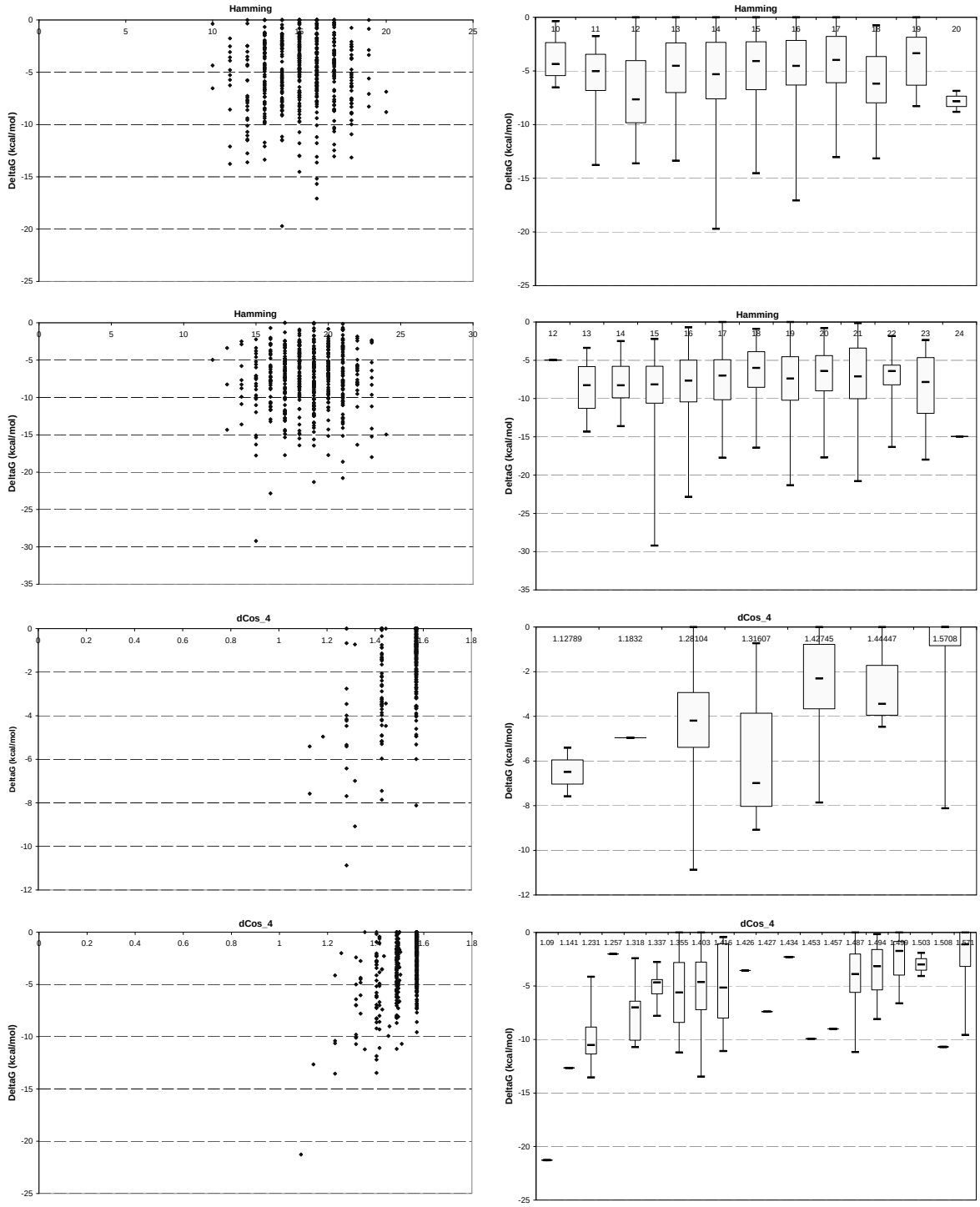


Abbildung 4.5: Scatter- und Boxplots der Messungen mit den zwei schwächsten und zwei der stärksten Korrelationen. Von oben nach unten: Hamming-Distanz über 20-mere und über 25-mere ($r = 0.06$), d_4^{cos} über 10-mere und über 15-mere ($r = 0.57$)

8-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.69	-0.72	-0.72	-0.71
Edit-Dist.		-0.97	-0.94	-0.89
NW(1,-2)			0.99	0.97
NW(1,-1)				0.99

10-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.67	-0.71	-0.71	-0.71
Edit-Dist.		-0.97	-0.94	-0.90
NW(1,-2)			0.99	0.97
NW(1,-1)				0.99

15-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.57	-0.59	-0.58	-0.57
Edit-Dist.		-0.97	-0.92	-0.87
NW(1,-2)			0.99	0.96
NW(1,-1)				0.99

20-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.57	-0.59	-0.59	-0.59
Edit-Dist.		-0.97	-0.94	-0.89
NW(1,-2)			0.99	0.96
NW(1,-1)				0.99

25-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.50	-0.50	-0.50	-0.49
Edit-Dist.		-0.97	-0.93	-0.89
NW(1,-2)			0.99	0.96
NW(1,-1)				0.99

30-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.58	-0.55	-0.54	-0.52
Edit-Dist.		-0.97	-0.93	-0.88
NW(1,-2)			0.99	0.96
NW(1,-1)				0.99

50-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.49	-0.50	-0.48	-0.47
Edit-Dist.		-0.96	-0.92	-0.88
NW(1,-2)			0.99	0.96
NW(1,-1)				0.99

10-30-mere	Edit-Dist.	NW(1,-2)	NW(1,-1)	NW(2,-1)
H-Maß	0.92	-0.77	-0.48	-0.04
Edit-Dist.		-0.90	-0.64	-0.19
NW(1,-2)			0.91	0.59
NW(1,-1)				0.87

Tabelle 4.4: Korrelationskoeffizienten der Maße H-Maß, Edit-Distanz und globalem Alignment untereinander. Bei den mit dem Algorithmus von Needleman-Wunsch (NW) berechneten Alignments sind in Klammern die Belohnung für Watson-Crick-Basenpaare sowie die Strafe für Mismatches und Lücken angegeben.

Länge	8-mere	10-mere	15-mere	20-mere	25-mere	30-mere	50-mere	10-30-mere
r	0.86	0.86	0.88	0.87	0.86	0.87	0.83	0.82

Tabelle 4.5: Korrelationskoeffizienten der beiden komplexitätsbasierten Distanzmaße d^* und d_1^{**} untereinander.

siehe Tabelle im Anhang). Insbesondere korrelieren Maße mit gleicher Tupellänge besonders stark.

Paare unterschiedlich langer Sequenzen Für die Hamming-Distanz, H-Maß und H-Distanz sinkt die Korrelation mit der freien Enthalpie über die Menge der 10- bis 30-mere ins Negative (auf jeweils $r = -0.29, r = -0.25$ und $r = -0.27$, s. Tab. 4.3). Diese Maße korrelieren für diese Menge untereinander sehr stark ($r \geq 0.89$, s. Anhang). Bei der Korrelation der Edit-Distanz und der drei globalen Alignments mit der freien Enthalpie zeigen sich sowohl Verbesserungen als auch Verschlechterungen gegenüber den Mengen mit fixer Sequenzlänge. Auch die Korrelation dieser Maße untereinander zeigt nun eine größere Breite, sie geht von $r = -0.19$ (Edit-Distanz vs. NW(2,-1)) bis $r = 0.91$ (NW(1,-2) vs. NW(1,-1)). Die komplexitätsbasierten Maße korrelieren nun nur etwas schwächer mit der freien Enthalpie. Bei den L-Tupel-basierten Maßen schlagen die Korrelationen der euklidischen Distanzen ins Negative um, die standardisierten euklidischen Distanzen verschlechtern sich stark, bleiben aber knapp positiv, während d_L^{LCC}, d_L^{cos} und d_L^{Evol} nur etwas schwächere Korrelation mit der freien Enthalpie zeigen.

4.2.4 Diskussion

Distanzmaße vs. freie Enthalpie Da für viele Distanzmaße pro Menge von 500 Messungen nur viel weniger als 500 verschiedene Distanzwerte gemessen wurden (Tab. 4.2), sind die entsprechenden Korrelationskoeffizienten mit Vorsicht zu betrachten. Trotz der vielen Messungen ist die Anzahl der Freiheitsgrade in der entsprechenden Meßreihe nur klein, der Korrelationskoeffizient also nur bedingt verläßlich. Dies zeigt aber auch ein grundsätzliches Problem dieser Distanzmaße. Eine große Menge unterschiedlicher Sequenzpaare mit sehr unterschiedlichen Duplexstabilitäten werden auf dieselbe Distanz abgebildet (Abb. 4.5). Wenn man die

Bereiche der freien Enthalpie betrachtet, die auf jeweils einen Distanzwert abgebildet werden, so sieht man, daß diese Bereiche sich weitgehend überschneiden, eine auch nur halbwegs eindeutige Zuordnung von freier Enthalpie zu Distanz ist also nicht möglich. Diese Distanzmaße abstrahieren also zu stark von der chemischen Realität und erscheinen als Maß für die Hybridisierungswahrscheinlichkeit ungeeignet.

Bei einigen Distanzmaßen steigt die Anzahl unterschiedlicher gemessener Distanzwerte mit der Sequenzlänge (Tab. 4.2), so daß die Korrelationskoeffizienten zumindest für längere Sequenzen als aussagekräftig anzusehen sind. Diese bestätigen in etwa die Korrelationskoeffizienten, die bei kurzen Sequenzen gemessen wurde (s. z. B. d_6^{SE}). Die Distanzmaße, die recht gut abschneiden, also mit einem Korrelationskoeffizient von ca. 0.5 (dies sind d_4^{LCC}, d_4^{cos} und d_4^{Evol}), zeigen auch im Scatter- bzw. Boxplot eine Tendenz (Abb. 4.5), die dem erwarteten Zusammenhang (Abb. 4.3) zumindest grob entspricht. Sie sind somit für eine Grobsortierung von Sequenzkandidaten im Sequenzdesign verwendbar, modellieren die Hybridisierungswahrscheinlichkeit aber auch nur sehr ungenau.

Es entspricht der theoretischen Überlegung, daß die Hamming-Distanz die geringste Korrelation mit der freien Enthalpie zeigt, da sie keinerlei Freiheit bei der Ausrichtung der beiden Sequenzen zueinander gewährt. Läßt man Verschiebungen der Sequenzen gegeneinander zu (s. H-Maß und Homologie), so verbessert sich die Korrelation deutlich, was ebenfalls der theoretischen Betrachtung entspricht. Interessanterweise führt die Hinzunahme von Lücken (s. Edit-Distanz und globale Alignments) im Mittel über alle sieben Mengen zu keiner weiteren Verbesserung, obwohl dies eine realistischere Modellierung der Hybridisierung zweier Sequenzen erlaubt. Genauer betrachtet ist die Korrelation von Edit-Distanz und globalem Alignment bei langen Sequenzen (30- und 50-mere) aber durchaus höher als die von H-Maß und Homologie, bei kurzen Sequenzen dagegen geringer. Da eine Lücke im Wesentlichen eine Verschiebung von Teilsequenzen gegeneinander zur Folge hat, ist dies auch anschaulich nachvollziehbar, da bei kurzen Sequenzen eine Verschiebung von Teilsequenzen keinen allzu großen Unterschied gegenüber der Verschiebung der gesamten Sequenzen macht, während die Verschiebung von Teilsequenzen bei langen Sequenzen die Vielfalt der möglichen Anordnungen stark erhöht. Dies zeigt sich auch in der Korrelation zwischen dem H-Maß und der Edit-Distanz bzw. dem globalen Alignment, die mit zunehmender Sequenzlänge schwächer wird ($|r| \approx 0.7$ für 8-mere, $|r| \approx 0.5$ für 50-mere, Tab. 4.4). Daß die H-Distanz schwächere Korrelation zeigt als das H-Maß, liegt an der Einführung der Poligos. Diese hat zur Folge, daß die H-Distanz mal den Abstand zu einer Sequenz und mal zu deren Komplement mißt, während das H-Maß immer den Abstand zum Komplement mißt, wie es die Modellierung der Hybridisierung erfordert.

Die komplexitätsbasierten und die L-Tupel-basierten Distanzmaße sind in der Modellierung insofern weiter vom Hybridisierungsprozeß entfernt als die bisher betrachteten Maße, als daß sie kein Alignment, also keine Anordnung der beiden Sequenzen zueinander, betrachten. Dennoch sind die Korrelationskoeffizienten z. T. mit denen der Alignment-basierten Maße vergleichbar. Die Abnahme der Korrelation der L-Tupel-Maße mit wachsendem L zeigt, daß die Berücksichtigung kurzer Teilsequenzen wichtiger ist als die längerer Teilsequenzen. Dies ist nachvollziehbar, da z. B. eine gemeinsame Subsequenz der Länge 6 sich auch als drei gemeinsame Subsequenzen der Länge 4 äußern, umgekehrt aber eine gemeinsame Subsequenz der Länge 4 in den 6-Tupel-Maßen ignoriert wird. Maße mit kürzeren Tupeln sind also sensibler gegenüber kürzeren Abschnitten, die hybridisieren könnten. Besonders ungeeignet sind Statistiken über Tupel, die nicht viel kürzer sind als die zu vergleichenden Sequenzen (s. die 6-Tupel-Maße angewandt auf 8-mere). Hierbei enthält jede Sequenz nur sehr wenige Tupel (in diesem Fall je drei) von vielen möglichen (hier $4^6 = 4096$), daher ist die Wahrscheinlichkeit, daß überhaupt zwei Sequenzen mindestens ein Tupel gemeinsam haben, gering, die c-Vektoren

sind also mit hoher Wahrscheinlichkeit gleich lang und stehen zueinander orthogonal. Daher ist auch keine Varianz der Distanzen zu erwarten.

Gibt man die Tupellänge nicht vor, sondern läßt Tupel unterschiedlicher Länge zu, welche der Algorithmus selbst findet, wie dies bei den Lempel-Ziv-basierten Maßen der Fall ist, so scheint dies keinen Vorteil gegenüber der Statistik über kurze Tupel fester Länge zu bringen. Dies könnte daran liegen, daß gemeinsame Subsequenzen unabhängig von ihrer Länge bewertet werden, obwohl mit der Länge die Stabilität der Hybridisierung wächst.

H-Maß und H-Distanz Die eher schwache Korrelation zwischen H-Maß und H-Distanz resultiert aus dem oben angeführten Grund für die unterschiedliche Korrelation beider Maße zur freien Enthalpie. Die stärkere Korrelation, die beim Vergleich von Sequenzen unterschiedlicher Länge zu beobachten ist, resultiert aus der Dominanz der langen ungepaarten Subsequenzen an den Enden der längeren Sequenz. Die tatsächliche Ähnlichkeit der kürzeren Sequenz mit einem Teil der längeren verliert also an Gewicht. Damit spielt es auch eine geringere Rolle, welche der beiden Sequenzen eines Polimers die H-Distanz bestimmt. Der gleiche Grund erklärt die starke Korrelation zwischen Hamming-Distanz und H-Maß für die 10- bis 30-mere. Die starke Korrelation von H-Maß und Homologie ist nicht überraschend, da die Homologie ein längennormalisiertes H-Maß ist. Beim Vergleich von Sequenzen unterschiedlicher Länge nimmt diese Korrelation ab, da die Homologie unabhängig von der Sequenzlänge nie größer als 1 wird, während längere dangling Ends zu einem höheren H-Maß führen.

Edit-Distanz und globales Alignment Die starke Korrelation zwischen den vier Alignment-Varianten (incl. der Edit-Distanz) ist durchaus nicht selbstverständlich. Die verschiedenen Parameterisierungen führen nicht nur zu unterschiedlichen Bewertungen gleicher Alignments, sondern i. a. auch zu unterschiedlichen optimalen Alignments. Die hier untersuchten Unterschiede der Parameter sind aber anscheinend zu gering, um sich in den Ergebnissen bemerkbar zu machen.

Die sinkende Korrelation des H-Maßes mit den Alignment-Varianten bei wachsender Sequenzlänge resultiert aus der zunehmenden Bedeutung von Lücken im Alignment (s. o.). Die sehr großen Unterschiede in den Korrelationen der fünf Maße untereinander für die Menge der 10- bis 30-mere beruhen auf den unterschiedlichen Bewertungen von Matches gegenüber Mismatches und Lücken. H-Maß, Edit-Distanz und NW(1,-2) bestrafen Mismatches und Lücken stärker als sie Matches belohnen, daher korrelieren sie noch recht stark untereinander. Bei NW(2,-1) ist es genau umgekehrt, die Matches wirken sich stärker auf die Distanz aus, daher ist die Korrelation mit den anderen Maßen nur gering.

Komplexitätsbasierte Maße Die durchgehend starke Korrelation der beiden komplexitätsbasierten Distanzmaße ist nicht überraschend, da die eigentliche Ähnlichkeitsmessung bei der Berechnung der Komplexitäten der Sequenzen und ihrer Konkatenationen geschieht, welche von den beiden Maßen dann nur noch unterschiedlich zu je einer Zahl akkumuliert werden.

***L*-Tupel-basierte Maße** Ähnlich wie bei den komplexitätsbasierten Maßen ist auch bei den L-Tupel-basierten Maßen bei gleicher Tupellänge L die eigentliche Ähnlichkeitsmessung, nämlich die Erhebung der L-Tupelhäufigkeiten bzw. -frequenzen, für alle Maße gleich und wird nur mit verschiedenen Methoden zu einer Zahl zusammengefaßt. Die starke Korrelation ist hier also zu erwarten gewesen. Auch Maße verschiedener Tupellängen zeigen noch einen deutlichen

Zusammenhang, da sich ein längerer gemeinsamer Tupel im Maß mit größerem L auch als Anzahl kürzerer gemeinsamer Tupel im Maß mit kleinerer Tupellänge zeigt.

Paare unterschiedlich langer Sequenzen Die starke Korrelation zwischen Hamming-Distanz, H-Maß und H-Distanz für die Menge der 10- bis 30-mere resultiert aus dem Einfluß der dangling Ends, die bei diesen Maßen gleich stark bestraft werden. Dieser Einfluß dominiert bei diesen Sequenzlängen sogar die Unterschiede zwischen einer Sequenz und deren Komplement, die beim Vergleich gleich langer Sequenzen zu einer deutlich schwächeren Korrelation zwischen H-Maß und H-Distanz führt. Die unterschiedlichen Längen nicht nur innerhalb eines Paares, sondern auch zwischen verschiedenen Sequenzpaaren, erklären auch die schlechtere Korrelation gegenüber der freien Enthalpie. Ein längeres Sequenzpaar hat mehr Möglichkeiten für Mismatches und ungepaarte Basen, wird also in der Erwartung mit einer höheren Distanz bewertet werden, da die ebenfalls häufiger werdenden Matches nicht belohnt werden, also nicht in die Distanz eingehen. Gleichzeitig sind Hybridisierungen längerer Moleküle aber tendentiell stabiler, was kleinere Werte für die freie Enthalpie bedeutet. Für die Modellierung der Hybridisierung unterschiedlich langer Sequenzen sind diese Maße also völlig ungeeignet.

Diese Überlegung wird durch die vier Alignment-Varianten (incl. Edit-Distanz) gestützt. Auch hier verschlechtert sich die Korrelation bei den Maßen, die Mismatches und Lücken stärker betonen als Matches (Edit-Distanz und NW(1,-2)), während sie sich bei gleicher Gewichtung nicht stark verändert (NW(1,-1)) und bei stärkerer Betonung der Matches sogar etwas besser wird (NW(2,-1)).

Die leichte Verschlechterung der Korrelation der letzten drei L-Tupel-basierten Maße mit der freien Enthalpie beruht auf einem gewissen Mindestabstand, mit dem unterschiedlich lange Sequenzen auf jeden Fall bewertet werden, da sie aus unterschiedlich vielen L-Tupeln bestehen. Ähnliches gilt für die komplexitätsbasierten Maße. Die negativen Korrelationen bei den euklidischen Distanzmaßen begründen sich ähnlich wie oben. Je länger die Sequenzen sind, desto kleiner die zu erwartende freie Enthalpie, desto mehr Dimensionen gibt es aber auch für die c-Vektoren, sich zu unterscheiden, wodurch der Abstand größer wird. Die standardisierte euklidische Distanz kann diesen Effekt zum Teil abfangen, da mit wachsender Sequenzlänge auch die Varianz, durch die geteilt wird, zunimmt.

Fazit Die Hamming-Distanz ist zur Abschätzung der Hybridisierungswahrscheinlichkeit völlig ungeeignet. Ihre Erweiterungen H-Maß und Homologie sind wesentlich realistischer, leiden aber auch wesentlich unter der Abbildung vieler verschiedener freier Enthalpien auf sehr wenige Distanzen. Die Berücksichtigung von Lücken, wie bei Edit-Distanz und globalem Alignment, bringt nur bei längeren Sequenzen Vorteile. Die Alignment-freien, auf Subsequenzstatistiken beruhenden Distanzmaße sind bei kurzer Subsequenzlänge zumindest für eine grobe Sortierung geeignet, was die im nächsten Kapitel ausgeführte Verwendung der n_b-Uniqueness für den in dieser Arbeit vorgestellten Sequenz-Design-Algorithmus motiviert. Bei Alignment-basierten Maßen sollten Matches mindestens ebenso stark gewichtet in die Distanz eingehen wie Mismatches und Lücken. Dangling Ends sollten, wenn überhaupt, nur längenunabhängig in die Distanz einfließen. Für die Modellierung der Hybridisierung von Sequenzen unterschiedlicher Länge sind die meisten untersuchten Maße noch weniger geeignet als für Sequenzen gleicher Länge.

4.3 Eigenschaften von Sequenzmengen

Da man im allgemeinen größere Mengen von Sequenzen generieren möchte, ist es auch sinnvoll, Eigenschaften zu betrachten, die nicht paarweise gemessen werden, sondern sich auf eine beliebig große Sequenzmenge beziehen. Dies können sowohl Kumulationen von paarweisen Eigenschaften sein (Energielücke, Computational Incoherence), als auch direkt sequenzmengenbezogene Eigenschaften (n_b-Uniqueness).

4.3.1 Einmaligkeit von Subsequenzen

Um zu vermeiden, daß Teilbereiche von DNA-Molekülen hybridisieren, sollten nicht nur die kompletten Sequenzen, sondern auch schon Subsequenzen nicht komplementär sein. Dies motiviert die Definition der n_b-*Uniqueness* [147, 54].

Definition 3 *Eine Menge $M \subset \Sigma^*, \Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, von DNA-Sequenzen heißt* n_b-unique *für ein $n_b \in \mathbb{N}$, wenn für alle Sequenzen $X \in M$ gilt: Ist $Y \in \Sigma^*$ Subsequenz von X mit $L_Y = n_b$, so folgt*

a) Es gibt keine Subsequenz Z der Länge n_b einer beliebigen Sequenz aus M mit $Z = Y$.

b) Es gibt keine Subsequenz Z der Länge n_b einer beliebigen Sequenz aus M mit $Z = \overline{Y}$.

Voraussetzung a) verhindert, daß sich zwei Sequenzen bereits in Teilen zu ähnlich sind, und so von einer dritten, zu einer der beiden komplementären Sequenz „verwechselt“ werden. Voraussetzung b) dient der Vermeidung von abschnittsweisen Hybridisierungen zweier Sequenzen. Je nach beabsichtigter Anwendung (und damit je nach konkreter Instanz des DNA-Sequenz-Design-Problems, siehe Abschnitt 3.3), kann es sinnvoll sein, nur a) oder nur b) zu fordern. Die „beliebige Sequenz“ aus der Definition schließt insbesondere auch die Sequenz X selbst mit ein.

Offensichtlich ist für die Vermeidung von Kreuzhybridisierungen n_b zu minimieren. Insbesondere sollte n_b aber deutlich kleiner als die Länge der kompletten Sequenzen sein, um eine genügend große Unähnlichkeit garantieren zu können. Betrachtet man z. B. die Sequenzen $X = \mathtt{ACGTGAGTGCA}$ und $Y = \mathtt{TGCACGCACGT}$, so ist die Menge $M = \{X, Y\}$ wegen des A-G-Mismatches in der Mitte 6-unique, der Duplex $\frac{\mathtt{ACGTGAGTGCA}}{\mathtt{TGCACGCACGT}}$ ist aber nicht viel weniger stabil als der entsprechende perfekte Duplex ohne Mismatch. Allgemein haben zwei n_b-unique Sequenzen der Länge n_s schlechtestenfalls $\left\lfloor \frac{n_s}{n_b} \right\rfloor$ Mismatches.

Ein wesentlicher Vorteil der n_b-Uniqueness ist die Existenz von Algorithmen zur systematischen Konstruktion von Sequenzmengen mit dieser Eigenschaft (siehe nächstes Kapitel).

4.3.2 Erweiterung auf Konkatenationen

In vielen Anwendungen werden DNA-Sequenzen aneinandergehängt, so daß längere Sequenzen entstehen. Zur theoretischen Erfassung von Sequenzunähnlichkeit von Konkatenationen wurden in [86, 77, 80] verschiedene Begriffe definiert und untersucht, u. a.:

- *complement-compliant*: Das Komplement einer Sequenz aus M kommt nicht als Subsequenz einer anderen Sequenz aus M vor.
- *comma-free*: Keine Sequenz aus M ist Subsequenz einer Konkatenation zweier beliebiger Sequenzen aus M.

- *complement-free*: Das Komplement einer Sequenz aus M ist nicht Subsequenz einer Konkatenation zweier beliebiger Sequenzen aus M.
- *complement-subword compliant*: Das Komplement einer Subsequenz der Länge k einer Sequenz aus M kommt nicht als Subsequenz in derselben Sequenz vor.
- *complement-k-code*: Das Komplement einer Subsequenz der Länge k einer Sequenz aus M ist Subwort keiner Sequenz aus M.

Bisher liegen nur wenige grundlegende theoretische Erkenntnisse zu Sprachen mit diesen Eigenschaften vor, z. B. daß sie für reguläre Sprachen entscheidbar, für kontext-freie Sprachen aber bereits unentscheidbar sind [77], und daß diese Eigenschaften i. a. nicht über die Konkatenation abgeschlossen sind [80]. Einige Methoden zur systematischen Konstruktion von Sequenzmengen wurden vorgeschlagen [77, 87], es ist jedoch fraglich, wie sinnvoll deren Anwendung in der Praxis ist (siehe nächstes Kapitel).

4.3.3 Thermodynamische Eigenschaften

Ein einfaches, aber rechenintensives Maß für die Erfassung des Stabilitätsunterschieds zwischen erwünschter und unerwünschter Hybridisierung ist die *Energielücke* [1]. Sei Σ_B die Menge der freien Enthalpien ΔG_B der beabsichtigten und Σ_I die Menge der freien Enthalpien ΔG_I der fehlerhaften Hybridisierungen. Dann ist die Energielücke

$$\delta F = \min_{\substack{\Delta G_B \in \Sigma_B, \\ \Delta G_I \in \Sigma_I}} (\Delta G_I - \Delta G_B) \tag{4.19}$$

die Differenz der freien Enthalpien der stabilsten unerwünschten und der instabilsten beabsichtigten Hybridisierung. Es werden also nur die schlechtesten Vertreter beider Gruppen betrachtet, die Verteilung der Stabilitäten aller anderen Hybridisierungen wird nicht berücksichtigt. Zur Vermeidung von Fehlhybridisierungen ist diese Lücke natürlich zu maximieren.

Die Gleichgewichtszustände sämtlicher möglicher Hybridisierungsformen werden bei der *Computational Incoherence* ξ berücksichtigt [135]. Diese gibt das Mengenverhältnis von fehlerhaft hybridisierten Molekülen zu allen hybridisierten Molekülen im Gleichgewichtspunkt der Reaktion an. Seien C_i^0 und C_j^0 die Startkonzentrationen der Sequenzen i und j, $Z_e^{i,j}$ die Summe der Gleichgewichtskonstanten für alle unerwünschten Konfigurationen zwischen den Sequenzen i und j, und $Z_c^{i,j}$ die Partition Function für alle Konfigurationen zwischen den Sequenzen i und j. Dann ist die Computational Incoherence

$$\xi = \frac{\sum_{i,j \geq i} C_i^0 C_j^0 Z_e^{i,j}}{\sum_{i,j \geq i} C_i^0 C_j^0 Z_c^{i,j}}. \tag{4.20}$$

Die Berechnung von ξ vereinfacht sich z. B. unter Annahme des staggering Zipper-Modells, wie es in [135] für kurze Oligos vorgeschlagen wird. Sind die Anfangskonzentrationen aller beteiligten DNA-Moleküle gleich, vereinfacht sich die Berechnung weiter zu

$$\xi = \frac{\sum_{i,j \geq i} Z_e^{i,j}}{\sum_{i,j \geq i} Z_c^{i,j}}. \tag{4.21}$$

Beide thermodynamische Eigenschaften sind theoretisch fundiert und, abhängig vom gewählten Modell zur Berechnung der freien Enthalpien und Gleichgewichtskonstanten, vermutlich realistischer als die Subsequenz-basierten. Beide sind aber nur zur Bewertung einer

bereits vorliegenden Sequenzmenge geeignet, es gibt (bisher) keinen Algorithmus, der die Einhaltung vorgegebener Grenzen bzgl. dieser Eigenschaften per Konstruktion garantiert, und es scheint schwierig, einen solchen Algorithmus zu entwerfen.

Kapitel 5

Sequenz-Design-Algorithmen anderer Gruppen

Im Verlauf des letzten Jahrzehnts haben im wesentlichen zwei Forschungsbereiche die Entwicklung von DNA-Sequenz-Design-Algorithmen und -Programmen motiviert und vorangetrieben. Das Sequenz-Design-Problem wurde schon früh als ein zentrales Problem des DNA-Computing erkannt, um die Fehlerrate der *in vitro*-Berechnungen zu minimieren. Die Genomforschung dagegen verlangte nach zuverlässigen PCR-Primern und Sonden für Microarrays. Für beide Bereiche sind eine Reihe von Designprogrammen entstanden, die sowohl allgemeine Suchverfahren wie Evolutionäre Algorithmen als auch speziell für das Sequenzdesign entwickelte Algorithmen verwenden.

5.1 Typen von Algorithmen

Einige generelle Suchalgorithmen werden häufiger für das Sequenzdesign verwendet und sollen zunächst allgemein beschrieben werden.

Der Begriff *Evolutionärer Algorithmus* (EA) bezeichnet eine Klasse von Such- und Optimierungsverfahren, die wichtige Mechanismen der natürlichen Evolution imitieren, um Lösungen für ein gegebenes Problem zu „evolvieren“ [20, 18]. Dabei verwendet man eine Menge von Lösungskandidaten (eine Population von Individuen). Diese können sich vermehren, d. h. Kopien von sich selbst erzeugen, wobei die Nachkommen durch zufällige Mutationen (punktuelle Veränderungen der Lösung) sowie durch Rekombination (Austausch von „Teillösungen“) verändert werden. Eine Fitness-Funktion bewertet die Eignung der Kandidaten als Lösung für das gegebene Problem, eine auf dieser Fitness basierende Selektion sorgt dafür, daß die besseren Lösungskandidaten überleben und weitere Nachkommen produzieren, während die schlechteren aussterben, also aus der Population entfernt werden. EA werden in vielen Anwendungen aus verschiedenen Gebieten erfolgreich eingesetzt. Für das DWD stellt jedes Individuum eine Menge von Sequenzen dar. Hier wird sehr häufig eine bestimmte EA-Variante angewandt, *Genetische Algorithmen* (GA). Bei einem GA sind die Individuen Bitstrings, was der Kodierung von DNA-Sequenzen entgegenkommt, da man jede Base mit 2 Bit darstellen kann.

Simulated Annealing (SA) ist ein EA-ähnliches Verfahren, bei der nur ein Individuum mit einem Nachkommen verglichen wird, meist das bessere und gelegentlich (zufällig) das schlechtere Individuum übernommen wird und die Stärke der Veränderungen durch die Mutation über die Laufzeit abnimmt.

Bei einem *Adaptive Walk* wird eine meist zufällig gewählte Initiallösung immer wieder mu-

tiert, hier also z. B. eine Sequenz aus der Menge durch eine andere, zufällig erzeugte ersetzt. Ist die mutierte Lösung besser als die alte bzgl. eines Fitnesskriteriums, so wird sie als aktuelle Lösung übernommen, anderenfalls wird die alte Lösung beibehalten. Dieser Zyklus aus Mutation und Selektion wird in einer festgelegten Anzahl wiederholt.

Eine stochastische lokale Suche ist ähnlich wie ein Adaptive Walk, allerdings wird bei der Selektion mit einer gewissen Wahrscheinlichkeit die schlechtere Lösung beibehalten. Dies soll die frühzeitige Konvergenz gegen lokale Minima verhindern.

5.2 DNA-Word-Design

Adleman wählte die von ihm im ersten DNA-Computing-Experiment verwendeten DNA-Sequenzen rein zufällig aus [2]. Da die Menge aller Sequenzen einer bestimmten Länge sehr groß ist, ist die Wahrscheinlichkeit, daß bei der Auswahl nur sehr weniger Sequenzen diese sehr unterschiedlich sind, vermutlich hoch. Allerdings gibt es bislang keine Untersuchungen zu dieser Wahrscheinlichkeit, und selbst bei kleinen Wahrscheinlichkeiten für eine schlechte Auswahl kann der Anwender Pech haben.

Eine zufällige Suche nach Sequenzen führen auch Arita et al. durch, überprüfen aber die Sequenzen auf bestimmte Eigenschaften (Vorkommen von Restriktionsschnittstellen, GC-Gehalt, Hamming-Distanz aller Sequenzpaare, Vorkommen von `TTT` oder `AAA` und starke Komplementarität an den $3'$-Enden), und verwerfen ungeeignete Sequenzen [17]. Somit ist wenigstens ein durch die Schwellenwerte der Kriterienbewertungen, ab denen eine Sequenz verworfen wird, definiertes Mindestmaß an Qualität garantiert, jedoch kann die Laufzeit beliebig lang werden.

Durch Wahl eines geeigneten Graphen läßt sich das DWD auf das Independend Set Problem abbilden, also der Suche nach einer (möglichst großen) Menge von Knoten, die nicht durch Kanten miteinander verbunden sind [43]. In diesem Graph entsprechen die Knoten Oligomeren einer bestimmten Länge, eine Kante zwischen zwei Knoten bedeutet eine Neigung der beiden verbundenen Oligomere zur Hybridisierung. Diese wird anhand der minimalen freien Enthalpie eines Duplex aus beiden Sequenzen geschätzt, die mit Hilfe von Dynamischer Programmierung und thermodynamischen Parametern des nearest-Neighbor-Modells berechnet wird. Überschreitet diese freie Enthalpie einen Schwellenwert, so wird eine Kante gesetzt. Zur Suche nach Independend Sets schlagen Deaton et al. einen greedy Algorithmus vor, der solange (zufällig gewählte) Knoten zur Ergebnismenge hinzufügt, bis kein weiterer Knoten mehr unabhängig, also mit keinem Knoten der Menge über eine Kante verbunden, wäre. Für das DWD-Problem haben die Autoren eine große Menge von Zufallssequenzen erzeugt, aus diesen wie beschrieben einen Graph konstruiert und mit dem greedy Algorithmus eine Menge von kreuzhybridisierungsfreien Sequenzen gesammelt. Da das Independend Set Problem NP-vollständig ist, wurde durch die Abbildung auch gezeigt, daß das DWD (zumindest in der in diesem Artikel verwandten Formulierung) ebenfalls NP-vollständig ist. Von der Abbildung auf einen Graphen abgesehen führt der Algorithmus auch nur Filtern von Zufallssequenzen durch.

Eine ebenfalls zufallsbasierte Suche wird von Faulhammer et al. eingesetzt, um geeignete RNA-Sequenzen für die Darstellung von Bitvektoren mit der Lipton-Kodierung ([102], s. Abschnitt 3.2.1) zu finden [52]. In allen 2^{10} möglichen Konkatenationen soll es höchstens 5 identische Basen in beliebigen Fenstern von 20 Basen Länge geben, kein Komplement einer Subsequenz der Länge 7 oder länger soll vorkommen, es werden keine Guanin-Basen verwendet, und die Schmelztemperatur soll bei 45 °C liegen. Das Programm PERMUTE startet mit Zufallssequenzen über {`A,C,U`} und mutiert diese durch Neuauswürfeln einzelner Basen solange, bis alle Kriterien erfüllt sind. Nicht nur kann dies schlechtestenfalls endlos dauern, auch die

ständigen Überprüfungen der Kriterien sind sehr rechenaufwendig.

Rose et al. haben in einem Genetischen Algorithmus (GA) die Computational Incoherence als Fitness verwendet [136, 137]. Deren Berechnung basiert auf einem erweiterten staggering Zipper-Modell (s. Abschnitt 2.2), das auch Single Base Mismatches und Single Base Bulges berücksichtigt. Ebenfalls einen GA, aber den Logarithmus der Computational Incoherence als Fitness mit dem einfachen staggering Zipper-Modell verwenden Garzon et al. in ihrer Software *Edna* [59].

Ein GA, der Sequenzmengen auf die Einhaltung von n_b-Uniqueness (s. Abschnitt 4.3.1) optimiert, kam bei Ruben et al. zum Einsatz [140]. Individuen werden bzgl. ihrer Fitness bestraft, wenn Subsequenzen bestimmter Länge mehrfach vorkommen oder Komplemente von Subsequenzen ebenfalls vorkommen. Varianten berücksichtigen auch die Übergangsbereiche bei Konkatenation der Sequenzen oder beschränken die Auswahl der Basen auf {A,C,T}. Außerdem können die Sequenzen mit Hilfe des Vienna RNA Package, einem Programmpaket zur Sekundärstrukturvorhersage [75], auf Sekundärstrukturen untersucht und diese bei der Fitness berücksichtigt werden.

Deaton et al. messen die Fitness in ihrem GA, indem sie zählen, wieviele Paare von Sequenzen der zu optimierenden Menge eine Kreuzhybridisierung ermöglichen [44, 45]. Diese wiederum wird bei Unterschreiten einer Hamming-Distanz-Schwelle angenommen.

Eine gewichtete Summe aus Bewertungen der Kriterien Vorkommen von Restriktionsschnittstellen, GC-Gehalt, Hamming-Distanz aller Sequenzpaare, Vorkommen von TTT oder AAA und starke Komplementarität an den 3′-Enden dient als Fitness für einen GA von Arita et al. [17].

Simulated Annealing (SA) wurde von Tanaka et al. eingesetzt [162]. Als Kriterien dienten Bewertungen der Sequenzen bezüglich des H-Maßes der Sequenzen zueinander und zu den Komplementärsequenzen, der Selbstkomplementarität von Sequenzen, der Abweichung von GC-Gehalt und Schmelztemperatur von einem vorgegebenen Wert, dem Aufeinanderfolgen identischer Basen sowie des Vorliegens komplementärer Subsequenzen an den 3′-Enden (falls die Sequenzen mit Polymerase verlängert werden sollen). Diese Bewertungen wurden als gewichtete Summe zu einem Fitness-Wert zusammengesetzt. Einen EA mit den gleichen Bewertungskriterien verwenden Shin et al., ebenfalls mit einer gewichteten Summe als Fitness [153], und Kim et al., diese allerdings mit einer nicht-dominierten Sortierung der Individuen [90]. Letztere realisiert eine Suche nach *pareto-optimalen* Lösungen, also Lösungen, die man bzgl. keines einzelnen Optimierungskriteriums mehr verbessern kann, ohne sie gleichzeitig bzgl. mindestens eines anderen Kriteriums zu verschlechtern. Im Idealfall bietet der EA nach Terminierung eine Menge sehr unterschiedlicher pareto-optimaler Lösungskandidaten an, die jeweils besser bzgl. verschiedener Kritierien sind, so daß der Benutzer unter diesen auswählen kann, je nachdem, welche Kriterien ihm wichtiger erscheinen.

Die Evolutionären Algorithmen scheinen ein erfolgreiches Werkzeug zum DNA-Word-Design darzustellen. Insbesondere die mehrkriteriellen Algorithmen sind interessant, da man die Sequenzen für viele Anwendungen auf mehr als nur eine Anforderung hin optimieren muß. Hierbei ist die Suche nach Lösungen auf der Pareto-Front nützlicher als der Einsatz einer gewichteten Summe als Fitness, da der Benutzer nach der Optimierung sehen kann, welche Lösung welches Kriterium wie gut erfüllt und danach eine geeignete Sequenzmenge auswählen kann. Die gewichtete Summe erfordert *a priori* die Festlegung der Gewichte, ohne daß der Benutzer wirklich abschätzen kann, wie sich rein numerische Unterschiede der einzelnen Kriterienbewertungen in Qualitätsunterschieden wiederspiegeln, insbesondere bei sehr unterschiedlichen Kriterien. Ein Problem der EA ist, daß jede im Verlauf der Evolution auftauchende Sequenzmenge bewertet werden muß, d. h. sämtliche relevanten Kriterien müssen immer wieder für

jede Menge neu berechnet werden. Dies kann bzgl. der Rechenzeit sehr teuer sein, so ist z. B. die Überprüfung der n_b-Uniqueness bei Ruben et al. sehr aufwendig [140]. Generell wäre es wünschenswert, gerade die Einhaltung der paar- bzw. mengenweisen Kriterien bereits in der Konstruktionsmethode zu erzwingen.

Mit einem stochastischen lokalen Suchalgorithmus haben Tulpan et al. die Anzahl der Verletzungen vorgegebener Beschränkungen minimiert [165, 166]. Diese betreffen die Hamming-Distanz der Sequenzen untereinander und zu ihren Komplementen, den GC-Gehalt, die Schmelztemperatur, die freie Enthalpie sowie das Vorkommen verbotener Subsequenzen. Bei der Suche wird immer wieder zufällig eine Sequenz ausgewählt, die mindestens eine der Beschränkungen verletzt, und von dieser alle Mutationen erzeugt, die sich um genau eine Base von der gewählten Sequenz unterscheiden. Die gewählte Sequenz wird durch den Mutanten mit der geringsten Anzahl an Beschränkungsverletzungen ersetzt, mit einer gewissen Wahrscheinlichkeit erfolgt die Ersetzung durch einen zufällig gewählten Mutanten.

Mit einem Adaptive Walk haben Penchovsky und Ackermann eine Bibliothek von 24 26-meren erzeugt [127]. Sie starten mit einer Zufallsmenge, wählen dann in jedem Zyklus eine Sequenz zufällig aus, erzeugen eine weitere Sequenz zufällig, und übernehmen die bessere der beiden wieder in die Menge, während die schlechtere verworfen wird. Kriterien für die Güte der Sequenzen sind Subsequenzen aus unmittelbar aufeinanderfolgenden identischen Basen (sollen nicht länger als 3 Basen sein), Vorliegen von mehr als zwei aufeinanderfolgenden Cytosin-Basen an den Enden, eine möglichst gleichmäßige Schmelztemperatur, sowie eine möglichst große Energielücke zwischen geplanten und unerwünschten Hybridisierungen. Geprüft werden dabei auch sämtliche möglichen Konkatenationen von Sequenzen. Ackermann und Gast haben einen Adaptive Walk mit der Energielücke, also die Differenz zwischen der freien Enthalpie der instabilsten gewünschten und der der stabilsten unerwünschten Hybridisierung, als Gütemaß eingesetzt, um 24 16-mere zu optimieren [1].

Adaptive Walk und stochastische lokale Suche sind als sehr einfache Varianten von EA sicherlich auch für das DWD nützliche Algorithmenklassen, da aber nur ein Lösungskandidat nach und nach verbessert wird, ist der explorierte Teil des Lösungsraums eher klein, die Gefahr einer Konvergenz gegen ein suboptimales lokales Optimum ist also höher als bei EA mit größeren Populationen.

DeBruijn-Sequenzen sind zyklische Sequenzen über einem beliebigen Alphabet Σ, in denen jede mögliche Subsequenz einer bestimmten Länge k genau einmal vorkommt [41]. Smith schlägt eine Erweiterung dieser Idee vor, so daß das Komplement einer vorkommenden Subsequenz nicht auch in der DeBruijn-Sequenz vorkommt [156]. Anschließend müßte man nur noch Sequenzen gewünschter Länge aus der modifizierten DeBruijn-Sequenz ausschneiden. Leider gibt Smith keinen Algorithmus zur Erzeugung solcher modifizierter DeBruijn-Sequenzen an. Denkbar wäre es, z. B. eine der Methoden aus [131] anzupassen und zu verwenden. Eine sehr einfache Methode beginnt mit einer Subsequenz der Länge k und fügt das jeweils größte Symbol an, so daß der neuentstandene Suffix der Länge k noch nicht in der Sequenz vorkommt („das größte Symbol“ heißt hierbei das Symbol an der höchsten Position gemäß einer beliebigen Sortierung aller Symbole des Alphabets). Dies wird iteriert, bis alle möglichen Sequenzen der Länge k als Subsequenz vorkommen, und somit eine DeBruijn-Sequenz generiert wurde. Durch Hinzufügen von Backtracking und der Wahl des jeweils nächstkleineren Symbols lassen sich mit diesem Verfahren alle möglichen DeBruijn-Sequenzen aufzählen. Ein anderen Algorithmus verwendet einen Graph, dessen Knoten mit allen möglichen Sequenzen der Länge $k-1$ markiert sind. Eine gerichtete Kante von einem Knoten v zu einem Knoten w ist mit Symbol $s \in \Sigma$ markiert, wenn sich die Markierung von w aus Anhängen von s an den Suffix von v der Länge $k-2$ ergibt. Z. B. führt eine mit `C` markierte Kante vom Knoten mit der Markierung

ACGT zum Knoten mit der Markierung CGTC. Da jeder Knoten genau $|\Sigma|$ eingehende und ebenso viele ausgehende Kanten hat, existiert mindestens ein Eulerpfad (ein Pfad, der jede Kante genau einmal durchläuft) dieses Graphen. Verfolgt man einen solchen Eulerpfad und notiert nacheinander die Kantenmarkierungen, so ergibt sich eine DeBruijn-Sequenz.

Die nur einmalige Verwendung von Subsequenzen einer bestimmten Länge wird auch in der in dieser Arbeit vorgestellten Design-Methode verwendet, ebenfalls ähnelt der graphbasierte Algorithmus dem hier benutzten (s. nächsten Abschnitt). Allerdings ist bei dem oben beschriebenen Algorithmus zur Erzeugung von DeBruijn-Sequenzen nicht klar, wie die von Smith vorgeschlagene Modifikation bzgl. der Komplemente der Subsequenzen mit den Eulerpfaden in Einklang gebracht werden kann. Entsteht bei einem Schritt, der durch den Graph gemacht wird, der k-Suffix X, so muß die Kante an anderer Stelle des Graphen, die dort bei Durchlauf den k-Suffix $\overline{X}$ entstehen lassen würde, „gesperrt“ werden, so daß $\overline{X}$ nicht in derselben modifizierten DeBruijn-Sequenz vorkommen kann wie X. Dadurch wird aber ein Eulerpfad zerrissen und kann nicht bis zum Ende verfolgt werden. Es ist auch nicht ersichtlich, ob ein Eulerpfad unter Ausschluß der jeweils „komplementären“ Kanten überhaupt noch existiert. Desweiteren ist nicht offensichtlich, wie weitere Sequenzeigenschaften wie z. B. die Schmelztemperatur berücksichtigt werden können.

Baum beschreibt in [25] ebenfalls eine Methode zur Erzeugung n_b-uniquer Sequenzmengen, bei der eine Sequenz immer weiter verlängert wird, die Auswahl der Basen zur Verlängerung dabei aber eingeschränkt werden. Sowohl diese Einschränkungen als auch die Überlegungen, die zu diesen Einschränkungen führen, sind recht kompliziert, erlauben ein Sequenzentwurf von Hand in Form einer Knobelei, es ist jedoch nicht ersichtlich, wie die Überlegungen und die sich daraus ergebende Entwurfsmethode automatisiert werden können.

Für universelle DNA-Microarrays verwenden Gerry et al. „Zip-Code-Sequenzen“, die als Verlängerung an den eigentlichen Sonden hängen, so daß auf dem Microarray die Komplementärsequenzen der Zip-Codes aufgebracht werden können [65]. Damit können die Microarrays für die Untersuchung beliebiger Sequenzen verwendet werden, es müssen nur jeweils neue Sonden, verlängert um die Zip-Codes, hergestellt werden. Das Problem der Sondensuche entspricht dem bei normalen Microarrays, das später behandelt wird, die Suche nach geeigneten Zip-Codes entspricht aber dem Word-Design-Problem. Die Autoren erstellen durch Filtern zunächst eine Menge von Tetrameren (Sequenzen der Länge 4), so daß alle Tetramere mindestens einen Hamming-Abstand von 2 zueinander haben. Die Komplemente von in der Menge vorhandenen Tetrameren werden ausgeschlossen, ebenso selbstkomplementäre Tetramere. Jeweils sechs Tetramere werden anschließend (nicht-überlappend) aneinandergehängt, wobei zwei so entstehende 24-mere sich um mindestens drei Tetramere unterscheiden. Somit ist ein Hamming-Abstand von mindestens 6 zwischen zwei beliebigen Zip-Codes gewährleistet, der Hamming-Mindestabstand der einzelnen Tetramere sorgt für eine gewisse Verteilung der Mismatches über die Sequenz. Im schlechtesten Fall liegen aber die drei zugelassenen gleichen Tetramere zweier Zip-Codes direkt hintereinander und führen so zu identischen Subsequenzen der Länge 12, also der halben Sequenzlänge. Außerdem sind die Zip-Codes und deren Komplemente auf dem Microarray nur so lange wiederverwendbar, wie die eigentlichen Sonden bzw. die nachzuweisenden Zielsequenzen nicht mit ihnen interferieren.

Die Template-Map-Methode beruht auf der klassischen Kodierungstheorie, genauer auf binären, fehlerkorrigierenden Codes [57]. Anstelle von DNA-Sequenzen erzeugt man zunächst eine (relativ kleine) Menge von Bitstrings einer bestimmten Länge l, die sogenannten *Templates*. Diese Menge wird z. B. so konstruiert, daß alle Strings einen Mindestabstand zueinander bzgl. der Hamming-Distanz oder einem für Bitstrings angepaßten H-Maß haben. Ein fehlerkorrigierender Code mit Bitstrings der Länge l, die den gleichen Mindestabstand zueinander haben,

Template	00110110
Map	10100101
Sequenz	TAGCAGCT

Abbildung 5.1: Beispiel für eine DNA-Sequenz, die aus einem Template- und einer Map-Bitstring erzeugt wird. Die Bits im Template legen fest, ob ein C-G- oder ein A-T-Basenpaar an die entsprechende Stelle kommen soll, die Bits in der Map bestimmen, ob jeweils die erste (C oder A) oder die zweite (G oder T) Base gewählt wird.

dient als Menge der *Maps*. Durch Abbildung aller möglichen Paare aus Template und Map auf eine DNA-Sequenz erhält man eine Sequenzmenge, deren Größe das Produkt der Größen von Template- und Map-Menge ist. Bei dieser Abbildung legt z. B. für eine Position i das i-te Bit des Templates fest, ob an Position i der DNA-Sequenz ein C-G- oder ein A-T-Basenpaar stehen soll, während das i-te Map-Bit bestimmt, ob die erste (also C bzw. A) oder die zweite Base (also G bzw. T) gewählt wird (Abb. 5.1). Dabei bleibt die minimale Hamming-Distanz unter dieser Abbildung erhalten. Damit teilt sich das DNA-Word-Design-Problem in zwei Teilprobleme, nämlich die Suche nach geeigneten Template- und Map-Mengen, wobei beide Mengen aber kleiner sind als die gesuchte DNA-Sequenz-Menge. Während man die fehlerkorrigierenden Codes der Map-Mengen mit Methoden der klassischen Kodierungstheorie konstruieren kann, bleibt das Designproblem für Templates bestehen. In bisherigen Veröffentlichungen wurde diese mit vollständiger Aufzählung und Filtern gefunden, was für die sehr kleine Anzahl und die geringe Länge (≤ 30 Basen) der dort gesuchten Sequenzen noch akzeptablen Aufwand bedeutet [16, 105]. Ben-Dor et al. konstruieren Templates aus DeBruijn-Sequenzen über dem Alphabet {S,W}, wobei S für ein G-C- und W für ein A-T-Basenpaar steht [26].

Arita hat mit zwei Templates und einem fehlerkorrigierenden Code mit 56 Bitstrings 112 Sequenzen der Länge 12 erzeugt, die mindestens vier Mismatches untereinander und zu ihren Komplementen, auch bzgl. aller Verschiebungen und Konkatenationen haben [15]. Mit Templates, die nach Vorüberlegungen, wie sie auszusehen haben, von Hand entworfen wurden, erzeugten Frutos et al. 108 8-mere, die mindestens 4 Mismatches untereinander und gegenüber ihren Komplementen haben, für das oberflächenbasierte DNA-Computing [57]. Li et al. demonstrierten die Erzeugung von Sequenzen der Länge $4k$ mit jeweils mindestens $2k$ Mismatches untereinander für $k = 2, 3, 4, 5$. Garzon et al. stellen eine auf einem Tensor-Produkt basierende Methode vor, um sogar mehr Sequenzen als das Produkt der Anzahlen von Templates und Maps erzeugen zu können [60]. Die Template-Map-Methode kann auch verwendet werden, um eine *bond-free*[1] Sequenzmenge zu erzeugen [87].

Die Template-Map-Methode vereinfacht das DWD Problem, indem es die Größe der Probleminstanz reduziert, da man aus zwei kleinen Sequenzmenge eine wesentlich größere erzeugen kann, unter Beibehaltung einer Mindest-Hamming-Distanz. Es bleibt offen, wie gut sich andere Kriterien, z. B. andere Distanzmaße, in dieser Strategie realisieren lassen. Außerdem bleibt das Problem der Suche nach geeigneten Templates bestehen, es ist durch die reduzierte Größe vielleicht einfacher geworden, aber noch nicht gelöst. Das im vorhergehenden Kapitel durchgeführte Experiment hat zudem gezeigt, daß die Hamming-Distanz zur Abbildung der Hybridisierungsneigung eher ungeeignet ist.

Es gibt eine Reihe von Vorschlägen zu Methoden, mit denen man theoretisch beliebig große

[1] Die Eigenschaft bond-free ist verwandt mit comma-free und complement-free (s. Abschnitt 4.3.2) und bezieht sich auf einen minimalen Hammingabstand zwischen Subsequenzen bestimmter Länge.

Sequenzmengen erzeugen kann, die complement-free sind oder ähnliche Eigenschaften erfüllen (s. Abschnitt 4.3.2) [77, 87, 81, 80], allerdings sind diese Methoden noch nicht in der Praxis erprobt worden. Außerdem enthalten größere Sequenzmengen natürlich auch längere Sequenzen, und es ist abzusehen, daß ab einer gewissen Länge die Eigenschaft „complement-free“ und deren Verwandte nicht ausreichen werden, um Kreuzhybridisierungen bzw. Sekundärstrukturen zu vermeiden.

Phan und Garzon haben einen Shuffle-Operator $\oslash$ definiert, mit dem sie Wortmengen erzeugen können [129]. Seien $x_1, x_2, \ldots, x_k$ Sequenzen der Länge m, dann ist $x_1 \oslash \ldots \oslash x_k = x_{11} \ldots x_{k1} x_{12} \ldots x_{k2} \ldots x_{1m} \ldots x_{km}$, wobei x_{ij} die j-te Base der Sequenz x_i ist. Der DWD-Algorithmus erzeugt zunächst die Menge S_m aller DNA-Sequenzen der Länge m. Dann wird für jede Sequenz aus S_m ihr Komplement aus S_m entfernt. Schließlich sei $S = \{x_1 \oslash \ldots \oslash x_\tau : \forall x_i \in S_m$ und $x_i \neq x_j$ falls $i \neq j\}$. Dann enthält S Sequenzen der Länge $n = \tau m$, die paarweise ein H-Maß von mindestens τ zueinander haben. In einer Variante läßt man nur Sequenzen mit dem GC-Gehalt w in die Menge S_m. Damit haben die Sequenzen in S einen GC-Gehalt von τw. Weitere Anforderungen an die Sequenzen müßten vermutlich durch Filtern der Sequenzen aus S implementiert werden.

Vollständige Aufzählung wurde von Hartemink et al. verwendet, um eine Sequenzmenge für eine spezielle Anwendung, die „programmierte Mutagenese“, bei der sich die Sequenzen nur geringfügig unterscheiden müssen, zu finden [71]. Ein von Hand entworfenes Muster von Mismatches, die die Sequenzen haben sollten, wurde vorgegeben, was die Anzahl der zu untersuchenden Sequenzmengen stark einschränkt. Gefiltert wurde nach Kriterien wie Schmelztemperatur, Unterschiede in der Schmelztemperatur, maximale Stabilität von Hairpin Loops und Kreuzhybridisierungen.

Mit Hilfe eines Eulergraphen finden Pancoska et al. zu einer gegebenen Sequenz eine Menge von Sequenzen mit gleicher Schmelztemperatur [125]. Der Graph enthält vier mit `A`, `C`, `G` und `T` markierte Knoten. Die Eingabesequenz wird durchlaufen, für jedes Nachbarpaar von Basen wird eine gerichtete Kante von dem Knoten, der mit der ersten Base des Paars markiert ist, zum Knoten, der mit der zweiten Base markiert ist, eingefügt. Schließlich wird eine solche Kante auch für das Paar aus der letzten und der ersten Base der Eingabesequenz eingefügt. Man kann leicht nachvollziehen, das der Graph einen Eulergraph bildet, da ein Durchlaufen der Kanten in der gleichen Reihenfolge, in der sie eingefügt wurden, einen Eulerpfad bildet. Findet man nun sämtliche anderen Eulerpfade, so bestehen diese aus den selben Kanten, die diesen Pfaden entsprechenden DNA-Sequenzen (die Knotenmarkierungen in der Reihenfolge, in der sie besucht werden) bestehen also aus den selben Nachbarpaaren von Basen und haben somit nach dem nearest-Neighbor-Modell dieselbe Schmelztemperatur. Leider dürften auch viele der so gefundenen Sequenzen sehr ähnlich sein, für eine tatsächliche Anwendung müssen diese also zunächst noch auf Kreuzhybridisierungen untersucht und selektiert werden.

Eine *in vitro*-Methode zur Suche nach kreuzhybridisierungsfreien Sequenzen wurde von Deaton et al. vorgeschlagen [42]. Zunächst befinden sich im Reagenzglas zufällige Sequenzen, die an beiden Enden mit vorgegebenen Primerbindungsstellen verlängert wurden. Die Idee besteht darin, daß kreuzhybridisierungsfreie Moleküle einzelsträngig bleiben und man sie daher mit PCR (s. Abschnitt 2.1) amplifizieren kann, während DNA-Stränge, die Kreuzhybridisierungen eingehen, zumindest teilweise doppelsträngig vorliegen und somit die Herstellung von Kopien durch die Polymerase verhindern. Hauptsächliches Problem dieses Ansatzes ist das Vorliegen von Sequenzen, die perfekt komplementär zu den kreuzhybridisierungsfreien Sequenzen sind, nach dem ersten Kopierzyklus. Um zu verhindern, daß diese Duplexe bilden und die kreuzhybridisierungsfreien Sequenzen nicht weiter amplifiziert werden können, muß die Reaktion bei einer so hohen Temperatur stattfinden, daß Kreuzhybridisierungen anderer Sequenzen

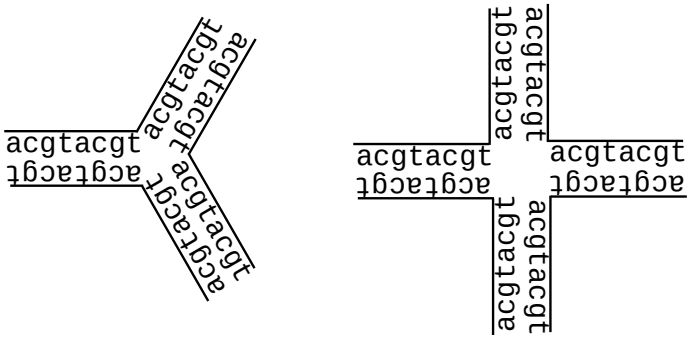


Abbildung 5.2: Beispiele für 3- und 4-armige DNA-Junctions. Die Linien deuten die Backbones an. Die Basen dienen nur zur Skizzierung und stellen keinen Entwurfsvorschlag dar.

erst recht unwahrscheinlich werden, und diese somit auch für die Amplifizierung zur Verfügung stehen.

5.3 Struktur-Design

Auf der n_b-Uniqueness beruht SEQUIN, ein Programm von Nadrian Seeman [147]. Dieses Programm dient zur Suche nach Sequenzen, die 3- und 4-armige Junctions bilden (Abb. 5.2). Es läßt zum Einen das Hinzufügen einzelner Basen zur zu konstruierenden Sequenz von Hand zu, und zeigt für jede hinzugekommene Base die Verletzungen der n_b-Uniqueness an. Zum Anderen fügt ein semi-automatischer Modus kurze Sequenzstücke ein, woraufhin der Benutzer diese bestätigen oder wiederum von Hand ändern kann. Verletzungen der n_b-Uniqueness werden angezeigt, aber zugelassen. Weitere Anforderungen an die Sequenzen, wie fixe Subsequenzen, ein bestimmter GC-Gehalt usw., sind prinzipiell möglich, müssen aber durch den Benutzer im Designprozeß realisiert werden.

Im Vienna RNA Package [75] finden sich nicht nur Programme zur Vorhersage und zum Vergleich, sondern auch zum Design von Sekundärstrukturen. *RNAinverse* findet zu einer vorgegebenen Sekundärstruktur eine RNA-Sequenz, die sich zu dieser Struktur falten wird. Der Algorithmus beruht auf einem Adaptive Walk, bei dem Basen, die noch nicht die gewünschte Rolle in der Struktur einnehmen, zufällig mutiert werden, und diese Mutation übernommen wird, wenn für die mutierte Sequenz eine Struktur vorhergesagt wird, die der Zielstruktur ähnlicher ist als die der Sequenz vor der Mutation. Da für jede Bewertung der Vorhersagealgorithmus aufgerufen werden muß, der auch in effizienter Implementierung noch eine Rechenzeit von $\Theta(n^3)$ benötigt, wobei n die Sequenzlänge ist, verwenden Hofacker et al. eine strukturelle Dekomposition, bei der die Zielstruktur in kleiner Teilstrukturen zerlegt und für diese der Design-Algorithmus rekursiv aufgerufen wird. Sind geeignete Sequenzen für die Teilstrukturen gefunden worden, so werden sie wieder zusammengesetzt und per Vorhersage überprüft, ob sich die zusammengesetzte Sequenz auch zur zusammengesetzten Struktur faltet. Ggf. müssen einzelne Teilstrukturen dann neu entworfen werden.

Andronescu et al. verwenden ebenfalls strukturelle Dekomposition und die fold-Funktion aus dem Vienna RNA Package für das gleiche Problem [13, 14], allerdings setzen sie eine stochastische lokale Suche statt eines Adaptive Walk ein, d. h. mit einer gewissen Wahrscheinlichkeit werden auch Mutationen, die zu schlechteren Strukturen führen, übernommen. Weitere Verbesserungen an der Dekomposition (Aufteilung in möglichst gleich große Teilstrukturen zur

Verringerung der Rekursionstiefe) und der Initialisierung der Sequenz erhöhen zum Einen die Ausbeute an gelösten Probleminstanzen und verringern zum Anderen die Rechenzeit, verglichen mit RNAinverse aus dem Vienna RNA Package [14].

Die beiden zuletzt genannten Algorithmen beziehen sich nur auf das Design von Sekundärstrukturen einzelsträngiger Moleküle, eine Erweiterung auf beliebige Strukturen aus mehreren Strängen sollte jedoch möglich sein. Erforderlich wäre dafür allerdings ein entsprechender Vorhersagealgorithmus, den es bisher nicht gibt. Ebenfalls einfach zu implementieren sein dürften weitere Kriterien wie fixe Subsequenzen, verbotene Subsequenzen, die Einschränkung der Basenauswahl usw.

5.4 Primer- und Microarray-Design

Es gibt zwei wesentliche Unterschiede bei der Suche nach PCR-Primern und Microarray-Sonden gegenüber der Erzeugung von DNA-Wörtern für das DNA-Computing. Erstens sollen sowohl Primer als auch Sonden an bestimmte Gene (oder andere bestimmte, bereits vorgegebene Sequenzen) hybridisieren, daher ist die Menge der Kandidaten für die Primer und Sonden von vornherein stark eingeschränkt und entspricht der Menge von Subsequenzen z. B. des zu identifizierenden Gens, wobei die Länge der Subsequenzen meist stark beschränkt ist. Zweitens gibt es eine i. a. sehr große Bibliothek von Sequenzen (andere Gene, intergenomische Sequenzen), mit denen die Primer bzw. Sonden nicht hybridisieren sollen. Für viele Organismen umfassen diese Bibliotheken mehrere Milliarden Basenpaare.

Die erstgenannte Eigenschaft legt einen grundlegenden Suchalgorithmus nahe, der auch tatsächlich meist angewandt wird: Ein Fenster bestimmter Größe läuft über die Kandidatensequenz (z. B. das Gen), die in diesem Fenster liegende Subsequenz wird auf ihre Eignung als Primer bzw. Sonde überprüft.

Primer-/Sonden-Kandidaten werden meist auf die folgenden Eigenschaften getestet:

a) keine Sekundärstrukturen der einzelnen Sequenzen

b) keine Hybridisierung der beiden Primer eines Paars

c) ggf. keine Hybridisierung zwischen Primern aus verschiedenen Paaren (für PCR) oder zwischen verschiedenen Zielsequenzen (für Microarrays)

d) gleiche Hybridisierungseffizienz beider Primer eines Paars oder mehrerer Zielsequenzen

e) Schmelztemperaturen nahe einer vorgegebenen Temperatur

f) keine Hybridisierung mit Sequenzen der Bibliothek der anderen Gene und intergenomischer Sequenzen, bei Microarray zumindest nicht mit anderen Sonden

g) kein Ausfransen am $3'$-Ende von Primern, da hier die Polymerase zur Verlängerung angreifen können muß.

Eigenschaften a) und b) (und ggf. c)) werden über die Hamming-Distanz [130], Alignment mit Dynamic Programming [85], oder thermodynamischen Vorhersageprogrammen wie mfold [167, 138] gemessen, d) anhand der Schmelztemperatur oder des GC-Gehalts abgeschätzt [130, 85], f) mit Alignment oder Alignment-basierten Programmen wie BLAST (z. B. [98, 138, 119]), nur selten mit einem thermodynamischen Modell [78, 84, 138] überprüft, und g) durch Verwendung von `G-C`-Basenpaaren am $3'$-Ende sichergestellt [37, 180]. Oft berücksichtigen Primer-Designprogramme nur eine echte Teilmenge dieser Eigenschaften.

Die Auswahl von geeigneten Primern erfolgt meist, indem Kandidaten verworfen werden, die bestimmte Schwellenwerte der gemessenen Eigenschaften überschreiten (z. B. [37]). Alternativ werden die Primer gewählt, deren Eigenschaftsvektor die geringste Distanz zu einem vom Benutzer vorgegebenen Zielvektor hat [85].

Wu et al. verwenden einen Genetischen Algorithmus ([180]), bei dem die Individuen aus den Startpositionen und Längen der beiden Primer bestehen. Für die Fitness werden die Eigenschaften ggf. binarisiert, indem sie auf Verletzung von Beschränkungen getestet werden, anschließend wird eine gewichtete Summe der Eigenschaftsbits gebildet. Ebenfalls eine gewichtete Summe, allerdings für nichtbinäre Eigenschaftsbewertungen und in einem Simulated Annealing-Algorithmus, wurde von Hoover und Lubkowski gewählt [76].

Tobler et al. untersuchten die Eignung von Methoden des Maschinellen Lernens zur Vorauswahl von Sondenkandidaten [164]. Die Aufgabe bestand darin, für einen 24-mer aus einem Gen vorherzusagen, ob dessen Signalintensität im Drittel der stärksten Intensitäten über alle möglichen Sonden dieses Gens liegt. Naïve Bayes und Künstliche Neuronale Netze erwiesen tatsächlich als nützlich, Decision Tree Induction und einfaches Ranking nach der Schmelztemperatur dagegen nicht.

Li und Stormo setzen zusätzlich zu oben genannten Eigenschaften und Methoden *suffix arrays* und *sequence landscapes*, zwei Datenstrukturen zur Erfassung des Vorkommens von Subsequenzen, ein, um Sonden zu finden, deren Subsequenzen möglichst selten im Genom außerhalb des zu untersuchenden Gens vorkommen [98]. Einen ähnlichen Ansatz mit *suffix trees* verfolgen Emrich et al. [51].

Anstatt ein Fenster vorgegebener Länge über die Zielsequenz zu schieben, kann man auch für jede Startposition die Primer- bzw. Sonden-Länge ermitteln, für die die Schmelztemperatur einer vorgegebenen am nächsten ist [78, 73].

Ansätze zur Erzeugung von universellen Microarrays, bei denen die eigentlichen Sonden an sogenannte Tags oder Zip-Codes angehängt und deren Komplementärsequenzen (Anti-Tags) auf dem Chip fixiert sind, fallen unter das Word-Design-Problem und werden dort behandelt (s. o.).

Kapitel 6

Ein graphbasierter Sequenz-Design-Algorithmus

6.1 Der Basisstrang-Graph

Der in [53] erstmals erstellte und im Rahmen dieser Arbeit weiterentwickelte und untersuchte Designalgorithmus erzwingt bereits bei der Konstruktion der Sequenzen die Einhaltung der n_b-Uniqueness. Dementsprechend werden DNA-Sequenzen nicht als aus einzelnen Basen, sondern aus sich überlappenden Subsequenzen einer fixen Länge n_b (den sog. *Basissträngen*) bestehend betrachtet (Abb. 6.1).

Zur Sequenzgenerierung werden die Basisstränge in einem Graph G_{bs} angeordnet, der dem Graph zur Erzeugung von DeBruijn-Sequenzen ähnelt (s. Abschnitt 5.2). Sei $S_{bs} = \{\texttt{A,C,G,T}\}^{n_b}$ die Menge aller möglichen Basisstränge der Länge n_b. Dann gibt es für jeden Basisstrang $x \in S_{bs}$ einen mit x markierten Knoten in G_{bs}. Zwischen zwei Knoten mit den Markierungen $x = x_1x_2 \ldots x_{n_b}$ und $y = y_1y_2 \ldots y_{n_b}$ existiert eine gerichtete, mit der Base $b \in \{\texttt{A,C,G,T}\}$ markierte Kante genau dann, wenn $x_2 \ldots x_{n_b} = y_1 \ldots y_{n_b-1}$ und $y_{n_b} = b$ gilt, wenn also x und y zwei aufeinanderfolgende, überlappende Basisstränge in einer längeren Sequenz sein können und y mit b endet (Abb. 6.2).

Damit läßt sich eine Sequenz der Länge n_s auf einen Pfad aus $n_s - n_b + 1$ Knoten durch G_{bs} abbilden, wobei die Folge der Knotenmarkierungen entlang des Pfads der Folge der überlappenden Subsequenzen in der Sequenz entsprechen. Durch diese Abbildung wird das DWD zur Suche nach einer Menge von Pfaden durch G_{bs}. Die n_b-Uniqueness erfordert, daß keine Subsequenz der Länge n_b mehr als einmal in der Sequenzmenge vorkommt, also beschränkt sich die Suche auf Pfade, die keine gemeinsamen Knoten haben. Desweiteren folgt aus der n_b-Uniqueness auch, daß für jede in der Menge vorhandene Subsequenz der Länge n_b deren

```
acgccctca
acgccc
 cgccct
  gccctc
   ccctca
```

Abbildung 6.1: Aufbau einer Sequenz aus Basissträngen. Vier Basisstränge der Länge $n_b = 6$ bilden überlappend eine Gesamtsequenz der Länge $n_s = 9$.

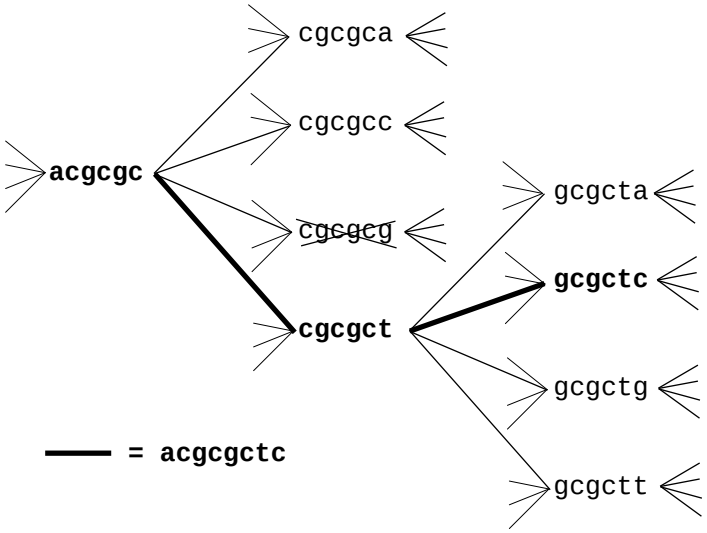


Abbildung 6.2: Ausschnitt aus dem Basisstrang-Graph G_{bs} für $n_b = 6$. Jeder Knoten hat genau vier Vorgänger und vier Nachfolger. Die Nachfolger-Beziehung zwischen zwei Knoten entspricht dem überlappenden Aufeinanderfolgen der entsprechenden Basisstränge in einer längeren Sequenz. Ein Pfad durch den Graph definiert somit eine DNA-Sequenz. Als Beispiel ist hier der Pfad für die Sequenz `acgcgctc` hervorgehoben. Der Basisstrang `cgcgcg` ist selbstkomplementär und wird für die Sequenzsuche als verboten markiert. Der Übersicht halber wurden die Pfeilspitzen weggelassen, die Kanten sind von links nach rechts gerichtet.

Komplement nicht auch in der Sequenzmenge vorkommen darf, daher dürfen auch die mit diesen Komplementen markierten Knoten nicht in den Pfaden vorkommen. Konsequenterweise scheiden Knoten, die mit selbstkomplementären Basissträngen markiert sind, von vornherein aus.

6.2 Ein greedy Algorithmus

Zur Suche nach einer Menge von knotendisjunkten Pfaden durch den Graph G_{bs} läßt sich ein einfacher greedy Algorithmus mit Backtracking verwenden [53]. Er wählt zufällige Startknoten und von dort ausgehend iterierend zufällig einen Nachfolgeknoten, der nicht die n_b-Uniqueness verletzt, bis entweder ein Pfad gewünschter Länge vollständig ist, oder es keinen geeigneten Nachfolgeknoten mehr gibt. In letzterem Fall wird Backtracking angewandt, um einen anderen Pfad zu finden. Führt das Backtracking auf den Startknoten zurück, ohne daß dieser noch geeignete, noch nicht probierte Nachfolgeknoten hat, wird ein neuer Startknoten gezogen. Nach Vollendung eines Pfades wird die zu ihm korrespondierende DNA-Sequenz zur Ausgabemenge hinzugefügt und ebenfalls ein neuer Startknoten gezogen.

Der Einfachheit halber werden im Folgenden Knoten und die Basisstränge, mit denen sie markiert sind, sprachlich nicht unterschieden.

Algorithmus 1

Eingabe: Sequenzlänge n_s, Basisstranglänge n_b
Ausgabe: Menge von DNA-Sequenzen der Länge n_s, die n_b-unique ist.

1. Erzeuge den Graph G_{bs} aller Basisstränge $\in S_{bs}$ der Länge n_b.
2. Setze die Menge der potentiellen Startknoten $V_{start} = S_{bs}$.
3. Markiere alle Knoten, die mit selbstkomplementären Basissträngen markiert sind, als verboten. Entferne diese Knoten aus V_{start}.
4. Markiere alle Knoten als unbenutzt.
5. Solange $V_{start} \neq \emptyset$:
   (a) Ziehe zufällig einen Knoten $x \in V_{start}$ und entferne x aus V_{start}.
   (b) Setze den aktuellen Pfad gleich x.
   (c) Markiere sowohl x als auch $\overline{x}$ als benutzt.
   (d) Setze $i = 1$.
   (e) Solange $i < n_s - n_b + 1$:
   - Wähle zufällig einen Nachfolgeknoten y von x, der weder benutzt noch verboten ist, und für den die Kante (x, y) nicht als probiert markiert ist.
   - Wenn einen solchen Knoten gibt:
     – Markiere y und $\overline{y}$ als benutzt.
     – Verlängere den aktuellen Pfad um y.
     – Setze $i = i + 1, x = y$.
   - Wenn es keinen solchen Knoten mehr gibt:
     – Markiere x und $\overline{x}$ als unbenutzt.
     – Falls $i = 1 \Rightarrow$ breche innere Schleife ab.
     – Sonst sei u der direkte Vorgänger von x im aktuellen Pfad.
       * Markiere die Kante (u, x) als probiert.
       * Lösche alle „probiert“-Markierungen auf von x ausgehenden Kanten.
       * Entferne x aus dem aktuellen Pfad.
       * Setze $i = i - 1$ und $x = u$.
   (f) Falls $i = n_s - n_b + 1$:
   - Entferne alle Knoten des aktuellen Pfades und deren Komplemente aus V_{start}.
   - Konstruiere die dem aktuellen Pfad entsprechende DNA-Sequenz und füge sie zur Ausgabemenge hinzu.

In jeder Iteration der äußeren Schleife wird ein Pfad gesucht. Die innere Schleife implementiert die eigentliche Pfadsuche inkl. Backtracking. Die „probiert“-Markierungen stellen sicher, daß beim Backtracking ein bereits untersuchter Nachfolger nicht erneut gewählt wird. In der angegeben Form erzeugt der Algorithmus soviele Sequenzen, wie er finden kann. Natürlich läßt er sich leicht mit einem Zähler und einer geeigneten Schleifenbedingung so ändern, daß er nur eine maximale Anzahl von Sequenzen sucht.

6.3 Erweiterungen

Neben der n_b-Uniqueness gibt es eine Reihe weiterer Anforderungen an die Sequenzen, die ebenfalls implementiert wurden.

kompatible Sequenzen Eine bereits vorhandene Sequenzmenge, die die n_b-Uniqueness erfüllt, kann unter Beibehaltung dieser Eigenschaft erweitert werden. Hierzu werden die vorgegebenen Sequenzen in ihre Basisstränge zerlegt und diese im Graph als verboten markiert. Sind vorgegebene Sequenzen kürzer als n_b, so müssen sie für die Erhaltung der n_b-Uniqueness nicht weiter beachtet werden [53].

verbotene Subsequenzen Für Sequenzen, die nicht als echte Subsequenz vorkommen dürfen, würde es genügen, zu überprüfen, ob so eine Sequenz je komplett vorkommt. Um auch eine Ähnlichkeit zu ihnen zu vermeiden, werden sie aber ähnlich behandelt wie eine zu erweiternde Sequenzmenge. Zusätzlich werden für verbotene Sequenzen, die kürzer als n_b sind, alle Basisstränge, die solche Sequenzen als Subsequenzen enthalten, ebenfalls als verboten markiert.

Basenwiederholungen Ein Sonderfall von verbotenen Subsequenzen sind kontinuierliche Wiederholungen ein und derselben Base. Hier wird die aufeinanderfolgende Verwendung solcher Basen während der Pfadsuche mitgezählt, bei Erreichen eines vom Benutzer vorgegebenen Schwellenwertes wird die bisher wiederholte Base von der Auswahl der Kante zum Nachfolgeknoten ausgeschlossen. Auf diese Weise lassen sich nicht nur Wiederholungen einzelner Basen, sondern auch Abfolgen von Basen einer Teilmenge von {`A,C,G,T`} verbieten. Für diese Teilmengen verwendet man die IUPAC-Notation [39] (s. Tabelle 6.1). Ein Verbot von `SSS` schließt also die Subsequenzen `CCC`, `CCG`, `CGC`, `CGG`, `GCC`, `GCG`, `GGC` und `GGG` aus.

fixe Subsequenzen Einzelne Basen oder ganze Subsequenzen der zu erzeugenden Sequenzen können vorgegeben werden. Für diese Basen werden im Suchalgorithmus die entsprechenden Nachfolger nicht zufällig gewählt, sondern es wird jeweils die der vorgegebenen Base entsprechenden Kante verfolgt. Auch die fixen Subsequenzen lassen sich in der IUPAC-Notation (Tab. 6.1) angeben, so daß die Nachfolgerauswahl auf mehr als einen Knoten eingeschränkt werden kann. Haben fixierte Subsequenzen ein Länge von n_b oder länger, werden ihre Basisstränge extrahiert und sowohl sie als auch ihre Komplemente als benutzt markiert. Durchläuft die Pfadsuche diese Subsequenzen für die vorgegebene Sequenz und Position, wird die „benutzt“-Markierung dieser Basisstränge ignoriert, sie wird nur berücksichtigt, wenn die Basisstränge an anderer Stelle in der Pfadsuche zur Auswahl stünden.

GC-Gehalt, Schmelztemperatur, freie Enthalpie Eigenschaften, die komplette Sequenzen betreffen, wie GC-Gehalt, Schmelztemperatur [53] und freie Enthalpie, lassen sich nicht im Verlauf der Sequenzkonstruktion berücksichtigen, sondern erst nach Vervollständigung der Sequenz prüfen. Verletzt die neuentstandene Sequenz dabei vom Benutzer vorgegeben Einschränkungen, so wird ebenfalls Backtracking eingeleitet und eine neue Sequenz gesucht. Um nicht nur z. B. einen GC-Gehalt von 50 % zu erhalten, sondern auch eine gleichmäßige Verteilung der `A-T`- und `G-C`-Basenpaare über die Sequenz, kann auch der GC-Gehalt der Basisstränge eingeschränkt werden, so daß z. B. jeder Basisstrang selbst einen GC-Gehalt von 50 % hat. Basisstränge, die gegen eine solche Einschränkung verstoßen, werden als verboten markiert.

Symbol	Basen	Erklärung
A	A	**A**denin
C	C	**C**ytosin
G	G	**G**uanin
T	T	**T**hymin
R	G oder A	Pu**r**ine
Y	T oder C	P**y**rimidine
M	A oder C	A**m**ino
K	G oder T	**K**eto
S	G oder C	**s**trong (starke Bindung, 3 H-Brücken)
W	A oder T	**w**eak (schwache Bindung, 2 H-Brücken)
H	A oder C oder T	nicht G (H folgt im Alphabet auf G)
B	C oder G oder T	nicht A
V	A oder C oder G	nicht T (bei RNA: nicht U)
D	A oder G oder T	nicht C
N	A oder C oder G oder T	a**n**y base

Tabelle 6.1: Nomenklatur für nicht vollständig spezifizierte Basen nach dem Vorschlag der IUPAC-IUB. Angegeben sind jeweils das in der Sequenz zu verwendende Symbol, die Basen, für die dieses Symbol stehen kann, sowie eine kurze Begründung für die Wahl des jeweiligen Symbols [39].

Homologie und andere Maße Da es in ungünstigen Fällen trotz n_b-Uniqueness zu hoher Sequenzähnlichkeit kommen kann (s. Abschnitt 4.3.1), hat der Benutzer die Möglichkeit, diese Ähnlichkeit bzgl. eines Distanzmaßes zu beschränken. Implementiert wurde dies für die Homologie (s. Abschnitt 4.1), die Erweiterung auf weitere Distanzmaße läßt sich sehr einfach realisieren. Nach Vervollständigung einer Sequenz wird deren Distanz zu allen anderen bereits gefundenen Sequenzen berechnet. Verletzt diese die vorgegebene Einschränkung, wird wiederum Backtracking ausgelöst [53].

Konkatenationen Werden die DNA-Sequenzen im Anwendungsprotokoll aneinandergehängt, so ist es natürlich wünschenswert, daß auch die resultierenden, längeren Sequenzen noch die n_b-Uniqueness erfüllen. Leider läßt sich aber aus der n_b-Uniqueness der Einzelsequenzen nicht auf die der zusammengesetzten schließen. Bei Konkatenation zweier Sequenzen „entstehen“ $n_b - 1$ neue Subsequenzen der Länge n_b, die zu berücksichtigen sind (Abb. 6.3) [53].

Die Abbildung der Sequenzkonstruktion auf die Pfadsuche bietet hierfür eine recht einfache Lösung. Wenn der Benutzer angibt, daß zwei Sequenzen aneinandergehängt werden sollen, so wird zunächst die erste Sequenz durch normale Pfadsuche erzeugt, anschließend wird dieser Pfad um n_s Knoten verlängert und somit die zweite Sequenz erzeugt, wobei die ersten $n_b - 1$ Knoten dieser Verlängerung den Basissträngen des Übergangsbereichs entsprechen.

I. a. ist es möglich, daß eine solche Erzeugungsreihenfolge der Sequenzen nicht offensichtlich ist. Sollen z. B. zwei Sequenzen X und Y in der Anwendung zu XYX konkateniert werden, so würde die oben beschriebene Verlängerung von X zu Y zwar den XY-Übergang berücksichtigen, nicht aber den YX-Übergang. Analog problematisch wäre die Verlängerung von Y zu X. Um sämtliche Übergangsbereiche berücksichtigen zu können, muß zunächst der Benutzer oder eine anwendungsspezifische, dem Suchalgorithmus übergeordnete Routine eine sinn-

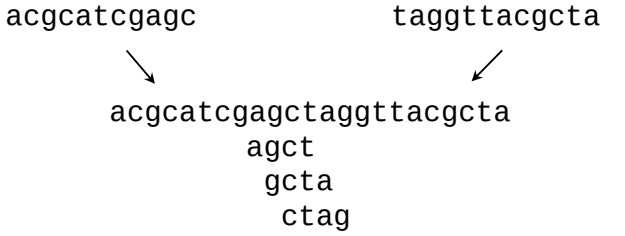


Abbildung 6.3: Konkatenation zweier Sequenzen. Für $n_b = 4$ entsteht bei der Konkatenation zweier Sequenzen ein Übergangsbereich, der aus 3 Basissträngen besteht. Hier verletzt die Konkatenation die n_b-Uniqueness, da `agct` selbstkomplementär ist und `gcta` bereits als letzter Basisstrang der rechten Sequenz vorkommt.

volle Generierungsreihenfolge auswählen. Dann werden zunächst die in dieser Reihenfolge zu verlängernden Sequenzen erzeugt. Anschließend wird für die Verlängerungen ein „Rahmen“ aus Basissträngen vorgegeben (Abb. 6.4). Im XYX-Beispiel besteht dieser Rahmen aus dem letzten bzw. ersten Basisstrang von X. Für die Generierung von Y dient anstatt eines zufällig gewählten Knotens der letzte Basisstrang von X als Startknoten. Wurden XY-Übergang und Y selbst durch Pfadsuche konstruiert, so läuft die Pfadsuche für den YX-Übergang weiter, wobei die Nachfolgersuche hierbei wie beim Einbau fixer Subsequenzen durch die Basen des ersten Basisstranges von X determiniert ist. Wird dabei ein Basisstrang erreicht, der als benutzt oder verboten markiert ist, wird Backtracking ausgelöst. Anderenfalls endet das Pfadwachstum $n_b - 1$ Knoten nach Vollendung von Y, was der Länge des Übergangsbereichs entspricht.

Weiterhin muß berücksichtigt werden, daß es für eine Sequenz ggf. mehrere Konkatenationspartner geben kann. Dies gilt sowohl für Verlängerungen als auch für zu verlängernde Sequenzen, und für $3'$- wie $5'$-Ende. Z. B. sollen die Sequenzen X, Y, A, B, C, D und E generiert werden, und zwar zuerst X und Y, dann A, B, C, D und E. Dann können die Konkatenationen XA, XB, XC vorgesehen sein, ebenso AX, BX, CX, oder AX, AY, oder XA, YA. Daher müssen sowohl am Anfang als auch am Ende der Verlängerungen verzweigende bzw. zusammenlaufende Pfade gesucht werden können.

Für die Konkatenationen XA, XB, XC, XD und XE z. B. wäre es naheliegend, einfach fünf Verlängerungen von X zu suchen, ausgehend vom letzten Basisstrang von X. Allerdings hat der entsprechende Knoten in G_{bs} nur vier direkte Nachfolger, spätestens nach der Erzeugung der vierten Verlängerung sind diese alle als benutzt markiert, die Suche nach einer fünften Verlängerung muß also scheitern.

Will man die Konkatenationsmöglichkeiten nicht auf eine maximal vierfache Verzweigung beschränken, was viele Anwendungen unmöglich machen würde, so muß man an dieser Stelle also die mehrfache Verwendung von Basissträngen zulassen. Allerdings muß, um das Ausmaß der Verletzung der n_b-Uniqueness zu minimieren, sichergestellt werden, daß das mehrfache Vorkommen der betroffenen Basisstränge wirklich nur auf die jeweilige Verzweigung beschränkt ist.

Erreicht wird dies durch eine parallele Pfadsuche für alle Verlängerungen. Dazu werden sämtliche Rahmensequenzen (also Konkatenationsnachbarn in $5'$- und in $3'$-Richtung) aller Verlängerungen gesammelt. Für jedes Paar aus zu generierender Sequenz und $5'$-Nachbar wird zunächst ein eigener Pfad angelegt, der jeweils mit dem letzten Basisstrang des $5'$-Nachbarn initialisiert wird (Abb. 6.5). Nun werden diese Pfade parallel verlängert, d. h. in jedem Schritt der Pfadsuche werden *alle* Pfade um einen Knoten verlängert. Dabei wird zunächst für jeden

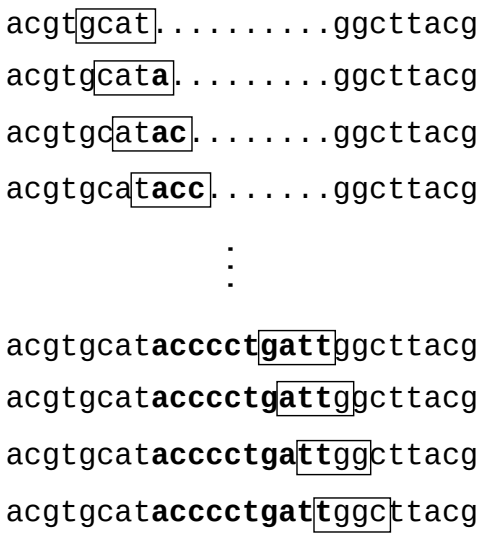


Abbildung 6.4: Behandlung von Konkatenationen. Die 5′- und 3′-Nachbarn der zu suchenden Sequenz bilden einen Rahmen, der durch die Pfadsuche aufgefüllt wird. Die Punkte stehen für die zu findenden Basen der Verlängerung. Als Startknoten wird der letzte Basisstrang des 5′-Nachbars gewählt (hier für $n_b = 4$). Der Pfad wird verlängert, bis die mittlere Sequenz (fett hervorgehoben) vollständig ist, anschließend werden noch die $n_b - 1$ Basisstränge des Übergangs zum 3′-Nachbarn überprüft.

Pfad je ein Nachfolgeknoten gesucht. Pfade, die zu ein und derselben zu generierenden Sequenz zusammenlaufen sollen, folgen dabei Kanten mit derselben Markierung. Ist ein Nachfolgeknoten in einem Pfad als benutzt markiert, so wird überprüft, ob diese Verwendung *in diesem Schritt* bei einer Verzweigung oder einem Zusammenlaufen von Pfaden geschieht, unter denen auch der aktuell betrachtete Pfad ist. Ist dies der Fall (und sind solche Mehrfachverwendungen nach Vorgabe des Benutzers erlaubt), so wird der Nachfolgeknoten trotz der „benutzt"-Markierung zugelassen. Wird ein Nachfolgeknoten in einem Pfad verworfen, der mit anderen zu einer gemeinsamen Verlängerung zusammenläuft, so können auch in diesen anderen Pfaden die Nachfolger mit derselben Kantenmarkierung nicht gewählt werden. Wird Backtracking für einen Pfad ausgelöst, so wird es parallel für alle Pfade, die zur selben Verlängerung gehören, durchgeführt, alle anderen Pfade werden solange „eingefroren", also nicht weiter verlängert oder verkürzt, bis die im Backtracking neu gesuchten Pfade wieder die gleiche Länge erreicht haben wie die eingefrorenen.

Seien z. B. die Sequenzen $X_i, i = 1, \ldots, 5$ bereits generiert und enden für $n_b = 4$ mit den Basissträngen $x_{i1}x_{i2}x_{i3}x_{i4}$, die Verlängerungen $V_j, j = 1, \ldots, 5$ werden noch gesucht, wobei die Konkatenationen $X_1V_1, X_2V_1, X_3V_1, X_4V_1, X_5V_1, X_1V_2, X_1V_3, X_1V_4, X_1V_5$ vorkommen sollen. Dann gibt es neun Pfade, fünf davon werden mit dem Basisstrang $x_{11}x_{12}x_{13}x_{14}$ initialisiert. Wird im ersten Schritt sowohl für V_1 als auch für V_5 als erste Base $b \in \{\texttt{A,C,G,T}\}$ gewählt, so erreichen beide Pfade den Knoten $x_{12}x_{13}x_{14}b$. Da diese Mehrfachverwendung in einer Verzweigung auftritt, kann (und sollte) sie toleriert werden. Werden in den ersten drei Schritten für V_1 die Basen b_1, b_2, b_3 gewählt und sei z. B. $x_{24} = x_{44}$, so würde im dritten Schritt für die Pfade X_2V_1 und X_4V_1 derselbe Basisstrang $x_{24}b_1b_2b_3$ gewählt. Da diese Pfade zu derselben Verlängerung V_1 zusammenlaufen, ist auch das tolerierbar. Wäre eine solche Mehrfachverwendung nicht erlaubt (was sinnvoll ist, wenn weniger als fünf Pfade zusammenlaufen), so würde

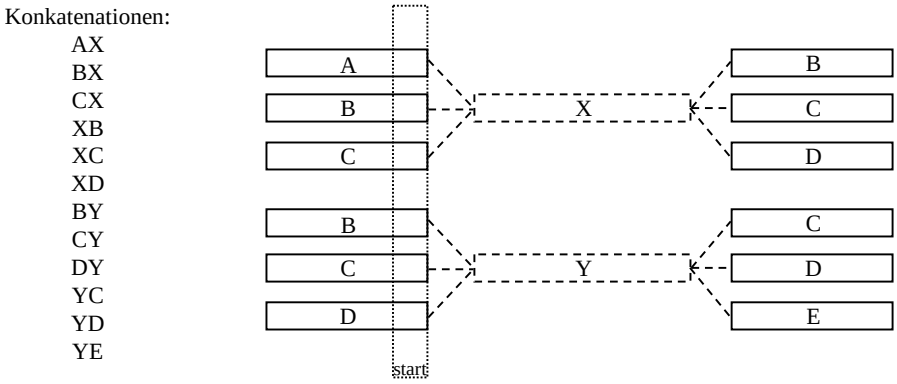


Abbildung 6.5: Parallele Pfadverlängerung. Die Sequenzen A bis E sind gegeben, X und Y noch zu erzeugen. Für jede zu suchende Verlängerung gibt es soviele Pfade wie Konkatenationsnachbarn auf einer Seite. Alle Pfade werden mit dem letzten Basisstrang des jeweiligen $5'$-Nachbarn als Startknoten initialisiert. Das mehrfache Auftauchen der Sequenzen B und C als $5'$-Nachbarn bedeutet, daß sich die zugehörigen Pfade verzweigen, ggf. muß hier also die Mehrfachverwendung von Basissträngen erlaubt werden.

paralleles Backtracking für die Pfade $X_iV_1, i = 1, \ldots, 5$, die ja in jedem Schritt gleichen Kantenmarkierungen folgen müssen, ausgelöst, während die Pfade $X_1V_j, j = 2, \ldots, 5$ eingefroren werden, bis die erstgenannten Pfade wieder bis zum dritten Schritt verlängert sind. Für ein konkretes Beispiel siehe Abbildung 6.6.

Sind die Übergangsbereiche für alle Konkatenationen erzeugt, werden die Verlängerungen weiterhin parallel generiert. Bei den Übergängen zu den $3'$-Nachbarn der Verlängerungen werden analoge Überprüfungen vorgenommen. Sollen die Verlängerungen verschieden lang sein, so werden sie bis zum jeweils letzten Basisstrang, der noch komplett in der Verlängerung liegt, erzeugt und dann ggf. eingefroren, bis auch die längeren Sequenzen komplett sind. Erst dann werden die $3'$-Übergänge wieder für alle Pfade gesucht.

Strukturen Der Algorithmus läßt sich nicht nur für die Suche nach DNA-Wörtern verwenden, sondern auch für das Struktur-Design-Problem. Der Benutzer oder eine übergeordnete Programm-Routine zerlegt dazu die gewünschte Struktur in lineare Abschnitte, die dann als Wörter generiert werden können. Abschnitte, die in der Zielstruktur benachbart sind, werden wie oben beschrieben als Konkatenationen behandelt. Eine solche Erweiterung das Algorithmus für das Struktur-Design ist der *DNA-Sequence-Compiler* (s. Abschnitt 6.5.3).

Beim durch diese Software unterstützen Entwurf von Junctions ist eine besondere Form der Nachbarschaft zu beachten. Betrachtet man ein 3-armiges Molekül, das in einzelne Sequenzen aufgeteilt ist wie in Abbildung 6.7 dargestellt, erkennt man, daß die Armsequenzen in der Mitte der Junction benachbart sind. Allerdings trifft hier jeweils das $5'$-Ende von Arm_i auf das $3'$-Ende von $\overline{Arm_{i-1}}$, also das Komplement des $5'$-Endes von Arm_{i-1}. Daher wird bei der Wahl der Startknoten für die Armsequenzen zusätzlich überprüft, ob die Konkatenation jeweils aus Startknoten eines Arms und Komplement des Startknotens eines benachbarten Arms die n_b-Uniqueness verletzt.

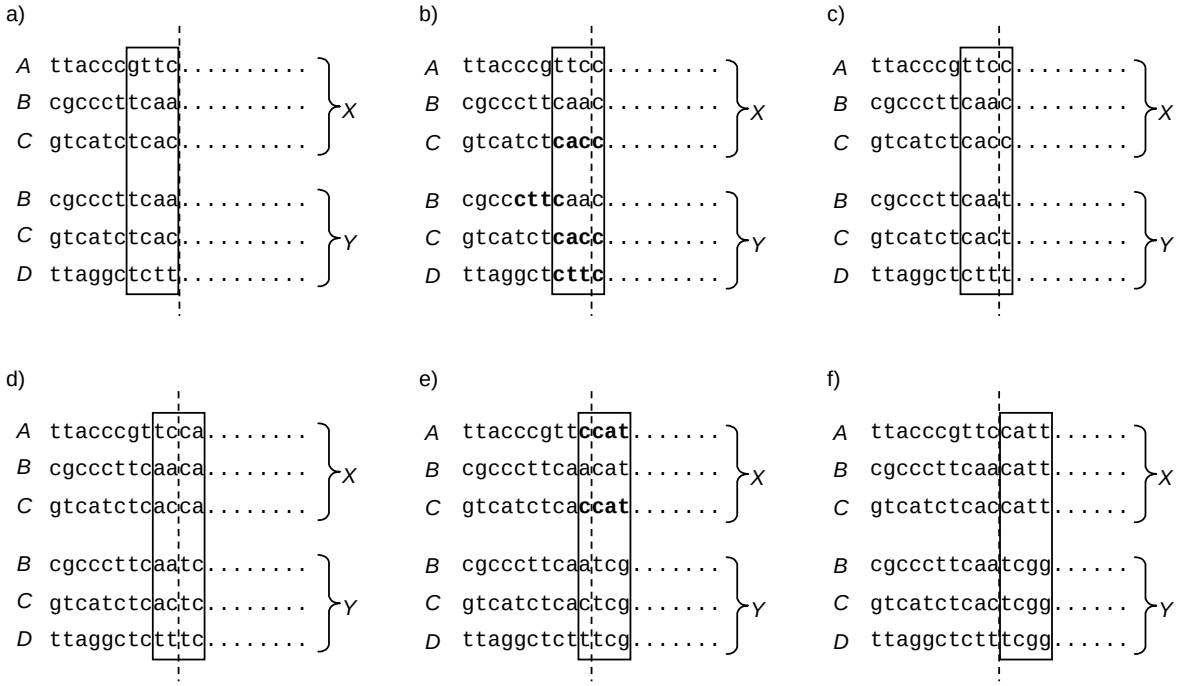


Abbildung 6.6: Erzeugung des Übergangs von den $5'$-Nachbarn zu den Verlängerungen für $n_b = 4$. a) Startsituation wie in Abbildung 6.5. Die Punkte markieren noch zu suchende Basen, die vertikale gestrichelte Linie die Schnittstelle zwischen $5'$-Nachbar und Verlängerung, der Rahmen die aktuell betrachteten Basisstränge. b) 1. Schritt: Würden sowohl X als auch Y mit `c` anfangen, so wäre beim CY-Übergang der Basisstrang `cacc` bereits benutzt. Der Algorithmus würde feststellen, daß die andere Verwendung dieses Basisstrangs im gleichen Schritt im CX-Übergang liegt, also in der von C ausgehenden Verzweigung. Diese Verletzung der n_b-Uniqueness könnte also toleriert werden. (Gleiches gilt für den Basisstrang `caac` in BX und BY.) Allerdings ist auch `cttc` bereits markiert, und zwar schon als Basisstrang in B. Dieses mehrfache Vorkommen kann also nicht toleriert werden, für Y wird ein anderer Nachfolger gesucht. c) Alternativer 1. Schritt: X und Y fangen mit verschiedenen Basen an, alle Basisstränge werden hier zum ersten Mal verwendet. d) 2. Schritt: Auch hier ist alles in Ordnung. e) 3. Schritt: Im CX-Übergang ist der Basisstrang `ccat` bereits als benutzt markiert. Da die zweite Verwendung im AX-Übergang liegt, also im Zusammenlaufen der Pfade zu X, und im selben Schritt erfolgt ist, kann dies erlaubt werden. f) 4. Schritt: Die Übergänge sind komplett, ab jetzt liegen die Basisstränge komplett in den Verlängerungen. Diese können also weiter generiert werden, ohne auf Verzweigungen achten zu müssen. Ein analoges Vorgehen wie hier beschrieben findet dann wieder bei den Übergängen zu den $3'$-Nachbarn statt.

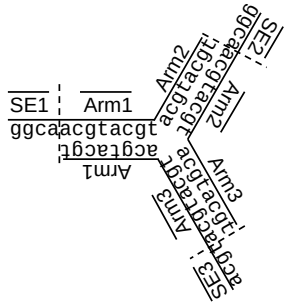


Abbildung 6.7: 3-armige Junction mit Sequenzidentifikatoren. Jeder Arm ist in eine doppelsträngige Hauptsequenz und ein einzelsträngiges sticky End aufgeteilt. Die Arme stoßen am Kreuzungspunkt mit ihren 5′-Enden aneinander.

6.4 Diskussion

Bei Betrachtung des Algorithmus ergeben sich die folgenden wichtigen Aspekte, sowohl positive als auch negative.

Erzwungene Unähnlichkeit Der Algorithmus erzwingt eine wichtige Eigenschaft zur Unähnlichkeit der Sequenzen, die n_b-Uniqueness, bei der Konstruktion der Sequenzen, anstatt z. B. Zufallssequenzen zu filtern, bis eine solche Eigenschaft erfüllt ist.

Leichte Erweiterung auf neue Anforderungen Er ist leicht auch für weitere Anforderungen an die Sequenzen erweiterbar. Diese können entweder durch Verbieten von Basissträngen implementiert werden (verbotene Sequenzen, GC-Gehalt der Basisstränge), oder durch Prüfen und ggf. Auslösen von Backtracking nach Vollendung einer Sequenz (Schmelztemperatur, GC-Gehalt, Homologie). Letzteres entspricht zwar wiederum dem o. g. Filtern von Sequenzen, allerdings haben diese durch die n_b-Uniqueness bereits eine gewisse Grundqualität.

Eingeschränktes Verwerfen von Sequenzen Der hier gewählte Weg hat einen Vorteil gegenüber den (modifizierten) DeBruijn-Sequenzen. Verletzt eine Sequenz z. B. die Einschränkung der Schmelztemperatur, so wird diese nicht komplett verworfen, sondern zunächst repariert. Auch wenn das Backtracking doch zu einem Verwerfen der kompletten Sequenz (also dem Verwerfen des Startknotens) geführt hat, wird nur diese einzelne Sequenz verworfen, nicht die ganze Menge. Bei den DeBruijn-Sequenzen bzw. den Algorithmen zur Erzeugung dieser ist es nicht offensichtlich, ob ein solches Reparieren von Teilen der DeBruijn-Sequenzen möglich ist. I. a. muß bei den Algorithmen, die nicht mit Backtracking arbeiten, die ganze DeBruijn-Sequenz verworfen werden, auch wenn Teilsequenzen sich bereits als geeignete DNA-Wörter herausgestellt haben.

Automatische Suche Die Sequenzgenerierung läuft automatisch ab, es ist keine Interaktion während des Suchvorgangs nötig wie bei SEQUIN, das ebenfalls die n_b-Uniqueness von Sequenzen implementiert ([147], s. o.).

Erweiterung auf Strukturdesign Die n_b-Uniqueness läßt sich leicht auf die Konkatenation von Sequenzen erweitern. Dadurch läßt sich der Algorithmus auch für das Design von

Strukturen verwenden.

Sequenzausbeute Der greedy Algorithmus erzielt eine Ausbeute an Sequenzen, die nahe an das theoretische Maximum unter Bewahrung der n_b-Uniqueness herankommt. Für eine nähere Betrachtung siehe Abschnitt 7.1.

Hohe Abstraktion Die n_b-Uniqueness abstrahiert die Neigung zur (Kreuz-)Hybridisierung auf das Vorkommen von gleichen bzw. komplementären Subsequenzen fixer Länge. Dies ist eine starke Abstraktion, da tatsächlich unterschiedliche Subsequenzen gleicher Länge eine unterschiedliche Neigung zur Hybridisierung haben. Allerdings gibt es zahlreiche Hinweise darauf, daß die n_b-Uniqueness ein sinnvolles Instrument zur Vermeidung von Fehlhybridisierungen ist. Dirks et al. haben in [47] die n_b-Uniqueness (im Artikel unter dem Namen Sequence-Symmetry-Minimization) mit anderen Kriterien (Minimierung der freie Enthalpie der Zielstruktur und Zielstruktur ist die Struktur mit der minimalen freien Enthalpie) für das Design einer einzelsträngigen Sekundärstruktur verglichen und festgestellt, daß die n_b-Uniqueness und das minimale-freie-Enthalpie-Kriterium geeigneter sind als die Minimierung der freien Enthalpie und auch deutlich besser als die zufällige Sequenzsuche. Die Gruppe von Nadrian Seeman entwirft und baut seit Jahren erfolgreich komplexe DNA-Strukturen unter Beachtung der n_b-Uniqueness, und erzielt damit auch *in vitro* hervorragende Ergebnisse. Nicht zuletzt zeigt ein im Rahmen dieser Arbeit durchgeführtes Experiment, daß die Einschränkung der Basisstranglänge n_b zu einer Vergrößerung der Energielücke und damit zu einer Verbesserung der Spezifität der Hybridisierung führt (s. Abschnitt 7.3). Auch theoretisch ist die Verwendung von Subsequenzen durchaus motiviert, da sie eine Nukleation sowie mehrere Stackings von Basenpaaren darstellen, also den Hybridisierungsprozeß realistisch abbilden, aber gerade für kurze Subsequenzen auch genügend Flexibilität zur Berücksichtigung von Lücken bieten.

Kleiner Hamming Ein wesentlicher Kritikpunkt an der n_b-Uniqueness ist die Möglichkeit, doch sehr ähnliche Sequenzen zu erzeugen. So sind z. B. die Sequenzen
`ACGTGAGTGCA` und
`TGCACGCACGT`
wegen des `A-G`-Mismatches in der Mitte 6-unique, ein Duplex aus den beiden wäre aber nicht viel weniger stabil als ein perfekt komplementärer Duplex. Zum Einen ist aber die Wahrscheinlichkeit, daß zwei Sequenzen, die sich oder ihrem Komplement so ähnlich sind, in einer Menge gemeinsam auftauchen, recht gering. Ist eine Sequenz der Länge n_s gegeben, so ist die Wahrscheinlichkeit, daß eine weitere Sequenz, die mit der ersten die n_b-Uniqueness erfüllt, den Hamming-Abstand 1 von der ersten hat,

$$p(H = 1) = \frac{3}{4^{n_s}}, \tag{6.1}$$

da es insgesamt 4^{n_s} Sequenzen gibt, alle Basen außer der mittleren festgelegt sind, und es für die mittlere Base drei Möglichkeiten gibt, ein Mismatch zu erzeugen. Verallgemeinert man diese Argumentation auf einen Hamming-Abstand von k, so ergibt sich die Wahrscheinlichkeit

$$p(H = k) = \frac{3^k}{4^{n_s}}. \tag{6.2}$$

Für einige beispielhafte Belegungen von n_b, n_s und k sind diese Wahrscheinlichkeiten in Tabelle 6.2 aufgeführt und zeigen, daß sie recht gering sind.

n_b	4	5	6	7	8	9	10
n_s	7	9	11	13	15	17	19
$p(H=1)$	$1.83 \cdot 10^{-4}$	$1.14 \cdot 10^{-5}$	$7.15 \cdot 10^{-7}$	$4.47 \cdot 10^{-8}$	$2.79 \cdot 10^{-9}$	$1.75 \cdot 10^{-10}$	$1.09 \cdot 10^{-11}$
$p(H=2)$	$5.49 \cdot 10^{-4}$	$3.43 \cdot 10^{-5}$	$2.15 \cdot 10^{-6}$	$1.34 \cdot 10^{-7}$	$8.38 \cdot 10^{-9}$	$5.24 \cdot 10^{-10}$	$3.27 \cdot 10^{-11}$
$p(H=3)$	$1.65 \cdot 10^{-3}$	$1.03 \cdot 10^{-4}$	$6.44 \cdot 10^{-6}$	$4.02 \cdot 10^{-7}$	$2.51 \cdot 10^{-8}$	$1.57 \cdot 10^{-9}$	$9.82 \cdot 10^{-11}$

Tabelle 6.2: Wahrscheinlichkeiten, daß zwei Sequenzen der Länge $n_s = 2n_b - 1$ den Hamming-Abstand $H = 1, 2, 3$, haben.

Zum Anderen gilt diese Betrachtung für $n_s = 2n_b - 1$. Es ist jedoch klar, daß n_b so klein wie möglich gewählt werden sollte, insbesondere deutlich kleiner als n_s, um eine ausreichende Unähnlichkeit der Sequenzen zu gewährleisten. Ein größerer Unterschied zwischen n_b und n_s verringert aber auch die Wahrscheinlichkeit für das Auftreten der hohen Ähnlichkeit zusätzlich. Dieses Problem stellt sich also im Wesentlichen nur bei ungeschickter Wahl der Parameter, darf aber aufgrund der positiven Wahrscheinlichkeit nicht völlig ignoriert werden.

Sekundärstrukturen Ein weiterer häufig geäußerter Kritikpunkt an der n_b-Uniqueness betrifft die Vermeidung einzelsträngiger Sekundärstrukturen, die angeblich nicht gegeben sei. Dies ist bereits theoretisch nicht korrekt, da die n_b-Uniqueness nicht nur inter- sondern auch intramolekular gilt. Ist ein Basisstrang also Subsequenz eines DNA-Strangs, so darf natürlich weder er noch sein Komplement noch einmal in derselben Sequenz auftauchen. Also sollten auch verschiedene Bereiche ein und derselben Sequenz untereinander genügend nicht-komplementär sein, um die Bildung von einzelsträngigen Sekundärstrukturen unwahrscheinlich zu machen.

Untersuchungen mit *in silico* durchgeführten Experimenten, bei denen Sekundärstrukturen generierter Sequenzen vorhergesagt wurden, zeigen, daß die gefundenen Strukturen im Mittel nur relativ geringe Stabilität aufweisen (s. Abschnitt 7.2). Allerdings zeigen die Ergebnisse auch, daß die durchschnittliche Stabilität der Sekundärstrukturen nur für sehr kleine n_b geringer ist als die von für Zufallssequenzen vorhergesagten Strukturen.

Laufzeit Die größte Schwäche des greedy Algorithmus ist seine Laufzeit. Zwar werden im best case für die Erzeugung einer Sequenz der Länge n_s mit Basisstranglänge n_b nur $n_s - n_b + 1$ Schritte benötigt, da in diesem Fall immer der erste getestete Knoten verfügbar ist. Der worst case sieht aber leider wesentlich schlechter aus. In einem solchen Szenario wird immer erst die vollständige Sequenz betrachtet und dann z. B. wegen Verletzung der Einschränkung der Schmelztemperatur verworfen. Betrachtet man zunächst die innere Schleife, also die Pfadsuche von einem Startknoten aus, so bleiben durch die Festlegung des Startknotens für einen Pfad noch $n_s - n_b$ Knoten zu suchen. Ist diese Suche erfolglos, so wird durch das Backtracking schlimmstenfalls für jeden verfolgten Zweig der vollständige Pfad durchlaufen. Das Backtracking spannt also einen vollständigen Baum[1] der Nachfolgerknoten auf. Da es für jede Verzweigung des Baums vier mögliche Nachfolger gibt, werden maximal $4^{n_s - n_b}$ Knoten durchlaufen, bevor die Pfadsuche von diesem Startknoten aus als gescheitert betrachtet wird. Da es 4^{n_b} mögliche Startknoten in G_{bs} gibt, ergibt sich mit der äußeren Schleife eine worst-case-Laufzeit von $4^{n_b} \cdot 4^{n_s - n_b} = 4^{n_s}$. Dies entspricht der Laufzeit der vollständigen Aufzählung aller Sequenzen der Länge n_s, ist also tatsächlich als äußerst schlecht zu beurteilen.

[1] In diesem gedachten Baum können Knoten des Graphen mehrfach vorkommen, da sie über verschiedene Pfade vom Startknoten aus erreichbar sein können. Der Baum ist also kein Teilgraph von G_{bs}, sondern dient nur der Veranschaulichung des Laufzeitverhaltens.

I. a. erzeugt man nicht nur eine Sequenz, sondern mehrere, und mit jeder fertigen Sequenz sinkt die Wahrscheinlichkeit, noch eine weitere zu finden, die ebenfalls die gestellten Anforderungen erfüllt. Hier kann es also durchaus sinnvoll sein, eine zu lange Suche abzubrechen, die bisher gefundenen Sequenzen zu verwerfen und mit einem neuen Startwert für den Zufallszahlengenerator neu zu beginnen. Dabei hofft man, daß die neuen Instanzen der Sequenzen, die man bereits gefunden hatte, vielleicht eher eine weitere Sequenz zulassen, z. B. durch eine günstigere Zerteilung des Graphen. Damit verzichtet man allerdings auf oben genannten Vorteil gegenüber den DeBruijn-Sequenzen. Erste, sehr subjektive Erfahrungen mit der Generierung von Sequenzen unter verschiedenen Einschränkungen von Eigenschaften deuten an, daß erfolgreiche Suchen meist relativ schnell beendet sind, während lang andauernde Suchen meistens in erfolglosen vollständigen Aufzählungen enden. Es bleibt zu untersuchen, ob dieser Eindruck tatsächlich stimmt. Eine theoretische Analyse ist allerdings aufgrund der vielen verschiedenen Einschränkungen, die gemacht werden können, sehr schwierig. Eine empirische Untersuchung wäre nur für sehr kleine n_b durchführbar, da man mehrfach tatsächlich vollständige Aufzählungen durchführen müßte. Eine Heuristik, die nach einer gewissen Laufzeit bzw. einer bestimmten, von n_b und n_s abhängigen Anzahl von Schritten die Suche abbricht bietet sich aber als Gegenstand weiterer Untersuchung und Entwicklung an.

Eine Verbesserung der Laufzeit durch etablierte effiziente Graph-Algorithmen ist zumindest zur Zeit nicht in Aussicht. Die Suche nach einer möglichst großen Menge knotendisjunkter Pfade, also das *maximum fixed-length disjoint paths problem* ist für interessante Pfadlängen NP-vollständig [58]. Schlimmer noch, für nah verwandte Probleme wurde gezeigt, das es NP-schwer ist, eine optimale Lösung gut zu approximieren [108, 68]. Gute Approximationsalgorithmen gibt es nur für planare Graphen [91], jedoch ist G_{bs} für $n_b \geq 2$ nicht planar. Es würden zudem praktische Fragen offen bleiben. So beinhalten die erwähnten Probleme vorgegebene Start- und Endknoten der Pfade, was hier nur für die Verlängerung existierender Pfade zu Konkatenation von Sequenzen der Fall ist. Für das allgemeinene DWD-Problem bliebe die Frage nach einer sinnvollen Wahl dieser ausgezeichneten Knoten. Ebenfalls unberücksichtigt bleibt bei den Standardproblemen, daß während der Pfadkonstruktion die Knoten, deren Basisstränge komplementär zu den verwendeten sind, ebenfalls als benutzt markiert werden. Diese zusätzliche, dynamische Randbedingung könnte die Güte von Approximationsalgorithmen weiter verschlechtern oder ihre Effizienz beeinträchtigen.

6.5 Software

Die Software-Sammlung CANADA (**C**omputer **A**ided **N**ucleic **A**cid **D**esign p**A**ckage) enthält unter anderem zwei Designprogramme, die auf dem Graph-Algorithmus beruhen: Den DNA-Sequence-Generator (DSG) für das DNA-Word-Design-Problem und den DNA-Sequence-Compiler (DSC) für eine Klasse von Struktur-Design-Problemen. Beide Programme gibt es sowohl mit einer graphischen Benutzeroberfläche als auch Konsolen-basiert. Während bei Ersteren die Eingaben über Dialogfenster gemacht werden, geschieht dies bei Letzteren mit Hilfe der Sequenzbeschreibungssprache DeLaNA.

6.5.1 DeLaNA

Als Format für Ein- und Ausgabedateien der Programme in CANADA wurde die Beschreibungssprache DeLaNA (**De**scription **La**nguage for **N**ucleic **A**cid molecules) entworfen. In dieser werden Sequenzen als Objekte beschrieben, die bestimmte Eigenschaften haben. In Eingabedateien dient die Festlegung dieser Eigenschaften als Beschränkung des Suchraums (Abb.

```
// File:           diss_example.dln
// Author:         Udo Feldkamp
// Last Change:    14.01.2005
// Comment:        sample input file for dsg

SEQUENCE x0 {
  length = 15;
  gc_ratio = [0.4;0.6]; }

SEQUENCETYPE mytype {
  gc_ratio = 0.5;
  length = 10; }

mytype x1, x2, x3;

mytype x4 {
  Tm = [59;61];
  length = 20; }

POOL mypool {
  sequences = x0, x1, x2, x3, x4;
  n_uniqueness = 6;
  Na_conc = .05;
  sample_conc = 2e-7;
  Formamide_conc = 0; }
```

Abbildung 6.8: Beispiel einer DeLaNA-Eingabedatei. Zeilen, die mit `//` beginnen, sind Kommentarzeilen und werden ignoriert. Die Sequenz `x0` soll die in geschweiften Klammern angegebenen Eigenschaften (eine Länge von 15 und einen GC-Gehalt zwischen 40 und 60 %) haben. Die Sequenzen `x1, x2, x3` sollen alle die Eigenschaften des Prototyps `mytype` haben. `x4` ist vom gleichen Sequenztyp, soll also auch einen GC-Gehalt von 50 % haben, allerdings ist die Länge geändert und eine Einschränkung der Schmelztemperatur T_m hinzugefügt worden. Das Objekt `mypool` beschreibt die Eigenschaften der ganzen Sequenzmenge (s. Text).

6.8), in Ausgabedateien sind die tatsächlichen (vom Rechner prognostizierten) Eigenschaften angegeben (Abb. 6.9).

Die Definition eines Sequenzobjekts besteht aus dem Schlüsselwort `SEQUENCE`, einem oder mehreren Objektidentifikatoren, und in geschweiften Klammern die Festlegung der Eigenschaften. Solche Sequenzeigenschaften sind die Länge, der GC-Gehalt, die Schmelztemperatur und freie Enthalpie eines perfekten Duplex aus dieser Sequenz und ihrem Komplement, oder eine vorgegebene Sequenzmaske in IUPAC-Notation (Tab. 6.1), die mit konkreten Basen aus {`A,C,G,T`} gefüllt werden soll. Falls eine Menge von Sequenzen mit gleichen Eigenschaften beschrieben werden sollen, kann ein Sequenztyp mit diesen Eigenschaften definiert werden. Die Sequenzen müssen dann nur noch als zu diesem Typ gehörig deklariert werden. Bei der Definition einzelner Sequenzen eines Typs können auch Eigenschaften des Typs „überschrieben“ werden, so können auch Sequenzen, die sich in vielen, aber nicht allen Eigenschaften gleichen, einfach beschrieben werden.

Ein weiterer Objekttyp sind die Pools, also Sequenzmengen. Deren wichtigste Eigenschaft ist zunächst die Liste der Sequenzen (genauer: der Identifikatoren der Sequenzobjekte), die zu dieser Menge gehören. Weitere Eigenschaften sind die Basisstranglänge n_b, Grenzwerte für Hamming-Distanz, H-Maß oder Homologie, sowie Methode, thermodynamische Parameter und Reaktionsbedingungen für die Berechnung von Schmelztemperatur und freier Enthalpie. Für die Generierung von Konkatenationen mit Verzweigungen gibt es als weitere Eigenschaft eine

```
// File name: diss_example_out.dln
// Created by CANADA v0.1.1

SEQUENCE x0 {
  NA_type = DNA;
  length = 15;
  GC_ratio = 0.6;
  Tm = 55.2532;
  seq_mask = "gatctagcgtgcgag";
}

SEQUENCETYPE mytype {
  NA_type = DNA;
  length = 10;
  GC_ratio = 0.5;
  Tm = [0;100];
  seq_mask = "";
}

SEQUENCE x1 {
  NA_type = DNA;
  length = 10;
  GC_ratio = 0.5;
  Tm = 28.4599;
  seq_mask = "tgcatgtgtc";
}

SEQUENCE x2 {
  NA_type = DNA;
  length = 10;
  GC_ratio = 0.5;
  Tm = 31.1799;
  seq_mask = "ttgttgcagc";
}

SEQUENCE x3 {
  NA_type = DNA;
  length = 10;
  GC_ratio = 0.5;
  Tm = 29.9473;
  seq_mask = "agacctttcg";
}

SEQUENCE x4 {
  NA_type = DNA;
  length = 20;
  GC_ratio = 0.5;
  Tm = 60.6904;
  seq_mask = "attaagtattccagcccgcg";
}

POOL mypool {
  sequences = x0,
              x1,
              x2,
              x3,
              x4;
  n_uniqueness = 6;
  Hamming = 0;
  H_distance = 0;
  sample_conc = 2e-007;
  Na_conc = 0.05;
  formamide_conc = 0;
}
```

Abbildung 6.9: Ausgabedatei, die DSG zur Eingabedatei von Abbildung 6.8 erzeugt hat. Hier wird jede erzeugte Sequenz einzeln mit ihren Eigenschaften aufgeführt. Die Eigenschaft `seq_mask`, die bei der Eingabe der Einschränkung der Basenauswahl an bestimmten Positionen der Sequenz dient, enthält hier jeweils die fertige Sequenz.

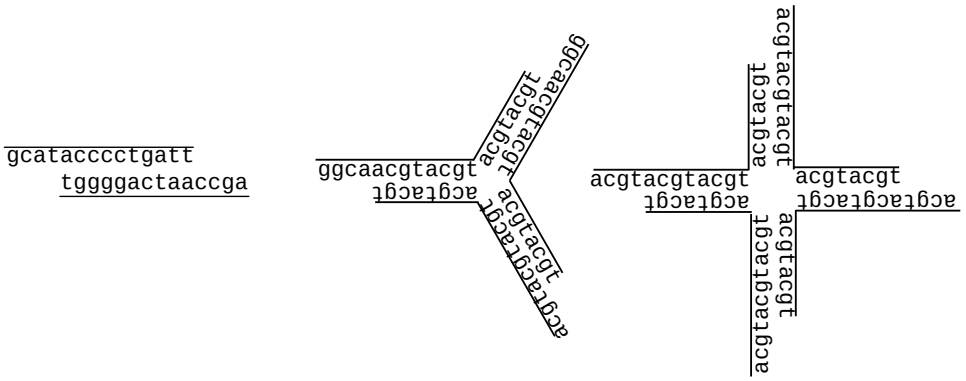


Abbildung 6.10: DNA-Stab, 3-armige und 4-armige DNA-Junction. Sämtliche Arme sind mit sticky Ends versehen. Die Abbildung soll nur die Form der Moleküle zeigen, die Basen sind willkürlich gewählt und hoch unspezifisch.

Grenze, für wieviele Schritte von Beginn der Verzweigung an die Mehrfachverwendung von Basissträngen erlaubt sein soll. Der Benutzer kann mehrere Pool-Objekte definieren, wenn z. B. verschiedene Sequenzmengen eine unterschiedliche Unähnlichkeit der Sequenzen haben sollen.

Für den Entwurf von DNA-Sequenzen, die bestimmte Strukturen bilden sollen, sind einige Makros definiert. Diese werden auch als Objekte definiert, enthalten aber mehrere Sequenzen, die zu einem Baustein der Struktur gehören. Bisher sind drei Typen von Bausteinen definiert: doppelsträngige, lineare Stäbe mit sticky Ends, sowie 3- und 4-armige Junctions, deren Arme ebenfalls mit sticky Ends versehen werden können (Abb. 6.10).

Die Elemente eines Makros sind Sequenzobjekte, deren Eigenschaften entweder innerhalb dieses Makros definiert werden, oder die als normales Sequenzobjekt außerhalb des Makros definiert und im Makro nur noch mit dem Objektnamen referenziert werden (Abb. 6.11). Letzteres bietet die Möglichkeit, eine Sequenz als Teil mehrerer Bausteine zu verwenden. Ein Stab besteht aus einer Kernsequenz (dem doppelsträngigen Teil) und den beiden sticky Ends, eine Junction aus den drei bzw. vier doppelsträngigen Armen und ebensovielen sticky Ends.

6.5.2 DNA-Sequence-Generator

Der DNA-Sequence-Generator (DSG) dient zur Erzeugung einer Menge von DNA-Wörtern. Als Eigenschaften können außer der Basisstranglänge n_b Grenzen für den GC-Gehalt kompletter Sequenzen und von Basissträngen sowie für die Schmelztemperatur und die freie Enthalpie vorgegeben werden. Weitere Optionen in der Version mit graphischer Benutzeroberfläche (Abb. 6.12) sind das Verbot der Teilsequenzen `GGG` (zur Vermeidung von Quadruplexen und `G-G`-Basenpaaren) sowie `ATG`, `GTG` und `TTG` (die bei Verwendung der Sequenz in Organismen zu *Startcodons* übersetzt würden, die eine Interpretation der nachfolgenden Sequenz als Gen zur Folge haben könnten). Außerdem läßt sich die Auswahl des ersten und letzten Basenpaars auf `G-C`-Basenpaare einschränken, um ein Ausfransen (Fraying) der Duplexe zu vermeiden. Der Generator kann nicht nur Sequenzen komplett *de novo* erzeugen, sondern auch teilweise vorgegebene Sequenzen vervollständigen.

Die Variante mit graphischer Benutzeroberfläche zeigt die generierten Sequenzen und ihre wichtigsten Eigenschaften im Hauptfenster an (Abb. 6.13). Ändert man Methode, Parametersatz oder Reaktionsbedingungen für die Berechnung der Schmelztemperatur, so werden die

```
SEQUENCETYPE Terminal {
          length = 20;
          Tm = [55;65]; }

SEQUENCETYPE Variable {
          length = 10;
          GC_ratio = 0.5; }

Terminal x, y, z;

Variable A, B, C;

MACRO_ROD r {
  left_sticky_end.length = 10;
  left_sticky_end.GC_ratio = 0.6;
  core_sequence = x;
  right_sticky_end.length = 10;
  right_sticky_end.GC_ratio = 0.4;  }

MACRO_3WAYJUNCTION j1 {
  arm1 = x;
  arm2 = y;
  arm3 = z;
  sticky_end1 = A;
  sticky_end2 = B;
  sticky_end3 = C;  }

POOL p {
  N_uniqueness = 6;
  Violation_tolerance = 0;
  Sample_conc = 2e-7;
  Na_conc = 0.05;
  Tm_method = NNSantaLucia;   }
```

Abbildung 6.11: Beispiel für eine DeLaNA-Eingabedatei mit Macros für Strukturbausteine. Das Schlüsselwort `MACRO_ROD` kennzeichnet einen DNA-Stab, `MACRO_3WAYJUNCTION` eine 3-armige DNA-Junction. Die Eigenschaften der einzelnen Sequenzen in den Makros können innerhalb der Makrodefinition festgelegt werden, wie bei `left_sticky_end` in Makro `r`, oder es werden vorher definierte Sequenzobjekte durch ihren Identifikator referenziert, wie im Makro `j1`. Das letztere Vorgehen erlaubt die Mehrfachverwendung von Sequenzen, wie bei Sequenz `x`.

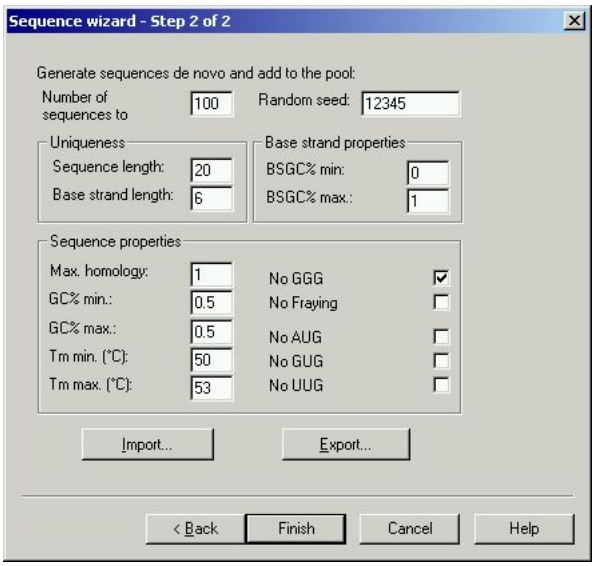


Abbildung 6.12: Dialogfenster zum Einstellen der Eingabeparameter für DSG.

Angaben im Hauptfenster dementsprechend aktualisiert. Das Programm läßt sich daher auch als Schmelztemperaturrechner für Sequenzmengen zweckentfremden.

6.5.3 DNA-Sequence-Compiler

Die Bezeichnung „Compiler“ bezieht sich auf die Programmierung des programmable Self-Assembly. Der Benutzer definiert (programmiert) die Strukturen, die sich durch Self-Assembly bilden sollen, in einer höheren Sprache, der Compiler übersetzt diese in „Maschinenanweisungen“, also in DNA-Moleküle, die *in vitro* der Programmierung folgen. Die höhere Sprache ist bei der Konsolen-basierten Variante DeLaNA, bei der älteren Version mit graphischer Benutzeroberfläche besteht sie aus Regeln einer regulären Grammatik. Diese Regeln haben die Form $A \rightarrow xB$, wobei A und B Variablen sind und x ein Terminalsymbol. Bei den DNA-Stäben, auf deren Entwurf die ältere Version beschränkt ist, entspricht die Kernsequenz dem Terminalsymbol, die sticky Ends stellen die Variablen dar. So wie durch wiederholte Variablenersetzung durch Anwendung der Grammatik-Regeln Terminalsymbole sukzessive aneinander gehängt werden, so führen die sticky Ends zur Aneinanderreihung der Stäbe, wobei die Kernsequenzen allerdings durch die (doppelsträngig gewordenen) sticky-End-Bereiche voneinander getrennt bleiben. Für ein Beispiel hierzu siehe Abschnitt 8.1.

Die neuere, Konsolen-basierte Version des Compilers erlaubt auch den Entwurf von 3- und 4-armigen Junctions. Da sich diese nicht mehr auf Regeln einer regulären Grammatik abbilden lassen, wurden entsprechende Makros für DeLaNA definiert (s. o.).

Der Compiler erzeugt zunächst die Kernsequenzen von Stäben und die Armsequenzen der Junctions, wobei bei letzteren die Auswahl der Startknoten wie in Abschnitt 6.3. beschrieben modifiziert wurde. Anschließend werden die sticky Ends generiert, indem die jeweils benachbarten Kern- und Armsequenzen verlängert werden.

Beispiele für mit dem Compiler erzeugte Sequenzen für das Strukturdesign sind in Kapitel 8 zu finden.

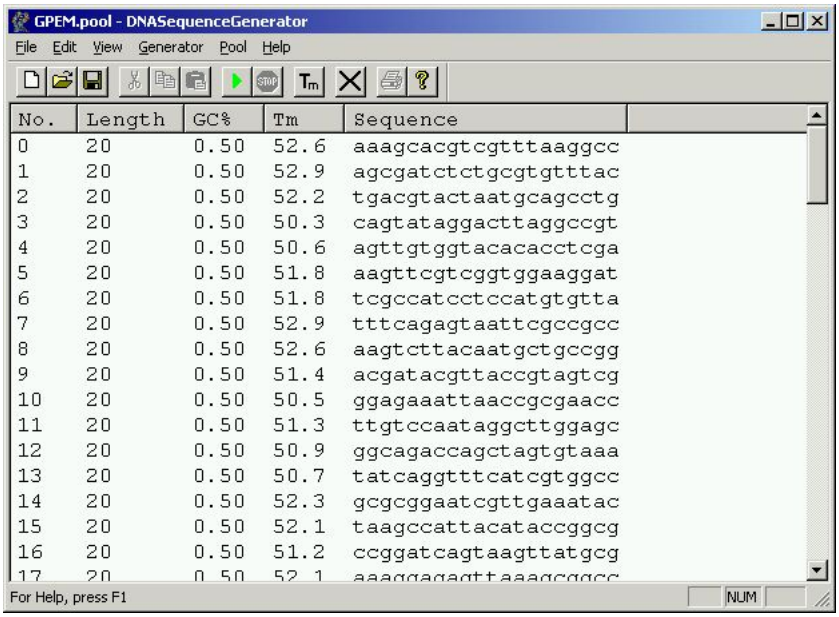


Abbildung 6.13: Hauptfenster von DSG. Gezeigt werden die erzeugten Sequenzen sowie ihre Eigenschaften.

Kapitel 7

Experimente zum DNA-Sequence-Generator

Die folgenden Experimente dienen dazu, die Güte der DNA-Sequenzen zu überprüfen, die mit dem DNA-Sequence-Generator generiert wurden. Die meisten Experimente sind *in silico*, d. h. im Rechner durchgeführt worden. Daher basieren die Aussagen zur Güte der Sequenzen nur auf Modellen, aufgrund derer bestimmte Moleküleigenschaften prognostiziert werden. In den meisten Fällen wurde mit der Berechnung der freien Enthalpie von Hybridisierungsreaktionen aber ein Modell gewählt, das z. Z. als am realistischsten angesehen wird.

Schließlich wird auch ein erstes *in vitro* durchgeführtes Experiment vorgestellt. Es prüft die tatsächliche Güte einer kleinen Stichprobe von entworfenen Sequenzen.

7.1 Größe der erzeugten Sequenzmengen

7.1.1 Einleitung

Um größtmögliche Unähnlichkeit zwischen den generierten Sequenzen zu erzielen, sollte beim Sequenzdesign, das auf der n_b-Uniqueness basiert, ein möglichst kleiner Wert für n_b gewählt werden. Allerdings beschränkt ein konkretes n_b die Anzahl der Sequenzen, die ohne Verletzung der n_b-Uniqueness erzeugt werden können [53]. Die Anzahl N_{bs} von Basissträngen der Länge n_b ist

$$N_{bs} = 4^{n_b}. \tag{7.1}$$

Da für jeden verwendeten Basisstrang dessen Komplement ausgeschlossen wird, darf nur die Hälfte der N_{bs} Basisstränge in den generierten Sequenzen vorkommen. Ist n_b gerade, so müssen außerdem die selbstkomplementären Basisstränge ausgeschlossen werden. Es gibt keine selbstkomplementären Basisstränge ungerader Länge, da die mittlere Base nicht komplementär zu sich selbst sein kann. Für die Anzahl N_{useful} der Basissequenzen, die tatsächlich verwendet werden können, ergibt sich somit

$$N_{useful} = \begin{cases} \frac{N_{bs} - 4^{n_b/2}}{2} & \text{falls } n_b \text{ gerade,} \\ \frac{N_{bs}}{2} & \text{sonst.} \end{cases} \tag{7.2}$$

Da eine Sequenz der Länge n_s aus $n_s - n_b + 1$ Basissträngen besteht, gilt für die Anzahl N_{seq} der Sequenzen, die sich erzeugen lassen

$$N_{seq} = \left\lfloor \frac{N_{useful}}{n_s - n_b + 1} \right\rfloor . \tag{7.3}$$

Diese Anzahl ist eine obere Schranke, es lassen sich also maximal N_{seq} Sequenzen erzeugen. Tatsächlich zerschneiden die gefundenen Pfade den Graphen derart, daß oft einzelne Basisstränge ungenutzt bleiben, die keinen vollständigen Pfad mehr bilden können, da sie nicht direkt verbunden sind.

Dieses Experiment soll ermitteln, wie groß dieser „Verschnitt“ an nicht mehr zu verwendenden Basissträngen ist, wie nahe also die Größen der erzeugten Sequenzmengen an die errechneten maximalen Größen herankommen.

7.1.2 Material und Methoden

Für alle Basisstranglängen $n_b \in \{4, 5, 6, 7\}$ und alle Sequenzlängen $n_s \in \{10, \ldots, 40\}$ wurden mit dem DNA-Sequence-Generator je 10 Sequenzmengen generiert und deren Größe gemittelt. Es wurden keine zusätzlichen Einschränkungen bzgl. der Sequenzeigenschaften gemacht.

Um die Auswirkungen der Beschränkungen von Sequenzeigenschaften zu messen, wurden für $n_b = 4$ und $n_s = 10$ für die Beschränkung des GC-Gehalts auf 0.0, 0.1, …, 0.9 und 1.0 je zehn Sequenzmengen generiert und deren Größe gemessen. Gleiches wurde für die Einschränkung der Schmelztemperatur auf 0–20, 20–24, 24–28, 28–32, 32–36, 36–40 und 40–100 °C durchgeführt.

7.1.3 Ergebnisse und Diskussion

Tabelle 7.1 zeigt die durchschnittlichen Größen der Sequenzmengen über zehn Versuche, die theoretische obere Schranke für die Größe und den prozentualen Anteil ersterer an letzterer. In den meisten untersuchten Fällen werden 80 bis 90 % der maximal möglichen Sequenzen gefunden. Da wie oben ausgeführt dieses Maximum eher zu hoch geschätzt ist, kann die Ausbeute als zufriedenstellend angesehen werden. Eine Untersuchung der Topologie der Basisstranggraphen wäre nötig, um nachzuweisen, ob die maximale Anzahl überhaupt erreicht werden kann, oder ob die gefundenen Pfade für bestimmte Belegungen von n_b und n_s den Graphen zwangsläufig so zerschneiden, daß eine Mindestanzahl nicht mehr zu verwendender Basisstränge übrigbleibt. Leider ist eine solche topologische Untersuchung nicht einfach. Interessant wäre sie auch im Zusammenhang mit der in Abschnitt 5.2 diskutierten Frage, ob Eulerpfade überhaupt noch möglich sind, wenn man selbstkomplementäre Knoten sowie solche, die zu verwendeten Basissträngen komplementär sind, ausschließt.

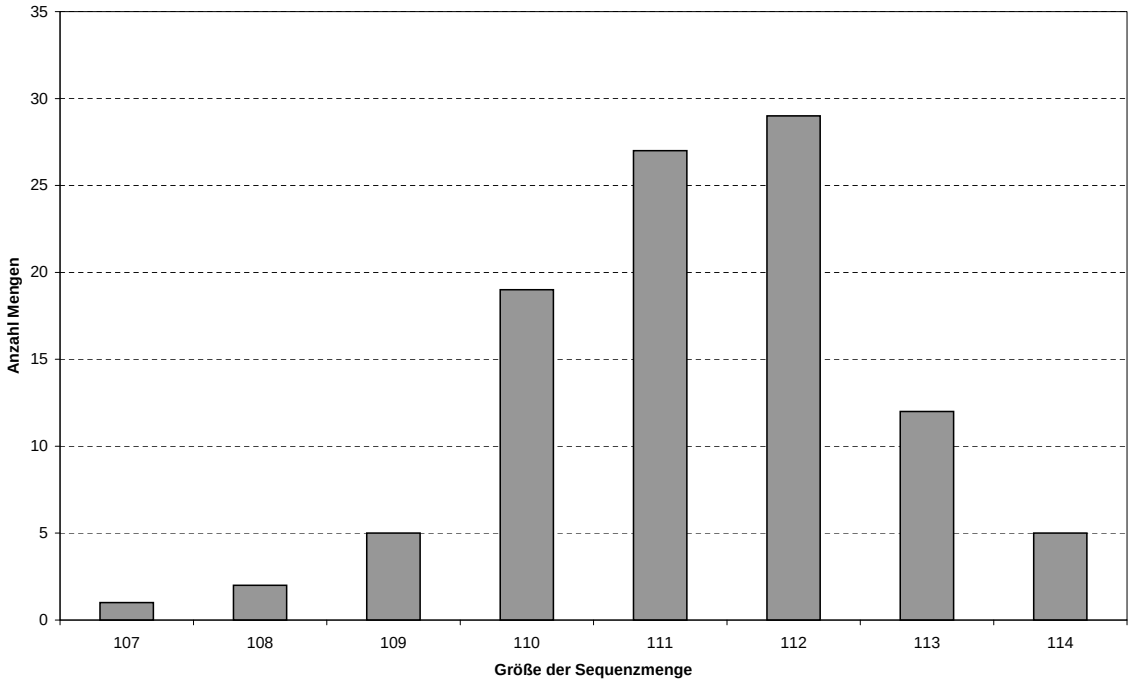
Die Stichprobengröße wurde mit jeweils zehn Sequenzmengen recht klein gewählt, um die Rechenzeit zu beschränken. Für einzelne (n_b, n_s)-Paare wurden weitere Sequenzmengen erzeugt (für Stichprobengrößen von 20, 50 bzw. 100), die die hier gezeigten Mittelwerte bestätigen (Daten hier nicht gezeigt). Zudem zeigte sich, daß die Sequenzmengengröße recht stabil ist, sie schwankt über mehrere Versuche nur um wenige Sequenzen (Abb. 7.1). Aufgrund dieser Stabilität kann eine Stichprobengröße von 10 als ausreichend angesehen werden.

Die hier erzeugten Sequenzen wurden bei der Generierung in ihren Eigenschaften nicht eingeschränkt. Beschränkungen wie verbotene Subsequenzen würden zusätzliche Basisstränge von der Pfadsuche ausschließen, die Sequenzausbeute also verringern. Bei anderen Beschränkungen, die Eigenschaften kompletter Sequenzen wie die Schmelztemperatur oder den GC-Gehalt

betreffen, ist eine solche Auswirkung nicht derart offensichtlich. Sequenzen, die diese Einschränkungen erfüllen, bilden zwar eine Teilmenge A aller Sequenzen der Länge n_s, eine n_b-unique Menge ist selbst aber auch nur eine recht kleine Teilmenge $B \subset \{\texttt{A,C,G,T}\}^{n_s}$. Gilt $|A| \gg |B|$, ist die gleichzeitige Erfüllung von n_b-uniqueness und der anderen Sequenzeigenschaften ohne Verlust an Sequenzausbeute durchaus denkbar.

Wie Abbildung 7.2 zeigt, verringert eine solche Beschränkung tatsächlich nicht automatisch die Sequenzausbeute. Läßt man nur Sequenzen mit einem GC-Gehalt von 50 % zu, ergibt sich praktisch kein Verlust (13 Sequenzen im Mittel gegenüber 13.2 ohne Beschränkung). Hierbei muß kein Basisstrang von der Verwendung ausgeschlossen werden, da für jeden Basisstrang mit einem GC-Gehalt von r % ein Basisstrang mit einem GC-Gehalt von $100 - r$ % existiert, der somit den GC-Gehalt der gesamten Sequenz wieder auf 50 % ausgleichen kann. Wird der GC-Gehalt dagegen z. B. auf 0 % beschränkt, so werden alle Basisstränge von der Verwendung ausgeschlossen, die mindestens eine Guanin- oder Cytosin-Base enthalten.

Bei der Schmelztemperatur gibt es einen Maximalbereich bei 28 – 32 °C (12.9 Sequenzen im Mittel). Dieser Bereich ist vermutlich auch für Zufallssequenzen gleicher Länge der wahrscheinlichste, da für die mit dem Graphalgorithmus generierten Sequenzen eine Gleichverteilung der nearest-Neighbor-Paare angenommen werden kann.



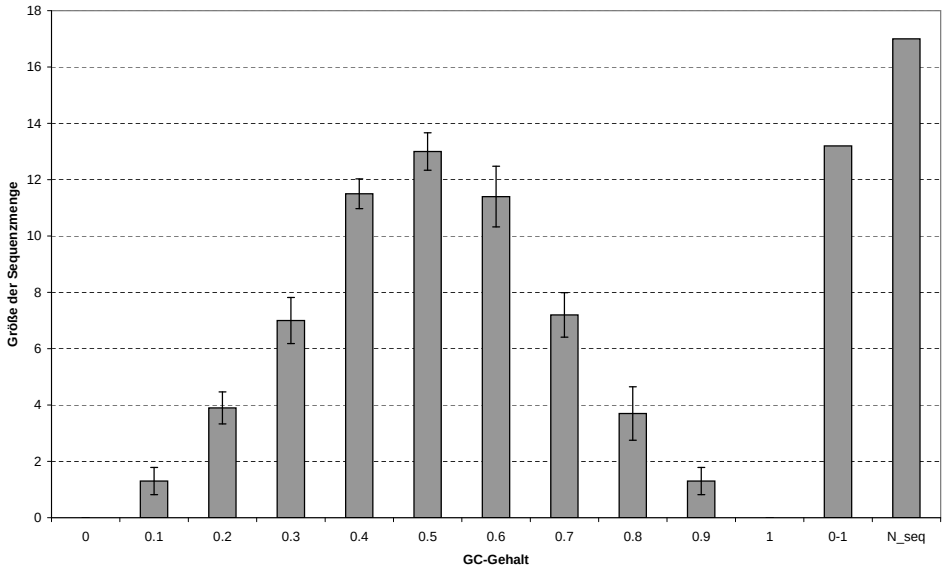
Anz. Sequenzen	107	108	109	110	111	112	113	114
Anz. Mengen	1	2	5	19	27	29	12	5

Abbildung 7.1: Histogramm der Größen von 100 generierten Sequenzmengen mit $n_b = 6$ und $n_s = 20$. Gezeigt wird für jede Anzahl von Sequenzen in einer Menge die Anzahl der gefundenen Mengen, die genau so viele Sequenzen enthalten. Mehr als die Hälfte der Mengen enthalten 111 oder 112 Sequenzen, drei Viertel enthalten 110 – 112 Sequenzen.

n_s \ n_b	4	5	6	7
10	13.2 von 17 **(77.6%)**	70.6 von 85 **(83.1%)**	327.8 von 403 **(81.3%)**	1693.7 von 2048 **(82.7%)**
11	11.9 von 15 **(79.3%)**	59.8 von 73 **(81.9%)**	274.5 von 336 **(81.7%)**	1353.2 von 1638 **(82.6%)**
12	10.4 von 13 **(80.0%)**	53.1 von 64 **(83.0%)**	233.5 von 288 **(81.1%)**	1129.6 von 1365 **(82.8%)**
13	9.3 von 12 **(77.5%)**	47.1 von 56 **(84.1%)**	206.3 von 252 **(81.9%)**	969.9 von 1170 **(82.9%)**
14	8.4 von 10 **(84.0%)**	42.5 von 51 **(83.3%)**	184.0 von 224 **(82.1%)**	849.9 von 1024 **(83.0%)**
15	8.1 von 10 **(81.0%)**	38.6 von 46 **(83.9%)**	165.4 von 201 **(82.3%)**	757.8 von 910 **(83.3%)**
16	7.3 von 9 **(81.1%)**	35.6 von 42 **(84.8%)**	150.5 von 183 **(82.2%)**	682.1 von 819 **(83.3%)**
17	6.8 von 8 **(85.0%)**	33.0 von 39 **(84.6%)**	138.5 von 168 **(82.4%)**	623.4 von 744 **(83.8%)**
18	6.1 von 8 **(76.3%)**	30.7 von 36 **(85.3%)**	127.3 von 155 **(82.1%)**	573.3 von 682 **(84.1%)**
19	5.8 von 7 **(82.9%)**	28.4 von 34 **(83.5%)**	118.4 von 144 **(82.2%)**	531.1 von 630 **(84.3%)**
20	5.6 von 7 **(80.0%)**	26.6 von 32 **(83.1%)**	111.4 von 134 **(83.1%)**	494.8 von 585 **(84.6%)**
21	5.1 von 6 **(85.0%)**	25.1 von 30 **(83.7%)**	103.6 von 126 **(82.2%)**	463.0 von 546 **(84.8%)**
22	5.0 von 6 **(83.3%)**	24.1 von 28 **(86.1%)**	97.7 von 118 **(82.8%)**	433.5 von 512 **(84.7%)**
23	4.9 von 6 **(81.7%)**	22.3 von 26 **(85.8%)**	92.9 von 112 **(82.9%)**	409.6 von 481 **(85.2%)**
24	4.2 von 5 **(84.0%)**	21.7 von 25 **(86.8%)**	88.6 von 106 **(83.6%)**	387.6 von 455 **(85.2%)**
25	4.0 von 5 **(80.0%)**	20.8 von 24 **(86.7%)**	84.5 von 100 **(84.5%)**	367.4 von 431 **(85.2%)**
26	4.0 von 5 **(80.0%)**	19.8 von 23 **(86.1%)**	80.1 von 96 **(83.4%)**	350.8 von 409 **(85.8%)**
27	4.0 von 5 **(80.0%)**	19.0 von 22 **(86.4%)**	76.6 von 91 **(84.2%)**	334.1 von 390 **(85.7%)**
28	3.7 von 4 **(92.5%)**	18.2 von 21 **(86.7%)**	73.7 von 87 **(84.7%)**	320.3 von 372 **(86.1%)**
29	3.4 von 4 **(85.0%)**	17.4 von 20 **(87.0%)**	70.5 von 84 **(83.9%)**	307.1 von 356 **(86.3%)**
30	3.3 von 4 **(82.5%)**	16.9 von 19 **(88.9%)**	67.8 von 80 **(84.8%)**	294.0 von 341 **(86.2%)**
31	3.0 von 4 **(75.0%)**	16.4 von 18 **(91.1%)**	65.4 von 77 **(84.9%)**	283.4 von 327 **(86.7%)**
32	3.0 von 4 **(75.0%)**	15.7 von 18 **(87.2%)**	62.8 von 74 **(84.9%)**	272.5 von 315 **(86.5%)**
33	3.0 von 4 **(75.0%)**	15.0 von 17 **(88.2%)**	60.6 von 72 **(84.2%)**	262.8 von 303 **(86.7%)**
34	3.0 von 3 **(100.0%)**	14.9 von 17 **(87.6%)**	58.8 von 69 **(85.2%)**	253.4 von 292 **(86.8%)**
35	2.8 von 3 **(93.3%)**	14.2 von 16 **(88.8%)**	56.6 von 67 **(84.5%)**	245.1 von 282 **(86.9%)**
36	3.0 von 3 **(100.0%)**	13.9 von 16 **(86.9%)**	54.7 von 65 **(84.2%)**	237.7 von 273 **(87.1%)**
37	2.9 von 3 **(96.7%)**	13.3 von 15 **(88.7%)**	53.0 von 63 **(84.1%)**	230.0 von 264 **(87.1%)**
38	2.9 von 3 **(96.7%)**	12.8 von 15 **(85.3%)**	51.7 von 61 **(84.8%)**	223.0 von 256 **(87.1%)**
39	2.4 von 3 **(80.0%)**	12.5 von 14 **(89.3%)**	50.1 von 59 **(84.9%)**	217.1 von 248 **(87.5%)**
40	2.0 von 3 **(66.7%)**	12.2 von 14 **(87.1%)**	48.2 von 57 **(84.6%)**	211.1 von 240 **(88.0%)**

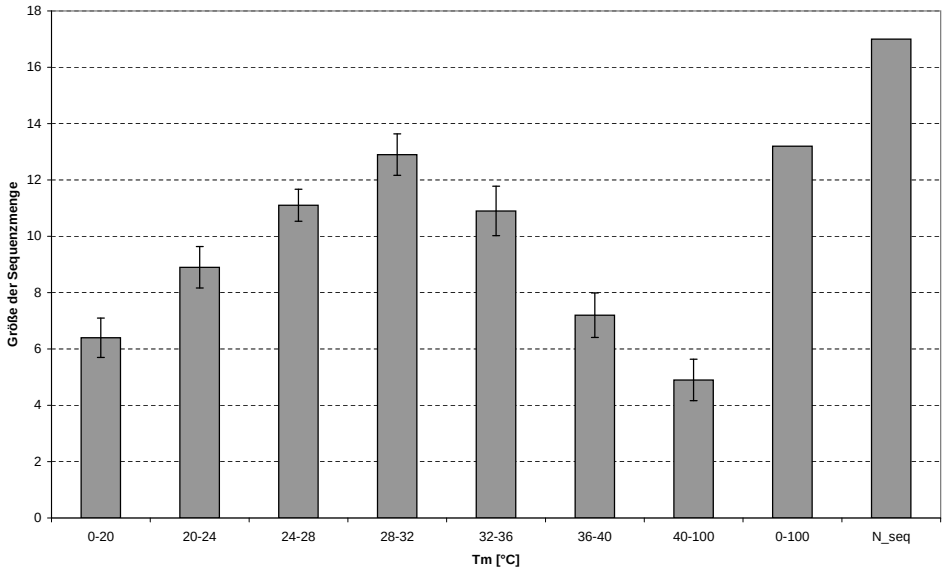
Tabelle 7.1: Größe generierter Sequenzmengen. Für jedes Paar von Basisstranglänge n_b und Sequenzlänge n_s sind im Format „X von Y (Z %)“ die durchschnittliche Anzahl gefundener Sequenzen pro generierter Sequenzmenge, gemittelt über zehn Versuche (X), die maximale Sequenzmengengröße (Y), und der prozentuale Anteil von X an Y (Z) angegeben.

a)



GC-Gehalt	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0-1	N_{seq}
$\overline{N}$	0	1.3	3.9	7	11.5	13	11.4	7.2	3.7	1.3	0	13.2	17

b)



T_m (°C)	0-20	20-24	24-28	28-32	32-36	36-40	40-100	0-100	N_{seq}
$\overline{N}$	6.4	8.9	11.1	12.9	10.9	7.2	4.9	13.2	17

Abbildung 7.2: Sequenzmengengrößen bei Einschränkung von Sequenzeigenschaften. Angegeben ist für jede Einschränkung der Mittelwert $\overline{N}$ über 10 Mengen mit $n_b = 4$ und $n_s = 10$, sowie die Ausbeute ohne Einschränkung und das theoretische Maximum (aus Tab. 7.1). a) Einschränkung des GC-Gehalts. b) Einschränkung der Schmelztemperatur T_m.

7.2 Sekundärstrukturen erzeugter Sequenzen

7.2.1 Einleitung

Eine häufig geäußerte Kritik an der Anwendung der n_b-Uniqueness ist die Behauptung, daß einzelsträngige Sekundärstrukturen nicht vermieden werden. Theoretisch müßte aber, so wie eine geringe Hybridisierungsneigung zwischen zwei verschiedenen Sequenzen durch die n_b-Uniqueness erzielt wird, auch ein Mangel an Hybridisierungsneigung zwischen verschiedenen Bereichen einer Sequenz vorliegen. Kommt ein Basisstrang X in einer Sequenz vor, kann sein Komplement $\overline{X}$ nicht in einem anderen Bereich dieser Sequenz auftauchen. Aber natürlich wächst die Gefahr von Sekundärstrukturbildungen mit wachsender Basisstranglänge n_b. In diesem Experiment soll untersucht werden, wie groß die Neigung zur Bildung einzelsträngiger Sekundärstrukturen für verschiedene Werte von n_b tatsächlich ist.

7.2.2 Material und Methoden

Aus den Basisstranglängen $n_b \in \{4, 5, 6, 7, 8, 9, 10\}$ und den Sequenzlängen $n_s \in \{8, 10, 12, 16, 20, 25, 30, 50, 100\}$ wurden alle (n_b, n_s)-Paare gebildet, für die $n_b < n_s$ gilt. Für jedes Paar wurden 1000 Sequenzen mit dem DNA-Sequence-Generator erzeugt, die jeweils für sich betrachtet die n_b-Uniqueness erfüllen. Außerdem wurde für jedes n_s eine Menge von je 1000 Zufallssequenzen erzeugt. Zusätzlich wurden für jedes n_s drei Sequenzen von Hand konstruiert, die stabile Sekundärstrukturen, genauer gesagt Hairpin Loops mit möglichst vielen Basenpaaren verschiedener Stabilität, bilden sollten. Die jeweils erste konstruierte Sequenz besteht nur aus Wiederholungen von `AT`-Dinukleotiden, da `AT` und `TA` die Nachbarpaare sind, die im Parametersatz für das nearest-Neighbor-Modell von [144] die geringsten Beiträge zur freien Enthalpie liefern. Für diese Sequenz sollte sich eine Hairpin Loop bilden, bei der fast alle Basen der ersten Hälfte mit fast allen Basen der zweiten Hälfte hybridisieren. Ausgenommen sind mindestens drei Basen, die die eigentliche Schleife bilden. Entsprechend besteht die zweite Sequenz nur aus `GC`-Dinukleotiden, da `GC` und `CG` die höchsten Summanden für die freien Enthalpie liefern. Die dritte Sequenz sollte ebenfalls eine Hairpin Loop ausbilden, bei der der doppelsträngige Stamm aber nur etwa ein Viertel der gesamten Sequenzlänge einnimmt. Dieser Stamm besteht auch aus `GC`-Wiederholungen, während die Schleife nur `A` enthält.

Für jede der 68027 Sequenzen wurden mit dem Programm RNAstructure, einer Windows-Variante von mfold ([112, 186]), ihre Sekundärstruktur vorhergesagt. RNAstructure berechnet dabei nicht nur die Struktur mit minimaler freier Enthalpie, sondern auch weitere, weniger stabile Strukturen. Es wurden jeweils die freien Enthalpien ΔG_i aller Strukturen berechnet, für die $\Delta G_i < 0$ gilt, wobei i ein Index über alle berechneten Strukturen der untersuchten Sequenz ist. Als Parametersatz für das nearest-Neighbor-Modell wurden die Resultate für DNA aus [144] eingesetzt. Die Berechnungen beziehen sich auf eine Reaktionstemperatur von 37 °C und eine Na^+-Konzentration von 1 M. Das Programm wurde mit

```
RNAstructure /fold -s fold_me.seq -c out.ct -d
```

aufgerufen, wobei `fold_me.seq` der Name einer Eingabedatei mit der zu faltenden Sequenz ist, `out.ct` der Name der Ausgabedatei mit den freien Enthalpien, und der Schalter `-d` angibt, daß es sich um DNA (und nicht um RNA) handelt.

Aus den freien Enthalpien ΔG_i der Sekundärstrukturen einer Sequenz wurde die Partition Function $q_c = \sum_i e^{-\Delta G_i / RT}$ gebildet (s. Abschnitt 2.3). Für handlichere Werte und um sich wieder im Bereich der freien Enthalpien zu bewegen, wurde aus der Partition Function die *Ensemble-Energie* $\Delta G_{EE} = -\log(q_c) \cdot RT$ berechnet. Die Ensemble-Energie stellt ein realistischeres kumuliertes Maß für die Neigung zu Sekundärstrukturen dar als z. B. die Summe

der einzelnen freien Enthalpien $\sum_i \Delta G_i$, da mehrere verschiedene Konformationen auch untereinander konkurrieren, eine Verdoppelung der Anzahl möglicher Sekundärstrukturen ähnlicher Stabilität also nicht zu einer Verdopplung der Summe der Konzentrationen von Molekülen führt, die diese Konformationen annehmen.

7.2.3 Ergebnisse und Diskussion

In Tabelle 7.2 sind für jedes (n_b, n_s)-Paar sowie für die Zufallssequenzen Mittelwerte, Standardabweichungen und Minima der Ensemble-Energien, sowie die Anzahl von Sequenzen, für die Sekundärstrukturen vorhergesagt wurden, angegeben. Zusätzlich sind Ensemble-Energien für die von Hand konstruierten Hairpin Loops angefügt, um zu verdeutlichen, in welchen Wertebereichen wirklich stabile Sekundärstrukturen zu finden sind. Boxplots, die jeweils Maximum, oberes Quartil, Median, unteres Quartil und Minimum der Ensemble-Energien zeigen, veranschaulichen diese Werte für die größten vier Sequenzlängen (Abb. 7.3 bis 7.6).

Für kurze Sequenzen ($n_s = 8$ bis $n_s = 25$) wurden nur wenige Sekundärstrukturen gefunden. Da diese Einträge eine Ensemble-Energie von 0.0 zugewiesen bekommen, tragen sie nicht zum Mittelwert bei, wohl aber zur Standardabweichung, die daher größer ist als der Mittelwert. Ein Trend, der ab $n_s = 16$ zu erkennen ist, zeigt den Einfluß der Basisstranglänge auf die Stabilität der Sekundärstrukturen. Für $n_b = 4$ und $n_b = 5$, bei längeren Sequenzen auch für $n_b = 6$, sind die Mittelwerte niedriger als für längere Basisstränge. Mit wachsendem n_b nimmt auch die mittlere Ensemble-Energie zu, jedoch ergeben sich ab $n_b = 7$ keine nennenswerten Änderungen mehr. Die Mittelwerte scheinen sich dem der Zufallssequenzen anzunähern, für $n_b = 10$ sind sie von diesen nicht mehr zu unterscheiden. Die Einschränkung der Stabilität einzelsträngiger Sekundärstrukturen ist also nur für sehr kurze Basisstranglängen im Mittel besser als bei reinen Zufallssequenzen. Eine Beschränkung auf $n_b = 4$ oder $n_b = 5$ schränkt aber auch die Anzahl der erzeugbaren Sequenzen stark ein. Ob sich größere Anzahlen von Sequenzen mit ähnlich geringer Stabilität überhaupt finden lassen können, müßte eine ähnliche Untersuchung wie die hier gemachte zeigen, aber mit vollständiger Aufzählung aller möglichen Sequenzen pro Länge, was aber aufgrund der immensen Rechenzeit, die dafür notwendig wäre, nicht durchführbar ist.

Tabelle 7.2 und die Boxplots in Abb. 7.3 bis 7.6 zeigen auch, daß mit wachsender Basisstranglänge die gemessenen minimalen Ensemble-Energien tendentiell kleiner werden, also stabilere Konformationen gefunden wurden. Eine Lockerung der n_b-Uniqueness durch größeres n_b erlaubt also größere Ausreißer nach unten. Die relativ gleichbleibenden unteren Quartile in den Boxplots zeigen, daß es sich tatsächlich nur um Ausreißer handelt, nicht um eine Verschiebung aller Werte unterhalb des Medians zu kleineren Energien hin.

Sowohl für die mit DSG als auch für die zufällig erzeugten Sequenzen werden im Mittel Sekundärstrukturen mit wesentlich geringeren Stabilität als für die von Hand konstruierten Sequenzen vorhergesagt. Für sehr kurze Sequenzlängen ist auch hier aufgrund der geringen Anzahl gefundener Faltungen kein sinnvoller Vergleich möglich. Ab $n_s = 16$ zeigt sich aber, daß dieser Unterschied mit wachsender Sequenzlänge ebenfalls zunimmt. Betrachtet man statt der Mittelwerte die Minima, zeigt sich ein etwas anderes Bild. Für kurze Sequenzen (bis $n_s = 25$) gibt es Ausreißer, deren Ensemble-Energie niedriger liegt als die der beiden weniger stabilen konstruierten Sequenzen (AT und GAC in Tab. 7.2). Bei $n_s = 30$ sind die Werte etwa vergleichbar groß, bei $n_s = 50$ liegen die Minima noch in der Nähe der Ergebnisse für die instabilste konstruierte Sequenz (AT), unter den generierten Sequenzen der Länge $n_s = 100$ finden sich nur noch instabilere Ausreißer. Bei geringen Sequenzlängen sollte man also an die Generierung (mit DSG oder einem Zufallssequenzgenerator) einen Sortier- oder Filterschritt anfügen,

$n_s = 8$	4	5	6	7	Z	AT	GC	GAC
Mittelwert	0.00	0.00	0.00	-0.01	-0.01	0.00	0.00	0.00
St.Abw.	0.04	0.07	0.07	0.11	0.12			
Minimum	-1.30	-1.70	-2.30	-2.30	-3.20			
Anzahl	1	2	1	3	2			

$n_s = 10$	4	5	6	7	8	9	Z	AT	GC	GAC
Mittelwert	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	0.00	-1.60	0.00
St.Abw.	0.17	0.15	0.10	0.14	0.14	0.15	0.13			
Minimum	-2.90	-2.50	-1.40	-1.40	-2.20	-2.30	-1.90			
Anzahl	37	32	38	41	31	39	36			

$n_s = 12$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-0.07	-0.08	-0.08	-0.09	-0.07	-0.07	-0.07	-0.08	0.00	-3.60	-1.20
St.Abw.	0.28	0.35	0.33	0.33	0.27	0.30	0.31	0.32			
Minimum	-2.50	-4.40	-3.70	-2.70	-2.70	-3.60	-3.40	-3.80			
Anzahl	96	91	109	108	104	102	100	112			

$n_s = 16$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-0.18	-0.28	-0.29	-0.28	-0.30	-0.32	-0.30	-0.31	-0.60	- 8.00	-3.00
St.Abw.	0.47	0.61	0.66	0.64	0.69	0.76	0.68	0.67			
Minimum	-4.40	-4.10	-4.50	-4.20	-6.60	-7.80	-4.20	-4.00			
Anzahl	235	275	289	285	282	292	291	286			

$n_s = 20$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-0.35	-0.53	-0.58	-0.58	-0.61	-0.55	-0.57	-0.57	-2.00	-12.40	-5.40
St.Abw.	0.64	0.86	0.94	0.93	1.02	0.92	0.95	0.94			
Minimum	-3.80	-5.20	-6.30	-7.60	-7.30	-7.70	-6.70	-7.40			
Anzahl	406	436	462	482	479	466	451	450			

$n_s = 25$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-0.70	-0.93	-1.05	-1.07	-1.13	-1.06	-1.09	-1.04	-3.80	-17.60	-6.90
St.Abw.	0.89	1.07	1.24	1.31	1.34	1.28	1.28	1.22			
Minimum	-5.90	-6.00	-6.90	-7.00	-6.90	-8.90	-7.40	-9.80			
Anzahl	607	665	675	663	689	666	688	673			

$n_s = 30$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-1.01	-1.47	-1.48	-1.56	-1.62	-1.55	-1.65	-1.63	-5.60	-23.60	-9.10
St.Abw.	1.03	1.38	1.37	1.57	1.54	1.55	1.62	1.54			
Minimum	-5.60	-7.50	-7.50	-11.20	-8.30	-10.80	-11.30	-9.90			
Anzahl	733	795	800	798	801	817	820	804			

$n_s = 50$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-2.54	-3.39	-3.77	-3.96	-3.97	-3.94	-4.05	-4.03	-12.64	-45.60	-18.70
St.Abw.	1.42	1.72	1.96	2.10	2.20	2.12	2.21	2.18			
Minimum	-8.34	-9.75	-10.48	-11.50	-16.80	-12.21	-13.06	-14.00			
Anzahl	975	986	986	986	983	984	986	989			

$n_s = 100$	4	5	6	7	8	9	10	Z	AT	GC	GAC
Mittelwert	-6.78	-9.04	-10.09	-10.46	-10.47	-10.61	-10.65	-10.71	-30.08	-100.40	-46.80
St.Abw.	1.78	2.54	2.88	3.22	2.98	3.17	3.18	3.12			
Minimum	-14.04	-18.90	-19.77	-25.33	-21.09	-24.41	-21.40	-22.28			
Anzahl	1000	1000	1000	1000	1000	1000	1000	1000			

Tabelle 7.2: Ensemble-Energien der vorhergesagten Sekundärstrukturen. Jede Tabelle enthält die Ergebnisse für eine Sequenzlänge $n_s \in \{8, 10, 12, 16, 20, 25, 30, 50, 100\}$, über jeweils variierende Basisstranglänge n_b und die Zufallssequenzen (Z). Angegeben sind Mittelwert, Standardabweichung und Minimum der Ensemble-Energien ΔG_{EE} in kcal/mol, sowie die Anzahl der Sequenzen, für die mindestens eine Sekundärstruktur vorhergesagt wurde (von 1000). In den drei ganz rechten Spalten sind die Ensemble-Energien der konstruierten stabilen Sequenzen angegeben, die nur aus `AT`-Binucleotiden bestehen (AT), nur aus `GC`-Binucleotiden (GC), und aus einem `GC`-Stamm mit einer `A`-Schleife (GAC). In diesen Spalten wurde jeweils nur ein Wert gemessen, daher sind weder Standardabweichung noch Minimum angegeben.

der die Ausreißer, also Sequenzen mit vorhergesagten Sekundärstrukturen hoher Stabilität, verwirft.

7.3 Kreuzhybridisierung erzeugter Sequenzen

7.3.1 Einleitung

Das Modell zur Vermeidung von Kreuzhybridisierungen, das im in dieser Arbeit vorgestellten Algorithmus zur Anwendung kommt, also die n_b-Uniqueness, ist rein Zeichenketten-orientiert, abstrahiert also stark von thermodynamisch detaillierten Modellen der Hybridisierung. Die Hoffnung ist, daß durch eine genügend starke qualitative Einschränkung der Sequenzähnlichkeit auch eine quantitative Einschränkung der Stabilität von Kreuzhybridisierungen erzielt wird (siehe auch die theoretische Diskussion in Kapitel 4).

In diesem Experiment soll untersucht werden, wie realistisch diese Hoffnung ist. Dazu werden die Stabilitätsunterschiede von erwünschten und unerwünschten Hybridisierungen für Sequenzmengen, die mit verschiedenen Parametern erzeugt wurden, gemessen und verglichen.

7.3.2 Material und Methoden

Für die Basisstranglängen $n_b \in \{4, 5, 6, 7\}$ und die Sequenzlängen $n_s \in \{8, 10, 12, 16, 20, 25, 30, 50, 100\}$ wurden je 10 Sequenzmengen mit dem DNA-Sequence-Generator erzeugt. Es wurden keine weiteren Einschränkungen der Sequenzeigenschaften vorgenommen. Zum Vergleich wurden außerdem für jede Sequenzlänge 10 Mengen mit Zufallssequenzen generiert. Da die Berechnung der beiden hier verwendeten Gütemaße für Sequenzmengen sehr rechenzeitintensiv sind, wurden die Größen der untersuchten Mengen auf 150 Sequenzen beschränkt.

Für jede Sequenzmenge wurden zwei Werte gemessen. Die Energielücke δF ist die Differenz der freien Enthalpien der stabilsten unerwünschten und der instabilsten beabsichtigten Hybridisierung [1] (s. Abschnitt 4.3.3). Außerdem wurde eine Mittelwert-Variante δF_A gemessen. Für diese werden zunächst sowohl für die Menge der erwünschten als auch für die Menge der unerwünschten Hybridisierungen jeweils der Mittelwert der freien Enthalpien berechnet, und anschließend die Differenz dieser beiden Mittelwerte gebildet. Als erwünschte Duplexe galten die Paarungen von Sequenzen X mit ihrem perfekten Komplement $\overline{X}$, als unerwünscht Hybridisierungen sowohl zwischen zwei Sequenzen X und Y der Menge, wobei auch $X = Y$ gelten kann, und zwischen X und $\overline{Y}$ mit $X \neq Y$. Die minimale freie Enthalpie für einen Duplex wurde wie in Abschnitt 4.2.2 beschrieben berechnet. Die beiden Sequenzen des Duplex wurden mit einer 16 Basen langen Linkersequenz zu einem Einzelstrang verbunden, anschließend wurde mit dem Programm RNAfold aus dem Vienna RNA Package [75] die Sekundärstruktur mit minimaler freier Enthalpie berechnet. Für eine ausführlichere Beschreibung und Diskussion dieser Methode siehe Abschnitt 4.2.2. Hier wurden allerdings die Berechnungen für eine Reaktionstemperatur von 37 °C durchgeführt, der Aufruf von RNAfold lautete hier also:

```
RNAfold -noGU -T 37 -C < sequences.txt > out.txt
```

Für jedes (n_b, n_s)-Paar und bei den Zufallssequenzen für jedes n_s wurden die gemessenen Werte von δF und δF_A über die jeweils zehn Sequenzmengen gemittelt.

7.3.3 Ergebnisse und Diskussion

In Tabelle 7.3 sind die Ergebnisse, also die Mittelwerte und Standardabweichungen von δF und δF_A über je 10 Sequenzmengen zusammengefaßt. Für alle Sequenzlängen n_s ist eine Ver-

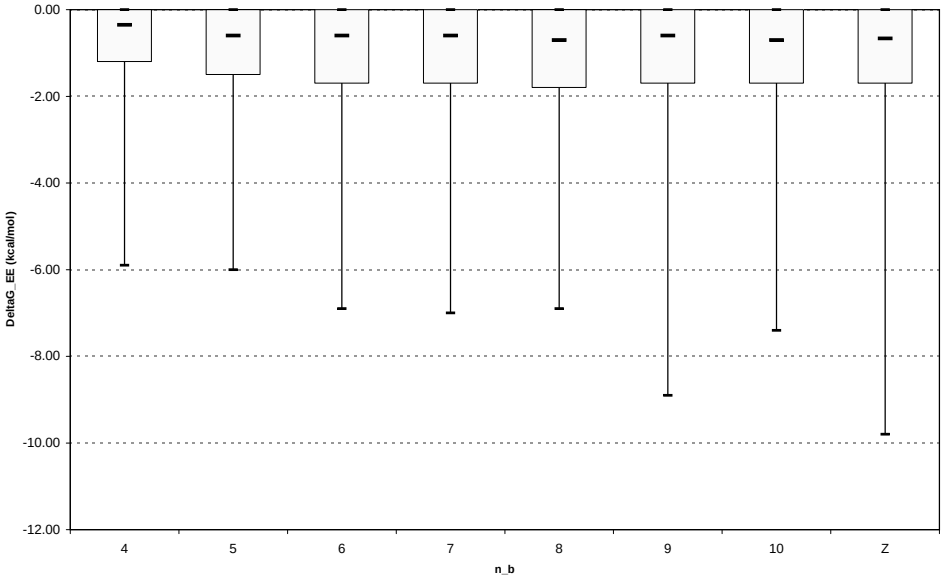


Abbildung 7.3: Boxplot der Ensemble-Energien ΔG_{EE} für die Sequenzlänge $n_s = 25$, über alle Basisstranglängen n_b und die Zufallssequenzen (Z).

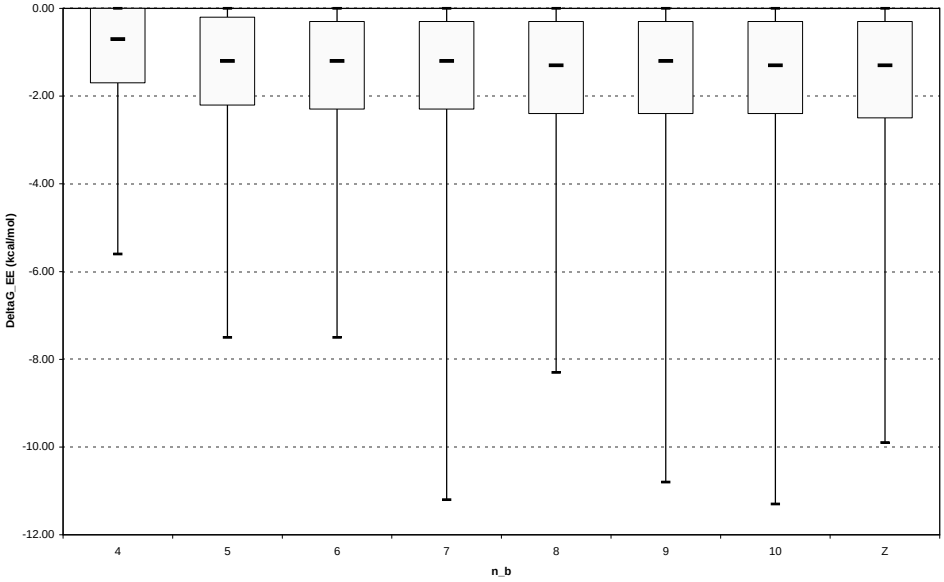


Abbildung 7.4: Boxplot der Ensemble-Energien ΔG_{EE} für die Sequenzlänge $n_s = 30$, über alle Basisstranglängen n_b und die Zufallssequenzen (Z).

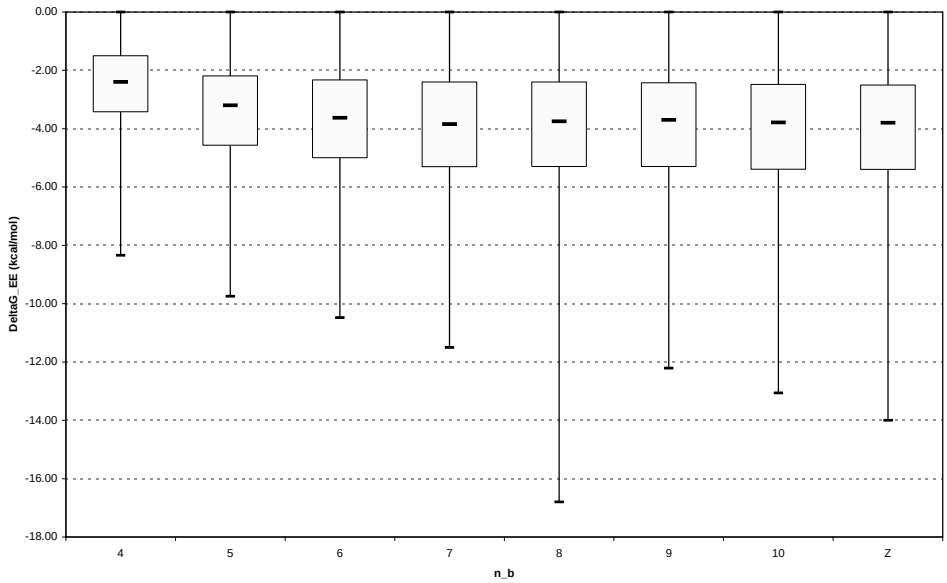


Abbildung 7.5: Boxplot der Ensemble-Energien ΔG_{EE} für die Sequenzlänge $n_s = 50$, über alle Basisstranglängen n_b und die Zufallssequenzen (Z).

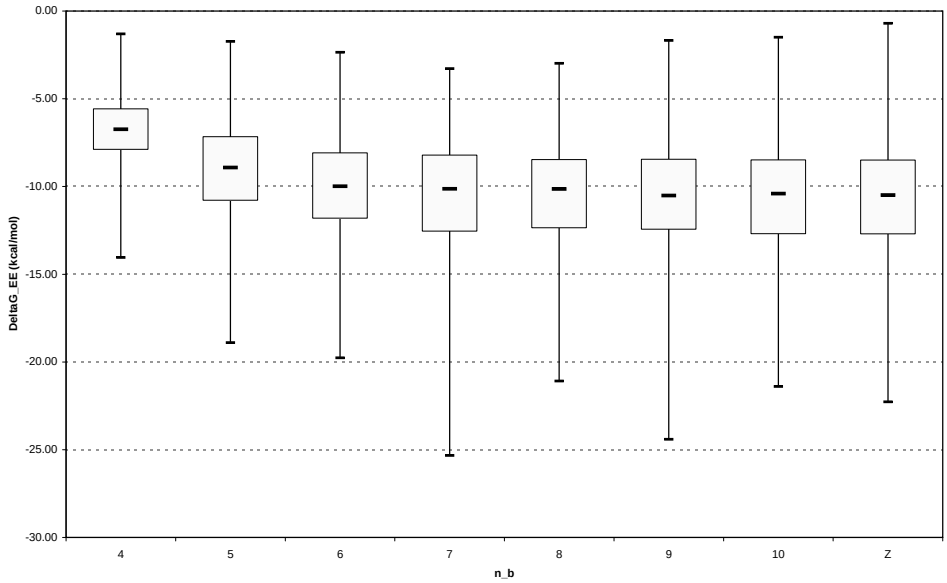


Abbildung 7.6: Boxplot der Ensemble-Energien ΔG_{EE} für die Sequenzlänge $n_s = 100$, über alle Basisstranglängen n_b und die Zufallssequenzen (Z).

ringerung der Energielücke δF mit wachsender Basisstranglänge n_b zu erkennen (Abb. 7.7). Es bestätigt sich also, daß eine Einschränkung der Sequenzähnlichkeit mit der n_b-Uniqueness zumindest bei kleinen n_b zu einer Erhöhung der Spezifität der Hybridisierung führt. Für die meisten Sequenzlängen sind die Energielücken der 7-uniquen Mengen ähnlich denen der Zufallsmengen. Anscheinend bringt nur eine Einschränkung von $n_b \leq 6$ eine Verbesserung der Spezifität gegenüber den Zufallssequenzen. Ob die mit DSG erzeugten Mengen weniger Ausreißer nach unten, also Mengen mit geringer Energielücke aufweisen als Zufallsmengen, kann mit einem Stichprobenumfang von 10 Mengen nicht beantwortet werden, ist aber untersuchenswert.

Für mehrere (n_b,n_s)-Kombinationen sind die Energielücken sogar negativ, d. h. es gibt Kreuzhybridisierungen, die stabiler sind als die gewünschten Duplexe. Dies ist für 8-mere mit beliebigem n_b der Fall, aber auch noch für $n_s = 20$ und $n_b = 7$. Insgesamt zeigt sich, daß mit wachsender Sequenzlänge n_s auch die Basisstranglänge n_b größer wird, ab der die Energielücke negativ wird. Entscheidend für die Vermeidung von Kreuzhybridisierung ist also nicht alleine die Wahl eines kleinen n_b, sondern die Berücksichtigung eines geeigneten Verhältnissen von n_s zu n_b.

Die Mittelwert-Variante der Energielücke δF_A ist für ein festes n_s über alle Werte von n_b und auch für die Zufallssequenzen gleich, die geringen Unterschiede der Mittelwerte liegen innerhalb der Standardabweichungen. Im Mittel sind die Stabilitäten sowohl der erwünschten als auch der unerwünschten Hybridisierungen unabhängig von der Basisstranglänge, die größeren Energielücken δF bei kleineren Werten von n_b müssen also daraus resultieren, daß die einzelnen freien Enthalpien der Sequenzpaare näher beieinander liegen. Es ist unwahrscheinlich, daß ein solches Zusammenrücken unter den erwünschten Hybridisierungen geschieht, da es keinen ersichtlichen Grund gibt, warum eine Verkürzung der Basisstranglänge alleine zu stabileren Sequenzen führen sollte. Eine Verkleinerung der Streubreite der Kreuzhybridisierungen bei gleichbleibendem Mittelwert bedeutet aber, daß nicht nur extrem stabile, sondern auch extrem instabile Kreuzhybridisierungen verhindert werden. Eine genauere Ursachenanalyse würde eine Untersuchung der entsprechenden extrem stabil bzw. instabil kreuzhybridisierenden Sequenzpaare erfordern.

7.4 Vergleich mit veröffentlichten Bibliotheken

7.4.1 Einleitung

Um die Qualität der Ausgabe des Graphalgorithmus mit der anderer Algorithmen zu vergleichen, wurden einige veröffentlichte Sequenzbibliotheken gewählt, mit dem DNA-Sequence-Generator Mengen gleicher Größe erzeugt, und mehrere Eigenschaften jeweils beider Mengen gemessen [54].

7.4.2 Material und Methoden

Für jede Sequenzbibliothek aus der Literatur wurde mit dem DNA-Sequence-Generator eine Menge mit gleich vielen Sequenzen gleicher Länge erzeugt. Dabei wurden die Sequenzeigenschaften wie unten angegeben beschränkt. Gemessen wurden die minimale Länge einzigartiger Subsequenzen n_b (entsprechend der n_b-Uniqueness), die paarweise Homologie sowohl zwischen den Sequenzen als auch zwischen Sequenzen und den Komplementen der anderen Sequenzen (s. Abschnitt 4.1), die Energielücke (s. Abschnitt 4.3.3), sowie die Schmelztemperatur. Die Energielücke δF wurde nicht nur, wie von Ackermann und Gast definiert, als Differenz der freien

			δF					δF_A		
$n_s = 8$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	-1.10	-6.70	-8.35	-10.69	-9.51	7.59	7.53	7.50	7.50	7.52
St.Abw.	1.60	1.61	0.67	0.89	1.66	0.18	0.07	0.10	0.17	0.12

			δF					δF_A		
$n_s = 10$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	3.03	-4.27	-6.85	-7.75	-9.50	11.85	11.64	11.53	11.61	11.66
St.Abw.	1.12	1.89	1.92	1.28	1.61	0.24	0.14	0.13	0.14	0.20

			δF					δF_A		
$n_s = 12$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	5.17	-3.20	-6.39	-5.83	-8.51	15.85	15.70	15.61	15.54	15.57
St.Abw.	1.48	2.36	2.31	1.59	2.93	0.34	0.22	0.19	0.19	0.23

			δF					δF_A		
$n_s = 16$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	9.44	3.68	-1.93	-3.43	-4.37	23.52	23.59	23.29	23.29	23.40
St.Abw.	3.01	2.32	1.43	2.19	2.56	0.48	0.13	0.10	0.17	0.15

			δF					δF_A		
$n_s = 20$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	21.38	12.01	2.09	-0.13	-0.56	31.87	31.11	30.88	30.82	30.85
St.Abw.	2.99	2.98	3.55	2.24	3.41	0.60	0.21	0.10	0.19	0.24

			δF					δF_A		
$n_s = 25$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	29.61	19.89	9.82	7.56	5.19	41.38	40.55	40.28	40.00	40.06
St.Abw.	3.73	1.69	2.47	2.03	2.98	1.09	0.25	0.14	0.29	0.20

			δF					δF_A		
$n_s = 30$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	39.71	25.60	20.17	13.80	9.53	50.50	49.88	49.40	49.25	49.12
St.Abw.	4.03	4.87	2.91	3.17	4.24	1.17	0.22	0.14	0.13	0.22

			δF					δF_A		
$n_s = 50$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	79.25	64.04	47.29	38.55	37.31	89.03	87.12	85.78	85.44	85.37
St.Abw.	6.77	4.59	4.48	2.10	4.77	2.27	0.41	0.24	0.20	0.30

			δF					δF_A		
$n_s = 100$	4	5	6	7	Z	4	5	6	7	Z
Mittelwert	182.02	149.10	135.26	116.64	117.06	182.02	178.41	176.48	175.54	175.37
St.Abw.	7.50	10.65	9.21	5.15	5.79	7.50	1.40	0.56	0.19	0.58

Tabelle 7.3: Energielücke für Sequenzmengen mit verschiedenen Parametern. Jede Teiltabelle zeigt die Ergebnisse für eine Sequenzlänge $n_s \in \{8, 10, 12, 16, 20, 25, 30, 50, 100\}$. Für jedes n_s wurden Sequenzmengen mit $n_b \in \{4, 5, 6, 7\}$ sowie eine Menge mit Zufallssequenzen (Z) untersucht. Angegeben sind Mittelwerte und Standardabweichungen für Energielücke δF (links) und der Mittelwert-Variante δF_A (rechts) jeweils über 10 Mengen.

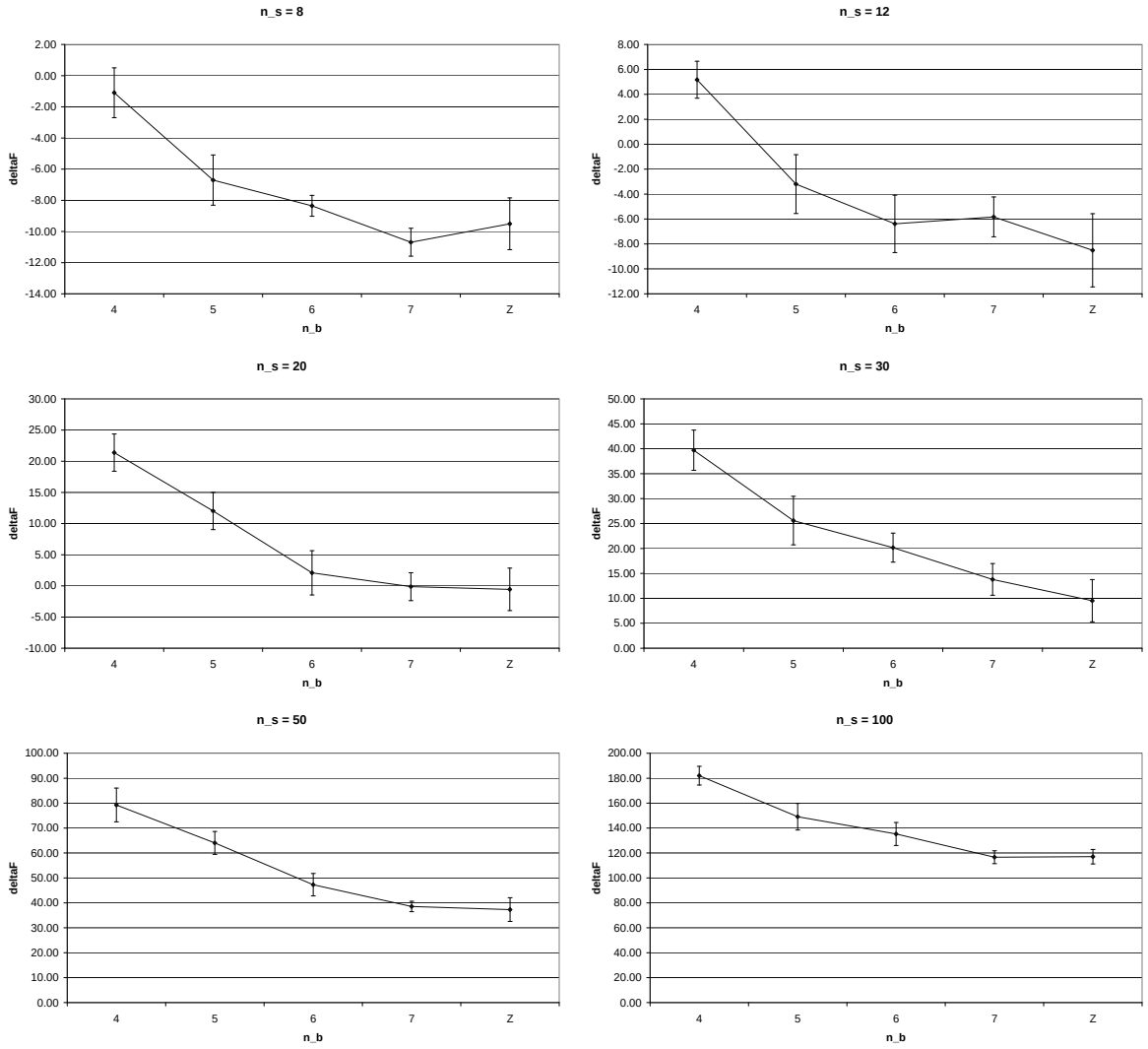


Abbildung 7.7: Vergleich der Energielücken. Für die Sequenzlängen $n_s \in \{8, 12, 20, 30, 50, 100\}$ sind die Mittelwerte der Energielücken δF in Abhängigkeit der Basisstranglänge n_b angegeben. Z kennzeichnet die Zufallsmengen. Man sieht deutlich die Verschlechterung der Energielücke mit wachsendem n_b. Ungünstige Verhältnisse von n_s zu n_b führen außerdem zu negativen Energielücken.

Enthalpien von instabilster gewünschter und stabilster unerwünschter Hybridisierung gemessen [1], sondern auch in einer Mittelwert-Variante δF_A. Für diese werden zunächst sowohl für die Menge der erwünschten als auch für die Menge der unerwünschten Hybridisierungen jeweils der Mittelwert der freien Enthalpien berechnet, und anschließend die Differenz dieser beiden Mittelwerte gebildet. Das Vorgehen bei der Berechnung der freien Enthalpien war dasselbe wie beim Experiment zu Kreuzhybridisierungen in erzeugten Sequenzmengen (s. Abschnitt 7.3). Sämtliche Schmelztemperaturen wurden nach dem nearest-Neighbor-Modell mit Parametern der SantaLucia-Gruppe [145] und für eine DNA-Konzentration von $2 \cdot 10^{-7}$ M und eine Na^+-Konzentration von 0.05 M berechnet.

Für genauere Beschreibungen der Generierungsalgorithmen aus der Literatur siehe Kapitel 5. Die Sequenzen selbst sind unter „Ergebnisse und Diskussion“ angegeben.

Neun 15-mere und drei 20-mere von Arita et al. Diese Sequenzen wurden mit einem Genetischen Algorithmus gefunden, der Hamming-Distanz, Basenwiederholung, GC-Gehalt, falsche Positionierung von Restriktionsschnittstellen und Hybridisierung am $3'$-Ende berücksichtigt [17].

Zur Erzeugung einer Konkurrenzmenge wurden folgende Anforderungen an den DNA-Sequence-Generator gestellt:

- $n_b = 5$
- für die 20-mere ein GC-Gehalt von 50 %, für die 15-mere zwischen 50 und 54 % (wegen der ungeraden Sequenzlänge ist ein GC-Gehalt von genau 50 % nicht möglich)
- für die 20-mere eine Schmelztemperatur zwischen 62 und 63 °C, für die 15-mere zwischen 49 und 52 °C
- keine Subsequenz aus drei oder mehr aufeinanderfolgenden Guanin-Basen
- eine maximale Homologie von 0.5

Sieben 20-mere von Deaton et al. Auch diese Bibliothek wurde mit einem GA erzeugt [45]. Hierbei sollten möglichst wenige Sequenzpaare eine bestimmte Hamming-Distanz unterschreiten.

Zur Erzeugung einer Konkurrenzmenge wurden folgende Anforderungen an den DNA-Sequence-Generator gestellt:

- $n_b = 5$
- ein GC-Gehalt von 50 %
- eine Schmelztemperatur zwischen 58 und 60 °C
- keine Subsequenz aus drei oder mehr aufeinanderfolgenden Guanin-Basen
- eine maximale Homologie von 0.35

Zwanzig 15-mere von Faulhammer et al. Diese Sequenzen wurden mit dem Programm PERMUTE solange zufällig mutiert, bis sie mehrere Anforderungen erfüllten: Zwei 20 Basen lange Subsequenzen beliebiger Konkatenationen[1] müssen sich um mindestens 15 Basen unterscheiden, eine Subsequenz der Länge 7 darf nicht zusammen mit ihrem Komplement vorkommen, die Schmelztemperaturen sollten nahe bei 45 °C liegen [52].

Zur Erzeugung einer Konkurrenzmenge wurden folgende Anforderungen an den DNA-Sequence-Generator gestellt:

- $n_b = 5$
- ein GC-Gehalt von 40 %
- eine Schmelztemperatur zwischen 44 und 46 °C
- keine Subsequenz aus drei oder mehr aufeinanderfolgenden Guanin-Basen
- eine maximale Homologie von 0.5

Sieben 20-mere von Shin et al. Auch hier wurde ein Evolutionärer Algorithmus verwendet, der verschiedene auf dem H-Maß basierende Unähnlichkeitsbewertungen, GC-Gehalt, Schmelztemperatur, Selbstkomplementarität und kontinuierliche Basenwiederholungen als Fitness verwendet [153].

Zur Erzeugung einer Konkurrenzmenge wurden folgende Anforderungen an den DNA-Sequence-Generator gestellt:

- $n_b = 5$
- ein GC-Gehalt von 50 %
- eine Schmelztemperatur zwischen 60 und 62 °C
- keine Subsequenz aus drei oder mehr aufeinanderfolgenden Guanin-Basen
- eine maximale Homologie von 0.4

Vierzehn 20-mere von Tanaka et al. Mit ähnlichen Fitnesskriterien wie bei Shin et al., aber durch Simulated Annealing wurde diese Sequenzmenge erzeugt [162].

Zur Erzeugung einer Konkurrenzmenge wurden folgende Anforderungen an den DNA-Sequence-Generator gestellt:

- $n_b = 5$
- ein GC-Gehalt von 50 %
- eine Schmelztemperatur zwischen 59 und 62 °C
- keine Subsequenz aus drei oder mehr aufeinanderfolgenden Guanin-Basen
- eine maximale Homologie von 0.4

	Arita et al.	DSG
Sequenzen	ccgtcttcttctgct	aaagccgtcgtttcc
	ttccctccctctctt	ttgtggtactctgcg
	cgtcctcctcttgtt	tattagatggccgcc
	ccccttcttgtcctt	ctagctcctttgtcg
	tgcccctcttgttct	gcattgtagtggctg
	ctcctcttccttgct	ggcatatagcgtgac
	cttctcccttcctct	gttattgcgacctcg
	ccttccttccctctt	agtcatggaccaacg
	tcccctttgtgtgtgt	gaacggttaccgatc
	gagagagaggccccctatcc	aaagacgtgtgaagtgcgct
	gaagagaagggcacccctcc	gacgaaagttcagcagcgaa
	gtggtgttgcgtcccttccc	tgttaaaatcaggctcgcgc
n_b	10	5
Homologie	0.00 bis 0.73	0.27 bis 0.47
	0.38 (± 0.19)	0.39 (± 0.06)
δF (kcal/mol)	2.60	12.2
δF_A (kcal/mol)	24.09	23.62
T_m (°C) (15-mere)	49.29 bis 52.79	49.29 bis 51.76
	50.61 (± 1.12)	50.63 (± 0.85)
T_m (°C) (20-mere)	62.56 bis 62.72	62.08 bis 62.56
	62.61 (± 0.13)	62.34 (± 0.24)

Tabelle 7.4: Vergleich der Arita-Bibliothek [17] mit einer mit DSG erzeugten Bibliothek. Gezeigt werden die Sequenzen, die minimale Länge einzigartiger Subsequenzen n_b, sowie Homologie, Energielücke δF, deren Mittelwert-Variante δF_A und Schmelztemperatur T_m. In den Feldern für Homologie und T_m stehen in der ersten Reihe Minimal- und Maximalwert in der jeweiligen Sequenzmenge, in der zweiten Durchschnitt und Standardabweichung. Die Schmelztemperatur wurde jeweils für die beiden Teilmengen der 15- und der 20-mere getrennt gemittelt.

7.4.3 Ergebnisse und Diskussion

Vergleich mit Arita et al. Bemerkenswert ist an der Arita-Menge der sehr kleine Bereich der Schmelztemperaturen, der erreicht wurde (Tab. 7.4), obwohl der GA nur auf einen möglichst gleichen GC-Gehalt der Sequenzen hin optimiert, also ein eigentlich nur sehr ungenaues Modell für die Schmelztemperatur verwendet. Allerdings bestehen die 15-mere hauptsächlich aus Cytosin und Thymin, es werden also nur wenige nearest-Neighbor-Paare immer wieder verwendet, was die Ähnlichkeit der Schmelztemperaturen, die nach dem nearest-Neighbor-Modell berechnet wurden, erklärt. Gleiches gilt für die untereinander sehr ähnlich aussehenden 20-mere.

Der DSG erzielt mit expliziter Einschränkung der Schmelztemperatur eine ähnlich schmale Bandbreite. Allerdings sind diese Sequenzen nicht nur dem Augenschein nach bereits wesentlich unähnlicher als die Arita-Sequenzen, auch die Messungen bestätigen diesen Eindruck. Insbesondere ist die minimale Länge einmaliger Subsequenzen n_b für die Arita-Menge wesentlich höher als für die DSG-Menge. Nicht nur kommt der 9-mer `cctcttgtt` zweimal vor (zweite und vierte Sequenz), was bei 15-meren schon zu einer hohen Verwechslungsgefahr führen kann, es gibt 17 7-mere, die als Subsequenz doppelt vorkommen.

Der Unterschied in der Sequenzähnlichkeit zeigt sich auch in der Homologie. Zwar sind die Durchschnittswerte fast gleich, aber in der Arita-Menge schwankt die paarweise Homologie sehr stark. Die Paare mit „perfekter Unähnlichkeit“ (Homologie = 0.0) ergeben sich dadurch, daß in einigen 15-meren weder Adenin noch Guanin vorkommt, deren Komplemente also nur aus Adenin und Guanin bestehen. Da bei der Homologiemessung auch mit den Komplementärsequenzen verglichen wurde, kann bei einem solchen Sequenzpaar keine Base übereinstimmen. Die Betrachtung der Ähnlichkeit der 15-mere unter sich, also ohne die Komplemente, ergibt sehr hohe Homologien, insbesondere auch das Maximum von 0.73. Die DSG-Menge zeigt eine viel schmalere Bandbreite von Homologien, da die erzwungene n_b-Uniqueness zu eine Unähnlichkeit sowohl der Sequenzen untereinander als auch gegenüber ihren Komplementen führt.

Auch die Energielücke spiegelt die mangelnde Sequenzunähnlichkeit der Arita-Menge wieder. Während die Mittelwerte von erwünschten und unerwünschten Hybridisierungen in beiden Mengen jeweils gleich weit voneinander entfernt liegen, zeigen die δF-Werte einen deutlichen Unterschied. Die instabilste erwünschte Konformation ist in der Arita-Menge also kaum stabiler als die stabilste unerwünschte.

Vergleich mit Deaton et al. Mit einer minimalen Länge einzigartiger Subsequenzen von $n_b = 9$ ist die Sequenzähnlichkeit in der Deaton-Menge deutlich höher als in der DSG-Menge (Tab. 7.5). Immerhin fünf 6-mere und drei 7-mere kommen doppelt vor. Die durchschnittliche Homologie bei beiden Mengen ist ähnlich. Während aber in der DSG-Menge fast alle Sequenzpaare eine Homologie von 0.35 besitzen, gibt es in der Deaton-Menge mehr Ausreißer.

Ein ähnliches Bild zeigen die Energielücken. Während für die Mittelwert-Variante bei beiden Mengen etwa gleiche Werte gemessen wurden, ist δF gegenüber der DSG-Menge nahezu halbiert.

Die Schmelztemperaturen fallen bei der DSG-Menge in einen wesentlich kleineren Bereich als bei der Deaton-Menge. Dies ist nicht überraschend, da die Schmelztemperatur bei der Erzeugung der Deaton-Menge überhaupt nicht beachtet wurde, nicht einmal in Form des GC-Gehalts.

[1] die Konkatenationen enthalten dabei noch kurze Spacer-Sequenzen, die hier nicht betrachtet werden.

	Deaton et al.	DSG
Sequenzen	cttgtgaccgcttctgggga	aaagccgtcgtttaaggacc
	cattggcggcgcgtaggctt	accattttggaggtggaacg
	atagagtggatagttctggg	tatatcgtagagccacacgc
	gatggtgcttagagaagtgg	tccgcgtactgataatcctc
	tgtatctcgttttaacatcc	atatgcttaggcacggttgg
	gaaaaaggaccaaaagagag	tctcgtgaattggtctggac
	ttgtaagcctactgcgtgac	ttactcatctctgtgacgcc
n_b	9	5
Homologie	0.25 bis 0.50	0.25 bis 0.35
	0.35 (± 0.07)	0.34 (± 0.03)
δF (kcal/mol)	12.84	22.73
δF_A (kcal/mol)	30.78	30.94
T_m (°C)	52.67 bis 68.91	58.34 bis 59.86
	59.21 (± 5.57)	59.18 (± 0.61)

Tabelle 7.5: Vergleich der Deaton-Bibliothek [45] mit einer mit DSG erzeugten Bibliothek. Das Format entspricht dem von Tabelle 7.4.

Vergleich mit Faulhammer et al. Die minimale Länge einzigartiger Subsequenzen n_b ist auch in der Faulhammer-Menge höher als in der dazu konkurrierenden DSG-Menge (Tab. 7.6). Es gibt zwar nur einen doppelt auftauchenden 7-mer (`tttctcc`), aber 17 doppelt und zwei dreifach vorkommende 6-mere, sowie drei selbstkomplementäre 6-mere. Dies verletzt offensichtlich das Designkriterium, daß innerhalb eines 20-Basen-Fensters über Konkatenationen mindestens 15 Mismatches bestehen müssen. Bzgl. der durchschnittlichen Homologie sind sich beide Mengen ähnlich, jedoch ist auch hier die Schwankung in der vorgegebenen Bibliothek größer als in der mit DSG erzeugten.

Diese hohen Sequenzähnlichkeiten in der Faulhammer-Menge schlagen sich auch in den Energielücken nieder. Die Mittelwert-Varianten sind vergleichbar gut, die Extremwert-Lücke δF ist dagegen für die Faulhammer-Menge nicht nur schlechter als für die DSG-Menge, sie ist sogar negativ. D. h. die stabilste unerwünschte Konformation ist stabiler als die instabilste erwünschte. Eine genauere Analyse zeigt, daß die stabilste Fehlhybridisierung ($\Delta G_{37} = -11.17$ kcal/mol) zwischen der achten und dem Komplement der dritten Sequenz prognostiziert wurde (Abb. 7.8), während der instabilste erwünschte Duplex aus der zwanzigsten Sequenz und ihrem Komplement besteht. Dessen geringe Stabilität wird schon durch den geringen GC-Gehalt angedeutet. Faulhammer et al. berichten im Artikel auch von Problemen, die zu einer falschen Lösung geführt haben, führen diese aber auf Deletionen und Punktmutationen nach Klonierung zurück.

Daß die Sequenzen der Faulhammer-Menge hier eine durchschnittliche Schmelztemperatur von 42 °C haben anstatt wie vom Suchalgorithmus beabsichtigt 45 °C, liegt vermutlich an verschiedenen Parametern zur Berechnung der Schmelztemperaturen (in der Veröffentlichung finden sich dazu leider keine Angaben). Davon abgesehen zeigen die Schmelztemperaturen breitere Schwankungen als in der DSG-Menge.

Vergleich mit Shin et al. Die Shin-Bibliothek zeigt eine nur etwas größere minimale Länge einzigartiger Subsequenzen als die entsprechende DSG-Menge (Tab. 7.7). Es gibt nur einen

	Faulhammer et al.	DSG
Sequenzen	ctcttactcaattct	aaagccgtcaaatac
	catatcaacatctta	tacctttttgtctcg
	atcctccacttcaca	taagtatatcgtgcc
	ttaaaatcttccctc	agtgacactagcatt
	ctatttctccacacc	aagctattgattggc
	gcttcaaacaattcc	cttctctcacctata
	aactctcaaattcaa	ttacagcgttttacc
	ctaacctttacttca	ggcaagaggaataat
	cattccttatcccac	tggtaggccatttaa
	caccctttctcctct	cacttgagtacaaca
	tcctcacattactta	ggatgtccttgttta
	acttcctttatatcc	gcgaaaattaactcc
	ttataacaaacatcc	gtctgagctgataaa
	acataaccctcttca	acaggcgtatctaat
	accttactttccata	gatccggttactaaa
	gtacattctccctac	atgaggcagtcttta
	cataatcttatattc	tgcgactatgttatg
	ataatcacatacttc	acctgactcgtaata
	tccaccaactaccta	accaaaccatgatga
	ttttaaatttcacaa	gtaccgttgaattgt
n_b	8	5
Homologie	0.13 bis 0.67	0.20 bis 0.47
	0.39 ($\pm$ 0.10)	0.39 ($\pm$ 0.06)
δF (kcal/mol)	-0.18	9.89
δF_A (kcal/mol)	17.78	19.37
T_m (°C)	33.44 bis 40.00	44.01 bis 46.00
	42.00 ($\pm$ 4.12)	45.25 ($\pm$ 0.59)

Tabelle 7.6: Vergleich der Faulhammer-Bibliothek [52] mit einer mit DSG erzeugten Bibliothek. Das Format entspricht dem von Tab. 7.4.

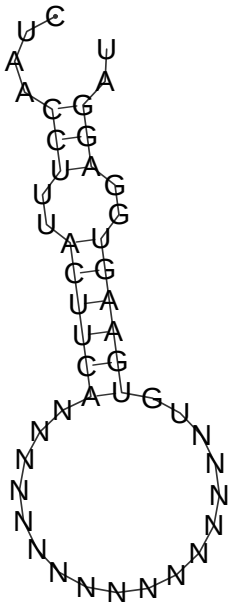


Abbildung 7.8: Stabilste unerwünschte Konformation der Faulhammer-Bibliothek mit ΔG_{37} = -11.17 kcal/mol. Die `N` bezeichnen die nichthybridisierenden Basen der Linkersequenz. Außerdem ist `T`(hymin) durch `U`(racil) ersetzt, da RNAfold die Sequenzen als RNA betrachtet, nicht als DNA.

	Shin et al.	DSG
Sequenzen	`aggcgagtatggggtatatc`	`aaagccgtcgtttaaggagc`
	`cctgtcaacattgacgctca`	`tagtcgcgtgatttggaagg`
	`ttatgattccactggcgctc`	`tacgtctcgaactgatagcg`
	`atcgtactcatggtccctac`	`cttcgtgtcggccatcatat`
	`cgctccatccttgatcgttt`	`tcatgttggcaccgtatgca`
	`cttcgctgctgataacctca`	`ggttcttacgctctactgca`
	`gagttagatgtcacgtcacg`	`ttacacttgaagctggctcg`
n_b	7	5
Homologie	0.25 bis 0.55	0.30 bis 0.40
	0.37 ($\pm$ 0.06)	0.37 ($\pm$ 0.04)
δF (kcal/mol)	23.93	23.23
δF_A (kcal/mol)	31.43	30.46
T_m (°C)	56.95 bis 62.08	60.30 bis 61.08
	59.76 ($\pm$ 2.00)	60.79 ($\pm$ 0.24)

Tabelle 7.7: Vergleich der Shin-Bibliothek [153] mit einer mit DSG erzeugten Bibliothek. Das Format entspricht dem von Tab. 7.4.

	Tanaka et al.	DSG
Sequenzen	cgagacatcgtgcatatcgt	aaagccgtcgtttaaggagc
	tatagcacgagtgcgcgtat	tagtcgcgtgatttggaagg
	gatctacgatcatgagagcg	tacgtctcgaactgatagcc
	tctgtactgctgactcgagt	gctgtctttcgtcaataccg
	cgagtagtcacacgatgaga	tgatcttgtaaaggccaggc
	agatgatcagcagcgacact	tgcagaaaaactatgccgcc
	tgtgctcgtctctgcatact	ctgaacggaatctagtagcg
	agacgagtcgtacagtacag	tacgatacttggcgagccat
	atgtacgtgagatgcagcag	gcgcggacaattcattggtt
	atcactactcgctcgtcact	aatcgcagtacagatggtgg
	tcagagatactcacgtcacg	gtctacggttctcttacgct
	gacagagctatcagctactg	cttaggcaggtgccacatat
	gctgacatagagtgcgatac	ggatgaccagagcacttcaa
	acatcgacactactacgcac	ccgcaatccggtgaaattag
n_b	9	5
Homologie	0.25 bis 0.45	0.25 bis 0.40
	0.38 ($\pm$ 0.05)	0.36 ($\pm$ 0.04)
δF (kcal/mol)	14.96	19.32
δF_A (kcal/mol)	29.16	30.74
T_m (°C)	58.82 bis 62.81	59.34 bis 61.69
	60.62 ($\pm$ 1.18)	60.33 ($\pm$ 0.73)

Tabelle 7.8: Vergleich der Tanaka-Bibliothek [162] mit einer mit DSG erzeugten Bibliothek. Das Format entspricht dem von Tab. 7.4.

doppelt vorkommenden 6-mer sowie einen dreifach und sieben doppelt vorkommende 5-mere. Auch der Unterschied in der Homologie ist zwischen beiden Mengen recht gering, es zeigen sich aber auch hier wieder größere Schwankungen bei der vorgegebenen Menge.

Die Energielücken sind bei beiden Mengen etwa gleich groß. Die Schmelztemperaturen liegen bei der DSG-Menge in einem deutlich engeren Bereich, dessen Breite ist jedoch auch bei der Shin-Menge bereits akzeptabel.

Vergleich mit Tanaka et al. Mit $n_b = 9$ zeigt die Tanaka-Menge eine deutlich größere minimale Länge einzigartiger Subsequenzen als die DSG-Menge (Tab. 7.8). Sie enthält zwei doppelt vorkommende 8-mere und einen selbstkomplementären 8-mer sowie 13 doppelt vorkommende 7-mere. Dafür ist die Homologie bei beiden Mengen ähnlich, auch in der Größe der Schwankungen.

Die Energielücke δF ist bei der Tanaka-Menge deutlich kleiner als bei der DSG-Menge, während die Mittelwert-Variante in beiden Fällen etwa gleich ist. Die Schmelztemperaturen zeigen wiederum bei der DSG-Menge eine geringere Streuung, diese ist aber auch bei der Tanaka-Bibliothek akzeptabel.

Fazit Wie der Vergleich zeigt, kann der DNA-Sequence-Generator Sequenzmengen erzeugen, die nicht nur mit anderen, veröffentlichten Mengen vergleichbar sind, sondern die oft sogar besser sind, sowohl bzgl. der Sequenzunähnlichkeit als auch bzgl. der Schmelztemperatur. Bei

den meisten Mengen gelang die Generierung einer ausreichenden Anzahl von Sequenzen unter den angegebenen Einschränkungen der Eigenschaften bereits im ersten Versuch, für keine Menge mußten mehr als drei Versuche unternommen werden. Außerdem wurden auch die Einschränkungen nicht erst nach Analyse der veröffentlichten Bibliotheken gewählt, um diese auf jeden Fall zu schlagen, sondern es wurde versucht, eine möglichst gute Menge zu erzeugen, und erst diese anschließend mit der veröffentlichten verglichen.

Stärkste Konkurrenz für den graphbasierten Algorithmus sind die evolutionären Verfahren, die viele verschiedene Eigenschaften als Fitness-Kriterien berücksichtigen.

Desweiteren suggerieren die Ergebnisse, daß eine größere Basisstranglänge n_b mit einer geringeren Energielücke δF, also einem kleineren Störabstand zwischen erwünschter und unerwünschter Hybridisierung, korrespondiert.

7.5 Erstellung einer Oligomer-Bibliothek für die DDI

7.5.1 Einleitung

Ein wichtiges Werkzeug zur Untersuchung der Funktion von Proteinen sind Protein-Microarrays. Die direkte Immobilisierung der Proteine auf dem Träger ist problematisch, da u. a. die Funktionalität der Proteine auf der Oberfläche eingeschränkt wird. Bei der *DNA-Directed Immobilization* (DDI) werden die Proteine mit DNA-Oligomeren verbunden, die komplementär zu Sonden auf einem DNA-Microarray sind. Durch die Hybridisierung der DNA-Moleküle werden die Proteine immobilisiert, ohne daß sie mit der Oberfläche in Berührung kommen müssen (s. Abschnitt 3.2.2). Hierfür ist es erforderlich, daß die Oligomere spezifisch sowie mit hoher und für alle Oligomere gleicher Effizienz hybridisieren. Die gleiche Problemstellung ergibt sich natürlich auch für andere Anwendungen für DNA-Microarrays, wie z. B. dem oberflächenbasierten DNA-Computing (s. Abschnitt 3.2.1).

In diesem Experiment, durchgeführt in Zusammenarbeit mit der Arbeitsgruppe von Prof. Niemeyer am Fachbereich Chemie der Universität Dortmund. wurde eine Sequenz-Bibliothek für die DDI generiert und ihre Qualität *in vitro* überprüft [55]. Zum Vergleich wurde eine veröffentlichte Bibliothek einer anderen Gruppe ebenfalls getestet.

7.5.2 Material und Methoden

Zum Vergleich wurde die Bibliothek aus 14 20-meren von Tanaka et al. [162] gewählt, da diese im theoretischen Vergleich sehr gut abgeschnitten hatte (s. Abschnitt 7.4). Die Sequenzen von Shin et al. sind zwar nach den dabei untersuchten Kriterien noch besser, allerdings ist die Bibliothek mit nur sieben Sequenzen zu klein.

Mit dem DNA-Sequence-Generator wurde eine Menge von 14 22-meren mit folgenden Eigenschaften erzeugt:

- Die Basisstranglänge beträgt $n_b = 5$
- Der GC-Gehalt wurde auf 50 % beschränkt.
- Die Schmelztemperatur wurde auf 62 bis 64 °C beschränkt.
- Nicht mehr als zwei Guanin-Basen dürfen unmittelbar aufeinander folgen.

Die Schmelztemperaturen wurden nach dem nearest-Neighbor-Modell mit den Parametern der SantaLucia-Gruppe [145] für eine DNA-Konzentration von $2 \cdot 10^{-7}$ M und eine Na^+-Konzentration von 0.05 M berechnet.

Name	Sequenz	ΔG^{ss}_{25} (kcal/mol)	ΔG^{ss}_{37} (kcal/mol)	T_m (°C)	ΔG^{pm}_{37} (kcal/mol)
F1	CCTGCGTCGTTTAAGGAAGTAC	-0.45	0.00	62.2	-26.96
F2	CAGCCAAGATTCTTTTACCGCC	0.00	0.00	63.1	-27.16
F3	CCATCATGTGTGCCGAGATATG	0.00	0.00	62.6	-26.29
F4	CTTCTCCTAACTGCACGGAATG	-0.25	0.00	63.0	-26.50
F5	GGTCCGGTCATAAAGCGATAAG	-0.52	0.00	62.2	-26.61
F6	GTCCTCGCCTAGTGTTTCATTG	0.00	0.00	62.2	-26.77
F7	GGATCTGGCGCATAGACAATTC	-0.23	0.00	62.7	-26.93
F8	CACGTCACTGTTAATCCGAAGC	-0.14	0.00	62.7	-27.36
F9	GTGGAAAGTGGCAATCGTGAAG	-0.24	0.00	62.4	-27.41
F10	GGACGAATACAAAGGCTACACG	0.00	0.00	62.1	-26.89
F11	CAAGGTCTGCTTGATTTGGAGG	-2.82	-1.70	62.2	-26.55
F12	GTTTTGAACGTAGTAGAGCCGG	-0.79	-0.30	62.2	-26.96
F13	GTAGGTGTCGGTGCGAAATTAG	-1.20	-0.40	62.1	-26.89
F14	CTAGAACCGTTACGAGTTTGCG	-1.44	0.00	63.5	-27.28

Tabelle 7.9: Oligomer-Sequenzen der F-Bibliothek, generiert mit dem DNA-Sequence-Generator. Gezeigt werden die Sequenzen in $5' \rightarrow 3'$-Richtung, die minimale freie Enthalpie von Sekundärstrukturen gemäß Vorhersage mit RNAfold für 25 °C (ΔG^{ss}_{25}) und 37 °C (ΔG^{ss}_{37}). Schmelztemperatur (T_m) und freie Enthalpie der perfekten Duplexe ΔG^{pm}_{37} wurden nach dem nearest-Neighbor-Modell mit Parametern von SantaLucia et al. berechnet [145]. Die ersten zehn Sequenzen wurden für die *in vitro*-Untersuchung ausgewählt.

Um die Gefahr von einzelsträngigen Sekundärstrukturen zu minimieren, wurden für alle 14 Sequenzen Sekundärstrukturen mit Hilfe des RNAfold Webservers vorhergesagt, der auch thermodynamische Parameter für DNA zur Verfügung stellt [74]. Die zehn Sequenzen $F1$ bis $F10$, deren vorhergesagten Sekundärstrukturen die geringste Stabilität für 25 °C besitzen, bilden die endgültige F-Bibliothek. Diese Oligos zeigen praktisch keine Sekundärstrukturen für 37 °C (Tab. 7.9). Dieselbe Auswahl wurde für die Sequenzen der Tanaka-Menge durchgeführt (Tab. 7.10), so daß die T-Bibliothek die 10 Oligomere $T1$ bis $T10$ mit den instabilsten vorhergesagten Sekundärstrukturen enthält. Zum Suchalgorithmus für die Tanaka-Menge siehe Kapitel 5, für eine theoretische Analyse *in silico* siehe Abschnitt 7.4.

Für die *in vitro*-Analyse, die von der Niemeyer-Gruppe durchgeführt wurde, wurden die Komplemente der ausgewählten Oligos ($cF1$ bis $cF10$ und $cT1$ bis $cT10$) auf einem Glasträger befestigt. An die Oligomere $F1$ bis $F10$ und $T1$ bis $T10$ wurden Streptavidin-Moleküle angehängt, um deren Einfluß auf die Hybridisierung bei der DDI zu berücksichtigen. Die Streptavidin-Moleküle waren außerdem mit dem Fluoreszenz-Farbstoff Cy5 markiert, um die Hybridisierungseffizienz durch Lichtintensität messen zu können. Anschließend wurden die Oligomer-Streptavidin-Konjugate auf das Microarray gegeben und so Hybridisierung ermöglicht. Schließlich wurden Fluoreszenzintensitäten gemessen. Für biochemische Details dieses Protokolls verweise ich auf [55].

7.5.3 Ergebnisse und Diskussion

Die Signalintensitäten für beide Bibliotheken sind in den Tabellen 7.11 und 7.12 zusammengefaßt und in Abbildung 7.9 veranschaulicht. Die Angaben sind in „arbitrary units“(a.u.) gemes-

Name	Sequenz	ΔG^{ss}_{25} (kcal/mol)	ΔG^{ss}_{37} (kcal/mol)	T_m (°C)	ΔG^{pm}_{37} (kcal/mol)
T1	CGAGACATCGTGCATATCGT	-1.84	-1.00	61.8	-24.95
T2	TCTGTACTGCTGACTCGAGT	0.00	0.00	60.5	-24.25
T3	CGAGTAGTCACACGATGAGA	-1.42	-0.40	60.3	-23.93
T4	AGATGATCAGCAGCGACACT	0.00	0.00	62.1	-25.15
T5	TGTGCTCGTCTCTGCATACT	-1.68	-0.70	61.6	-24.59
T6	AGACGAGTCGTACAGTACAG	-1.41	-0.60	58.8	-23.92
T7	ATGTACGTGAGATGCAGCAG	0.00	0.00	60.9	-24.48
T8	ATCACTACTCGCTCGTCACT	0.00	0.00	61.0	-24.93
T9	GCTGACATAGAGTGCGATAC	-0.35	0.00	59.6	-23.78
T10	ACATCGACACTACTACGCAC	-0.46	0.00	59.0	-24.53
T11	TATAGCACGAGTGCGCGTAT	-2.93	-2.20	62.8	-24.98
T12	GATCTACGATCATGAGAGCG	-2.76	-1.90	60.6	-23.86
T13	TCAGAGATACTCACGTCACG	-1.78	-1.10	59.5	-23.93
T14	GACAGAGCTATCAGCTACTG	-4.47	-2.60	60.0	-23.01

Tabelle 7.10: Oligomer-Sequenzen der T-Bibliothek, aus der Veröffentlichung von Tanaka et al. entnommen [162]. Es gilt die Beschreibung zu Tabelle 7.9.

sen und beziehen sich auf eine Normierung durch das DNA-Microarray-Lesegerät. Wie man an den Intensitäten der perfekten Duplex-Hybridisierungen ($F1$ mit $cF1$ usw.) erkennen kann, erzielte die F-Bibliothek mit einer durchschnittlichen Intensität von 29849 ± 8395 a.u. wesentlich stärkere Signale als die T-Bibliothek mit 4727 ± 3550 a.u. Relativ gesehen sind dies Abweichungen von ±28% ($F1 - F10$) bzw. ±75% ($T1 - T10$). Insbesondere zeigen die T-Sequenzen eine maximale Abweichung von 156%, bei den F-Sequenzen beträgt diese nur 52%. Es ist nicht klar, warum die T-Bibliothek viel schwächere Signale zeigt als die F-Bibliothek, obwohl die berechneten Schmelztemperaturen sehr ähnlich sind.

Um die Kreuzhybridisierung der Oligonukleotid-Streptavidin-Konjugate mit Sonden zu untersuchen, die nicht das jeweilige perfekte Komplement sind, wurde auf Basis der Werte in den Tabellen 7.11 und 7.12 ein Schwellenwert von 15 % der maximalen Signalintensität eines perfekten Duplex festgelegt, ab dem ein Signal als Kreuzhybridisierung angesehen wird. Dies entspricht 3019 a.u. bei der F-Bibliothek und 79 a.u. bei der T-Bibliothek. Signale unterhalb dieser Intensitätsschwelle gelten als Hintergrundrauschen, d. h. Signale durch Moleküle, die sich ohne Hybridisierung auf der Oberfläche des Microarrays angelagert haben. Mit diesem Schwellenwert zeigt die T-Bibliothek fünf Kreuzhybridisierungen, die F-Bibliothek dagegen keine.

Die starken Schwankungen der Hybridisierungseffizienz in der T-Bibliothek lassen sich nicht durch die Schmelztemperaturen erklären, da diese hier ähnlich nah beieinander liegen (±1.13 °C) wie in der F-Bibliothek (±0.36 °C). Schmelztemperaturen scheinen also nur ein schwaches Maß für die Hybridisierungseffizienz zu sein, was sich auch mit Erkenntnissen in der Literatur deckt [34]. Deutlichere Unterschiede zwischen den Bibliotheken sind zum Einen die wesentlich größere Basisstranglänge für n_b-Uniqueness in der T-Bibliothek (9 gegenüber 5 in der F-Bibliothek), zum Anderen die kleinere Energielücke (14.96 kcal/mol gegenüber 19.32 kcal/mol) zwischen stabilster unerwünschter und instabilster erwünschter Hybridisierung. Auch wenn diese erste *in vitro*-Prüfung nicht umfangreich genug ist, um wirklich sichere Schlüsse ziehen zu können, gibt das Ergebnis doch Grund zur Annahme, daß das für den DNA-Sequence-

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
cF10	56	110	68	70	110	111	73	91	60	**22655**
cF9	58	61	84	72	64	82	142	73	**20124**	469
cF8	57	180	115	132	293	591	249	**25847**	59	57
cF7	56	77	1722	1545	176	117	**20459**	66	59	59
cF6	56	171	75	59	101	**45428**	65	195	66	57
cF5	58	115	89	113	**33477**	80	198	65	59	62
cF4	58	89	110	**43646**	1703	72	2328	62	54	58
cF3	58	326	**30814**	76	62	82	128	396	58	56
cF2	71	**28430**	144	57	59	89	62	60	219	79
cF1	**27605**	203	203	356	243	277	161	690	63	59

Tabelle 7.11: Fluoreszenzsignal-Intensitäten des DDI-Experiments mit der F-Bibliothek des DNA-Sequence-Generators. Die Intensitäten der perfekten Duplexe sind fett gedruckt. Es zeigt sich keine Kreuzhybridisierung bei einem Schwellenwert von 15 % der niedrigsten Intensität eines perfekten Duplex (= 3019 a.u.).

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
cT10	42	47	44	1092	51	44	49	70	1326	**4174**
cT9	43	48	47	59	54	45	51	59	**5087**	73
cT8	45	47	45	57	52	51	52	**525**	56	58
cT7	50	51	100	64	52	52	**3360**	46	57	68
cT6	45	46	50	54	88	**4506**	58	40	48	56
cT5	41	44	45	57	**1522**	44	48	45	60	52
cT4	45	49	51	**12128**	54	48	59	50	63	79
cT3	52	49	**604**	62	55	49	53	60	59	67
cT2	75	**5881**	52	65	56	53	58	47	56	75
cT1	**9483**	105	56	64	57	53	62	45	53	66

Tabelle 7.12: Fluoreszenzsignal-Intensitäten des DDI-Experiments mit der T-Bibliothek aus [162]. Die Intensitäten der perfekten Duplexe sind fett gedruckt. Es zeigt sich eine stärkere Abweichung unter diesen Intensitäten als in der F-Bibliothek. Außerdem sind für einen Schwellenwert von 15 % der niedrigsten Intensität eines perfekten Duplex (= 79 a.u.) mehrere Kreuzhybridisierungen zu beobachten (unterstrichen)

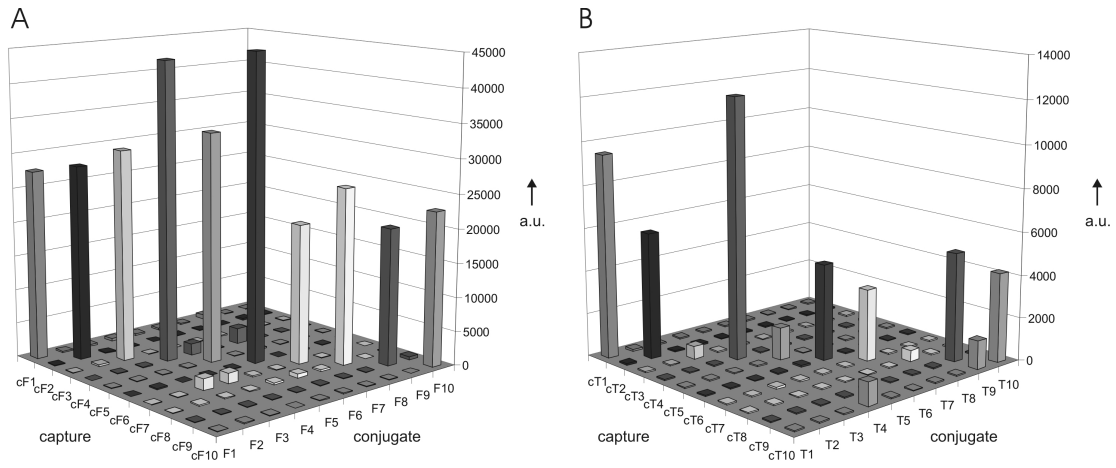


Abbildung 7.9: Hybridisierungssignale aus dem DDI-Experiment mit der F-Bibliothek des DNA-Sequence-Generators (A) und der T-Bibliothek aus [162] (B). Gezeigt werden die Fluoreszenzsignal-Intensitäten nach Hybridisierung der Oligomer-Streptavidin-Cy5-Konjugate. Die F-Bibliothek in (A) zeigt höhere und homogenere Signalintensitäten als die T-Bibliothek in (B).

Generator gewählte Konzept der Sequenzunähnlichkeit, die n_b-Uniqueness, einen realistischen Ansatz zur Vermeidung von Kreuzhybridisierungen darstellt.

Kapitel 8

Demonstrationen des DNA-Sequence-Compilers

Die folgenden vier Experimente dienen der beispielhaften Demonstration, welche Strukturen bzw. Bausteine für Strukturen man mit dem DNA-Sequence-Compiler entwerfen kann.

8.1 Ein DNA-Zufallszahlengenerator

8.1.1 Einleitung

Ein physikalischer Zufallszahlengenerator mit hoher Ausbeute soll konstruiert werden [53]. Dazu werden Nullen und Einsen in doppelsträngige DNA-Moleküle mit sticky Ends codiert, die dann *in vitro* zu Bitstrings beliebiger Länge hybridisieren.

8.1.2 Material und Methoden

Für die Sequenzgenerierung wurde die ältere Version des DNA-Sequence-Compilers verwendet, der nur lineare doppelsträngige DNA-Stäbe entwerfen kann und als Eingabe reguläre Grammatiken verwendet. Die Produktionsregeln der Grammatik für den Zufallszahlengenerator sind $P = \{S \to sA, A \to 0A, A \to 1A, A \to e\}$. Die Moleküle, die diese Regeln repräsentieren, sowie Beispiele für hybridisierte Supramoleküle werden in Abbildung 8.1 gezeigt[1]. Die Sequenzen s und e bilden dabei immer Anfang und Ende eines Supramoleküls. Sie dienen als Primer-Bindungsstellen für die PCR zum Auslesen der Binärzahlen. Vor der Sequenz s wird die Schnittstelle für das Restriktionsenzym HindIII (`AAGCTT`, wird im oberen wie unteren Strang zwischen den Adenin-Basen geschnitten) angefügt, nach der Sequenz e die Schnittstelle für das Enzym BamHI (`GGATCC`, wird zwischen den Guaninen geschnitten). Diese Schnittstellen dürfen in den zu erzeugenden Sequenzen nicht vorkommen und dienen in der Anwendung dazu, ein Supramolekül in die Plasmid-DNA eines Bakteriums zu klonieren, um es zu isolieren. Für biochemische Details zum Auslesen der Moleküle verweise ich auf [132].

Das sticky End A wurde als 10-mer entworfen, sollte für sich 4-unique sein, einen GC-Gehalt von 50 % haben und nicht mehr als zwei aufeinanderfolgende Guanin-Basen enthalten. Die Kernsequenzen s, e, 0 und 1 haben die Länge 20, eine Basisstranglänge von 6, einen GC-Gehalt von 50 %, eine Schmelztemperatur zwischen 50 und 51 °C, und enthalten ebenfalls

[1] Für das Startsymbol S wird keine Sequenz erzeugt

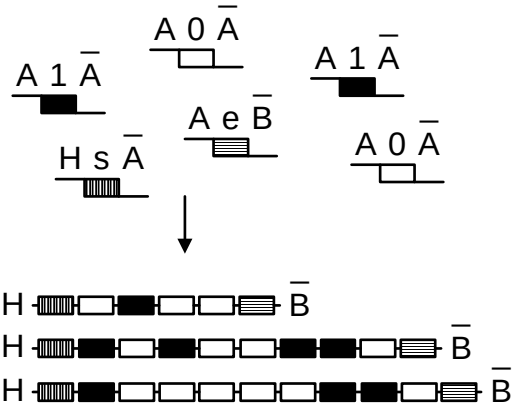


Abbildung 8.1: Self-assembly von Zufallszahlen (aus [132]). Oben: Die Regeln der regulären Grammatik werden durch doppelsträngige Moleküle mit sticky Ends repräsentiert, wobei der doppelsträngige Teil ein Terminal und jedes sticky End eine Variable darstellt. Unten: Durch Hybridisierung der sticky Ends bilden sich lange dsDNA-Moleküle, die zufällige Binärzahlen verschiedener Länge darstellen.

Name	Sequenz	T_m (°C)
A	`aaatcgtcgg`	8.4
s	`acggcctatactagctctac`	50.5
0	`tacagagtccggtttggtga`	50.8
1	`cagatcgagtgtatgaggag`	50.1
e	`ataacctcgttggaccacct`	50.4

Tabelle 8.1: DNA-Sequenzen für den Zufallszahlengenerator. Angegeben sind die Sequenzen für das sticky End und die Kernsequenzen mit ihrer Schmelztemperatur T_m.

keine Guanin-Folgen der Länge 3 oder länger. Die Schmelztemperaturen der sticky Ends wurden nach der Wallace-Regel, die der Kernsequenzen nach dem nearest-Neighbor-Modell mit Parametern der Sugimoto-Gruppe [159] für eine DNA-Konzentration von $2 \cdot 10^{-7}$ M und eine Na^+-Konzentration von 0.05 M berechnet.

8.1.3 Ergebnisse und Diskussion

Die generierten Sequenzen sind in Tabelle 8.1 aufgelistet und die daraus zu bildenden DNA-Stäbe in Abbildung 8.2 zu sehen. Da nur sehr wenige Sequenzen gebraucht wurden, konnten die Eigenschaften sehr stark eingeschränkt werden. Eine Erzeugung der Sequenzen mit den angegebenen Beschränkungen war bereits beim zweiten Versuch (mit verschiedenen Startwerten des Pseudozufallszahlengenerators) erfolgreich.

Tatsächlich werden die erzeugten Supramoleküle nicht beliebig lang werden, sondern mit wachsender Länge werden immer weniger Bitketten dieser Länge *in vitro* vorhanden sein [132].

```
S -> sA
        agcttacggcctatactagctctac
            atgccggatatgatcgagatgtttagcagcc

A -> 0A
        aaatcgtcggtacagagtccggtttggtga
                  atgtctcaggccaaaccactttagcagcc

A -> 1A
        aaatcgtcggcagatcgagtgtatgaggag
                  gtctagctcacatactcctctttagcagcc

A -> e
        aaatcgtcggataacctcgttggaccacctg
                  tattggagcaacctggtggacctag
```

Abbildung 8.2: DNA-Stabmoleküle für den Zufallszahlengenerator. Gezeigt werden die DNA-Stabmoleküle, die die Regeln der Eingabegrammatik repräsentieren. Am $5'$-Ende von s und $\overline{e}$ sind die Klonierstellen so angefügt, wie sie von den Restriktionsenzymen geschnitten würden.

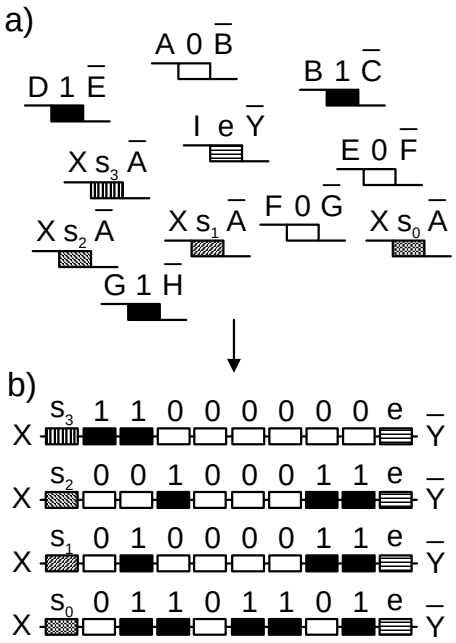


Abbildung 8.3: Self-assembly von 4-Byte-Binärzahlen (aus [132]). a) Die Regeln der regulären Grammatik werden durch doppelsträngige Moleküle mit sticky Ends repräsentiert, wobei der doppelsträngige Teil ein Terminal und jedes sticky End eine Variable darstellt. b) Durch Hybridisierung der sticky Ends bilden sich lange dsDNA-Moleküle, die zufällige Binärzahlen der Länge 8 Bit darstellen. Durch die verschiedenen Priming Sites s_0 bis s_3 lassen sich 4 Byte unterscheiden.

8.2 Eine 32-Bit-Datenstruktur

8.2.1 Einleitung

Um nicht Zufallszahlen beliebiger, sondern definierter Länge zu erzeugen, benötigt man mehrere verschiedene sticky Ends. Ein sticky End liegt dabei immer zwischen den beiden gleichen Bitpositionen. So lassen sich z. B. Moleküle für 8-Bit-Supramoleküle definieren. Variiert man dann noch die Priming-Site, kann man Datenstrukturen mit mehreren Bytes herstellen. Hier werden Sequenzen für eine 4-Byte-Datenstruktur erzeugt (Abb. 8.3) [53].

8.2.2 Material und Methoden

Auch hier wurde die ältere, auf regulären Grammatiken beruhende Version des DNA-Sequence-Compilers verwendet. Die Produktionsregeln für die 32-Bit-Datenstruktur sind

$$
\begin{aligned}
P = \{ & S \to s_0 A, \quad S \to s_1 A, \quad S \to s_2 A, \quad S \to s_3 A, \\
& A \to 0B, \quad A \to 1B, \quad B \to 0C, \quad B \to 1C, \\
& C \to 0D, \quad C \to 1D, \quad D \to 0E, \quad D \to 1E, \\
& E \to 0F, \quad E \to 1F, \quad F \to 0G, \quad F \to 1G, \\
& G \to 0H, \quad G \to 1H, \quad H \to 0I, \quad H \to 1I, \\
& I \to e\}
\end{aligned}
$$

Die sticky Ends $A, B, \ldots, I$ definieren die Reihenfolge, in der die Stabmoleküle zusammengefügt werden. Auch hier dienen die Sequenzen s und e als Priming-Sites für die PCR und sind mit Restriktionsschnittstellen für die Klonierung versehen (s. Abschnitt 8.1).

Wie bei den Zufallszahlen beliebiger Länge haben die sticky Ends die Länge 10, die Kernsequenzen die Länge 20. Die sticky Ends sind unter sich 4-unique, die Menge aller Sequenzen ist 6-unique. Für alle Sequenzen ist der GC-Gehalt auf 50 % beschränkt, für die Kernsequenzen darf außerdem die Schmelztemperatur nur zwischen 50 und 51 °C liegen. Damit die doppelsträngigen Kernsequenzen der Stabmoleküle nicht an ihren Enden ausfransen, mußte an 5′- und 3′-Ende jeweils ein `G-C`-Basenpaar eingesetzt werden. Da sich die Pfade der Kernsequenzen bei Konkatenation mit den sticky Ends stark verzweigen müssen, wurde die kontrollierte Mehrfachverwendung von Basissträngen in den gesamten Übergangsbereichen erlaubt (s. Abschnitt 6.3). Die Schmelztemperaturen der sticky Ends wurden nach der Wallace-Regel, die der Kernsequenzen nach dem nearest-Neighbor-Modell mit Parametern der Sugimoto-Gruppe [159] für eine DNA-Konzentration von $2 \cdot 10^{-7}$ M und eine Na^+-Konzentration von 0.05 M berechnet.

8.2.3 Ergebnisse und Diskussion

Die gefundenen Sequenzen werden in Tabelle 8.2, die Stabmoleküle in Abbildung 8.4 gezeigt. Die Beschränkungen der Sequenzeigenschaften sind sehr restriktiv, insbesondere die Schmelztemperaturen liegen in einem sehr schmalen Bereich. Daher wurden 5 Versuche mit verschiedenen Initialisierungen des Pseudozufallszahlengenerators benötigt, um diese Sequenzen erfolgreich generieren zu können. Relativ großzügig ist der Bereich für die erlaubten Mehrfachverwendungen gewählt, mit 5 Schritten von der eigentlichen Verzweigung aus werden diese Verletzungen der n_b-Uniqueness im gesamten Übergangsbereich zwischen zwei konkatenierten Sequenzen toleriert. Eine Verkürzung dieses Bereichs war nur möglich, wenn andere Restriktionen gelockert wurden, z. B. die Schmelztemperaturen oder die Einschränkung der Basenpaaren an den Sequenzenden zur Verhinderung des Ausfransens. Prinzipiell wurde aber ein erfolgreicher Sequenzentwurf auch bei starker Verzweigung der Pfade demonstriert.

Die Verwendung von k verschieden Priming-Sites $s_0, \ldots, s_{k-1}$ erlaubt die Herstellung einer k-Byte Datenstruktur, ohne eine k-fache Menge an sticky Ends generieren zu müssen. Zum Auslesen des i-ten Byte wird für die PCR die Primer s_i eingesetzt, jedes Byte ist also einzeln adressierbar.

8.3 Bausteine für ein DNA-Band

8.3.1 Einleitung

Um auch die Erzeugung von komplexeren Strukturen zu demonstrieren, sollen vierarmige Junctions als Bausteine für ein DNA-Band ähnlich wie in [182] generiert werden.

8.3.2 Material und Methoden

Zur Generierung wurde die neuere Variante des DNA-Sequence-Compilers verwendet. Für das Self-Assembly eines DNA-Bands werden zwei Typen von vierarmigen Junctions benötigt (Abb. 8.5). Alle Arme sind 20 Basenpaare lang (ohne sticky Ends) und sind mit 8 Basen langen sticky Ends versehen. Die Arme, die nach dem Self-Assembly außen liegen werden, besitzen keine sticky Ends. Die Schmelztemperatur der Armsequenzen ist auf 55-57 °C eingeschränkt, der

```
S -> s0A
agcttcaacacatggagttacacgc
    agttgtgtacctcaatgtgcggcctttgtag

S -> s2A
agcttgctcctagaagtctacaagc
    acgaggatcttcagatgttcggcctttgtag

A -> 0B
cggaaacatcggatttggcaacaacctgag
          cctaaaccgttgttggactcgaaaatcggg

B -> 0C
cttttagcccggatttggcaacaacctgag
          cctaaaccgttgttggactccctctaatgg

C -> 0D
ggagattaccggatttggcaacaacctgag
          cctaaaccgttgttggactcggcgtttatc

D -> 0E
ccgcaaatagggatttggcaacaacctgag
          cctaaaccgttgttggactcgtctcgtatg

E -> 0F
cagagcatacggatttggcaacaacctgag
          cctaaaccgttgttggactcgcatcttgac

F -> 0G
cgtagaactgggatttggcaacaacctgag
          cctaaaccgttgttggactcctgccaatag

G -> 0H
gacggttatcggatttggcaacaacctgag
          cctaaaccgttgttggactcgacttcactg

H -> 0I
ctgaagtgacggatttggcaacaacctgag
          cctaaaccgttgttggactccagaacacag

I -> e
gtcttgtgtccttgtttaatacaggggcgcg
          gaacaaattatgtccccgcgcctag
```

```
S -> s1A
agcttgaaaaaattggactcggggc
    actttttttaacctgagccccggcctttgtag

S -> s3A
agcttcttctgccatacaactaggc
    agaagacggtatgttgatccggcctttgtag

A -> 1B
cggaaacatccaaccaggattaagccatgc
          gttggtcctaattcggtacggaaaatcggg

B -> 1C
cttttagccccaaccaggattaagccatgc
          gttggtcctaattcggtacgcctctaatgg

C -> 1D
ggagattacccaaccaggattaagccatgc
          gttggtcctaattcggtacgggcgtttatc

D -> 1E
ccgcaaatagcaaccaggattaagccatgc
          gttggtcctaattcggtacggtctcgtatg

E -> 1F
cagagcataccaaccaggattaagccatgc
          gttggtcctaattcggtacggcatcttgac

F -> 1G
cgtagaactgcaaccaggattaagccatgc
          gttggtcctaattcggtacgctgccaatag

G -> 1H
gacggttatccaaccaggattaagccatgc
          gttggtcctaattcggtacggacttcactg

H -> 1I
ctgaagtgaccaaccaggattaagccatgc
          gttggtcctaattcggtacgcagaacacag
```

Abbildung 8.4: DNA-Stabmoleküle für die 32-Bit-Datenstruktur. Am $5'$-Ende von s_0, s_1, s_2, s_3 und $\overline{e}$ sind die Klonierstellen so angefügt, wie sie von den Restriktionsenzymen geschnitten würden.

Name	Sequenz	T_m (°C)
A	cggaaacatc	8.4
B	cttttagccc	8.4
C	ggagattacc	8.4
D	ccgcaaatag	8.4
E	cagagcatac	8.4
F	cgtagaactg	8.4
G	gacggttatc	8.4
H	ctgaagtgac	8.4
I	gtcttgtgtc	8.4
s0	caacacatggagttacacgc	50.2
s1	gaaaaaattggactcggggc	50.2
s2	gctcctagaagtctacaagc	50.1
s3	cttctgccatacaactaggc	50.3
0	ggatttggcaacaacctgag	50.0
1	caaccaggattaagccatgc	50.6
e	cttgtttaatacaggggcgc	50.9

Tabelle 8.2: DNA-Sequenzen für die 32-Bit-Datenstruktur. Angegeben sind die Sequenzen für die sticky Ends und die Kernsequenzen mit ihrer Schmelztemperatur T_m.

GC-Gehalt der sticky Ends auf 50 %. Die Menge aller Sequenzen soll 5-unique sein. Die T_m-Berechnungen verwenden das nearest-Neighbor-Modell mit den Parametern von SantaLucia et al. [145] und gelten für eine DNA-Konzentration von $2 \cdot 10^{-7}$ M und eine Na^+-Konzentration von 0.05 M. Die DeLaNA-Eingabedatei ist in Abbildung 8.6 gezeigt.

8.3.3 Ergebnisse und Diskussion

Abbildung 8.7 zeigt die DeLaNA-Beschreibung der gefundenen Sequenzen, Abbildung 8.8 eine schematische Darstellung der so erzeugten vierarmigen Junctions. Dieses recht kleine Beispiel ist tatsächlich keine wirkliche Herausforderung an den DNA-Sequence-Compiler (bereits der zweite Generierungsversuch war erfolgreich), die Ausgaben sind aber übersichtlich genug, um hier als Beispiel dargestellt werden zu können. Ein größeres Beispiel findet sich im nächsten Abschnitt.

8.4 Bausteine für einen DNA-Würfel

8.4.1 Einleitung

In diesem etwas umfangreicheren Beispiel für den Entwurf komplexer Strukturen sollen dreiarmige Junctions generiert werden, die sich per Self-Assembly zu einem würfelförmigen Gerüst verbinden können.

8.4.2 Material und Methoden

Jede zu generierende dreiarmige Junction repräsentiert eine der acht Ecken des Würfels. Die Arme dienen somit als Würfelkanten, genauer als halbe Kanten, die in der Mitte per sticky

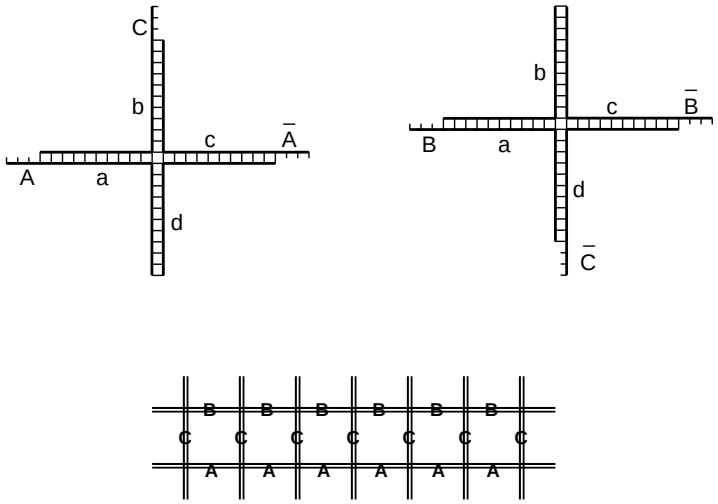


Abbildung 8.5: Junctions für ein zweireihiges DNA-Band. Oben sind die beiden Typen von vierarmigen Junctions dargestellt. Das linke Molekül wird durch die sticky Ends A und $\overline{A}$ beim Self-Assembly zur unteren Reihe verbunden, das rechte durch B und $\overline{B}$ zur oberen Reihe. Die beiden Reihen werden durch die sticky Ends C und $\overline{C}$ aneinandergefügt. Nach außen hin gibt es keine sticky Ends. Die Armsequenzen a, b, c und d werden in beiden Junctions verwendet, um Sequenzen zu sparen. Ein getrenntes Assembly beider Junctiontypen vor dem Zusammenführen zur Banderzeugung sollte Hybridisierungen zwischen Armsequenzen verschiedener Junctiontypen verhindern. Unten ist ein Abschnitt eines bereits zusammengefügten Bands mit Angabe der sticky Ends skizziert.

```
SEQUENCETYPE arm {
          length = 20;
          GC_ratio = [0;1];
          Tm = [55;57]; }

SEQUENCETYPE se {
          length = 8;
          GC_ratio = 0.5; }

arm a, b, c, d;

se A, B, C;

se blunt {
  length = 0; }

MACRO_4WAYJUNCTION j1 {
  arm1 = a;
  arm2 = b;
  arm3 = c;
  arm4 = d;
  sticky_end1 = A;
  sticky_end2 = C;
  sticky_end3 = A;
  orientation3 = 1;
  sticky_end4 = blunt;
  }

MACRO_4WAYJUNCTION j2 {
  arm1 = a;
  arm2 = b;
  arm3 = c;
  arm4 = d;
  sticky_end1 = B;
  sticky_end2 = blunt;
  sticky_end3 = B;
  orientation3 = 1;
  sticky_end4 = C;
  orientation4 = 1;
  }


POOL p {
          N_uniqueness = 5;
          Violation_tolerance = 0;
          Sample_conc = 2e-7;
          Na_conc = 0.05;
          Tm_method = NNSantaLucia;
          }
```

Abbildung 8.6: DeLaNA-Eingabedatei für das DNA-Band. Das sticky End *blunt* mit der Länge Null dient nur der Vollständigkeit der Beschreibung. Die `orientation`-Einträge geben an, ob ein sticky End aus der eigentlichen Sequenz oder deren Komplement bestehen soll. Eine 1 führt zur Komplementbildung (vergleiche Abb. 8.5), fehlende Angaben werden als 0 gelesen.

```
// File name: ribbon_out.dln
// Created by CANADA v0.1.1

SEQUENCETYPE arm {
               NA_type = DNA;
               length = 20;
               GC_ratio = [0;1];
               Tm = [55;57];
               seq_mask = "";
}


SEQUENCETYPE se {
               NA_type = DNA;
               length = 8;
               GC_ratio = 0.5;
               Tm = [0;100];
               seq_mask = "";
}


SEQUENCE a {
               NA_type = DNA;
               length = 20;
               GC_ratio = 0.55;
               Tm = 55.5875;
               seq_mask = "cagatagcgatggctttgcc";
}


SEQUENCE b {
               NA_type = DNA;
               length = 20;
               GC_ratio = 0.55;
               Tm = 55.9435;
               seq_mask = "ttaccggcacctggatacac";
}


SEQUENCE c {
               NA_type = DNA;
               length = 20;
               GC_ratio = 0.55;
               Tm = 56.5951;
               seq_mask = "gtggtgagcggaggctaaaa";
}


SEQuENCE d {
               NA_type = DNA;
               length = 20;
               GC_ratio = 0.55;
               Tm = 56.747;
               seq_mask = "aatacggtccttccgggtag";
}


SEQUENCE A {
               NA_type = DNA;
               length = 8;
               GC_ratio = 0.5;
               Tm = 4.88809;
               seq_mask = "tatgcgag";
}


SEQUENCE B {
               NA_type = DNA;
               length = 8;
               GC_ratio = 0.5;
               Tm = 10.1763;
               seq_mask = "cgtttgtc";
}


SEQUENCE C {
               NA_type = DNA;
               length = 8;
               GC_ratio = 0.5;
               Tm = 7.43886;
               seq_mask = "tgccaatc";
}


SEQUENCE blunt {
               NA_type = DNA;
               length = 0;
               GC_ratio = 0;
               Tm = 0;
               seq_mask = "";
}


MACRO_4WAYJUNCTION j1 {
               Sticky_end1 = A;
               Orientation1 = 0;
               Sticky_end2 = C;
               Orientation2 = 0;
               Sticky_end3 = A;
               Orientation3 = 1;
               Sticky_end4 = blunt;
               Orientation4 = 0;
               Arm1 = a;
               Arm2 = b;
               Arm3 = c;
               Arm4 = d;
}


MACRO_4WAYJUNCTION j2 {
               Sticky_end1 = B;
               Orientation1 = 0;
               Sticky_end2 = blunt;
               Orientation2 = 0;
               Sticky_end3 = B;
               Orientation3 = 1;
               Sticky_end4 = C;
               Orientation4 = 1;
               Arm1 = a;
               Arm2 = b;
               Arm3 = c;
               Arm4 = d;
}


POOL p {
               sequences = ;
               n_uniqueness = 5;
               Hamming = 0;
               H_distance = 0;
               sample_conc = 2e-007;
               Na_conc = 0.05;
               formamide_conc = 0;
}
```

Abbildung 8.7: DeLaNA-Ausgabedatei des DNA-Bands. Es sind sämtliche Sequenzen mit ihren Eigenschaften einzeln aufgeführt.

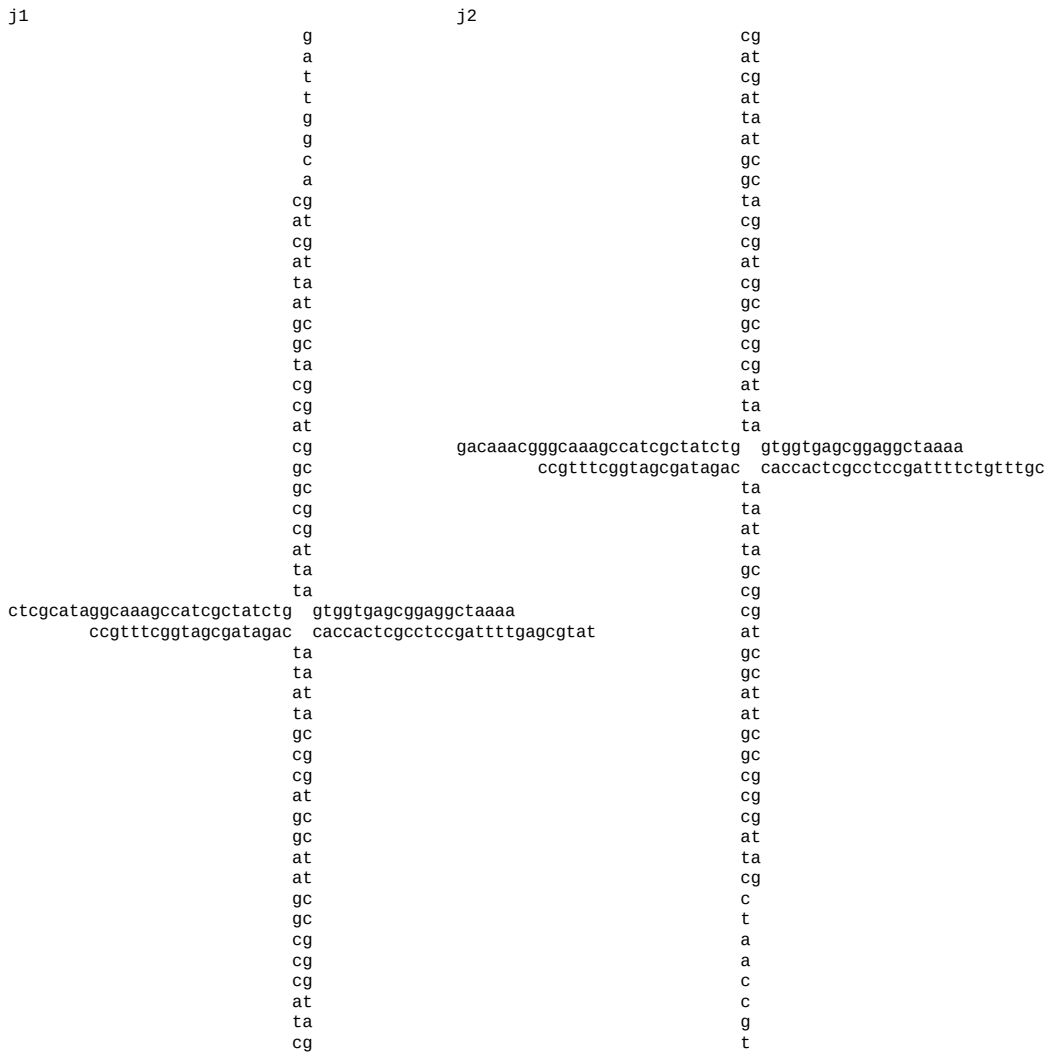
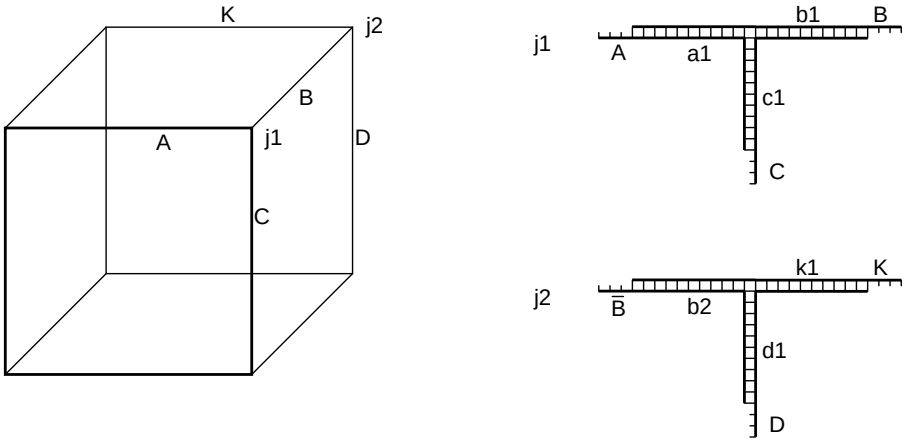


Abbildung 8.8: Generierte vierarmige Junctions für das DNA-Band. Hier sind die Sequenzen aus Abbildung 8.7 gemäß der Definition der Junction-Makros angeordnet.




```
MACRO_3WAYJUNCTION j1 {
  arm1 = a1;
  arm2 = b1;
  arm3 = c1;
  sticky_end1 = A;
  sticky_end2 = B;
  sticky_end3 = C;
  }
```

```
MACRO_3WAYJUNCTION j2 {
  arm1 = b2;
  arm2 = k1;
  arm3 = d1;
  sticky_end1 = B;
  orientation1 = 1;
  sticky_end2 = K;
  sticky_end3 = D;
  }
```

Abbildung 8.9: Skizze des DNA-Würfels und Beispiele für Junctions. Der Würfel besteht aus acht dreiarmigen Junctions $j1$ bis $j8$, die die Ecken repräsentieren. Jeder Arm einer Junction stellt eine halbe Kante dar, die sticky Ends dienen dazu, diese Kantenhälften zu verbinden. Im Würfel links oben sind zwei Ecken und die dazu inzidenten Kanten mit Identifikatoren markiert. Rechts oben sind die beiden entsprechenden Junctions schematisch dargestellt. Die Kante B des Würfels teilt sich auf in die Armsequenzen $b1$ von $j1$ und $b2$ von $j2$ sowie das sticky End B. Unten sind die DeLaNA-Beschreibungen der beiden Junctions angegeben. Der Eintrag `orientation1 = 1` gibt an, daß dieses sticky End das Komplement der Sequenz B ist.

End mit der anderen Hälfte der Kante (dem entsprechenden Arm der benachbarten Junction) hybridisieren sollen (s. Abb. 8.9).

Zur Generierung der Sequenzen wird auch hier die neuere Version des DNA-Sequence-Compilers verwendet. Die Armsequenzen sind 11 Basenpaare lang, die sticky Ends haben die Länge 8, so daß nach Hybridisierung jede Kante des Würfels 30 Basenpaare lang ist, was genau drei Umdrehungen der Doppelhelix entspricht. Die Schmelztemperatur der Armsequenzen ist auf 25-30 °C eingeschränkt, der GC-Gehalt der sticky Ends auf 50 %. Die Menge aller Sequenzen soll 6-unique sein. Die T_m-Berechnungen verwenden das nearest-Neighbor-Modell mit den Parametern von SantaLucia et al. [145] und gelten für eine DNA-Konzentration von $2{\cdot}10^{-7}$ M und eine Na^+-Konzentration von 0.05 M. Auf eine vollständige Angabe der DeLaNA-Eingabedatei wird hier aus Platzgründen verzichtet, sie ist im Anhang zu finden.

8.4.3 Ergebnisse und Diskussion

Auch die DeLaNA-Ausgabedatei und die erzeugten Junctions werden nicht hier, sondern im Anhang gezeigt. Tabelle 8.3 listet die generierten Sequenzen auf.

linker Arm		sticky End		rechter Arm	
ID	Sequenz	ID	Sequenz	ID	Sequenz
a1	cgtcgatgcta	A	ctcccttt	a2	tgtagccaaac
b1	tactatcgcgt	B	tcgacttc	b2	tgacgaaatgc
c1	gaccgtcctat	C	attctccg	c2	caagacaagcc
d1	cgagcaggtta	D	gaaatcgc	d2	caaggggcata
e1	ttacctcagcc	E	ccaccatt	e2	gcctccgaata
f1	agagatgagcg	F	cacccata	f2	gaactccacga
g1	ttaaactgccg	G	ccgtatgt	g2	gggattaaggc
h1	ctaattgcgcc	H	cgctaatg	h2	tgattgcttgc
i1	ccgttctctag	I	gagtctct	i2	caccggctata
j1	attccgagtga	J	gctatcac	j2	gcgcgtaatag
k1	cgaactgggta	K	gtattccc	k2	gcgtagagttg
l1	ggacatgcttc	L	tcagagca	l2	aggagatacgt

Tabelle 8.3: Generierte Sequenzen für den DNA-Würfel. In jeder Zeile sind die drei Sequenzen angegeben, die zusammen eine der zwölf Kanten des Würfels bilden. Links und rechts befinden sich die Arme der beiden Junctions, die durch die Kante verbunden werden, in der Mitte das sticky End.

Diese 36 Sequenzen zu generieren war bereits deutlich schwieriger als beim im letzten Abschnitt beschriebenen Beispiel. Von 50 Versuchen, die Sequenzen zu erzeugen, waren nur 5 erfolgreich. Die 45 erfolglos beendeten Versuche scheiterten alle bereits an der Suche nach Startknoten für die Armsequenzen. Obwohl jeder Arm nur zwei Nachbarsequenzen hat, mit denen er in der Mitte der Junction zusammenkommt, scheint die Beachtung der n_b-Uniqueness in diesen verzweigenden Sequenzübergängen eine starke Einschränkung darzustellen.

Kapitel 9

Zusammenfassung und Ausblick

Der Entwurf von Nukleinsäuresequenzen ist sowohl für das DNA-Computing als auch für die Verwendung von DNA-Self-Assembly in der Nanotechnolgie wichtig, aber auch in alltäglichen Labortechniken wie der Polymerasekettenreaktion oder DNA-Microarrays. Ziel des Entwurfs ist dabei die Programmierung des Self-Assembly, d. h. die Sequenzen sollen derart gewählt werden, daß in vitro die gewünschten Hybridisierungen stattfinden, aber auch nur diese. Neben verschiedenen anwendungsabhängigen Anforderungen an die Sequenzen und ihre Eigenschaften ist das wichtigste Designziel daher eine spezifische Hybridisierung, bei der die gewünschten Hybridisierungen stabil sind und mit hoher Wahrscheinlichkeit und Ausbeute geschehen, während alle unerwünschten Formen der Hybridisierung unwahrscheinlich und instabil sind. Aufgrund der großen Anzahl möglicher Sequenzkandidaten sowie der zum Teil aufwendigen Berechnung der Eigenschaften von Sequenzen und Sequenzmengen ist die Unterstützung durch den Computer unerläßlich.

Eine Entwurfsstrategie besteht im Wesentlichen aus zwei Teilen, einem Modell für die Hybridisierungswahrscheinlichkeit von DNA-Molekülen in Abhängigkeit von ihrer Basensequenz, und einem Algorithmus, der Sequenzen konstruiert oder auswählt, die nach diesem Modell spezifisch hybridisieren. Eine gerade für Informatiker naheliegende Modellierung der Hybridisierungsneigung ist die Sequenzähnlichkeit, die man mit verschiedenen Distanzmaßen für Zeichenketten messen kann. Wie in dieser Arbeit gezeigt wurde, korrelieren die verschiedenen Distanzmaße eher schlecht mit einem thermodynamisch detaillierten, theoretisch begründeten, und daher als realistisch angenommenen Modell, dem nearest-Neighbor-Modell. Da Distanzmaße paarweise berechnet werden, ist die Bewertung einer ganzen Sequenzmenge auch sehr rechenintensiv.

Die in dieser Arbeit vorgestellte Entwurfsstrategie verwendet als Modell der Hybridisierungsneigung eine Sequenzmengeneigenschaft, die n_b-Uniqueness. Diese beruht auf der Einmaligkeit von Subsequenzen der Länge n_b, wobei auch Komplementärsequenzen vorkommender Subsequenzen ausgeschlossen werden. Die Methode zur Sequenzgenerierung beruht auf einem Graphen der Subsequenzen. Ein greedy Algorithmus sucht knotendisjunkte Pfade durch diesen Graphen, die einer Menge von Sequenzen entsprechen, die die n_b-Uniqueness erfüllen.

Dieser Algorithmus erzwingt bereits bei der Konstruktion der Sequenzen die wichtigste Eigenschaft, eben die n_b-Uniqueness, anstatt beliebige Sequenzen zu erzeugen und ggf. wieder zu verwerfen. Weitere Einschränkungen von Sequenzeigenschaften wie Schmelztemperatur, fixe oder verboten Subsequenzen, oder Homologie lassen sich einfach zusätzlich implementieren. Will man nicht nur eine Menge einzelner Sequenzen generieren, sondern sollen die DNA-Moleküle in der Anwendung Konkatenationen und komplexere Strukturen bilden, so läßt sich

der Algorithmus auch für diese Problemstellungen erweitern. Die größte Schwäche der Algorithmus liegt in seiner worst-case-Rechenzeit, die der einer vollständigen Enumeration entspricht.

Basierend auf diesem Algorithmus ist Software für das DNA-Sequenzdesign entwickelt worden. Für die Generierung von Mengen einzelner Sequenzen ist dies der DNA-Sequence-Generator, während der DNA-Sequence-Compiler dem Entwurf von Sequenzen für verschiedene Strukturen dient.

Untersuchungen haben gezeigt, daß die Ausbeute des Algorithmus dem theoretischen Maximum recht nahe kommt. Ebenfalls gezeigt wurde, daß die Einschränkung der Sequenzähnlichkeit durch die n_b-Uniqueness tatsächlich Auswirkungen auf die Spezifität der Hybridisierung hat, und daß sie auch Einfluß auf die Neigung der DNA-Moleküle zur Bildung einzelsträngiger Sekundärstrukturen hat. Hierbei ist es allerdings wichtig, die Länge einmaliger Subsequenzen n_b so klein wie möglich zu wählen, insbesondere deutlich kleiner als die Länge der Gesamtsequenzen. Ein Vergleich mit veröffentlichten Sequenzbibliotheken demonstrierte, daß der in dieser Arbeit vorgestellte Algorithmus anderen Sequenzdesignverfahren überlegen bzw. zumindest ebenbürtig ist. Schließlich konnte die Güte von Sequenzen, die mit der auf diesem Algorithmus basierenden Software generiert wurden, auch in vitro nachgewiesen werden.

Allerdings bleibt noch Raum für weitere Untersuchungen und Entwicklungen. Insbesondere der greedy Algorithmus zur Pfadsuche kann sicherlich verbessert werden. Die Laufzeit könnte verbessert werden, wenn eine Heuristik früh genug aussichtslosen Sequenzgenerierungsversuche erkennen und abbrechen würde. Hierzu wären sowohl experimentelle Untersuchungen zu Laufzeit und Erfolgswahrscheinlichkeit nötig als auch die Entwicklung einer solchen Heuristik, die auf den Ergebnissen der Experimente beruht. Denkbar wäre auch ein umfangreiche Untersuchung der Eignung von anderen, graphentheoretisch fundierten Verfahrens zur Suche knotendisjunkter Pfade für das Sequenzdesign. Z. B. wäre es untersuchenswert, inwieweit sich Eulerpfade auch unter Ausschluß der Komplemente von Basissträngen verwenden lassen, oder wie die tatsächliche Güte von Approximationsalgorithmen zum *maximum disjoint paths problem* bei Anwendung auf den Basisstranggraph ist. Eine andere interessante Erweiterung des Algorithmus wäre das Einbinden zusätzlicher Information, nämlich der Unähnlichkeit der Basisstränge untereinander, so daß wenn möglich nicht nur verschiedene, sondern auch möglichst unähnliche Basisstränge in den Sequenzen vorkommen. Ebenfalls vorstellbar ist die Berücksichtigung von thermodynamischen Daten bereits während der Konstruktion einer Sequenz, ein solcher Algorithmus stellt z. Z. noch eine große Herausforderung dar.

Der DNA-Sequence-Compiler kann bisher nur einige wenige einfache Strukturen entwerfen. Die Erweiterung dieses Programms auf komplexere Strukturen ist gerade für die Verwendung in der Nanotechnologie äußerst wichtig. Double und Triple-Crossover-Kacheln z. B. bilden durch Self-Assembly flächige Strukturen und bieten eine aussichtsreiche Methode zum Nanogerüstbau. Ultimatives Ziel ist hier die Fähigkeit des Programms, beliebige Strukturen zu entwerfen. Erforderlich wäre dafür, daß das Programm solche Strukturen in Einzelsequenzen zerlegt und selbständig eine sinnvolle Reihenfolge wählt, in der diese zu generieren sind.

Auch wenn das nearest-Neighbor-Modell aus den genannten Gründen als das z. Z. realistischste gilt, so ist es doch nur ein Modell. Weitere Untersuchungen *in vitro* sind dringend notwendig, vor allem auch mit größeren Sequenzmengen, die unter verschiedenen Gesichtspunkten und mit verschiedenen Hybridisierungsmodellen und Algorithmen entworfen wurden. Leider sind solche Untersuchungen sehr kostenaufwendig. Um Experimente wirklich im größeren Maßstab durchführen zu können, müßte die Synthetisierung von DNA-Molekülen deutlich preiswerter werden.

Künstliche Nukleinsäuren wie PNA und LNA werden zunehmend in molekularbiologischen Verfahren verwendet und sind auch für die Nanotechnologie attraktiv. Für diese gibt es aber

noch keine systematischen Untersuchungen zur Abhängigkeit der Hybridisierungsstabilität von der Basensequenz, so daß das nearest-Neighbor-Modell für diese Moleküle bisher nicht angewendet werden kann.

Über den Autor

Von 1992 bis 1995 Studium der Informatik mit Nebenfach Physik an der Universität Kaiserslautern, von 1995 bis 1999 Fortsetzung des Studiums an der Universität Dortmund, Abschluß (Diplom) im Oktober 1999. Von 2000 bis 2004 wissenschaftlicher Mitarbeiter am Lehrstuhl für Systemanalyse des Fachbereichs Informatik in der Arbeitsgruppe von Prof. Wolfgang Banzhaf. Seit 2005 wissenschaftlicher Mitarbeiter am Lehrstuhl für Biologisch-Chemische Mikrostrukturtechnik des Fachbereichs Chemie in der Arbeitsgruppe von Prof. Christof M. Niemeyer. Forschungsschwerpunkte sind naturanaloge Berechnungsmethoden und Computational Intelligence, Bioinformatics im Sinne von Genomics und Proteomics, sowie DNA-Sequenz-Design für DNA-Computing und Nanotechnologie.

Publikationen

Udo Feldkamp (2000): Ein DNA-Sequenz-Compiler, *Technical Report of the Systems Analysis Research Group*, University of Dortmund, Department of Computer Science, SYS–2/00.

Udo Feldkamp, Wolfgang Banzhaf und Hilmar Rauhe (2000): A DNA sequence compiler, in Anne Condon and Grzegorz Rozenberg (Eds.), *Preproceedings of the 6th International Workshop on DNA-Based Computers, DNA 2000*, Leiden, The Netherlands, June 2000, 253 (Poster).

Hilmar Rauhe, Gaby Vopper, Udo Feldkamp, Wolfgang Banzhaf und Jonathan C. Howard (2000): Digital DNA molecules, in Anne Condon and Grzegorz Rozenberg (Eds.), *Preproceedings of the 6th International Workshop on DNA-Based Computers, DNA 2000*, Leiden, The Netherlands, June 2000, 271 (Poster).

Udo Feldkamp, Sam Saghafi, Wolfgang Banzhaf und Hilmar Rauhe (2002): DNASequenceGenerator: A Program for the construction of DNA sequences, in Natasha Jonoska and Nadrian C. Seeman (Eds.), *DNA Computing, 7th International Workshop on DNA-Based Computers, DNA 2001*, Tampa, USA, 10-13 June 2001, Springer Verlag, 23-32.

Michael Rosskopf, Udo Feldkamp und Wolfgang Banzhaf (2003): Classification of leukemia classes by GP-based DNA-chip analysis, in A. Barry (Ed.), *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2003) – Workshop Program*, Chicago IL, 2003, 81-82.

Udo Feldkamp, Hilmar Rauhe und Wolfgang Banzhaf (2003): Software Tools for DNA Sequence Design, *Genetic Programming and Evolvable Machines*, 4(2): 153-171.

Udo Feldkamp, Ron Wacker, Hendrik Schroeder, Wolfgang Banzhaf und Christof M. Niemeyer (2004): Microarray-Based in vitro Evaluation of DNA Oligomer Libraries Designed in silico, *ChemPhysChem*, 5(3): 367-372.

Literaturverzeichnis

[1] J. Ackermann and F.-U. Gast. Word design for biomolecular information processing. *Zeitschrift für Naturforschung*, 58a:157–161, 2003.

[2] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, November 1994.

[3] L. M. Adleman. On the potential of molecular computing. *Science*, 268:483–484, 1995.

[4] L. M. Adleman. On constructing a molecular computer. In E. B. Baum and R. J. Lipton, editors, *DNA Based Computers*, volume 27 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science.* AMS, 1996.

[5] P. Alberti and J.-L. Mergny. DNA duplex-quadruplex exchange as the basis for a nano-molecular machine. *Proceedings of the National Academy of Sciences*, 100(4):1569–1573, 2003.

[6] H. T. Allawi, N. Peyret, P. A. Seneviratne, and J. SantaLucia, Jr. DNA mismatch thermodynamics and structure. *Abstracts of Papers of the American Chemical Society*, 213:270, April 1997.

[7] H. T. Allawi and J. SantaLucia, Jr. Thermodynamics and NMR of internal G·T mismatches in DNA. *Biochemistry*, 36(34):10581–10594, 1997.

[8] H. T. Allawi and J. SantaLucia, Jr. Nearest neighbor thermodynamic parameters for internal G·A mismatches in DNA. *Biochemistry*, 37(8):2170–2179, 1998.

[9] H. T. Allawi and J. SantaLucia, Jr. Nearest-neighbor thermodynamics of internal A·C mismatches in DNA: Sequence dependence and pH effects. *Biochemistry*, 37(26):9435–9444, 1998.

[10] H. T. Allawi and J. SantaLucia, Jr. Thermodynamics of internal C·T mismatches in DNA. *Nucleic Acids Research*, 26(11):2694–2701, 1998.

[11] American Mathematical Society. *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University, June 10-12, 1996*, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science., 1996.

[12] M. Amos and P. E. Dunne. DNA simulation of boolean circuits. Technical Report CTAG-97009, Department of Computer Science, University of Liverpool, UK, December 1997.

[13] M. Andronescu, R. Aguirre-Hernández, A. Condon, and H. H. Hoos. RNAsoft: a suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Research*, 31(13):3416–3422, 2003.

[14] M. Andronescu, A. P. Fejes, F. Hutter, H. H. Hoos, and A. Condon. A new algortihm for RNA secondary structure design. *Journal of Molecular Biology*, 336:607–624, 2003.

[15] M. Arita. Writing information into DNA. In Jonoska et al. [82], pages 23–35.

[16] M. Arita and S. Kobayashi. DNA sequence design using templates. *New Generation Computing*, 20:263–277, 2002.

[17] M. Arita, A. Nishikawa, M. Hagiya, K. Komiya, H. Gouzu, and K. Sakamoto. Improving sequence design for DNA computing. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 875–882. Morgan Kaufmann, 2000.

[18] T. Bäck. *Evolutionary Algorithms in Theory and Practice — Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.

[19] T. Bäck, J. N. Kok, and G. Rozenberg. Cross-fertilization between evolutionary computation and DNA-based computing. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, pages 980–987, 1999.

[20] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[21] W. Banzhaf. Self-organizing algorithms derived from RNA interactions. In W. Banzhaf and F. H. Eeckman, editors, *Evolution and Biocomputation*, volume 899 of *LNCS*, pages 69–102. Springer, Berlin, 1995.

[22] W. Banzhaf. *Encyclopedia of Physical Science and Technology*, volume 14, chapter Self-organizing Systems, pages 589–598. Academic Press, New York, 2002.

[23] W. Banzhaf. Artificial chemistries - toward constructive dynamical systems. *Solid State Phenomena*, 97/98:43–50, 2004.

[24] W. Banzhaf, P. Dittrich, and H. Rauhe. Emergent computation by catalytic reactions. *Nanotechnology*, 7:307–314, 1996.

[25] E. B. Baum. DNA sequences useful for computation. Available under http://www.neci.nj.nec.com/homepages/eric/seq.ps, June 1996.

[26] A. Ben-Dor, R. Karp, B. Schwikowski, and Z. Yakhini. Universal DNA tag systems: A combinatorial design scheme. *Journal of Computational Biology*, 7(3):503–519, 2000.

[27] Y. Benenson, R. Adar, T. Paz-Elilzur, Z. Livneh, and E. Shapiro. DNA molecule provides a computing machine with both data and fuel. *Proceedings of the National Academy of Sciences*, 100(5):2191–2196, March 2003.

[28] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro. An autonomous molecular computer for logical control of gene expression. *Nature*, 429:423–429, 2004.

[29] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414:430–434, November 2001.

[30] S. Bommarito, N. Peyret, and J. SantaLucia, Jr. Thermodynamic parameters for DNA sequences with dangling ends. *Nucleic Acids Research*, 28(9):1929–1934, 2000.

[31] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. Adleman. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 296:499–502, 2002.

[32] E. Braun, Y. Eichen, U. Sivan, and G. Ben-Yoseph. DNA-templated assembly and electrode attachment of a conducting silver wire. *Nature*, 391:775–778, February 1998.

[33] K. J. Breslauer, R. Frank, and H. Blöcker. Predicting DNA duplex stability from the base sequence. *Proceedings of the National Academy of Sciences*, 83(4):3746–3750, 1986.

[34] C. R. Cantor and P. R. Schimmel. *Biophysical Chemistry Part III: The Behavior of Biological Macromolecules*. W. H. Freeman and Company, 1980.

[35] A. Carbone and N. C. Seeman. Circuits and programmable self-assembling DNA structures. *Proceedings of the National Academy of Sciences*, 99(20):12577–12582, 2002.

[36] J. Chen and J. Reif, editors. *DNA Computing: 9th International Workshop on DNA Based Computers, DNA9, Madison, WI, USA, June 1-3, 2003. Revised Papers*, volume 2943 of *LNCS*. Springer, 2004.

[37] S. H. Chen, C. Y. Lin, C. S. Cho, C. Z. Lo, and C. A. Hsiung. Primer design assistant (PDA): a web-based primer design tool. *Nucleic Acids Research*, 31(13):3751–3754, 2003.

[38] C. T. Clelland, V. Risca, and C. Bancroft. Hiding messages in DNA microdots. *Nature*, 399:533–534, June 1999.

[39] A. Cornish-Bowden. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic Acids Research*, 13(9):3021–3032, 1985.

[40] J. C. Cox, D. S. Cohen, and A. D. Ellington. The complexities of DNA computing. *Trends in Biotechnology*, 17:151–154, 1999.

[41] N. G. de Bruijn. A combinatorial problem. *Proceedings of the Koninklijke Nederlandsche Akademie van Wetenschappen*, 49(6–10):758–764, 1946.

[42] R. Deaton, J. Chen, H. Bi, M. Garzon, H. Rubin, and D. H. Wood. A PCR-based protocol for in vitro selection of non-crosshybridizing oligonucleotides. In Hagiya and Ohuchi [70], pages 196–204.

[43] R. Deaton, J. Chen, H. Bi, and J. A. Rose. A software tool for generating non-crosshybridizing libraries of DNA oligonucleotides. In Hagiya and Ohuchi [70], pages 252–261.

[44] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr. Good encodings for DNA-based solutions to combinatorial problems. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University, June 10-12, 1996* [11], pages 159–171.

[45] R. Deaton, R. C. Murphy, J. A. Rose, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr. Genetic search for reliable encodings for DNA-based computation. In MIT Press, editor, *First Conference on Genetic Programming*, Stanford University, 1996.

[46] C. Dekker and M. Ratner. Electronic properties of DNA. *Physics World*, August 2001.

[47] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

[48] W. U. Dittmer, A. Reuter, and F. C. Simmel. A DNA-based machine that can cyclically bind and release thrombin. *Angewandte Chemie International Edition*, 43:3550–3553, 2004.

[49] R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[50] L. H. Eckardt, K. Naumann, W. M. Pankau, M. Rein, M. Schweitzer, N. Windhab, and G. von Kiedrowski. Chemical copying of connectivity. *Nature*, 420:286, 2002.

[51] S. J. Emrich, M. Lowe, and A. L. Delcher. PROBEmer: a web-based software tool for selecting optimal DNA oligos. *Nucleic Acids Research*, 31(13):3746–3750, 2003.

[52] D. Faulhammer, A. R. Cukras, R. J. Lipton, and L. F. Landweber. Molecular computation: RNA solutions to chess problems. *Proceedings of the National Academy of Sciences*, 97(4):1385–1389, February 2000.

[53] U. Feldkamp. Ein DNA-Sequenz-Compiler. Technical Report of the Systems Analysis Research Group SYS–2/00, Universität Dortmund, Fachbereich Informatik, November 2000. (basiert auf Diplomarbeit).

[54] U. Feldkamp, H. Rauhe, and W. Banzhaf. Software tools for DNA sequence design. *Genetic Programming and Evolvable Machines*, 4(2):153–171, June 2003.

[55] U. Feldkamp, R. Wacker, H. Schroeder, W. Banzhaf, and C. M. Niemeyer. Microarray-based in vitro evaluation of DNA oligomer libraries designed in silico. *ChemPhysChem*, 5(3):367–372, 2004.

[56] C. Ferretti, G. Mauri, and C. Zandron, editors. *Preliminary Proceedings of the tenth International Meeting on DNA Computing, June 7-10, 2004 - University of Milano-Bicocca*, 2004.

[57] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith, and R. M. Corn. Demonstration of a word design strategy for DNA computing on surfaces. *Nucleic Acids Research*, 25(23):4748–4757, 1997.

[58] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[59] M. Garzon, R. J. Deaton, J. A. Rose, and D. R. Franceschetti. Soft molecular computing. In E. Winfree and D. K. Gifford, editors, *Proceedings of the 5th DIMACS Workshop on DNA Based Computers, held at the Massachussetts Institute of Technology, Cambridge, MA, USA June 14 - June 15, 1999*, pages 91–100. American Mathematical Society, 1999.

[60] M. H. Garzon, K. V. Bobba, and B. P. Hyde. Digital information encoding on DNA. In Jonoska et al. [82], pages 152–166.

[61] M. H. Garzon, R. Deaton, P. Neathery, D. R. Franceschetti, and R.C. Murphy. A new metric for DNA computing. In J. R. Koza, D. Kalyanmoy, D. Marco, M. Fogel, D. B.and Garzon, I. Hitoshi, and R. L. Riolo, editors, *Conference on Genetic Programming*, Stanford University, Stanford, California, July13–16, 1997. Special Track on DNA computing.

[62] M. H. Garzon and R. J. Deaton. Biomolecular computing and programming: a definition. *Künstliche Intelligenz*, (1):39–40, 2000.

[63] H. G. Gassen and K. Minol, editors. *Gentechnik*, volume 1290 of *UTB*. Gustav Fischer Verlag, Stuttgart, 4. edition, 1996.

[64] Genosys. http://www.genosys.co.uk/oligos/tech_info/melting_temp.pdf. (23.11.2004).

[65] N. P. Gerry, N. E. Witowski, J. Day, R. P. Hammer, G. Barany, and F. Barany. Universal DNA microarray method for multiplex detection of low abundance point mutations. *Journal of Molecular Biology*, 292:251–262, 1999.

[66] R. P. Goodman, R. M. Berry, and A. J. Turberfield. The single-step synthesis of a DNA tetrahedron. *Chemical Communications*, (12):1372–1373, 2004.

[67] F. Guarnieri, M. Fliss, and C. Bancroft. Making DNA add. *Science*, 273(5272):220–223, July 12 1996.

[68] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 67(3):473–496, 2003.

[69] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[70] M. Hagiya and A. Ohuchi, editors. *DNA Computing : 8th International Workshop on DNA-Based Computers, DNA8 Sapporo, Japan, June 10-13, 2002. Revised Papers*, volume 2568 of *LNCS*. Springer, 2003.

[71] A. J. Hartemink, D. K. Gifford, and J. Khodor. Automated constraint-based nucleotide sequence selection for DNA-computation. In L. Kari, H. Rubin, and D. H. Wood, editors, *Proceedings of the 4th DIMACS Workshop on DNA Based Computers, held at the University of Pennysylvania, Philadelphia, USA June 15 - 19, 1998*, pages 227–235, 1998. University of Pennsylvania, Philadelphia, June 15 – 19, 1998.

[72] P. Hazarika, B. Ceyhan, and C. M. Niemeyer. Reversible switching of DNA-gold nanoparticle aggregation. *Angewandte Chemie*, 116:6631–6633, 2004.

[73] K. E. Herold and A. Rasooly. OligoDesign: a computer program for development of probes for oligonucleotide microarrays. *BioTechniques*, 35:1216–1221, 2003.

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

[75] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125:167–188, 1994.

[76] D. M. Hoover and J. Lubkowski. DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis. *Nucleic Acids Research*, 30(10):e43, 2002.

[77] S. Hussini, L. Kari, and S. Konstantinidis. Coding properties of DNA languages. In Jonoska and Seeman [83], pages 57–69.

[78] D. Hyndman, A. Cooper, S. Pruzinsky, D. Coad, and M. Mitsuhashi. Software to determine optimal oligonucleotide sequences based on hybridization simulation data. *BioTechniques*, 20:1090–1097, 1996.

[79] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[80] N. Jonoska and K. Mahalingam. Languages of DNA based code words. In Chen and Reif [36], pages 61–73.

[81] N. Jonoska and K. Mahalingam. Methods for constructing coded DNA languages. In Jonoska et al. [82], pages 241–253.

[82] N. Jonoska, G. Paun, and G. Rozenberg, editors. *Aspects of Molecular Computing, Essays Dedicated to Tom Head on the Occasion of His 70th Birthday*, volume 2950 of *Lecture Notes in Computer Science*. Springer, 2004.

[83] N. Jonoska and N. C. Seeman, editors. *DNA Computing, 7th International Workshop on DNA-Based Computers, DNA 2001, Tampa, U.S.A., 10-13 June 2001*, LNCS. University of South Florida, Tampa, FL, Springer, 2002.

[84] L. Kaderali, A. Deshpande, J. P. Nolan, and P. S. White. Primer-design for multiplexed genotyping. *Nucleic Acids Research*, 31(6):1796–1802, 2003.

[85] T. Kämpke, M. Kieninger, and M. Mecklenburg. Efficient primer design algorithms. *Bioinformatics*, 17(3):214–225, 2001.

[86] L. Kari, R. Kitto, and G. Thierrin. Codes, involutions and DNA encodings. In W. Brauer, H. Ehrig, J. Karhumäki, and A. Salomaa, editors, *Formal and Natural Computing*, volume 2300 of *LNCS*, pages 376–393. Springer, 2002.

[87] L. Kari, S. Konstantinidis, and P. Sosik. Bond-free languages: Formalizations, maximality and construction methods. In Ferretti et al. [56]. to appear in Springer LNCS series.

[88] K Keren, R. S. Berman, E. Buchstab, U. Sivan, and E. Braun. DNA-templated carbon nanotube field-effect transistor. *Science*, 302:1380–1382, 2003.

[89] K. Keren, M. Krueger, R. Gilad, G. Ben-Yoseph, U. Sivan, and E. Braun. Sequence-specific molecular lithography on single DNA molecules. *Science*, 297:72–75, 2002.

[90] D. Kim, S.-Y. Shin, I.-H. Lee, and B.-T. Zhang. NACST/Seq: A sequence design system with multiobjective optimization. In Hagiya and Ohuchi [70], pages 242–251.

[91] J. M. Kleinberg. *Approximation Algorithms for Disjoint Path Problems*. PhD thesis, Massachusetts Institute of Technology, 1996.

[92] R. Knippers. *Molekulare Genetik*. Georg Thieme Verlag, 6. edition, 1995.

[93] S. Kobayashi, T. Kondo, and M. Arita. On template method for DNA sequence design. In Hagiya and Ohuchi [70], pages 205–214.

[94] T. LaBean, H. Yan, J. Kopatsch, F. R. Liu, and E. Winfree. Construction, analysis, ligation, and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122:1848–1860, 2000.

[95] J. D. Le, Y. Pinto, N. C. Seeman, K. Musier-Forsyth, T. A. Taton, and R. A. Kiehl. DNA-templated self-assembly of metallic nanocomponent arrays on a surface. *Nano Letters*, 4(12):2343–2347, 2004.

[96] A. Leier, C. Richter, W. Banzhaf, and H. Rauhe. Cryptography with DNA binary strands. *BioSystems*, 57:13–22, 2000.

[97] A. Lempel and J. Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22(1):75–81, January 1976.

[98] F. Li and G. D. Stormo. Selection of optimal DNA oligos for gene expression arrays. *Bioinformatics*, 17(11):1067–1076, 2001.

[99] H. Li, S. H. Park, J. H. Reif, T. H. LaBean, and H. Yan. DNA-templated self-assembly of protein and nanoparticle linear arrays. *Journal of the American Chemical Society*, 126:418–419, 2004.

[100] Y. Li, Y. D. Tseng, S. Y. Kwon, L. d'Espaux, and J. S. Bunch. Controlled assembly of dendrimer-like DNA. *Nature Materials*, 3:38–42, 2004.

[101] M. Linial and N. Linial. On the potential of molecular computing. *Science*, 268:481, 1995.

[102] R. J. Lipton. DNA solution of hard computational problems. *Science*, 268:542–545, April 1995.

[103] D. Liu, S. H. Park, J. H. Reif, and T. H. LaBean. DNA nanotubes self-assembled from triple-crossover tiles as templates for conductive nanowires. *Proceedings of the National Academy of Sciences*, 101(3):717–722, 2004.

[104] Q. Liu, L. Wang, A. G. Frutos, A. E. Condon, R. M. Corn, and L. M. Smith. DNA computing on surfaces. *Nature*, 403:175–179, January 2000.

[105] W. Liu, S. Wang, L. Gao, F. Zhang, and J. Xu. DNA sequence design based on template strategy. *Journal of Chemical Information and Computer Sciences*, 43:2014–2018, 2003.

[106] M. S. Livstone, D. van Noort, and L. F. Landweber. Molecular computing revisited: a moore's law? *Trends in Biotechnology*, 21(3):98–101, 2003.

[107] Y.-M. D. Lo, K. F. C. Yu, and S. L. Wong. On the potential of molecular computing. *Science*, 268:481–482, 1995.

[108] B. Ma and L. Wang. On the inapproximability of disjoint paths and minimum steiner forest with bandwidth constraints. *Journal of Computer and System Sciences*, 60:1–12, 2000.

[109] C. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 407:493–496, 2000.

[110] C. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules (errata). *Nature*, 408:750, 2000.

[111] C. Mao, W. Sun, Z. Shen, and N. C. Seeman. A nanomechanical device based on the B-Z transition of DNA. *Nature*, 397:144–146, January 1999.

[112] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences, USA*, 101:7287–7292, 2004.

[113] A. P. Mills, Jr. Gene expression profiling diagnosis through DNA molecular computation. *Trends in Biotechnology*, 20(4):137–140, 2002.

[114] C. A. Mirkin. A DNA-based methodology for preparing nanocluster circuits, arrays, and diagnostic materials. *MRS Bulletin*, 25(1):43–54, January 2000.

[115] C. A. Mirkin, R. L. Letsinger, R. C. Mucic, and J. J. Storhoff. A DNA-based method for rationally assembling nanoparticles into macroscopic materials. *Nature*, 382:607–609, 1996.

[116] J. C. Mitchell and B. Yurke. DNA scissors. In Jonoska and Seeman [83], pages 258–268.

[117] J.-M. Nam, C. S. Thaxton, and C. A. Mirkin. Nanoparticle-based bio-bar codes for the ultrasensitive detection of proteins. *Science*, 301:1884–1886, 2003.

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

[119] H. B. Nielsen, R. Wernersson, and S. Knudsen. Design of oligonucleotides for microarrays and perspectives for design of multi-transcriptome arrays. *Nucleic Acids Research*, 31(13):3491–3496, 2003.

[120] C. M. Niemeyer and M. Adler. Nanomechanical deveces based on DNA. *Angew. Chem. Int. Ed.*, 41(20):3779–3783, 2002.

[121] C. M. Niemeyer, W. Bürger, and J. Peplies. Kovalente DNA-Streptavidin-Konjugate als Bausteine für neuartige biometallische Nanostrukturen. *Angewandte Chemie*, 110(16):2391–2395, 1998.

[122] C. M. Niemeyer, B. Ceyhan, and D. Blohm. Functionalization of covalent DNA-streptavidin conjugates by means of biotinylated modular components. *Bioconjugate Chemistry*, 10:708–719, 1999.

[123] C. M. Niemeyer, T. Sano, C. L. Smith, and C. R. Cantor. Oligonucleotide-directed self-assembly of proteins: semisynthetic DNA-streptavidin hybrid molecules as connectors for the generation of macroscopic arrays and the construction of supramolecular bioconjugates. *Nucleic Acids Research*, 22(25):5530–5539, 1994.

[124] J. S. Oliver. Computation with DNA: Matrix multiplication. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University, June 10-12, 1996* [11], pages 113–122.

[125] P. Pancoska, Z. Moravek, and U. M. Moll. Rational design of DNA sequences for nanotechnology, microarrays and molecular computers using eulerian graphs. *Nucleic Acids Research*, 32(15):4630–4645, 2004.

[126] G. Păun. From cells to computers: Membrane computing — a quick overview. In Ferretti et al. [56]. to appear in Springer LNCS series.

[127] R. Penchovsky and J. Ackermann. DNA library design for molecular computation. *Journal of Computational Biology*, 10(2):215–229, 2003.

[128] N. Peyret, P. A. Seneviratne, H. T. Allawi, and J. SantaLucia, Jr. Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A·A, C·C, G·G and T·T mismatches. *Biochemistry*, 38:3468–3477, 1999.

[129] V. Phan and M. H. Garzon. The capacity of DNA for information encoding. In Ferretti et al. [56]. to appear in Springer LNCS series.

[130] G. Raddatz, M. Dehio, T. F. Meyer, and C. Dehio. PrimeArray: genome-scale primer design for DNA-microarry construction. *Bioinformatics*, 17(1):98–99, 2001.

[131] A. Ralston. De Bruijn sequences — a model exampel of the interaction of discrete mathematics and computer science. *Mathematics Magazine*, 55(3):131–143, May 1982.

[132] H. Rauhe, G. Vopper, U. Feldkamp, W. Banzhaf, and J. C. Howard. Digital DNA molecules. In A. E. Condon and G. Rozenberg, editors, *Preproceedings of the 6th International Workshop on DNA-Based Computers, DNA 2000, Leiden, The Netherlands, June 2000*, page 271. Leiden center for natural computing, 2000. Poster, manuscript available under http://ls11-www.informatik.uni-dortmund.de/molcomp/Publications/publications.html.

[133] J. H. Reif. The design of autonomous DNA nanomechanical devices: Walking and rolling DNA. In Hagiya and Ohuchi [70], pages 22–37.

[134] V. I. Risca. DNA-based steganography. *Cryptologia*, 25(1):37–49, January 2001.

[135] J. A. Rose, R. J. Deaton, D. R. Franceschetti, M. Garzon, and S. E. Stevens, Jr. A statistical mechanical treatment of error in the annealing biostep of DNA computation. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 2, pages 1829–1834. Morgan Kaufmann, 1999.

[136] J. A. Rose, R. J. Deaton, M. Hagiya, and A. Suyama. The fidelity of the tag-antitag system. In Jonoska and Seeman [83], pages 138–149.

[137] J. A. Rose, M. Hagiya, and A. Suyama. The fidelity of the tag-antitag system II: Reconciliation with the stringency picture. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, pages 2740–2747. IEEE press, 2003.

[138] J.-M. Rouillard, M. Zuker, and E. Gulari. OligoArray 2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids Research*, 31(12):3057–3062, 2003.

[139] S. Roweis, E. Winfree, R. Burgoyne, N. V. Chelyapov, M. F. Goodman, P. W. K. Rothemund, and L. M. Adleman. A sticker based model for DNA computation. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University, June 10-12, 1996* [11], pages 1–29.

[140] A. J. Ruben, S. J. Freeland, and L. F. Landweber. PUNCH: An evolutionary algorithm for optimizing bit set selection. In Jonoska and Seeman [83], pages 150–160.

[141] W. Rychlik and R. E. Rhoads. A computer program for choosing optimal oligonucleotides for filter hybridization, sequencing and in vitro amplification of DNA. *Nucleic Acids Research*, 17(21):201–209, 1989.

[142] W. Saenger. *Principles of Nucleic Acid Structure*. Springer Advanced Texts in Chemistry. Springer, 1984.

[143] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya. Molecular computation by DNA hairpin formation. *Science*, 288:1223–1226, 2000.

[144] J. SantaLucia, Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95:1460–1465, February 1998.

[145] J. SantaLucia, Jr., H. T. Allawi, and P. A. Seneviratne. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35(11):3555–3562, 1996.

[146] M. Scheffler, A. Dorenbeck, S. Jordan, M. Wüstefeld, and G. von Kiedrowski. Self-assembly of trisoligonucleotidyls: The case for nano-acetylene and nano-cyclobutadiene. *Angewandte Chemie Int. Ed.*, 38(22):3122–3315, 1999.

[147] N. C. Seeman. De novo design of sequences for nucleic acid structural engineering. *Journal of Biomolecular Structure & Dynamics*, 8(3):573–581, 1990.

[148] N. C. Seeman. The design and engineering of nucleic acid nanoscale assemblies. *Current Opinion in Structural Biology*, 6:519–526, 1996.

[149] N. C. Seeman. DNA nanotechnology: Novel DNA constructions. *Annual Review of Biophysics and Biomolecular Structure*, 27:225–248, 1998.

[150] N. C. Seeman. DNA engineering and its application to nanotechnology. *Trends in Biotechnology*, 17:437–443, November 1999.

[151] D. Sen and W. Gilbert. A sodium-potassium switch in the formation of four-stranded G4-DNA. *Nature*, 344:410–414, 1990.

[152] W. M. Shih, J. D. Quispe, and G. F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–621, 2004.

[153] S.-Y. Shin, D.-M. Kim, I.-H. Lee, and B.-T. Zhang. Evolutionary sequence generation for reliable DNA computing. In *Proceedings of the 2002 Congress on Evolutionary Computing CEC'02*, pages 79–84. IEEE, 2002.

[154] S.-Y. Shin, B.-T. Zhang, and S.-S. Jun. Solving travelling salesman problem using molecular programming. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, pages 994–1000, 1999.

[155] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:185–197, 1981.

[156] W. D. Smith. DNA computers in vitro and in vivo. In *Proceedings of a DIMACS Workshop, held at Princeton University, 4 April 1995, Amer. Math. Soc., 1996*, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science., pages 121–186. American Mathematical Society, 1995.

[157] M. N. Stojanovic and D. Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21(9):1069–1074, 2003.

[158] X. Su and L. M. Smith. Demonstration of a universal surface DNA computer. *Nucleic Acids Research*, 32(10):3115–3123, 2004.

[159] N. Sugimoto, S. Nakano, M. Yoneyama, and K. Honda. Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes. *Nucleic Acids Research*, 24(22):4501–4505, 1996.

[160] W. I. Sundquist and A. Klug. Telomeric DNA dimerizes by formation of guanine tetrads between hairpin loops. *Nature*, 342:825–829, 1989.

[161] F. Tanaka, A. Kameda, M. Yamamoto, and A. Ohuchi. Nearest-neighbor thermodynamics of DNA sequences with single bulge loop. In Chen and Reif [36], pages 170–179.

[162] F. Tanaka, M. Nakatsugawa, M. Yamamoto, T. Shiba, and A. Ohuchi. Towards a general-purpose sequence design system in DNA computing. In *Proceedings of the 2002 Congress on Evolutionary Computing CEC'02*, pages 73–78. IEEE, 2002.

[163] K. Tanaka, A. Tengeiji, T. Kato, N. Toyama, and M. Shionoya. A discrete self-assembled metal array in artificial DNA. *Science*, 299:1212–1213, 2003.

[164] J. B. Tobler, M. N. Molla, E. F. Nuwaysir, R. D. Green, and J. W. Shavlik. Evaluating machine learning approaches for aiding probe selection for gene-expression arrays. *Bioinformatics*, 18(Suppl. 1):S164–S171, 2002.

[165] D. C. Tulpan, H. H. Hoos, and A. E. Condon. Stochastical local search algorithms for DNA word design. In Hagiya and Ohuchi [70], pages 229–241.

[166] D. C. Tulpan, H. H. Hoos, A. E. Condon, M. Shortreed, S. C. Bong, and L. M. Smith. Thermodynamically based DNA code design. In Ferretti et al. [56]. to appear in Springer LNCS series.

[167] C. Varotto, E. Richly, F. Slamini, and D. Leister. GST-PRIME: a genome-wide primer design software for the generation of gene sequence tags. *Nucleic Acids Research*, 29(21):4373–4377, 2001.

[168] J. C. Venter et al. The sequence of the human genome. *Science*, 291(5507):1304–51, 2001.

[169] S. Vinga and J. Almeida. Alignment-free sequence comparison — a review. *Bioinformatics*, 19(4):513–523, 2003.

[170] G. von Kiedrowski and S. Müller. Programmierbare biomolekulare Nanokonstrukte, Molekulare Kopiermaschinen. *chemie RUBIN*, pages 12–16, 2003.

[171] R. Wacker and C. M. Niemeyer. DDI-μFIA — a readily configurable microarray-fluorescence immunoassay based on DNA-directed immobilization of proteins. *ChemBioChem*, 5:453–459, 2004.

[172] R. Wacker, H. Schröder, and C. M. Niemeyer. Performance of antibody microarrays fabricated by either DNA-directed immobilization, direct spotting, or streptavidin-biotin attachment: a comparative study. *Analytical Biochemistry*, 330:281–287, 2004.

[173] J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[174] J. G. Wetmur. Physical chemistry af nucleic acid hybridization. In D. Wood, editor, *DNA3*, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science., pages 1–23, Providence, RI, 1997. American Mathematical Society.

[175] G. M. Whitesides, J. P. Mathias, and C. T. Seto. Molecular self-assembly and nanochemistry: A chemical strategy for the synthesis of nanostructures. *Science*, 254:1312–1319, November 1991.

[176] K. A. Williams, P. T. M. Veenhuizen, B. G. de la Torre, R. Eritja, and C. Dekker. Carbon nanotubes with DNA recognition. *Nature*, 420:761, 2002.

[177] E. Winfree, X. Yang, and N. C. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In *DNA2* [11].

[178] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, Santa Fe, NM, USA, 1995.

[179] H. Wu. An improved surface-based method for DNA computation. *BioSystems*, 59:1–5, 2001.

[180] J.-S. Wu, C. Lee, C.-C. Wu, and Y.-L. Shiue. Primer design using genetic algorithm. *Bioinformatics*, 20(11):1710–1717, 2004.

[181] T.-J. Wu, J. P. Burke, and D. B. Davison. A measure of DNA sequence dissimilarity based on mahalanobis distance between frequencies of words. *Biometrics*, 53:1431–1439, December 1997.

[182] H. Yan, S. H. Park, G. Finkelstein, J. H. Reif, and T. H. LaBean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301:1882–1884, 2003.

[183] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–65, January 2002.

[184] B. Yurke, A. J. Turberfield, A. P. Mills, Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, August 2000.

[185] M. Zheng, A. Jagota, E. D. Semke, B. A. Diner, R. S. McLean, S. R. Lustig, R. E. Richardson, and N. G. Tassi. DNA-assisted dispersion and separation of carbon nanotubes. *Nature Materials*, 2:338–342, 2003.

[186] M. Zuker, D.H. Mathews, and D.H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. In J. Barciszewski and B.F.C. Clark, editors, *RNA Biochemistry and Biotechnology*, NATO ASI Series, pages 11–43. Kluwer Academic Publishers, Dordrecht, NL, 1999.

Anhang A

Tabellen zum Distanzmaßvergleich

8-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^{LE}	d_5^{LE}	d_6^{LE}	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d^{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.42	0.20	-0.42	0.75	-0.68	-0.62	-0.56	0.04	0.07	0.11	0.05		0.17	0.14	0.06		0.13	-0.15	-0.08		0.15	0.08		0.15	0.08		0.10
H-Maß		1.00	0.55	-1.00	0.69	-0.72	-0.72	-0.71	0.34	0.36	0.32	0.26		0.50	0.33	0.27		0.40	-0.42	-0.28		0.42	0.28		0.42	0.28		0.43
H-Distanz			1.00	-0.55	0.35	-0.34	-0.34	-0.33	0.12	0.13	0.24	0.23		0.22	0.27	0.25		0.30	-0.33	-0.25		0.33	0.25		0.33	0.25		0.33
Homologie				1.00	-0.69	0.72	0.72	0.71	-0.34	-0.36	-0.32	-0.26		-0.50	-0.33	-0.27		-0.40	0.42	0.28		-0.42	-0.28		-0.42	-0.28		-0.43
Edit-Distanz					1.00	-0.97	-0.94	-0.89	0.23	0.26	0.23	0.17		0.37	0.24	0.18		0.28	-0.30	-0.21		0.30	0.21		0.30	0.21		0.32
NW (1, -2)						1.00	0.99	0.97	-0.27	-0.29	-0.26	-0.19		-0.44	-0.27	-0.19		0.30	0.33	0.22		-0.33	-0.22		-0.33	-0.22		-0.35
NW (1, -1)							1.00	0.99	-0.29	-0.31	-0.28	-0.19		-0.47	-0.28	-0.19		-0.31	0.33	0.22		-0.33	-0.22		-0.33	-0.22		-0.35
NW (2, -1)								1.00	-0.29	-0.32	-0.30	-0.20		-0.50	-0.29	-0.19		-0.33	0.34	0.22		-0.34	-0.22		-0.34	-0.22		-0.36
d^*									1.00	0.86	0.26	0.17		0.44	0.25	0.14		0.29	-0.29	-0.20		0.29	0.20		0.29	0.20		0.29
d_1^{**}										1.00	0.29	0.18		0.49	0.27	0.16		0.31	-0.31	-0.21		0.31	0.21		0.31	0.21		0.27
d_4^{LE}											1.00	0.67		0.64	0.96	0.57		0.85	-0.80	-0.54		0.80	0.55		0.80	0.55		0.42
d_5^{LE}												1.00		0.40	0.69	0.97		0.90	-0.59	-0.84		0.60	0.84		0.56	0.84		0.34
d_6^{LE}																												
$d^2(4,6)$														1.00	0.56	0.29		0.62	-0.48	-0.29		0.48	0.29		0.48	0.29		0.42
d_4^{SE}															1.00	0.65		0.89	-0.86	-0.60		0.86	0.60		0.86	0.60		0.44
d_5^{SE}																1.00		0.88	-0.60	-0.87		0.61	0.87		0.57	0.87		0.33
d_6^{SE}																												
$d^{SE*}(4,6)$																		1.00	-0.79	-0.79		0.80	0.79		0.78	0.79		0.47
d_4^{LCC}																			1.00	0.70		-1.00	-0.70		-1.00	-0.70		-0.50
d_5^{LCC}																				1.00		-0.71	-1.00		-0.66	-1.00		-0.38
d_6^{LCC}																												
d_4^{cos}																						1.00	0.71		1.00	0.71		0.50
d_5^{cos}																							1.00		0.66	1.00		0.38
d_6^{cos}																												
d_4^{Evol}																									1.00	0.66		0.50
d_5^{Evol}																										1.00		0.38
d_6^{Evol}																												
ΔG_{25}																												1.00

Tabelle A.1: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 8-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 8, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}. In den leeren Zeilen und Spalten (Distanzmaße mit Tupeln der Länge 6) konnte wegen einer Varianz von 0 kein Korrelationskoeffizienz berechnet werden.

10-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^E	d_5^E	d_6^E	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d^{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.40	0.22	-0.40	0.72	-0.66	-0.62	-0.58	0.07	0.08	0.10	0.10	0.04	0.13	0.14	0.14	0.07	0.13	-0.17	-0.19	-0.12	0.17	0.19	0.12	0.17	0.19	0.12	0.24
H-Maß		1.00	0.59	-1.00	0.67	-0.71	-0.71	-0.71	0.33	0.34	0.30	0.18	0.06	0.46	0.34	0.21	0.09	0.25	-0.44	-0.29	-0.17	0.44	0.29	0.17	0.44	0.30	0.17	0.44
H-Distanz			1.00	-0.59	0.39	-0.40	-0.39	-0.38	0.17	0.21	0.18	0.13	0.08	0.24	0.23	0.18	0.12	0.21	-0.30	-0.23	-0.15	0.30	0.23	0.15	0.29	0.23	0.15	0.27
Homologie				1.00	-0.67	0.71	0.71	0.71	-0.33	-0.34	-0.30	-0.18	-0.06	-0.46	-0.34	-0.21	-0.09	-0.25	0.44	0.29	0.17	-0.44	-0.29	-0.17	-0.44	-0.30	-0.17	-0.44
Edit-Distanz					1.00	-0.97	-0.94	-0.90	0.25	0.25	0.20	0.12	0.04	0.34	0.24	0.16	0.07	0.18	-0.31	-0.22	-0.14	0.31	0.22	0.14	0.31	0.22	0.14	0.34
NW (1, -2)						1.00	0.99	0.97	-0.28	-0.28	-0.25	-0.14	-0.05	-0.41	-0.27	-0.16	-0.07	-0.19	0.34	0.22	0.14	-0.34	-0.22	-0.14	-0.34	-0.22	-0.14	-0.35
NW (1, -1)							1.00	0.99	-0.29	-0.30	-0.27	-0.15	-0.05	-0.44	-0.29	-0.16	-0.07	-0.20	0.36	0.22	0.14	-0.36	-0.22	-0.14	-0.36	-0.22	-0.14	-0.36
NW (2, -1)								1.00	-0.30	-0.32	-0.30	-0.16	-0.06	-0.48	-0.31	-0.17	-0.07	-0.21	0.37	0.23	0.15	-0.37	-0.23	-0.15	-0.37	-0.23	-0.15	-0.36
d^*									1.00	0.86	0.29	0.19	0.12	0.40	0.31	0.21	0.13	0.25	-0.34	-0.22	-0.13	0.35	0.22	0.13	0.35	0.22	0.13	0.21
d_1^{**}										1.00	0.30	0.18	0.12	0.44	0.32	0.19	0.13	0.25	-0.37	-0.23	-0.16	0.38	0.23	0.16	0.38	0.23	0.16	0.25
d_4^E											1.00	0.75	0.49	0.76	0.96	0.64	0.33	0.67	-0.68	-0.41	-0.24	0.70	0.42	0.24	0.70	0.42	0.24	0.42
d_5^E												1.00	0.74	0.57	0.69	0.95	0.59	0.84	-0.42	-0.65	-0.43	0.44	0.65	0.43	0.42	0.65	0.43	0.28
d_6^E													1.00	0.38	0.48	0.74	0.94	0.90	-0.24	-0.43	-0.63	0.25	0.43	0.64	0.23	0.41	0.64	0.15
$d^2(4,6)$														1.00	0.69	0.46	0.25	0.55	-0.49	-0.28	-0.18	0.50	0.29	0.18	0.50	0.29	0.18	0.39
d_4^{SE}															1.00	0.66	0.39	0.72	-0.77	-0.47	-0.27	0.78	0.47	0.27	0.78	0.47	0.27	0.47
d_5^{SE}																1.00	0.69	0.89	-0.48	-0.75	-0.50	0.49	0.75	0.50	0.47	0.75	0.50	0.31
d_6^{SE}																	1.00	0.89	-0.25	-0.46	-0.70	0.26	0.47	0.70	0.23	0.45	0.70	0.16
$d^{SE*}(4,6)$																		1.00	-0.51	-0.63	-0.62	0.52	0.63	0.63	0.50	0.62	0.63	0.34
d_4^{LCC}																			1.00	0.62	0.35	-1.00	-0.62	-0.35	-1.00	-0.62	-0.35	-0.57
d_5^{LCC}																				1.00	0.67	-0.62	-1.00	-0.67	-0.59	-1.00	-0.67	-0.38
d_6^{LCC}																					1.00	-0.36	-0.67	-1.00	-0.32	-0.64	-1.00	-0.21
d_4^{cos}																						1.00	0.62	0.36	1.00	0.63	0.36	0.57
d_5^{cos}																							1.00	0.67	0.59	1.00	0.67	0.38
d_6^{cos}																								1.00	0.32	0.64	1.00	0.21
d_4^{Evol}																									1.00	0.59	0.32	0.57
d_5^{Evol}																										1.00	0.64	0.38
d_6^{Evol}																											1.00	0.21
ΔG_{25}																												1.00

Tabelle A.2: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 10-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 10, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

15-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d^*_1	d^E_4	d^E_5	d^E_6	$d^2(4,6)$	d^{SE}_4	d^{SE}_5	d^{SE}_6	$d^{SE*}(4,6)$	d^{LCC}_4	d^{LCC}_5	d^{LCC}_6	d^{cos}_4	d^{cos}_5	d^{cos}_6	d^{Evol}_4	d^{Evol}_5	d^{Evol}_6	ΔG_{25}
Hamming	1.00	0.26	0.15	-0.26	0.59	-0.54	-0.50	-0.46	0.12	0.12	0.11	0.06	0.04	0.13	0.13	0.09	0.04	0.08	-0.18	-0.11	-0.08	0.18	0.11	0.08	0.18	0.11	0.08	0.23
H-Maß		1.00	0.60	-1.00	0.57	-0.59	-0.58	-0.57	0.30	0.30	0.27	0.22	0.18	0.34	0.29	0.24	0.19	0.26	-0.38	-0.31	-0.23	0.38	0.31	0.23	0.38	0.31	0.23	0.43
H-Distanz			1.00	-0.60	0.27	-0.26	-0.25	-0.23	0.13	0.12	0.15	0.14	0.10	0.11	0.17	0.15	0.10	0.14	-0.24	-0.24	-0.20	0.24	0.24	0.20	0.24	0.24	0.20	0.23
Homologie				1.00	-0.57	0.59	0.58	0.57	-0.30	-0.30	-0.27	-0.22	-0.18	-0.34	-0.29	-0.24	-0.19	-0.26	0.38	0.31	0.23	-0.38	-0.31	-0.23	-0.38	-0.31	-0.23	-0.43
Edit-Distanz					1.00	-0.97	-0.92	-0.87	0.32	0.35	0.24	0.16	0.12	0.37	0.24	0.16	0.12	0.19	-0.33	-0.21	-0.15	0.33	0.21	0.15	0.33	0.21	0.15	0.40
NW (1, -2)						1.00	0.99	0.96	-0.35	-0.37	-0.27	-0.18	-0.14	-0.43	-0.26	-0.18	-0.14	-0.21	0.33	0.22	0.15	-0.34	-0.22	-0.15	-0.34	-0.22	-0.15	-0.42
NW (1, -1)							1.00	0.99	-0.36	-0.39	-0.28	-0.20	-0.16	-0.46	-0.26	-0.19	-0.15	-0.22	0.33	0.22	0.15	-0.34	-0.22	-0.15	-0.34	-0.22	-0.15	-0.43
NW (2, -1)								1.00	-0.37	-0.40	-0.28	-0.20	-0.16	-0.48	-0.27	-0.19	-0.16	-0.23	0.33	0.22	0.14	-0.33	-0.22	-0.14	-0.34	-0.22	-0.14	-0.43
d^*									1.00	0.88	0.36	0.27	0.20	0.51	0.37	0.29	0.21	0.32	-0.46	-0.37	-0.25	0.46	0.37	0.26	0.46	0.37	0.25	0.43
d^{**}_1										1.00	0.40	0.29	0.21	0.56	0.40	0.30	0.22	0.33	-0.53	-0.41	-0.30	0.53	0.41	0.30	0.53	0.41	0.30	0.49
d^E_4											1.00	0.81	0.56	0.74	0.97	0.76	0.51	0.73	-0.69	-0.56	-0.34	0.72	0.57	0.34	0.72	0.57	0.34	0.38
d^E_5												1.00	0.78	0.57	0.80	0.98	0.74	0.90	-0.55	-0.70	-0.49	0.57	0.70	0.49	0.56	0.70	0.50	0.33
d^E_6													1.00	0.43	0.55	0.77	0.99	0.94	-0.33	-0.47	-0.65	0.34	0.48	0.65	0.33	0.46	0.65	0.27
$d^2(4,6)$														1.00	0.69	0.52	0.39	0.58	-0.53	-0.41	-0.25	0.55	0.41	0.25	0.55	0.41	0.25	0.45
d^{SE}_4															1.00	0.79	0.52	0.76	-0.74	-0.60	-0.36	0.76	0.60	0.36	0.76	0.60	0.36	0.41
d^{SE}_5																1.00	0.77	0.92	-0.58	-0.73	-0.51	0.60	0.74	0.51	0.59	0.74	0.52	0.36
d^{SE}_6																	1.00	0.94	-0.34	-0.48	-0.66	0.35	0.49	0.66	0.34	0.47	0.66	0.29
$d^{SE*}(4,6)$																		1.00	-0.53	-0.63	-0.62	0.55	0.64	0.63	0.54	0.63	0.63	0.38
d^{LCC}_4																			1.00	0.79	0.46	-1.00	-0.79	-0.46	-1.00	-0.79	-0.46	-0.57
d^{LCC}_5																				1.00	0.71	-0.79	-1.00	-0.71	-0.77	-1.00	-0.71	-0.50
d^{LCC}_6																					1.00	-0.47	-0.72	-1.00	-0.44	-0.69	-1.00	-0.41
d^{cos}_4																						1.00	0.79	0.47	1.00	0.79	0.47	0.57
d^{cos}_5																							1.00	0.72	0.77	1.00	0.72	0.50
d^{cos}_6																								1.00	0.44	0.69	1.00	0.41
d^{Evol}_4																									1.00	0.77	0.44	0.56
d^{Evol}_5																										1.00	0.69	0.49
d^{Evol}_6																											1.00	0.41
ΔG_{25}																												1.00

Tabelle A.3: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 15-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 15, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

20-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^E	d_5^E	d_6^E	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d^{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.25	0.07	-0.25	0.49	-0.45	-0.41	-0.38	0.06	0.02	0.10	0.08	0.06	0.11	0.10	0.07	0.06	0.08	-0.11	-0.09	-0.09	0.11	0.09	0.09	0.11	0.09	0.09	0.06
H-Maß		1.00	0.55	-1.00	0.57	-0.59	-0.59	-0.59	0.14	0.15	0.18	0.13	0.13	0.31	0.16	0.10	0.11	0.16	-0.23	-0.18	-0.21	0.23	0.18	0.21	0.23	0.18	0.21	0.36
H-Distanz			1.00	-0.55	0.28	-0.28	-0.27	-0.26	0.06	0.08	0.02	0.02	0.09	0.09	0.02	0.01	0.08	0.07	-0.09	-0.07	-0.13	0.09	0.07	0.13	0.09	0.06	0.13	0.22
Homologie				1.00	-0.57	0.59	0.59	0.59	-0.14	-0.15	-0.18	-0.13	-0.13	-0.31	-0.16	-0.10	-0.11	-0.16	0.23	0.18	0.21	-0.23	-0.18	-0.21	-0.23	-0.18	-0.21	-0.36
Edit-Distanz					1.00	-0.97	-0.94	-0.89	0.19	0.19	0.19	0.11	0.11	0.31	0.16	0.08	0.09	0.13	-0.24	-0.15	-0.16	0.24	0.15	0.16	0.24	0.15	0.16	0.32
NW (1, -2)						1.00	0.99	0.96	-0.21	-0.22	-0.21	-0.13	-0.12	-0.35	-0.17	-0.09	-0.09	-0.15	0.26	0.17	0.17	-0.26	-0.17	-0.17	-0.26	-0.17	-0.18	-0.35
NW (1, -1)							1.00	0.99	-0.22	-0.24	-0.22	-0.14	-0.12	-0.37	-0.18	-0.10	-0.10	-0.15	0.26	0.17	0.18	-0.27	-0.17	-0.18	-0.27	-0.17	-0.18	-0.36
NW (2, -1)								1.00	-0.23	-0.26	-0.24	-0.15	-0.13	-0.40	-0.20	-0.11	-0.10	-0.16	0.27	0.18	0.18	-0.28	-0.18	-0.18	-0.28	-0.18	-0.18	-0.36
d^*									1.00	0.87	0.34	0.24	0.14	0.38	0.34	0.25	0.14	0.26	-0.42	-0.32	-0.18	0.43	0.32	0.18	0.43	0.32	0.18	0.35
d_1^{**}										1.00	0.37	0.27	0.16	0.43	0.37	0.28	0.17	0.29	-0.44	-0.35	-0.20	0.45	0.35	0.20	0.45	0.35	0.21	0.34
d_4^E											1.00	0.74	0.45	0.75	0.98	0.69	0.39	0.69	-0.71	-0.50	-0.31	0.74	0.51	0.31	0.75	0.52	0.31	0.37
d_5^E												1.00	0.75	0.51	0.72	0.98	0.70	0.89	-0.51	-0.69	-0.51	0.53	0.69	0.51	0.53	0.70	0.51	0.26
d_6^E													1.00	0.33	0.44	0.74	0.98	0.92	-0.36	-0.59	-0.75	0.37	0.59	0.75	0.36	0.58	0.75	0.20
$d^2(4,6)$														1.00	0.70	0.46	0.29	0.53	-0.51	-0.32	-0.19	0.53	0.33	0.20	0.54	0.33	0.20	0.44
d_4^{SE}															1.00	0.72	0.41	0.70	-0.74	-0.52	-0.32	0.77	0.53	0.32	0.77	0.53	0.32	0.38
d_5^{SE}																1.00	0.73	0.91	-0.53	-0.71	-0.53	0.55	0.72	0.53	0.54	0.73	0.53	0.26
d_6^{SE}																	1.00	0.92	-0.36	-0.60	-0.77	0.37	0.61	0.77	0.36	0.59	0.77	0.20
$d^{SE*}(4,6)$																		1.00	-0.56	-0.70	-0.69	0.58	0.70	0.69	0.57	0.70	0.69	0.31
d_4^{LCC}																			1.00	0.74	0.51	-1.00	-0.74	-0.51	-1.00	-0.74	-0.52	-0.52
d_5^{LCC}																				1.00	0.79	-0.74	-1.00	-0.79	-0.72	-1.00	-0.79	-0.44
d_6^{LCC}																					1.00	-0.51	-0.79	-1.00	-0.49	-0.78	-1.00	-0.33
d_4^{cos}																						1.00	0.74	0.51	1.00	0.74	0.52	0.52
d_5^{cos}																							1.00	0.79	0.73	1.00	0.80	0.44
d_6^{cos}																								1.00	0.49	0.78	1.00	0.33
d_4^{Evol}																									1.00	0.73	0.49	0.52
d_5^{Evol}																										1.00	0.78	0.43
d_6^{Evol}																											1.00	0.33
ΔG_{25}																												1.00

Tabelle A.4: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 20-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 20, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

25-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d^*_1	d^E_4	d^E_5	d^E_6	$d^2(4,6)$	d^{SE}_4	d^{SE}_5	d^{SE}_6	$d^{SE*}(4,6)$	d^{LCC}_4	d^{LCC}_5	d^{LCC}_6	d^{cos}_4	d^{cos}_5	d^{cos}_6	d^{Evol}_4	d^{Evol}_5	d^{Evol}_6	ΔG_{25}
Hamming	1.00	0.24	0.14	-0.24	0.44	-0.42	-0.40	-0.37	-0.01	-0.04	-0.07	-0.09	-0.11	-0.02	-0.06	-0.08	-0.11	-0.10	0.03	0.05	0.07	-0.04	-0.05	-0.07	-0.04	-0.05	-0.07	0.06
H-Maß		1.00	0.60	-1.00	0.50	-0.50	-0.50	-0.49	0.21	0.23	0.21	0.14	0.10	0.27	0.22	0.15	0.10	0.17	-0.28	-0.22	-0.17	0.28	0.22	0.17	0.28	0.22	0.17	0.31
H-Distanz			1.00	-0.60	0.22	-0.23	-0.23	-0.23	0.15	0.18	0.12	0.10	0.08	0.11	0.13	0.10	0.08	0.11	-0.17	-0.14	-0.13	0.17	0.14	0.13	0.17	0.14	0.13	0.21
Homologie				1.00	-0.50	0.50	0.50	0.49	-0.21	-0.23	-0.21	-0.14	-0.10	-0.27	-0.22	-0.15	-0.10	-0.17	0.28	0.22	0.17	-0.28	-0.22	-0.17	-0.28	-0.22	-0.17	-0.31
Edit-Distanz					1.00	-0.97	-0.93	-0.89	0.22	0.22	0.17	0.10	0.06	0.27	0.17	0.10	0.06	0.12	-0.22	-0.14	-0.10	0.22	0.14	0.10	0.22	0.14	0.10	0.28
NW (1, -2)						1.00	0.99	0.96	-0.24	-0.25	-0.20	-0.12	-0.08	-0.32	-0.19	-0.12	-0.08	-0.15	0.24	0.17	0.13	-0.25	-0.17	-0.13	-0.25	-0.17	-0.13	-0.31
NW (1, -1)							1.00	0.99	-0.25	-0.26	-0.22	-0.14	-0.09	-0.35	-0.21	-0.14	-0.10	-0.16	0.26	0.18	0.14	-0.27	-0.18	-0.14	-0.27	-0.18	-0.15	-0.33
NW (2, -1)								1.00	-0.26	-0.28	-0.23	-0.16	-0.11	-0.37	-0.23	-0.16	-0.11	-0.18	0.28	0.20	0.16	-0.28	-0.20	-0.16	-0.29	-0.20	-0.16	-0.34
d^*									1.00	0.86	0.45	0.36	0.25	0.46	0.45	0.36	0.25	0.37	-0.43	-0.33	-0.25	0.45	0.34	0.25	0.45	0.34	0.26	0.33
d^{**}_1										1.00	0.47	0.38	0.25	0.51	0.46	0.37	0.24	0.38	-0.47	-0.40	-0.31	0.48	0.40	0.31	0.48	0.40	0.31	0.36
d^E_4											1.00	0.81	0.54	0.78	0.99	0.78	0.50	0.77	-0.68	-0.54	-0.38	0.73	0.55	0.38	0.74	0.55	0.38	0.39
d^E_5												1.00	0.75	0.61	0.81	0.99	0.73	0.91	-0.53	-0.70	-0.55	0.57	0.71	0.55	0.57	0.71	0.55	0.33
d^E_6													1.00	0.41	0.54	0.75	0.99	0.93	-0.39	-0.59	-0.74	0.41	0.60	0.74	0.41	0.59	0.74	0.27
$d^2(4,6)$														1.00	0.75	0.58	0.38	0.62	-0.50	-0.38	-0.28	0.54	0.39	0.28	0.54	0.39	0.28	0.40
d^{SE}_4															1.00	0.80	0.52	0.78	-0.71	-0.56	-0.38	0.76	0.57	0.38	0.76	0.57	0.39	0.40
d^{SE}_5																1.00	0.74	0.92	-0.55	-0.72	-0.55	0.59	0.74	0.56	0.59	0.74	0.56	0.35
d^{SE}_6																	1.00	0.92	-0.40	-0.60	-0.74	0.42	0.61	0.75	0.41	0.60	0.75	0.27
$d^{SE*}(4,6)$																		1.00	-0.56	-0.70	-0.69	0.60	0.71	0.69	0.60	0.70	0.69	0.37
d^{LCC}_4																			1.00	0.76	0.52	-1.00	-0.76	-0.52	-1.00	-0.76	-0.52	-0.52
d^{LCC}_5																				1.00	0.80	-0.76	-1.00	-0.79	-0.75	-1.00	-0.80	-0.44
d^{LCC}_6																					1.00	-0.52	-0.80	-1.00	-0.51	-0.79	-1.00	-0.36
d^{cos}_4																						1.00	0.76	0.52	1.00	0.77	0.52	0.52
d^{cos}_5																							1.00	0.80	0.75	1.00	0.80	0.44
d^{cos}_6																								1.00	0.51	0.79	1.00	0.36
d^{Evol}_4																									1.00	0.76	0.51	0.52
d^{Evol}_5																										1.00	0.79	0.44
d^{Evol}_6																											1.00	0.36
ΔG_{25}																												1.00

Tabelle A.5: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 25-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 25, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

30-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^E	d_5^E	d_6^E	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d^{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.35	0.23	-0.35	0.46	-0.42	-0.39	-0.37	0.07	0.09	0.15	0.11	0.05	0.15	0.14	0.10	0.05	0.09	-0.19	-0.14	-0.07	0.19	0.14	0.07	0.19	0.14	0.07	0.18
H-Maß		1.00	0.63	-1.00	0.58	-0.55	-0.54	-0.52	0.12	0.14	0.20	0.18	0.13	0.24	0.20	0.18	0.12	0.18	-0.29	-0.23	-0.12	0.28	0.23	0.12	0.28	0.23	0.12	0.29
H-Distanz			1.00	-0.63	0.35	-0.33	-0.31	-0.29	-0.02	0.00	0.11	0.10	0.07	0.13	0.12	0.11	0.06	0.10	-0.15	-0.10	-0.04	0.15	0.10	0.04	0.15	0.10	0.04	0.14
Homologie				1.00	-0.58	0.55	0.54	0.52	-0.12	-0.14	-0.20	-0.18	-0.13	-0.24	-0.20	-0.18	-0.12	-0.18	0.29	0.23	0.12	-0.28	-0.23	-0.12	-0.28	-0.23	-0.12	-0.29
Edit-Distanz					1.00	-0.97	-0.93	-0.88	0.13	0.16	0.25	0.23	0.14	0.33	0.22	0.21	0.12	0.20	-0.28	-0.24	-0.08	0.29	0.24	0.09	0.29	0.24	0.09	0.36
NW (1, -2)						1.00	0.99	0.96	-0.14	-0.17	-0.26	-0.23	-0.15	-0.37	-0.23	-0.20	-0.13	-0.20	0.28	0.23	0.08	-0.29	-0.24	-0.08	-0.29	-0.24	-0.08	-0.39
NW (1, -1)							1.00	0.99	-0.16	-0.18	-0.27	-0.23	-0.16	-0.39	-0.24	-0.20	-0.13	-0.21	0.29	0.24	0.09	-0.30	-0.24	-0.09	-0.30	-0.24	-0.09	-0.41
NW (2, -1)								1.00	-0.18	-0.20	-0.28	-0.24	-0.16	-0.41	-0.25	-0.21	-0.14	-0.22	0.30	0.25	0.10	-0.31	-0.25	-0.10	-0.31	-0.25	-0.10	-0.42
d^*									1.00	0.87	0.37	0.25	0.14	0.41	0.38	0.25	0.14	0.26	-0.41	-0.32	-0.19	0.42	0.32	0.19	0.42	0.32	0.20	0.20
d_1^{**}										1.00	0.41	0.28	0.16	0.46	0.41	0.28	0.16	0.29	-0.46	-0.36	-0.22	0.48	0.36	0.22	0.48	0.36	0.22	0.23
d_4^E											1.00	0.78	0.54	0.80	0.98	0.76	0.50	0.75	-0.66	-0.47	-0.23	0.73	0.48	0.24	0.74	0.49	0.24	0.41
d_5^E												1.00	0.79	0.59	0.77	0.99	0.76	0.91	-0.49	-0.65	-0.46	0.55	0.67	0.46	0.54	0.67	0.46	0.35
d_6^E													1.00	0.44	0.53	0.78	0.99	0.94	-0.30	-0.49	-0.63	0.33	0.51	0.63	0.33	0.50	0.63	0.26
$d^2(4,6)$														1.00	0.76	0.55	0.40	0.62	-0.50	-0.33	-0.17	0.56	0.34	0.17	0.56	0.35	0.17	0.42
d_4^{SE}															1.00	0.77	0.51	0.76	-0.69	-0.48	-0.24	0.76	0.50	0.25	0.76	0.50	0.25	0.40
d_5^{SE}																1.00	0.77	0.92	-0.52	-0.68	-0.48	0.57	0.70	0.48	0.56	0.70	0.48	0.33
d_6^{SE}																	1.00	0.93	-0.30	-0.51	-0.65	0.34	0.52	0.65	0.33	0.51	0.65	0.25
$d^{SE*}(4,6)$																		1.00	-0.49	-0.60	-0.57	0.54	0.62	0.58	0.53	0.61	0.58	0.35
d_4^{LCC}																			1.00	0.75	0.45	-1.00	-0.75	-0.45	-0.99	-0.75	-0.45	-0.46
d_5^{LCC}																				1.00	0.75	-0.74	-1.00	-0.75	-0.73	-1.00	-0.75	-0.38
d_6^{LCC}																					1.00	-0.44	-0.75	-1.00	-0.43	-0.74	-1.00	-0.23
d_4^{cos}																						1.00	0.74	0.44	1.00	0.74	0.44	0.47
d_5^{cos}																							1.00	0.75	0.73	1.00	0.75	0.38
d_6^{cos}																								1.00	0.43	0.74	1.00	0.23
d_4^{Evol}																									1.00	0.73	0.43	0.46
d_5^{Evol}																										1.00	0.74	0.38
d_6^{Evol}																											1.00	0.23
ΔG_{25}																												1.00

Tabelle A.6: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 30-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 30, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

50-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^E	d_5^E	d_6^E	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d_{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.20	0.10	-0.20	0.38	-0.34	-0.31	-0.29	0.06	0.04	-0.01	-0.02	0.02	0.05	0.01	-0.01	0.03	0.02	-0.06	-0.04	-0.03	0.05	0.03	0.03	0.05	0.03	0.03	0.15
H-Maß		1.00	0.65	-1.00	0.49	-0.50	-0.48	-0.47	0.09	0.05	0.09	0.01	-0.02	0.21	0.10	0.02	-0.01	0.04	-0.20	-0.10	-0.05	0.19	0.10	0.05	0.19	0.10	0.05	0.27
H-Distanz			1.00	-0.65	0.29	-0.29	-0.28	-0.28	0.04	0.04	0.04	-0.01	-0.03	0.12	0.04	-0.02	-0.03	0.00	-0.11	-0.08	-0.03	0.10	0.08	0.03	0.10	0.08	0.03	0.19
Homologie				1.00	-0.49	0.50	0.48	0.47	-0.09	-0.05	-0.09	-0.01	0.02	-0.21	-0.10	-0.02	0.01	-0.04	0.20	0.10	0.05	-0.19	-0.10	-0.05	-0.19	-0.10	-0.05	-0.27
Edit-Distanz					1.00	-0.96	-0.92	-0.88	0.20	0.19	0.19	0.11	0.07	0.32	0.18	0.10	0.07	0.13	-0.26	-0.16	-0.10	0.25	0.16	0.10	0.26	0.16	0.10	0.47
NW (1, -2)						1.00	0.99	0.96	-0.21	-0.22	-0.22	-0.13	-0.08	-0.37	-0.20	-0.12	-0.08	-0.15	0.27	0.16	0.10	-0.27	-0.16	-0.10	-0.28	-0.16	-0.10	-0.49
NW (1, -1)							1.00	0.99	-0.22	-0.23	-0.23	-0.14	-0.08	-0.39	-0.22	-0.13	-0.08	-0.15	0.28	0.15	0.10	-0.28	-0.15	-0.10	-0.28	-0.16	-0.10	-0.50
NW (2, -1)								1.00	-0.24	-0.25	-0.25	-0.15	-0.10	-0.41	-0.23	-0.14	-0.09	-0.17	0.28	0.16	0.10	-0.29	-0.16	-0.10	-0.29	-0.16	-0.10	-0.50
d^*									1.00	0.83	0.37	0.32	0.27	0.38	0.37	0.32	0.26	0.34	-0.40	-0.37	-0.30	0.41	0.37	0.30	0.41	0.37	0.30	0.28
d_1^{**}										1.00	0.40	0.34	0.29	0.42	0.40	0.33	0.28	0.37	-0.44	-0.41	-0.35	0.45	0.41	0.35	0.46	0.41	0.35	0.30
d_4^E											1.00	0.82	0.60	0.78	0.99	0.80	0.58	0.80	-0.72	-0.58	-0.42	0.81	0.60	0.43	0.81	0.61	0.43	0.35
d_5^E												1.00	0.81	0.58	0.81	0.99	0.79	0.93	-0.58	-0.71	-0.59	0.65	0.74	0.59	0.65	0.74	0.60	0.25
d_6^E													1.00	0.46	0.60	0.81	1.00	0.95	-0.42	-0.59	-0.73	0.48	0.61	0.74	0.47	0.61	0.74	0.17
$d^2(4,6)$														1.00	0.76	0.56	0.44	0.64	-0.61	-0.46	-0.36	0.67	0.47	0.36	0.68	0.48	0.36	0.38
d_4^{SE}															1.00	0.81	0.59	0.81	-0.72	-0.58	-0.43	0.81	0.60	0.43	0.82	0.61	0.43	0.34
d_5^{SE}																1.00	0.80	0.93	-0.58	-0.72	-0.60	0.65	0.74	0.60	0.65	0.75	0.60	0.24
d_6^{SE}																	1.00	0.94	-0.42	-0.59	-0.74	0.47	0.62	0.74	0.47	0.61	0.74	0.17
$d^{SE*}(4,6)$																		1.00	-0.58	-0.68	-0.70	0.66	0.71	0.70	0.65	0.71	0.70	0.26
d_4^{LCC}																			1.00	0.80	0.55	-0.99	-0.80	-0.55	-0.99	-0.80	-0.56	-0.46
d_5^{LCC}																				1.00	0.81	-0.80	-1.00	-0.81	-0.79	-1.00	-0.81	-0.34
d_6^{LCC}																					1.00	-0.56	-0.81	-1.00	-0.55	-0.80	-1.00	-0.28
d_4^{cos}																						1.00	0.80	0.56	1.00	0.80	0.56	0.45
d_5^{cos}																							1.00	0.81	0.79	1.00	0.81	0.34
d_6^{cos}																								1.00	0.55	0.80	1.00	0.28
d_4^{Evol}																									1.00	0.80	0.55	0.46
d_5^{Evol}																										1.00	0.80	0.34
d_6^{Evol}																											1.00	0.28
ΔG_{25}																												1.00

Tabelle A.7: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 50-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen der Länge 50, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

10-30-mere	Hamming	H-Maß	H-Distanz	Homologie	Edit-Distanz	NW (1, -2)	NW (1, -1)	NW (2, -1)	d^*	d_1^{**}	d_4^E	d_5^E	d_6^E	$d^2(4,6)$	d_4^{SE}	d_5^{SE}	d_6^{SE}	$d^{SE*}(4,6)$	d_4^{LCC}	d_5^{LCC}	d_6^{LCC}	d_4^{cos}	d_5^{cos}	d_6^{cos}	d_4^{Evol}	d_5^{Evol}	d_6^{Evol}	ΔG_{25}
Hamming	1.00	0.91	0.91	-0.44	0.87	-0.70	-0.41	0.02	0.22	0.15	0.66	0.70	0.70	0.66	0.08	0.08	0.09	0.09	0.02	0.00	-0.04	-0.15	-0.07	0.01	-0.16	-0.07	0.00	-0.29
H-Maß		1.00	0.98	-0.62	0.92	-0.77	-0.48	-0.04	0.28	0.23	0.67	0.70	0.71	0.70	0.08	0.08	0.10	0.09	-0.04	-0.05	-0.09	-0.09	-0.02	0.06	-0.10	-0.02	0.06	-0.25
H-Distanz			1.00	-0.56	0.90	-0.75	-0.46	-0.03	0.26	0.19	0.65	0.69	0.70	0.67	0.09	0.09	0.11	0.11	0.00	-0.02	-0.07	-0.12	-0.04	0.04	-0.13	-0.05	0.03	-0.27
Homologie				1.00	-0.51	0.67	0.70	0.58	-0.37	-0.27	-0.05	-0.01	0.00	-0.16	-0.19	-0.18	-0.19	-0.19	0.19	0.21	0.22	-0.21	-0.23	-0.22	-0.20	-0.22	-0.22	-0.24
Edit-Distanz					1.00	-0.90	-0.64	-0.19	0.29	0.26	0.70	0.71	0.71	0.75	0.08	0.07	0.08	0.08	-0.09	-0.07	-0.08	-0.04	0.00	0.05	-0.05	-0.01	0.04	-0.19
NW (1, -2)						1.00	0.91	0.59	-0.37	-0.30	-0.42	-0.38	-0.37	-0.52	-0.16	-0.13	-0.15	-0.15	0.17	0.14	0.11	-0.11	-0.11	-0.10	-0.11	-0.10	-0.10	-0.09
NW (1, -1)							1.00	0.87	-0.38	-0.29	-0.06	0.02	0.04	-0.21	-0.21	-0.18	-0.18	-0.19	0.21	0.18	0.12	-0.24	-0.20	-0.13	-0.24	-0.20	-0.13	-0.35
NW (2, -1)								1.00	-0.31	-0.20	0.37	0.47	0.50	0.21	-0.22	-0.19	-0.19	-0.19	0.21	0.19	0.11	-0.33	-0.26	-0.15	-0.34	-0.26	-0.15	-0.57
d^*									1.00	0.82	0.13	0.06	0.04	0.22	0.12	0.06	0.05	0.06	-0.34	-0.29	-0.19	0.34	0.29	0.19	0.34	0.29	0.19	0.20
d_1^{**}										1.00	0.25	0.17	0.14	0.33	0.06	-0.02	-0.04	-0.02	-0.44	-0.34	-0.23	0.41	0.33	0.22	0.41	0.33	0.22	0.21
d_4^E											1.00	0.96	0.93	0.94	0.06	-0.03	-0.06	-0.04	-0.26	-0.15	-0.14	0.08	0.06	0.09	0.08	0.05	0.09	-0.37
d_5^E												1.00	0.99	0.89	-0.04	-0.06	-0.07	-0.07	-0.10	-0.09	-0.11	-0.10	-0.02	0.06	-0.11	-0.03	0.06	-0.47
d_6^E													1.00	0.86	-0.07	-0.08	-0.08	-0.08	-0.03	0.00	-0.07	-0.18	-0.10	0.01	-0.19	-0.11	0.01	-0.52
$d^2(4,6)$														1.00	0.06	-0.02	-0.04	-0.02	-0.25	-0.15	-0.14	0.09	0.06	0.10	0.08	0.06	0.10	-0.27
d_4^{SE}															1.00	0.97	0.95	0.96	-0.23	-0.21	-0.16	0.26	0.22	0.16	0.26	0.22	0.17	0.13
d_5^{SE}																1.00	0.99	1.00	-0.08	-0.16	-0.13	0.11	0.18	0.14	0.11	0.18	0.14	0.08
d_6^{SE}																	1.00	1.00	-0.03	-0.10	-0.11	0.06	0.12	0.11	0.06	0.12	0.11	0.05
$d^{SE*}(4,6)$																		1.00	-0.05	-0.12	-0.11	0.08	0.13	0.12	0.08	0.13	0.12	0.06
d_4^{LCC}																			1.00	0.75	0.54	-0.98	-0.74	-0.54	-0.97	-0.74	-0.54	-0.39
d_5^{LCC}																				1.00	0.81	-0.74	-0.99	-0.81	-0.72	-0.99	-0.81	-0.32
d_6^{LCC}																					1.00	-0.53	-0.80	-1.00	-0.50	-0.79	-1.00	-0.19
d_4^{cos}																						1.00	0.75	0.54	1.00	0.76	0.54	0.50
d_5^{cos}																							1.00	0.81	0.73	1.00	0.81	0.38
d_6^{cos}																								1.00	0.51	0.80	1.00	0.22
d_4^{Evol}																									1.00	0.74	0.52	0.51
d_5^{Evol}																										1.00	0.80	0.39
d_6^{Evol}																											1.00	0.22
ΔG_{25}																												1.00

Tabelle A.8: Korrelationskoeffizienten der Distanzmaßuntersuchung aus Abschnitt 4.2 für 10-30-mere. Gezeigt werden die linearen Korrelationskoeffizenten aller Paare von Distanzmaßen gemessen über 500 Paare von Sequenzen mit Längen zwischen 10 und 30, sowie die Korrelation aller Distanzmaße mit der freien Enthalpie ΔG_{25}.

Anhang B

Ein-/Ausgabedateien für den DNA-Würfel

```
SEQUENCETYPE arm {
  length = 11;
  GC_ratio = [0;1];
  Tm = [25;30]; }

SEQUENCETYPE se {
  length = 8;
  GC_ratio = 0.5; }

arm a1, a2, b1, b2, c1, c2, d1, d2,
  e1, e2, f1, f2, g1, g2, h1, h2,
  i1, i2, k1, k2, l1, l2, m1, m2;

se A, B, C, D, E, F, G, H, I, K,
  L, M;

MACRO_3WAYJUNCTION j1 {
  arm1 = a1;
  arm2 = b1;
  arm3 = c1;
  sticky_end1 = A;
  sticky_end2 = B;
  sticky_end3 = C;
  }

MACRO_3WAYJUNCTION j2 {
  arm1 = b2;
  arm2 = m1;
  arm3 = d1;
  sticky_end1 = B;
  orientation1 = 1;
  sticky_end2 = M;
  sticky_end3 = D;
  }

MACRO_3WAYJUNCTION j3 {
  arm1 = d2;
  arm2 = k1;
  arm3 = e1;
  sticky_end1 = D;
  orientation1 = 1;
  sticky_end2 = K;
  sticky_end3 = E;
  }

MACRO_3WAYJUNCTION j4 {
  arm1 = c2;
  arm2 = e2;
  arm3 = f1;
  sticky_end1 = C;
  orientation1 = 1;
  sticky_end2 = E;
  orientation2 = 1;
  sticky_end3 = F;
  }

MACRO_3WAYJUNCTION j5 {
  arm1 = g1;
  arm2 = f2;
  arm3 = i1;
  sticky_end1 = G;
  sticky_end2 = F;
  orientation2 = 1;
  sticky_end3 = I;
  }

MACRO_3WAYJUNCTION j6 {
  arm1 = a2;
  arm2 = g2;
  arm3 = l1;
  sticky_end1 = A;
  orientation1 = 1;
  sticky_end2 = G;
  orientation2 = 1;
  sticky_end3 = L;
  }

MACRO_3WAYJUNCTION j7 {
  arm1 = l2;
  arm2 = h1;
  arm3 = m2;
  sticky_end1 = L;
  orientation1 = 1;
  sticky_end2 = H;
  sticky_end3 = M;
  orientation3 = 1;
  }

MACRO_3WAYJUNCTION j8 {
  arm1 = i2;
  arm2 = k2;
  arm3 = h2;
  sticky_end1 = I;
  orientation1 = 1;
  sticky_end2 = K;
  orientation2 = 1;
  sticky_end3 = H;
  orientation3 = 1;
  }


POOL p {
  N_uniqueness = 6;
  Violation_tolerance = 0;
  Sample_conc = 2e-7;
  Na_conc = 0.05;
  Tm_method = NNSantaLucia;
}
```

Abbildung B.1: Eingabedatei für den DNA-Würfel aus Abschnitt 8.4. In der Beschreibungssprache DeLaNA werden die acht Junctions, ihre Arme und die sticky Ends definiert. Sticky Ends mit `orientation = 1` sind komplementär zur angegebenen Sequenz.

```
// File name: cube_out.dln
// Created by CANADA v0.1.1

SEQUENCETYPE arm {
  NA_type = DNA;
  length = 11;
  GC_ratio = [0;1];
  Tm = [25;30];
  seq_mask = "";
}

SEQUENCETYPE se {
  NA_type = DNA;
  length = 8;
  GC_ratio = 0.5;
  Tm = [0;100];
  seq_mask = "";
}

SEQUENCE a1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 29.0136;
  seq_mask = "cgtcgatgcta";
}

SEQUENCE a2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 25.3599;
  seq_mask = "tgtagccaaac";
}

SEQUENCE b1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 25.8801;
  seq_mask = "tactatcgcgt";
}

SEQUENCE b2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 27.7208;
  seq_mask = "tgacgaaatgc";
}

SEQUENCE c1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.3784;
  seq_mask = "gaccgtcctat";
}

SEQUENCE c2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 29.0611;
  seq_mask = "caagacaagcc";
}

SEQUENCE d1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.4493;
  seq_mask = "cgagcaggtta";
}

SEQUENCE d2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 27.4005;
  seq_mask = "caaggggcata";
}

SEQUENCE e1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 27.3757;
  seq_mask = "ttacctcagcc";
}

SEQUENCE e2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.3505;
  seq_mask = "gcctccgaata";
}

SEQUENCE f1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.6692;
  seq_mask = "agagatgagcg";
}

SEQUENCE f2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 29.705;
  seq_mask = "gaactccacga";
}
```

Abbildung B.2: Ausgabedatei für den DNA-Würfel aus Abschnitt 8.4. In der Beschreibungssprache DeLaNA werden die vom DNA-Sequence-Compiler gefundenen Sequenzen mit ihren Eigenschaften angegeben. Die Darstellung der Datei wird in den Abbildungen B.3 bis B.5 fortgesetzt.

```
SEQUENCE g1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 26.7616;
  seq_mask = "ttaaactgccg";
}

SEQUENCE g2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 27.4892;
  seq_mask = "gggattaaggc";
}

SEQUENCE h1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 29.9132;
  seq_mask = "ctaattgcgcc";
}

SEQUENCE h2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 26.8849;
  seq_mask = "tgattgcttgc";
}

SEQUENCE i1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 25.4282;
  seq_mask = "ccgttctctag";
}

SEQUENCE i2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 26.592;
  seq_mask = "caccggctata";
}

SEQUENCE k1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 26.5671;
  seq_mask = "attccgagtga";
}

SEQUENCE k2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 29.1065;
  seq_mask = "gcgcgtaatag";
}

SEQUENCE l1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.3467;
  seq_mask = "cgaactgggta";
}

SEQUENCE l2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.2432;
  seq_mask = "gcgtagagttg";
}

SEQUENCE m1 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.545455;
  Tm = 28.6148;
  seq_mask = "ggacatgcttc";
}

SEQUENCE m2 {
  NA_type = DNA;
  length = 11;
  GC_ratio = 0.454545;
  Tm = 25.1696;
  seq_mask = "aggagatacgt";
}
```

Abbildung B.3: Ausgabedatei für den DNA-Würfel aus Abschnitt 8.4 (Fortsetzung von Abbildung B.2).

```
SEQUENCE A {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 4.93989;
            seq_mask = "ctcccttt";
}

SEQUENCE B {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 7.29005;
            seq_mask = "tcgacttc";
}

SEQUENCE C {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 6.95945;
            seq_mask = "attctccg";
}

SEQUENCE D {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 10.381;
            seq_mask = "gaaatcgc";
}

SEQUENCE E {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 6.85551;
            seq_mask = "ccaccatt";
}

SEQUENCE F {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 1.01721;
            seq_mask = "cacccata";
}

SEQUENCE G {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 6.75561;
            seq_mask = "ccgtatgt";
}

SEQUENCE H {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 5.5678;
            seq_mask = "cgctaatg";
}

SEQUENCE I {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 2.12562;
            seq_mask = "gagtctct";
}

SEQUENCE K {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 2.9053;
            seq_mask = "gctatcac";
}

SEQUENCE L {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 1.93717;
            seq_mask = "gtattccc";
}

SEQUENCE M {
            NA_type = DNA;
            length = 8;
            GC_ratio = 0.5;
            Tm = 4.82968;
            seq_mask = "tcagagca";
}

MACRO_3WAYJUNCTION j1 {
            Sticky_end1 = A;
            Orientation1 = 0;
            Sticky_end2 = B;
            Orientation2 = 0;
            Sticky_end3 = C;
            Orientation3 = 0;
            Arm1 = a1;
            Arm2 = b1;
            Arm3 = c1;
}
```

Abbildung B.4: Ausgabedatei für den DNA-Würfel aus Abschnitt 8.4 (Fortsetzung von Abbildung B.3).

```
MACRO_3WAYJUNCTION j2 {
  Sticky_end1 = B;
  Orientation1 = 1;
  Sticky_end2 = M;
  Orientation2 = 0;
  Sticky_end3 = D;
  Orientation3 = 0;
  Arm1 = b2;
  Arm2 = m1;
  Arm3 = d1;
}

MACRO_3WAYJUNCTION j3 {
  Sticky_end1 = D;
  Orientation1 = 1;
  Sticky_end2 = K;
  Orientation2 = 0;
  Sticky_end3 = E;
  Orientation3 = 0;
  Arm1 = d2;
  Arm2 = k1;
  Arm3 = e1;
}

MACRO_3WAYJUNCTION j4 {
  Sticky_end1 = C;
  Orientation1 = 1;
  Sticky_end2 = E;
  Orientation2 = 1;
  Sticky_end3 = F;
  Orientation3 = 0;
  Arm1 = c2;
  Arm2 = e2;
  Arm3 = f1;
}

MACRO_3WAYJUNCTION j5 {
  Sticky_end1 = G;
  Orientation1 = 0;
  Sticky_end2 = F;
  Orientation2 = 1;
  Sticky_end3 = I;
  Orientation3 = 0;
  Arm1 = g1;
  Arm2 = f2;
  Arm3 = i1;
}

MACRO_3WAYJUNCTION j6 {
  Sticky_end1 = A;
  Orientation1 = 1;
  Sticky_end2 = G;
  Orientation2 = 1;
  Sticky_end3 = L;
  Orientation3 = 0;
  Arm1 = a2;
  Arm2 = g2;
  Arm3 = l1;
}

MACRO_3WAYJUNCTION j7 {
  Sticky_end1 = L;
  Orientation1 = 1;
  Sticky_end2 = H;
  Orientation2 = 0;
  Sticky_end3 = M;
  Orientation3 = 1;
  Arm1 = l2;
  Arm2 = h1;
  Arm3 = m2;
}

MACRO_3WAYJUNCTION j8 {
  Sticky_end1 = I;
  Orientation1 = 1;
  Sticky_end2 = K;
  Orientation2 = 1;
  Sticky_end3 = H;
  Orientation3 = 1;
  Arm1 = i2;
  Arm2 = k2;
  Arm3 = h2;
}

POOL p {
  sequences = ;
  n_uniqueness = 6;
  Hamming = 0;
  H_distance = 0;
  sample_conc = 2e-007;
  Na_conc = 0.05;
  formamide_conc = 0;
}
```

Abbildung B.5: Ausgabedatei für den DNA-Würfel aus Abschnitt 8.4 (Fortsetzung von Abbildung B.4).

```
j1
aaagggagtagcatcgacg--tactatcgcgt
        atcgtagctgc  atgatagcgcaagctgaag
                    cg
                    ta
                    gc
                    gc
                    cg
                    at
                    gc
                    gc
                    at
                    ta
                    at
                    t
                    a
                    a
                    g
                    a
                    g
                    g
                    c
j2
tcgacttcgcatttcgtca--ggacatgcttc
        cgtaaagcagt  cctgtacgaagagtctcgt
                    gc
                    cg
                    ta
                    cg
                    gc
                    ta
                    cg
                    cg
                    at
                    at
                    ta
                    c
                    t
                    t
                    t
                    a
                    g
                    c
                    g
```

```
j3
gaaatcgctatgccccttg--attccgagtga
        atacggggaac  taaggctcactcgatagtg
                    at
                    at
                    ta
                    gc
                    gc
                    at
                    gc
                    ta
                    cg
                    gc
                    gc
                    g
                    g
                    t
                    g
                    g
                    t
                    a
                    a
j4
attctccgggcttgtcttg--gcctccgaata
        ccgaacagaac  cggaggcttatttaccacc
                    ta
                    cg
                    ta
                    cg
                    ta
                    at
                    cg
                    ta
                    cg
                    gc
                    cg
                    g
                    t
                    g
                    g
                    g
                    t
                    a
                    t
```

Abbildung B.6: Junctions für den DNA-Würfel aus Abschnitt 8.4. Die gezeigten dreiarmigen Junctions bilden die Ecken des Würfels und sind aus den Armsequenzen und sticky Ends aus den Abbildungen B.2 bis B.4 zusammengesetzt. Die Striche `--` in der jeweils oberen Sequenz markieren, daß dies ein durchgehender Strang ist. Die Darstellung wird in Abbildung B.7 fortgesetzt.

```
j5
acatacggcggcagtttaa--gaactccacga
        gccgtcaaatt  cttgaggtgctatacccac
                    gc
                    gc
                    cg
                    at
                    at
                    gc
                    at
                    gc
                    at
                    ta
                    cg
                    c
                    t
                    c
                    a
                    g
                    a
                    g
                    a
j6
ctccctttgtttggctaca--gggattaaggc
        caaaccgatgt  ccctaattccgtgtatgcc
                    gc
                    cg
                    ta
                    ta
                    gc
                    at
                    cg
                    cg
                    cg
                    at
                    ta
                    c
                    a
                    t
                    a
                    a
                    g
                    g
                    g
```

```
j7
gtattccccaactctacgc--ctaattgcgcc
        gttgagatgcg  gattaacgcgggcgattac
                    ta
                    cg
                    cg
                    ta
                    cg
                    ta
                    at
                    ta
                    gc
                    cg
                    at
                    a
                    c
                    g
                    a
                    g
                    a
                    c
                    t
j8
gagtctcttatagccggtg--gcgcgtaatag
        atatcggccac  cgcgcattatccactatcg
                    at
                    cg
                    ta
                    at
                    at
                    cg
                    gc
                    at
                    at
                    cg
                    gc
                    g
                    t
                    a
                    a
                    t
                    c
                    g
                    c
```

Abbildung B.7: Junctions für den DNA-Würfel aus Abschnitt 8.4 (Fortsetzung von Abbildung B.6).

Index