Ein Sequenzdesign-Algorithmus für verzweigte DNA-Strukturen

DISSERTATION

zur Erlangung des Akademischen Grades Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

Dipl.-Inf. Jan Seiffert

geboren am 7. Februar 1977 in Borna

Gutachter:

Prof. Dr. Erwin Stoschek	Institut für Systemarchitektur, TU Dresden (emeriti)
Prof. Dr. Wolfgang Pompe	Institut für Werkstoffwissenschaft, TU Dresden (emeriti)
PD Dr. Peter Dittrich	Biologisch-Pharmazeutische Fakultät, Friedrich-Schiller-Universität Jena

eingereicht am 26. November 2007
verteidigt am 7. November 2008

Dresden, den 26. November 2008

Danksagung

Ich danke allen, die zu dieser Arbeit beigetragen haben. Besonderer Dank geht an meine beiden Doktorväter Prof. Stoschek und Prof. Pompe; an Stefan Hecker und Lenore Keschka, die für mich die Bilder erstellt haben; an die Korrekturleser Kerstin Krüger, Bianca Franz und Jens Seiffert sowie an meine Arbeitskollegen, vor allem an Robert Müller, Nicola Seriani, Martin Mkandawire, Alexander Huhle und Michael Mertig.

Vielen Dank an den Freistaat Sachsen, der mich mit einem Stipendium unterstützt hat.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Nanotechnologie ist ein sehr aktuelles Forschungsgebiet, das nicht nur im Fokus der wissenschaftlichen Fachwelt steht, sondern zunehmend auch die öffentliche Wahrnehmung erreicht. Das Forschungsgebiet ist sehr breit gefächert und reicht von der Materialwissenschaft über Computertechnik bis hin zur Molekularbiologie und Medizin. Gemeinsam ist allen Teilgebieten, dass sich die entscheidenden Effekte im Nanometerbereich abspielen. Viele chemische Elemente oder Verbindungen verändern im Nanometerbereich ihre Eigenschaften. Ein Beispiel dafür ist Gold. Es ist normalerweise sehr reaktionsträge und geht daher kaum Bindungen mit anderen Stoffen ein. Das trägt erheblich zu seinem allgemeinen Wert bei, macht es jedoch andererseits für die Chemie uninteressant. Im Gegensatz dazu wird ein Goldcluster mit einer Größe von nur wenigen Nanometern viel reaktionsfreudiger, kann andere Elemente an sich binden und dadurch zum Beispiel auch als Katalysator wirken. Dabei sind Fragestellungen gegeben wie: Welche Clustergrößen und -anordnungen ziehen welche Effekte nach sich? und: Wie können solche Größen und Arrangements erzeugt werden?

In der Schaltkreisproduktion wird schon seit längerer Zeit im Nanometerbereich gearbeitet. Zur Zeit geht man bereits auf Strukturgrößen unter 30 Nanometer. Über welche Techniken lässt sich diese Miniaturisierung und damit einherschreitende Leistungssteigerung fortsetzen?

Der Nanometerbereich ist auch die Welt der großen Biomoleküle, wie Proteine, Enzyme und Nukleinsäuren. Die Erforschung der Strukturen und Funktionsweisen dieser Moleküle bringt nicht nur ein besseres Verständnis der biochemischen Abläufe in lebenden Organismen, sondern auch die Möglichkeit der molekularbiologischen Stoffsynthese und molekularen Konstruktion mit sich.

Möchte man Strukturen im Nanometerbereich gezielt beeinflussen bzw. erzeugen, kann man nicht einfach auf die Konstruktionsprinzipien unserer makroskopischen Welt zurückgreifen. Ein Würfel mit einem Zentimeter Kantenlänge lässt sich sehr leicht durch Sägen, Schleifen oder Gießen herstellen. Bei einer Kantenlänge von zehn oder fünf Nanometern ist das ungleich schwieriger, da die notwendigen Werkzeuge nicht zur Verfügung stehen und andere chemische und physikalische Kräfte zu beachten sind. Die Materialien lassen sich nicht mehr einfach von außen in eine gewünschte Form bringen. Das Top-Down genannte Konstruktionsprinzip stößt hier an seine Grenzen. Deshalb greift man verstärkt auf die umgekehrte Vorgehensweise zurück. Es werden größere Einheiten durch Selbstorganisation der kleineren Bauelemente gebaut. Dieses Konstruktionsprinzip wird Bottom-Up

genannt.

Für das Bottom-Up-Prinzip eignet sich besonders gut die Desoxyribonukleinsäure (DNA). Die DNA ist ein kettenförmiges Molekül, dessen Grundelemente – die Nukleotide – neben ihrer Bindung in der Kette zusätzlich noch gezielt mit Nukleotiden anderer DNA-Moleküle eine Bindung eingehen können. Bei geeigneten Bedingungen geschieht das automatisch. Die Anordnung der Nukleotide in den beteiligten Molekülen legt dabei fest, welche Moleküle sich wie miteinander verbinden. Die Struktur der Bauelemente bestimmt also die Gestalt des Gesamtkonstruktes. Durch die gezielte Festlegung einzelner Nukleotidsequenzen kann so die gewünschte DNA-Zielstruktur vorbestimmt werden. Unterschiedlichste DNA-Strukturen wurden bereits geplant und erzeugt. So gibt es einfache und komplexere Verzweigungen, die als Bauelemente dienen, Netze und auch dreidimensionale Objekte wie Tetraeder, Würfel oder Röhren. Sogar bewegliche Nanomaschinen wurden schon aus DNA hergestellt. Die Strukturen dienen dann oft als Grundgerüste für weitergehende Arrangements, zum Beispiel eine bestimmte Anordnung von Proteinen und Metallclustern oder einem Nanodraht.

Beim Design einer Zielstruktur und den dazu passenden Nukleotidsequenzen sind bestimmte Anforderungen und Randbedingungen zu beachten. Deren Zusammenspiel führt im Falle des Sequenzdesigns zu einer so hohen Komplexität, dass zu deren Lösung auf die Unterstützung durch Computer zurückgegriffen werden muss.

Es existieren bereits mehrere Sequenzdesign-Algorithmen mit dazugehörigen Software-Programmen. Sie sind jedoch entweder nicht für verzweigte DNA-Strukturen geeignet oder, da nur teilautomatisiert, sehr arbeitsaufwendig für den Benutzer. Obwohl trotz dieser Einschränkungen bereits große Erfolge in der DNA-Nanotechnologie zu verzeichnen sind, ist es wünschenswert, einen schnellen und vollautomatischen Algorithmus und eine Implementierung dessen zu besitzen. Dies würde die Konstrukteure von DNA-Netzwerkstrukturen entlasten und somit mehr Raum für andere Tätigkeiten bereitstellen.

Die vorliegende Dissertation widmet sich genau diesem Problem. Ihr Ziel ist es, einen vollautomatischen Sequenzdesign-Algorithmus samt Implementierung zur Verfügung zu stellen, der bei der Konstruktion beliebiger DNA-Strukturen Verwendung finden kann. Als Einführung wird zuerst das DNA-Molekül selbst und dessen Potential als Werkstoff vorgestellt. Danach werden die Anforderungen an die Nukleotidsequenzen formuliert und schließlich wird ein entsprechender Algorithmus und seine Implementierung vorgestellt.

Kapitel 2

DNA als Werkstoff

2.1 Das DNA-Molekül

2.1.1 Nukleotide

DNA (desoxyribonucleic acid, engl. für Desoxyribonukleinsäure) ist ein Polymer aus Nukleotiden. Jedes Nukleotid besteht aus einer Pentose, einer Nukleobase und einer Phosphatgruppe (siehe Abb. 2.1). Pentose und Nukleobase ohne Phosphatgruppe nennt man Nukleosid. Die Pentose ist eine 2-Desoxyribose in Ringform (siehe Abb. 2.2). Am C1-Atom der Pentose hängt die Nukleobase. Die Phosphatgruppe ist mit dem C5-Atom verknüpft. Das C3-Atom trägt eine OH-Gruppe, an der weitere Bindungen stattfinden können.

In der DNA gibt es vier verschiedene Nukleobasen: Guanin und Adenin, welche Purin-Derivate sind, sowie die Pyrimidin-Derivate Thymin und Cytosin (siehe Abb. 2.3). Es gibt noch eine fünfte Nukleobase: das Uracil, welches im Schwestermolekül der DNA, der RNA (ribonucleic acid, engl. für Ribonukleinsäure), das Thymin ersetzt.

Die Phosphatgruppe am C5-Atom der Pentose kann ein Monophosphat, ein Diphosphat oder auch ein Triphosphat sein (siehe Abb. 2.4).

Es gibt also jeweils vier Nukleoside, an welche unterschiedliche Phosphatgruppen gebunden sein können. Tabelle 2.1 listet die Namen und Abkürzungen aller Nukleoside mit Mono-, Di- und Triphosphaten auf. Sehr bekannt sind zusätzlich zwei Nukleosidphosphate der RNA: das Adenosintriphosphat (ATP) und das Adenosindiphosphat (ADP). Beide spielen eine wichtige Rolle im Energiehaushalt von Organismen. ATP wird als Energielieferant für einige molekularbiologische Reaktionen benutzt (siehe Kap. 2.3).

2.1.2 Einzelstränge

Ein einzelsträngiges DNA-Molekül entsteht durch Verkettung von Nukleotiden mit Monophosphatgruppen (Desoxynukleosidmonophosphate, dNMP). Die Phosphatgruppen gehen dabei jeweils mit dem C5-Atom der einen und dem C3-Atom der nächsten Desoxyribose eine Phosphoresterbindung (C—O—P) ein. Zwei benachbarte Nukleoside sind also kovalent durch eine Phosphordiesterbindung verknüpft. Es entsteht ein Strangrückgrat aus alternierenden Desoxyribose-Ringen und Phosphatgruppen, von dem die Nukleobasen nach außen abstehen. Diese grundlegende Struktur des DNA-Moleküls als eine Kette aus Nukleotiden ist die Primär-Struktur der DNA.

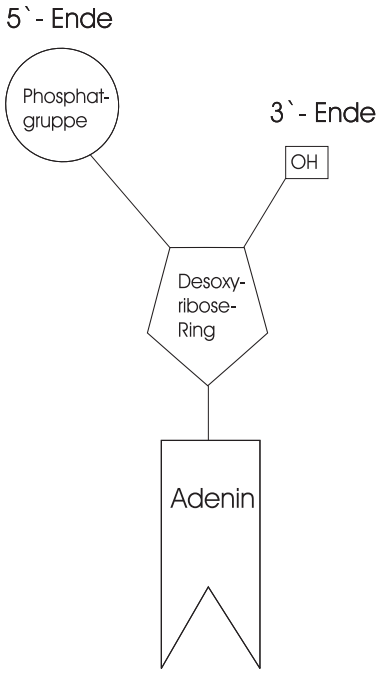


Abbildung 2.1: Struktur eines Nukleotids

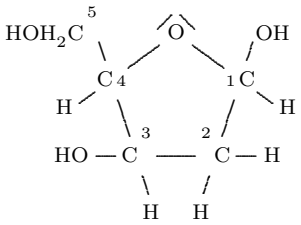
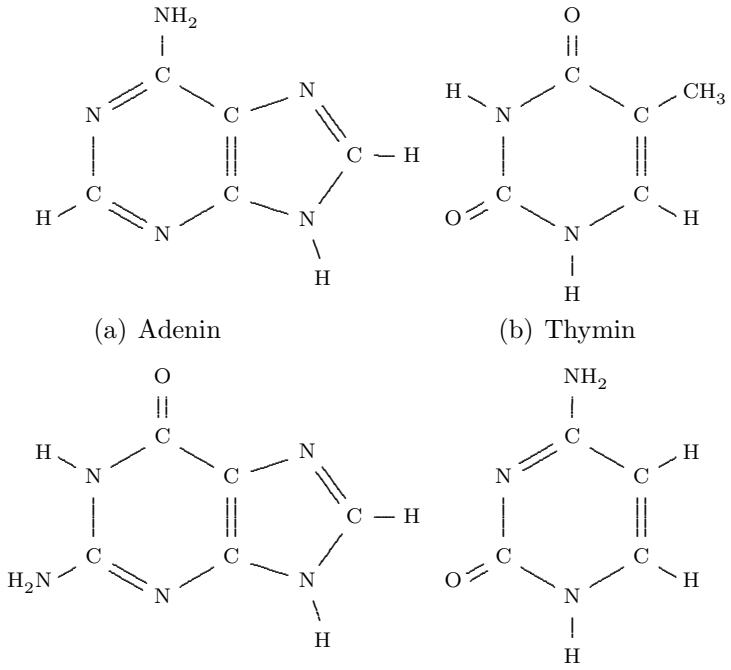


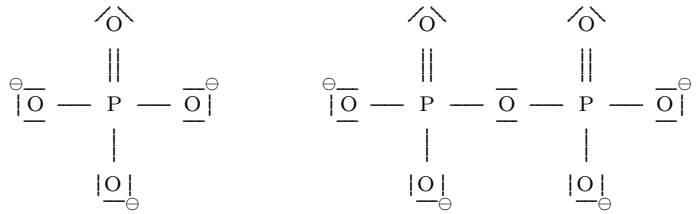
Abbildung 2.2: Strukturformel der Desoxyribose



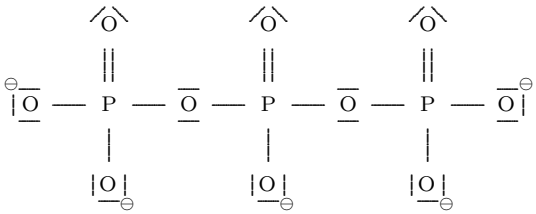
(a) Adenin (b) Thymin

(c) Guanin (d) Cytosin

Abbildung 2.3: Strukturformeln der Nukleobasen



(a) Monophosphat (b) Diphosphat



(c) Triphosphat

Abbildung 2.4: Strukturformeln der Phosphatgruppen

Nukleobase	Nukleosid	Nukleotide
Adenin (A)	Desoxyadenosin	Desoxyadenosinmonophosphat (dAMP) Desoxyadenosindiphosphat (dADP) Desoxyadenosintriphosphat (dATP)
Thymin (T)	Desoxythymidin	Desoxythymidinmonophosphat (dTMP) Desoxythymidindiphosphat (dTDP) Desoxythymidintriphosphat (dTTP)
Guanin (G)	Desoxyguanosin	Desoxyguanosinmonophosphat (dGMP) Desoxyguanosindiphosphat (dGDP) Desoxyguanosintriphosphat (dGTP)
Cytosin (C)	Desoxycytidin	Desoxycytidinmonophosphat (dCMP) Desoxycytidindiphosphat (dCDP) Desoxycytidintriphosphat (dCTP)

Nukleoside (= Pentose + Nukleobase)
Nukleotide (= Pentose + Nukleobase + Phosphatgruppe)

Tabelle 2.1: Namen der Nukleobasen der DNA

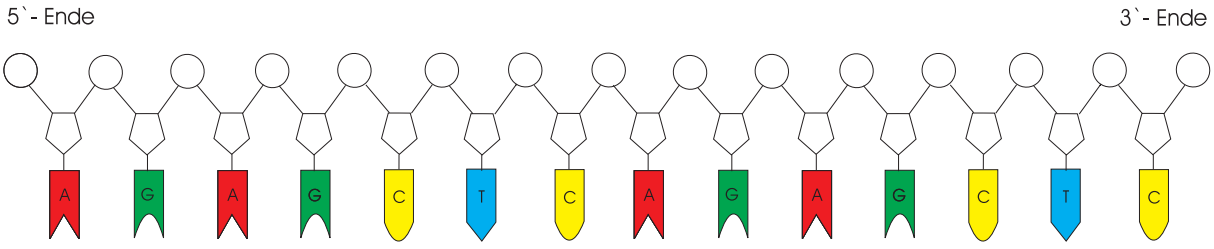


Abbildung 2.5: Struktur eines DNA-Einzelstrang-Moleküls

Ein DNA-Strang hat zwei unterschiedliche Enden. An einem Ende liegt das C3-Atom der Desoxyribose des äußersten Nukleotids. Am entgegengesetzten Ende liegt das C5-Atom der Desoxyribose des anderen äußersten Nukleotids. Die Basensequenz kann also in zwei möglichen Leserichtungen angegeben werden, entweder vom sogenannten 3'-Ende zum sogenannten 5'-Ende oder umgekehrt. Wenn nicht anders angegeben, wird in dieser Arbeit die übliche Auflistung vom 5'- zum 3'-Ende benutzt.

In jeder Phosphatgruppe ist ein Sauerstoffatom (O) nur einfach gebunden, wodurch die Gruppe eine negative Ladung besitzt. Im gesamten Strang sind diese Ladungen gleich orientiert. Das Molekül wird dadurch hochpolar. Dies kann, wie wir später sehen werden, für Analyse- und Positionierungszwecke benutzt werden.

2.1.3 Basenpaare und Doppelstränge

Jeweils eine Purin- und eine Pyrimidin-Base können über Wasserstoffbrücken ein Basenpaar bilden. Aus geometrischen Gründen bindet Guanin mit Cytosin über drei Wasserstoffbrücken und Adenin mit Thymin über zwei Wasserstoffbrücken. Dies sind die sogenannten Watson-Crick-Basenpaare. Die Guanin-Cytosin-Bindung ist wegen der höheren Zahl an Wasserstoffbrücken stärker als die Adenin-Thymin-Bindung.

Zwei Einzelstränge oder Einzelstrangabschnitte mit komplementären Basensequenzen können sich über die Basenpaarungen zu einem Doppelstrang verbinden. Komplementär sind die Sequenzen dann, wenn die eine Sequenz in umgekehrter Leserichtung die Watson-Crick-Partner der anderen Sequenz enthält. Zum Beispiel ist der Strang $5' - \mathrm{TTCTCGGA} - 3'$ komplementär zu $5' - \mathrm{TCCGAGAA} - 3'$ und beide können den Doppelstrang

$$\begin{aligned} &5' - \mathrm{TTCTCGGA} - 3' \\ &3' - \mathrm{AAGAGCCT} - 5' \end{aligned}$$

bilden. Ein Strang besitzt eine selbstkomplementäre Sequenz, wenn er mit einer Kopie seiner selbst eine Bindung eingehen kann. Dies ist der Fall bei der Sequenz $5'-\mathrm{ACGATCGT}-3'$, denn zwei Stränge mit dieser Sequenz können sich verbinden und den Doppelstrang

$$\begin{aligned} &5' - \mathrm{ACGATCGT} - 3' \\ &3' - \mathrm{TGCTAGCA} - 5' \end{aligned}$$

formen.

Die Basenpaar-Bildung legt die Sekundär-Struktur eines DNA-Moleküls fest. Natürlicherweise ist die Sekundär-Struktur so ausgelegt, dass zwei lange Einzelstränge komplett zu einem Doppelstrang verbunden sind. Doch auch in der Natur gibt es zeitweise, zum Beispiel während der Zellteilung oder der Genexpression, Abweichungen davon. Ein großer Teil der DNA-Nanotechnologie besteht darin, ebenfalls abweichende, aber stabile Sekundärstrukturen zu finden, die sich für nützliche oder interessante Anwendungen nutzen lassen.

2.1.4 Helix-Konformationen

Aufgrund der Molekülgeometrie und der Eigenschaften des Lösungsmittels bleibt ein DNA-Doppelstrang nicht planar, sondern verdreht sich zur allseits bekannten Doppelhelix, bei der die beiden Strangrückgrate nach außen zeigen und die Basenpaare im

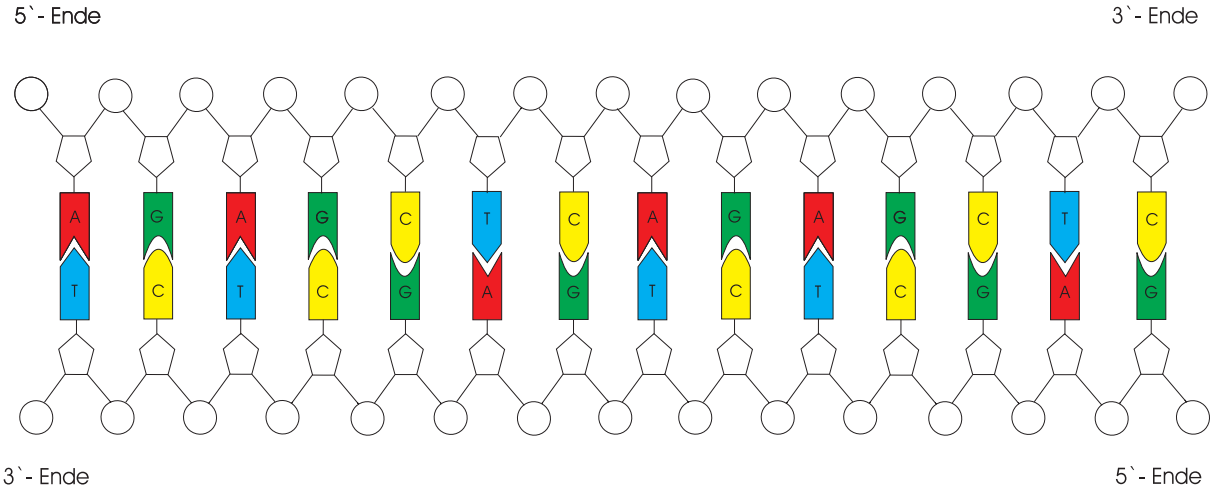


Abbildung 2.6: Struktur eines DNA-Doppelstrang-Moleküls

	B-Form	A-Form	Z-Form
Drehrichtung	rechtsgängig	rechtsgängig	linksgängig
Helixdurchmesser	23.7 Å	25.5 Å	18.4 Å
Windungshöhe	35.4 Å, 10.4 bp	25.3 Å, 11 bp	45.6 Å, 12 bp
Basenpaar-Neigung	1^o	19^o	9^o
Vorkommen	in normaler Lösung	in Lösungsmitteln mit geringer Polarität	in Salzlösungen mit hoher Konzentration, GC-Sequenzen in Alkohollösung

Die Zahlenwerte wurden übernommen von R. Rohs [29].

Tabelle 2.2: Merkmale verschiedener Doppelhelix-Konformationen

Inneren übereinander gestapelt sind. Diese Anordnung ist sehr stabil und im Gegensatz zum Einzelstrang bis zu einer gewissen Länge auch relativ unflexibel. Man kann ein Doppelstrang-Molekül bis zu einer Länge von 150 Basenpaaren als starr oder persistent betrachten, weshalb diese Länge als Persistenzlänge bezeichnet wird. Die Starrheit hat eine hohe Bedeutung für den Einsatz der DNA als Werkstoff. Es ist dadurch möglich, gerüstartige Strukturen zu schaffen.

Die genaue Form der Helix (Windungshöhe, Drehrichtung etc.) wird vor allem durch die Eigenschaften der Lösung, in welcher sich das Molekül befindet, aber auch durch die Basensequenz bestimmt. Man hat mehrere DNA-Konformationen entdeckt, von denen die drei häufigsten in Tabelle 2.2 aufgeführt sind (siehe dazu auch [29]).

Die gängigste und auch in der Natur vorherrschende Form ist die sogenannte B-Form. Sie ist eine rechtsgängige Helix mit einer Windungshöhe von durchschnittlich 10.4 Basenpaaren oder 35.4 Å und einem Helixdurchmesser von 23.7 Å. Die meisten der später beschriebenen DNA-Strukturen basieren auf dieser Konformation. Verschiedene Konformationen können ebenfalls als Konstruktionselemente benutzt werden. So kann zum Beispiel dieselbe Helix unter verschiedenen Bedingungen die B-Form und die Z-Form annehmen,

was zur Konstruktion von Nanoapparaten genutzt werden kann [37].

2.2 Schmelztemperatur von DNA-Doppelsträngen

Wie später aufgezeigt wird, besteht ein großer Teil der DNA-Nanotechnologie darin, aus Einzelsträngen durch Doppelstrangbildung größere Moleküle zu erzeugen. Dazu muss man abschätzen können, ob Einzelstränge unter bestimmten Bedingungen in einer Lösung einzeln oder gebunden vorliegen. Der Schmelzpunkt hängt von verschiedenen Faktoren ab. Wesentlich sind die Temperatur, aber auch die Salzkonzentration der Lösung sowie die Länge und die Basenfolge der Sequenz. Meist ist dabei die Temperatur die zu bestimmende Größe bei gegebenen Randbedingungen. Im Folgenden werden zwei Methoden zur Bestimmung der Schmelztemperatur von DNA-Doppelsträngen vorgestellt.

2.2.1 GC-Formel

Die GC-Formel bietet eine sehr einfache Methode zur Bestimmung der Schmelztemperatur eines Doppelstranges. Sie stützt sich auf den G/C-Basenpaaranteil der Strangsequenz. Daher auch ihr Name. Zusätzlich gibt es Korrekturterme, die den Einfluss der Salzkonzentration in der Lösung beschreiben. Für unterschiedliche Stranglängen besitzt die GC-Formel unterschiedliche Varianten. Sie lautet für bis zu 13 Basenpaare

$$T_M = 2 \cdot \#\mathrm{AT} + 4 \cdot \#\mathrm{GC}^o C \tag{2.1}$$

[53] und ab 14 Basenpaaren

$$T_m = 64.9 + \frac{41 \cdot \#\mathrm{GC} - 672.4^o}{l} C, \tag{2.2}$$

[50] wobei jeweils l die Stranglänge in Basenpaaren, #AT die Anzahl der A/T-Basenpaare und #GC die Anzahl der G/C-Basenpaare sind. Es wird dabei angenommen, dass die Strangkonzentration 50 nM und die Salzkonzentration 50 mM bei einem pH-Wert von 7 betragen.

Für abweichende Salzkonzentrationen werden beide Formeln verändert. Für bis zu 13 Basenpaare lautet sie dann

$$T_M = 2 \cdot \#\mathrm{AT} + 4 \cdot \#\mathrm{GC} + 16.6 \cdot \log_{10}(\frac{[\mathrm{Na}^+]}{0.05\mathrm{M}})^o C \tag{2.3}$$

[38] und ab 14 Basenpaaren

$$T_m = 78.9 + \frac{41 \cdot \#\mathrm{GC} - 820}{l} + 16.6 \cdot \log_{10}(\frac{[\mathrm{Na}^+]}{0.05\mathrm{M}})^o C \tag{2.4}$$

[51]. $[\mathrm{Na}^+]$ bezeichnet jeweils die Konzentration monovalenter Kationen (Natrium, Kalium) in der Lösung. Die Konzentration divalenter Kationen (Magnesium, Calcium) $[\mathrm{Mg}^{2+}]$ kann durch die Formel

$$[\mathrm{Na}^+] = 4 \cdot [\mathrm{Mg}^{2+}] \tag{2.5}$$

in eine monovalente Salzkonzentration umgewandelt und in die Berechnung einbezogen werden.

Alle GC-Formeln sind Näherungsverfahren und nicht in ihrem Wirkungsbereich gleich genau. So hat zum Beispiel die Gleichung 2.4 ihre größte Genauigkeit zwischen 18 und 25 Basenpaaren. Darunter und darüber sind Ergebnisse weniger exakt. Trotzdem ist die GC-Formel sehr nützlich, weil sie leicht und schnell zu berechnen ist.

2.2.2 Nearest-Neighbor-Methode

Eine genauere, dafür aber auch aufwendigere Methode zur Bestimmung der Schmelztemperatur benutzt einen thermodynamischen Ansatz. Man betrachtet Bildung und Aufspaltung des Doppelstranges als eine Gleichgewichtsreaktion, beschrieben durch die Gleichgewichtskonstante K_{eq}. Für K_{eq} gibt es zwei unterschiedliche Definitionen. Die erste ist eine thermodynamische Definition und lautet

$$K_{eq} = \beta \exp\left(-\frac{\Delta G^0}{RT}\right) \tag{2.6}$$

mit ΔG^0 als der freien Enthalpie der Reaktion, der universellen Gaskonstante $R = 1.986 \frac{\text{cal}}{\text{mol K}}$ und der Temperatur T. Der Faktor β dient zur Anpassung der Einheit (siehe unten). Die freie Enthalpie ΔG^0 enthält einen enthalpischen Anteil ΔH^0 und einen entropischen Anteil ΔS^0 und ist definiert als

$$\Delta G^0 = \Delta H^0 - T \cdot \Delta S^0. \tag{2.7}$$

Ersetzt man in Gleichung 2.6 die freie Enthalpie durch diesen Ausdruck und stellt sie dann nach der Temperatur um, erhält man

$$T = \frac{\Delta H^0}{\Delta S^0 + R \cdot \ln(\frac{\beta}{K_{eq}})}. \tag{2.8}$$

Die zweite Definition der Gleichgewichtskonstante K_{eq} erfolgt über die Stoffkonzentrationen in der Lösung. Angenommen wird, dass das Gleichgewicht mit der Reaktionsgleichung

$$\text{A} \quad + \quad \text{B} \quad \longleftrightarrow \quad \text{AB} \tag{2.9}$$

beschrieben werden kann, wobei A und B zwei Einzelstränge und AB der Doppelstrang aus beiden sind. K_{eq} ist definiert als

$$K_{eq} = \frac{[\text{AB}]_{eq}}{[\text{A}]_{eq} \cdot [\text{B}]_{eq}}. \tag{2.10}$$

$[\text{A}]_{eq}$, $[\text{B}]_{eq}$ und $[\text{AB}]_{eq}$ sind die Konzentrationen der Einzelstränge und des Doppelstranges im Gleichgewicht.

Die Reaktion startet mit den Ausgangskonzentrationen [A], [B] und [AB] = 0. Da bei der angenommenen Reaktionsgleichung 2.9 jeweils ein Strang A und ein Strang B einen Doppelstrang AB bilden, gilt $[\text{A}]_{eq} = [\text{A}] - [\text{AB}]_{eq}$ und $[\text{B}]_{eq} = [\text{B}] - [\text{AB}]_{eq}$. Daraus folgt

$$K_{eq} = \frac{[\mathrm{AB}]_{eq}}{([\mathrm{A}] - [\mathrm{AB}]_{eq}) \cdot ([\mathrm{B}] - [\mathrm{AB}]_{eq})}. \tag{2.11}$$

Gesucht ist das Gleichgewicht im Schmelzpunkt, der allgemein als jener Punkt bezeichnet wird, an welchem die Hälfte der Einzelstränge gebunden vorliegen. Unter der vereinfachenden Annahme, dass $[\mathrm{A}] = [\mathrm{B}] = c$, bedeutet dies, dass $[\mathrm{A}]_m = [\mathrm{B}]_m = \frac{c}{2}$. Eingesetzt in Gleichung 2.11, erhält man für die Gleichgewichtskonstante im Schmelzpunkt die Formel

$$K_m = \frac{2}{c}. \tag{2.12}$$

Die Gleichgewichtskonstante in Gleichung 2.8 kann damit durch einen Ausdruck der Stoffkonzentration ersetzt werden. Für die angenommene Reaktionsgleichung hat K_m die Einheit M^{-1}, weshalb der Faktor β aus Gleichung 2.6 den Wert $1\mathrm{M}^{-1}$ erhält. Für die Schmelztemperatur ergibt sich dann

$$T_M = \frac{\Delta H^0}{\Delta S^0 + R \cdot \ln(\frac{\beta c}{x})}. \tag{2.13}$$

Der Faktor x, der in dieser Formel neu eingeführt wird, müsste nach Gleichung 2.12 den Wert 2 haben. In Experimenten hat sich aber gezeigt, dass Stränge mit selbstkomplementären Sequenzen ein anderes Schmelzverhalten als nicht-selbstkomplementäre Stränge aufweisen. Im selbstkomplementären Fall nimmt darum x den Wert 1 an. In allen anderen Fällen gilt $x = 4$. Ist $x = 1$, wird der Term $R \cdot \ln(\frac{\beta c}{x})$ größer. Der absolute Wert des Nenners sinkt dadurch, da die Entropie bei genügend großer Stranglänge einen negativen Wert annimmt. Letztendlich steigt die Schmelztemperatur. Demnach haben selbstkomplementäre Stränge eine höhere Schmelztemperatur.

Für eine genaue Berechnung der Schmelztemperatur fehlen nun noch Angaben über die Enthalpie und die Entropie der Reaktion. An dieser Stelle setzt das Nearest-Neighbor-Modell (engl. für nächster Nachbar) ein [41,46]. Es geht davon aus, dass die freie Enthalpie vor allem durch die Abfolge benachbarter Basenpaare festgelegt ist. Es gibt insgesamt 10 mögliche Kombinationen benachbarter Basenpaare. Für jede Kombination wurden experimentell Werte für Enthalpie und Entropie ermittelt und veröffentlicht [10, 41]. Sie sind in Tabelle 2.3 aufgelistet. Es wird dann angenommen, dass die Gesamtenthalpie, abgesehen von einigen Korrekturtermen, die Summe aller Enthalpien der einzelnen Basenpaar-Nachbarn ist. Gleiches gilt für die Entropie. Es gilt:

$$\begin{array}{rcl} \Delta H^0 & = & \Delta H^0_{init} + \Delta H^0_{term} + \Delta H^0_{stack} \\ \Delta H^0_{stack} & = & \sum_{i=0}^{l-2} \Delta H^0_{(i,i+1)} \\ \Delta S^0 & = & \Delta S^0_{init} + \Delta S^0_{term} + \Delta S^0_{stack} \\ \Delta S^0_{stack} & = & \sum_{i=0}^{l-2} \Delta S^0_{(i,i+1)} \end{array} \tag{2.14}$$

Die Werte ΔH^0_{init} und ΔS^0_{init} sind Ausgangswerte, mit denen jede Berechnung startet. Für die beiden Basenpaare an den Enden des Doppelstranges werden zusätzliche Werte ΔH^0_{term} und ΔS^0_{term} hinzugefügt, wobei die Werte für G/C-Paare gleich 0 sind. ΔH^0_{stack} und ΔS^0_{stack} enthalten die summierten Beiträge aller Basenpaar-Nachbarn. l bezeichnet

	ΔH^0 $[\frac{\text{kcal}}{\text{mol}}]$	ΔS^0 $[\frac{\text{cal}}{\text{mol K}}]$	ΔG^0_{37} $[\frac{\text{kcal}}{\text{mol}}]$
AA TT	-7.6	-21.3	-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	+0.2	-5.7	+1.96
term AT	+2.2	+6.9	+0.05
term GC	0	0	0

Die Daten wurden bei einer Konzentration von 1M NaCl gemessen und stammen aus einer Veröffentlichung von Santalucia und Hicks [10].

Tabelle 2.3: Thermodynamische Grunddaten des Nearest-Neighbor-Modells

die Länge des Doppelstranges. Mit den so berechneten Werten für die Enthalpie und die Entropie kann dann durch Gleichung 2.13 die Schmelztemperatur berechnet werden.

In den bisherigen Gleichungen spielte die Salzkonzentration der Lösung keine Rolle, obwohl sie Einfluss auf den Schmelzpunkt hat. Die Basiswerte in Tabelle 2.3 wurden für eine monovalente Salzkonzentration von $[Na^+] = 1M$ ermittelt. Für abweichende Konzentrationen müssen die Werte für die Entropie ΔS^0 angepasst werden. Es gilt

$$\Delta S^0_{[Na^+]} = \Delta S^0_{[1M NaCl]} + 0.368 \cdot \frac{N}{2} \cdot \ln([Na^+]), \tag{2.15}$$

wobei N die Anzahl der Phosphatgruppen im Doppelstrang (in der Regel $2 \cdot (l-1)$) ist. Die Konzentration divalenter Kationen (Magnesium, Calcium) kann wiederum mit der Formel $[Na^+] = 4 \cdot [Mg^{2+}]$ in eine Konzentration monovalenter Kationen umgerechnet werden.

Die Enthalpie ΔH^0 wird als unabhängig von der Salzkonzentration angenommen, solange $0.05M < [Na^+] < 1.1M$ gilt.

Betrachtet wird als Beispiel für eine Schmelztemperaturberechnung der Doppelstrang

$$\begin{array}{l} 5' - \text{TGCCACTCTG} - 3' \\ 3' - \text{ACGGTGAGAC} - 5' \end{array}.$$

Er besteht aus den Basenpaar-Nachbarn

$$\begin{array}{llllllllll} 5' - \text{TG} & \text{GC} & \text{CC} & \text{CA} & \text{AC} & \text{CT} & \text{TC} & \text{CT} & \text{TG} - 3' \\ 3' - \text{AC} & \text{CG} & \text{GG} & \text{GT} & \text{TG} & \text{GA} & \text{AG} & \text{GA} & \text{AC} - 5' \end{array}.$$

Der Doppelstrang soll in einer Konzentration von $c = 5 \cdot 10^{-6}M$ in der Lösung vorliegen. Die monovalente Kationenkonzentration beträgt $[Na^+] = 0.2M$. Divalente Kationen treten nicht auf oder sind bereits in die monovalente Salzkonzentration einbezogen. Unter Zuhilfenahme von Tabelle 2.3 ergibt sich für die Enthalpie

$$\begin{array}{rcll} \Delta H^0 & = & \Delta H^0_{init} + \Delta H^0_{term} + \Delta H^0_{stack} & \text{Gl. 2.14} \\ & = & \Delta H^0_{init} + \Delta H^0_{termAT} + \Delta H^0_{termGC} + \Delta H^0_{TG} + \Delta H^0_{GC} + & \\ & & \Delta H^0_{CC} + \Delta H^0_{CA} + \Delta H^0_{AC} + \Delta H^0_{CT} + \Delta H^0_{TC} + \Delta H^0_{CT} + & \\ & & \Delta H^0_{TG} & \\ & = & (0.2 + 2.2 + 0.0 - 8.5 - 9.8 & \text{Tab. 2.3} \\ & & -8.0 - 8.5 - 8.4 - 7.8 - 8.2 - 7.8 & \\ & & -8.5)\frac{\text{kcal}}{\text{mol}} & \\ \Delta H^0 & = & -73,1\frac{\text{kcal}}{\text{mol}} & \end{array}$$

und für die Entropie bei 1M Salzkonzentration

$$\begin{array}{rcll} \Delta S^0_{[1M NaCl]} & = & \Delta S^0_{init} + \Delta S^0_{term} + \Delta S^0_{stack} & \text{Gl. 2.14} \\ & = & \Delta S^0_{init} + \Delta S^0_{termAT} + \Delta S^0_{termGC} + \Delta S^0_{TG} + \Delta S^0_{GC} + & \\ & & \Delta S^0_{CC} + \Delta S^0_{CA} + \Delta S^0_{AC} + + \Delta S^0_{CT} + \Delta S^0_{TC} + & \\ & & \Delta S^0_{CT} + \Delta S^0_{TG} & \\ & = & (-5.7 + 6.9 + 0.0 - 22.7 - 24.4 + & \text{Tab. 2.3} \\ & & -19.9 - 22.7 - 22.4 - 21.0 - 22.2 + & \\ & & -21.0 - 22.7)\frac{\text{cal}}{\text{mol K}} & \\ \Delta S^0_{[1M NaCl]} & = & -197.8\frac{\text{cal}}{\text{mol K}}. & \end{array}$$

Sequenz	Länge	#GC	T_m (GC-Formel)	T_m (Nearest-Neighbor)
GGAAATACTT	10	3	36.0^{o}C	29.2^{o}C
TGCCACTCTG	10	6	42.0^{o}C	44.5^{o}C
GGTCGGAGGC	10	8	46.0^{o}C	50.7^{o}C
AATAGCAGAGTAAGG	15	6	50.6^{o}C	50.4^{o}C
GGTGCCCGAGTGTCC	15	12	67.0^{o}C	69.2^{o}C
CAGACATAATCTAAACGGAG	20	8	64.3^{o}C	58.2^{o}C
GGGGAGCCGCAGGCGATGCC	20	16	70.7^{o}C	78.5^{o}C

Die Strangkonzentration beträgt bei allen Strängen $5 \cdot 10^{-6}$M, die monovalente Salzkonzentration beträgt immer 0.2M.

Tabelle 2.4: Schmelztemperaturen verschiedener DNA-Doppelstränge

Die an die verlangte Salzkonzentration angepasste Entropie beträgt

$$\begin{aligned} \Delta S^0_{[\mathrm{Na}^+]} &= \Delta S^0_{[1\mathrm{M}\mathrm{NaCl}]} + 0.368 \cdot \tfrac{N}{2} \cdot \ln([\mathrm{Na}^+]) \quad \text{Gl. 2.15} \\ &= -197.8 + 0.368 \cdot \tfrac{18}{2} \cdot \ln(0.2) \tfrac{\mathrm{cal}}{\mathrm{mol\ K}} \\ \Delta S^0_{[\mathrm{Na}^+]} &= -203.1 \tfrac{\mathrm{cal}}{\mathrm{mol\ K}}. \end{aligned}$$

Daraus ergibt sich schließlich die Schmelztemperatur

$$\begin{aligned} T_m &= \tfrac{\Delta H^0}{\Delta S^0_{[\mathrm{Na}^+]} + R \cdot \ln(\frac{\beta c}{x})} \quad \text{Gl. 2.13} \\ &= \tfrac{-73.1 \cdot 1000}{-203.1 + 1.986 \cdot \ln(\frac{5 \cdot 10^{-6}}{4})} \mathrm{K} \\ T_m &= 317.7\mathrm{K}\ (44.5^o C). \end{aligned}$$

Der Doppelstrang besitzt also unter den gegebenen Randbedingungen eine Schmelztemperatur von 44.5^{o}C.

Die aktuellen Werte für das Nearest-Neighbor-Modell sind ausreichend genau bei Stranglängen zwischen 8 und 60 Basenpaaren. Außerhalb dieser Grenzen treten auch bei diesem Modell größere Ungenauigkeiten auf.

In Tabelle 2.4 sind beispielhaft die Schmelztemperaturen mehrerer Doppelstränge, berechnet jeweils mit der GC-Formel und dem Nearest-Neighbor-Modell, aufgeführt.

2.3 Methoden zur DNA-Manipulation

In den letzten Jahrzehnten wurde eine ganze Reihe von molekularbiologischen Techniken entwickelt, um die DNA analysieren und manipulieren zu können. Dieses Kapitel beleuchtet diejenigen Techniken, die auf dem Gebiet der DNA-Nanotechnologie zur Anwendung kommen.

2.3.1 Synthese

Es ist möglich, DNA mit einer beliebigen Basensequenz aus einzelnen Nukleotiden zu synthetisieren. Den einzig limitierenden Faktor bildet die Länge des zu synthetisierenden

Stranges. Biochemische Techniken sind immer mit einer Fehlerrate behaftet. Ab einer bestimmten Stranglänge führt das dazu, dass die Ausbeute an korrekten Strängen zu gering und damit unpraktikabel wird. Heutzutage kann man über 100 Basen lange DNA-Einzelstränge mit beliebiger Sequenz kommerziell erwerben [56, 57].

Die Synthese erfolgt schrittweise nach dem Prinzip der wachsenden Kette. Ein Träger (zum Beispiel Glasträger) wird so präpariert, dass Nukleoside, die alle die erste Base der Sequenz enthalten, aus einer Lösung heraus daran binden kann. Alle noch freien Nukleoside werden danach ausgewaschen und es beginnen die Schritte des Kettenwachstums. Jeder Wachstumsschritt besteht aus zwei Phasen. In der ersten Phase werden an die Desoxyriboseringe des letzten Nukleosids Phosphatgruppen gebunden. In der zweiten Phase werden neue Nukleoside vom aktuellen Typ über den Träger geleitet und binden dort an die soeben angebrachten Phosphatgruppen. Alle Nukleoside, die nirgendwo anbinden konnten, werden wiederum ausgewaschen. Beide Phasen werden mit dem jeweiligen Nukleosidtyp solange wiederholt, bis die gesamte Sequenz fertig synthetisiert ist. Auf dem Träger wachsen so parallel viele einzelne Stränge mit der gleichen Sequenz. Natürlich kommt es vor, dass während eines Wachstumsschrittes ein Strang auf dem Träger ohne neues Nukleotid bleibt. Dieser Strang hat dann zwar eine falsche Sequenz, ist aber auch kürzer. Nach dem Lösen der Einzelstränge vom Träger können diese falschen, kürzeren Sequenzen durch eine Gel-Elektrophorese (siehe Kap. 2.3.7) entfernt werden.

2.3.2 Melting/Annealing

Die beiden wichtigsten Techniken für die DNA-Nanotechnologie sind die Hybridisierung und die Dehybridisierung der Einzelstrangmoleküle, hier genannt Annealing (engl. für Tempern, Abkühlen) und Melting (engl. für Schmelzen). Bei der Hybridisierung verbinden sich zwei Einzelstränge mit komplementären Sequenzen über die Wasserstoffbrückenbindungen ihrer Basen zu einem Doppelstrang. Die Dehybridisierung ist die gegensätzliche Reaktion des Aufbrechens der Wasserstoffbrücken.

Das Melting (Dehybridisierung) erfolgt durch Erhitzen der Lösung, in welcher sich das DNA-Material befindet, auf über 95°C. Bei dieser Temperatur sind alle Wasserstoffbrückenbindungen aufgespalten. Es liegen nur noch Einzelstränge in der Lösung vor.

Beim Annealing (Hybridisierung) kühlt man die vorher durch ein Melting erhitzte Lösung bis unter die Schmelztemperatur der zu erwartenden Doppelstränge (siehe Kap. 2.2) herunter. Dabei bilden sich die Wasserstoffbrücken wieder aus und die Einzelstränge binden wieder aneinander. In diesem Vorgehen steckt ein großes Potenzial. Während des Abkühlens finden sich Stränge mit komplementären Sequenzen, binden aneinander und formen größere und komplexere DNA-Moleküle. Durch die Wahl der Basensequenzen der Stränge kann festgelegt werden, welche Stränge sich miteinander verbinden. Man kann damit durch die Basensequenzen die Sekundärstruktur der entstehenden komplexeren Moleküle vorherbestimmen. Die Assemblierung erfolgt durch einfaches Abkühlen des DNA-Materials ohne weitere zusätzliche Eingriffe.

Der Hybridisierungsprozess kann durch Art und Geschwindigkeit des Abkühlens beeinflusst werden. Schnelles Abkühlen (bis zu nur wenigen Sekunden) bevorzugt kürzere Doppelstrangabschnitte, langsames Abkühlen (bis zu mehreren Tagen) bietet dagegen längeren Strangabschnitten genügend Zeit, aneinander zu binden. Das Annealing kann auch in mehreren Etappen erfolgen, indem man zum Beispiel sehr schnell auf eine be-

stimmte Temperatur abkühlt, dort längere Zeit verharrt und erst später den Kühlprozess fortsetzt. Damit kann erreicht werden, dass sich Doppelstränge, deren Schmelzpunkt bei dieser Temperatur liegt, besonders bevorzugt ausbilden. Das genaue Vorgehen hängt von der gewünschten DNA-Struktur ab.

2.3.3 Ligation

Eine ebenfalls sehr wichtige Technik der DNA-Nanotechnologie ist die Ligation. Sie ist eine enzymatische Reaktion, bei welcher DNA-Doppelstränge miteinander verkettet werden können. Das eingesetzte Enzym, eine DNA-Ligase, katalysiert dabei die Verbindung der Strangenden durch eine Phosphatgruppe. Ohne das Enzym würde diese Reaktion unter normalen Bedingungen nicht stattfinden. Die Ligasen benötigen deshalb auch ein Cosubstrat als Energielieferant. Je nach Enzym ist das entweder ATP (Adenosintriphosphat) oder NAD (Nicotinamidadenindinukleotid).

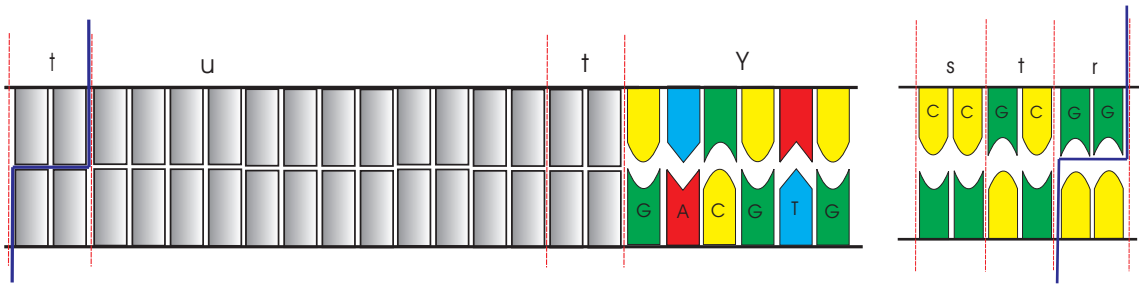
Vom verwendeten Enzym hängt auch ab, wie die Strangenden geartet sein müssen. Manche Enzyme verknüpfen nur Doppelstränge mit komplementären Einzelstrangüberhängen (im engl.: sticky ends). Andere können auch Doppelstränge mit glatten Enden (engl.: blunt ends) miteinander verbinden. In beiden Fällen ist es erforderlich, dass die 5'-Enden der Einzelstränge an der Verbindungsstelle mit einer Phosphatgruppe versehen sind.

Im Labor erfolgt die Ligation durch Inkubation der zu verknüpfenden DNA-Moleküle mit dem Enzym und dem energieliefernden Cosubstrat. Eine übliche Reaktionszeit ist 12 Stunden bei einer Temperatur von 37^oC. Die genauen Reaktionsbedingungen hängen jedoch vom verwendeten Enzym ab. Übliche Enzyme sind die T4-Ligase, die sowohl blunt- als auch sticky-Enden verknüpft, oder die *E.coli*-Ligase, welche jedoch nur sticky-Enden miteinander verbindet.

Während der Ligationsreaktion finden und verbinden sich DNA-Doppelstränge zu größeren Molekülen. Gesteuert werden kann dieser Prozess einerseits durch die Basensequenzen von Einzelstrangüberhängen (nur komplementäre Überhänge binden aneinander) und andererseits durch das Anbringen oder Entfernen von Phosphatgruppen an den Strangenden (nur phosphorylisierte Strangenden werden verknüpft). Die Ligation hat deshalb ein ähnliches Potenzial wie das Annealing. Durch die Wahl von Basensequenzen oder Molekülgruppen an einzelnen DNA-Strängen kann die Struktur eines größeren DNA-Moleküls vordefiniert werden.

2.3.4 Restriktion

Die gegensätzliche Reaktion zur Ligation ist die Restriktion, oft auch Digestion (engl.: Verdauen) genannt. Durch spezielle Enzyme, die Restriktionsendonukleasen oder kurz Restriktionsenzyme, werden DNA-Doppelstränge zerschnitten. Jedes Enzym besitzt eine spezifische Erkennungssequenz. An Stellen, wo diese Sequenz auftaucht, dockt das Enzym an das DNA-Molekül und katalysiert das Zerschneiden des Strangrückgrates. Die Art des Schnittes ist ebenfalls enzymspezifisch. Es gibt sowohl glatte Schnitte, als auch solche, die zu Einzelstrangüberhängen führen. Die Schnittstelle kann innerhalb der Erkennungssequenz, aber auch mehrere Basenpaare davon entfernt liegen. Restriktionsenzyme,



(a) *BsgI* (b) *Cfr42I*

Abbildung 2.7: Erkennungssequenz und Schnittstelle zweier Restriktionsenzyme

die außerhalb der Erkennungssequenz schneiden, benötigen als Energielieferant ATP als Cosubstrat.

In Organismen dienen die Restriktionsenzyme dazu, fremde DNA zu zerstören und damit unschädlich zu machen. Aus verschiedensten Bakterienstämmen werden Restriktionsendonukleasen extrahiert, welche mit ihren Erkennungssequenzen und Schnittstellen in Katalogen verzeichnet sind und kommerziell beschafft werden können [58, 59]. Abbildung 2.7 zeigt Erkennungssequenz und Schnittverhalten zweier Enzyme als Beispiele.

Im Labor wird die zu zerschneidende DNA zusammen mit dem Enzym und dem eventuell notwendigen ATP inkubiert. Üblich ist eine Reaktionszeit von 3 Stunden bei einer Temperatur von 37^{o}C. Die genauen Reaktionsbedingungen hängen aber vom jeweiligen Enzym ab.

In der DNA-Nanotechnologie wird die Restriktion hauptsächlich zu Analysezwecken verwendet. Wenn man eine DNA-Struktur gebaut hat, muss untersucht werden, ob diese der gewünschten Struktur entspricht. Eine Methode dafür ist, die generierte Struktur durch Restriktionsenzyme wieder zu zerschneiden und an Hand der Eigenschaften der Bruchstücke (siehe Kap. 2.3.7) auf die Gestalt der Gesamtstruktur zu schließen. Man kann die Restriktion aber auch als einen echten Schritt beim Aufbau der gewünschten DNA-Struktur verwenden.

2.3.5 Polymerisation und Polymerase-Kettenreaktion

Allgemein bedeutet Polymerisation das Zusammensetzen mehrerer Monomere zu einem Polymer. Auf die DNA bezogen bedeutet es, Nukleotide zu einem DNA-Strang aneinanderzureihen. Die Reaktion wird durch ein Enzym, eine Polymerase, katalysiert. Die Sequenz des entstehenden Stranges wird durch einen bereits existierenden Einzelstrang, an dem die Polymerisation entlang verläuft, festgelegt. Als Ausgangspunkt dient ein Doppelstrang mit einem 5'-Einzelstrangüberhang. Dieser Überhang wird, ausgehend vom bereits vorhandenen Doppelstrang, nacheinander mit Nukleosidtriphosphaten (*dNTP*s) zu einem kompletten Doppelstrang aufgefüllt. Dabei bindet das Triphosphat des jeweiligen Nukleotids unter Abspaltung von Diphosphat an das 3'-Ende des Vorgängers. Das Abspalten des Diphosphates liefert die für diese Reaktion notwendige Energie. Der Einzelstrang wächst somit in Richtung seines 3'-Endes. Die Reihenfolge der Nukleotide wird durch die Basensequenz des ursprünglichen Überhanges bestimmt.

Gleichzeitig zum Auffüllen der 5'-Überhänge werden 3'-Überhänge durch das Enzym abgebaut.

Die DNA-Polymerisation ist die Grundlage der Erbgut-Replikation während der Zellteilung. In der Biotechnologie wird sie benutzt, um sehr schnell viele Kopien von DNA-Material zu produzieren. Die Technik, mit der dies geschieht, ist die Polymerase-Kettenreaktion (PCR, engl.: polymerase chain reaction). Sie ist eine zyklische Folge der Techniken Melting, Annealing und Polymerisation.

Möchte man einen DNA-Doppelstrang mittels PCR vervielfältigen, werden eine thermostabile Polymerase und kurze DNA-Stücke (die sogenannten Primer) mit den Startsequenzen (am 5'-Ende) der beiden Einzelstränge benötigt. Die Primer sind in der Regel 10 bis 20 Basen lang. Man gibt sie zusammen mit dem DNA-Templat, der Polymerase und genügend freien Nukleotidtriphosphaten in eine Lösung und führt dann mehrmals hintereinander eine Melting-, eine Annealing- und eine Polymerisationsphase durch.

In der Melting-Phase wird die Lösung auf über 95^oC erhitzt. Dadurch werden alle Doppelstränge aufgespalten. In der Annealing-Phase wird die Lösung sehr schnell bis unter die Schmelztemperatur der Primer (siehe Kap. 2.2) abgekühlt. Das schnelle Abkühlen bevorzugt die Anlagerung der kurzen Primer gegenüber der kompletten Hybridisierung der ursprünglichen Doppelstränge. Die Primer binden also an die zu vervielfältigenden Einzelstränge, und zwar an deren 3'-Enden. Es entstehen kurze Doppelstränge mit langen 5'-Einzelstrangüberhängen. Diese Überhänge werden dann in der Polymerisationsphase mit den freien Nukleotiden aufgefüllt.

Wenn die verwendete Polymerase thermostabil ist, lässt sich die gesamte Prozedur sehr leicht über die Temperatur steuern. Es gibt die sogenannten Cycler (vom engl. cycle für Kreis), in die man den Temperaturverlauf des kompletten PCR-Schrittes einprogrammieren kann, und die dann selbstständig mehrere solcher Schritte hintereinander durchführen. Üblich sind bis zu 40 PCR-Schritte. Bei jedem Schritt verdoppelt sich die Zahl der zu kopierenden DNA-Stränge, was einen exponentiellen Mengenzuwachs nach sich zieht.

In der DNA-Nanotechnologie findet die Polymerisation vor allem in Form der PCR breite Anwendung bei der Synthese und Analyse. Stränge mit frei synthetisierten Basensequenzen können durch die PCR vervielfältigt werden. Gleiches gilt für DNA-Material, das durch andere Prozesse erzeugt wurde (zum Beispiel durch Annealing und Ligation) und welches nach einer PCR besser nachgewiesen werden kann. Beim eigentlichen Aufbau einer DNA-Struktur spielt die Polymerisation bis jetzt aber keine Rolle.

2.3.6 Label

DNA lässt sich auf verschiedenste Weise mit anderen Molekülen markieren oder funktionalisieren. Das Spektrum reicht von einer einfachen Markierung des gesamten Moleküls mit einem Farbstoff bis hin zur Funktionalisierung einzelner Nukleotide mit einer spezifischen chemischen Gruppe. Hier soll auf drei der wichtigsten Funktionalisierungen eingegangen werden: die Phosphorylisierung, die Biotinylisierung und die Thiolysierung.

Phosphatgruppen

Bei einem DNA-Einzelstrang kann das äußerste Nukleotid am 5'-Ende mit einer Phosphatgruppe versehen sein oder nicht. Mit Hilfe enzymatischer Reaktionen können diese

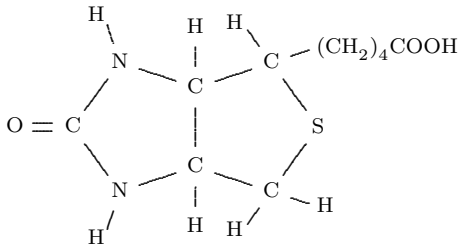


Abbildung 2.8: Strukturformel des Biotin

Gruppen gezielt an- oder abgebaut werden.

Die Phosphorylisierung wird durch ein Polynukleotidkinease-Enzym katalysiert. Unter Verbrauch von ATP, welches neben der Energie auch die Phosphatgruppe bereitstellt, ersetzt es die OH-Gruppe am C5-Atom der Desoxyribose. Die Dephosphorylisierung erfolgt unter Einwirkung einer Phosphatase. Dieses Enzym spaltet die Phosphatgruppe wieder vom C5-Atom der Desoxyribose ab und ersetzt sie durch eine OH-Gruppe. Beide Reaktionen finden üblicherweise bei einer Temperatur von 37^{o}C über einen Zeitraum von einer Stunde hinweg statt. Danach ist das Nukleotid zum Nukleosid geworden. Am C5-Atom seiner Desoxyribose befindet sich nur noch eine OH-Gruppe.

Durch die An- bzw. Abwesenheit dieser Gruppe lässt sich eine Ligations-Reaktion steuern (siehe Kap. 2.3.3). Ist keine Phosphatgruppe vorhanden, findet keine Ligation statt. Mit Hilfe von Enzymen kann man Phosphatgruppen an die Strangenden anbringen.

Biotin / Streptavidin

Biotin ist ein Vitamin und als solches auch als Vitamin B_7 oder Vitamin H bekannt. Es spielt bei vielen enzymatischen Reaktionen beim Stoffwechsel eine bedeutende Rolle. Die Struktur des Biotin ist in Abbildung 2.8 dargestellt.

Biotin besitzt eine Carboxylgruppe (—COOH), welche es sehr reaktions- und bindungsfreudig macht. Gibt man Biotin in eine Lösung mit DNA, bindet es unter Abspaltung von Wasser an das 5'-Ende der Stränge, wenn diese mit einer OH-Gruppe (und nicht mit einer Phosphatgruppe) versehen sind. Dies ist eine sehr einfache und wirksame Reaktion. Sie ist jedoch nicht ohne weiteres rückgängig zu machen und blockiert außerdem die Strangenden der DNA. Ligationen sind an diesen Stellen nicht mehr möglich.

Es gibt darum auch Verfahren, die das Biotin anders an ein Nukleotid binden. Üblich ist die Methode, das Biotin über einen Kohlenstoff-Arm (engl.: spacer, linker) an ein Stickstoff-Atom in der Nukleobase zu binden. Beispiele dafür sind das Biotin-14-dCTP und das Biotin-21-dUTP. Beides sind Desoxynukleosidtriphosphate, an die das Biotin über einen 14C- bzw. einen 21C-Arm gebunden ist. Die Kohlenstoff-Arme sorgen für eine Distanz zwischen dem Biotin einerseits und den Nukleotiden andererseits, so dass die normale Struktur der DNA nicht zu sehr gestört wird.

Mit Biotin versehene Nukleotide können heute kommerziell beschafft werden. Sie werden dann sowohl bei reiner Synthese von DNA als auch bei der Polymerisation verwendet, um Biotin-Moleküle an genau definierte Stellen in einem DNA-Strang zu bringen.

Was ist nun durch ein Biotin-Molekül an einem DNA-Strang gewonnen? Biotin bindet sehr stark, jedoch nicht-kovalent, an Avidin und Streptavidin. Beides sind Proteine aus vier identischen Untereinheiten. An jede Einheit kann jeweils ein Biotin-Molekül binden. Daraus ergeben sich viele Möglichkeiten in der Nanotechnologie. Man kann zum Beispiel vier biotinylierte DNA-Stränge über ein Streptavidin-Molekül zu einem vierarmigen Komplex verbinden. Avidin und Streptavidin lassen sich aber auch zu makroskopischen Kügelchen formen, an welche sehr viel biotinylierte DNA binden kann. Durch Zentrifugieren kann man dann diese DNA von derjenigen, die nicht mit Biotin versehen war, räumlich trennen. Es besteht die Möglichkeit, bestimmte DNA-Stränge von anderen zu unterscheiden.

Nicht zuletzt können über Avidin und Streptavidin viele andere Substanzen an DNA gebunden werden. Da man das Biotin an fest definierten Stellen im Molekül platzieren kann, sind diese Substanzen an genau festgelegten Orten. Die DNA kann dadurch sehr ortsgenau funktionalisert oder markiert werden.

Thiolgruppen

Die Thiolgruppe ist eine funktionelle Gruppe, bestehend aus einem Schwefel- und einem Wasserstaffatom (—SH). Ihre Struktur gleicht der Hydroxylgruppe (—OH), die Bindung des Wasserstoffs ist jedoch schwächer, da Schwefel weniger elektronegativ als Sauerstoff ist.

Es gibt verschiedene Arten, eine Thiolgruppe in die DNA zu integrieren. Zum einen ist es möglich, eine Thiolgruppe über ein Phenylmethyl an der Phosphatgruppe am 5'-Ende eines Stranges zu installieren. Wie schon beim Biotin ist damit jedoch dieses Strangende für weitere Reaktionen gesperrt. Man kann aber auch in die Nukleobasen ein Schwefelatom anstelle eines Stickstoffatoms einbauen. Zusammen mit dem Wasserstoff eines benachbarten Kohlenstoff-Atoms bildet der Schwefel dann eine Pseudo-Thiolgruppe. Das Anbringen der Thiolgruppe an einem Kohlenstoff-Arm ist ebenfalls möglich. Die thioliserten Nukleotide lassen sich dann wiederum über Synthese und Polymerisation in DNA-Stränge einbauen.

An die Thiolgruppe können sich andere Substanzen binden. Für die DNA-Nanotechnologie am wichtigsten ist dabei die hohe Affinität zu Gold. Bringt man Gold in eine Lösung mit thiolisierter DNA, so bindet diese daran. DNA-Stränge können dadurch fixiert oder separiert werden. Andersherum lassen sich Gold-Cluster auf die gleiche Weise an genau definierte Stellen auf einem DNA-Strang befestigen [9].

2.3.7 Gel-Elektrophorese

Die Gel-Elektrophorese ist eine Methode, DNA-Stränge nach ihrer Länge zu separieren. Man bringt dazu die DNA-Probe in ein elektrisches Feld. Durch die negative Ladung der Phosphatgruppen wandern die Stränge von der negativ geladenen Kathode zur positiv geladenen Anode. Der ganze Prozess findet in einem Gel statt, welches die Bewegung der Moleküle bremst. Die Bremswirkung hängt im Wesentlichen von der Porengröße des Gels ab. Agarose-Gele besitzen relativ große Poren von 50 bis 150 nm. Dagegen sind die Poren in Polyacrylamid-Gelen nur zwischen 3 und 6 nm klein.

Die Geschwindigkeit, mit der sich ein DNA-Strang im Gel bewegt, hängt neben

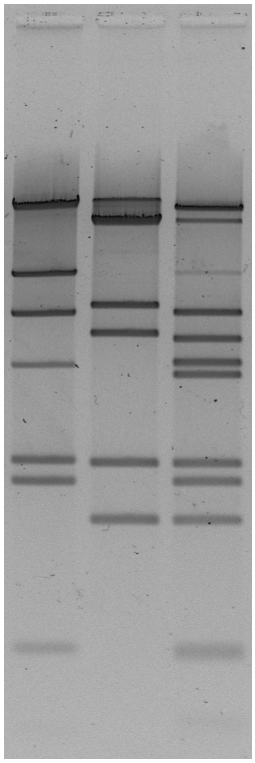


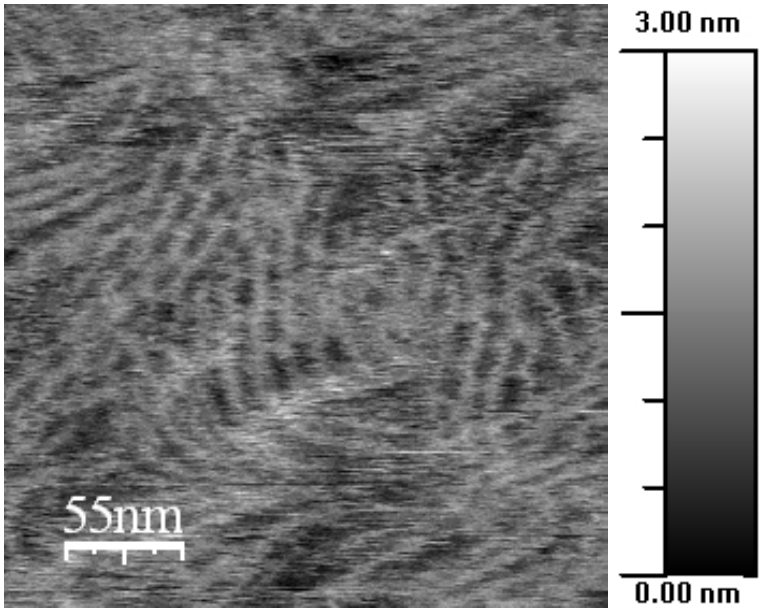
Abbildung 2.9: Auswertungsbild einer Gel-Eletrophorese

der Porengröße des Gels auch von der angelegten Spannung, seiner Ladung und seiner Stranglänge ab. In der Regel laufen kurze Stränge schneller als lange. Nach einer bestimmten Laufzeit haben sich deshalb die Stränge in Bereiche mit gleichen Längen separiert (die sogenannten Banden). Nach der Elektrophorese wird das DNA-Material gefärbt und unter Ultraviolett-Licht (UV) analysiert. Ein Beispiel ist in Abbildung 2.9 zu sehen. Dort gibt es drei Bahnen, in elchem unterschiedliches DNA-Material im selben Gel von oben nach unten gelaufen ist. Die verschieden langen DNA-Stücke haben sich dabei in den dunklen Banden angereichert.

Die Gel-Elektrophorese wird benutzt, um die Existenz bestimmter DNA-Fragmente nachzuweisen. Damit kann auf Erfolg oder Misserfolg von vorher durchgeführten Operationen geschlossen werden. Zum Beispiel können nach einer Digestion die verbliebenen DNA-Stücke erkannt werden. Fehlen diese, hat die Restriktion nicht funktioniert. Ähnlich, nur umgekehrt, erfolgt der Nachweis einer Ligation. Weiterhin ist auch die Bildung größerer DNA-Komplexe durch Selbstassemblierung in einem Gel sichtbar.

2.3.8 Rasterkraft-Mikroskopie (AFM)

Die Rasterkraft-Mikroskopie (AFM, von engl.: atomic force microscopy) ist eine Methode, Strukturen im Nanometerbereich sichtbar zu machen. Dabei wird eine an einer Blattfeder, dem Cantilever, befestigte Messspitze zeilenweise über eine Probe geführt. Es gibt diverse Modi, in denen die Rasterung vorgenommen wird. Sie unterscheiden sich danach, ob die Messspitze im Kontakt mit der Probe steht oder nicht, und nach unterschiedlichen



Die Aufnahme wurde von Alexander Huhle am Max-Bergmann-Zentrum für Biomaterialien der TU Dresden gemacht.

Abbildung 2.10: DNA in einer AFM-Aufnahme

Auswerteverfahren. Im Kontakt-Modus steht die Messspitze in direktem mechanischen Kontakt mit der Probe. Deren Oberfläche lenkt die Spitze beim Abtasten ab, was zu einer Verbiegung des Cantilever führt. Im Nicht-Kontakt-Modus wird die Messspitze dagegen in einigem Abstand über die Probe geführt. Hierbei führen die atomaren Wechselwirkungen zwischen Messspitze und Probenoberfläche zur Ablenkung der Spitze und zu einer Verbiegung des Cantilever. Diese Verbiegung des Cantilever kann gemessen werden, wodurch sich ein Höhenprofil der Probe erstellen lässt.

Für die Darstellung von DNA im Rasterkraft-Mikroskop eignet sich besonders der Tapping-Modus. In diesem Modus wird die Messspitze nicht einfach über die Probe gezogen, sondern zusätzlich gehoben und gesenkt. Dadurch tippt die Spitze immer nur kurz auf die Probe. Lose Moleküle auf der Probenoberfläche, wie die DNA, werden so nicht von der Messspitze verschoben. In anderen Modi kommt so etwas vor.

Je nach Beschaffenheit der Nadel und dem Messmodus erreicht man Auflösungen von 0.1 bis 10 nm, wodurch im Idealfall sogar einzelne Atome sichtbar werden. Die gemessene Topographie wird schließlich als Graustufen- oder Farbbild ausgegeben.

Die AFM-Aufnahme in Abbildung 2.10 zeigt langgestreckte DNA-Moleküle ähnlich der Struktur in Abbildung 2.17 auf Seite 33, jedoch mit kurzen senkrecht abstehenden Seitenarmen. Die Arme sorgen dafür, dass sich die einzelnen DNA-Ketten in einem gewissen Abstand voneinander entfernt auf der Oberfläche ablagern. Es entsteht ein ziegelsteinartiges Muster.Die Aufnahme wurde von Alexander Huhle am Max-Bergmann-Zentrum für Biomaterialien der TU Dresden gemacht.

Außer der Oberflächentopographie können mit dem AFM noch weitere Eigenschaften der Probe ermittelt werden. Dazu gehören elektrische und magnetische Feldstärken, chemische Kräfte sowie Steifigkeit und Bindungsfestigkeit einzelner Moleküle. Die Ma-

gnetfeldmessung wird vor allem bei der Produktion von Computer-Festplatten als Qualitätskontrolle verwendet.

2.4 DNA-Nanostrukturen

2.4.1 Selbstassemblierung

DNA-Nanostrukturen sind im weiteren Sinne alle Strukturen, die wesentlich durch DNA gebildet oder stabilisiert werden, und im engeren Sinne alle Strukturen, die abgesehen von einzelnen Funktionalisierungen komplett aus DNA bestehen. Sie werden unter Verwendung der soeben beschriebenen Techniken erzeugt und untersucht. Die bei weitem häufigste Herstellungsmethode ist die Selbstassemblierung der gewünschten DNA-Struktur durch das Annealing. Wie in Kapitel 2.3.2 beschrieben, finden bei dieser Technik während eines Abkühlungsprozesses Einzelstränge oder Abschnitte von Einzelsträngen mit komplementären Basensequenzen zueinander und bilden Wasserstoffbrücken zwischen ihren Basen aus. Dies geschieht bei geeigneten Bedingungen ohne sonstiges Zutun. Die Gestalt der entstehenden DNA-Strukturen hängt von der Basensequenz der beteiligten Einzelstränge ab. Sie kann also durch geeignete Wahl der Sequenzen vorherbestimmt werden. Da man Einzelstränge mit jeder beliebiger Basensequenz synthetisieren kann, sind hier auch keine Grenzen gesetzt. Lediglich längere Stränge mit deutlich mehr als 100 Basen müssen zuvor aus kleineren Stücken zusammengesetzt werden.

Damit der Zusamenbau gelingt, müssen die Basensequenzen bestimmten Anforderungen genügen. Zuallererst sollten die entstehenden Doppelstränge stabil sein, das heißt ihre Schmelztemperaturen müssen genügend hoch sein (siehe Kap. 2.2). Als kleinste sinnvolle Doppelstranglänge haben sich 5 Basenpaare herausgebildet. Der Maximallänge sind prinzipiell keine Grenzen gesetzt.

Des Weiteren müssen die Basensequenzen so ausgelegt sein, dass die gewünschte Zielstruktur mit sehr hoher Wahrscheinlichkeit entsteht. Für eine bestimmte Sequenzkonfiguration sind immer mehrere Strukturkonstellationen denkbar, denn jede Base kann ja mit jeder anderen komplementären Base, von denen es viele gibt, eine Bindung eingehen. Allerdings sind die meisten Konstellationen äußerst instabil, weil die unerwünschten Paarungen sehr kurz sind. Beim Sequenzdesign muss also dafür gesorgt werden, dass Fehlkonstellationen möglichst instabil sind und deshalb gegenüber der Zielstruktur eine sehr viel geringere Auftrittswahrscheinlichkeit besitzen. Die Fehlkonstellationen treten dann im Experiment zwar auf, jedoch nur in sehr geringer Zahl, so dass die Erzeugung der Zielstruktur nicht gestört wird.

Auf dem Problem des Sequenzdesigns für die Selbstassemblierung liegt der Schwerpunkt dieser Arbeit. Es wird im Kapitel 3.2 umfassend behandelt.

Auch mit der Ligation lassen sich DNA-Nanostrukturen assemblieren. Durch eine enzymatische Reaktion werden dabei Doppelstränge mit Einzelstrangüberhängen oder mit glatten Enden verknüpft (siege Kap. 2.3.3). Im Fall der Einzelstrangüberhänge, bestimmen auch hier die Basensequenzen, welche Komponenten miteinander verknüpft werden. Allerdings müssen die Sequenzen außer der Komplementarität keinen weiteren Anforderungen genügen. Stabilitätsfragen spielen keine Rolle, weil das Ligase-Enzym die Strangrückgrate vollständig verbindet.

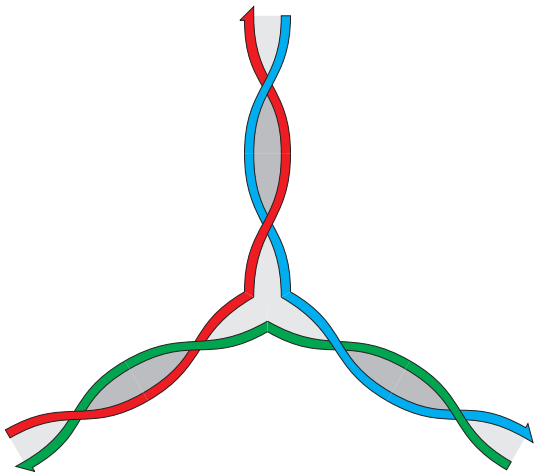


Abbildung 2.11: Darstellung einer dreiarmigen Verzweigung

Man verwendet die Ligation, um Grundelemente zu einer größeren Struktur zusammenzubauen, noch öfter aber, um die bei der Selbstassemblierung entstanden Lücken im Strangrückgrat zu schließen. Die Stabilität der entstandenen DNA-Struktur wird dadurch erheblich erhöht.

In den letzten Jahren wurden viele verschiedene DNA-Nanostrukturen meist durch Selbstassemblierung erzeugt. Die folgenden Unterkapitel beschreiben einige Grundstrukturen aber auch komplexere Gebilde, um einen Eindruck vom Potenzial der DNA als Werkstoff zu vermitteln. Eine noch ausführlichere Darstellung kann in [2] nachgelesen werden.

2.4.2 Einfache Verzweigungen

Die lineare Doppelhelix ist die natürliche Form der DNA unter normalen Bedingungen. Aber schon in der lebenden Zelle treten während der DNA-Replikation Abweichungen in Form von Verzweigungen (engl.: junction) auf [52]. Diese Verzweigungen sind allerdings nur ein Übergangszustand und nicht stabil. Die Strukturen dienen als Vorbild für stabile, künstliche Verzweigungen.

Beginnen wir mit dem Einfachsten: einer dreiarmigen Verzweigung. Abbildung 2.11 zeigt eine solche Struktur, an welcher drei DNA-Einzelstränge beteiligt sind. Jeder Strang ist an seinem 5'-Ende mit dem ersten und an seinem 3'-Ende mit dem zweiten der beiden anderen Stränge verbunden. Solche Konstrukte wurden schon sehr früh erzeugt [47]. Die drei Arme richten sich bei der Ablage auf einer Oberfläche in einem Winkel von ungefähr 120^o aus. Allerdings ist dieser Winkel weder immer gleich noch fest. Zug oder Druck an den Armen kann die Winkelkonstellation erheblich verzerren. Es kommt auch vor, dass einer der Arme zwischen die beiden anderen klappt, so dass eine $60^o/60^o/240^o$-Winkelverteilung entsteht.

Damit sich die drei Einzelstränge zu der dreiarmigen Verzweigung zusammenfinden können, müssen ihre Basensequenzen zueinander passen. Der Anfang des ersten Stranges muss komplementär zum Ende des dritten Stranges und sein Ende komplementär zum Anfang des zweiten Stranges sein. Ebenfalls komplementär müssen das Ende des zweiten

Strang	Sequenz
A:	CCGTGGATACCTTAGTCGCC
B:	GGCGACTAAGAAATGAGCAG
C:	CTGCTCATTTGTATCCACGG

Tabelle 2.5: Passende Basensequenzen für eine dreiarmige Verzweigung

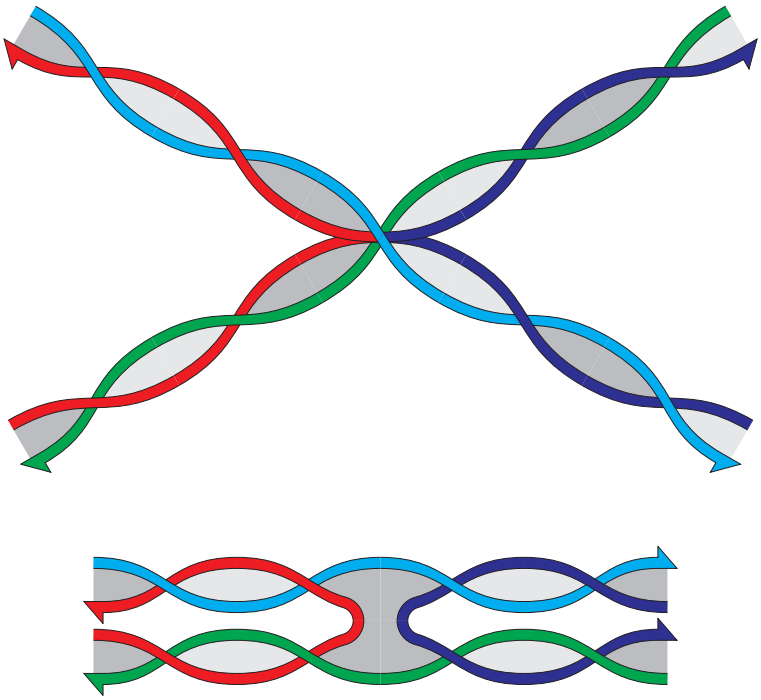


Abbildung 2.12: Darstellung einer vierarmigen Verzweigung in Draufsicht (oben) und Seitenansicht (unten)

und der Anfang des dritten Stranges sein. Tabelle 2.5 zeigt passende Basensequenzen für eine solche Struktur.

Sind die Sequenzen passend und einige Randbedingungen erfüllt, bildet sich die Verzweigung durch Abkühlen der DNA-Lösung (Annealing) durch Selbstassemblierung.

Wird das dreiarmige Ensemble um einen zusätzlichen Strang erweitert, entsteht eine vierarmige Verzweigung, wie sie in Abbildung 2.12 zu sehen ist. Experimente legen nah, dass sich die Arme bei dieser Konstellation unter normalen Bedingungen nicht in einem 90^o Winkel, sondern in einem $120^o/60^o$ Arrangement anordnen. Es entstehen dabei zwei Doppelhelices, die am Kreuzungspunkt zwei Einzelstränge untereinander austauschen [35]. Die beiden Helices liegen außerdem in zwei Ebenen übereinander.

Ein fünfter, sechster oder auch siebenter Strang kann gleichermaßen dazugenommen werden, um Verzweigungen mit noch mehr Armen zu erzeugen. Allerdings sind diese Konstrukte immer instabiler und schwerer zu beherrschen. Deshalb haben sich im Wesentlichen nur die drei- und vierarmigen Verzweigungen als Bauelemente durchgesetzt.

Vielarmige Verzweigungen lassen sich auch durch Verknüpfung von Verzweigungen

mit weniger Armen erzeugen. Verbindet man jeweils einen Arm von zwei dreiarmigen Verzweigungen, entsteht ein vierarmiges Gebilde [26]. An die vier Enden können weitere Verzweigungen angelagert werden, so dass vielarmige Verbindungen ohne prinzipielle Stabilitätsprobleme möglich werden. Allerdings sind solche Strukturen aus einfachen Verzweigungen äußerst flexibel und haben deshalb nur wenig Anwendungen gefunden.

Neben diesen reinen DNA-Verzweigungen lassen sich Einzel- oder Doppelstränge aber auch über andere Stoffe miteinander verbinden. Zum Beispiel können über Biotin vier DNA-Stränge an ein Streptavidin-Molekül gebunden werden. Die Stränge sind dann tetraederartig um das Streptavidin angeordnet. Für die Herstellung vielarmiger Verzweigungen eignen sich sehr gut Goldcluster mit wenigen Nanometern Durchmesser. Thiolisierte DNA-Stränge können daran binden. Wie viel Arme in diesem Fall vom Verzweigungspunkt ausgehen, ist schwieriger zu bestimmen, kann aber unter anderem durch die Clustergröße beeinflusst werden. Die DNA-funktionalisierten Goldcluster können dann zum Beispiel zu einem Gold-Komposit verbunden werden [34, 42].

2.4.3 DNA-Netze

Aus den einfachen Verzweigungen können größere DNA-Strukturen zusammengesetzt werden. Eine große Gruppe solcher Strukturen sind zweidimensionale Netze. Meist werden sie erzeugt, indem mehrere einfache Verzweigungen zu einem oder mehreren Grundelementen zusammengefasst werden, welche sich dann über Einzelstrangüberhänge zum Netz verbinden. Die Grundelemente werden meist durch Selbstassemblierung erzeugt und gegebenenfalls durch Ligation stabilisiert. Das Zusammensetzen des Netzes erfolgt dann entweder auch durch Selbstassemblierung (das komplette Netz kann dann auch in nur einem Annealing-Schritt produziert werden) oder durch Ligation.

Eines der ersten DNA-Netze war ein Gitter aus Rhomben und Parallelogrammen [35]. Das Grundelement dieses Netzes ist ein Rhombus mit vier Helixwindungen Kantenlänge. Das entspricht 42 Basenpaaren und einer Länge von ca. 14 Nanometern. Es wird von zehn Einzelsträngen geformt. Vier vierarmige Verzweigungen bilden die Ecken des Rhombus. Entsprechend den Winkeln der Basisverzweigung sind die Winkel im Rhombus 60^o und 120^o. Das Element ist in Abbildung 2.13 dargestellt.

Die Kanten des Rhombus gehen über die Eckpunkte hinaus und enden in Einzelstrangüberhängen. Dadurch stehen insgesamt acht Verknüpfungspunkte zur Verfügung, an denen sich benachbarte Elemente verbinden können. Durch die geeignete Wahl der Basensequenzen der Überhänge wurde erreicht, dass sich die Grundelemente zu dem regelmäßigen Netz in Abbildung 2.14 zusammensetzten. Zwei benachbarte Rhomben liegen zwei Helixwindungen voneinander entfernt. In den Lücken entstehen Parallelogramme mit zwei und vier Windungen Seitenlänge und kleinere Rhomben mit zwei Windungen Seitenlänge. Betrachtet man das Gitter als einen zweidimensionalen Kristall, so besteht dessen Elementarzelle aus einem großen und einem kleinen Rhombus sowie zwei Parallelogrammen. Die Ausdehnung der Elementarzelle beträgt inklusive den Überhängen acht Helixwindungen.

Die erzeugten Netze dieser Gestalt sind mehrere Mikrometer breit und hoch. Eine Variante mit veränderten Verknüpfungspunkten führte zu einer eindimensionalen Verbindung des Grundelementes. Es entstand so eine Kette aus alternierenden Rhomben und Parallelogrammen. Das Konstrukt ist nur sechs Helixwindungen (63 bp, 21 nm) breit,

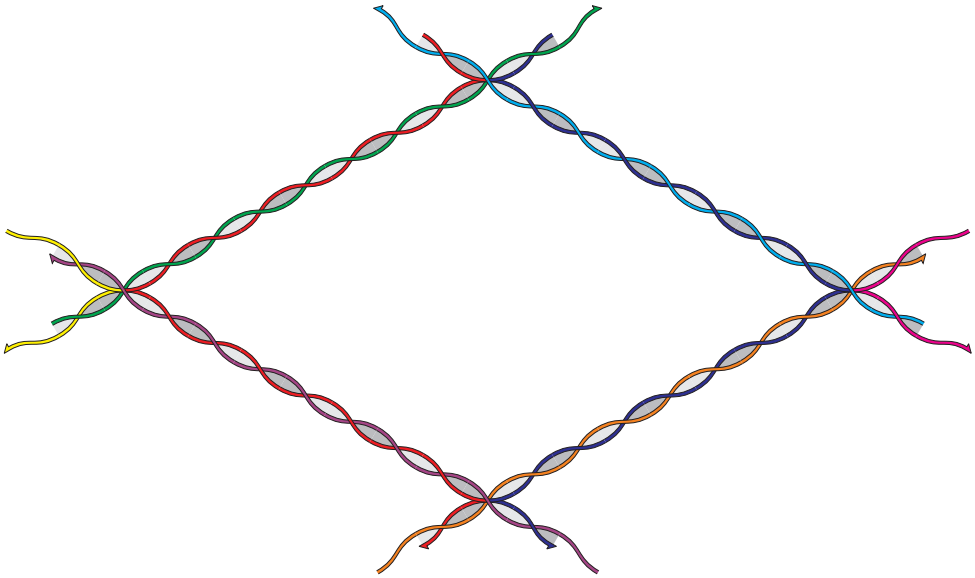


Abbildung 2.13: Grundelement für das Rhombus-Netz

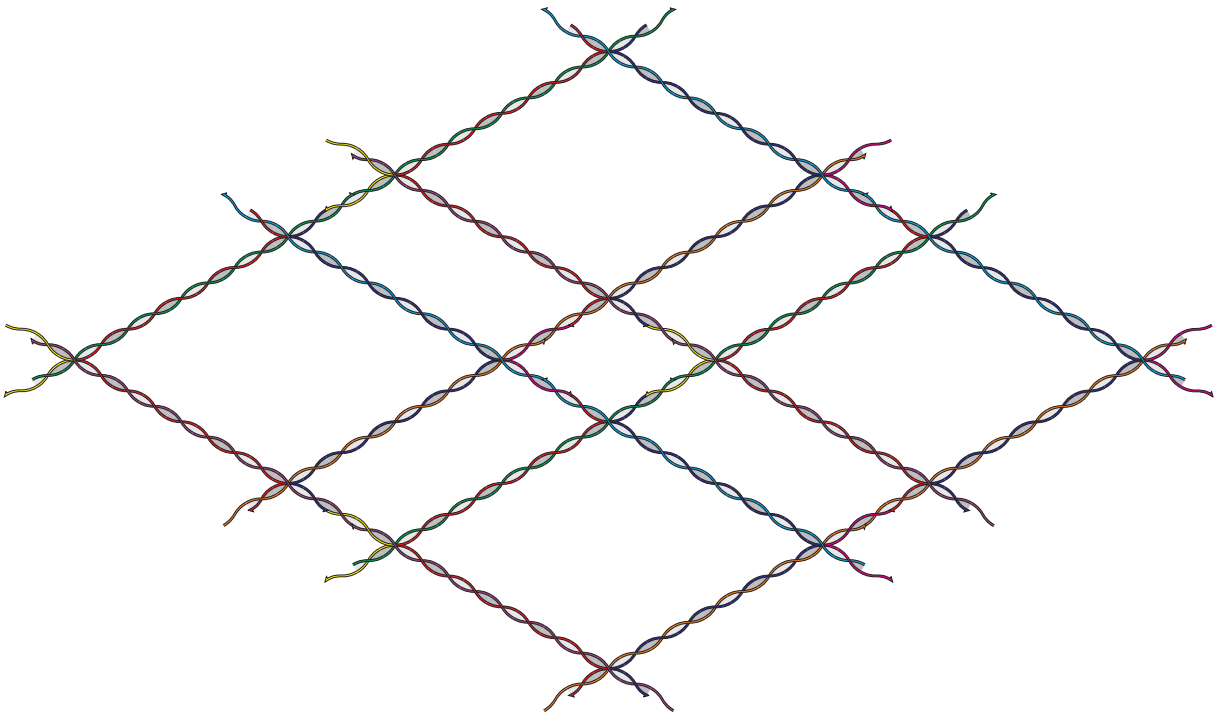


Abbildung 2.14: Gitter aus Rhombus-Grundelementen

Abbildung 2.15: Darstellung eines Double-Crossover-(DX)-Moleküls

trotzdem aber mehrere Mikrometer lang.

Völlig anders geartete Grundelemente sind die Double-Crossover-Moleküle (DX) [33]. Sie bestehen aus zwei Doppelhelices, die an zwei Kreuzungspunkten Einzelstränge untereinander austauschen. Nach außen hat das Molekül vier Strangenden. Es lässt sich also auch als eine, wenn auch komplexere, vierarmige Verzweigung betrachten. Anhand der Lage der Kreuzungspunkte können mehrere Varianten des DX-Moleküls unterschieden werden [33]. Eine oft verwendete Form ist in Abbildung 2.15 dargestellt. Bei dieser sind die beiden Kreuzungspunkte zwei Helixwindungen voneinander entfernt. Es gibt einen oberen und einen unteren Seitenstrang (rot und orange), welche beide durch das gesamte Molekül laufen. Verbunden sind sie durch drei innere Stränge (hellblau, grün und dunkelblau), wobei der mittlere grüne Strang einen Ring formt.

Durch die Verknüpfung der beiden Doppelhelices ist ein DX-Molekül sehr viel steifer als eine einfache Verzweigung. Es lässt sich daher sehr gut zur Konstruktion größerer Strukturen verwenden. Gesteuert wird die Konstruktion durch die Basensequenzen an den Einzelstrangüberhängen der vier äußeren Arme des DX-Elementes. In Abbildung 2.16 ist als Beispiel ein Netz aus DX-Molekülen der Abbildung 2.15 zu sehen.

Bei diesem Arrangement sind die Sequenzen der Einzelstrangüberhänge so definiert, dass ausgehend von Abbildung 2.15 das obere linke Ende an das obere rechte und das untere linke Ende an das untere rechte Ende eines benachbarten Elementes binden kann. Die Kreuzungspunkte benachbarter Elemente liegen zweieinhalb Helixwindungen voneinander entfernt. Deshalb sind die Elemente alternierend um jeweils 180^o um ihre Längsachse zueinander verdreht. Kleinere Verkrümmungen, die ein DX-Molekül aufweisen kann, werden dadurch ausgeglichen. Da es im dargestellten Netz nur einen Typ Grundelement gibt, ist das wichtig, weil sich die leichte Krümmung bei gleicher Orientierung sonst aufsummieren würde.

Ein ähnliches Arrangement, allerdings mit vier unterschiedlichen DX-Elementen wurde in der Veröffentlichung [9] vorgestellt. Einige der DX-Elemente haben dabei in ihrem zentralen Ring eine zusätzliche Verzweigung, deren einer Arm aus der Gitterebene herausragt. An diesen Armen sind Goldcluster gebunden, so dass eine regelmäßigere Anordnung von Goldclustern erreicht werden konnte.

Mit einem etwas veränderten Design der Einzelstrangüberhänge des DX-Moleküls in Abbildung 2.15 lässt sich erreichen, dass sich die Grundelemente nicht zu einem Netz, sondern zu einer Kette formieren. Diese Konstellation ist in Abbildung 2.17 zu sehen. Auf diese Weise erhält man lang gestreckte, aber trotzdem noch recht steife DNA-Konstrukte, die zum Beispiel als Vorlage für Nanodrähte benutzt werden können.

Ein weiteres Grundelement ist das 4X4-Molekül [18]. Es besteht aus vier vierarmigen Verzweigungen, von denen jeweils zwei Arme über einen zentralen Ringstrang miteinander verbunden sind. Die übrigen Arme der Verzweigungen (insgesamt acht) weisen nach außen und stellen Verknüpfungsstellen zur Verfügung. Zwischen zwei benachbarten Verzweigungen ist in den Ringstrang eine Schlaufe aus vier ungepaarten Thymin-Basen eingebaut.

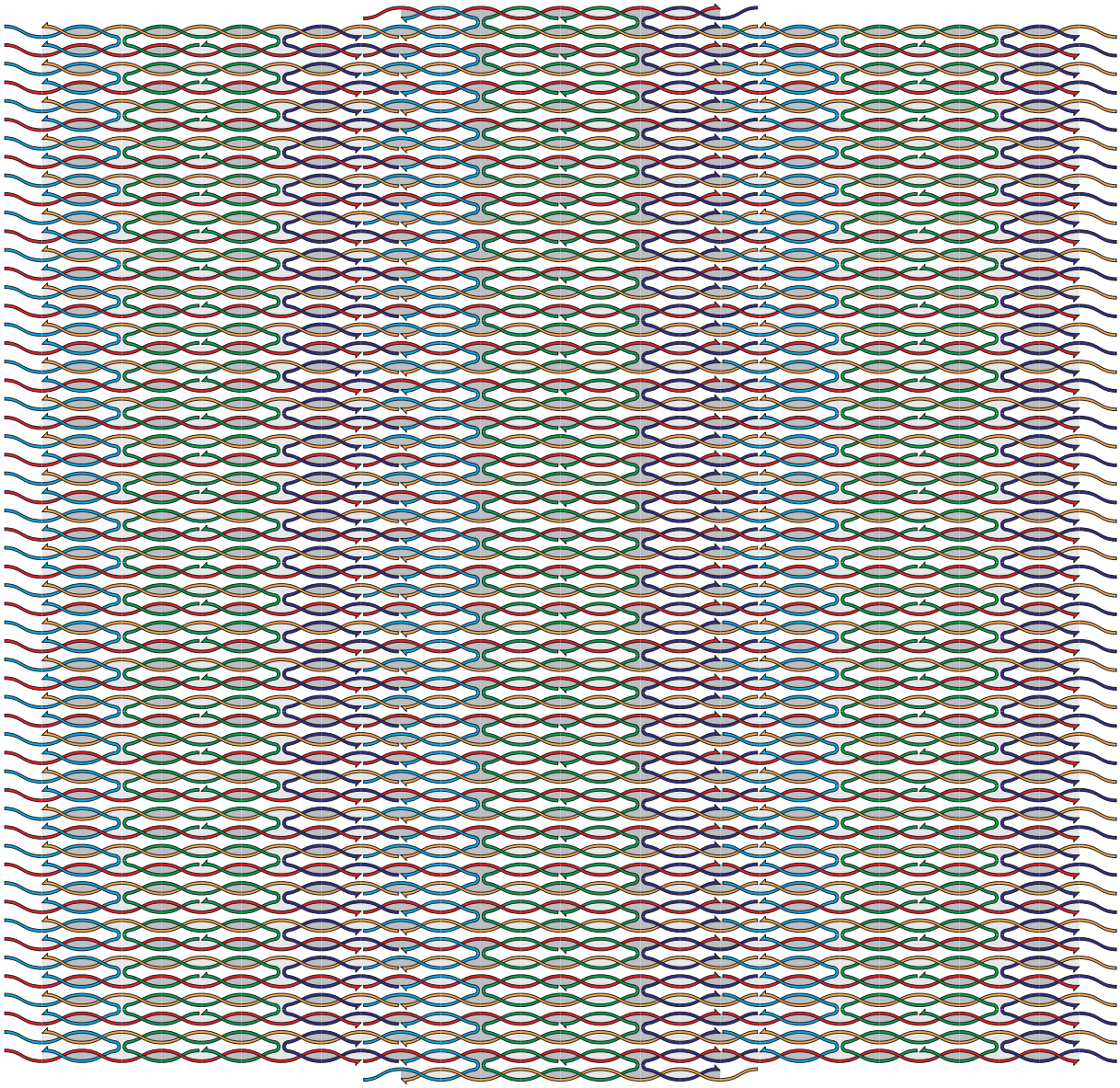


Abbildung 2.16: Netz aus DX-Molekülen

Abbildung 2.17: Darstellung einer Kette aus DX-Molekülen

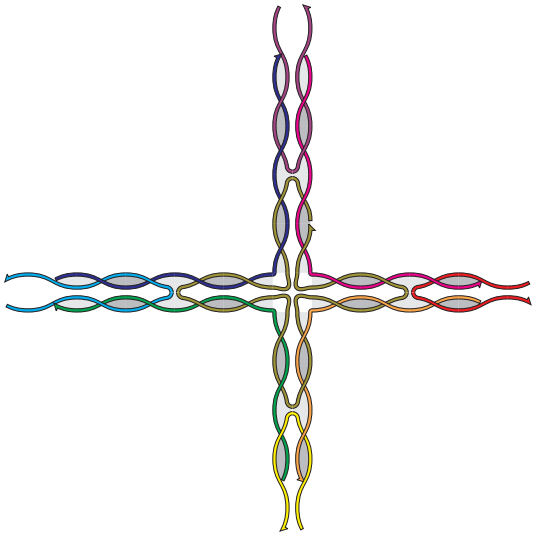


Abbildung 2.18: Darstellung des 4X4-Moleküls

Die Doppelhelices zwischen den Verzweigungspunkten werden dadurch unterbrochen und können an diesen Stellen knicken. Das Molekül erhält dadurch die Form eines Kreuzes. Eine Darstellung des 4X4-Moleküls ist in Abbildung 2.18 zu sehen.

Durch geeignete Wahl der Sequenzen der Einzelstrangüberhänge kann wiederum ein Netz konstruiert werden. In der Veröffentlichung [18] wurde ein Netz aus einem Grundelement generiert. Die beiden unteren Enden des Elementes verbinden sich dabei mit den beiden oberen Enden und die beiden linken mit den beiden rechten. So entsteht ein regelmäßiges Quadratgitter, welches in Abbildung 2.19 zu sehen ist. Bei eingehender Betrachtung wird ersichtlich, dass durch die Verknüpfung der einzelnen Grundelemente wieder DX-Elemente ähnlich denen aus Abbildung 2.15 entstehen. Diese bilden die Kanten des Quadratgitters. Die Knoten sind die durch die Thymin-Schlaufen herbeigeführten Knicke.

Jedes 4X4-Element wurde zusätzlich in seiner Mitte mit einem Protein besetzt, so dass ein regelmäßiges Protein-Gitter entstand. In der gleichen Arbeit wurde auch eine abgewandelte Form des Netzes vorgestellt, das in der einen Dimension nur zwei Grundelemente breit ist. Durch Metallisierung [13, 27] dieser Anordnung entstanden Nanodrähte.

Mit einer veränderten Variante des 4X4-Elements, welches nur drei Arme besitzt (sozusagen ein 3X4-Element), hat man auf ähnliche Weise ein Hexagon-Netz erzeugt [6].

Auch irreguläre Netze oder Graphen lassen sich aus unterschiedlichen Grundelementen herstellen, wie in der Arbeit [11] gezeigt wurde.

2.4.4 3D-Strukturen

Einzelne Grundelemente lassen sich auch zu dreidimensionalen Objekten zusammensetzen. Wenn man zum Beispiel vier geeignete einfache dreiarmige Verzweigungen wie in Abbildung 2.11 miteinander verknüpft, kann ein Tetraeder wie in Abbildung 2.20 entstehen. In [4] wurde der dargestellte Tetraeder sowie einige weitere Abwandlungen erzeugt.

Um den Grundelementen die für diese Konstruktion notwendige hohe Flexibilität zu

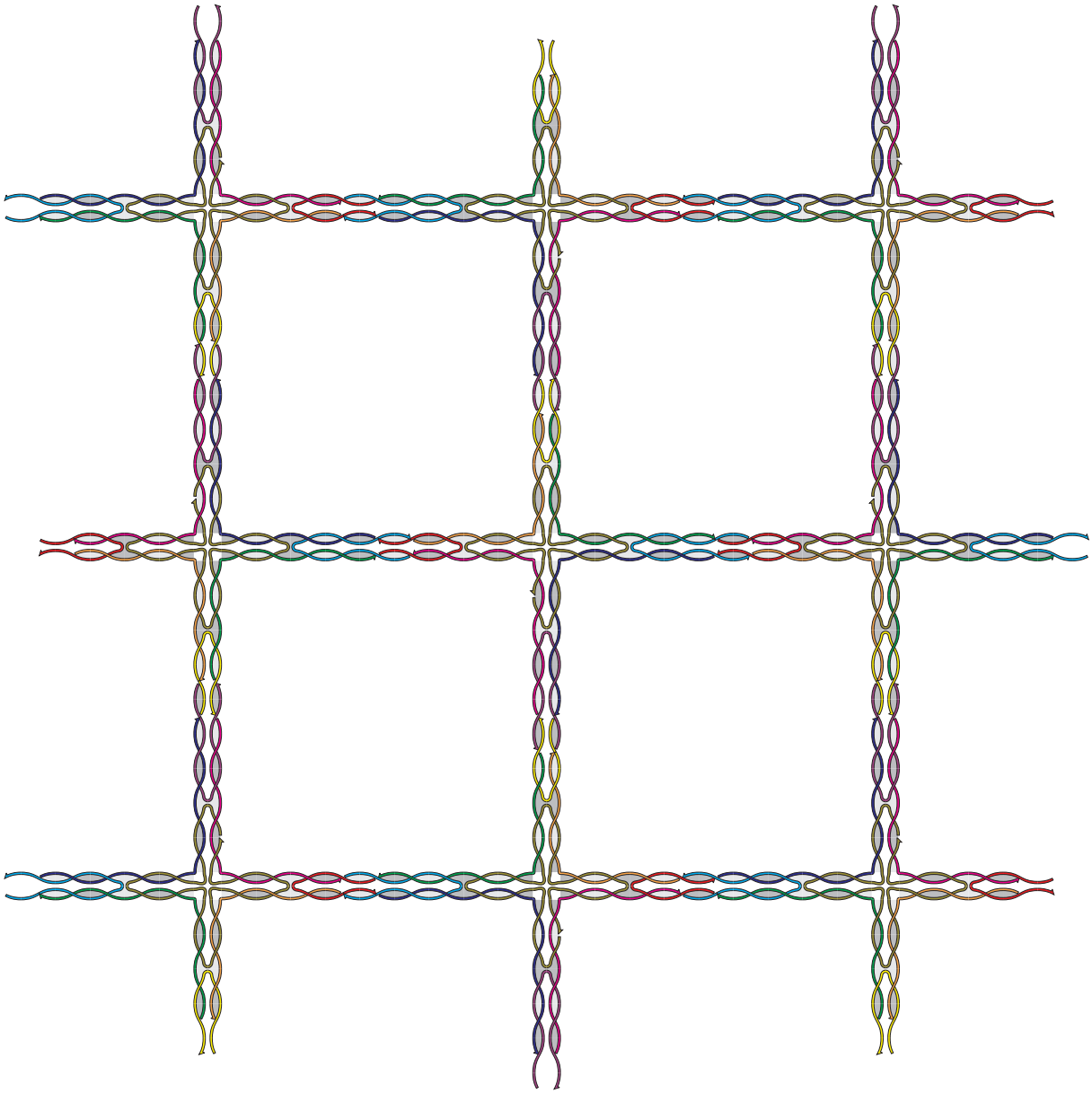


Abbildung 2.19: Netz aus 4X4-Molekülen

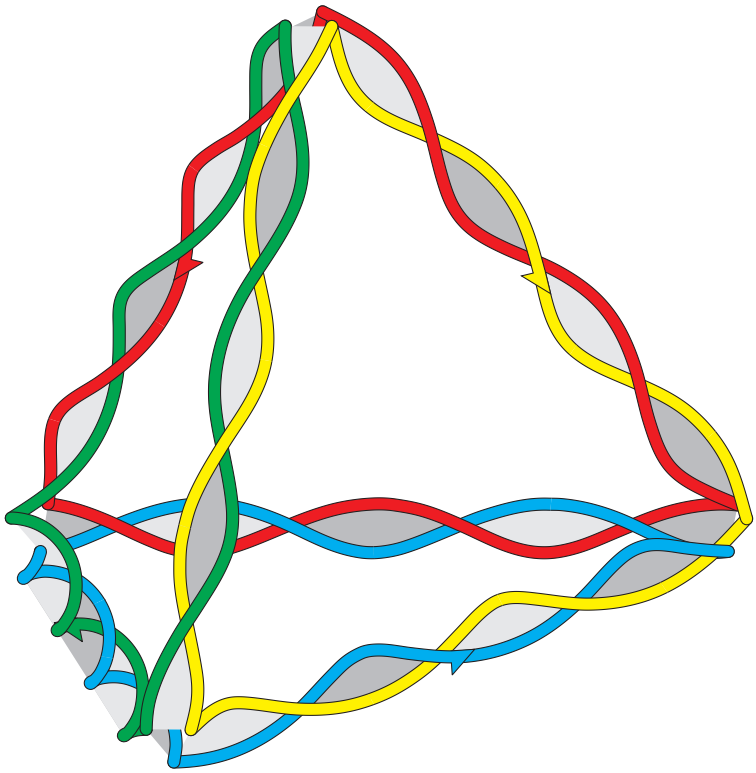


Abbildung 2.20: Tetraederförmiges DNA-Molekül

geben, sind um die Verzweigungspunkte herum jeweils drei ungepaarte Basen eingebaut. Diese ermöglichen den Armen eine hohe Beweglichkeit in alle Raumrichtungen. Der gesamte Tetraeder ist durch seine geometrische Form wieder formstabil.

Durch geeignete Verknüpfung von acht dreiarmigen Verzweigungen entsteht auf ähnliche Weise ein Würfel oder Quader [44]. Einige weitere dreidimensionale Objekte konnten bisher erfolgreich synthetisiert werden, darunter Oktaeder [14], ein Oktaeder mit abgeschnittenen Ecken [43] und verschiedene Arten von Röhren [3, 19].

Die Röhrenkonstruktionen nutzen eine vorhandene Krümmung eines Bauelementes. In der Arbeit [3] falten sich mehrere Kopien des gleichen Stranges zu einem DX-Molekül zusammen. Diese Grundelemente verbinden sich wiederum zu einem Band mit einer Breite von einigen wenigen DX-Elementen. Durch die Krümmung der Elemente und ihrer immer gleichen Orientierung krümmt sich das Band zu einer Einfachhelix. So entsteht eine Röhre mit einem Durchmesser von 20 bis 30 Nanometern. In der zweiten Arbeit [19] werden Triple-Crossover-Moleküle [33] als Grundelemente verwendet. Dies sind Moleküle, die drei Doppelhelices durch vier Kreuzungspunkte verbindet. Auch TX-Moleküle weisen eine Eigenkrümmung auf. Bei Erzeugung eines Gitters ohne alternierende Orientierung der Grundelemente führt das zu einer Röhrenbildung.

Kapitel 3

Design von DNA-Strukturen

Im vorangegangenen Kapitel wurde beschrieben, wie mit der DNA Nanostrukturen aufgebaut werden. Dieses Kapitel widmet sich nun den Herausforderungen des Designs solcher Strukturen.

Der Aufbau einer DNA-Nanostruktur erfolgt in drei Schritten:

1. Strukturdesign,
2. Sequenzdesign und
3. Herstellung der Struktur im Labor.

Ein vierter Schritt wird notwendig, wenn das Ergebnis verifiziert werden soll, was bei wissenschaftlichen Arbeiten natürlich immer notwendig ist.

In der Strukturdesign-Phase wird geklärt, welche DNA-Struktur erzeugt werden soll und wie diese exakt aussehen muss. Dabei spielen die Materialeigenschaften der DNA, ihre Flexibilität und ihre Windungseigenschaften sowie die angestrebte Funktionalität der Struktur eine Rolle. Im Kapitel 2.4 war zu sehen, dass die verwendeten Bauelemente immer komplexer werden. Benutzte man anfangs einfache drei- und vierarmige Verzweigungen, kommen nun verstärkt Crossover-Moleküle oder noch komplexere Gebilde, wie zum Beispiel das 4X4-Element, zum Einsatz. Man verwendet ungepaarte Basen, um bestimmte Flexibilitäten (beim Tetraeder) oder Knicke (T-Schlaufen beim 4X4-Element) zu erzeugen. Die Vielgestaltigkeit der Strukturen nimmt immer mehr zu.

Die zweite Phase ist das Sequenzdesign. Dort wird die Frage geklärt, wie die Basensequenzen für die gewünschte Zielstruktur aussehen müssen. Durch die Basensequenzen wird die Gestalt der entstehenden Strukturen vorherbestimmt, weil sie bei der Selbstassemblierung als Programmierung fungieren. Da die Strukturen immer komplexer werden, steigen auch die Anforderungen an die Basensequenzen. Die Strukturen enthalten immer mehr Verzweigungspunkte verschiedenster Art.

Schon lange kann die Aufgabe des Sequenzdesigns ohne Unterstützung von Computern nicht mehr effektiv behandelt werden. Ältere Algorithmen stoßen aber zunehmend an ihre Grenzen, da sie für einfachere Anwendungsfälle entwickelt wurden.

Im Folgenden wird nun zuerst das Strukturdesign betrachtet und dann das Hauptaugenmerk auf das Sequenzdesign gerichtet.

3.1 Strukturdesign

Die Gestalt einer DNA-Struktur wird zu allererst von ihrem Zweck, das heißt der angestrebten Funktionalität, bestimmt. Benötigt man nur einen einfachen Abstandshalter, genügt eine einfache Doppelhelix mit der entsprechenden Länge. Möchte man dagegen ein DNA-Netz erzeugen, um darauf zum Beispiel Proteine oder Goldcluster zu immobilisieren, benötigt man Grundelemente mit Verzweigungsstellen und funktionalisierten Nukleotiden.

Grundsätzlich stehen drei Strukturelemente zur Verfügung: Verzweigungspunkte, Einzelstränge und Doppelstränge. An den Verzweigungspunkten können theoretisch beliebig viele, praktisch bis zu sechs und üblicherweise drei oder vier Doppelstränge miteinander verknüpft werden. Sie entstehen durch Hybridisierung von Einzelsträngen, wobei jeder Strang an mindestens zwei Armen teilnimmt. Alternativ kann die Verbindung auch durch Nicht-DNA-Moleküle (z. B. Streptavidin) erfolgen. Einfache drei- und vierarmige Verzweigungen (siehe Kap. 2.4.2) sind relativ flexibel. Sie besitzen zwar eine bevorzugte Gestalt (dreiarmig: 120^o, vierarmig: $120^o/60^o$), verbiegen sich aber ohne weitere Stabiliserung sehr leicht. Durch die Verknüpfung mehrerer einfacher Verzweigungen durch Doppelstrangabschnitte können sehr viel starrere Gebilde, wie zum Beispiel die DX-Moleküle, entstehen (siehe Kap. 2.4.3). Andererseits erhöhen ungepaarte Basen direkt am Verzweigungspunkt dessen Flexibilität noch erheblich. Das ist manchmal nötig, um eine gewünschte Gestalt zu erhalten, wie zum Beispiel das Tetraeder in Kapitel 2.4.4. Ohne die erhöhte Flexibilität der Verzweigungspunkte würde die Zielstruktur entweder gar nicht entstehen können oder aber auf Grund der hohen Torsionskräfte sehr instabil sein.

Doppelstränge können bis zu einer Länge von 150 Basenpaaren (50 nm) als starr angesehen werden. Sie bilden deshalb das Grundgerüst einer DNA-Struktur. Ihre Gestalt ist die einer Doppelhelix. Welche Form die Helix genau hat, hängt von den Umgebungsbedingungen und teilweise auch von der Basensequenz ab (siehe Kap. 2.1). In den meisten Fällen wird die B-Form mit einer Windungshöhe von 10.4 Basenpaaren (3.4 nm) vorliegen. Die Windungsphase entscheidet darüber, in welcher Ebene einzelne Arme an Verzweigungspunkten abzweigen. Für zweidimensionale Strukturen ist eine halbe Helixwindung die Basis-Abstandseinheit. Verzweigungspunkte werden ihre Arme dann in annähernd der gleichen Ebene ausstrecken. Vierteldrehungen führen dagegen zu Ausrichtungen in die dritte Dimension. Designt man Strukturen, die von ihrer Art her eigentlich in einer Ebene liegen (z. B. ein DX-Molekül), achtet aber nicht auf die halben Helixwindungen, entstehen Torsionen in dem Molekül, was zu einer verminderten Stabilität führt.

Einzelstränge sind sehr flexibel. Sie sind eine Art Perlenkette aus aufgereihten Nukleotiden. Man setzt sie ein, um gezielt Flexibilität in ein Molekül zu bringen.

Flache DNA-Komplexe wie das DX-Molekül weisen meist eine leichte Krümmung auf. Setzt man aus ihnen größere Netze zusammen, werden auch diese sich in eine Richtung biegen. Dieses Designelement wird benutzt, um räumliche Strukturen zu erzeugen. Ist die Krümmung jedoch nicht erwünscht, kann man sie neutralisieren, indem benachbarte Grundelemente mit jeweils entgegengesetzter Orientierung verbunden werden. Die Krümmung des einen Elements wird dann durch den Nachbarn ausgeglichen.

Eine noch tiefer gehende Übersicht über das Design von DNA-Strukturen ist in [2] nachzulesen.

3.2 Sequenzdesign

Nachdem geklärt ist, wie genau die Zielstruktur aussehen soll, stellt sich die Frage nach dazu passenden Basensequenzen. Die DNA-Struktur wird meist in einem oder mehreren Annealingschritten aufgebaut. Dabei bestimmen die Basensequenzen der Einzelstränge durch die Watson-Crick-Paarungen die Gestalt der entstehenden Struktur (Bottom-up Design). Unpassende Sequenzen führen zu unerwünschten Strukturen.

Die grundlegendste Anforderung ist, dass Sequenzen innerhalb eines Doppelstranges komplementär zueinander sein müssen. Das ist sehr einfach zu erfüllen. Wesentlich schwieriger ist es, dafür zu sorgen, dass außer diesen erwünschten keine weiteren unerwünschten Basenpaarungen stattfinden. Nun lassen sich Fehlpaarungen prinzipiell nicht vermeiden. Zum Beispiel kann jede Adenin-Base mit jeder Thymin-Base irgendwo in der DNA-Struktur eine Bindung eingehen. Ähnlich ist es mit kurzen Sequenzen, zum Beispiel CT. Die komplementäre Sequenz AG wird selbst bei kleinen Strukturen mehrfach auftreten. Anderenfalls wäre die Sequenzvariabilität zu sehr eingeschränkt. Die Stabilität solcher kurzer Fehlpaarungen ist allerdings im Vergleich zu den längeren erwünschten Doppelsträngen sehr gering (siehe Kap. 2.2). Gelänge es, die Länge unerwünschter Basenpaarungen auf ein bestimmtes Maß zu begrenzen, würde damit auch deren Auftrittswahrscheinlichkeit begrenzt. Sie würden zwar vorkommen, könnten aber die Erzeugung der Zielstruktur nicht wesentlich behindern.

Eine Methode, die Begrenzung der Fehlerlängen zu formulieren, ist das im Folgenden beschriebene Critonkonzept.

3.2.1 Das Critonkonzept

Das Critonkonzept wurde in den 1980er Jahren von Nadrian C. Seeman publiziert [48, 49] und in das Sequenzdesign-Programm SEQUIN [45] eingearbeitet. Mit Hilfe dieses Programmes wurden seither die Sequenzen für die meisten der DNA-Strukturen aus Kapitel 2.4 generiert. Es ist darum ein sehr bewährtes Konzept.

Ein Criton ist ein Abschnitt auf einem DNA-Einzelstrang mit einer fest definierten Länge, zum Beispiel 3. Diese feste Länge heißt Critonlänge und soll im Folgenden mit L_C bezeichnet werden. Jeder Einzelstrang der DNA-Struktur wird vollständig in Critons zerlegt, und zwar so, dass sich benachbarte Critons um $L_C - 1$ Basen überlappen. Daraus folgt, dass ein Einzelstrang der Länge l in $l - (L_C - 1)$ Critons der Länge L_C zerfällt, so zum Beispiel der acht Basen lange Strang $5' - \mathrm{TTCTCGGA} - 3'$, der aus den sechs Critons TTC, TCT, CTC, TCG, CGG und GGA zusammengesetzt ist.

Alle Critons haben einen Vorgänger und einen Nachfolger auf dem Strang, außer natürlich die jeweils ersten und letzten, bei denen einer der Nachbarn fehlt. Critons, die sich auf den beiden Einzelsträngen eines Doppelstranges gegenüberliegen, werden als zueinander komplementär bezeichnet. Ihre Sequenzen sind wegen der Watson-Crick-Basenpaarungen immer komplementär.

Eine DNA-Struktur mit passender Basensequenz muss die folgenden vier Critonregeln erfüllen:

1. Die Sequenz jedes Critons existiert nur ein einziges Mal in der gesamten DNA-Struktur.

2. Die komplementären Sequenzen von Critons, die nicht vollständig und ununterbrochen in ein und demselben Doppelstrang liegen, kommen nicht vor.

3. Selbstkomplementäre Sequenzen existieren nicht. Ist die Critonlänge L_C eine ungerade Zahl, kommen auch keine selbstkomplementären Sequenzen der Länge $L_C + 1$ vor.

4. In dem Basenpaar-Ring eines Verzweigungspunktes kommt jeder Basenpaar-Typ (G/C, C/G, A/T und T/A) höchstens zweimal vor und befindet sich dann auf benachbarten Armen.

Was ist mit diesen Critonregeln gewonnen? Die Critons repräsentieren alle Subsequenzen der Länge L_C in der DNA-Struktur. Es gilt, die Länge unerwünschter Paarungsstellen so gering wie möglich zu halten. Wenn nun, wie in Critonregel 1 gefordert, jede Subsequenz der Länge L_C höchstens einmal in der gesamten Struktur vorkommt, so können die Sequenzen zweier komplementärer Critons in einem Doppelstrangabschnitt nirgendwo ein zweites Mal auftauchen. Es gibt deshalb keine Sequenz länger als $L_C - 1$, welche ebenfalls die Sequenz des Doppelstranges hat.

Es gibt aber auch Critons, die nicht in Doppelsträngen liegen. Das betrifft Critons in Einzelstrangabschnitten, an Verzweigungspunkten und gegenüber von Lücken im Strangrückgrat. Ihre Komplementärsequenzen sollen gar nicht vorkommen. Critonregel 2 verbietet diese deshalb.

Während des Aufbauprozesses liegt jeder Strang tausend- und millionenfach in der Lösung vor. Bei selbstkomplementären Sequenzen (z. B. GATC) könnten zwei Kopien desselben Stranges miteinander eine Bindung länger als $L_C - 1$ Basen eingehen. Deshalb untersagt Regel 3 selbstkomplementäre Sequenzen. Der Zusatz für ungerade Critonlängen in dieser Regel ist notwendig, da Sequenzen mit ungerader Länge nie selbstkomplementär sind, aber trotzdem größere selbstkomplementäre Abschnitte formen können. Betrachten wir als Beispiel den Einzelstrang $5' - \text{TTGATCCG} - 3'$. Er wird in die Critons TTG, TGA, GAT, ATC, TCC und CCG zerlegt. Keines dieser Critons ist selbstkomplementär, trotzdem weist der Strang die Sequenz GATC auf, die eine Verbindung mit Länge 4 ermöglicht. Deshalb müssen für ungerade Critonlängen auch die nächst größeren geraden Sequenzlängen getestet werden.

Die ersten drei Critonregeln stellen also zusammen sicher, dass unerwünschte Paarungen auf Längen unterhalb der Critonlänge beschränkt sind. Ist die Critonlänge klein genug, ist die Stabilität der Fehlstellen sehr gering.

Regel 4 geht über die Begrenzung der Fehlstellen hinaus und verhindert mobile Verzweigungspunkte. In Abbildung 3.1 ist eine Verzweigung zu sehen, welche diese Regel verletzt. In der linken Konfiguration besteht der Basenpaar-Ring (alle Basenpaare direkt neben dem Verzweigungspunkt) aus den Basenpaaren C/G, A/T, G/C und nochmals A/T. Das Paar A/T kommt also zweimal vor, was noch kein Problem darstellt, liegt aber auf den nicht benachbarten Armen 2 und 4. Dadurch ist diese Verzweigungsstelle nicht die einzig mögliche Konfiguration. Das A auf Arm 2 kann sich auch mit dem T auf Arm 4 verbinden, ebenso ihre ursprünglichen Partner. Eines der beiden neuen Paare würde dann auf den ersten, das andere auf den dritten Arm wandern. Abhängig von den jeweiligen Sequenzen kann sich dieser Prozess weiter fortsetzen. Der Vorgang ist in Abbildung 3.1 dargestellt, wo in der rechten Konfiguration die Arme 1 und 3 um jeweils drei Basenpaare

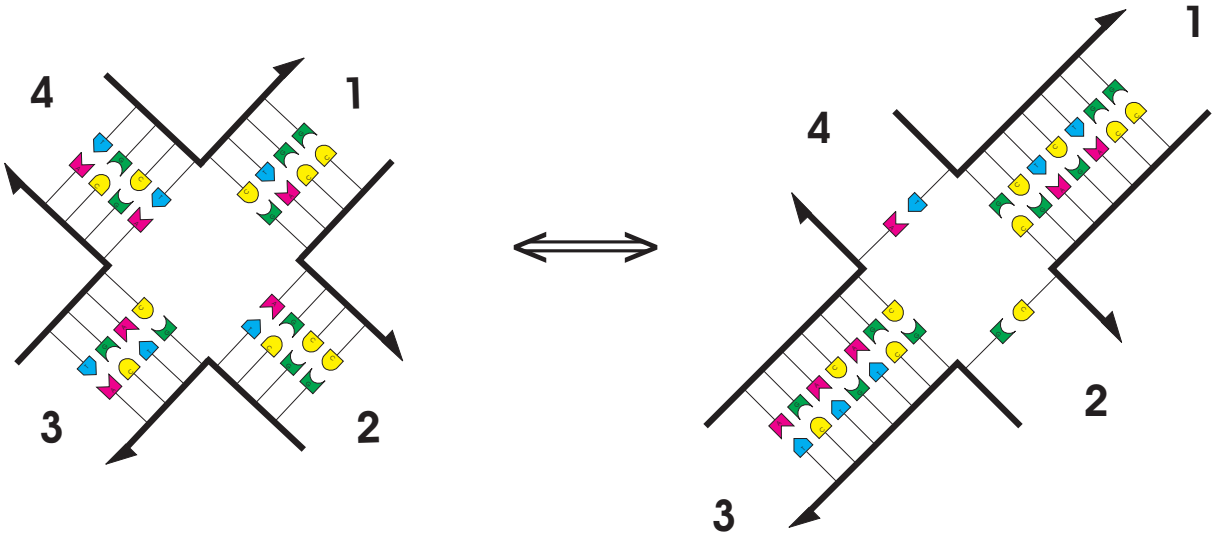


Abbildung 3.1: Mobiler Verzweigungspunkt

verlängert und die beiden anderen Arme entsprechend verkürzt sind. Eine erfüllte Regel 4 verhindert diese Mobilität.

An dieser Stelle soll auf die Besonderheiten von drei- und vierarmigen Verzweigungen hingewiesen werden. Eine vierarmige Verzweigung, die gegen Regel 4 verstößt, verletzt automatisch auch Regel 2. Betrachtet man dazu wieder die linke Konfiguration in Abbildung 3.1. Das Criton, welches mit zwei Basen auf dem ersten Arm und mit einer Base auf dem zweiten Arm liegt, besitzt die Sequenz AGA. Das Criton, welches mit einer Base auf Arm 4 und mit zwei Basen auf Arm 1 liegt, hat die Sequenz TCT. Die Sequenzen sind komplementär zueinander. Da aber beide Critons nicht vollständig und ununterbrochen in genau einem Doppelstrangabschnitt liegen – der Verzweigungspunkt ist eine Unterbrechung – dürften ihre komplementären Sequenzen laut Regel 2 überhaupt nicht vorkommen. Der gleiche Effekt tritt bei einer dreiarmigen Verzweigung auf, bei der ein Basenpaar zweimal vorkommt. Der Basenpaar-Ring wäre in diesem Fall zwar korrekt, Regel 2 würde jedoch verletzt werden. Für drei- und vierarmige Verzweigungen könnte man also auf einen Test der 4. Critonregel verzichten.

3.2.2 Weitere Anforderungen an Sequenzen

Das Critonkonzept ist eine gute Basis, um passende Sequenzen für einen Laborversuch zu generieren. Es gibt jedoch noch weitergehende Anforderungen, die, abhängig von der zu erzeugenden Struktur, wünschenswert sein könnten.

Es hat sich als günstig herausgestellt, die Enden eines Doppelstranges, soweit sie nicht an einem Verzweigungspunkt liegen, mit einem G/C-Basenpaar abzuschließen. Ein Aufspalten des Doppelstranges wird dadurch erschwert, weil die G/C-Bindung stärker als die A/T-Bindung ist. Ebenfalls als günstig hat sich erwiesen, Abschnitte, in denen mehr als zwei Guanin-Basen hintereinander auftreten, zu unterdrücken. Auch hier ist die hohe Bindungskraft des G/C-Paares die Ursache. Längere Guanin-Abschnitte gehen auch Bindungen mit fast komplementären Sequenzen (z. B. CCAC) ein und erzeugen dadurch

Fehlpaarungen. Es kann auch wünschenswert sein, den G/C-Anteil eines Strangabschnittes festlegen zu können, denn der G/C-Anteil eines Doppelstranges hat wesentlichen Einfluss auf dessen Stabilität (siehe Kap. 2.3.2).

Manchmal wird es erforderlich sein, bestimmte Sequenzen in der DNA-Struktur vorzudefinieren, so zum Beispiel die Erkennungssequenzen von Restriktionsenzymen (siehe Kap. 2.3.4) oder auch einzelne funktionalisierte Basen (siehe Kap. 2.3.6). Diese selbstdefinierten Sequenzen können möglicherweise gegen die harten Critonregeln verstoßen, insbesondere dann, wenn es sich um selbstkomplementäre Sequenzen handelt, wie es häufig bei Restriktionsenzymen der Fall ist. Sequenzen, die den Critonregeln genügen, könnten dann auf keinen Fall erzeugt werden. Es muss daher auch eine Möglichkeit geben, bestimmte Gebiete in der DNA-Struktur gezielt zu maskieren, um anzuzeigen, dass dort Verletzungen der Critonregeln möglich sind.

3.2.3 Anforderungen an den Sequenzdesign-Algorithmus

Aus den vorangegangenen Kapiteln ergibt sich eine Reihe von Anforderungen an einen Sequenzdesign-Algorithmus. Er muss beliebige DNA-Strukturen handhaben können, die Sequenzen nach den Critonregeln generieren und verschiedene wünschenswerte Nebenaspekte beachten. Die folgende Auflistung fasst die Anforderungen zusammen:

1. Der Algorithmus muss jede beliebige DNA-Struktur erfassen und behandeln können. Es ist dabei jedoch nicht notwendig, die angegebene Zielstruktur auf ihre Sinnhaftigkeit zu überprüfen.

2. Die erzeugten Sequenzen entsprechen den Critonregeln.

3. Sequenzen können an beliebigen Stellen in der Zielstruktur vordefiniert werden. Die Vordefinition kann explizit erfolgen (Base x in Strang y ist G), aber auch Variablilität enthalten (Base x in Strang y ist G oder C).

4. Beliebige Sequenzen können für die gesamte Struktur verboten werden.

5. Der G/C-Basenpaaranteil eines Doppelstranges kann in bestimmten Grenzen vordefiniert werden.

6. Beliebige Regionen der Zielstruktur können von den Critonregeln befreit werden.

7. Eine Sequenz kann als selbstkomplementär definiert werden.

In dieser Auflistung bilden die ersten beiden Punkte den Kern. Wichtig ist, dass jede theoretisch denkbare Zielstruktur behandelt werden kann. Das macht den Algorithmus unabhängig von konkreten Anwendungsfällen. Mit Erfüllung der Critonregeln ist sichergestellt, dass die Zielstruktur (wenn sie sinnvoll gestaltet ist) im Labor auch wirklich erzeugt werden kann. Alle weiteren Anforderungen sind Zusatzforderungen, welche sich in den konkreten Fällen als nützlich bzw. notwendig erwiesen haben. Für andere zukünftige Anwendungsfälle könnten sich noch weitere Nebenbedingungen ergeben.

Die bisherigen Sequenzdesign-Algorithmen bzw. -Programme sind bezüglich der Anforderungen noch nicht zufriedenstellend. Die meisten können nicht beliebige DNA-Strukturen behandeln, sondern sind nur für einfache lineare Sequenzen anwendbar

[21, 28, 36, 55]. Nur zwei Programme sind für verzweigte DNA-Strukturen benutzbar: das schon erwähnte SEQUIN [45] und ein neueres Programm namens TileSoft [16]. Mit SEQUIN wurden in den letzten Jahren die Sequenzen für fast alle DNA-Struktur-Experimente generiert. Allerdings ist dieses Programm nur teilautomatisch und erfordert noch sehr viel Arbeit vom Benutzer. Zudem stößt es bei einigen neueren Strukturen an seine Grenzen. TileSoft scheint diese Unzulänglichkeiten nicht aufzuweisen. Allerdings steht von diesem Programm noch keine allgemein zugängige Version zum Testen zur Verfügung.

3.2.4 Das Basissequenz-Konzept

Das Kernstück des im nächsten Kapitel beschriebenen Algorithmus wird sein, die erzeugten Sequenzen auf die Critonregeln zu testen. Der Test wird im Wesentlichen darin bestehen zu klären, ob ein Criton seine aktuelle Sequenz annehmen darf oder nicht. Da diese Frage sehr oft zu beantworten sein wird, braucht man eine schnelle und effiziente Methode dafür.

Der erste und einfachste Ansatz wäre, die Sequenzen aller Critons in eine Liste zu schreiben und bei jedem Neueintrag zu prüfen, ob diese Sequenz korrekt ist (wurde sie schon benutzt, ist sie selbstkomplementär etc.). Bei jedem Test müsste dabei allerdings die gesamte immer größer werdende Liste abgeglichen werden, was einen zu großen Suchaufwand bedeutet.

Organisiert man die Liste in Form eines Wörterbuches, verringert sich der Suchaufwand erheblich und beträgt dann unabhängig von der Größe der Liste nur noch maximal L_C Schritte, wenn L_C die Critonlänge ist.

Es gibt jedoch eine noch schnellere Methode, die zudem noch zusätzliche positive Effekte mit sich bringt und auf sogenannten Basissequenzen beruht [21,36]. Basissequenzen sind alle denkbaren Sequenzen einer bestimmten Länge L_B. Im Gegensatz zu den Sequenzen der Critons sind die Basissequenzen unveränderlich, können also in einer konstanten Struktur festgehalten werden. Alle Basissequenzen werden deshalb in einem Sequenzgraphen zusammengefasst und durch Nachbarschaftsbeziehungen miteinander verknüpft. Zwei Basissequenzen sind dann benachbart, wenn sie sich um die verminderte Basissequenzlänge $L_B - 1$ Basenwerte überlappen. Dadurch entstehen Vorgänger und Nachfolger. So ist zum Beispiel die Basissequenz GGG der Nachfolger von AGG, TGG, CGG und von sich selbst, aber auch der Vorgänger von GGA, GGT, GGC und wiederum auch von sich selbst.

Der Sequenzgraph besteht demzufolge aus den Basissequenzen als Knoten und den Nachfolgebeziehungen als Kanten. Er kann notiert werden als $SG = (V, E)$ mit der Knotenmenge

$$V = \{v = v_0 \dots v_{L_B-1} | v_i \in \{\text{G, A, T, C}\}, 0 \leq i < L_B\}$$

und der Kantenmenge

$$E = \{(v, w) | v, w \in V \cup (v_1 \dots v_{L_B-1}) = w_0 \dots w_{L_B-2}\}.$$

L_B bezeichnet die Basissequenzlänge.

Abbildung 3.2 zeigt einen Sequenzgraphen mit Basissequenzlänge $L_B = 2$.

Ein Pfad durch den Graphen über l_p jeweils benachbarte Basissequenzen repräsentiert eine Sequenz der Länge $l = l_p + (L_B - 1)$, zum Beispiel die eines Einzelstranges. Im

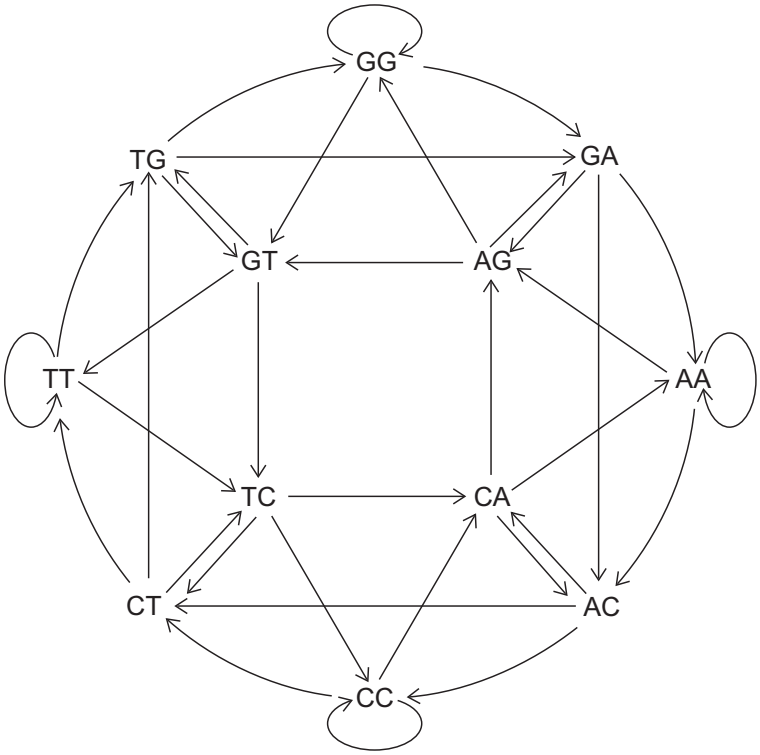


Abbildung 3.2: Darstellung eines Sequenzgraphen mit Basissequenzlänge 2

Critonkonzept (siehe Kap. 3.2.1) wurde jeder Einzelstrang in eine Kette sich ebenfalls überlappender Critons mit Länge L_C zerlegt. Derselbe l Basen lange Strang besteht dann aus $l-(L_C-1)$ Critons. Setzt man die Basissequenzlänge L_B gleich der Critonlänge L_C, so ist die Pfadlänge l_p gleich der Anzahl der Critons $l-(L_C-1)$, und man kann jedem Criton eine Basissequenz zuordnen. Eine in Critons zerlegte DNA-Struktur lässt sich so auch als eine Menge von Pfaden im Sequenzgraph betrachten.

Wie müssen nun die Pfade, welche die Sequenzen der Einzelstränge einer DNA-Struktur repräsentieren, aussehen, damit die Critonregeln erfüllt sind? Critonregel 1 fordert, dass die Sequenz jedes Critons nur einmal auftaucht. Für eine Basissequenz bedeutet dies, dass sie höchstens einmal in maximal einem Pfad Mitglied sein kann, oder anders formuliert, dass sich ein Pfad nicht mit sich selbst oder einem anderen Pfad überlagern darf. Um das gewährleisten zu können, bedarf es eines Attributes, welches festhält, ob eine Basissequenz schon in einem Pfad benutzt wird oder nicht. Wird bei der Sequenzgenerierung einem Criton C auf einem Einzelstrang S eine Sequenz zugewiesen, so wird die entsprechende Basissequenz X dem zum Einzelstrang gehörenden Pfad p_S zugeordnet. Soll Critonregel 1 erfüllt sein, geht das nur, wenn die Basissequenz X noch nicht von einem Pfad benutzt wird. In diesem Fall wird X als von C benutzt markiert. In Doppelsträngen wird durch die Festlegung der Sequenz eines Critons auch die Sequenz seines Komplements auf dem gegenüberliegenden Einzelstrang definiert. Dieses Criton $\overline{C}$ gehört zu einem zweiten Strang S_2 und erhält die komplementäre Basissequenz $\overline{X}$. Diese muss ebenfalls noch unbenutzt sein, damit sie dem Pfad p_{S_2} zugeordnet werden kann. Anderenfalls müssen beide Critons C und $\overline{C}$ eine andere Sequenz erhalten.

In Critonregel 2 wird verlangt, dass die komplementären Sequenzen von Critons, die nicht vollständig und ununterbrochen in Doppelsträngen liegen, nicht vorkommen. Das betrifft Critons in Einzelstrangabschnitten, an Verzweigungspunkten und gegenüber von Lücken im Strangrückgrat. Sie zeichnen sich dadurch aus, dass sie kein komplementäres Criton auf einem gegenüberliegenden Strang besitzen. Um die Regel 2 zu erfüllen, müssen aber auch in diesem Fall die komplementären Basissequenzen geprüft und markiert werden, um sie für eine Benutzung durch andere Critons/Pfade zu sperren.

Relativ einfach lässt sich Critonregel 3 erfüllen, welche selbstkomplementäre Sequenzen verbietet. Für den Fall, dass die Criton-/ Basissequenzlänge eine gerade Zahl ist, werden einfach alle selbstkomplementären Basissequenzen (z. B. GCGC) aus dem Sequenzgraph entfernt. Sie können dann von keinem Pfad mehr benutzt werden. Im ungeraden Fall werden keine Sequenzen entfernt, sondern Nachbarschaftsbeziehungen, die zu selbstkomplementären Sequenzen führen (z. B. GCG $\rightarrow$ CGC) unterdrückt. Bei der Zuordnung der Basissequenzen zu einem Criton/Pfad, muss dann zusätzlich noch geprüft werden, ob Verbindungen zu den Vorgänger- und Nachfolgersequenzen bestehen.

Die letzte Critonregel, die für stabile Verzweigungspunkte sorgen soll, kann mit dem Basissequenz-Konzept nicht oder nur sehr umständlich geprüft werden und bedarf deshalb einer gesonderten Behandlung.

Die Sequenzgenerierung für einen Einzelstrang läuft dann wie folgt ab: Der Einzelstrang wird in Critons zerlegt. Dem ersten Criton im Strang C_0 wird zufällig eine noch unbenutzte Basissequenz X zugewiesen. Diese Sequenz und ihr Komplement $\overline{X}$ werden als benutzt markiert. Für das zweite Criton C_1 wird nun eine Basissequenz Y gesucht, welche Nachfolger von X und unbenutzt ist. Gibt es eine solche, werden Y und $\overline{Y}$ wiederum als benutzt markiert. Für das dritte Criton C_2 bedarf es einer Basissequenz Z die unbenutzt und ein Nachfolger von Y ist. Kann einmal für ein Criton C_i keine Basissequenz gefunden werden, dann muss die Zuordnung des vorangegangenen Critons C_{i-1} rückgängig gemacht und für diesen eine neue Sequenz gefunden werden. Erst danach kann C_i erneut angegangen werden.

Dieser Generierungsalgorithmus arbeitet sich an dem Einzelstrang entlang. Kann er einem Criton eine Basissequenz zuordnen, geht er einen Schritt voran zum nächsten. Bei einem Misserfolg geht er ein Criton zurück. Gelingt es, dem letzten Criton auf dem Strang eine Basissequenz zuzuweisen, ist die Sequenz des Einzelstranges erfolgreich generiert. Ist es nicht mehr möglich, dem zweiten Criton C_1 eine Sequenz zuzuordnen, muss für das erste Criton C_0 eine andere bisher ungetestete Basissequenz gesucht werden. Wurden alle unbenutzten Sequenzen ohne Erfolg getestet, ist die komplette Generierung fehlgeschlagen. Auf diese Weise können nacheinander die Sequenzen für alle Einzelstränge einer DNA-Struktur erzeugt werden. Der in Kapitel 4 beschriebene Algorithmus wird die Sequenzen der Einzelstränge nicht in einem Stück sondern abschnittsweise generieren.

Die Abbildungen 3.3, 3.4, und 3.5 zeigen ein Beispiel für die Sequenzgenerierung unter Verwendung des Basissequenzgraphen. Als Zielstruktur dient eine sehr kleine dreiarmige Verzweigung, deren Arme aus jeweils zwei Basenpaaren besteht. Für diese Struktur reicht gerade noch die Criton- bzw. Basissequenzlänge 2 aus. Der Sequenzgraph und die Sequenzgenerierung sind deshalb noch recht übersichtlich. In den Abbildungen ist jeweils links der Sequenzgraph und rechts die Zielstruktur mit den aktuellen Basenwerten dargestellt. Nach und nach werden im Graph Pfade generiert. Die Farben dieser Pfade korrespondieren mit den Farben der Einzelstränge in der Zielstruktur (rot, blau und grün). Gestrichelt um-

rahmte Basissequenzen werden zwar benutzt, kommen aber in keinem Einzelstrang vor. Es sind die komplementären Sequenzen der Critons, die über dem Verzweigungspunkt liegen.

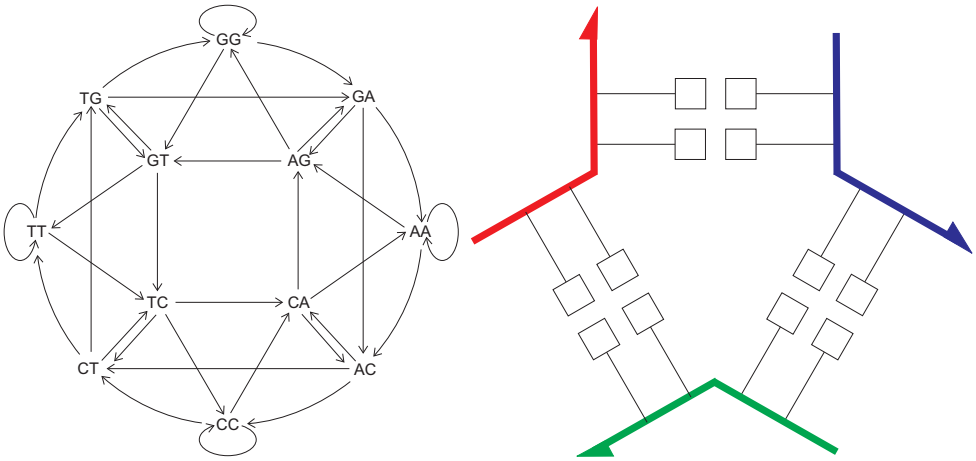
Abbildung 3.3(a) zeigt den völlig unbenutzten Sequenzgraphen und die Zielstruktur mit undefinierten Basenwerten (N). Im Sequenzgraph werden alle selbstkomplementären Basissequenzen (GC, CG, AT und TA) weggelassen, um selbstkomplementäre Subsequenzen prinzipiell auszuschließen. In den Abbildungen 3.3(b), 3.3(c) und 3.4(a) wird die Basensequenz des roten Stranges generiert. Dabei werden nacheinander die Basissequenzen GA, AA und AC als von Rot benutzt markiert. Gleichzeitig werden die komplementären Sequenzen TC dem grünen, TT ebenfalls dem roten und GT dem blauen Strang zugewiesen. TT ist dabei die komplementäre Basissequenz zu AA, die über dem Verzweigungspunkt liegt. Sie befindet sich auf keinem Einzelstrang und ist deshalb gestrichelt dargestellt. Nach diesen drei Schritten besitzt der rote Strang die Basensequenz GAAC. Die Basensequenzen der anderen beiden Stränge sind schon teilweise definiert. Der blaue Strang ist dann GTNN, der grüne NNTC.

Die Abbildungen 3.4(b), 3.4(c) und 3.5(a) zeigen die Sequenzgenerierung für den blauen Einzelstrang. Da nur noch zwei Basen undefiniert sind, bräuchte es eigentlich nur noch zwei Schritte. In Abbildung 3.4(b) kommt es jedoch zu einem Konflikt. Der dritten Base wird der Wert T zugewiesen. Der blaue Strang würde damit die beiden Basissequenzen TT und AA benötigen, um seinen Pfad im Sequenzgraphen fortzusetzen. Diese sind jedoch schon im roten Pfad integriert. Es muss also ein anderer Basenwert gefunden werden, was in Abbildung 3.4(c) geschieht. Die Base erhält den Wert G. Die nun benötigten Basissequenzen TG und CA sind noch frei und können deshalb von Blau benutzt werden. Der letzten Base im vierten Strang wird in Abbildung 3.5(a) ebenfalls der Wert G zugewiesen. Der blaue Pfad wird dadurch um GG erweitert. Gleichzeitig wird dem grünen Pfad die Basissequenz CC hinzugefügt. An dieser Stelle sind bereits alle Basenwerte definiert. Der rote Strang besitzt unverändert die Sequenz GAAC, der blaue die Sequenz GTGG und der grüne die Sequenz CCTC.

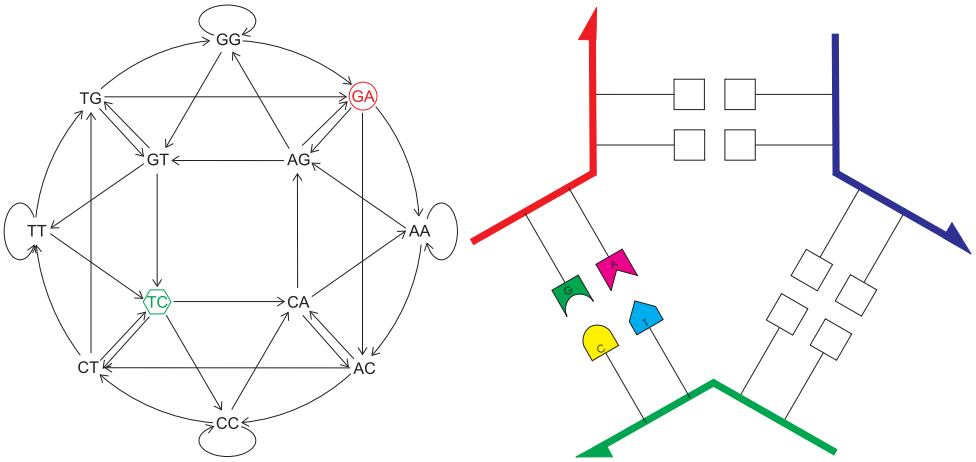
Der grüne Strang ist aber noch nicht vollständig getestet und der grüne Pfad im Graphen noch nicht komplett. Es muss noch überprüft werden, ob die Sequenz CT, die auf dem grünen Strang über dem Verzweigungspunkt liegt, noch frei ist. Glücklicherweise ist dies, ebenso wie bei dem Komplement AG, der Fall und beide können dem grünen Pfad zugeordnet werden, zu sehen in Abbildung 3.5(b). Würde an dieser Stelle ein Konflikt auftreten, müsste man bis zum Zustan aus Abbildung 3.4(c) zurückgehen und dort für die dritte Base des blauen Stranges einen anderen Basenwert finden. Hier erhält man einen Eindruck von den Schwierigkeiten der Sequenzgenerierung in der Nähe von Verzweigungspunkten. Bei dem eben beschriebenen Vorgehen werden während der Sequenzgenerierung in den Strängen ungetestet die Sequenzen anderer Stränge festgelegt. Eventuell auftretende Fehler werden dann zwar später entdeckt, um sie zu beheben, muss aber unter Umständen wieder sehr weit zurückgegangen werden.

Im vorliegenden Fall war die Sequenzgenerierung jedoch ohne solche Probleme erfolgreich. Die Zielstruktur besitzt danach korrekte Sequenzen im Sinne der Critonregeln, denn jede Subsequenz der Länge 2 kommt nur einmal in der Struktur vor (Regel 1), die komplementären Subsequenzen direkt über dem Verzweigungspunkt werden geblockt (Regel 2) und selbstkomplementäre Subsequenzen sind von vornherein ausgeschlossen (Regel 3). Die letzte Critonregel ist bei dreiarmigen Verzweigungspunkten automatisch erfüllt, wenn

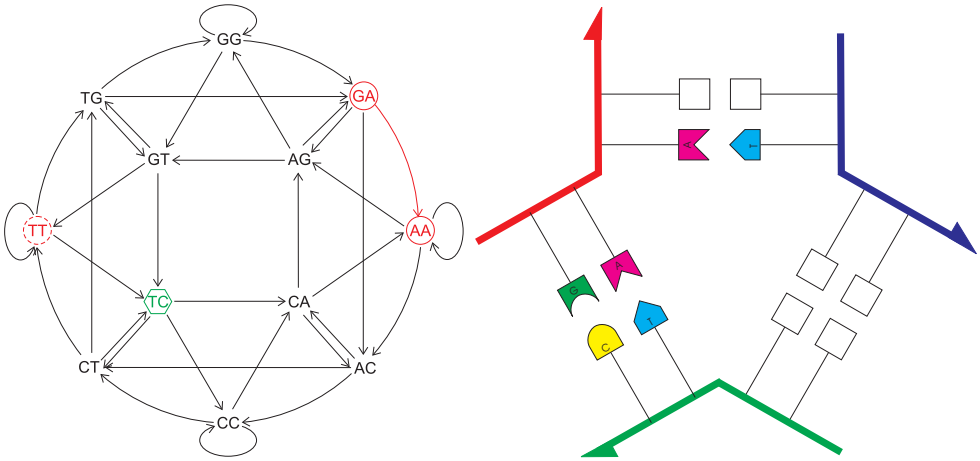
die ersten drei erfüllt sind. Der Verzweigungspunkt der Zielstruktur ist also stabil.



(a)

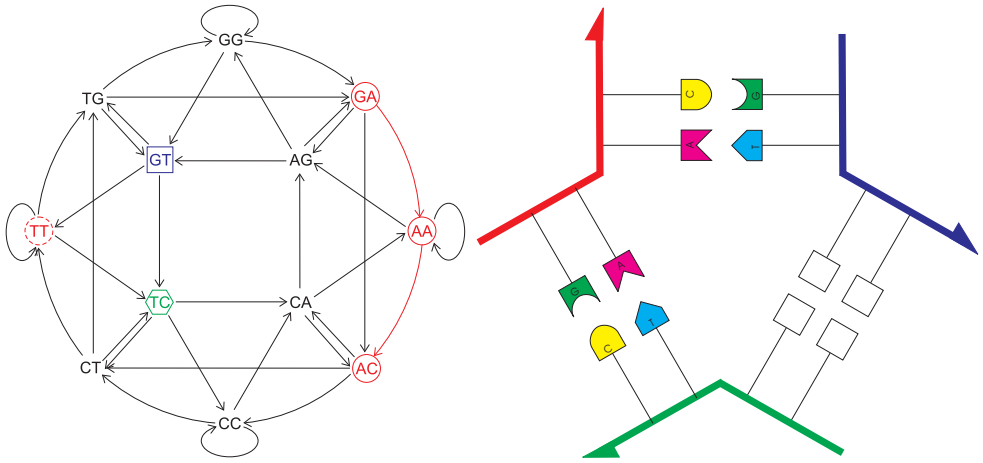


(b)

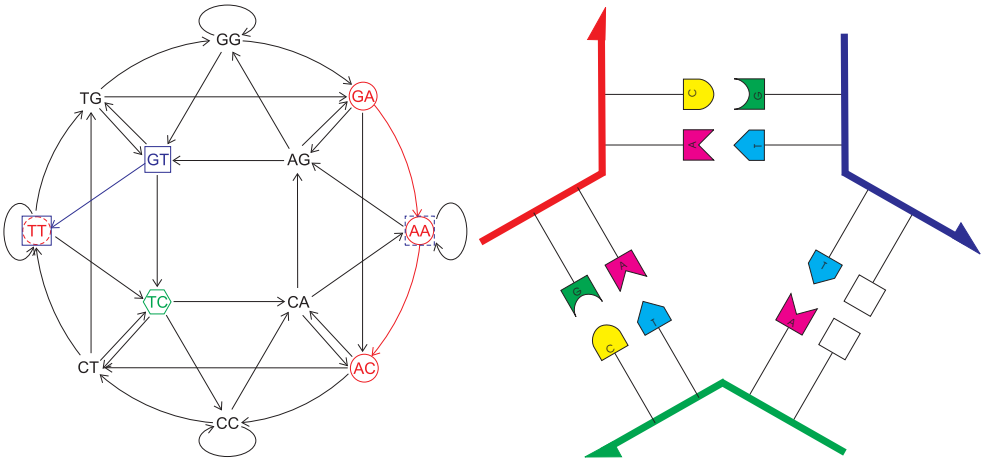


(c)

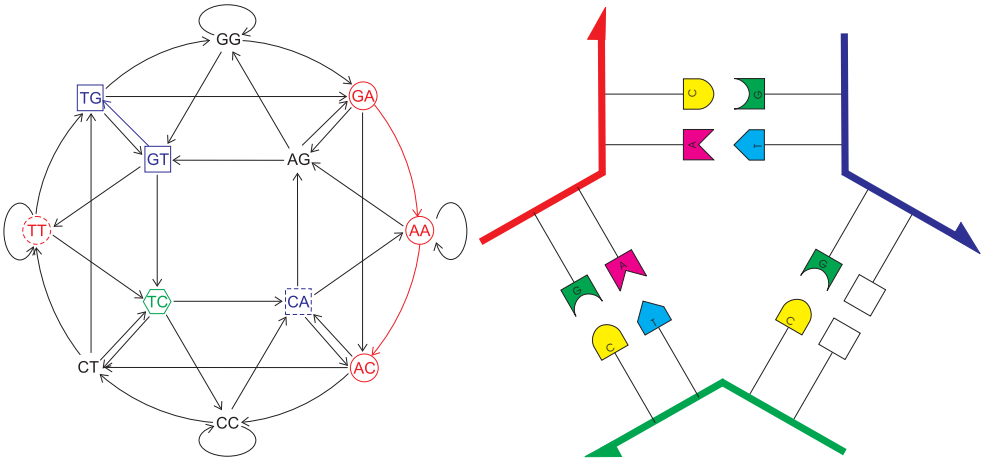
Abbildung 3.3: Sequenzgenerierung mit dem Basissequenz-Konzept



(a)

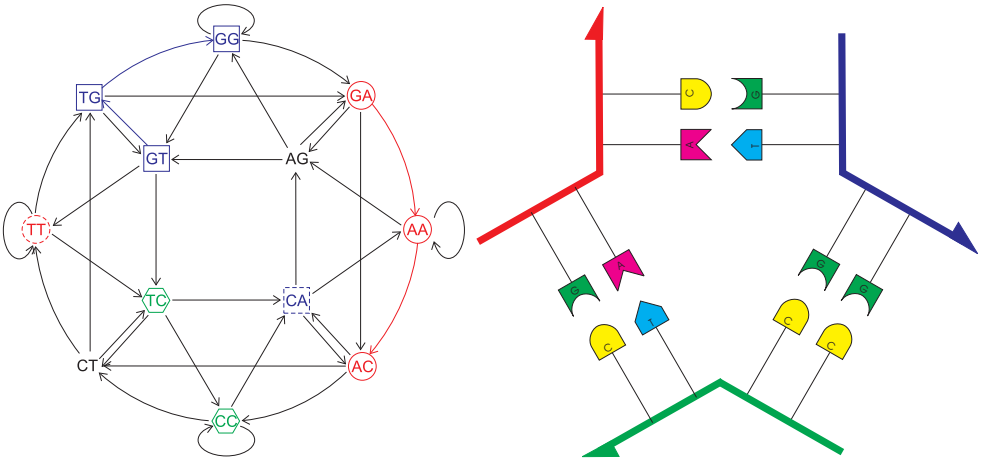


(b)

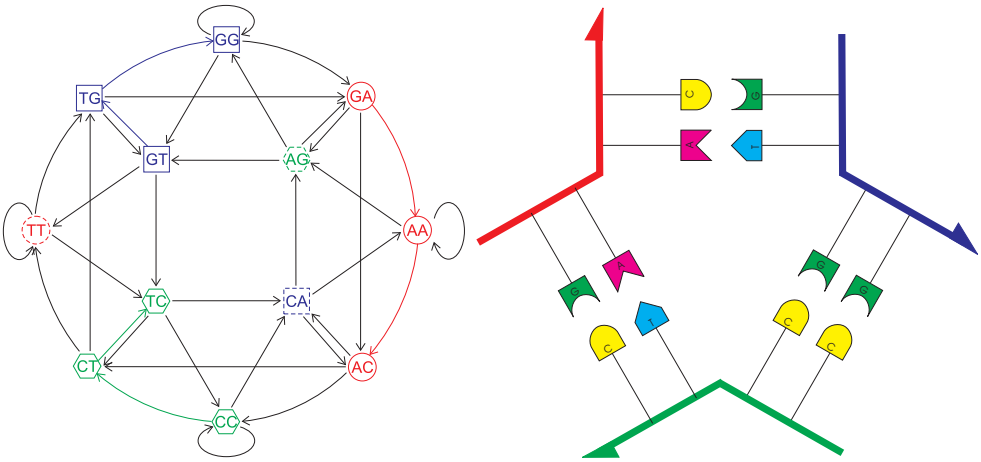


(c)

Abbildung 3.4: Sequenzgenerierung mit dem Basissequenz-Konzept



(a)



(b)

Abbildung 3.5: Sequenzgenerierung mit dem Basissequenz-Konzept

Kapitel 4

Ein vollautomatischer Sequenzdesign-Algorithmus

Der im Folgenden beschriebene Algorithmus hat in seiner Grundversion das Ziel, für jede denkbare DNA-Struktur korrekte Basensequenzen im Sinne der Critonregeln generieren zu können. Zusätzlich sollen beliebige Sequenzen vordefiniert und Sequenzen kleiner oder gleich der Critonlänge verboten werden. In einer Erweiterung kommt dann die Möglichkeit hinzu, den Anteil von G/C-Basenpaaren und dadurch indirekt die Schmelzpunkte von Doppelsträngen zu bestimmen. Ebenfalls wird es möglich sein, beliebige Abschnitte der DNA Struktur zu maskieren, um an diesen Stellen gezielt Verstöße gegen die Critonregeln zu erlauben. Dadurch kann man auch selbstkomplementäre Sequenzen (z. B. die Erkennungssequenzen von Restriktionsenzymen) in die Struktur einfügen oder identische Sequenzbereiche erzeugen.

Der Algorithmus läuft in den folgenden fünf Phasen ab:

1. Einlesen der DNA-Zielstruktur,
2. Normalisierung der DNA-Zielstruktur,
3. Vorbereitung der Sequenzgenerierung,
4. Sequenzgenerierung und
5. Ausgabe der Ergebnisse.

In den folgenden Unterkapiteln wird jede dieser Phasen eingehend beschrieben.

4.1 Einlesen der DNA-Zielstruktur

Zuerst muss die gewünschte DNA-Zielstruktur angegeben werden. Eine DNA-Struktur ist formal betrachtet eine Menge von DNA-Einzelsträngen, welche in Doppelstrangabschnitten durch Basenpaare miteinander verbunden sind. Jeder Einzelstrang hat einen eindeutigen Namen und ist eine Folge von Basen. Die Leserichtung der Basenfolge ist immer die vom 5'- zum 3'-Ende des Stranges (siehe 2.1.2). Eine Base ist genau einem Einzelstrang an einer fest definierten Position zugeordnet und mit ihrer Vorgänger- und

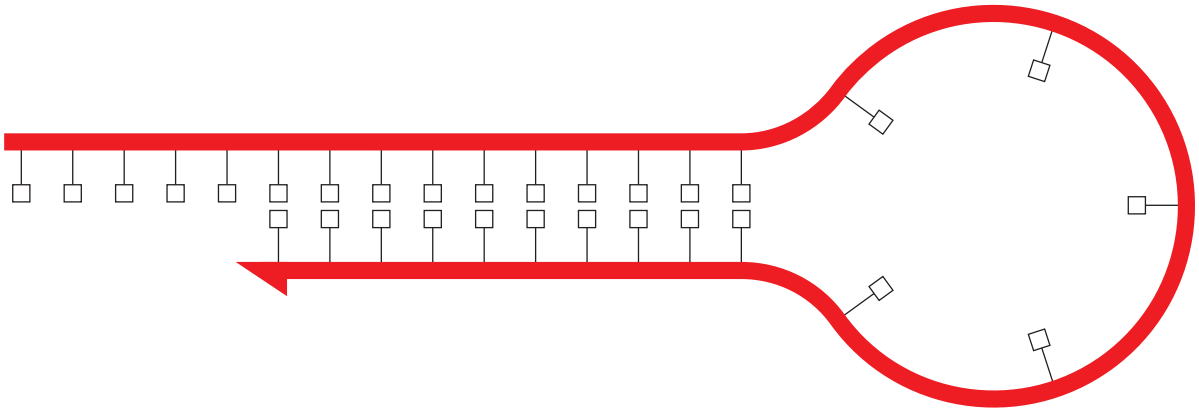


Abbildung 4.1: Darstellung einer DNA-Zielstruktur

Nachfolge-Base im Strang verknüpft. Die Verknüpfungen repräsentieren das Strangrückgrat. Jede Base besitzt einen Basenwert, welcher einer der folgenden fünf sein kann: G für Guanin, A für Adenin, T für Thymin, C für Cytosin oder N für einen noch unbestimmten Wert. Unbestimmte Basenwerte werden während der Sequenzgenerierung durch konkrete Werte ersetzt. Bei der Eingabe kann jeder Base eine Beschränkung der Werte, die sie annehmen kann, auferlegt werden, zum Beispiel nur G oder nur G und C oder auch kein T. Außerdem kann zu jedem Zeitpunkt der aktuelle Basenwert einer Base konstant gesetzt und auch wieder freigegeben werden.

Eine Base kann mit einer beliebigen anderen Base ein Basenpaar bilden. Die Basenwerte der beiden Partner sind in diesem Fall immer komplementär zueinander (G/C, A/T oder N/N). Wird der Wert der einen Base verändert, ändert sich der Wert der anderen Base entsprechend auch. Gleichfalls komplementär sind die Variabilitäten der Basenwerte. Ist zum Beispiel der eine Basenwert nur A oder C, so ist der andere entsprechend nur T oder G. Wird der Wert der einen Base konstant gesetzt, so auch der andere, und umgekehrt.

Ein Doppelstrang ist eine Folge von Basenpaaren, durch welche zwei Einzelstränge oder zwei Abschnitte desselben Stranges ununterbrochen miteinander verbunden sind. Jeder Doppelstrang wird definiert durch die Startbase auf dem ersten Strang, die Startbase auf dem zweiten Strang sowie der Gesamtlänge des Doppelstranges. Dadurch ist die Position des Doppelstranges in der DNA-Struktur eindeutig bestimmt. Eine konkrete Basensequenz wird nicht festgelegt. Durch die Basenpaare ist aber sichergestellt, dass die beiden miteinander verbundenen Strangabschnitte immer komplentär zueinander sind. Doppelstränge überlappen sich nicht.

Die Einzelstränge und die Doppelstrangabschnitte, welche die Einzelstränge verbinden, definieren die Zielstruktur. Abbildung 4.1 zeigt als Beispiel einen einfachen Hairpin-Loop, der aus nur einem Einzelstrang S und einem Doppelstrangabschnitt besteht. Der Strang enthält die 30 Basen $5' - S_0 \dots S_{29} - 3'$. Der Pfeil zeigt das 3'-Ende von S an. Die Basen $S_5 \dots S_{14}$ binden an die Basen $S_{20} \dots S_{29}$ und bilden dadurch einen Doppelstrang. Es entsteht ein dreiarmiger Verzweigungspunkt, dessen einer Arm der Doppelstrang ist und dessen anderen beiden Arme in den Hairpin-Loop hineinreichen.

4.2 Normalisierung der DNA-Zielstruktur

Mit Basen, die Einzelstränge formen, welche wiederum in Doppelsträngen miteinander verbunden sind, kann jede theoretisch denkbare DNA-Struktur formuliert werden. Die möglichen Gestalten, die die Strukturen aufweisen können, sind dementsprechend unerschöpflich vielfältig. Es kann Einzel- und Doppelstrangabschnitte, Verzweigungspunkte und Lücken im Strangrückgrat in beliebiger Zahl und Kombination geben. Um diese Vielfalt auf ein einfacher zu handhabendes Maß zu reduzieren, wird die eingegebene Zielstruktur normalisiert.

Die Normalisierung besteht aus zwei Schritten. Im ersten Schritt werden alle ungebundenen Basen mit virtuellen Basen verknüpft. Diese virtuellen Basen gehören zu keinem Einzelstrang. Sie dienen dazu, die Struktur überall doppelsträngig zu machen. Somit muss nicht mehr zwischen einzelsträngig und doppelsträngig unterschieden werden. Im zweiten Schritt werden alle Lücken in den Strangrückgraten geschlossen. Davon sind auch die zuvor eingeführten virtuellen Basen betroffen. Dadurch ist die Struktur nun nicht nur überall doppelsträngig, die Doppelstrangabschnitte sind nun außer an den Verzweigungspunkten auch nirgendwo mehr unterbrochen.

Die normalisierte DNA-Zielstruktur besitzt dann nur noch zwei Strukturelemente: völlig homogene Doppelstrangabschnitte und Verzweigungspunkte, welche Doppelstrangabschnitte miteinander verbinden. Durch die Normalisierung können die Einzelstränge der Originalstruktur verändert werden. Sie können sich durch Hinzunahme virtueller Basen erweitern und/oder sich mit anderen Strängen verketten. Die Doppelstrangabschnitte verbinden diese neu entstandenen Einzelstränge miteinander.

Ein Verzweigungspunkt ist ein Ort in einer normalisierten Struktur, an welchem mindestens drei Doppelstrangabschnitte miteinander verknüpft sind. Die Basenpaare um den Verzweigungspunkt herum bilden den Basenpaar-Ring dieses Punktes. Die Doppelstrangabschnitte bilden die Arme.

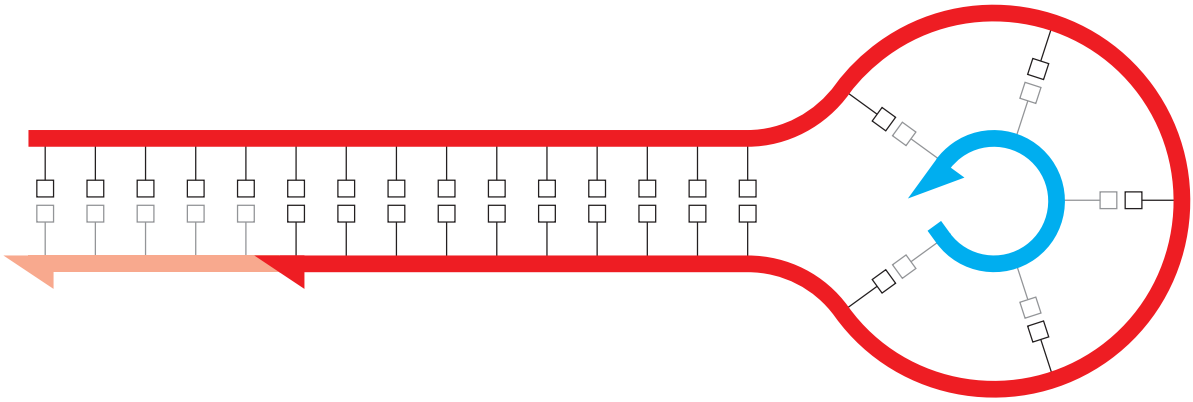
Abbildung 4.2 zeigt die normalisierte Zielstruktur aus Abbildung 4.1. An deren ungepaarte Basen wurden virtuelle Basen gebunden (hellgrau dargestellt). Der Strang S wurde dadurch um 5 virtuelle Basen erweitert und es entstand ein zweiter rein virtueller Einzelstrang im Inneren des Hairpin-Loops. Die normalisierte Zielstruktur besteht aus zwei Doppelstrangabschnitten – dem verlängerten Original-Doppelstrang und einem, der den Loop ausbildet – sowie einem dreiarmigen Verzweigungspunkt.

4.3 Vorbereitung der Sequenzgenerierung

In der Vorbereitungsphase werden die Critonstruktur und der Sequenzgraph erzeugt. Dazu benötigt man die Critonlänge. Ist diese nicht bereits definiert, muss sie jetzt ermittelt werden.

4.3.1 Bestimmung der Critonlänge

Es ist wünschenswert, die Critonlänge L_C so gering wie möglich zu halten, da mit ihr auch die maximale Fehlerlänge festgelegt wird (siehe Kap. 3.2.1). Andererseits bestimmt L_C aber auch die maximale Zahl der Subsequenzen, welche die Critons annehmen können, nämlich 4^{L_C}. Da jede Subsequenz höchstens einmal erscheinen darf (Critonregel 1), können



Hellgrau dargestellte Basen sind virtuell.

Abbildung 4.2: Darstellung der normalisierten Zielstruktur

auch nur höchstens 4^{L_C} korrekte Critons existieren. Die Größe der DNA-Struktur ist also bei gegebener Critonlänge begrenzt. Benötigt man mehr Critons für eine größere DNA-Struktur, muss die Critonlänge und damit auch die maximale Fehlerlänge erhöht werden.

Wenn N_C die Zahl der benötigten Critons einer DNA-Struktur angibt, so lässt sich die optimale Critonlänge L_C unter Berücksichtigung der eben genannten Randbedingungen formulieren als

$$4^{L_C-1} < N_C \leq 4^{L_C}. \tag{4.1}$$

Daraus folgt,

$$L_C - 1 < \log_4 N_C \leq L_C. \tag{4.2}$$

Da die Critonlänge eine natürliche Zahl sein muss, kommt nur die nächste ganze Zahl $\geq \log_4 N_C$ in Frage.

Es ist sehr schwierig, die genaue Zahl der benötigten Critons für eine DNA-Struktur zu ermitteln, weil der Wert von der Critonlänge selbst abhängt und auch gesperrte Subsequenzen nach Critonregel 2 und 3 mit einbezogen werden müssen. Es hat sich aber als hinreichend und praktikabel erwiesen, für jede Base (real und virtuell) in der normalisierten Zielstruktur ein Criton zu veranschlagen. Man vernachlässigt dabei zu sperrende Sequenzen an den Verzweigungspunkten, auf der anderen Seite aber auch, dass ein Strang in weniger Critons zerfällt, als er Basen besitzt. Außerdem ist es sowieso nicht günstig, die benötigten Critons zu knapp zu kalkulieren. Wie im Kapitel 4.5 zu sehen sein wird, kann der Algorithmus nur effizient arbeiten, wenn die Zahl der benötigten Critons kleiner als 85% der zur Verfügung stehenden Subsequenzen beträgt.

Die hier verwendete und normalisierte Zielstruktur enthält 20 Basenpaare. Es werden dafür also näherungsweise 40 Critons benötigt. Aus Gleichung 4.2 ergibt sich somit:

$$\begin{array}{rcccl} L_C - 1 & < & \log_4 40 & \leq & L_C \\ L_C - 1 & < & 2,66 & \leq & L_C \end{array}$$

Die Critonlänge muss also 3 sein.

4.3.2 Aufbau der Critonstruktur

Mit dem Wissen um die Critonlänge kann nun die normalisierte Zielstruktur in die Critons zerlegt werden. Dies geschieht, wie in Kapitel 3.2.1 beschrieben. Jeder Einzelstrang mit Länge l wird in $l - (L_C - 1)$ Critons der Länge L_C unterteilt. Wie schon weiter oben erwähnt, ist hierbei zu beachten, dass die Einzelstränge in der normalisierten Zielstruktur nicht notwendigerweise denen in der Originalstruktur entsprechen. Durch den Einsatz der virtuellen Basen und dem Schließen der Lücken im Strangrückgrat können sich Einzelstränge verlängern oder zusammenschließen. Benachbarte Critons überlappen sich um $L_C - 1$ Basen. Die Basenwerte aller Basen, die ein Criton umfasst, bilden dessen Sequenz.

Alle Critons werden in einem Graphen - der Critonstruktur - zusammengefasst und miteinander verknüpft. Jedes Criton wird mit seinem Vorgänger und Nachfolger auf dem Strang sowie mit seinem Komplement auf dem gegenüberliegenden Strang verbunden. Durch die Verknüpfungen kann später ein Criton in der Nähe eines anderen Critons leicht und schnell gefunden werden.

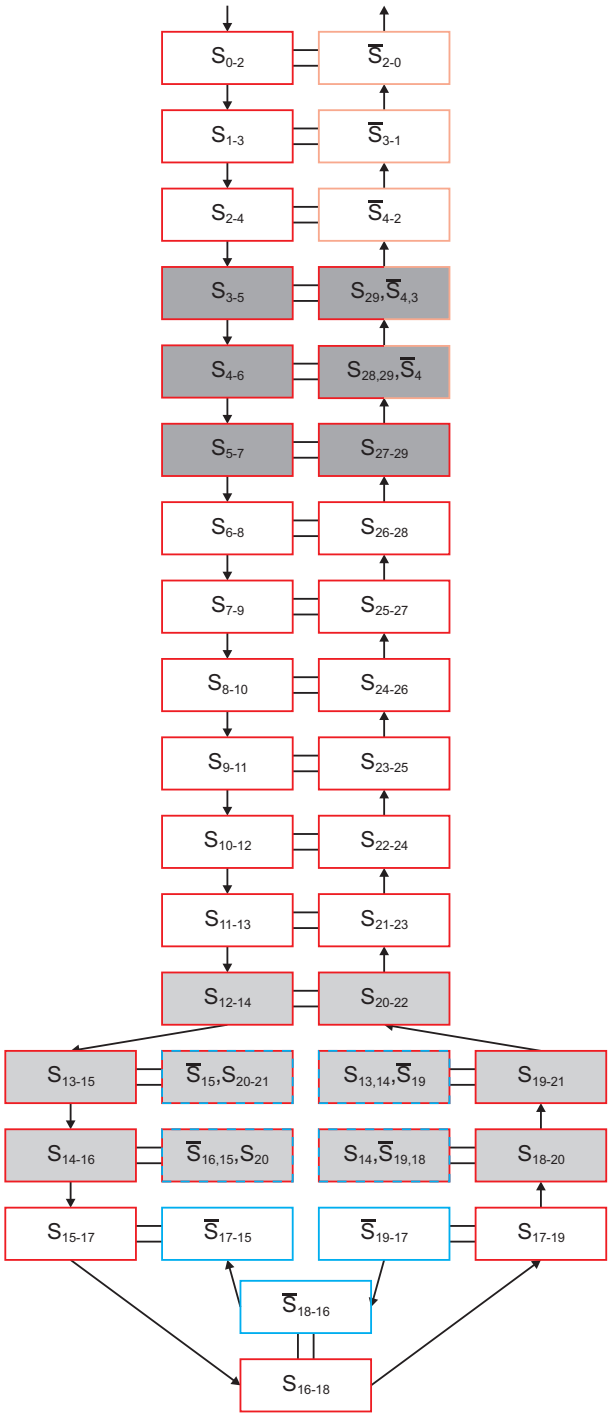
Critons, die über einem Verzweigungspunkt liegen und somit kein komplementäres Criton besitzen, erhalten als Komplement ein virtuelles Criton. In diesem virtuellen Criton werden die Komplemente aller Basen des realen Critons zusammengefasst. Es liegt daher auf mehreren Einzelsträngen. Während der Sequenzgenerierung dienen die virtuellen Critons dazu, bestimmte Sequenzen nach Critonregel 2 zu sperren.

Alle Critons werden in der Critonstruktur zusammengefasst. Abbildung 4.3 zeigt die Critonstruktur der normalisierten Zielstruktur aus Abbildung 4.2. Die Critonlänge beträgt 3. Ein Criton ist dargestellt als ein Kasten, welcher die Auflistung der Basen des Critons in 5'-3'-Richtung enthält. S_{0-2} entspricht den Basen S_0, S_1 und S_2. Die mit einem Querstrich gekennzeichneten Basen sind virtuell. Um den Verzw eigungspunkt herum existieren virtuelle Critons, die auf unterschiedlichen Strangabschnitten liegen. Diese Critons sind zweifarbig und gestrichelt umrandet. Bei Critons, die zweifarbig, aber nicht gestrichelt umrandet sind, wird dagegen nur angezeigt, dass sie teilweise reale und teilweise virtuelle Basen enthalten. Sie sind trotzdem reale Critons.

Die Critons werden zusätzlich den Assoziationsgruppen zugeordnet. Für jedes Basenpaar in der normalisierten Zielstruktur existiert eine Assoziationsgruppe. Die Assoziationsgruppe eines Basenpaares enthält alle Critons, die mindestens eine der beiden Basen des Basenpaares umfassen. Es ist auch denkbar, dass ein Criton beide Basen umfasst, zum Beispiel an einem Hairpin-Loop. Ein Criton ist deshalb Mitglied in maximal L_C Assoziationsgruppen.

Über eine Assoziationsgruppe kann man schnell auf jene Critons zugreifen, deren Sequenzen vom Basenwert des assoziierten Basenpaares abhängig sind. Dieser Zugriff wird von der Sequenzgenerierungsprozedur benötigt. Diese verändert einzelne Basenwerte und muss dann alle betroffenen Critons auf eine korrekte Sequenz hin überprüfen.

In Abbildung 4.3 sind die Assoziationsgruppen zweier Basenpaare markiert. Die mit Dunkelgrau hinterlegten Critons gehören zur Assoziationsgruppe des Basenpaares S_5/S_{29}, die mit Hellgrau hinterlegten Critons zur Assoziationsgruppe von S_{14}/S_{20}. Die hellgraue Assoziationsgruppe ist größer als die dunkelgraue, weil sich dessen Basenpaar neben einem Verzweigungspunkt befindet.



Virtuelle Critons sind gestrichelt umrahmt. Mit Dunkelgrau hinterlegte Critons gehören zur Assoziationsgruppe des Basenpaares S_5/S_{29}, die mit Hellgrau hinterlegten Critons zur Assoziationsgruppe von S_{14}/S_{20}.

Abbildung 4.3: Critonstruktur der normalisierten DNA-Zielstruktur aus Abb. 4.2

4.3.3 Aufbau des Sequenzgraphen

Der Sequenzgraph wird, wie in Kapitel 3.2.4 beschrieben, aufgebaut. Die Basissequenzlänge ist identisch mit der Critonlänge L_C. Selbstkomplementäre Basissequenzen (z. B. GCGC) oder Verbindungen, die zu solchen führen (z. B. GCG → CGC), werden nicht einbezogen. Dieser Schritt ist eigentlich unnötig. Er soll dazu dienen, Critonregel 3 zu erfüllen. Diese wird jedoch immer erfüllt sein, solange auch Critonregel 1, welche die Einzigartigkeit jeder Subsequenz fordert, erfüllt ist. Das liegt daran, dass die normalisierte Zielstruktur überall doppelsträngig ist und sogar die Critons über den Verzweigungspunkten ein Komplement besitzen. Nimmt ein Criton eine selbstkomplementäre Sequenz an, so hätte auch dessen Komplement die gleiche Sequenz. Die Einzigartigkeit ist damit verletzt. Die Sequenzkonfiguration ist also nicht zulässig. Trotzdem werden die selbstkomplementären Basissequenzen aus dem Sequenzgraph entfernt, denn sie dürfen ja sowieso nicht benutzt werden.

Beliebige andere Basissequenzen können zusätzlich aus dem Graph entfernt werden. Gibt es zum Beispiel keine Basissequenzen, welche drei Guanin-Basen hintereinander aufweisen, kann auch keine solche Konstellation in der DNA-Struktur auftauchen.

Jede Basissequenz kann als unbenutzt oder als von einem bestimmten Criton benutzt markiert sein. Zu Beginn sind alle Basissequenzen im Graph unbenutzt. Im Laufe der Sequenzgenerierung wird versucht, jedem Criton in der Critonstruktur eine Basissequenz zuzuweisen. Dadurch entstehen Pfade aus benutzten Basissequenzen im Sequenzgraph, welche den Sequenzen der DNA-Einzelstränge entsprechen. Man kann auch sagen: die Critonstruktur, welche selbst eine Abbildung der normalisierten DNA-Zielstruktur darstellt, wird in den Sequenzgraphen abgebildet.

4.4 Sequenzgenerierung

Die Sequenzgenerierung findet in den Doppelstrangabschnitten der normalisierten DNA-Zielstruktur statt und läuft in zwei Phasen ab:

- Setzen konstanter Sequenzbereiche und
- Generierung aller undefinierten Sequenzen.

Konstante Sequenzbereiche, also alle Basen mit einem vordefinierten und unveränderlichen Basenwert, besitzen keinerlei Variabilität. Die Critons müssen die entsprechenden Basissequenzen unbenutzt im Sequenzgraph vorfinden. Dies ist zu Beginn sehr wahrscheinlich, da dann sehr wenige Basissequenzen benutzt werden. Kommt es bereits in dieser Phase zu Konflikten, kann mit den vordefinierten Sequenzen keine korrekte Gesamtsequenz gefunden werden.

Nach erfolgreichem Setzen der konstanten Bereiche werden alle übrigen Sequenzen in den Doppelstrangabschnitten generiert und zwar jeweils ein Doppelstrangabschnitt nach dem anderen. Die Tabelle 4.1 zeigt den Programmablauf für einen Doppelstrangabschnitt. Die Prozedur erwartet als Parameter die Folge der Basenpaare des Abschnittes, die Folge der dazugehörigen Assoziationsgruppen, welche von der Critonstruktur bereitgestellt werden, sowie den Sequenzgraph. Die Critonlänge und die Basissequenzlänge müssen identisch sein.

```
generateSequence (BP,AG,SG): {
    Erzeuge für jedes Basenpaar BP[i] die Menge
            V[i] := {G,A,T,C} der ungetesteten Basenwerte
            (0 <= i < BP.length)
    Setze i := 0.
    Solange 0 <= i und i < BP.length: {
        Wenn in V[i] kein ungetesteter Basenwert für BP[i]
                mehr vorhanden ist (V[i] = {}): {
            Setze V[i] := {G,A,T,C}.
            Setze i := i-1.
        }
        Sonst: {
            Gib BP[i] zufällig einen ungetesteten Basenwert
                    v aus V[i].
            Entferne v aus V[i].
            Lösche alle Zuordnungen zwischen den Critons aus
                    AG[i] und Basissequenzen aus SG.
            Wenn BP[i] den Basenwert v annehmen darf
                    und jedem Criton aus AG[i] mit kompletter
                    Sequenz eine Basissequenz aus SG
                    zugeordnet werden kann: {
                Setze i := i+1.
            }
        }
    }
    Wenn i = BP.length:
        Erfolg.
    Sonst:
        Misserfolg.
}
```

Die Prozedur erwartet als Parameter die Folge von Basenpaaren `BP` und die Folge von Assoziationsgruppen `AG`. Die Länge beider Folgen ist gleich. `AG[i]` ist die Assoziationsgruppe des Basenpaares `BP[i]` ($0 \leq$ `i` $<$ `BP.length`). Zusätzlich wird der Sequenzgraph `SG` benötigt. Die Basissequenzlänge von `SG` ist identisch mit der Länge der Critons in den Assoziationsgruppen. Die Prozedur meldet Erfolg, wenn die komplette Sequenz generiert wurde. Anderenfalls wird ein Misserfolg zurückgegeben.

Tabelle 4.1: Programmablauf der Sequenzgenerierung für einen einzelnen Doppelstrangabschnitt

Zu Beginn wird für jedes Basenpaar aus der Folge `BP` eine Menge von ungetesteten Basenwerten erzeugt. Jede dieser Mengen wird mit allen vier möglichen Werten (G, A, T und C) initialisiert.

Danach betritt die Prozedur eine Schleife, in deren Verlauf die Variable `i` verändert wird. `i` wird mit 0 initialisiert und zeigt im weiteren Verlauf immer auf das jeweils aktuelle Basenpaar. Jeder Schleifendurchlauf bringt eines der folgenden drei Ergebnisse: `i` wird um 1 vermindert, `i` wird um 1 erhöht oder `i` bleibt gleich. Die Schleife wird erst wieder verlassen, wenn `i` entweder unter 0 fällt oder die Länge der Basenpaar-Folge erreicht.

Die Verminderung von `i` tritt dann ein, wenn für das aktuelle Basenpaar `BP[i]` kein ungetesteter Basenwert mehr zur Verfügung steht, wenn also `V[i]` leer ist. Das passiert dann, wenn alle möglichen Basenwerte schon zuvor erfolglos getestet wurden. In diesem Fall ist der Versuch gescheitert, `BP[i]` einen korrekten Basenwert zu geben. Der Algorithmus geht zum vorhergehenden Basenpaar zurück. Zuvor wird `v[i]` jedoch wieder mit allen vier Basenwerten aufgefüllt. Gelangt die Sequenzgenerierung später wieder zu diesem Basenpaar, stehen wieder alle Basenwerte ungetestet zur Verfügung.

Gibt es noch ungetestete Basenwerte in `V[i]`, wird zufällig einer davon ausgewählt, aus `V[i]` entfernt und dem aktuellen Basenpaar `BP[i]` zugewiesen. Durch den neuen Basenwert ändern sich die Sequenzen aller Critons in der Assoziationsgruppe `AG[i]`. Alle zuvor gemachten Zuordnungen zwischen diesen Critons und Basissequenzen aus dem Sequenzgraph `SG` müssen deshalb gelöscht und die betroffenen Basissequenzen wieder als unbenutzt markiert werden. Dann wird der neue Basenwert `v` getestet. Der Test ist erfolgreich, wenn:

1. das Basenpaar `BP[i]` den Basenwert `v` annehmen darf und
2. jedem Criton in der Assoziationsgruppe `AG[i]` mit vollständiger Sequenz eine Basissequenz im Sequenzgraph `SG` zugeordnet werden kann.

Das Basenpaar `BP[i]` darf den Basenwert `v` dann annehmen, wenn:

1. der Basenwert nicht durch Randbedingungen verboten ist (z. B. vordefinierte Sequenzen) und
2. alle Basenpaar-Ringe, in denen `BP[i]` Mitglied ist, eine korrekte Konfiguration nach Critonregel 4 (stabile Verzweigungspunkte) besitzen.

Dieser Liste können noch weitere Bedingungen hinzugefügt werden (siehe Kap. 4.6). Nach einem Erfolg dieses Testes wird versucht, jedem Criton in der Assoziationsgruppe von `BP[i]`, der eine vollständige Sequenz hat, eine Basissequenz im Sequenzgraph zuzuordnen. Die Sequenz eines Critons ist unvollständig, wenn einer oder mehrere seiner Basenwerte undefiniert (N) sind. Einem Criton `c` aus der Assoziationsgruppe kann dann eine Basissequenz zugeordnet werden, wenn:

1. die entsprechende Basissequenz im Sequenzgraph `SG` existiert,
2. diese Basissequenz `bs` noch unbenutzt ist,
3. die Basissequenz, die das Vorgänger-Criton von `c` benutzt, ein Vorgänger von `bs` in `SG` ist und

4. die Basissequenz, die das Nachfolger-Criton von `c` benutzt, ein Nachfolger von `bs` in `SG` ist.

Da jede Basissequenz nur von einem Criton benutzt werden kann, dürfen zwei unterschiedliche Critons nicht dieselbe Sequenz annehmen. Critonregel 1 wird dadurch sichergestellt. Die virtuellen Critons an den Verzweigungspunkten, die auch in den Assoziationsgruppen sind, blockieren die von Critonregel 2 verbotenen Sequenzen. Die dritte Regel, die selbstkomplementäre Sequenzen verbietet, war ja schon von vornherein erfüllt.

Wenn das Zuordnen der Critons erfolgreich verläuft, dann ist der Basenwert `v` richtig gewählt. Dem Basenpaar `BP[i]` wurde ein korrekter Wert zugewiesen. Die Variable `i` wird um 1 erhöht und die Prozedur geht zum nächsten Basenpaar.

Kann auch nur einem Criton mit vollständiger Sequenz keine Basissequenz zugeordnet werden oder wenn `BP[i]` den Basenwert `v` nicht annehmen darf, so bleibt `i` unverändert. Im nächsten Schleifendurchlauf wird ein anderer Basenwert aus `V[i]` getestet.

Auf diese Weise arbeitet sich die Prozedur an der Basenpaar-Folge entlang. Bei einem Erfolg geht sie zum nächsten Paar. Bei einem Misserfolg verharrt sie an gleicher Stelle oder geht schließlich ein Basenpaar zurück. Gelingt es, dem letzten Basenpaar einen korrekten Basenwert zuzuweisen, ist die Sequenz des Doppelstrangabschnittes erfolgreich generiert worden. Gelingt es andererseits einmal nicht, für das erste Basenpaar einen Basenwert zu finden, ist die Sequenzgenerierung gescheitert. Die Sequenzen aller Doppelstrangabschnitte werden in unbestimmter Reihenfolge nacheinander generiert. Eine erfolgreich generierte Sequenz eines Abschnittes ändert sich nicht mehr. Wenn alle Abschnitte erfolgreich generiert wurden, ist die Sequenz der gesamten DNA-Struktur korrekt. Scheitert ein Abschnitt, so ist die komplette Sequenzgenerierung gescheitert. Aufgrund der zufälligen Zuweisung der Basenwerte kann ein erneuter Versuch allerdings doch noch einen Erfolg bringen.

Es stellt sich die Frage nach der Behandlung der Verzweigungspunkte. Offensichtlich bedarf es zumindest vorerst keiner Sonderbehandlung. Die Assoziationsgruppen beinhalten jeweils alle Critons, die von der Veränderung eines bestimmten Basenwertes abhängig sind. Ob diese nun in einem unverzweigten Abschnitt oder an einem Verzweigungspunkt liegen, spielt dabei keine Rolle (siehe Abb. 4.3, S. 56). Alle Critons werden getestet und nur korrekte Konstellationen werden akzeptiert. Allerdings hat sich gezeigt, dass der Algorithmus auf diese einfache Weise nicht immer in der Lage ist, passende Sequenzen zu generieren. Durch die Festlegung von Sequenzen auf den Armen eines Verzweigungspunktes werden die Möglichkeiten auf den verbleibenden Armen zu sehr eingeschränkt. Die deshalb unbedingt nötigen Basissequenzen sind oft bereits an anderer Stelle benutzt. Um die Erfolgswahrscheinlichkeit zu erhöhen, behandelt man die Verzweigungspunkte separat, wie im Kapitel 4.6.1 beschrieben wird.

4.5 Komplexitätsbetrachtungen

Bei jedem Algorithmus stellt sich die Frage, welcher Rechenaufwand bei einer bestimmten Problemgröße zu erwarten ist. Als Basisoperation des hier beschriebenen Sequenzdesign-Algorithmus bietet sich der Basenwert-Test inklusive Zuordnung der Critons zu den Basissequenzen an (vgl. Tab. 4.1, S. 58). Man vernachlässigt dabei zwar den Umstand, dass nicht immer die gleiche Anzahl Critons zugeordnet werden muss, für eine Näherung ist das aber ausreichend. Als Problemgröße wird im Folgenden die Anzahl der Basenpaare in

der normalisierten DNA-Zielstruktur betrachtet. Anzahl und Art der Verzweigungspunkte werden nicht berücksichtigt.

Wie viele Basistests $O(bp)$ benötigt der Algorithmus bei einer Problemgröße von bp Basenpaaren? Der minimale Aufwand ist leicht zu ermitteln. Er beträgt

$$O_{min}(bp) = bp. \tag{4.3}$$

Dies entspricht dem Fall, dass für jedes Basenpaar nur ein Basenwert getestet werden muss und alle Tests erfolgreich sind. Ähnlich einfach ist der maximale Rechenaufwand zu bestimmen. Dieser ist

$$O_{max}(bp) = \sum_{i=1}^{bp} 4^i \tag{4.4}$$

und ist dann gegeben, wenn alle möglichen Sequenzkonstellationen getestet werden müssen und erst die allerletzte die korrekte ist, oder aber keine korrekte Konstellation existiert. Der Algorithmus hat also sowohl das Potenzial zu einem linear, als auch zu einem exponentiell mit der Problemgröße wachsenden Rechenaufwand. Zwischen den beiden Extremen bewegt sich der durchschnittliche Rechenaufwand $O_{avg}(bp)$.

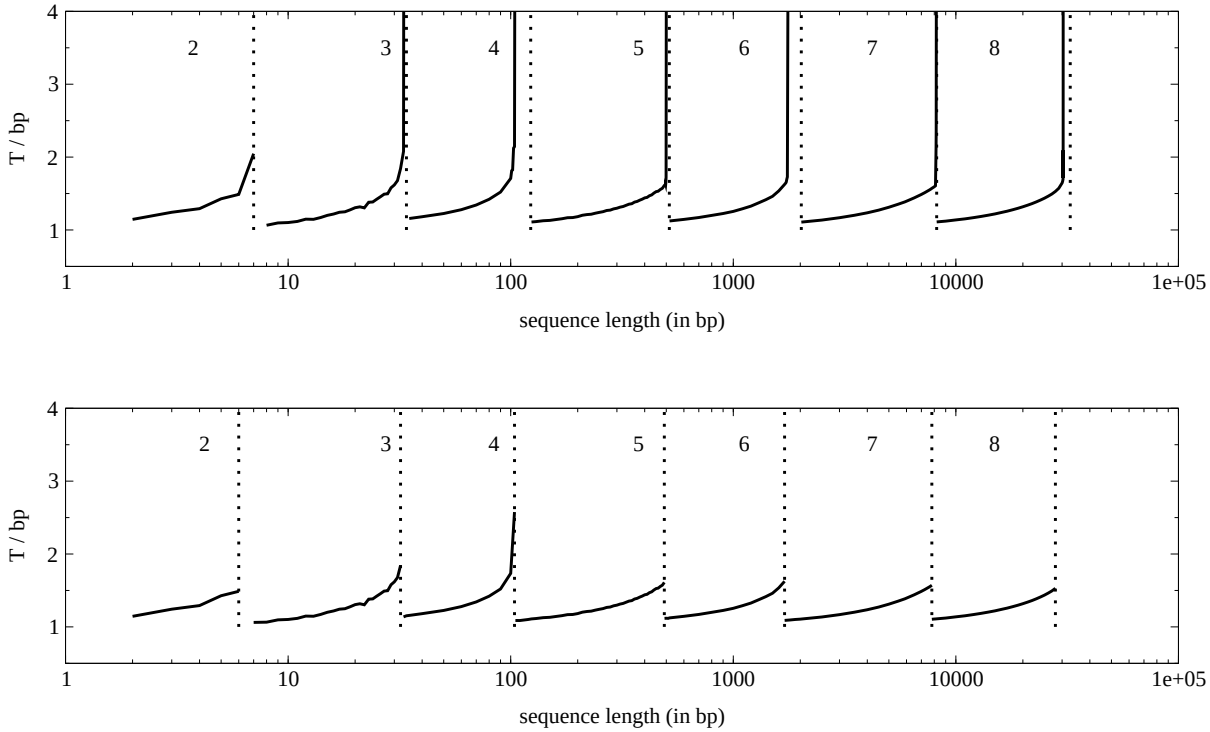
Um $O_{avg}(bp)$ zu bestimmen, wurden mit dem Programm Seed (siehe Kap. 4.7) Messungen mit Strukturgrößen von 2 bis 30000 Basenpaaren durchgeführt. Da Verzweigungspunkte vernachlässigt werden, bestanden die getesteten Strukturen jeweils nur aus einem Doppelstrang mit entsprechender Länge. Die Ergebnisse der Messungen sind in Abbildung 4.4(a) dargestellt. Das Diagramm zeigt die durchschnittlich gemessene Anzahl an Basistests pro Basenpaar in Abhängigkeit von der Größe der DNA-Struktur in Basenpaaren. Für jede beteiligte Strukturgröße wurden 100 Einzelmessungen vorgenommen. Eine Messung ermittelte, wie oft während einer erfolgreichen Sequenzgenerierung dem aktuellen Basenpaar (`BP[i]`) ein neuer Basenwert zugewiesen und getestet wurde (siehe dazu Tab. 4.1). Ein Messpunkt im Diagramm ist das arithmetische Mittel der Einzelmessungen geteilt durch die Anzahl der Basenpaare in der Struktur.

Für eine bessere Übersichtlichkeit ist die x-Achse des Diagrammes logarithmisch skaliert. Senkrechte gestrichelte Linien markieren die maximalen Strukturgrößen für eine bestimmte Critonlänge. Sie liegen bei 7, 34, 123, 516, 2021, 8198 und 32647 Basenpaaren. Das entspricht der halbierten Zahl der Basissequenzen im Sequenzgraph (4^{L_C} bzw. $4^{L_C} - 4^{\frac{L_C}{2}}$ wenn L_C gerade ist) plus $L_C - 1$ überzählige Basenpaare, da ein Strang der Länge l aus nur $l - (L_C - 1)$ Critons besteht. L_C ist dabei die im jeweiligen Größenbereich passende Critonlänge. Im Diagramm wird sie durch die Zahlen 2 bis 8 zwischen den senkrechten Trennlinien angezeigt.

Die Messkurve zeigt in der gewählten halblogarithmischen Darstellung ein quasi periodisches Verhalten in Abhängkeit von der verwendeten Critonlänge. In jedem Critonlängen-Bereich ist ein ähnlicher Kurvenverlauf zu beobachten: Vom Beginn des Bereiches bis zu ungefähr 85% der Maximalgröße, die durch die nächste senkrechte Trennlinie angezeigt wird, steigt die Messkurve nur sehr langsam und bewegt sich zwischen einem und zwei Basistests pro Basenpaar. Es gilt

$$bp \leq O_{avg1}(bp) \leq 2bp. \tag{4.5}$$

Der durchschnittliche Rechenaufwand kann in diesem Abschnitt also näherungsweise



(b) Anzahl der Basistests bei reduzierter Auslastung der Critonlängen-Bereiche

Abbildung 4.4: Gemessene Anzahl Basistests pro Basenpaar in Abhängkeit von der Gesamtzahl der Basenpaare

als linear ansteigend betrachtet werden. Entsprechend schnell und effizient arbeitet der Algorithmus.

Im letzten Fünftel eines Critonlängen-Bereiches gibt es eine Stelle, an der die Messkurve plötzlich extrem ansteigt. Ab da bedeutet jedes weitere Basenpaar eine Vervielfachung der benötigten Tests. Schnell werden sehr hohe Werte erreicht, die in der Abbildung nicht mehr sinnvoll darzustellen sind. Der Anstieg ist so extrem, dass nur wenige Messpunkte ermittelt werden konnten, denn eine einzige Sequenzgenerierung dauert nun nicht mehr nur wenige Sekunden, sondern mehrere Stunden oder Tage. Der durchschnittliche Rechenaufwand $O_{avg2}(bp)$ in diesem Abschnitt ist darum auch nicht gut zu bestimmen. Er nähert sich aber ganz offensichtlich $O_{max}(bp)$ an.

Nach weiterer Vergrößerung der DNA-Struktur über die Maximalgröße des Critonlängen-Bereiches hinaus fällt die Messkurve bei erhöhter Critonlänge wieder unter zwei Basistests pro Basenpaar.

Es ist gut zu sehen, dass die Stelle, an welcher die extreme Steigung beginnt, in Bereichen gerader Critonlängen deutlich eher auftritt als in Bereichen ungerader Critonlängen. Die Gründe dafür sind noch nicht geklärt.

Wie sind die unterschiedlichen Verhaltensweisen des Algorithmus zu erklären? Beim Basistest ist die Zuordnung der Critons zu den Basissequenzen im Sequenzgraph der wesentliche Faktor. Wie viele Basissequenzen zur Verfügung stehen, bestimmt die Critonlänge L_C. Es gibt 4^{L_C} Basissequenzen abzüglich der selbstkomplementären für den

Fall, dass L_C gerade ist. Die Größe der DNA-Struktur bestimmt die Anzahl der Basissequenzen, die benutzt werden. Innerhalb eines Critonlängen-Bereiches steigt der maximale Anteil benutzter Basissequenzen im Sequenzgraph mit wachsender Strukturgröße von 25% bis 100% kontinuierlich an, denn es werden immer mehr Basissequenzen benötigt, die Critonlänge und damit die Gesamtzahl der Basissequenzen bleibt aber konstant. Je höher der Sequenzgraph mit benutzten Basissequenzen ausgelastet ist, umso schwerer ist es, noch unbenutzte Sequenzen für die verbliebenen Critons zu finden. Offensichtlich gibt es eine bestimmte Auslastung, ab der das Zuordnen der Critons so schwierig wird, dass der Rechenaufwand extrem in die Höhe schnellt. Udo Feldkamp, welcher in seiner Diplomarbeit ebenfalls einen Generierungsalgorithmus auf Grundlage eines Sequenzgraphen benutzte [36], schrieb, dass mit diesem eine Auslastung des Graphen von 80% erreicht werden konnte. Der hier beschriebene Algorithmus schafft eine Auslastung von 85% und bei ungerader Critonlänge sogar bis zu 95%. Woher dieser doch sehr deutliche Unterschied zwischen geraden und ungeraden Critonlängen entsteht, ist, wie schon erwähnt, bisher noch nicht klar. Die erreichten Auslastungen des Sequenzgraphen erscheinen jedoch ausreichend gut, weil nur wenige Strukturgrößen von sehr hohem Rechenaufwand betroffen sind. Zudem lässt sich auch in diesen Fällen leicht Abhilfe schaffen, indem man die Critonlänge früher als theoretisch notwendig erhöht. Zum Beispiel kann eine DNA-Struktur mit 115 Basenpaaren noch mit der Critonlänge 4 behandelt werden. Es stehen dann 240 Basissequenzen zur Verfügung, gebraucht werden 224. Die Auslastung des Sequenzgraphen ist mit 93% aber schon recht hoch. Der Algorithmus braucht äußerst viele Rechenschritte für eine erfolgreiche Sequenzgenerierung. Führt man die Generierung dagegen mit Critonlänge 5 aus, stehen bei gleicher Strukturgröße 1024 Basissequenzen zur Verfügung. Der Sequenzgraph wird nur noch zu 22% ausgelastet und der Algorithmus ist wesentlich schneller.

Abbildung 4.4(b) zeigt Messungen unter den gleichen Bedingungen wie in 4.4(a), bei denen jedoch die Critonlängen in den kritischen Abschnitten erhöht wurden. Alle Critonlängen-Bereiche sind in diesem Diagramm dadurch leicht nach links verschoben. Die senkrechten Trennlinien liegen bei 6, 32, 104, 490, 1700, 7800 und 28000 Basenpaaren, was einer Reduzierung der maximalen Strukturgröße um 15% bei gerader und um 5% bei ungerader Critonlänge gegenüber 4.4(a) entspricht. Die Änderung hat zur Folge, dass die Messkurve flach bleibt und sich über den gesamten Messbereich fast ausschließlich zwischen einem und zwei Basisstests pro Basenpaar bewegt. Es spielt keine Rolle, ob die DNA-Struktur 20 oder 20000 Basenpaare enthält. Der Rechenaufwand lässt sich immer durch O_{avg1} aus Gleichung 4.5 abschätzen. Extreme Rechenzeiten werden vermieden.

Einziger Nachteil dieser Methode ist, dass die Sequenzen für die DNA-Strukturen in den betroffenen Größenbereichen erhöhte Fehlerlängen aufweisen. Da die Fehlerlänge aber jeweils nur um eins steigt und zudem wenige Größenbereiche betroffen sind, erscheint dies akzeptabel.

4.6 Erweiterungen des Algorithmus

Der in den vorangegangenen Unterkapiteln beschriebene Algorithmus kann nun an verschiedenen Stellen erweitert werden. Eine erste Erweiterung ist die Sonderbehandlung der Verzweigungspunkte.

4.6.1 Verbindungen

Wie bereits in Kapitel 4.4 erwähnt, kann der Algorithmus mit Verzweigungspunkten umgehen, erreicht aber in vielen Fällen kein Ergebnis. Der Grund dafür liegt darin, dass die Sequenzen in der Nähe der Verzweigungspunkte besonders sensibel sind. Die Festlegung der Sequenz eines Critons hat Einfluss auf die Sequenzen vieler anderer Critons auf abzweigenden Strängen. Es sind wesentlich mehr als innerhalb der linearen Doppelstrangabschnitte. Dieser Aspekt wird durch die Assoziationsgruppen berücksichtigt (siehe Abb. 4.3), so dass keine falschen Sequenzkonstellationen entstehen können. Da die Verzweigungspunkte aber zu beliebigen Zeitpunkten an der Reihe sind, stehen dann oft die nötigen Basissequenzen nicht mehr unbenutzt zur Verfügung. In der Grundversion des Algorithmus wird Doppelstrangabschnitt für Doppelstrangabschnitt generiert. Für einen Verzweigungspunkt bedeutet das, dass nacheinander in beliebiger Reihenfolge die Sequenzen der Arme generiert werden. Ist ein Arm festgelegt, ändert er sich nicht mehr. Die Critons, die über einem Verzweigungspunkt liegen, liegen auf zwei Armen. Bei der Generierung eines Armes, wird ihre Sequenz schon teilweise festgelegt und dadurch die Sequenzvariabilität der Gesamtsequenz eingeschränkt. Wird die Sequenz des nächsten Armes generiert, muss der Algorithmus mit dieser Einschränkung umgehen. Nun kann es passieren, dass bei der Generierung zweier Arme die Sequenzvariabilität eines Nachbararmes so sehr eingeschränkt wird, dass keine korrekte Konstellation mehr existiert. Eine kleine Änderung auf den anderen Armen würde diese Blockade auflösen, an die entsprechenden Basenpaare kommt der Algorithmus aber nicht mehr heran, da sie in einem anderen Doppelstrangabschnitt liegen. Die Sequenzgenerierung muss abbrechen und ist gescheitert, obwohl durchaus korrekte Konstellationen existieren.

Um dem Abhilfe zu schaffen, werden die Umgebungen der Verzweigungspunkte gesondert erfasst und während der Sequenzgenerierung extra behandelt. Die Datenstruktur dafür heißt Verbindung. Eine Verbindung enthält einen oder mehrere Verzweigungspunkte und alle Basenpaare bis zu einer bestimmten Entfernung. Diese Entfernung heißt Reichweite der Verbindung. Zwei Verzweigungspunkte, die innerhalb der Reichweite voneinander entfernt liegen, gehören zur gleichen Verbindung. Die Basenpaare innerhalb der Reichweite von einem der Verzweigungspunkte bilden die Umgebung der Verbindung. In den meisten Fällen wird eine Verbindung nur einen einzigen Verzweigungspunkt enthalten, es sind aber auch Konstellationen mit mehreren denkbar. Eine solche ist in Abb. 4.5 dargestellt. Diese multiple Verbindung umfasst drei dreiarmige Verzweigungspunkte, die jeweils zwei Basen voneinander entfernt liegen. Die Reichweite der Verbindung beträgt 3, weshalb alle Verzweigungspunkte zur selben Verbindung gehören. Die grau ausgefüllten Basen gehören zur Umgebung der Verbindung. Die hellgrau ausgefüllten Basen sind virtuell.

In der Umgebung der Verbindung sollen alle Critons vollständig enthalten sein, welche einen Verzweigungspunkt überspannen. Die Reichweite muss deshalb mindestens die um eins verminderte Critonlänge betragen.

Zu den bisher zwei Phasen der Sequenzgenerierung kann nun eine dritte hinzukommen. Die drei Phasen lauten dann:

- Setzen konstanter Sequenzbereiche,
- Generierung der Sequenzen in den Umgebungen der Verbindungen und

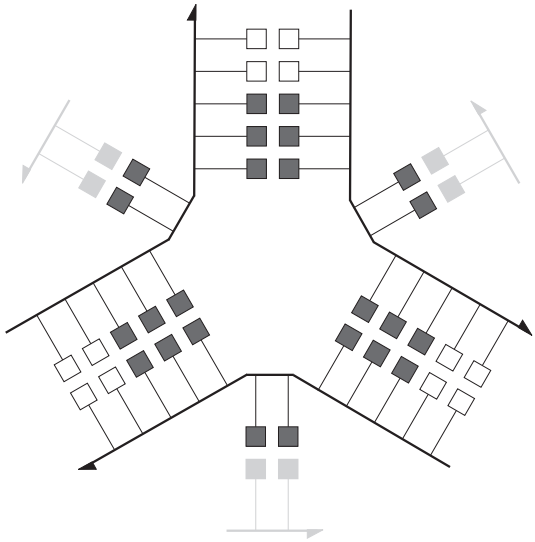


Abbildung 4.5: Verbindung mit Reichweite 3 und drei Verzweigungspunkten

- Generierung aller undefinierten Sequenzen in den Doppelstrangabschnitten.

Die Sequenzen in den Verbindungen werden also direkt nach dem Setzen der konstanten Abschnitte generiert. Das hat den zusätzlichen Vorteil, dass noch sehr viele unbenutzte Basissequenzen für diese sensiblen Bereiche zur Verfügung stehen. Die Verbindungen werden nacheinander in beliebiger Reihenfolge generiert. Nach erfolgreicher Generierung werden alle Basenwerte in den Umgebungen konstant gesetzt und bleiben im weiteren Verlauf deshalb unverändert. Hier liegt auch der Grund dafür, mehrere Verzweigungspunkte in eine Verbindung aufzunehmen. Liegen nämlich zwei Verzweigungspunkte zu nah beieinander, wird durch die Festlegung der Sequenz des einen Punktes auch die des anderen teilweise definiert. Es könnte also wieder zu großen Einschränkungen der Sequenzvariabilität und zu Blockaden kommen.

Wie verläuft nun die Sequenzgenerierung in den Verbindungen? Am günstigsten ist es, an den Verzweigungspunkten zu beginnen und dann auf allen Armen gleichzeitig sternförmig nach außen zu gehen. Die sensibelsten Stellen werden dadurch zuerst bearbeitet. Abbildung 4.6 stellt dieses Vorgehen für eine einfache dreiarmige Verbindung dar.

Solch ein Verhalten des Algorithmus kann auch sehr leicht erreicht werden, indem man alle Basenpaare der Verbindung in der gewünschten Reihenfolge anordnet und diese Folge zusammen mit der entsprechenden Folge der Assoziationsgruppen an die *generateSequence()*-Prozedur aus Tabelle 4.1 übergibt.

Man könnte die Sequenzen in den Umgebungen der Verbindungen auch mit einer verminderten Critonlänge generieren. Das hätte besonders dann Sinn, wenn die DNA-Zielstruktur sehr groß ist und daher mit großen Critonlängen gearbeitet werden muss. Dies zieht eine größere Länge eventueller Fehlstellen nach sich, die an den kritischen Verzweigungspunkten unerwünscht sein kann. Die Verbindungen bilden eine Teilstruktur der Zielstruktur, die weniger Critons benötigt und deren Sequenzen deshalb mit kleinerer Critonlänge generiert werden können. Man nutzt dazu eine zweite Critonstruktur und den

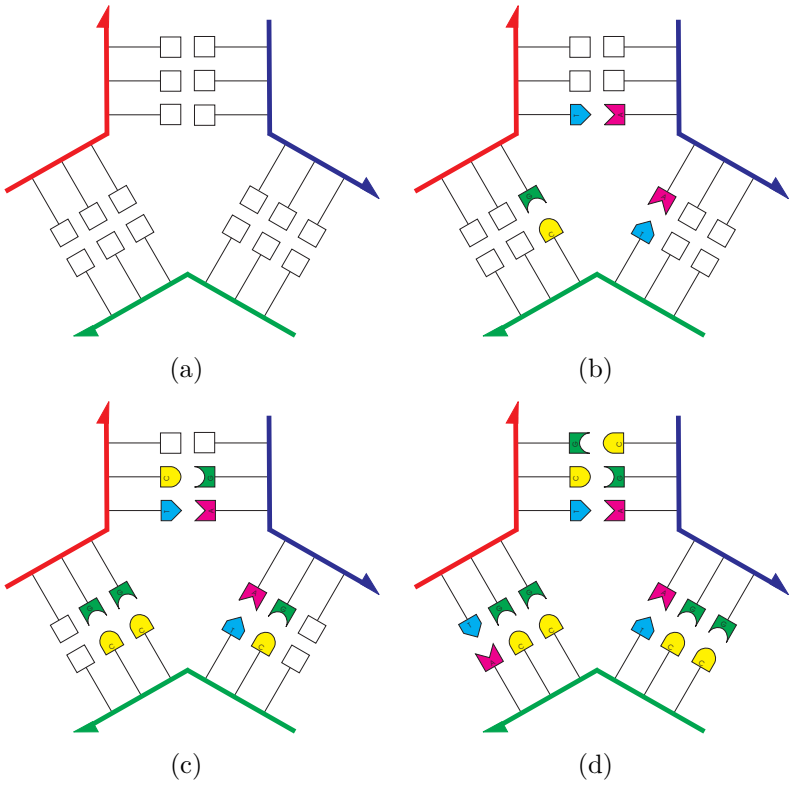


Abbildung 4.6: Sequenzgenerierung in einer Verbindung

dazugehörigen Sequenzgraph. Wichtig ist, dass die Reichweite der Verbindungen in einem solchen Fall an der Critonlänge für die Gesamtstruktur gemessen wird.

4.6.2 Thermodynamische Eigenschaften

In den Basisalgorithmus lassen sich sehr leicht zusätzliche Tests des Basenwertes einbauen. Ein sehr nützlicher ist der Test des G/C-Basenpaar-Anteils des Doppelstranges, zu dem das aktuelle Basenpaar `BP[i]` gehört. Der Basenwert-Test in Kapitel 4.4 auf Seite 59 würde dann um einen Punkt erweitert werden. Das aktuelle Basenpaar `BP[i]` darf den Basenwert `v` dann annehmen, wenn

1. der Basenwert `v` nicht durch Randbedingungen verboten ist (z. B. vordefinierte Sequenzen),
2. alle Basenpaar-Ringe, in denen `BP[i]` Mitglied ist, mit `v` eine korrekte Konfiguration nach Critonregel 4 (stabile Verzweigungspunkte) besitzen und
3. der Doppelstrang, zu dem `BP[i]` gehört, einen vordefinierten G/C-Anteil mit `v` noch erreichen kann.

Soll der G/C-Anteil eines Doppelstranges 50% betragen, bedeutet das, dass bei einer Länge von 10 Basenpaaren fünf davon ein G/C-Paar sein müssen. Im erweiterten Basenwert-Test, überprüft der Algorithmus, ob diese Forderung bei Verwendung des zu

testenden Basenwertes noch erfüllt werden kann. Sind schon alle fünf G/C-Paare definiert, darf kein weiteres auftreten. Sind dagegen erst drei festgelegt, besteht Spielraum für zwei zusätzliche G/C-Paare. Die G/C-Basenpaaranteile können mit Ober- und Untergrenzen für jeden Doppelstrang einzeln und/oder global für alle Doppelstränge vordefiniert werden.

Der Einbau dieses Testes erlaubt es, Einfluss auf die Schmelztemperatur der Doppelstränge zu nehmen, da ein höherer G/C-Anteil eine höhere Schmelztemperatur nach sich zieht. Insbesondere wird es möglich, Gruppen von Doppelsträngen mit ähnlichen Schmelztemperaturen zu generieren. Bei der globalen Definition eines G/C-Anteiles wäre das Ergebnis, dass Doppelstränge mit gleicher Länge auch annähernd gleiche Schmelztemperaturen aufweisen. Besonders wichtig ist dieser Umstand, wenn man komplexere Strukturen (z. B. Netze) aus Grundelementen aufbauen will und dabei in einem ersten Schritt die Grundelemente und in einem zweiten Schritt das Netz assemblieren möchte. Über einfache Temperaturregulierung ist dies dann erreichbar.

Anstelle des G/C-Testes könnte auch ein echter Test der erreichbaren Schmelztemperatur des aktuellen Doppelstranges stehen. Dieser Test ist jedoch aufwendiger.

4.6.3 Selbstkomplementäre Sequenzen und Masken

Eigentlich sind selbstkomplementäre Basensequenzen von den Critonregeln untersagt, weil sie die Festlegungen einer maximalen Fehlerlänge durchbrechen. Trotzdem wird es manchmal nötig oder wünschenswert sein, selbstkomplementäre Sequenzen in eine DNA-Struktur einzufügen. Viele Restriktionsenzyme besitzen selbstkomplementäre Erkennungssequenzen, zum Beispiel die beiden Enzyme aus Abbildung 2.7 auf Seite 21. Der Algorithmus sollte mit solchen vordefinierten Sequenzen umgehen können. Da die selbstkomplementären Sequenzen aber prinzipiell gegen die Critonregeln verstoßen, geht das nur, wenn man die entsprechenden Regionen von einer Prüfung der Regeln ausnimmt.

Dies geschieht durch Maskieren von Basenpaaren in der DNA-Struktur. Jedes Criton, welches mindestens eine maskierte Base enthält, ist dann von der Zuordnung zu einer Basissequenz ausgenommen. Bei der selbstkomplementären Sequenz GGATCC und einer angenommenen Critonlänge von 3 müssten mindestens die beiden inneren Basen A und T maskiert werden. Alle Critons, die diese Sequenz umfassen, sind dann von der Zuordnung zu einer Basissequenz befreit. Die nächsten Nachbarn dieser Critons, die nur teilweise innerhalb der Sequenz liegen, müssen dagegen wieder korrekt zugeordnet werden. Den gleichen Effekt erhält man, wenn man alle Basen der Sequenz maskiert, aber nur die Critons von der Zuordnung befreit, deren Basen alle maskiert sind. Hierfür sind verschiedene Varianten möglich.

Durch die Möglichkeit der Maskierungen lassen sich auch andere beliebige Sequenzen, die ihrer Natur nach gegen die Critonregeln verstoßen, in die DNA-Zielstruktur einfügen. Auch wenn mehrere Sequenzabschnitte mehrmals auftauchen sollen (z. B. mehrere Schlaufen aus Thymin-Basen, welche zur Erzeugung von Strangknicken benutzt werden), kann damit Abhilfe geleistet werden.

4.7 Seed - Eine Implementierung des Algorithmus

Das Programm namens Seed umfasst sowohl den Basisalgorithmus als auch die Erweiterungen aus Kapitel 4.6. Es ist so weit entwickelt, dass es allgemein zugänglich gemacht werden kann. Seed wurde in der Programmiersprache Java implementiert. Quell- und Binärcode von Seed, eine Nutzerdokumentation und einige Anwendungsbeispiele findet man im Anhang A.

Seed ist ein textorientiertes Programm, das über die Kommandozeile gestartet wird. Die wichtigste Eingabe beim Start ist die Beschreibung einer DNA-Zielstruktur, welche entweder aus einer Datei oder direkt aus der Kommandozeile entnommen wird. Die Strukturbeschreibung selbst erfolgt mittels fünf Typen von Strukturelementen. Es gibt Einzelstränge, Doppelstränge, Sequenzen, Variablen und Masken.

Einzel- und Doppelstränge definieren die physische Gestalt der Zielstruktur sowie die initiale Definition der Basenwerte. Jeder Basenwert kann einzeln festgelegt werden. Im einfachsten Fall ist er nicht definiert (N), er kann aber auch konstant (z. B. nur Guanin) oder eingeschränkt (z. B. nur Guanin oder Cytosin) sein. Mit Sequenzelementen kann die Beschreibung der Basensequenz von Einzelsträngen komplexer gestaltet werden. Sie ermöglichen es, Subsequenzen zu definieren, diese zum Beispiel als selbstkomplementär zu markieren und dann an beliebigen (auch mehreren) Stellen in die Struktur einzufügen. Der G/C-Basenpaar-Anteil jedes Doppelstranges kann individuell auf einen Zielwert festgelegt werden.

Variablen- und Maskierungselemente dienen nicht der eigentlichen Struktur- und Sequenzbeschreibung. Variablen sind ein Hilfsmittel, um beliebige Zahlen oder Zeichenketten, die in der Strukturbeschreibung öfters auftreten, speichern und ansprechen zu können. Masken definieren Sequenzbereiche, welche von der Überprüfung auf die Critonregeln ausgenommen sein sollen.

Beim Programmstart können neben der Strukturbeschreibung die folgenden Parameter angegeben werden:

- die Critonlänge,
- die Critonlänge in den Umgebungen der Verzweigungspunkte,
- die Reichweite der Umgebungen der Verzweigungspunkte,
- der globale Zielwert für den G/C-Basenpaar-Anteil,
- verbotene Subsequenzen,
- ein Maskierungslevel und
- eine zuvor ermittelte Sequenz-Konfiguration.

Zusätzlich besteht die Möglichkeit, selbstkomplementäre Sequenzen zu erlauben, alle Doppelstrangenden als G/C-Paar festzulegen und alle Basenpaar-Ringe der Verzweigungspunkte von der Prüfung auf die Critonregeln zu befreien. Außerdem gibt es Parameter, die die Informationsausgabe (generierte Sequenzen, Logdateien) steuern.

Außer der Strukturbeschreibung ist keiner der aufgeführten Parameter obligatorisch. Wird zum Beispiel keine Critonlänge spezifiziert, ermittelt Seed diese automatisch, wie in Kapitel 4.3.1 beschrieben.

Nach dem Programmstart arbeitet Seed alle Schritte aus den Kapiteln 4.1 bis 4.4 ab. Die DNA-Zielstruktur wird eingelesen (Kap. 4.1) und normalisiert (Kap. 4.2). Dabei werden automatisch die Verzweigungspunkte und die Doppelstrangabschnitte, in welchen letztendlich die Sequenzgenerierung stattfindet, lokalisiert. Danach folgt eine Vorbereitungsphase, in welcher, falls notwendig, die Critonlänge ermittelt (Kap. 4.3.1), die Critonstruktur aufgebaut (Abs 4.3.2) und die Verbindungen lokalisiert werden (4.6.1). Standardmäßig ist die Reichweite der Verbindungen die um eins verminderte Critonlänge. Sie kann vom Nutzer aber auch größer, jedoch nicht kleiner, eingestellt werden. Aus Effizienzgründen wurde der Aufbau des Sequenzgraphen (Kap. 4.3.3) aus der Vorbereitungs- in die Sequenzgenerierungsphase verlagert.

Während der eigentlichen Sequenzgenerierung versucht Seed zuerst, die Sequenzen der Umgebungen der Verzweigungspunkte (Verbindungen) mit einer eigenen, möglichst geringen Critonlänge zu generieren. Dadurch soll erreicht werden, dass bei großen DNA-Zielstrukturen, welche hohe Critonlängen erfordern, an den sensiblen Verzweigungspunkten weniger Fehlpaarungen auftreten. Gelingt die Generierung, werden die betroffenen Basenwerte anschließend konstant gesetzt und ändern sich im weiteren Verlauf nicht mehr. Falls nicht, werden die Verbindungen mit der globalen Critonlänge generiert, nachdem alle konstanten Sequenzbereiche ebenfalls mit der globalen Critonlänge in den Sequenzgraphen eingetragen wurden. Zuletzt erfolgt nacheinander die Sequenzgenerierung in den einzelnen Doppelstrangabschnitten.

Zu allen Schritten werden von Seed auf dem Bildschirm Meldungen ausgegeben, anhand derer der Programmablauf verfolgt werden kann. Informationen über die Sequenzgenerierung können auch in einer Logdatei ausgegeben werden.

Nach erfolgreicher Sequenzgenerierung werden die Sequenzen dargestellt, gespeichert und analysiert. Die Analyse zeigt an, welche Abschnitte in der Zielstruktur zueinander komplementär sind, welche G/C-Anteile und Schmelztemperaturen diese Bereiche aufweisen. Fehlpaarungen größer oder gleich der Critonlänge, die zum Beispiel durch Maskierungen verursacht werden, können so erkannt und bewertet werden. Die Sequenzanalyse kann auch ohne vorherige Sequenzgenerierung stattfinden. Dazu lässt sich eine vorher ermittelte Sequenz-Konfiguration in die Zielstruktur laden.

Schlägt eine Sequenzgenerierung fehl, kann das mehrere Ursachen haben. Wenn der Fehler beim Setzen der konstanten Sequenzbereiche auftritt, liegt das an Konflikten in der Sequenzbeschreibung selbst. Diese müssen durch Maskierung oder Änderung der Basenwerte gelöst werden. Tritt der Fehler während der Sequenzgenerierung in den Verbindungen oder Doppelstrangabschnitten auf, so kann ein erneuter Versuch trotzdem einen Erfolg bringen. Schlagen auch mehrere Versuche fehl bzw. dauert die Berechnung äußerst lange, kann es sein, dass wegen der Strukturgröße und/oder Einschränkungen in der Sequenzbeschreibung die Auslastung des Sequenzgraphen zu hoch ist (siehe Kap. 4.5). In diesem Fall schafft eine manuelle Erhöhung der Critonlänge Abhilfe.

Eine genaue Beschreibung des Progammes Seed findet sich in Anhang A.1. Seed Strukturbeschreibungen für einige DNA-Strukturen aus Kapitel 2.4 sind in Anhang A.2 nachzulesen.

Kapitel 5

Das DXL-Molekül – ein Experiment

Mit dem Sequenzdesign-Programm Seed bzw. dessen Vorläufern wurden die Basensequenzen für einige DNA-Strukturexperimente erstellt. Ein Experiment soll als Beispiel hier vorgestellt werden.

Ziel war es, eine langgestreckte und steife DNA-Struktur zu erzeugen, welche dann als Vorlage für einen Nanodraht dienen könnte. Natürlich kann jeder ganz normale DNA-Doppelstrang beliegig langgestreckt sein. Bei größeren Längen (über 150 Basenpaare) ist er aber nicht mehr steif, sondern faltet sich zusammen. Deshalb entstand die Idee, zwei Doppelstränge durch Kreuzungspunkte miteinander zu verknüpfen und dadurch die langgestreckte Form zu stabilisieren.

Als Grundelement für die Struktur dient ein Double-Crossover-Molekül (DX), dargestellt in Abbildung 5.1. Es besteht aus fünf DNA-Einzelsträngen: Einem zentralen Ringstrang (`RING`, grün), einem oberen und einem unteren Seitenstrang (`SO`, rot und `SU`, orange) sowie einem linken und einem rechten Verbindungsstrang (`VL`, hellblau und `VR`, dunkelblau). Nach der Terminologie in Arbeit [33] handelt es sich um ein DAE-Molekül. 'D' steht für Double-Crossover, 'A' für antiparallel, weil die untere und obere Helix entgegengesetzte Leserichtungen aufweisen, und 'E' steht für even (engl. für gerade), da zwischen beiden Kreuzungspunkten eine gerade Anzahl von halben Helixwindungen (in diesem Fall 2 ganze Windungen) liegt.

An den Enden des Grundelementes haben die beiden Verbindungsstränge `VL` und `VR` jeweils zwei Einzelstrangüberhänge. Über diese werden die Grundelemente zu einer Kette verbunden. Um eine eventuell auftretende Krümmung im Grundelement zu kompensieren, sind benachbarte Elemente zueinander um 180^o um ihre Längsachse verdreht. Erreicht wird das durch eine Überkreuz-Verknüpfung der Überhänge (der linke obere bindet an den rechten unteren und der linke untere an den rechten oberen) und durch einen Abstand von 2.5 Helixwindungen zwischen den Kreuzungspunkten benachbarter Elemente. Kleine Änderungen der Überhangkonfiguration könnten auch ein Netz erzeugen (siehe Abb. 2.15 und 2.16 auf S. 32 ff.).

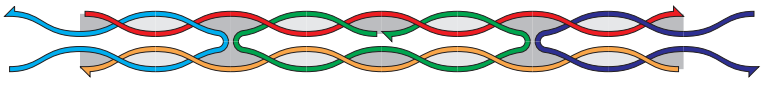


Abbildung 5.1: DXL-Grundelement

Abbildung 5.2: Darstellung der DXL-Struktur

```
# Definition der Einzelstraenge:
strand (RING, 42)   # grün
strand (SO, 41)     # rot
strand (SU, 41)     # orange
strand (VL, 32)     # hellblau
strand (VR, 32)     # dunkelblau

# Definition der Doppelstraenge des Grundelementes:
double (RING, SO, 0, 10, 11)
double (RING, SU, 11, 10, 21)
double (RING, SO, 32, 21, 10)
double (VL, SU, 6, 31, 10)
double (VL, SO, 16, 0, 10)
double (VR, SO, 6, 31, 10)
double (VR, SU, 16, 0, 10)

# Verknuepfung benachbarter Grundelemente:
double (VL, VR, 0, 0, 6)
double (VL, VR, 26, 26, 6)
```

Siehe Anhang A.1 für detaillierte Erläuterungen.

Tabelle 5.1: Seed-Strukturbeschreibung für die DXL-Struktur

Die entstehende langgestreckte Kette aus DX-Molekülen ist in Abbildung 5.2 zu sehen. Sie erhielt den Namen DXL: 'DX' für Double-Crossover und 'L' für lang.

Das Grundelement und die Verknüpfungen benachbarter Elemente wurden in einer Seed-Strukturbeschreibung formuliert. Diese ist in Tabelle 5.1 aufgelistet.

Mit der Strukturbeschreibung wurde eine Sequenzgenerierung durchgeführt. Zusätzliche Randbedingungen waren, dass keine Subsequenzen mit aufeinander folgenden Guanin-Basen auftreten und dass der G/C-Anteil aller Doppelstränge in etwa gleich sein sollte. Es stellte sich heraus, dass eine Critonlänge von 5 benötigt wird. Die Struktur mit 106 Basenpaaren liegt noch im Bereich von Critonlänge 4 (siehe Kap. 4.3.1). Allerdings erzeugt diese Strukturgröße einen sehr hohen Rechenaufwand (siehe Kap. 4.5), welcher durch die Einschränkungen aus den zusätzlichen Randbedingungen noch weiter erhöht wird. Die Erhöhung der Critonlänge brachte Abhilfe.

Die von Seed erzeugte Sequenzkonfiguration, mit welcher auch die Experimente durchgeführt wurden, ist in Tabelle 5.2 aufgelistet. Die dargestellten Sequenzen wurden mit dem Programmaufruf

```
# java Seed -lc 5 -gcf 0.5 -gcfr 0.05 -forbidden "GG" DXL.dat
```

`RING:`	GACTCGCTGTATCTCTAGTATGTGCTGCTTGCTCGTGAGTAC
`SO:`	CTCTCGAACTTACAGCGAGTCGTACTCACGATGTTCAGACG
`SU:`	GCTCATCTACGCAAGCAGCACATACTAGAGAAAACTGCGAC
`VL:`	CGAAACGTCGCAGTTTAGTTCGAGAGCTACAC
`VR:`	GTTTCGCGTCTGAACAGTAGATGAGCGTGTAG

Tabelle 5.2: Sequenzkonfiguration für die DXL-Struktur

erstellt, wobei `DXL.dat` die Strukturbeschreibung aus Tabelle 5.1 enthielt. Die Option `-lc 5` setzt die Critonlänge auf 5. Mit `-gcf 0.5` und `-gcf 0.05` wird die G/C-Anteile aller Doppelstränge auf Werte zwischen 45% und 55% festgelegt. Durch `-forbidden "GG"` sind zwei aufeinander folgende Guanin-Basen verboten.

Die Experimente fanden im Max-Bergmann-Zentrum für Biomaterialien der TU Dresden statt und wurden von Alexander Huhle durchgeführt. Das DNA-Material wurde bei der Biozym Scientific GmbH [56] beschafft.

Die Einzelstränge kamen in eine Pufferlösung mit 40 mM Tris Base, 2 mM EDTA und 12.5 mM $MgCl_2$. Die Endkonzentration der DNA lag bei 0.4 μM bei einer Menge von 500 μl.

Die Ausbildung der DXL-Moleküle erfolgte durch Abkühlen der Lösung von 95 ^{o}C auf 4 ^{o}C über einen Zeitraum von 16 Stunden. Mit dem Resultat wurden Gel-Elektrophoresen und AFM-Aufnahmen gemacht, um die Ergebnisse zu verifizieren.

Das Gel enthielt 6% Acrylamid (37,5:1 Acrylamid/Bisacrylamid) und einen Puffer aus 1xTBE und 10 mM $MgCl_2$. Das Gel lief eine Stunde mit einer Spannung von 12 $\frac{\mathrm{V}}{\mathrm{cm}}$ und wurde danach mit Gel Stain SYBR Green I von Molecular Probes [60] eingefärbt. Ergebnisse sind in Abbildung 5.3 zu sehen.

Für die AFM-Aufnahme wurde ein Tropfen (10 μl) der Lösung mit den assemblierten DXL-Molekülen auf einen Träger gegeben und dort für drei Minuten in Ruhe gelassen, um auf der Oberfläche absorbieren zu können. Danach wurden 15 μl Magnesiumchlorid-Lösung mit einer Konzentration von 10 mM hinzugegeben. Die Aufnahmen wurden in einer Flüssigzelle im Tapping-Modus mit einem Multimode NanoScope IIIa und NP-S-Spitzen von Veeco Instruments [61] durchgeführt. Eine Aufnahme ist in Abbildung 5.4 zu sehen.

Im Gelbild gibt es fünf vertikal verlaufende Bahnen mit Probenmaterial. Zwei zusätzliche Bahnen enthalten Maßstabsleitern. In jeder Probenbahn liefen unterschiedliche Strangkombinationen. Die jeweilige Kombination ist schematisch über der Bahn dargestellt. Am interessantesten ist die Bahn 5, bei welcher alle fünf Einzelstränge nach der Assemblierung analysiert werden. In dieser Bahn ist außer in der Starttasche keine klar abgegrenzte Bande mit DNA-Material zu erkennen. Lediglich ein leichter Schmier zieht sich über die gesamte Bahn. Man kann das so interpretieren, dass bei der Assemblierung sehr große Konstrukte entstanden, die im Gel gar nicht oder nur äußerst langsam laufen. Die meiste DNA bleibt deshalb in der Starttasche. Während der Stunde, in der die Gel-Elektrophorese läuft, zerbrechen jedoch einige der großen Konstrukte. Die Bruchstücke sind klein genug, um im Gel zu wandern. Da das Zerbrechen aber über die gesamte Laufzeit hinweg stattfindet, können sich gleich große Bruchstücke nicht in einer gemeinsamen Bande sammeln, sondern verteilen sich über die ganze Bahn.

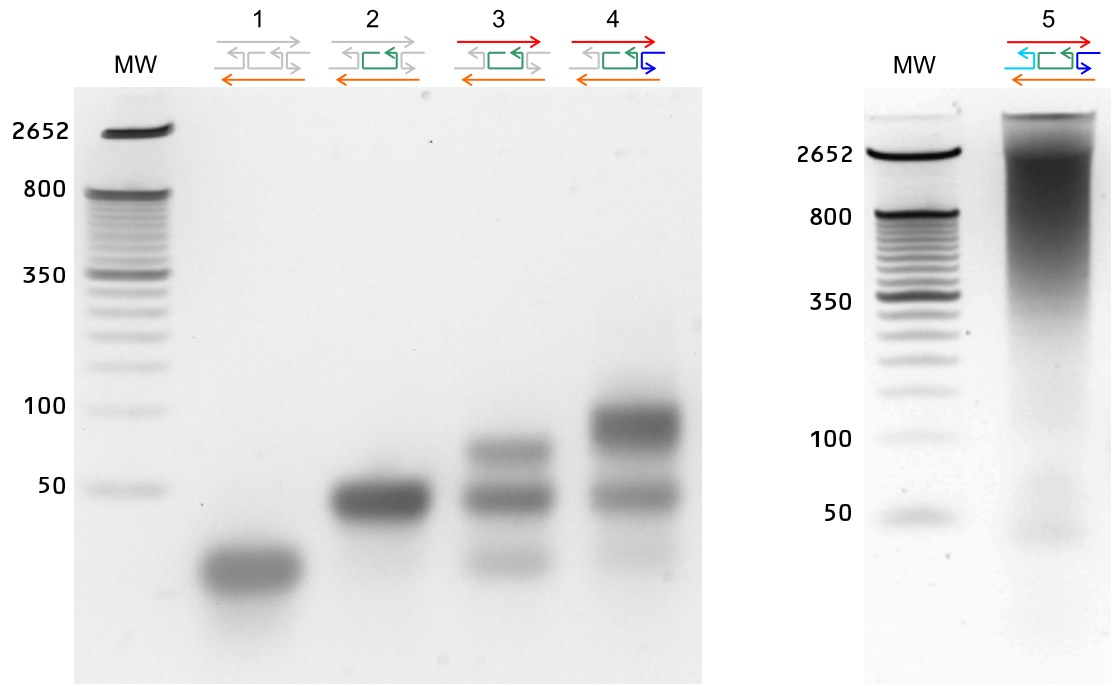


Abbildung 5.3: Ergebnisse der Gel-Elektrophorese

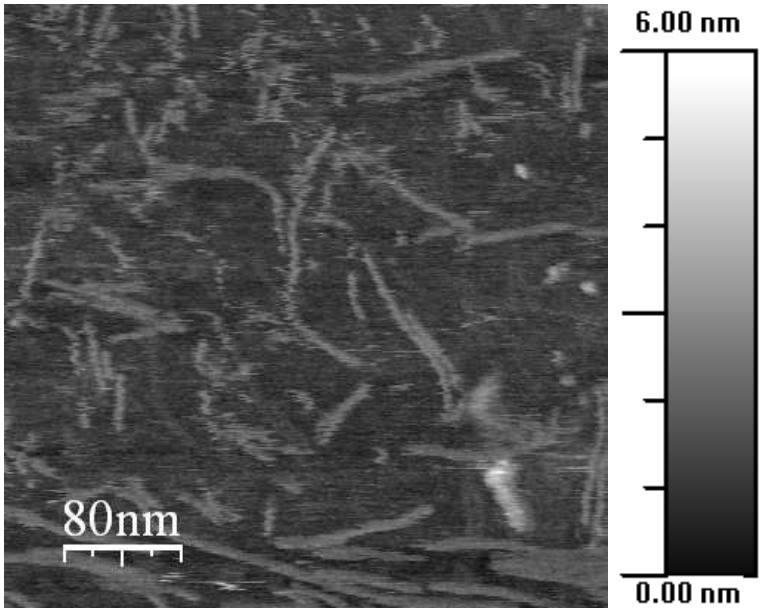


Abbildung 5.4: AFM-Aufnahme von DXL-Molekülen

Auskunft über die Gestalt der assemblierten Konstrukte liefert die AFM-Aufnahme. Auf ihr sind deutlich langgestreckte Formationen bis zu einer Länge von 240 Nanometern zu erkennen. Dies entspricht einem DXL-Molekül aus ca. 15 Grundelementen. Die Konstruktion war also erfolgreich. Man erkennt jedoch auch, dass die Moleküle nicht sehr gerade verlaufen, sondern Kurven und Knicke aufweisen. Ein Grund dafür könnten die relativ vielen Lücken im Strangrückgrat sein (sieh Abb. 5.2). Besonders die nah beieinander liegenden Lücken am Verknüpfungspunkt benachbarter Grundelemente könnten die Flexibilität verursachen. Eine Ligation, bei welcher die Lücken geschlossen würden, sollte da Abhilfe schaffen.

Kapitel 6

Zusammenfassung und Ausblick

Diese Dissertation hat einen Algorithmus vorgestellt, welcher für beliebige DNA-Zielstrukturen passende Sequenzkonfigurationen erzeugt. Eine passende Sequenzkonfiguration enthält neben den erwünschten komplementären Basensequenzen, welche den Aufbau der Zielstruktur steuern, keine weiteren unerwünschten komplementären Stellen über einer bestimmten Länge an Basen. Fehlpaarungen während der Selbstassemblierung können so den Aufbau der Zielstruktur im Labor nicht wesentlich stören.

Die Grundidee des Algorithmus ist es, alle Einzelstränge der DNA-Zielstruktur in gleich lange, sich überlappende Abschnitte, die Critons, zu zerlegen und jedem der Critons nach bestimmten Regeln eine Basissequenz zuzuordnen. Die Basissequenzen sind alle möglichen Sequenzen einer bestimmten Länge. Sie werden in einem Graphen gesammelt und miteinander verknüpft (siehe Kap. 3.2). Die Kombination von Critons und Basissequenzgraph erlaubt eine sehr schnelle Zuordnung der Sequenzen unter Beachtung der Regeln.

Der Sequenzdesign-Algorithmus ist deshalb für fast alle Strukturgrößen bis hin zu mehreren tausend Basenpaaren sehr schnell und effizient. Durch Messungen wurde ein durchschnittlicher Rechenaufwand von einem bis zwei Recheneinheiten (Basistests) pro Basenpaar ermittelt. Der exponentiell ansteigende Rechenaufwand in einigen wenigen Größenbereichen kann sehr einfach durch Erhöhung der zulässigen Fehlerlänge vermieden werden (siehe Kap. 4.5). Da die Fehlerlänge jeweils nur um eine Base steigt und nur fünf bis zehn Prozent aller Strukturen davon betroffen sind, erscheint dieses Vorgehen akzeptabel.

Der Algorithmus wurde in ein Java-Programm mit Namen Seed implementiert (siehe Kap. 4.7). Mit Seed wurden die Basensequenzen für einige DNA-Struktur-Experimente in der Arbeitsgruppe generiert. Die Software hat sich als nützlich und korrekt erwiesen. Eines der Experimente, bei welchem lang gestreckte DNA-Ketten aus Double-Crossover-Molekülen entstanden, wurde im Detail vorgestellt (siehe Kap. 5).

Der Algorithmus, das Programm Seed und das Experiment wurden veröffentlicht [1], um die Ergebnisse der Fachwelt vorzustellen.

Wenn das Sequenzdesign-Programm Seed auch in anderen Arbeitsgruppen Anwendung findet, ergeben sich sicherlich Änderungswünsche und Verbesserungsvorschläge. Diese können aus neuen Anwendungsfällen oder Wünschen der Benutzer hinsichtlich der Bedienung resultieren. Bereits jetzt sind einige Verbesserungsmöglichkeiten ersichtlich: Seed ist momentan ein rein textorientiertes Programm. Alle Angaben zur DNA-Zielstruktur

müssen per Tastatur erstellt und anhand der Text-Ausgaben des Programms verifiziert werden. Eine graphische Darstellung der eingelesenen Zielstruktur wie in Kapitel 2.4 oder 4.2 wäre da sehr hilfreich. Wegen der großen Vielfalt möglicher DNA-Strukturen und unterschiedlicher Strukturgrößen ist eine automatische Bildgenerierung eine große Herausforderung.

Ein nächster Schritt wäre eine graphische Erstellung der Strukturbeschreibung, so dass der Nutzer nur noch in Ausnahmefällen die Strukturdateien direkt bearbeiten muss. Dieser Schritt würde das Strukturdesign wesentlich beschleunigen. Er ist aber ebenfalls mit hohem Entwicklungsaufwand verbunden.

Der Kernalgorithmus selbst bietet ebenfalls Entwicklungsmöglichkeiten. Zum Beispiel können an den Stellen, an welchen die Basenwerte und die Sequenzen der Critons getestet werden, noch weitere bisher nicht berücksichtigte Bedingungen eingefügt werden. Beispiele dafür sind eine genaue Bestimmung der zu erwartenden Schmelztemperatur eines Doppelstranges oder eine Energieminimierung.

Aktuell läuft ein Projekt, welches den Algorithmus in seinem jetzigen Stand in eine Web-Anwendung integriert. Dies bietet für den Nutzer einige Vorteile: Er muss sich nicht mehr um Beschaffung, Installation und Pflege des Programms kümmern. Ein Nutzeraccount auf einer Webseite genügt. Zudem wird die Anwendung sehr viel mehr interaktiv gestaltet sein. Man kann dann direkt in die einzelnen Programmphasen eingreifen und Parameter verändern. Dadurch können schnell verschiedene Szenarien getestet werden. Letztendlich bietet die Web-Anwendung auch eine komfortablere Verwaltung der Strukturbeschreibungen und Sequenzkonfigurationen sowie ein Zugangsmanagement. Die Datenhaltung wird dadurch erleichtert.

Eine Herausforderung ist die genaue mathematische Beschreibung des Rechenaufwandes, den der Algorithmus für eine bestimmte Strukturgröße benötigt. Trotz einiger Anstrengungen ist das bisher noch nicht gelungen. Eine mathematische Beschreibung würde die Vorhersage, ab wann eine Sequenzgenerierung ineffizient wird, präzisieren. Die Einbeziehung von Verzweigungspunkten und von Randbedingungen zu den Sequenzen wäre ebenfalls nützlich. Damit könnte dann die für eine konkrete Zielstruktur erforderliche Critonlänge noch genauer bestimmt werden. Vor allem sollte eine mathematische Beschreibung die Frage klären, warum der Rechenaufwand bei Verwendung einer geraden Critonlänge deutlich eher ansteigt als bei einer ungeraden Critonlänge. Die dadurch gewonnenen Erkenntnisse könnten in eine Weiterentwicklung des Algorithmus einfließen und man könnte zudem ausschließen, dass es sich bei dem Effekt um ein Programmierartefakt handelt.

Literaturverzeichnis

[1] J. Seiffert and A. Huhle, *A Full-Automatic Sequence Design Algorithm for Branched DNA Structures*, J. Biomol. Struct. Dyn. *25*(5), 453-466 (2008)

[2] U. Feldkamp und C.M. Niemeyer, *Rationaler Entwurf von DNA-Nanoarchitekturen*, Angew. Chem. **118**(12), 1888-1910 (2006)

[3] H. Liu, Y. Chen, Y. He, A.E. Ribbe, and C. Mao, *Approaching the Limit: Can One DNA Oligonucleotide Assemble into Large Nanostructures*, Angew. Chem. Int. Ed. **45**(12), 1942-1945 (2006)

[4] R.P. Goodman, I.A.T. Schaap, C.F. Tardin, C.M. Erben, R.M. Berry, C.F. Schmidt, and A.J. Turberfield, *Rapid Chiral Assembly of Rigid DNA Building Blocks for Molecular Nanofabrication*, Science **310**(5754), 1661-1665 (2005)

[5] J. Malo, J.C. Mitchell, C. Vinien-Bryan, J.R. Harris, H. Wille, D.J. Sherratt, A.J. Turberfield, *Engineering a 2D Protein-DNA Crystal*, Angew. Chem. Int. Ed. **44**(20), 3057-3061 (2005)

[6] Y. He, Y. Chen, H. Liu, A.E. Ribbe, and C. Mao, *Self-Assembly of Hexagonal DNA Two-Dimensional (2D) Arrays*, J. Am. Chem. Soc. **127**(35), 12202-12203 (2005)

[7] D. Reishus, B. Shaw, Y. Brun, N. Chelyapov, and L. Adleman, *Self-Assembly of DNA Double-Double Crossover Complexes into High-Density, Doubly Connected, Planar Structures*, J. Am. Chem. Soc. **127**(50), 17590-17591 (2005)

[8] Z. Shen, H. Yan, T. Wang, and N.C. Seeman, *Paranemic Crossover DNA: A Generalized Holliday Structure with Applications in Nanotechnology*, J. Am. Chem. Soc. **126**(6), 1666-1674 (2004)

[9] 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)

[10] J. SantaLucia, Jr. and D. Hicks, *The Thermodynamics of DNA Structural Motifs*, Annu. Rev. Biophys. Biomol. Struct. **33**, 415-440 (2004)

[11] P. Sa-Ardyen, N. Jonoska, and N. C. Seeman, *Self-Assembly of Irregular Graphs Whose Edges Are DNA Helix Axes*, J. Am. Chem. Soc. **126**(21), 6648-6657 (2004)

[12] N. Chelyapov, Y. Brun, M. Gopalkrishnan, D. Reishus, B. Shaw, and L. Adleman, *DNA Triangles and Self-Assembled Hexagonal Tilings*, J. Am. Chem. Soc. **126**(43), 13924-13925 (2004)

[13] R. Seidel, L. Colombi Ciachi, M. Weigel, W. Pompe, and M. Mertig, *Synthesis of Platinum Cluster Chains on DNA Templates: Conditions for a Template-Controlled Cluster Growth*, J. Phys. Chem. B. **108**(30), 10801-10811 (2004)

[14] 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)

[15] 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*, Proc. Natl. Acad. Sci. **101**(3), 717-722 (2004)

[16] P. Yin, B. Guo, C. Belmore, W. Palmeri, E. Winfree, T.H. LaBean, and J.H. Reif, *TileSoft: Sequence Optimization Software For Designing DNA Secondary Structures*, http://www.cs.duke.edu/ py/paper/dnaTileSoft/, (2004)

[17] R.M. Dirks, M. Lin, E. Winfree, and N.A. Pierce, *Paradigms for Computational Nucleic Acid Design*, Nucl. Acids Res. **32**(4), 1392-1403 (2004)

[18] 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**(5641), 1882-1884 (2003)

[19] D. Liu, J.H. Reif, and T.H. LaBean, *DNA Nanotubes: Construction and Characterization of Filaments Composed of TX-tile Lattice*, pp. 10-21 in: *DNA Based Computers (DNA8)* (Eds: M. Hagiya and A. Ohuchi), Springer-Verlag, New York (2003)

[20] C.F. Monson and A.T. Woolley, *DNA-Templated Construction of Copper Nanowires*, Nano Letters **3**(3), 359-363 (2003)

[21] U. Feldkamp, H. Rauhe, and W. Banzhaf, *Software Tools for DNA Sequence Design*, Genet. Prog. Evolv. Mach. **4**(2), 153-171 (2003)

[22] H. Yan, T.H. LaBean, L. Feng, and J.H. Reif, *Directed Nucleation Assembly of DNA Tile Complexes for Barcode-Patterned Lattices*, Proc. Natl. Acad. Sci. **100**(14), 8103-8108 (2003)

[23] L. Feng, S.H. Park, J.H. Reif, and H. Yan, *A Two-State DNA Lattice Switched by DNA Nanoactuator*, Angew. Chem. Int. Ed. **42**(36), 4342-4346 (2003)

[24] N.C. Seeman, *DNA in a Material World*, Nature **421**, 427-431 (2003)

[25] T.H. LaBean, *Introduction to Self-Assembling DNA Nanostructures for Computation and Nanofabrication*, Chapter 2 in: *Computational Biology and Genome Informatics* (Eds.: J.T.L. Wang, C.H. Wu, and P.P. Wang), World Scientific Publishing, Singapore (2003)

[26] R. Seidel, *Methods for the Development of a DNA Based Nanoelectronics*, Dissertation, Technische Universität Dresden, Fakultät Mathematik und Naturwissenschaften (2003)

[27] M. Mertig, L. Colombi Ciacchi, R. Seidel, W. Pompe, and A. De Vita, *DNA as a Selective Metallization Template*, Nano Letters **2**(8), 841-844 (2202)

[28] M. Arita and S. Kobayashi, *DNA Sequence Design Using Templates*, New Gen. Comp. **20**(3), 263-277 (2002)

[29] R. Rohs, *Simulation der Strukturbildung und Ligandenbindung von Nukleinsäuren im Raum kollektiver und innerer Variablen*, Dissertation, Freie Universität Berlin, Fachbereich Biologie, Chemie, Pharmazie (2002)

[30] N.C. Seeman, *DNA Nicks and Nodes and Nanotechnology*, Nano Letters **1**(1), 22-26 (2001)

[31] A. Brenneman and A.E. Condon, *Strand Design for Bio-Molecular Computation*, Theor. Comp. Sci. **287**(1), 39-58 (2001)

[32] T.A. Taton, R.C. Mucic, C.A. Mirkin, and R.L. Letsinger, *The DNA-Mediated Formation of Supramolecular Mono- and Multilayered Nanoparticle Structures*, J. Am. Chem. Soc. **122**(26), 6305-6306 (2000)

[33] T.H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J.H. Reif, and N.C. Seeman, *Construction, Analysis, Ligation, and Self-Assembly of DNA Triple Crossover Complexes*, J. Am. Chem. Soc. **122**(9), 1848-1860, (2000)

[34] C.A. Mirkin, *Programming the Assembly of Two- and Three-Dimensional Architectures with DNA and Nanoscale Inorganic Building Blocks*, Inorg. Chem. **39**(11), 2258-2272 (2000)

[35] C. Mao, W. Sun, and N.C. Seeman, *Designed Two-Dimensional DNA Holliday Junction Arrays Visualized by Atomic Force Microscopy*, J. Am. Chem. Soc. **121**(23), 5437-5443 (1999)

[36] U. Feldkamp, *Ein DNA-Sequence-Compiler*, Diplomarbeit, Universität Dortmund, Fachbereich Informatik, Lehrstuhl XI (1999)

[37] 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 (1999)

[38] S. Nakano, M. Fujimoto, H. Hara, and N. sugimoto, *Nucleic Acid Duplex Stability: Influence of Base Composition on Cation Effects*, Nucl. Acids Res. **27**(14), 2957-2965 (1999)

[39] R.C. Mucic, J.J. Storhoff, C.A. Mirkin, and R.L. Letsinger, *DNA-Directed Synthesis of Binary Nanoparticle Network Materials*, J. Am. Chem. Soc. **120**(48), 12674-12675 (1998)

[40] E. Winfree, F. Liu, L.A. Wenzler, and N.C. Seeman, *Design and Self-Assembly of Two-Dimensional DNA Crystals*, Nature **394**, 539-544 (1998)

[41] J. SantaLucia, Jr. H.T. Alawi, and P. Ananda Seneviratne, *Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability*, Biochemistry **35**(11), 3555-3562 (1996)

[42] 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)

[43] Y. Zhang and N.C. Seeman, *Construction of a DNA-Truncated Octahedron*, J. Am. Chem. Soc. **116**(5), 1661-1669 (1994)

[44] J.H. Chen and N.C. Seeman, *Synthesis from DNA of a molecule with the connectivity of a cube*, Nature **350**, 631-633 (1991)

[45] N.C. Seeman, *De Novo Design of Sequences for Nucleic Acid Structural Engineering*, J. Biomol. Struct. Dyn. **8**(3), 573-581 (1990)

[46] L.J. Breslauer, R. Frank, H. Blocker, and L.A. Marky, *Predicting DNA Duplex Stability from the Base Sequence*, Proc. Natl. Acad. Sci. **83**(11), 3746-3750 (1986)

[47] N.C. Seeman, *An Immobile Nucleic Acid Junction Constructed from Oligonucleotides*, Nature **305**, 829-831 (1983)

[48] N.C. Seeman and N.R. Kallenbach, *Design of Immobile Nucleic Acid Junctions*, Biophys. J. **44**(2), 201-209 (1983)

[49] N.C. Seeman, *Nucleic Acid Junctions and Lattices*, J. Theor. Biol. **99**(2), 237-247 (1982)

[50] R.B. Wallace, J. Shaffer, R.P. Murphy, J. bonner, T. Hirose, and K. Itakura, *Hybridization of Synthetic Oligodeoxyribonucleotides to FX 174 DNA: The Effect of Single Base Pair Mismatch*, Nucl. Acids Res. **6**(11), 3543-3557 (1979)

[51] P.M. Howley, M.F. Israel, M.F. Law, and M.A. Martin, *A Rapid Method for Detecting and Mapping Homology between Heterologous DNAs: Evaluation of Polyomavirus Genomics*, J. Biol. Chem. **254**(11), 4876-4883 (1979)

[52] R. Holliday, *A Mechanism for Gene Conversion in Fungi*, Genet. Res. **5**, 282-304 (1964)

[53] J. Marmur and P. Doty, *Determination of the Base Composition of Desoxyribonucleic Acid from its Thermal Denaturation Temperature*, J. Mol. Biol. **5**, 109-118 (1962)

[54] J.D. Watson and F. Crick, *Molecular Structure of Nucleic Acids: A Structure for DNA*, Nature **171**, 737-738 (1953)

[55] *DNA Design Toolbox* (DNAdesign), http://www.dna.caltech.edu/DNAdesign/

[56] Biozym Scientific GmbH, http://www.biozym.com

[57] VBC Genomics, http://www.vbc-genomics.com

[58] New England Biolabs, http://www.neb.com

[59] Promega Corporation, http://www.promega.com

[60] Invitrogen Corporation, http://www.invitrogen.com

[61] Veeco Instruments, http://www.veeco.com

Anhang A

Seed

Das Sequenzdesign-Programm Seed kann auf der Internetseite

`http://nano.tu-dresden.de/~jseiffert/Seed/`

frei heruntergeladen werden. Das Programmpaket umfasst den ausführbaren Programmkode, ein Benutzerhandbuch und einige Strukturbeschreibungs- und Sequenzdateien. Es folgen hier gedruckt das Benutzerhandbuch und einige Beispiele für Strukturbeschreibungen mit Seed.

A.1 Seed User Manual

Seed - Sequence Design for branched DNA Structures
Version 1.0 (August 2007)

A.1.1 Introduction

This program Seed is meant to generate appropriate base sequences for the construction of DNA nano structures. The central point of the sequence generation algorithm is to limit the maximum mismatch length to a certain value by checking the sequences against the criton rules [N.C. Seeman, J. Theor. Bio. 99, 237-247]. Several additional constraints, such as G/C pair fraction or forbidden subsequences can be specified. A full description of the underlaying algorithm can be read in Seiffert & Huhle, J. Biomol. Struct. Dyn., ?? (2007).

A.1.2 Installation

Seed is installed by extracting the distributed zip-file. There will appear a directory called `Seed`. Change to it. It contains the subdirectories `bin`, `doc` and `structures`. In the `Seed/bin` directory, there is a jar-file `Seed.jar` containing the program. If you like, you can extract this archive and add the directory to your `CLASSPATH` variable to be found by Java.

The `Seed/doc` directory contains this manual. Furthermore, in `Seed/structures`, there are some structure description files and suitable sequence settings as examples.

Seed is to be started by the command:

```
# java -jar bin/Seed.jar [options] structure-file
```

or:

```
# java Seed [options] structure-file
```

if the jar-archive has been extracted and the `CLASSPATH` variable points to the `Seed/bin` directory. Seed has been developed using Java Version 1.4.2. You will need a suitable Java Version for your operating system. Look at http://java.sun.com to find it.

A.1.3 Running The Program

The last command line argument is expected to be the filename of a structure description file. Seed will read the description of the target DNA structure from this file. The syntax of structure description files will be explained in section A.1.4.

The following options can be added to the command:

`-h`	Print help screen.
`-q`	Quiet mode.
`-a`	Analyze sequences (no sequence generation).
`-seq file`	Load sequence setting from `file` to the target DNA structure. Sequence files are generated by Seed, but also can be created by the users themselves. The syntax is explained in section A.1.5.
`-lc n`	Force criton length. Default is auto-detect.
`-lcjunc n`	Force junction criton length. Default is auto-detect.
`-jrange n`	Force junction range. Default is criton length minus 1.
`-gcf x`	Set global target G/C pair fraction. Default is 0.5.
`-gcfr x`	Set global target G/C pair fraction range. Default is 1. After sequence generation, the G/C pair fraction of each doublestrand will be between `gcf-gcfr` and `gcf+gcfr`. Different values for individual doublestrands can be defined in the structure description file (section A.1.4).
`-gcends`	Set terminal base pairs of doublestrands to G/C. If a terminal base has different base values according to the structure description, those settings will be preferred.
`-forbidden string`	Define forbidden subsequences. The string must start and end with a quotation mark and may include several descriptions of subsequences, devided by a space character. A subsequence description may contain the characters `G`, `A`, `T`, `C` for the four bases and `N` for the undefined value. All other characters will be treated as `N`. Examples:

	`-forbidden "GGG"`
	`-forbidden "GGG AT"`
	The complements of a forbidden subsequence will also be forbidden.

`-sc`	Allow self-complementary sequences.
`-noloops`	Do not check base pair loops around branch points.
`-masklevel n`	Set mask level. Default is 1. Critons that include at least `n` masked bases will not be checked and therefore, can have any sequence.
`-result file`	Set result file name. Default is `results.log`. The result file will store the sequence setting produced by the sequence generation.
`-log`	Log sequence generation information.
`-logfile file`	Set file for storing sequence generation information. Default is `generation.log`.
`-nofile`	Handle last command line argument as structure description rather than a filename.

When starting Seed, it reads the structure data from the specified structure description file or from the command line. While reading, there might occur some errors, if the structure description is wrong. Seed will print the error information and break, then.

If reading has been finished successfully, the target DNA structure will be printed. Sequences and strands are displayed by their name, along with their length between brackets and the base sequence. Example:

```
A (25): <CAGAC>NNNNNNNNNNNNNNNNNNNN
```

All sequences are listed from the 5' to the 3' end. Sequences between `<` and `>` are constant. So, the displayed strand starts with five bases that have the constant sequence `CAGAC` followed by 20 bases with undefined base values.

A doublestrand is displayed by its location in the structure, its length between brackets, and its base sequence again. Example:

```
A:5-14/B:15-24 (10): NNNNNNNNNN
```

The doublestrands location includes all of its bases in the first strand (before the `/` character) and in the second strand (behind the `/` character). The displayed doublestrand ranges from base 5 to base 14 on the strand `A` and from base 15 to base 24 on the strand `B`. Its length is 10. The shown sequence is that of the bases of the first strand.

Variables are displayed by their name along with their values:

```
l = 10
```

Masks are displayed by their locations:

```
A:0-4 (5)
```

The example describes a mask over the first five bases of strand A. This sequence will not be checked during sequence generation.

Next step is the normalization of the DNA structure, which makes it double stranded everywhere. The normalized structure consists of double stranded sections and branch points. Those are printed. Sections are displayed like doublestrands, but their locations might be more complex, because sections can include parts of several strands as well as virtual bases. The character ~ indicates virtual bases. Example:

```
A:0-14/B:15-24,A:4-0~ (15): <CAGAC>NNNNNNNNNN'
```

This section ranges from base 0 to base 14 of strand A. The complementary bases are base 15 to 24 on strand B and five virtual bases that are bound to the first five bases of strand A. Its length is 15. The ' character at one end of the sequence indicates a branch point.

Branch points are displayed by their location in the structure, their number of arms, and the base values of their base pair loops. The location is described by a list of the bases that have the branch point at their 5' end. The shown base values are those of these bases. If the base pair loop of the branch point is closed this will be indicated at the end. For example:

```
A:15,B:15,C:15 (3): NNN (closed)
```

This branch point is located at the 5' end of the bases A:15, B:15 and C:15 and has three arms. The values of its base pair loop are undefined. The loop is closed.

After normalization, Seed makes some preparations, such as calculating the criton length (if not defined by command line option -lc), building the criton structure, and fixing the junctions. Junctions do contain branch points and the bases next to them. The bases form the junction environment. The dimension of the junction environments is defined by the junction range. Normally, it is the criton length minus 1, but can be changed with the -jrange option. All fixed junctions are printed, including all branch points of the junction and a description of the junction environment. This description contains a list of the base pairs in the environment followed by their base values. The base pairs are arranged by their distance to a branch point. For example:

```
1. branch point:
      A:15,B:15,C:15 (3): NNN (closed)
   junction section:
      A:15/C:14,C:15/B:14,B:15/A:14,\
      A:16/C:13,C:16/B:13,B:16/A:13,\
      A:17/C:12,C:17/B:12,B:17/A:12:\
      'N 'N 'N NNNNNN
```

The shown junction includes the branch point shown above and has a range of 3. Therefore, there are nine base pairs in the junction environment, three on each arm of the branch point.

Seed will try to generate the sequences of the junction environments first, using the smallest possible criton length for it. This junction criton length may also be defined by the -lcjunc option.

After preparation, the sequence generation starts. As already mentioned, Seed tries to generate the sequences in the junction environment with the smallest possible criton length first. Then, constant sections will be set and finally, the sequences of all double stranded sections will be generated.

Each of the generation steps might fail. Junction environment generation will be repeated 10 times at most. Each failure is indicated by a point. If each of them or any other generation step fails, the whole generation will fail. If the `-log` option is set, Seed will print internal generation information to the log file. Normally, its name is `generation.log`, but may be changed by the `-logfile` option.

If sequence generation has failed, another run may succeed, anyway. Another possibility is to increase the criton length by the `-lc` command line option. If this does not work, there certainly are some sequence restrictions according to the structure description that get in the way. Failures that occur during setting constant sections cannot be worked out by the program. The concerned sequences must be changed or masked.

If the sequence generation has been successful, Seed will print the sequences of all strands and the configuration of every branch point. The way of displaying them is described above. Branch points with an exclamation mark (`!`) in front do have an instable base pair loop configuration. The sequences will be stored in the `results.log` file. The name of the result file may be changed by the `-results` option.

Additionally, Seed makes a sequence analysis. It searches for sections with complementary sequences and shows them to the user. A complementary section is displayed like a normal double stranded section supplemented by its G/C pair fraction, its free energy ΔG^0_{37} and its melting temperature. The melting temperature of sequences shorter than 9 bases is calculated by

$$T_m = 4 \cdot \#GC + 2 \cdot \#AT + 16.6 \cdot \lg([\mathrm{Na}^+]/0.05),$$

where $\#GC$ is the number of guanine and cytosine bases, $\#AT$ is the number of adenine and thymine bases in the sequence, and $[\mathrm{Na}^+] = 0.2\mathrm{M}$ is the concentration of monovalent cations in the solution. Up to a length of 60 bases, the melting temperature is calculated using the nearest-neighbor model. Basic thermodynamic data is taken from Santalucia and Hicks (2004), Annu. Rev. Biophys. Biomol. Struct. 33:415-440 [10]. The required concentration of the DNA is set to $5 \cdot 10^{-6}\mathrm{M}$. $[\mathrm{Na}^+]$ is 0.2M again.

If the sequence is longer than that, the melting temperature is calculated by

$$T_m = 78.9 + (41 \cdot \#GC - 820)/length + 16.6 \cdot \lg([\mathrm{Na}^+]/0.05),$$

but these values may be not very accurate. The free energy is always calculated using the nearest-neighbor model.

Example:

```
A:5-14/B:15-20 (10): GGTTGGACTT
   gcf = 0.5, dG^0_{37} = -8.9289 kcal/mol, T_m = 39.4 grd C
```

This complementary section is the doublestrand already shown above, but now with a defined base sequence. Its melting temperature is 44.3 ^{o}C. If there are some complementary sections that are not meant to be complementary according to the structure description, they will be indicated by a exclamation mark.

If the `-a` option is set, the sequence analysis will be done without making a sequence generation before. By using the `-seq` option, various sequence settings for a target DNA structure can be analyzed in this way.

A.1.4 Structure Description Files

The description of the target DNA structure consists of a number of structure elements. There are five different element types: variables, sequences, strands, doublestrands, and masks. Strands and doublestrands constitute the DNA structure. Variables and sequences allow complex sequence specification and masks exclude certain sequence regions from checking.

The specification of each single structure element starts with a key word indicating the element type (`var, sequence, strand, double, mask`). Then, a tuple follows, which contains the description of the element.

Variable Elements

The key word for variable elements is `var`. A variable does have a name and a value. It is defined by:

```
var (name, value)
```

For example:

```
var (l, 10)
```

The value of a variable can be accessed by `$name` in later expressions. If the variable name contains other characters than letters, put it between quotation marks such as `$"l1"`. The base value sequences of strands and sequence elements also can be accessed in this way (see below). The variable value may be a simple number or string, but also a math expression, such as:

```
var (b, 2 * $a)
```

The following operators are supported:

Operator	Operation
`$x`	value of variable `x`
`x+y`	addition
`x-y`	subtraction
`x*y`	multiplication
`x/y`	division
`-x`	negation
`x^y`	exponentiation and string concatenation (`N^3` = `NNN`)
`//x`	square root
`y//x`	root
`log x y`	logarithm
`ln x`	natural logarithm
`sin x`	sinus
`cos x`	cosinus
`tan x`	tangent
`asin x`	arc sinus
`acos x`	arc cosinus
`atan x`	arc tangent

Strand and Sequence Elements

The key word for a strand element is `strand`. An element must contain the name of the strand and an expression that describes the base sequence:

```
strand (name, expression)
```

The simplest form of the sequence description is a number defining the length of the strand:

```
strand (A, 25)
```

In that case, this definition would produce a strand `A` of 25 bases length. Each base value would be undefined.

The sequence description may also be a simple string. In this case, the length of the string defines the length of the strand. Each character of the string defines the initial base values:

```
strand (A, CAGACNNNNNNNNNNNNNNNNNNN)
```

If so, the strand `A` would have a length of 25 bases, too, but its first five bases would have the constant values `CAGAC`. A single base value can be set to be undefined (`N`), constant (e.g. to guanine), or to some limited values (e.g. guanine or cytosine). The following characters may be used:

Character	Base Value(s)
G	guanine
A	adenine
T	thymine
C	cytosine
S	G or C
W	A or T
R	G or A
Y	T or C
M	A or C
K	G or T
H	A, T, or C (not G)
B	G, T, or C (not A)
V	G, A, or C (not T)
D	G, A, or T (not C)
N	G, A, T, or C

All other characters will be treated as N.

Additionally, the sequence description for a strand may consist of a number of expressions describing subsequences of the strand. For example, the strand A can also be described by:

```
strand (A, CAGAC N^20)
```

The first subsequence CAGAC is the one with constant base values. The second subsequence contains 20 bases with yet undefined values. All operators mentioned in the variable element section above may be used here, as well. To include the base values of other strands or sequences is allowed, too (see below).

The key word for a sequence element is sequence. Basically, a sequence is specified in the same way like a strand:

```
sequence (name, expression)
```

The difference is, that a sequence is not related to a certain strand. It is a list of base values, not of bases, as a strand is. A sequence is meant to be included into the sequence description of a strand.

```
sequence (S1, CAGAC)
strand (A, $"S1" N^20)
```

This definition would produce the same strand A, as described above.

A sequence can be marked to be self-complementary. An extra field containing the key word 'sc' must be added, then:

```
sequence (S, 6, sc)
```

This definition would produce a six base values long sequence that will be always self-complementary. If it is inserted into a strand, its respective subsequence will stay self-complementary during the sequence generation. If self-complementary sequences are

used, the `-sc` command line option must be set. Due to internal arrangements, self-complementary sequences do not have to be masked, although they violate the criton rules.

Sequences might be inserted at different locations in the DNA structure, such as:

```
sequence (loop, TTTT)
strand (L, N^10 $loop N^11 $loop n^10)
```

This would insert the base value of `loop` two times into strand `L`. If `loops` do not appear too often (≤ 4 times), it is not necessary to mask them.

The complementary sequence of a sequence element can be accessed by the `~` operator behind the sequence name.

Please note: variables, sequences and strands must have unique names.

Doublestrand Elements

The key word for doublestrand elements is `double`. Basically, there are five specifications to be made: the name of the two single strands involved, the positions of the bases, where the doublestrand starts, and the entire length:

```
double (name1, name2, start1, start2, length)
```

An example would be:

```
double (A, B, 5, 15, 10)
```

This would define a doublestrand that ranges from base 5 to 14 on strand `A` and from base 15 to 24 on strand `B`. Its length would be 10 base pairs. Of course, both strands must be defined before with suitable lengths. Note: The positions of bases on strands start with 0. A base can only bind to one other base. Therefore, defining overlapping doublestrands would cause errors.

The definition of the position of the doublestrand might also include math expressions that can be solved into numbers. Example:

```
double (A, B, 5, 5+$l, $l)
```

If the value of the variable `l` is 10, this definition would be equal to the one above.

Target G/C pair fractions can be given to every doublestrand. In this case, the doublestrand specification is extended by a sixth tuple field, whose content must start with the key word `gcf`, followed by a tuple containing the target G/C fraction and a range, both between 0 and 1:

```
double (name1, name2, start1, start2, length [, gcf (tgcf,range)])
```

After sequence generation, the G/C pair fraction of the specified doublestrand will lie between `tgcf-range` and `tgcf+range`. Example:

```
double (A, B, 5, 15, 10, gcf (0.6,0.1))
```

This would force the sequence generation to give this doublestrand a G/C pair fraction between 0.5 and 0.7. Predictions made for all doublestrands by the command line options `-gcf` and `-gcfr` will be overruled by settings in the structure description.

Mask Elements

It sometimes might be necessary to prevent some regions of the DNA structure from being checked against the criton rules. This can be achieved by masks. The key word for mask elements is `mask`. The definition itself contains the name of a strand or a sequence and start index and length of the region to be masked:

```
mask (name, start, length)
```

For example:

```
mask (A, 0, 5)
```

This would mask bases 0 to 4 on strand `A`.

Complements of masked bases will be masked, too. If a sequence is masked, the mask will hold for every base containing a base value of this sequence:

```
sequence (S1, CAGAC)
strand (A, $"S1" N^20)
mask (S1, 0, 5)
```

This would mask the whole sequence `S1`. Due to strand `A` includes `S1`, its first five bases will be masked, too. All critons that contain a certain number of masked bases won't be checked. The certain number is called mask level and can be set by the `-masklevel` command line option. By default, the mask level is set to 1. If the criton length would be 4, in the example this would concern criton `A:0-3`, `A:1-4`, `A:2-5`, `A:3-6`, `A:4-7`, and their complements.

Comments

All characters that stand behind a `#` character in a line are treated as comments.

A complete structure description for a simple three armed junction may look like this:

```
# Structure description for a
# three-armed junction
#
# Strand definition:
sequence (S1, CAGAC)
strand (A, $"S1" N^20)
strand (B, 25)
strand (C, NNNNNNNNNNNNNNNNNNNNNNNNN)
# doublestrand definition:
double (A, B, 15, 5, 10)
double (B, C, 15, 5, 10)
double (C, A, 15, 5, 10)
```

Please read the structure description files (`*.dat`) in the `Seed/structures` directory for some more examples.

A.1.5 Sequence Files

A sequence file contains a certain sequence configuration for a DNA structure. The definition is done by using sequence elements that were described in the previous section. Therefore, a sequence file is a structure description file that contains only sequence elements. An element in the sequence file does have a name, which in this case, must be the name of a strand or a sequence in the structure description, and a string defining the base sequence. For example:

```
sequence (S1, CAGAC)
sequence (A, CAGACGGTTGGACTTCCGCCTCCTT)
sequence (B, CGAATAAGGAGGCGGGTCGTGGTGA)
sequence (C, CACACTCACCACGACAAGTCCAACC)
```

This would be a sequence setting, which can be applied to the DNA structure described at the end of section A.1.4, by using the `-seq` command line option.

Seed stores the results of each successful sequence generation into a sequence file, named `results.log`. The name of this result file can be changed by the `-result` command line option. However, users may also create their own sequence files.

A.1.6 Bugs

There might be still some more or less important bugs in the program. Please report them to:

Jan Seiffert (seiffert@tmfs.mpgfk.tu-dresden.de).

Thank you very much.

If bigger criton lengths (> 7) are used, Java might throw an `OutOfMemoryException`. This is caused by the sequence graph, which contains up to $4^{\text{criton length}}$ items. This problem can be solved by using the `-Xmx` option of the Java machine, as far as your computer has enough memory.

A.2 Beispiel für Strukturbeschreibungs- und Sequenzdateien

Es folgen hier einige Beispiele für die Beschreibung von DNA-Strukturen mit Seed.

A.2.1 Dreiarmige Verzweigung

Seed Strukturbeschreibung für eine einfache dreiarmige Verzweigung wie in Abbildung 2.11 auf Seite 28, jedoch mit fünf Basen langen Einzelstrangüberhängen am 5'-Ende der Stränge:

```
# Definition der Stränge:
strand (A, 25) #rot
strand (B, N^25) #blau
strand (C, NNNNNNNNNNNNNNNNNNNNNNNNN) #grün
```

```
# Definition der Doppelstränge:
double (A, B, 15, 5, 10) #rot-cyan
double (B, C, 15, 5, 10) #cyan-grün
double (C, A, 15, 5, 10) #rot-grün
```

Dieselbe Struktur lässt sich auch mit variabler Arm- und Überhanglänge beschreiben. Dadurch können diese Größen schnell und einfach verändert werden.

```
# Definition von Variablen:
var (armLength, 10)
var (endLength, 5)

# Definition der Stränge:
strand (A, N^(2 * $armLength + $endLength))
strand (B, N^(2 * $armLength + $endLength))
strand (C, N^(2 * $armLength + $endLength))

# Definition der Doppelstränge:
double (A, B, $armLength + $endLength, $endLength, $armLength)
double (B, C, $armLength + $endLength, $endLength, $armLength)
double (C, A, $armLength + $endLength, $endLength, $armLength)
```

Die folgende Sequenzkonfiguration wurde mit Seed erstellt und ist als Sequenzdatei dargestellt. Die Critonlänge beträgt 4. Jeder der drei Doppelstränge hat einen G/C-Basenpaar-Anteil von 50%. Ihre freien Enden sind mit G/C-Paaren besetzt. Die Subsequenzen GGG, CCC, AAA und TTT treten nicht auf.

```
sequence (A, CAGACCACGAACTTCTCAGCCAATC)
sequence (B, ATGTTGATTGGCTGACTGCGATAAC)
sequence (C, TGTCGGTTATCGCAGGAAGTTCGTG)
```

A.2.2 Vierarmige Verzweigung

Durch Hinzufügen eines vierten Stranges zur dreiarmigen Konstellation entsteht eine vierarmige Verzweigung wie in Abbildung 2.12 auf Seite 29 dargestellt:

```
# Definition der Stränge:
strand (A, 25) #rot
strand (B, 25) #grün
strand (C, 25) #cyan
strand (D, 25) #blau

# Definition der Doppelstränge:
double (A, B, 15, 5, 10) #rot-grün
double (B, C, 15, 5, 10) #grün-cyan
double (C, D, 15, 5, 10) #cyan-blau
double (D, A, 15, 5, 10) #rot-blau
```

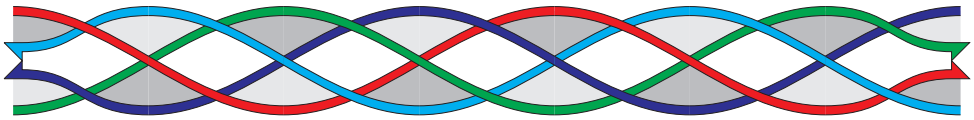


Abbildung A.1: Darstellung eines Paranemic-Crossover-Moleküls

Es folgt eine mit Seed erstellte Sequenzkonfiguration. Die Critonlänge ist 4. Alle vier Doppelstränge haben einen G/C-Anteil von 50%. Die Doppelstrangenden sind mit G/C-Paaren besetzt. Die Subsequenzen GGG und AAA (und damit auch CCC und TTT) treten nicht auf.

```
sequence (A, CTACGCAAGCGGTTACTTCGGAATC)
sequence (B, CCTATGATTCCGAAGGATGTTCACG)
sequence (C, AGTGCCGTGAACATCGCCACAATAC)
sequence (D, AGTTGGTATTGTGGCTAACCGCTTG)
```

A.2.3 Paranemic-Crossover-Moleküle

Bei Paranemic-Crossover-Molekülen (PX) [8] liegen zwei DNA-Helices nebeneinander in einer Ebene. An jeder möglichen Stelle, das heißt nach jeder halben Helixwindung, tauschen beide zwei Stränge überkreuz miteinander aus. Ein solches Molekül ist in Abbildung A.1 dargestellt.

Eine zu dieser Abbildung passende Strukturbeschreibung mit zusätzlichen fünf Basen langen Einzelstrangüberhängen am Anfang der Stränge könnte wie folgt lauten:

```
# Definition der Stränge:
strand (A, 33) #rot
strand (B, 33) #cyan
strand (C, 33) #blau
strand (D, 33) #grün

# Definition der Doppelstränge:
double (A, B, 5, 27, 6)
double (D, C, 5, 27, 6)
# Austausch A/D
double (D, B, 11, 22, 5)
double (A, C, 11, 22, 5)
# Austausch B/C
double (D, C, 16, 16, 6)
double (A, B, 16, 16, 6)
# Austausch D/A
double (A, C, 22, 11, 5)
double (D, B, 22, 11, 5)
# Austausch C/B
double (A, B, 27, 5, 6)
double (D, C, 27, 5, 6)
```

Eine von Seed ermittelte Sequenzkonfiguration mit Critonlänge 4 ist:

```
sequence (A, GCTTATCTGACCTCACTCGCCACAATGCCCCTT)
sequence (B, TAAGCAAGGGGACGGTTGGCGATTCCTGTCAGA)
sequence (C, TGGGTGCGTTTCATTGATGTTCGTGAGACTAAC)
sequence (D, ACCCAGTTAGTAGGAAGAACATACCGTAAACGC)
```

A.2.4 Rhombus-Gitter

Darstellungen des Rhombus-Grundelementes und des Rhombus-Gitters aus der Arbeit [35] finden sich in den Abbildungen 2.13 und 2.14 auf Seite 31 ff. Die entsprechende Seed-Strukturbeschreibung sieht wie folgt aus:

```
# Definition der Stränge:
strand (S1, 63)  #violet
strand (S2, 63)  #cyan
strand (S3, 63)  #orange
strand (S4, 63)  #grün
strand (S5, 100) #rot
strand (S6, 100) #blau
strand (S7, 26)  #gelb
strand (S8, 26)  #magenta

# Definition der Doppelstränge des Grundelementes:
# Kante A nach A':
double (S1, S5, 5, 92, 8)
double (S1, S6, 13, 50, 42)
double (S1, S8, 55, 5, 8)
# Kante B nach B':
double (S2, S7, 0, 13, 8)
double (S2, S5, 8, 8, 42)
double (S2, S6, 50, 0, 8)
# Kante C nach C':
double (S3, S6, 5, 92, 8)
double (S3, S5, 13, 50, 42)
double (S3, S7, 55, 5, 8)
# Kante D nach D':
double (S4, S8, 0, 13, 8)
double (S4, S6, 8, 8, 42)
double (S4, S5, 50, 0, 8)

# Verknüpfung benachbarter Elemente:
double (S1, S8, 0, 0, 5)      # A/A'
double (S2, S7, 58, 21, 5)    # B/B'
double (S3, S7, 0, 0, 5)      # C/C'
double (S4, S8, 58, 21, 5)    # D/D'
```

Die originale Sequenzkonfiguration aus der Arbeit in Form einer Sequenzdatei lautet:

```
sequence (S1, GTATGCTGATAGGACAATGAGTAGCTATTGGTGATCAACGTTAAGATACCAGTG \
              GACGAATCG) #  (63)
sequence (S2, CAGTATGGACGTAGATACTGTGCTAACGATATTCGAACTAGCGTCATCGGACGA \
              TCAGAGACG) # (63)
sequence (S3, CATTGGTAGTGCCTGTAATAATGTTGACTGCGGTTACCGTACTAATTGCTGTAC \
              CTGAGTGAG) # (63)
sequence (S4, TGACAGCCTGTCGAGTAGATCGTATGAATAGATGGCATCGCTGTAAATCCTGTG \
              TCACCTCAC) # (63)
sequence (S5, GTGACACACCGATGACGCTAGTTCGAATATCGTTAGCACAGTATCTACGTGGTA \
              CAGCAATTAGTACGGTAACCGCAGTCAACATTATTACACCTATCAG) #  (100)
sequence (S6, CTGATCGTGGATTTACAGCGATGCCATCTATTCATACGATCTACTCGACACCAC \
              TGGTATCTTAACGTTGATCACCAATAGCTACTCATTGTGGCACTAC) #  (100)
sequence (S7, CAATGCTCACTCACCATACTGCGTCT) # (26)
sequence (S8, CATACCGATTCGTGGCTGTCAGTGAG) # (26)
```

Diese Sequenzen weisen jedoch sehr viele bis zu 6 Basenpaare lange Fehlpaarungen auf, obwohl die Fehlerlänge bei dieser Strukturgröße auf 4 (Critonlänge = 5) beschränkt sein könnte.

A.2.5 4X4-Gitter

Darstellungen des 4X4-Elements und des 4X4-Gitters aus der Arbeit [18] finden sich in den Abbildungen 2.18 und 2.19 auf Seite 34 ff. Die entsprechende Seed-Strukturbeschreibung kann lauten:

```
# Definition der Stränge
sequence ("Tloop", TTTT)   # Sequenz der T-Schlaufen
strand (RING, (N^16) $Tloop (N^21) $Tloop (N^21) \
              $Tloop (N^21) $Tloop (N^5)) #ocker
strand (ENO, 42) #magenta
strand (ESO, 47) #orange
strand (ESW, 42) #grün
strand (ENW, 37) #blau
strand (VN, 26)  #violet
strand (VO, 36)  #rot
strand (VS, 36)  #gelb
strand (VW, 26)  #cyan

# Definition der Doppelstränge des Grundelementes:
double (RING, ENO, 0, 8, 6)
double (RING, ENW, 6, 19, 10)
double (RING, ENW, 20, 8, 11)
double (RING, ESW, 31, 24, 10)
double (RING, ESW, 45, 13, 11)
double (RING, ESO, 56, 24, 10)
```

```
double (RING, ESO, 70, 13, 11)
double (RING, ENO, 81, 19, 10)
double (RING, ENO, 95, 14, 5)

double (VN, ENW, 5, 29, 8)
double (VN, ENO, 13, 0, 8)
double (VO, ENO, 5, 29, 13)
double (VO, ESO, 18, 0, 13)
double (VS, ESO, 5, 34, 13)
double (VS, ESW, 18, 0, 13)
double (VW, ESW, 5, 34, 8)
double (VW, ENW, 13, 0, 8)

# Verknüpfung benachbarter Elemente:
double (VN, VS, 0, 0, 5)
double (VN, VS, 21, 31, 5)
double (VO, VW, 0, 0, 5)
double (VO, VW, 31, 21, 5)

# Maskierung der T-Schlaufen:
mask (Tloop, 1, 2)
```

Die T-Schlaufen müssen teilweise maskiert werden, da bei einer Critonlänge von 5 nicht genügend unterschiedliche Subsequenzen für die Umgebung der Schlaufen zur Verfügung stehen.

Die originale Sequenzkonfiguration lautet:

```
sequence (RING, CAGGCACCATCGTAGGTTTTCGTTCCGATCACCAACGGAGTTTTTTCTGCCG \
                TACACCAGTGAAGTTTTTCGATCCTAGCACCTCTGGAGTTTTTCTTGC)
sequence (ENO, ATGCAACCTGCCTGGCAAGACTCCAGAGGACTACTCATCCGT)
sequence (ESO, TCCGACTGAGCCCTGCTAGGATCGACTTCACTGGACCGTTCTACCGA)
sequence (ESW, ACCGGAGGCTTCCTGTACGGCAGAACTCCGTTGGACGAACAG)
sequence (ENW, ATAGCGCCTGATCGGAACGCCTACGATGGACACGCCG)
sequence (VN, GCGAGCGGCGTGTGGTTGCATCATGC)
sequence (VO, CTCTCACGGATGAGTAGTGGGCTCAGTCGGAGTCAG)
sequence (VS, CTCGCTCGGTAGAACGGTGGAAGCCTCCGGTGCATG)
sequence (VW, GAGAGCTGTTCGTGGCGCTATCTGAC)
```

Die originalen Sequenzen enthalten viele zum Teil sogar sehr lange Fehlpaarungen. Trotzdem war das Experiment erfolgreich.

A.2.6 Tetraeder

Es folgt die Seed-Strukturbeschreibung des Tetraeders aus Abbildung 2.20 auf Seite 36, welches in der Arbeit [4] vorgestellt wurde.

```
# Definition der Stränge:
strand (S1, N^11 A N^20 A N^20 A N^9) #cyan
strand (S2, N^11 A N^20 A N^20 A N^9) #rot
strand (S3, N^9 A N^20 A N^20 A N^11) #grün
strand (S4, N^9 A N^20 A N^20 A N^11) #gelb

# Definition der Doppelstränge:
# Kante A (im Paper):
double (S4, S1, 10, 0, 11)
double (S4, S1, 21, 54, 9)
# Kante B (im Paper):
double (S3, S4, 31, 31, 20)
# Kante C (im Paper):
double (S2, S4, 33, 0, 9)
double (S2, S4, 42, 52, 11)
# Kante D (im Paper):
double (S3, S2, 10, 0, 11)
double (S3, S2, 21, 54, 9)
# Kante E (im Paper):
double (S1, S2, 12, 12, 20)
# Kante F (im Paper):
double (S1, S3, 33, 0, 9)
double (S1, S3, 42, 52, 11)
```

Es folgt die originale Sequenzkonfiguration als Seed-Sequenzdatei:

```
sequence (S1, AGGCAGTTGAGACGAACATTCCTAAGTCTGAAATTTATCACCCGCCATAGTAGA \
              CGTATCACC)
sequence (S2, CTTGCTACACGATTCAGACTTAGGAATGTTCGACATGCGAGGGTCCAATACCGA \
              CGATTACAG)
sequence (S3, GGTGATAAAACGTGTAGCAAGCTGTAATCGACGGGAAGAGCATGCCCATCCACT \
              ACTATGGCG)
sequence (S4, CCTCGCATGACTCAACTGCCTGGTGATACGAGGATGGGCATGCTCTTCCCGACG \
              GTATTGGAC)
```

Diese Sequenzkonfiguration enthält 10 Fehlpaarungen mit Längen von 5 bis 7 Basenpaaren, obwohl die Critonlänge für diese Struktur bei 5 liegen könnte. Trotzdem war das Experiment erfolgreich.