Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays

von Manfred Nölte

Dissertation

zur Erlangung des Grades eines Doktors der
Naturwissenschaften
- Dr. rer. nat. -

Vorgelegt im Fachbereich 3 (Mathematik und Informatik) der

im Mai 2002

Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays

Inhaltsverzeichnis

Vorwort

Am Ende dieser Arbeit können in einer „Liste der verwendeten Symbole, Bezeichner und Abkürzungen“ deren Bedeutungen nachgeschlagen werden. In dieser Arbeit wurde eine konsistente Nomenklatur angestrebt.

Die mit „→“ gekennzeichneten Fachbegriffe können in dem Glossar nachgeschlagen werden, da mit Lesern gerechnet wird, die sich überwiegend in nur einem Fachgebiet – Informatik oder Biologie – dieser interdisziplinären Arbeit auskennen. Darunter gibt es sicher einige Informatiker, die nicht Wissen, was ein →Exon ist oder auch Biologen, die nichts mit einem →Gradientenabstiegs-Algorithmus anzufangen wissen. Ferner werden in dem Glossar weniger gebräuchliche aber für diese Arbeit zentrale Begriffe wie →Sequenzklasse und →Redundanz-Niveau kurz beschrieben.

Danksagung

Meinem Doktorvater Professor Manfred Bernd Wischnewsky danke ich für die ausgezeichnete Unterstützung und Betreuung. Das gute Klima in seiner Arbeitsgruppe ermöglicht es seinen Doktoranden konzentriert und erfolgreich zu arbeiten.

Viele konstruktive Ideen erhielt ich auch von meinen Kollegen Dr. Thomas Waschulzik, Dipl. Inform. Gerald Volkmann, Dr. Regina Rojek, Dr. Holger Dürer und Dr. Jun Zhao.

Den Professoren und Kollegen im Forschungs- und Entwicklungs-Verbund Gensensorik (jetzt umbenannt zu CAG – Centrum für angewandte Gensensorik) danke ich für ein kreatives interdisziplinäres Umfeld, in dem durch einen regen Informationsaustausch, ein sehr interessantes Forschungsgebiet an der Universität Bremen aufgebaut wurde. An erster Stelle ist hier Professor Dietmar Blohm zu nennen, ohne dessen herausragendes Engagement dieser Forschungs-Verbund nicht zustande gekommen wäre und der sich trotz der Anzahl von elf Arbeitsgruppen und entsprechend zahlreichen Doktoranden bei jeder Veröffentlichung die Zeit nahm, konstruktive Kritik zu üben.

Meiner Familie, ganz besonders meinem Bruder Harald, Rachel Ellis und den Teilnehmern meiner Fußball-Gruppe, insbesondere Professor Horst Herrlich, und der Universität Bremen Betriebssportgruppe danke ich für die Unterstützung im privaten Umfeld und für den sportlichen Ausgleich in einer freundschaftlichen Atmosphäre.

Zusammenfassung

Die „Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays“ ist eine Methodik zum Experiment-Design für die, in den Bereich der Nukleinsäure-Analytik einzuordnende, Technologie der DNA-Mikroarrays. Damit ist diese Arbeit ein Beitrag zur Bioinformatik und macht selbst intensiv von den Errungenschaften der Bioinformatik und →Molekularbiologie gebrauch.

DNA-Mikroarrays sind eine sich rapide entwickelnde Analysetechnologie im Bereich der modernen Biotechnologie. Mit DNA-Mikroarrays ist man in der Lage, massiv parallel, einfach, schnell und kostengünstig in Proben genetische Information (Nukleinsäuresequenzen) hochspezifisch und hochsensitiv nachzuweisen und damit die biologische Vielfalt gezielt zu untersuchen. Auf diesem Weg können Krankheitserreger, wie z.B. Viren und Bakterien, sowohl in Lebensmitteln als auch im Körpergewebe und in Körperflüssigkeiten in geringen Mengen nachgewiesen und exakt bestimmt werden.

Bei dem Einsatz von DNA-Mikroarrays finden bis zu mehrere zehntausend Reaktionen miniaturisiert statt, denn diese Anzahl von Nukleinsäure-Molekülen ist zuvor auf ein DNA-Mikroarray aufgebracht worden. Die Arbeit mit DNA-Mikroarrays, d.h. deren Entwicklung und Einsatz, ist geprägt durch eine Vielzahl von Parametern und Kriterien und es müssen große Datenmengen bearbeitet werden. Eine softwarebasierte Unterstützung zu entwickeln, drängt sich geradezu auf.

Bei der Entwicklung von DNA-Mikroarrays stellt sich die Frage: „Welche der soeben erwähnten Nukleinsäure-Moleküle müssen, bei einer vorgegebenen biologischen Aufgabenstellung (z.B. der Nachweis eines Virus in einer Blutprobe) verwendet werden, um eine sensitive und spezifische Analytik zu entwickeln?“. Diese Arbeit beschäftigt sich hauptsächlich mit dieser Frage, genauer mit dem Design von Oligonukleotid-Bibliotheken für DNA-Mikroarrays; denn kurze Stücke von Nukleinsäure-Molekülen (z.B. ACGTGGCT-AGCTAGCTGCTAGCT; →Sequenz von →Nukleotiden bzw. Basen) heißen „Oligonukleotide“. Die Sequenz dieser →Oligonukleotide und zahlreiche Eigenschaften müssen dabei optimiert und aufeinander abgestimmt werden.

Die Oligonukleotide werden so bestimmt, dass sie möglichst die Anwesenheit von →Ziel-Sequenzen (z.B. die →Nukleotid-Sequenz eines Virus) signalisieren, d.h. richtig-positive →Treffer erzeugen. Werden ebenfalls in der Probe vorhandene nicht nachzuweisenden Nukleinsäuren detektiert, dann sind das falsch-positive Treffer. Die Eigenschaften der Oligonukleotide lassen sich zunächst in zwei Klassen aufteilen. Zum einen bestimmen die erwähnten Treffer-Mengen die →Sensitivität und die →Spezifität eines Oligonukleotids oder einer ganzen Oligonukleotid-Bibliothek, zum anderen wird die Intensität des Signals (die Hybridisierungseffizienz) durch zahlreiche weitere Eigenschaften des Oligonukleotids selbst wie auch der nachzuweisenden Nukleinsäure bestimmt. Diese zwei Klassen von Eigenschaften dürfen jedoch nicht getrennt betrachtet werden. Die in dieser Arbeit entwickelten Bewertungsfunktionen verdeutlichen eine enge Verzahnung dieser Eigenschaften.

Für z.B. hochvariable Virengenome werden größere Anzahlen von Oligonukleotiden benötigt, um alle in der Viren-Population vorkommenden Varianten zu erfassen. Die effiziente Bestimmung dieser Oligonukleotide entspricht der Lösung eines „Set Cover“-Problems. In dieser Arbeit wurden drei Ansätze für kombinatorische Optimierung entwickelt, die das „Set Cover“-Problem heuristisch lösen und dabei die zuvor genannten Eigenschaften der Oligonukleotide berücksichtigen. Diese sind ein *Greedy Search* Ansatz, eine Kombination von →Gradientenabstieg und Kompetition und ein Ansatz über Genetische Algorithmen.

Die Ergebnisse dieser Arbeit sind in **Kapitel 4** eine umfangreiche Sammlung von Eigenschaften aller beim Design von Oligonukleotid-Bibliotheken relevanten Objekte, deren Wandlung

durch Bewertungsfunktionen in Zahlenwerte (damit werden sie für eine algorithmische Bearbeitung greifbar) und eine formale Aufgabenspezifikation. Weiterhin zählen in **Kapitel 5** die bereits erwähnten drei Ansätze für Optimierungs-Algorithmen, in **Kapitel 6** das internetbasierte Programm **optiNA** „**opti**mal **N**ucleic **A**cids“ und in **Kapitel 7** zwei innerhalb des FuE-Verbunds Gensensorik bearbeitete Anwendungen: die „Identifikation von Hepatitis C-Viren“ und die „Organismen-Identifikation: Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“ zu den Ergebnissen dieser Arbeit.

Mit dem in dieser Arbeit beschriebenen System können optimierte Oligonukleotid-Bibliotheken in kürzerer Zeit erstellt werden. *„A key element in microarray experiments is chip [microarray] design. This is the aspect that's often forgotten by users of commercial devices and commercial chips, because one benefit of those systems is that chip design has been done for you, by an expert before you ever think about doing an experiment. Chip design is a process that can take months“* [34]. Für Molekularbiologen entfällt damit die langwierige, manuelle und fehleranfällige Bearbeitung großer Mengen an →Sequenzen und deren Eigenschaften.

1. Einführung

Die Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays ist eine Bioinformatik-Methodik zum Experiment-Design für die, in den Bereich der Nukleinsäure-Analytik einzuordnende, Technologie der DNA-Mikroarrays. Die folgenden Abschnitte dieses Kapitels geben eine kurze Einführung in die Gebiete der Bioinformatik und der Nukleinsäure- bzw. DNA-Analytik und stellen deren Zusammenhang mit dieser Arbeit dar.

1.1. Bioinformatik

Wissenschaftlich ist diese Arbeit in die Bioinformatik (engl. Computational [Molecular] Biology oder Bioinformatics) [5], [34], [44], [57], [84], [87], [90], [103] einzuordnen. Die Bioinformatik ist die Disziplin, die die Aufgabe hat, große Mengen anfallender Daten der →Molekularbiologie systematisch zu verarbeiten. Dazu gehören die Entwicklung und Pflege von Datenbanken (Sequenzdatenbanken für →Proteine und Nukleinsäuren sowie Datenbanken für Experiment-Ergebnisse, Pathways, Funktionsvorhersagen und Makromolekül-Strukturen) und die Entwicklung neuer Algorithmen, die zum Beispiel für die Verarbeitung von DNA-Sequenzen geeignet sind. Die Bioinformatik stellt Programme für Molekularbiologen zur Verfügung, die deren tägliche Arbeit vereinfachen oder überhaupt erst ermöglichen. Ein großer Teil der Arbeit der Molekularbiologen besteht in der Durchführung von Labor-Experimenten, trotz eines zunehmenden Umgangs mit Computern, Datenbanken und Internet-Ressourcen; und dabei wird es sicher bleiben, denn „Bioinformatik ersetzt nicht Experimente, sondern hilft beim Design intelligenter Experimente“ [90]. Das Thema dieser Arbeit, nämlich die Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays, ist somit als Teil des Experiment-Designs eines →Hybridisierungs-Experiments in ein Anwendungsgebiet der Bioinformatik einzugliedern.

Ein weiteres Anwendungsfeld der Bioinformatik ist die Auswertung komplexer Resultate oder großer Datenmengen aus Experimenten. Gerade DNA-Mikroarrays liefern mit nur einem Hybridisierungs-Experiment eine sehr große Menge an Daten. Auf einer niedrigeren Ebene nennt [103] „*A classification of tasks in bioinformatics*“ als Ergebnis einer Fragebogenaktion folgende Aufgaben für die Bioinformatik: „*sequence similarity searching*[1]*, functional motif searching* (→Functional Genomics)*, sequence retrieval*[1]*, multiple sequence →alignment, →restriction mapping, secondary and tertiary structure prediction*[1]*, other DNA analysis including translation, primer design*[1]*, literature searching, phylogenetic analysis* (→Phylogenie)*, sequence assembly, location of expression*“.

1.2. Sequenzdatenbanken und DNA-Analytik

Eine wichtige Voraussetzung für die DNA-Analytik mit DNA-Mikroarrays ist die Kenntnis und die Verfügbarkeit der →Sequenzen der zu untersuchenden Organismen. In der Bioinformatik werden große Anstrengungen unternommen, Datenbanken mit Sequenz-Informationen aufzubauen. In diesen Sequenzdatenbanken werden die Sequenzen selbst, als Ergebnisse der Sequenzierungsprojekte (→Sequenzierung), zusammen mit Annotationen gespeichert.

Neben den großen und bekannten Sequenzierungsprojekten wie das des HUGO[2] „The Human Genome Organisation“ [47], [109] oder GABI[3] „Genomanalyse im Biologischen System Pflanze“ gibt es Sequenzierungsprojekte insbesondere zu Modellorganismen, wie der Hefe (c. serviciae), der Ackerschmalwand (Arabidopsis thaliana), einer einfachen Pflanze, der Tau-

[1] wird bei dem Design von Oligonukleotid-Bibliotheken benötigt bzw. ist ein verwandtes Problem.
[2] HUGO: Abk. für „The Human Genome Organisation“ [47], [109]; http://www.gene.ucl.ac.uk/hugo/
[3] GABI: Abk. für „Genomanalyse im Biologischen System Pflanze“; http://www.gabi.de/

fliege (Drosophila melanogaster) als Insekt, dem Bakterium Helicobakter pylori als einem von zahlreichen mikrobiellen Organismen, dem Zebrafisch, der Maus und zahlreichen Viren. Wegen dieser Anzahl an Sequenzierungsprojekten und weiterentwickelten Technologien bis hin zur *high throughput* Sequenzierung haben die Sequenzdatenbanken ein exponentielles Wachstum. Wie oben bereits erwähnt, sind diese Informationen Voraussetzung für die Organismen-Identifikation. Dieses Anwendungsgebiet der DNA-Analytik, das mit Hilfe der DNA und mit Hilfe der Sequenzinformation über DNA Aussagen über das Vorkommen eines Organismus oder eines Gewebetyps in einer Probe trifft, soll ein einfaches Beispiel verdeutlichen:

Die Sequenzinformation der DNA mehrerer Organismen bzw. Spezies (z.B. Bakterien oder Viren) werden nach der Sequenzierung in den Sequenzdatenbanken abgelegt. Ein DNA-analytisches Verfahren, welches diese Bakterien erkennen und unterscheiden soll, wird mit Hilfe dieser Sequenzinformation entwickelt. Ergebnisse aus der Anwendung dieses DNA-analytischen Verfahrens werden wiederum mit den Sequenzdatenbanken abgeglichen. Auf diese Weise kann man auf die Anwesenheit eines Bakteriums in einer Probe, quasi durch einen Blick in dessen Erbgut, schließen. Das heißt, es wird hier tatsächlich die genetische Information (der Genotyp) und nicht der Phänotyp (das Erscheinungsbild) identifiziert, wie dies bei anderen Nachweisverfahren der Fall ist.

Mittlerweile existieren umfangreiche Sequenzdatenbanken und sie enthalten mehrere vollständig sequenzierte →Genome. Viele Verfahren der Biotechnologie werden bereits industriell und im "high-throughput"-Verfahren durchgeführt. „Molekulare Techniken tragen in vielen Bereichen der medizinischen Diagnostik zu einer verbesserten Qualität der Tests und ihrer Ergebnisse bei. In der modernen Virus- und Mykobakteriendiagnostik sind diese Methoden nicht mehr wegzudenken; in der Erkennung von Krebs werden die ersten Ansätze bald das Forschungslabor verlassen.“ [72] Im Zentrum dieser Arbeit stehen die DNA-Mikroarrays, sie parallelisieren bis zu 100.000-fach, was sonst aufwändig mit anderen Verfahren der Molekularen Diagnostik wie →PCR, →Gelelektrophorese oder Southern-Blots durchgeführt werden müsste.

Die DNA-Analytik ist in mehrfacher Hinsicht eine schnelle Diagnostik. Das gilt insbesondere für DNA-Mikroarrays. Im Gegensatz zu serologischen Tests, die oft erst 3-4 Monaten nach einer Infektion, durch die Bildung von Anti-Körpern im Wirtsorganismus, mit vollständiger Sensitivität einsetzbar sind, ermöglicht die DNA-Analytik eine **Früherkennung**. „Für die Amplifikation und den Nachweis bakterieller und viraler →DNA und →RNA gibt es heute kommerziell verfügbare Testkits [...] und Geräte, die es erlauben, solche Tests mit großer Geschwindigkeit (Anm. d. Autors: **schneller Antwortzeit**) und **hohem Durchsatz** durchzuführen“ [72]. Aus dem folgenden Grund sind insbesondere DNA-Mikroarrays zusammen mit einer geeigneten Software zusätzlich **schnell entwickelt und auswertbar**. „Microarray experiments are amenable to computational (Anm. d. Autors: design and) analysis because of the uniform, standardized nature of (Anm. d. Autors: their setup and) their results“ [34]. Die in dieser Arbeit entwickelte Software **optiNA** wurde unter anderem mit dem Ziel entwickelt, das Design von DNA-Mikroarrays zu beschleunigen.

1.3. DNA-Mikroarrays

Im vorigen Abschnitt wurde festgestellt, das sich DNA-Mikroarrays aufgrund ihrer uniformen standardisierten Struktur für ein softwarebasiertes Design eignen. Aber was sind DNA-Mikroarrays, in welchen Kontext sind sie einzuordnen, wofür werden sie angewendet und was sind (neben der eben erwähnten Beschleunigung der Diagnostik) ihre Vorteile?

DNA-Mikroarrays sind eine sich rapide entwickelnde Analysetechnologie im Bereich der modernen Biotechnologie. Die Biotechnologie, seit der Nutzbarmachung von Bakterien und Pilzen in Produktionsprozessen wie zum Beispiel für Brot, Bier und Käse, und besonders die moderne Biotechnologie, seit 1973[4], haben bereits zahlreiche Anwendungen für Diagnose, Therapie und Produktion hervorgebracht. Ihre Anwendungsgebiete sind Umwelttechnologie, Landwirtschaft, Medizin, Lebensmittelproduktion und Gentechnologie. Die moderne Biotechnologie ist durch die →Molekularbiologie geprägt. Seit der Erfindung der →Sequenzierung (1977), der Klonierung (1972) und der Entwicklung der →Polymerasekettenreaktion PCR (1983) hat man Technologien zur Verfügung, um intensiv die Beschaffenheit und Funktion (→Functional Genomics) des Erbmaterials aller Organismen und Gewebetypen zu erforschen. „Nur sechs Jahre nach der Veröffentlichung des ersten komplett sequenzierten mikrobiellen Genoms leben wir bereits in dem, was man gemeinhin die ‚post-genomische' Phase nennt, ein Begriff, unter dem die neuen Techniken zusammengefasst werden, die unter Verwendung von Genomdaten den Zusammenhang von Sequenz, Funktion und Struktur im Regelwerk einer Zelle untersuchen." [90]

Mit DNA-Mikroarrays [97], [89] ist man in der Lage, massiv parallel, einfach, schnell und kostengünstig in Proben genetische Information (Nukleinsäuresequenzen) hochspezifisch (→Spezifität) und hochsensitiv nachzuweisen (→Sensitivität, →Nachweisgrenze) und damit die biologische Vielfalt gezielt zu untersuchen. Auf diesem Weg können Krankheitserreger wie z.B. Viren und Bakterien sowohl in Lebensmitteln als auch im Körpergewebe und in Körperflüssigkeiten in geringen Mengen nachgewiesen und exakt bestimmt werden. In der Medizin können DNA-Mikroarrays ferner zur Therapieoptimierung, z.B. bei Krebs und zur Prognose der Wirkung und der Verträglichkeit von Medikamenten, eingesetzt werden. In der Lebensmittel- und Futtermittelindustrie kann z.B. die Verwendung von gentechnisch veränderten Lebensmitteln in Fertiggerichten oder der Einsatz von unzulässigen biologischen Rohstoffen, wie z.B. Fleisch von bedrohten Tierarten, nachgewiesen werden. Im Bereich der forensischen Justiz und Strafermittlung können mit Hilfe von DNA-Mikroarrays Straftäter schneller identifiziert und somit Straftaten effizienter aufgedeckt werden. Der Vorteil der spezifischeren Identifikation wird von den Vorgängertechnologien der DNA-Analytik ohne Einschränkung übernommen, mit diesen wurden in den USA aufgrund ihrer Sicherheit und Aussagekraft bereits zum Tode verurteilte Menschen entlastet.

Die Vorteile der DNA-Analytik für die Medizin haben den Gesetzgeber in Deutschland Anfang 1999 veranlasst, für Tests auf Hepatitis C-Viren in Blutbanken DNA-analytische Verfahren vorzuschreiben. In einer Überarbeitung der Richtlinien aus dem Jahre 1996 von der Bundesärztekammer und dem Paul-Ehrlich-Institut heißt es: „Die Prüfung auf Hepatitis-C-Viren ist mit einer geeigneten Nukleinsäure-Amplifikationstechnik durchzuführen. Das Ergebnis muss negativ sein." [112] Die „Testung von Blutspenden auf Hepatitis-C-Virus mit Nukleinsäure-Nachweis-Techniken" [112] wurde im Bundesgesundheitsblatt (1998; 11, Seite 512) vorgeschrieben.

Bei der Analyse von Blutprodukten kann eine sensitive und aussagekräftige Analytik viel bewirken und das folgende Zitat verdeutlicht, dass die DNA-Analytik mehr und mehr zum Einsatz kommt:

> Durch die Einführung von empfindlichen Immuno-Assays zum Nachweis von anti-HIV, anti-HCV und HBsAg konnte dieses Risiko deutlich

[4] 1973: Herbert Boyer und Stanley Cohen klonieren ein erstes Gen. Vier Jahre später wurde von Allan Maxam, Walter Gilbert und Frederick Sanger Sequenzierungsmethoden entwickelt, um die Bausteinreihenfolge in Erbmolekülen zu bestimmen.

> verringert werden. Seit einiger Zeit (Anm. d. Autors: seit Mitte 2000) wird durch die Blutbanken zusätzlich die HCV-RNA mit äußerst sensitiven Nukleinsäuremethoden bestimmt: Das Risiko einer Infektion mit Hepatitisviren und HIV wurde auf das Niveau des Risikos eines Flugzeugabsturzes in den entwickelten Ländern gesenkt. [73]

Für jede Aufgabenstellung muss ein DNA-Mikroarray konfiguriert und optimiert werden. Einerseits, um die technische Realisierung zu vereinfachen oder zu ermöglichen und andererseits, um bei den DNA-Analysen möglichst gute Ergebnisse zu erzielen. Ferner ist für jede Aufgabenstellung eine spezifische Software für die Auswertung erforderlich, die im Rahmen des FuE-Verbunds Gensensorik (siehe Abschnitt 1.4) entwickelt wird.

Weltweit werden in Hunderten von Laboren und Firmen Systeme, Labor-Protokolle, Geschäftsmodelle und Testkits entwickelt, um das Potential der DNA-Analytik mit DNA-Mikroarrays auszuschöpfen. Es bleibt die Frage „Welche Sequenzinformationen aus den Sequenzdatenbanken müssen, bei einer vorgegebenen Organismen-Identifikation, verwendet werden, um eine sensitive und spezifische Analytik mit DNA-Mikroarrays für eine gegebene Aufgabenstellung zu entwickeln?“.

1.4. Die Fragestellung

Diese Arbeit beschäftigt sich hauptsächlich mit dieser Frage, nämlich mit der →Konfigurierung oder genauer mit dem Design von →Oligonukleotid-Bibliotheken für DNA-Mikroarrays. In [34] heißt es zu dieser Thematik: „A key element in microarray experiments is →chip design. This is the aspect that’s often forgotten by users of commercial devices and commercial chips, because one benefit of those systems is that chip design has been done for you, by an expert before you ever think about doing an experiment. Chip design is a process that can take months“. Der Abschnitt 3 „Stand der Technik“ und eine Kooperation mit dem UFT/BMG der Universität Bremen bei der manuellen Erstellung einer Oligonukleotid-Bibliothek für das Hepatitis C-Virus (HCV) bestätigen diese Thesen. Ziel dieser Arbeit ist die Beschleunigung, Qualitätssicherung und -verbesserung bei der Entwicklung von DNA-Mikroarrays durch die automatisierte und optimierte Erstellung von Oligonukleotid-Bibliotheken. Dazu wurde im Rahmen dieser Arbeit die internetbasierte Software **optiNA** entwickelt (siehe Kapitel 6).

Aber nicht nur in dem Gebiet der DNA-Analytik ist das dafür notwendige Sequenz-Design von Interesse. Ebenfalls spezielle Aufgabenstellungen des DNA-Computing (siehe Abschnitt 8.3) können davon profitieren, da in beiden Fällen Mengen von →Oligonukleotiden zusammengestellt werden und mehrere Optimierungskriterien ebenfalls im DNA-Computing anwendbar sind.

Die Konfigurierung beziehungsweise die Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays ist eine wissenschaftlich interessante Aufgabe bei der softwaretechnischen Unterstützung des Lebenszyklus von DNA-Mikroarrays. Für das zugrunde liegende Problem der kombinatorischen Optimierung kommen Verfahren wie Genetische Algorithmen [53], [74], Techniken aus der Theorie der →Neuronalen Netze [8], [18], [24], [91], [93], [116], "greedy search" [44] und Werkzeuge der Bioinformatik, wie BioPerl, EMBOSS, Fasta, mfold und das Vienna →RNA Package [110], [42] zum Einsatz.

1.5. FuE-Verbund Gensensorik

Diese Arbeit ist im Rahmen des BMBF-geförderten Forschungs- und Entwicklungsverbunds Gensensorik an der Universität Bremen entstanden. Dieser hat sich Mitte 2001 im Rahmen seiner Weiterentwicklung in CAG – Centrum für Angewandte Gensensorik umbenannt. Ziel

dieses Verbundes ist, vollintegrierte Systeme [11], [111] auf DNA-Mikroarray-Basis zu entwickeln. Im Sinne eines Messgerätes, ein solches Gerät wird Gensensor genannt, soll es schnelle und kostengünstige DNA-Analytik ermöglichen, die vor Ort durchgeführt werden kann.

Elf Arbeitsgruppen haben sich in diesem FuE-Verbund Gensensorik zusammengefunden und decken alle Fachgebiete ab, die für die Realisierung notwendig sind: Biologie, Chemie, Informatik, Mikrosystemtechnik, Robotik und Biosensorik. Von vier biologischen Arbeitsgruppen wenden sich drei der Anwendung zu (Prof. Bullerdiek, Prof. Hildebrand und Prof. Reinhold-Hurek), eine der Evaluierung dieser neuen Technologie (Dr. Amann) und eine der methodischen Untersuchung [14] und labortechnischen Realisierung (Prof. Blohm) [25], [76], [80]. Die Arbeitsgruppe der Chemie von Prof. Wöhrle forscht an der chemischen Aktivierung von Oberflächen zur kovalenten Anbindung von →Oligonukleotiden. Zwei Arbeitsgruppen der Informatik (Prof. Schlieder und Prof. Wischnewsky) bearbeiten Bioinformatik-Aufgabenstellungen. Die Mikrosystemtechnik von Prof. Binder und Prof. Benecke entwickelt eine miniaturisierte Hybridisierungskammer. In der Arbeitsgruppe von Prof. Metev wird ein Roboter für das Mikropipettiersystem entwickelt und schließlich der, wie Dr. Amann zur Phase 2 des FuE-Verbundes Gensensorik im Juni 2000 hinzugekommene, Prof. Gauglitz modifiziert die RIFS-Technologie, als Alternative zur "Fluoreszenz-Detektions-Technologie" für DNA-Mikroarrays.

Die Bioinformatik-Arbeitsgruppen entwickeln Algorithmen und Software für die Entwicklung und Anwendung von DNA-Mikroarrays [12]. In der Arbeitsgruppe von Professor Schlieder wird eine Software entwickelt, die das sogenannte Sequenzen-Retrieval unterstützt. Ausgehend von einer Fragestellung, zum Beispiel "Detektion des Virus HCV" [13], wird eine Vorauswahl von Sequenzen aus den oben erwähnten Sequenzdatenbanken ermittelt. Damit wird schließlich in der Arbeitsgruppe von Prof. Wischnewsky, in der diese Arbeit entstanden ist, die Konfigurierung von DNA-Mikroarrays mit der neuentwickelten Software **optiNA** durchgeführt [81], [83], [82], [85]. Die Auswertung und Interpretation einer mit einem DNA-Mikroarray durchgeführten Analyse wird ebenfalls softwaretechnisch unterstützt, so dass der gesamte „Lebenszyklus“ von DNA-Mikroarrays abgedeckt ist.

1.6. Zum Aufbau dieser Dissertation

Diese Arbeit ist gegliedert in sieben wesentliche Kapitel. Nach der Einführung werden in Kapitel 2 die Grundlagen der DNA-Analytik mit DNA-Mikroarrays, für die aus einem interdisziplinären Kreis erwarteten Leser, vorgestellt. Nach einer Darstellung des Stands der Technik (Kapitel 3) werden in den Kapiteln 4 und 5 die zwei zentralen Ergebnisse dieser Arbeit vorgestellt, die im folgenden Absatz kurz skizziert werden. In den Kapiteln 6 und 7 werden das internetbasierte Optimierungsprogramm **optiNA** „optimal Nucleic Acids“ und die mit dem FuE-Verbund Gensensorik be- und erarbeiteten Anwendungen und Ergebnisse vorgestellt. In Kapitel 8 werden in einer Diskussion die Ergebnisse und deren Einfluss auf die DNA-Analytik mit DNA-Mikroarrays bewertet und ein Ausblick gegeben.

Die zwei zentralen Ergebnisse dieser Arbeit in den Kapiteln 4 und 5 sind die Problemanalyse und Aufgabenspezifikation in Kapitel 4, in dem zusätzlich die in den Grundlagen und in der Aufgabenspezifikation angegebenen Problemtypen (z.B. das „Set Cover“-Problem mit Nebenbedingungen) und Objekteigenschaften (z.B. die Stabilität von Sekundärstrukturen) durch Bewertungsfunktionen in Zahlenwerte umgewandelt werden. Mit dem Ziel diese Zahlenwerte zu maximieren oder zu minimieren werden diese damit für einen algorithmischen Ansatz zugänglich gemacht. In Kapitel 5 werden drei Ansätze für Optimierungs-Algorithmen vorgestellt, mit denen optimierte Oligonukleotid-Bibliotheken für DNA-Mikro-

arrays erstellt werden können. Diese sind ein *Greedy Search* Ansatz, eine Kombination von →Gradientenabstieg und Kompetition und ein Ansatz über Genetische Algorithmen.

2. Grundlagen der DNA-Analytik mit DNA-Mikroarrays

Zusammenfassung: In der DNA-Analytik mit DNA-Mikroarrays werden Bestandteile der Erbinformation (DNA-Sequenzen) verwendet, um Organismen oder Gene zu identifizieren. Nach einer Beschreibung der zugrundeliegenden molekularen Prozesse und der Erzeugung von detektierbaren Signalen wird auf die →Thermodynamik und damit auf die wichtigsten Eigenschaften der DNA-Sequenzen im Zusammenhang mit DNA-Analytik eingegangen, nämlich die Schmelztemperatur und die Sekundärstrukturen. Es wird beschrieben, welchen Einfluss diese Eigenschaften auf die DNA-Analytik mit DNA-Mikroarrays haben und wie diese bei der Optimierung von Oligonukleotid-Bibliotheken berücksichtigt werden müssen.

In den letzten drei Abschnitten wird der Lebenszyklus von DNA-Mikroarrays von „Sequenzretrieval und Motivbestimmung“ über „Optimierung von Oligonukleotid-Bibliotheken“ bis zur „Auswertung einschließlich Interpretation“ beschrieben. Dabei werden besonders detailliert drei algorithmische Ansätze zur Optimierung von Oligonukleotid-Bibliotheken vorgestellt, die mittels iterativer Approximation oder heuristischer Verfahren ein zugrundeliegendes Problem kombinatorischer Optimierung lösen. Die Notwendigkeit ein sogenanntes „Set Cover“-Problem lösen zu müssen, welches eine kombinatorische Komplexität besitzt, wird an einem einfachen Beispiel erläutert. Die drei erwähnten algorithmische Ansätze zur Optimierung von Oligonukleotid-Bibliotheken sind „Greedy Set Covering“, ein Gradientenabstiegs-Verfahren, das in Abschnitt 5.2 mit einer kompetitiven Komponente kombiniert wird und ein Ansatz über Genetische Algorithmen.

In der DNA-Analytik mit DNA-Mikroarrays werden Bestandteile der Erbinformation verwendet, um Organismen oder Gene zu identifizieren. Ein Stück des Einzelstrangs der DNA des zu erkennenden Organismus wird durch ein kurzes passendes Gegenstück, welches auf dem DNA-Mikroarray immobilisiert ist, eingefangen. Die einzufangende, zunächst unbekannte, DNA in der zu untersuchenden Probe nennt man Ziel-DNA und das Gegenstück auf dem DNA-Mikroarray wird Fänger-DNA, Sonde (engl. *[capture] →probe*) oder manchmal auch →Oligonukleotid genannt, weil es aus einer →Sequenz von relativ wenigen (*griechisch „oligo“: wenig*) Basen besteht. „Eingefangen“ wird die Ziel-DNA mittels der Hybridisierung, wie der Übergang zweier Einzelstränge zu einem Doppelstrang bezeichnet wird.

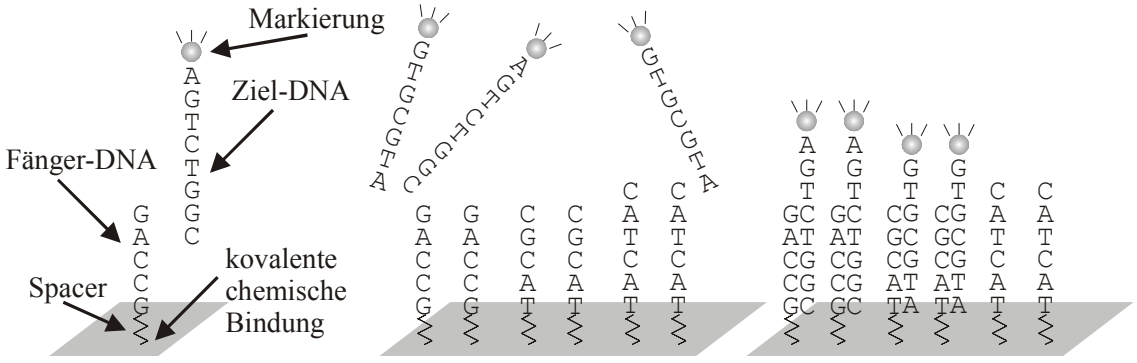


Abbildung 1.6-1: Das „Einfangen“ von Ziel-DNA auf einem DNA-Mikroarray

Die Abbildung 1.6-1/links stellt zunächst die kovalente Bindung und die Struktur von Fänger- und Ziel-DNA dar. Ein sogenanntes Spacer- oder Linker-Molekül [89] sorgt für einen Abstand der untersten Base der Fänger-DNA zur Oberfläche, die aus Glas, Silizium, Gold oder aus einer Nylon-Membran besteht. Die Abbildung 1.6-1/rechts stellt das Prinzip Ziel-

DNA-Einzelstränge mittels Hybridisierung einzufangen schematisch dar. Eine zu analysierende Probe enthält die Ziel-DNA, welche Fluoreszenz- oder radioaktive Marker enthält. Unter bestimmten Bedingungen, die im Abschnitt 2.1 detailliert beschrieben werden, kommt es zwischen der Fänger-DNA und der Ziel-DNA zur Hybridisierung.

Nach erfolgter Hybridisierung und dem Abwaschen von ungebundenem Material wird die Anwesenheit der Ziel-DNA über die Marker als sogenanntes Hybridisierungssignal detektiert [89], [97]. Dazu werden in Abhängigkeit von dem Typ der Markierung verschiedene bildgebende Verfahren eingesetzt. Die Detektion radioaktiver Strahlung, der Einsatz eines konfokalen Laser-Scanners oder einer einfachen CCD-Kamera gehören dazu. In der Abbildung 1.6-2 sind in 15 Zeilen und 18 Spalten 270 dieser Hybridisierungssignale dargestellt. Es handelt sich dabei um 270 Positionen oder Kavitäten (engl. *spots*), an denen winzige Mengen einer Lösung definierter DNA-Sequenzen zum Beispiel durch einen Mikropipettier-Roboter abgelegt wurden. Ein heller Punkt deutet auf die Anwesenheit von vielen Ziel-DNA-Molekülen und damit auf ein positives Hybridisierungssignal hin. In Abschnitt 2.6 wird kurz beschrieben, wie diese Hybridisierungssignale quantifiziert und für die Auswertung der DNA-Analyse verwendet werden.

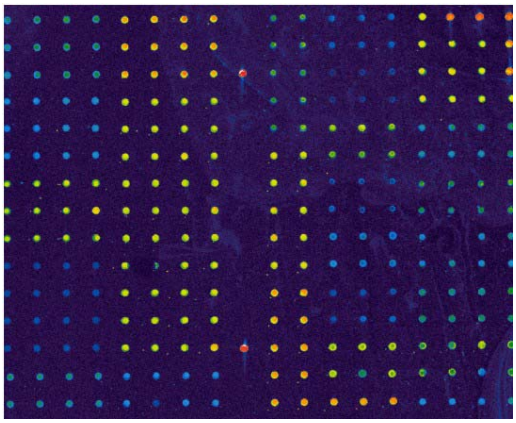


Abbildung 1.6-2: Hybridisierungssignale eines DNA-Mikroarrays

Ein Spot - z.B. Position (14, 10) - auf dem DNA-Mikroarray ist einem bestimmten →Oligonukleotid - z.B. 5´-ATCCGAAGCT-3´ - zugeordnet, sodass über die Position auf die eingefangene Ziel-DNA geschlossen werden kann - die Ziel-DNA enthält mit hoher Wahrscheinlichkeit die komplementäre Sequenz 3´-TAGGCTTCGA-5´. Von dieser Sequenz wiederum weiß man, dass sie einem Virus, einem Bakterium oder einem beliebigen Organismus zugeordnet ist, auf dessen Anwesenheit in der Probe somit geschlossen werden kann. Zur Absicherung der Ergebnisse werden in der Praxis sehr viele solcher Hybridisierungssignale erzeugt und ausgewertet.

Obwohl in der DNA-Analytik die DNA-Sequenz überwiegend nur als Zeichenkette verwendet wird, ist es sehr sinnvoll, möglichst viel über die Struktur und Funktion (→Functional Genomics) der DNA zu wissen. Denn man nimmt an, dass die Funktionen gewisser Sequenzabschnitte hauptsächlich in deren Sekundärstruktur liegt. Diese Sekundärstrukturen enthalten sogenannte Loops und Bulges, deren Basensequenzen (weitestgehend) unabhängig von der Sekundärstruktur sind. Daher sind die spezifischeren Sequenzen auf diesen Loops und Bulges zu erwarten. Die übrigen Sequenzabschnitte könnten bei einer →Mutation die Sekundärstruktur zerstören. Paarweise Mutationen, die zugleich auf beiden Seiten einer Helix auftreten, sind selten und werden zur sicheren Bestimmung von Sekundärstrukturen auf der Basis phylogenetischer Daten (→Phylogenie) herangezogen. Weiterhin beeinflussen die Sekundärstrukturen sehr stark, ob es zu einer Hybridisierung kommt. Dieser Sachverhalt wird in den Abschnitten 2.3 und 4.2.3 detailliert behandelt.

2.1. Hybridisierung und Denaturierung

Die Hybridisierung ist das Zusammengehen zweier DNA-Einzelstränge zu einer Doppelhelix (siehe Abbildung 2.1-1) und die Denaturierung ist der umgekehrte Prozess. Beide können inter- oder intramolekular ablaufen. Das intramolekulare Bilden von Helices wird in dem Abschnitt 2.3 zu den Sekundärstrukturen betrachtet. Für die DNA-Analytik mit DNA-Mikroarrays ist die Hybridisierung der eigentliche thermodynamische und hochspezifische Prozess, der durch geeignete Technologie und ein optimiertes →Hybridisierungsprotokoll sequenzspezifisch herbeigeführt werden soll.

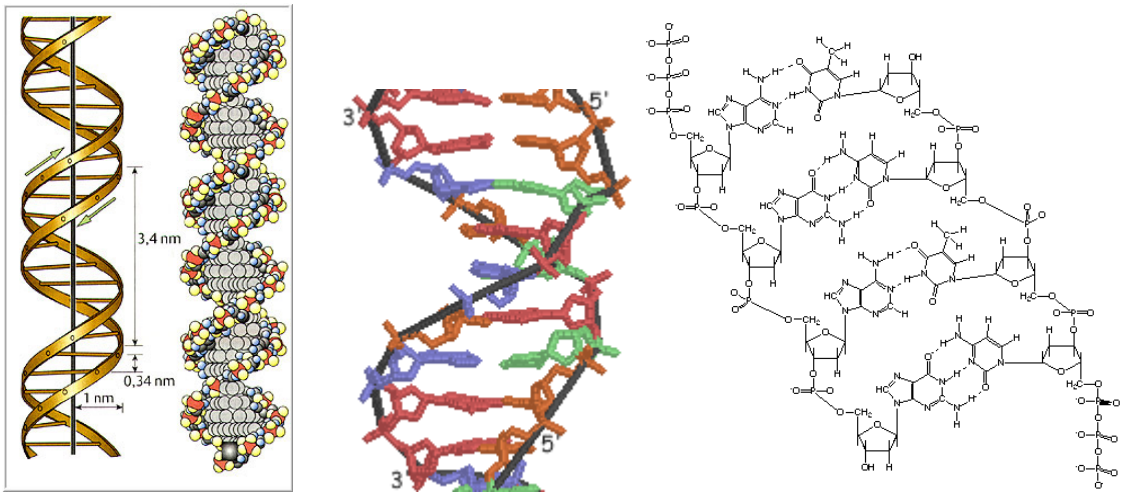


Abbildung 2.1-1: DNA-Doppelstrang (von links nach rechts) **als Helixstruktur, Kalottenmodell[5], dreidimensionale Struktur der →Nukleotide und Konstitutionsformel**

> Unter physiologischen Bedingungen ist die sogenannte B-Form die stabilste Konformation der DNA. Eine dreidimensionale Darstellung dieser Konformation ist in Abbildung 2.1-1 ersichtlich. In dem Kalottenmodell sind die Atome des Pentose-Phosphat-Rückgrats farbig hervorgehoben: die Phosphoratome sind rot, Sauerstoff ist gelb, Wasserstoff ist blau und der Kohlenstoff ist dunkelgrau. Die Atome der Purin und Pyrimidin Basen Guanin, Cytosin, Thymin und Adenin, die über Watson-Crick-Basenpaarung (Wasserstoffbrücken) zur Stabilisierung der Doppelhelix beitragen, erscheinen in dieser Darstellung grau. Die sogenannte breite und die schmale Furche sind deutlich zu erkennen. Das dreidimensionale Modell verdeutlicht die Ausrichtung der Pentoseringe des Rückgrats und der Stickstoffbasen. Die Ringe der Stickstoffbasen liegen orthogonal zur Längsachse der Helix und sind für die hydrophoben Basenstapel-Wechselwirkungen (engl. „*stacking forces*“) verantwortlich, die neben den Wasserstoffbrückenbindungen (gestrichelt in der Konstitutionsformel) den größeren Anteil an der thermodynamischen Stabilität einer Helixstruktur bewirken. Die farbig hervorgehobenen Bestandteile in dem dreidimensionalen Modell sind die →Nukleotide, die durch →Polymerisation zum DNA-Einzelstrang führen.

Zur Hybridisierung kommt es, wenn ein DNA-Einzelstrang bei geeigneten thermodynamischen Verhältnissen (siehe Abschnitt 2.2) ein passendes Gegenstück findet. Eine DNA-Sequenz ist dann das Gegenstück einer anderen, wenn es an allen Positionen zu Watson-Crick Basenpaarungen (A•T Adenin mit Thymin bildet 2 Wasserstoffbrückenbindungen aus; G•C Guanin mit Cytosin bildet 3 Wasserstoffbrückenbindungen) kommt. Dabei werden die DNA-

[5] Quelle: Buch „Molekulare Genetik“ von Rolf Knippers;
http://www.drd.de/helmich/bio/gen/reihe2/karte212.html

Sequenzen gerichtet, vom 5´- zum 3´-Ende, aufgeschrieben. Also ist 5´-AGCTTCGGAT-3´ das Gegenstück (engl. *reverse-complement*) zu 5´-ATCCGAAGCT-3´ (vgl. Abbildung 2.1-2).

```
5´ATCCGAAGCT3´
  ||||||||||
3´TAGGCTTCGA5´
```

Abbildung 2.1-2: Watson- und Crick-Strang einer DNA

Für das Zustandekommen guter Hybridisierungssignale auf einem DNA-Mikroarray ist es notwendig, die Oligonukleotide so auszuwählen, dass sie allesamt ähnliche Hybridisierungseigenschaften haben. Eine der wichtigsten Hybridisierungseigenschaften ist die Schmelztemperatur. Diese ist definiert als die Temperatur, bei der sich die Hälfte der Moleküle im Zustand des Einzelstrangs (engl.: *random coil state*) und die andere Hälfte im Zustand einer Doppelhelix (engl.: *double-helical state*) befinden. Bei einem Hybridisierungs-Experiment mit einem DNA-Mikroarray wird beispielsweise eine, in Bezug auf die im →Hybridisierungsprotokoll gewählte Hybridisierungs-Temperatur, zu niedrige Schmelztemperatur eines Oligonukleotids dazu führen, dass der entsprechende Spot auf einem DNA-Mikroarray kein Hybridisierungssignal enthält. Eine zu hohe Schmelztemperatur hingegen wird dazu führen, dass ebenfalls leicht abweichende Ziel-DNA (eine Ziel-DNA mit einem oder relativ zur Länge des Oligonukleotids wenigen Basenaustauschen) ein Hybridisierungssignal erzeugt. Die Molekularbiologen nennen das eine Kreuzhybridisierung oder „unspezifische Hybridisierung“.

Es ist unabdingbar, die Hybridisierungseigenschaften einzelner Oligonukleotide berechnen zu können [83] und in Kombination mit einer →Ziel-Sequenz die Qualität eines Hybridisierungssignals vorhersagen zu können (siehe Abschnitt 4.2). Für die Berechnung der Schmelztemperatur und anderen thermodynamischen Eigenschaften stehen zahlreiche Programme zur Verfügung, [16], [45], [86], [121]. Mit großem Aufwand werden die zugrundeliegenden thermodynamischen Parametersätze verbessert [22], [96], [95], [105] und die Programme und Parametersätze um weitere Eigenschaften, wie die Berechnung von Hybridisierungen mit →Mismatches [1], [3], [2], [4], [70], [117] und um die Berücksichtigung des „*helix initiation factors*“ [105], erweitert.

2.2. Thermodynamik

Die im vorigen Abschnitt erwähnte Hybridisierung und auch die Ausbildung von Sekundärstrukturen sind thermodynamische Prozesse. Beide beeinflussen direkt das Zustandekommen von Hybridisierungssignalen, welche das zentrale Ziel der DNA-Analytik mit DNA-Mikroarrays sind. Bildlich gesprochen, handelt es sich bei diesen beiden Prozessen, die in einer Zelle oder auch in der Probe einer „Hybridisierung eines DNA-Mikroarrays“[6] ablaufen, um das millionenfache Zustandekommen und Wiederauflösen von inter- und intramolekularen Doppelsträngen.

Man möchte diese Prozesse so genau wie möglich beschreiben, ist jedoch nicht in der Lage jedes einzelne Molekül zu betrachten. Daher werden in der →Thermodynamik sogenannte makroskopische messbare Zustandsgrößen (Beobachtungsgrößen) definiert. Es wird von den Details der atomaren und molekularen Welt abstrahiert. Die soeben erwähnten Prozesse erreichen im Allgemeinen einen Gleichgewichtszustand. „Überläßt man ein abgeschlossenes System sich selbst, so streben die Erwartungswerte physikalischer Größen im Laufe der Zeit erfahrungsgemäß gegen konstante ‚Gleichgewichtswerte‘. Den Zustand, in dem vom makros-

[6] In der DNA-Analytik wird das Spülen der Probe über das DNA-Mikroarray im Rahmen eines Hybridisierungsprotokolls ebenfalls als Hybridisierung bezeichnet.

kopischen Standpunkt aus keine messbaren Änderungen mehr festzustellen sind, nennt man auch einen Zustand im thermischen (oder thermodynamischen oder statistischen) Gleichgewicht.“ [15]

Zustandsgrößen von thermodynamischen Prozessen sind unter anderem Druck p, Volumen V, Temperatur T oder die Zusammensetzung. Die Entropie S, die innere Energie U, die Enthalpie H und die freie Enthalpie G sind Zustandsfunktionen, die von den Zustandsgrößen abhängen. Die Werte der Zustandsfunktionen sind unabhängig von dem Weg im Phasenraum, auf dem der Zustand erreicht wurde. Die Tabelle 2.2-1 stellt Zustandsfunktionen und ihrer Beziehungen dar.

Tabelle 2.2-1: Zustandsfunktionen

Entropie	$S = k \cdot \ln P$	mit k = Boltzmann-Konstante; P = Zustandswahrscheinlichkeit
Innere Energie	U	
Enthalpie	$H = U + p\,V$	p = Druck; V = Volumen
Freie Energie	$F = U - T\,S$	T = Temperatur
freie Enthalpie	$G = H - T\,S$	T = Temperatur

Vielen Lesern ist die Entropie als ein Maß für den Grad der Unordnung eines Systems bekannt. Für den hier interessierenden Gleichgewichtszustand gilt: alle Zustandsgrößen bleiben zeitlich konstant und es gibt eine Zustandsgleichung (z.B. bei idealen Gasen: $p\,V = m\,R\,T$ mit der Gaskonstanten R und der Masse m). Im Gleichgewichtszustand ist die Entropie S im Maximum, die freie Energie F ist im Minimum und bei isothermen-isobaren Systemen ist die freie Enthalpie ebenfalls im Minimum.

Für die Optimierung von Oligonukleotid-Bibliotheken ist die Schmelztemperatur von DNA-Doppelsträngen eine der wichtigsten Eigenschaften. Sie wird entweder gemessen, durch Simulation ermittelt oder mit Hilfe von thermodynamischen Parametersätzen, den *nearest neighbor interactions* (siehe Abschnitt 4.2.4.1), berechnet. Definiert ist die Schmelztemperatur als die Temperatur bei der 50% der DNA im hybridisierten Zustand vorliegt. Die Abbildung 2.2-1 stellt zwei mit dem POLAND-Server[7] berechnete Schmelzkurven dar, aus denen die Schmelztemperatur abgelesen werden kann. Zu jeder Temperatur stellt sich ein Gleichgewicht zwischen doppelsträngiger und einzelsträngiger DNA ein. Mit den Schmelzkurven wird dargestellt, in welchem Verhältnis doppel- und einzelsträngige DNA vorliegt.

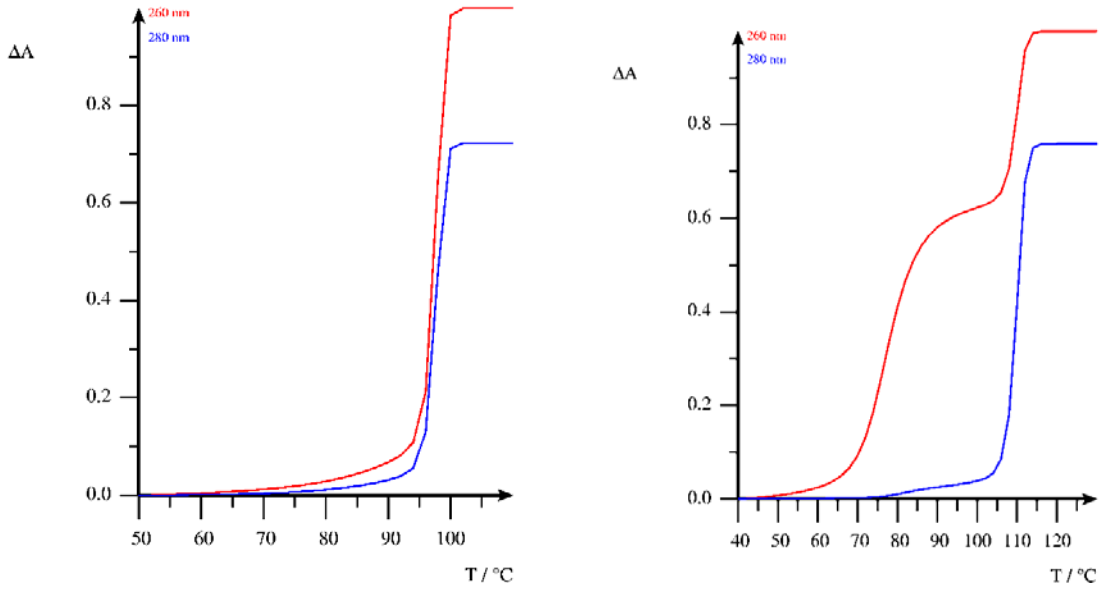


Abbildung 2.2-1: Schmelzkurven[7] zweier Sequenzen mit gleichverteiltem GC-Gehalt und mit Bereichen unterschiedlichen GC-Gehalts

Auf der x-Achse ist die Temperatur und auf der y-Achse die UV-Hypochromizität (UV-Absorption) bei 260 und 280 nm aufgetragen. In der Abbildung 2.2-1/links wurde eine →Sequenz mit annähernd gleichverteiltem GC-Gehalt angegeben. Daher ergibt sich hier der Prototyp einer sigmoidalen Schmelzkurve. In der Abbildung 2.2-1/rechts wurde eine Sequenz mit Bereichen unterschiedlichen GC-Gehalts vorgegeben. Einer der Bereiche bestand fast ausschließlich aus A und T, der andere überwiegend aus G und C. Die Schmelzkurve demonstriert mit dem zweistufigen Ansteigen der 260nm-Kurve, dass der entsprechende Doppelstrang stückweise aufschmilzt, zunächst zu einer Y-förmigen Struktur und anschließend zu zwei Einzelsträngen.

2.3. Sekundärstrukturen

Die Abfolge der Basen einer Nukleinsäure bildet die Primärstruktur und zugleich die höchste Abstraktionsstufe dieses komplexen Moleküls, das in der Lage ist, zahlreiche räumliche Konformationen einzugehen. Zwei konkretere Modelle sind die Sekundärstruktur, bei der intramolekulare →Basenpaarungen berücksichtigt werden, und die Tertiärstruktur, die dreidimensionale Anordnungen umfasst. Algorithmen zur Berechnung von Sekundärstrukturen verwenden dynamische Programmierung [123], von den →Neuronale Netzen das thermodynamisch motivierte Hopfield Netz [44] oder aufwändige der „Monte Carlo Methode“ ähnliche Simulationen auf verschiedenen Abstraktionsstufen, z.B. [26] „at the level of single base-pairing events“. Zu den bekanntesten Programmen zur Berechnung von Sekundärstrukturen gehören mfold von Michael Zuker [123], [69] und Vienna RNA [110], [42] von Ivo Hofacker. In [23] wird ein interessanter Ansatz vorgestellt, bei dem das Programm mfold mit experimentell gewonnenen Nebenbedingungen (*constraints*) unterstützt wird. In einem Experiment werden Enzyme verwendet, die bei Hairpin Strukturen mit einem Stem von mindestens 7 bp (→bp) an spezifischen Positionen (*cleavage sites*) schneiden. Aus der fragmentierten DNA werden diese Positionen ermittelt und als Nebenbedingungen für mfold umformuliert. „Incorporating constraint parameters obtained from experimental data into computational methods used to predict secondary structures can greatly improve the results.“ [23]

[7] Diese Schmelzkurven wurden mit dem POLAND-Server auf http://www.biophys.uni-duesseldorf.de berechnet.

Sekundärstrukturen können ganz beträchtlich das Zustandekommen von Hybridisierungssignalen behindern [19], [23], [75], [100]. Sekundärstrukturen der Fänger-DNA, wie auch der Ziel-DNA bzw. Ziel-RNA zerstören die Zugänglichkeit, die für das Zustandekommen einer Hybridisierung zwischen diesen beiden Molekülen notwendig ist. „DNA sequence analysis by oligonucleotide binding is often affected by interference with the secondary structure of the target DNA“ [23]. Eine Sekundärstruktur mit einer großen Stabilität ist problematischer als eine mit einer geringen Stabilität, die in jeder natürlich vorkommenden →Sequenz zu finden sein wird. Die Stabilität einer Sekundärstruktur wird in der freien Enthalpie[8] ΔG gemessen. Je größer das negative ΔG vom Betrag ist, umso stabiler ist die zugehörige Sekundärstruktur. Zu jeder Sequenz gibt es ein ganzes Ensemble von mehr oder weniger stabilen Sekundärstrukturen, zwischen denen sich ein Gleichgewicht herausbildet. Der Großteil der Moleküle befindet sich in dem Zustand der Sekundärstruktur mit dem stabilsten ΔG. Jedes einzelne Molekül befindet sich im ständigen Fluss zwischen mehreren Sekundärstrukturen. Weiter unten wird auf dieses Ensemble der möglichen Sekundärstruktur näher eingegangen.

Die Elemente einer Sekundärstruktur sind „dangling ends“ oder „single stranded regions“, Stacking regions oder auch Stems, Interior- und Haripin-Loops, Bulges und Multi-Loops, welche in der Abbildung 2.3-1 dargestellt sind. Im Allgemeinen gilt, dass eine Sekundärstruktur mit vielen, langen oder GC-reichen Stems sehr stabil ist.

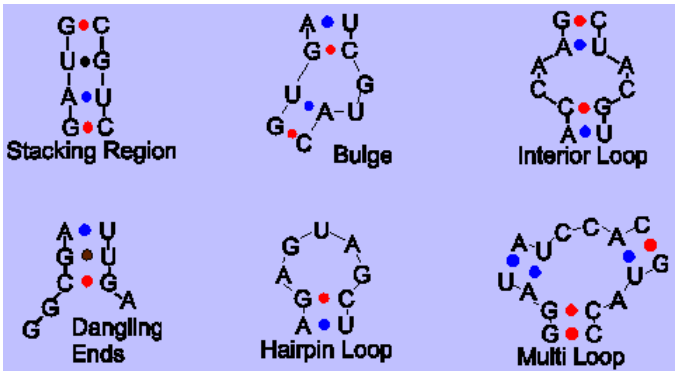


Abbildung 2.3-1: Elemente einer Sekundärstruktur

Die Abbildung 2.3-2 veranschaulicht eine ungünstige Sekundärstruktur. Nicht nur weil die Sekundärstruktur aus vielen, langen Stems besteht, sondern auch weil die rot eingezeichnete Sequenz, welche die Position eines →Oligonukleotids hervorheben soll, auf drei dieser Stems liegt und offensichtlich ebenfalls eine Sekundärstruktur von mindestens 5 Basenpaaren ausbildet. „For ASOs [→antisense oligonucleotides] to be effective, the complementary target sequence on →mRNA must be available for hybridization“ [19]. Trotz einer geringeren Stabilität gilt dieses Argument ebenso für DNA. „Our strategy is to focus on single-stranded regions in →RNA secondary structure, in particular those of at least four consecutive unpaired bases“ [19].

[8] Die freie Enthalpie ΔG wird häufig auch falsch mit freier Energie bezeichnet.

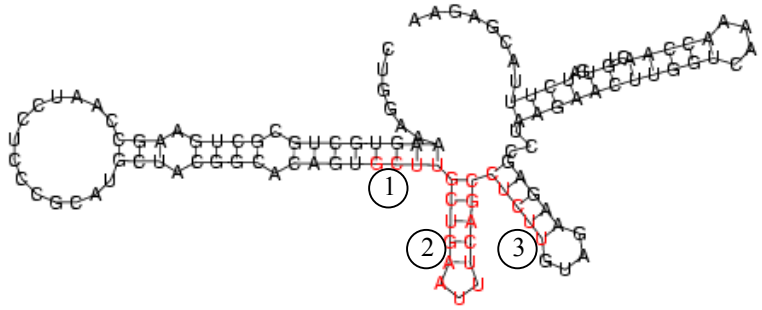


Abbildung 2.3-2: Beispiel einer schlechten Position eines Oligonukleotids auf einer Sekundärstruktur[9]

In der Abbildung 2.3-3 wurde unter den gleichen thermodynamischen Bedingungen (Temperatur, Salzgehalt usw.) eine andere Ziel-DNA gefaltet. Die resultierende Sekundärstruktur enthält erkennbar weniger lange Stems, und das Oligonukleotid liegt bis auf zwei →Basenpaarungen ausschließlich auf einzelsträngigen Bereichen. Somit ist hier zu erwarten, dass sich bei einer →Hybridisierung auf einem DNA-Mikroarray ein deutlich besseres Hybridisierungssignal ergibt.

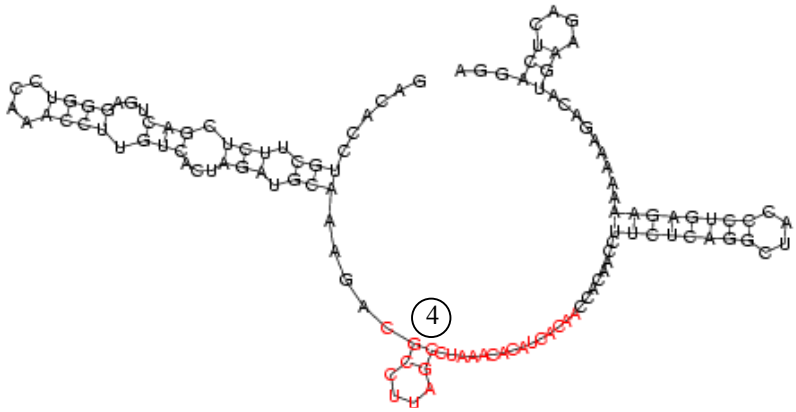


Abbildung 2.3-3: Beispiel einer guten Position eines Oligonukleotids auf einer Sekundärstruktur[9]

Diese Erwartung wurde durch Experimente in Zusammenarbeit mit dem UFT der Universität Bremen [77], [80] und in anderen Arbeitsgruppen [19], [23] bestätigt. Die Problematik der Sekundärstrukturen ist jedoch noch nicht gelöst. Zum einen muss die hier demonstrierte Interpretation einer Oligonukleotid-Position auf einer Ziel-DNA-Sekundärstruktur in einer Bewertungsfunktion (Score) dargestellt werden können, um sie im Batch-Betrieb einem →Algorithmus zugänglich zu machen. In dem Abschnitt 4.2.3 wird ein Verfahren dazu vorgestellt. Zum anderen bildet eine Ziel-DNA nicht nur eine Sekundärstruktur aus, sondern ein ganzes Ensemble von Sekundärstrukturen mit verschiedenen Stabilitäten bzw. freien Enthalpien ΔG.

[9] Für das Programm Vienna RNAfold wurde die Sequenz einer Ziel-DNA in eine RNA-Sequenz konvertiert. Daher sind in den Abbildungen die Thymin-Ts als Uracil-Us dargestellt.

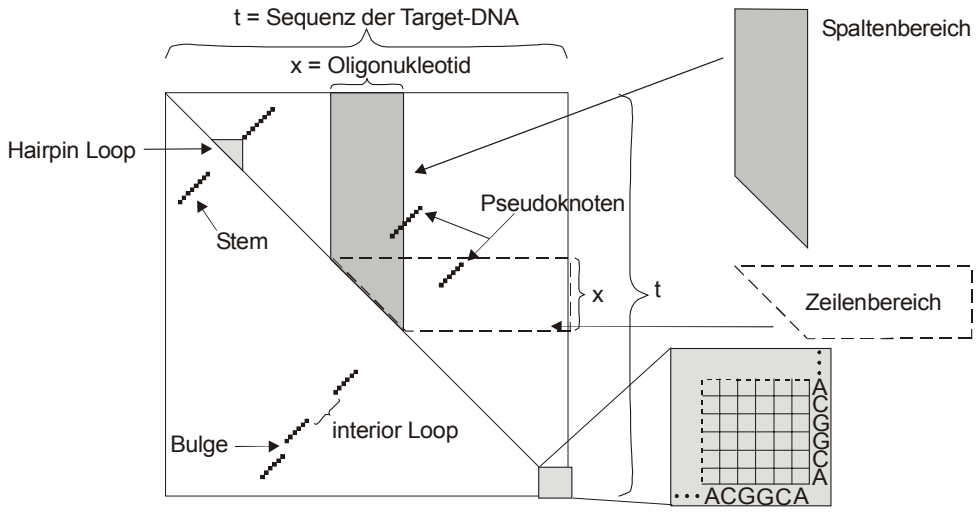


Abbildung 2.3-4: Strukturelemente und Erläuterungen zur Dotplot-Matrix

Die Abbildung 2.3-4 zeigt eine Visualisierungsform für Sekundärstrukturen, die in der Lage ist, für eine vorgegebene Sequenz alle möglichen Sekundärstrukturen gleichzeitig darzustellen, die Dot-Plot-Matrix. Die Sequenz, z.B. einer Ziel-DNA, und auch die Position eines Fänger-Oligonukleotids, welche in der Abbildung 2.3-2 rot hervorgehoben wurde, werden an den horizontalen Rändern von links nach rechts und an den vertikalen Rändern von oben nach unten aufgetragen. Die durch die Position des Fänger-Oligonukleotids definierten Spalten- und Zeilenbereiche heben die für eine Hybridisierung besonders kritischen Elemente der Sekundärstruktur hervor. Diese werden weiter unten am Beispiel der zwei bisher betrachteten Sekundärstrukturen näher erläutert.

Einige Elemente von Sekundärstrukturen (Stem, Bulge, Haripin Loop und Interior Loop; vgl. Abbildung 2.3-1) sind hier erneut schematisch dargestellt. Die obere Dreiecksmatrix ($i>j$) enthält im wesentlichen die gleiche Information wie die untere Dreiecksmatrix ($i<j$). An der Position (i, j) befindet sich die Wahrscheinlichkeit dafür, dass die i-te Base der Ziel-DNA mit der j-ten Base paart, auch Basenpaarwahrscheinlichkeiten genannt. Eine durch (i, j) adressierte →Basenpaarung auf einer der Dreiecksmatrizen bezeichnet ebenfalls die Basenpaarung (j, i) auf der jeweils anderen Dreiecksmatrix. Anstatt die untere Dreiecksmatrix wegzulassen, wird dort häufig die stabilste Sekundärstruktur, die sogenannte MFE-Struktur (für *minimal free energy*) abgebildet. Die obere Dreiecksmatrix enthält die Basenpaarwahrscheinlichkeiten, dargestellt durch mehr oder weniger große Quadrate aller vorkommenden Sekundärstrukturen. Die zusammenhängenden diagonal angeordneten Basenpaarwahrscheinlichkeiten entsprechen den Stems auf der Sekundärstruktur. Zwei Basenpaarwahrscheinlichkeiten, die zugleich auf einer Zeile oder einer Spalte liegen, schließen sich gegenseitig aus, da in dem hier betrachteten Modell eine Base jeweils nur mit einer weiteren Base paaren kann. D.h. die in der Natur nachgewiesenen Basen-Tripel werden in diesem Modell ausgeschlossen. Ebenfalls ausgeschlossen sind Pseudoknoten, die bereits zu den Tertiärstrukturen gezählt werden und durch Basenpaarungen (i, j) und (i´, j´) mit $i<i´<j<j´$ charakterisiert sind. Die Abbildung 2.3-4 verdeutlicht, dass ein Pseudoknoten letztlich aus Basenpaarungen (i´, j´) eines Stems mit den Basen i´´ des Loops $i<i´´<j$ eines zweiten Stems mit den Basenpaarungen (i, j) bestehen. Weiterhin sind Basenpaarwahrscheinlichkeiten an den Positionen $|i-j|<4$ ausgeschlossen, da Hairpin-Loops eine Mindestgröße von drei Basen haben.

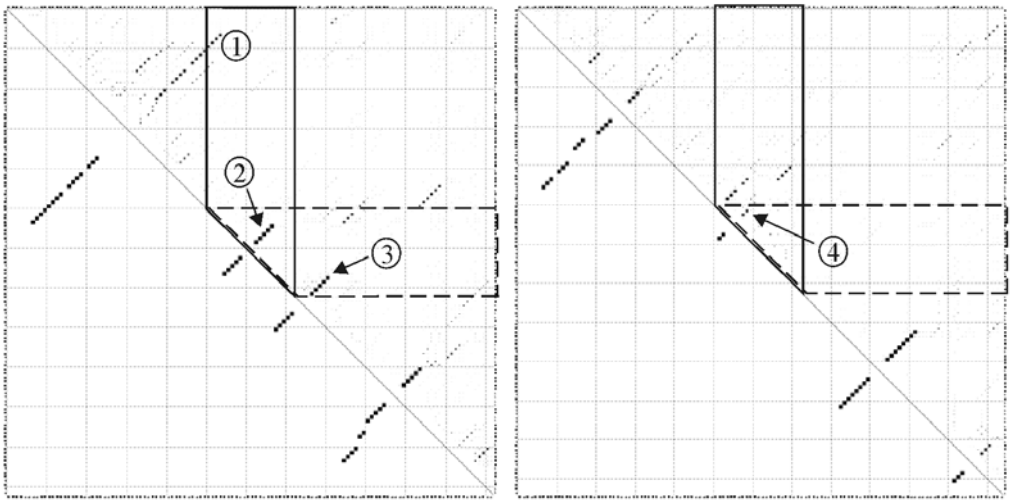


Abbildung 2.3-5: Dotplot-Matrizen

Mit etwas Übung erkennt man schnell die zwei oben abgebildeten MFE-Sekundärstrukturen. In der Abbildung 2.3-5/links ist die stabilere Sekundärstruktur aus der Abbildung 2.3-2 abgebildet, rechts die Sekundärstruktur mit den langen einzelsträngigen Bereichen aus der Abbildung 2.3-3. Links sind die vier Stems, zwei davon sehr lang unterbrochen durch interior Loops und Bulges, gut zu erkennen. Das Oligonukleotid ist durch die oben eingeführten Spalten- und Zeilenbereiche hervorgehoben. Die Überschneidungen des Oligonukleotids mit dreien der vier Stems erkennt man dadurch, dass die entsprechenden Basenpaarwahrscheinlichkeiten innerhalb der Spalten- oder Zeilenbereiche liegen. Die Stems sind hier und auch in Abbildung 2.3-2 und Abbildung 2.3-3 mit (1) bis (4) gekennzeichnet. Der Stem (2) liegt sogar in der Schnittmenge des Spalten- und Zeilenbereichs, dies ist der Stem, den die Ziel-DNA und das Oligonukleotid zugleich ausbilden. Weiterhin erkennt man, dass die Überlappung mit dem Stem (1) nur teilweise und die Überlappung mit dem Stem (3) vollständig ist.

In der Abbildung 2.3-5/rechts ist zu erkennen, dass sich nur zwei →Basenpaarungen der MFE-Sekundärstruktur, markiert mit (4), mit dem Oligonukleotid überlappen. Allerdings sind in dem Spaltenbereich ebenfalls zwei schwach ausgeprägte zur MFE-Struktur konkurrierende Stems zu sehen. In Abhängigkeit von der Dominanz der MFE-Struktur können diese Stems einer "suboptimalen" Sekundärstruktur ebenfalls das Hybridisierungssignal negativ beeinflussen.

Der Einfluss von Sekundärstrukturen auf Hybridisierungssignale wurde in den Arbeiten [80] und [77] untersucht. Dort wurde gefunden:

> „... solidphase hybridization studies have shown that individual oligonucleotides attached to the macromelecules display up to 100fold different hybridization efficiencies, depending on the specific nucleotide sequences. This is remarkable since the sequences of the →oligomers had been chosen for similar melting temperature (Tm), and thus comparable thermodynamic stability of the corresponding duplexes (ΔG). This indicated, that the sequence-specific hybridization efficiency is highly dependent on the presence of secondary structures, such as the formation of intramolecular hairpin loops ... The results from the microplate correlate with the formation of secondary structures ...“ [77]

Somit wurde bereits mehrfach eine Korrelation zwischen Hybridisierungssignalen und der visuell interpretierten Lage von Oligonukleotiden auf Sekundärstrukturen festgestellt. In dieser Arbeit wird in Abschnitt 4.2.3.2 ein Verfahren vorgestellt, dass eine vorhergesagte

Hybridisierungseffizienz in einer Bewertungsfunktion quantifiziert und auch suboptimale Sekundärstrukturen berücksichtigt.

2.4. Sequenzretrieval und Motivbestimmung

Das Ausgangsmaterial für die DNA-Analytik befindet sich in den großen internationalen Sequenzdatenbanken[10]. In einigen Fällen sind die interessierenden →Sequenzen, wie zum Beispiel bei der →Genexpression, bereits bekannt. Für einen flexiblen Einsatz der DNA-Analytik mit DNA-Mikroarrays für die Organismen-Identifikation ist jedoch zunächst ein Sequenzretrieval zu einer Fragestellung „Detektiere Organismus X und diskriminiere diesen gegenüber Y und Z“ nötig.

In der Arbeitsgruppe Prof. Schlieder des FuE-Verbundes Gensensorik wird ein System entwickelt, mit dem halbautomatisch und unter vorwiegender Kontrolle des Anwenders eine Bestimmung von hochsensitiven und hochspezifischen Sequenzmotiven vorgenommen werden kann. Dabei sind die folgenden Teilschritte vorgesehen: allgemeine Recherche, Datenbank-Sequenzretrieval, Untergruppenbestimmung, Datenvorverarbeitung, Bestimmung hoch sensitiver Sequenzmotive (→Sensitivität) und anschließend eine →Kontrollrecherche zur Sicherstellung der geforderten →Spezifität.

Aufgrund der voranschreitenden Sequenzierungsprojekte, z.B. des Humanen →Genom Projektes [47], [109], erfordert der rasant wachsende Datenbestand für bestehende DNA-Analytik-Projekte eine Aktualisierungskomponente, die neue, relevante Datenbank-Sequenzen berücksichtigt und gegebenenfalls vor der Verwendung veralteter Oligonukleotid-Bibliotheken warnt. Eine zunehmende Automatisierung des Sequenzretrieval-Prozesses ist ebenfalls wegen des rasant wachsenden Datenbestandes nötig, da der Umfang erzeugter Zwischenergebnisse kaum mehr ohne eine solche Unterstützung auszuwerten ist. Diese Auswertung erfordert jedoch domänenspezifisches Fachwissen, welches in der Arbeitsgruppe Prof. Schlieder modelliert und damit nach und nach der Gesamtprozess weitgehend automatisierbar wird. Bei diesem Sequenzretrieval-System werden Regionen eines Genoms (z.B. die 5´UTR oder die NS5-Region des Hepatitis C-Virus), Gene oder Motive bestimmt, die im Sinne eines „stepwise refinement“ der Optimierung einer Oligonukleotid-Bibliothek zugeführt werden.

2.5. Optimierung von Oligonukleotid-Bibliotheken

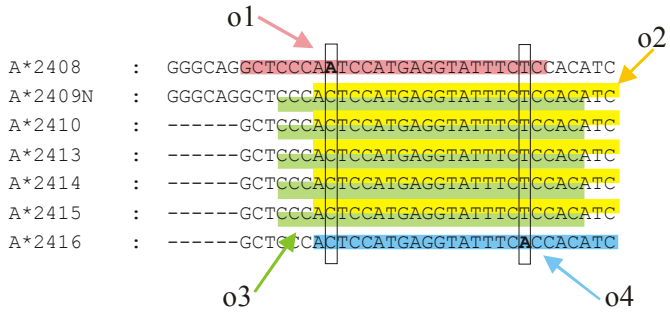
Ginge es bei der Optimierung von Oligonukleotid-Bibliotheken nur um die Optimierung der bereits erwähnten Hybridisierungs-Eigenschaften einzelner →Oligonukleotide, dann bräuchte man nur eine Bewertungsfunktion für einzelne Oligonukleotide konstruieren und man hätte die Gewissheit, dass man die optimale Lösung erhielte, wenn man nur nach dieser Bewertungsfunktion sortieren und die besten Oligonukleotide auswählen würde.

Für eine gute Sensitivität, d.h. Abdeckung z.B. einer Anzahl von Virus-Varianten, und Spezifität müssen die Oligonukleotide jedoch geeignet zusammengestellt bzw. kombiniert werden. Hier wird kombinatorische Optimierung benötigt. Probleme der kombinatorischen Optimierung besitzen eine sehr große Komplexität und in den meisten Fällen werden über Heuristiken nur angenäherte Lösungen berechnet, da eine exakte Lösung die Möglichkeiten der schnellsten Computer übersteigt.

[10] Die drei größten Sequenzdatenbanken sind die des EMBL (European Molecular Biology Laboratory) mit dem Hauptsitz in Heidelberg, des NCBI (National Center for Biotechnology Information) in den USA und die DDBJ (DNA Data Bank of Japan) des NIG (National Institute of Genetics).

Ein sehr einfaches Beispiel soll die Notwendigkeit der kombinatorischen Optimierung verdeutlichen. Dabei besteht die Bewertungsfunktion für dieses Beispiel nur aus der Sensitivität und Kriterien wie die Spezifität, die Schmelztemperatur oder Sekundärstrukturen des Fängers oder der Ziel-DNA, →Mismatch-Anzahl und -Position werden nicht betrachtet. Gegeben sei die Aufgabe, eine optimale Oligonukleotid-Bibliothek mit maximal drei Oligonukleotiden für die Identifikation eines Organismus zu erstellen. Wir nehmen an, dass dieser Organismus zu sieben verschiedenen Genotypen gehört, die in einem ausgewählten Sequenzbereich die in Abbildung 2.5-1 angegebenen Sequenzen enthalten.

Diese entsprechen den oben erwähnten →Ziel-Sequenzen und die farbig hervorgehobenen Sequenzen o1 bis o4 entsprechen den Fänger-Oligonukleotiden. Die Oligonukleotide o1 und o4 treffen nur eine Sequenz, während o2 und o3 fünf von sieben Sequenzen treffen. Die zwei vertikalen Kästchen heben die beiden Basenaustausche hervor, die diese Trefferanzahlen bewirken.



Sortierung der Oligonukleotide nach Sensitivität:

Oligo	Sensitivität
o2	5/7
o3	5/7
o1	1/7
o4	1/7

Abbildung 2.5-1: Beispiel-Sequenzen

Sortiert man nun die Oligonukleotide nach ihrer Güte, was hier der Sensitivität als einzigem Kriterium in der Bewertungsfunktion entspricht, dann erhält man die Reihenfolge o2, o3, o1, o4. Die Tabelle 2.5-1 verdeutlicht zusammen mit den resultierenden Sensitivitäten der gesamten Oligonukleotid-Bibliotheken L1 und L2, dass die Strategie S1, die besten drei Oligonukleotide zu wählen, nicht zur bestmöglichen Sensitivität führt. Die Strategie S1 führt zu einer Sensitivität von 6/7 und die Strategie S2 ergibt die bestmögliche Sensitivität.

Tabelle 2.5-1: Oligonukleotid-Bibliotheken zum obigen Beispiel

Oligonukleotid-Bibliothek	Strategie	Sensitivität
L1 = {o2, o3, o1}	S1 = „nehme die drei besten Oligonukleotide“	6/7
L2 = {o2, o1, o4}	S2 = „berücksichtige Kombinationen“	7/7

Zwar sind bei der kombinatorischen Optimierung für größere Probleme, wegen der großen Komplexität, exakte Lösungen nicht in akzeptabler Zeit berechenbar. Ein →Algorithmus jedoch, der unter Verwendung einer Heuristik, Kombinationen berücksichtigt und damit kombinatorisch optimiert, wird mit hoher Wahrscheinlichkeit zu besseren Ergebnissen als mit der Strategie S1 führen. Daher werden in dieser Arbeit heuristische kombinatorische Optimierungs-Algorithmen eingesetzt.

Auf einer abstrakteren Ebene betrachtet, stellt sich das soeben angegebene Beispiel für ein Optimierungsproblem, wie folgt dar. Die sieben Zeilen der Ziel-Sequenzen bilden die Menge

M = {1, 2, 3, 4, 5, 6, 7}. Die Oligonukleotide o1 bis o4 bilden Teilmengen dieser Menge M gemäß ihrer →Treffer auf den →Ziel-Sequenzen. Mit *Match* als Funktion auf der Menge aller Oligonukleotide K, die jedem Oligonukleotid eben diese Teilmenge von M zuordnet ist *Match*(o1) = {1}, *Match*(o2) = {2, 3, 4, 5, 6}, *Match*(o3) ist ebenfalls {2, 3, 4, 5, 6} und *Match*(o4) = {7}. Das oben angegebene Problem lässt sich nun mathematisch exakt formulieren: Finde die oder eine kleinste Teilmenge L aus K = {o1, o2, o3, o4} mit P = *Match*(K) = {*Match*(o1), *Match*(o2), *Match*(o3), *Match*(o4)}$\subset \wp$(M) der Potenzmenge[11] von M, sodass die Menge M vollständig überdeckt wird:

$$M = \bigcup_{x \in L} Match(x), \qquad L \subset K$$

Dieses Problem, welches für jede Menge M und jedes P$\subset \wp$(M) mit dem Tupel (M, P) vollständig charakterisiert ist, ist theoretisch bereits gut untersucht [17] und wird als „Set Cover"-Problem bezeichnet. Für das „Set Cover"-Problem gibt es eine gute heuristische Konstruktion einer suboptimalen Lösung, die im folgenden Abschnitt vorgestellt wird. Die bis hier ausgeblendeten Kriterien Spezifität unter Berücksichtigung eines hierarchischen Verwandtschaftsverhältnisses, Schmelztemperatur oder Sekundärstrukturen des Fängers oder der Ziel-DNA, Mismatch-Anzahl und -Position werden in den Kapiteln 4 und 5 behandelt.

2.5.1. Greedy Set Covering

Zu dem in Abbildung 2.5-1 angegebenen Optimierungsproblem und der Abstraktion zu einem (M, P) = ({1, 2, 3, 4, 5, 6, 7}, {*Match*(o1), *Match*(o2), *Match*(o3), *Match*(o4)}) „Set Cover"-Problem wird nun nach einer einfachen Heuristik eine möglicherweise suboptimale Lösung konstruiert. Es wird zunächst ein Oligonukleotid x ∈ K und damit ein *Match*(x) ∈ P gewählt, das möglichst viele Ziel-Sequenzen trifft, sodass *Match*(x) möglichst viele Elemente aus M überdeckt. Zum Beispiel o2 erfüllt diese Bedingung und wird als Teillösung der anfangs leeren Menge L hinzugefügt. Nun ist L = {o2} = {{2, 3, 4, 5, 6}}. Im nächsten Schritt wird dasjenige Oligonukleotid o1, o3 oder o4 ausgewählt, das am meisten Elemente aus M trifft, die bisher noch nicht getroffen wurden. Die Ziel-Sequenzen 1 und 7 werden von den Oligonukleotiden o1 und o4 getroffen, sodass in diesem Schritt zum Beispiel o1 der Menge L hinzugefügt wird. Es ist klar, dass dieser Algorithmus im dritten Schritt mit der Lösung L = {o2, o1, o4} abbricht. In diesem Fall ist diese Lösung sogar eine von mehreren optimalen Lösungen. Obwohl dieser Algorithmus bei diesem sehr einfachen Optimierungsproblem immer, d.h. unabhängig davon welches Oligonukleotid zu Beginn ausgewählt wird, zu einer optimalen Lösung gelangt, wird unten anhand von Beispielen als auch in theoretischen Betrachtungen aufgezeigt, dass häufig auch schlechtere Lösungen konstruiert werden.

Weil von Schritt zu Schritt jeweils der größtmögliche Zugewinn an Treffern oder Überdeckung von M angestrebt wird, heißt diese Strategie „Greedy Search" oder „Greedy Set Covering" [54], [99]. Der Greedy-Ansatz taucht als Heuristik in vielen Algorithmen der KI „Künstlichen Intelligenz" (z.B. beim Travelling Salesman Problem) und in der Bioinformatik (z.B. bei einigen Versionen von Algorithmen für das Berechnen von →Alignments) auf.

Die formale Angabe des Algorithmus für ein beliebiges (M, P) „Set Cover"-Problem ist sehr kurz. Zugunsten einer besseren Lesbarkeit werden hier nicht die Oligonukleotide x ∈ K, sondern deren über *Match*(x) ∈ P zugeordneten Teilmengen von M verwendet. D.h. die

[11] Die Potenzmenge von M = {1, 2, 3, 4, 5, 6, 7} ist die Menge aller Teilmengen $\wp$(M)={∅, {1}, {2}, ..., {1,2}, {1,3}, ..., M} von M, die $2^{|M|}$ Elemente enthält.

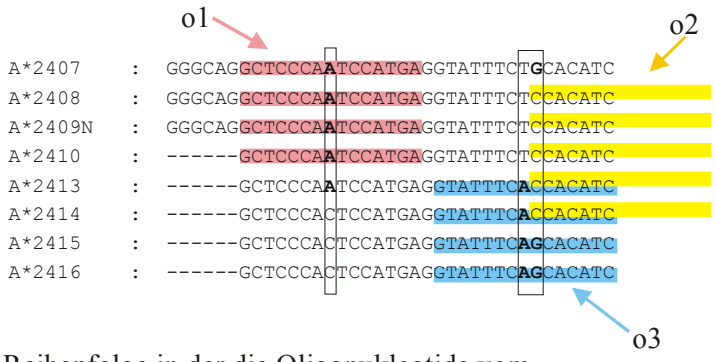
Lösung L´ ist hier nicht eine Teilmenge von K, sondern eine Teilmenge von P. Sei $M \neq \varnothing$ und $P \subset \wp(M)$:

1. L´ = { } ; hier L´ ⊂ P; unten L ⊂ K
2. sodass M - ($\bigcup_{y' \in L'} y' \cup y$) minimal ist
3. Setze L´ = L´ ∪ {y}
4. Wiederhole die Schritte 2 und 3 bis für alle y ∈ P gilt $\bigcup_{x \in L'} x \cup y = \bigcup_{x \in L'} x$

Schritt 2 ist der eigentliche Greedy-Schritt, in diesem wird, bei mehreren die Bedingung erfüllenden y ∈ P, eines beliebig ausgewählt. Die Menge L´ ⊂ P ist die Ausgabe dieses Greedy-Algorithmus. Mit L´ wird die Menge M ebenso gut überdeckt wie mit P, nur dass in den meisten Fällen weniger Elemente benutzt werden. Die Menge L von gesuchten Oligonukleotiden erhält man über $L = Match^{-1}(L')$, das Urbild von L´ unter der Abbildung $Match: K \rightarrow P$.

Leider garantiert dieser Algorithmus nicht, dass die oder eine optimale Lösung berechnet wird. Dazu müssten nach einem „Brute Force"-Ansatz $2^{|P|}$-1 Kombinationen angesehen werden. Die Komplexität dieses „Set Cover"-Problem ist $O(2^n)$ mit n = |P|. Es ist ein NP-vollständiges Problem, d.h. es ist kein deterministischer Algorithmus bekannt, der in jedem Fall in polynomialer Zeit eine optimale Lösung berechnet.

Um die Schwächen des Greedy-Algorithmus zu veranschaulichen, werden im folgenden weitere Beispiele konstruiert und theoretische Betrachtungen angestellt. Die typische Situation, die den Greedy-Algorithmus einen Fehler machen lässt, ist in einer Iteration ein Element von P in L´ aufzunehmen, das durch später aufgenommene Kombinationen von Elementen überflüssig wird.



Reihenfolge in der die Oligonukleotide vom Greedy-Algorithmus gewählt werden:

Oligo	Greedy-Score
o2	5
o3	2
o1	1

Abbildung 2.5-2: Problemfall für einen Greedy-Algorithmus

Für die Abbildung 2.5-2 wurde ein „Set Cover"-Problem konstruiert, das zu einer solchen Situation führt. Es kann formal mit (M, P) = ({1, 2, 3, 4, 5, 6, 7, 8}, {{1, 2, 3, 4}, {2, 3, 4, 5, 6}, {5, 6, 7, 8}}) angegeben werden.

Die dargestellten Oligonukleotide würden nach dem Greedy-Kriterium *„Wähle ein Oligonukleotid aus, das am meisten Elemente aus M trifft, die bis zu dieser Iteration noch nicht getroffen wurden"* oder in der Mengen-Version formuliert *„Wähle ein y ∈ P, das am*

meisten Elemente aus M überdeckt, die bis zu dieser Iteration noch nicht überdeckt wurden", d.h. nach Schritt 2 in dem oben angegebenen Algorithmus in der Reihenfolge o2, o3, o1 ausgewählt werden. Mit dem in der Abbildung 2.5-2 aufgeführten Greedy-Score eines Oligonukleotids wird die Anzahl der Elemente aus M bezeichnet, die von diesem Oligonukleotid getroffen werden, aber bis zu dieser Iteration noch nicht getroffen wurden. Es ist zu beachten, dass der Greedy-Score nur iterativ berechnet werden kann, da er von der Menge der bereits ausgewählten Oligonukleotide abhängt.

Hier würde der Greedy-Algorithmus mit einer Lösung L´ = {*Match*(o2), *Match*(o3), *Match*(o1)} terminieren (d.h. L = $Match^{-1}$(L´) = {o2, o3, o1}), während {o3, o1} die optimale Lösung ist. Damit würde ein Oligonukleotid zuviel verwendet werden. Das beabsichtigte mehrfache Treffen von Ziel-Sequenzen zur Schaffung von →Redundanz auf einem DNA-Mikroarray und Sicherheit bei der Auswertung von Hybridisierungssignalen wird an dieser Stelle ausgeblendet und in Abschnitt 4.2.2 behandelt.

Für heuristische Optimierungs-Algorithmen, die im Allgemeinen eine suboptimale Lösung berechnen, diese werden auch approximative Algorithmen genannt, ist es sehr wichtig zu wissen, wie stark die suboptimale Lösung von der optimalen abweicht. In dem betrachteten Beispiel enthielt die optimale Lösung 2 Oligonukleotide oder Elemente in L und die suboptimale 3, was zugleich wegen |P| = 3 die schlechtest mögliche Lösung ist.

In [17] wird für approximative Algorithmen der „Ratio Bound" ([99]: auch „worst case bound"; „classical harmonic upper bound" von 1978) definiert und für den „Greedy Set Cover"-Algorithmus berechnet. Das „Set Cover"-Problem aus diesem Abschnitt ist ein Minimierungs-Problem, da die Größe der resultierenden Menge L als „Kosten" aufgefasst werden kann, die selbstverständlich gering gehalten werden müssen. Der Begriff „Kosten", bezeichnet mit *C*, taucht in der folgenden Definition wieder auf, für das obige Beispiel gilt $C = |L|$ und für die Größe der Eingabe $n = |M|$. Mit C^* sind die Kosten einer optimalen Lösung bezeichnet.

> Definition: Ein approximativer Algorithmus hat den Ratio Bound $\rho(n)$, wenn für jede Eingabe der Größe *n* gilt:
>
> $$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

Wenn also $\rho(n)$ für alle möglichen Eingaben der Größe *n* eine obere Grenze für den Faktor ist, um den die Kosten *C* die Kosten der optimalen Lösung C^* übersteigen, dann ist diese definitionsgemäß der Ratio Bound $\rho(n)$. In der Definition ist neben dem $\frac{C}{C^*}$ für ein Minimierungs-Problem ($0 < C^* \leq C$) ebenfalls $\frac{C^*}{C}$ für ein Maximierungs-Problem ($0 < C \leq C^*$) eingebunden, und so gilt diese Definition für beide Problem-Typen.

Der Ratio Bound für das Beispiel aus Abbildung 2.5-2 ist nicht etwa 3/2, das Verhältnis der dort berechneten und der optimalen Lösung, da dieses Beispiel **nur eine** mögliche Eingabe für diesen Algorithmus ist.

In [17] wird der Ratio Bound für den „Greedy Set Cover"-Algorithmus zunächst mit $H(\max\{|x| : x \in P\})$ angegeben oder auch etwas schwächer mit (ln |M| + *1)*. Dabei ist *H(n)* eine Partialsumme der harmonischen Reihe. Damit ist für das betrachtete Beispiel

$H(\max\{|x| : x \in P\}) = H(5) = 2{,}28\overline{3}$ oder $(ln\ |\mathrm{M}| + 1) = (ln\ 8 + 1) = 3{,}079$. Das Beispiel war so konstruiert, dass jede Greedy-Iteration eindeutig ist. Ein Beispiel, das exakt die vorausberechnete obere Grenze $H(\max\{|x| : x \in P\})$ realisiert wäre z.B. (M, P) = ({1, 2, 3, 4}, {{1, 2}, {2, 3}, {3, 4}}). Hier ist bereits die erste Greedy-Iteration nicht eindeutig, da jedes Element in P gleichgroß ist. Wenn im „Worst Case" zuerst {2, 3} beliebig ausgewählt wird, dann ist mit $C = |L| = 3$ und mit $C^* = 2$ das Verhältnis $C / C^* = 3/2 = H(2) = H(\max\{|x| : x \in P\})$.

In vielen Fällen jedoch wird der „Greedy Set Cover"-Algorithmus recht gute Ergebnisse liefern und manchmal auch optimale, wie in dem Beispiel aus der Abbildung 2.5-1. Dort würden die Lösungen {o2, o1, o4} oder {o3, o1, o4} gefunden werden, die beide optimal sind. Weiterhin wurde durch die Heuristik ein NP-vollständiges Problem mit exponentieller Komplexität auf polynomialer Komplexität reduziert. Die Schleife des auf Seite 20 angegebenen Algorithmus kann mit einem Zeitaufwand von O(|P| |M|) implementiert werden, und sie wird höchstens mit einer Häufigkeit von min(|P| , |M|) durchlaufen. Somit kann der gesamte Algorithmus leicht mit einer polynomialen Komplexität von O(|P| |M| min(|P| , |M|)) implementiert werden. Es gibt allerdings auch eine Implementierung mit einem linearen Zeitaufwand von $O(\sum_{x \in P} |x|)$.

Diese beiden positiven Eigenschaften, nahezu optimale Ergebnisse zu liefern und dabei eine geringere Komplexitäts-Klasse als das ursprüngliche Problem zu besitzen, machen den „Greedy Set Cover"-Algorithmus sehr nützlich für das Design von Oligonukleotid-Bibliotheken, da Aufgaben mit einer großen Anzahl von Oligonukleotid-Kandidaten bearbeitet werden können, ohne allzu schlechte Ergebnisse oder eine zu lange Laufzeit befürchten zu müssen. In dieser Arbeit wird der „Greedy Set Cover"-Algorithmus mit zwei weiteren Optimierungs-Algorithmen verglichen, nämlich mit dem →Gradientenabstieg und mit Genetischen Algorithmen.

2.5.2. Gradientenabstiegs-Verfahren

Der →Gradientenabstieg, im eindimensionalen Fall auch als Newton-Verfahren bekannt, ist ein häufig eingesetztes Verfahren zur Minimierung oder Maximierung von Bewertungsfunktionen, die in einem bestimmten Kontext etwa Kosten- oder Gewinnfunktionen darstellen. Wird ein Problem so modelliert, dass die Bewertungsfunktion bezüglich des zu optimierenden Parameters differenzierbar ist, dann kann der Gradientenabstieg angewendet werden.

Bei künstlichen →Neuronalen Netzen und ganz besonders bei Backpropagation-Netzwerken kommen Varianten des Gradientenabstieg-Verfahrens zum Einsatz. Künstliche Neuronale Netze sind ein Modell für Informationsverarbeitung, die an der Informationsverarbeitung der Nervenzellen in Gehirnen angelehnt ist. Massive Parallelität einer großen Anzahl einfacher Prozessoren, Fehlertoleranz gegenüber unsicheren und verrauschten Daten und das Lernen aus einer vorgegebenen Mengen von Daten sind die wichtigsten Eigenschaften von künstlichen Neuronalen Netzen.

Ein im folgenden eingeführter fuzzyfizierter Zugehörigkeitsgrad eines Oligonukleotids zu einer Oligonukleotid-Bibliothek, eine Schmelztemperatur oder die mittlere Länge der Oligonukleotide könnten Parameter für die Optimierung einer Oligonukleotid-Bibliothek für ein DNA-Mikroarray mit Gradientenabstieg sein. In dem Abschnitt 5.2 wird eine Kombination von Kompetition und Gradientenabstieg vorgestellt.

Der fuzzyfizierte Zugehörigkeitsgrad soll nun anhand der Beispiele aus den Abbildungen Abbildung 2.5-1 und Abbildung 2.5-2 veranschaulicht werden. Eine komprimierte Form ein „Set Cover"-Problem zu visualisieren ist in anhand der Beispiele aus diesen Abbildungen in

Abbildung 2.5-3 dargestellt. Auf der x-Achse ist die zu überdeckende Menge M abgebildet, auf der y-Achse die Elemente der Menge P. Eine Spalte $t \in M$ stellt demnach eine Ziel-DNA (*t* für engl. *target DNA*) dar und eine Zeile ein Fänger-Oligonukleotid. Ein weißes Quadrat wird an der Position (*x*, *t*) gesetzt, wenn das Oligonukleotid der Zeile *x* tatsächlich die Ziel-DNA der Spalte *t* trifft.

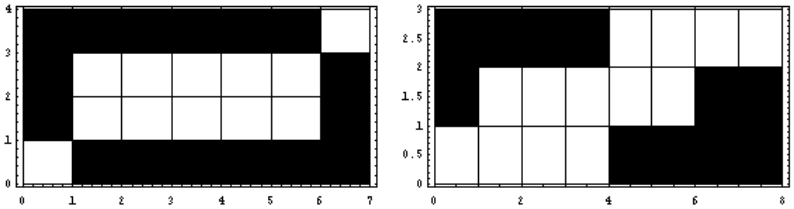


Abbildung 2.5-3: komprimierte Visualisierungen zu den Abbildungen Abbildung 2.5-1 und Abbildung 2.5-2

Diese Visualisierung eignet sich hervorragend, um den oben erwähnten fuzzyfizierten Zugehörigkeitsgrad darzustellen. In der Fuzzy-Logik können im Gegensatz zur klassischen Aussagenlogik Wahrheitswerte nicht nur mit 1 „wahr“ oder 0 „falsch“ dargestellt werden, sondern auch mit Wahrheitswerten zwischen 0 und 1. Ob man nun einen →Treffer bzw. ein Hybridisierungssignal an einer bestimmten Position erwartet oder ob man dieses Hybridisierungssignal für eine optimierte Oligonukleotid-Bibliothek benötigt, kann z.B. mit fuzzyfizierten Aussagen weich oder vage ausgedrückt werden. Die Abbildung 2.5-4 ist bei einem Ansatz zum letzteren Aussagentyp entstanden, ein dunkles Quadrat drückt aus, dass das Oligonukleotid *x* die Ziel-DNA *y* trifft, aber das dieses Hybridisierungssignal für die optimierte Oligonukleotid-Bibliothek nur einen geringen Beitrag leistet, weil z.B. andere Oligonukleotide bereits dieselbe Ziel-DNA treffen. Im Abschnitt 5.2 wird ein Optimierungs-Algorithmus für Oligonukleotid-Bibliotheken zu dieser Form der Kodierung angegeben.

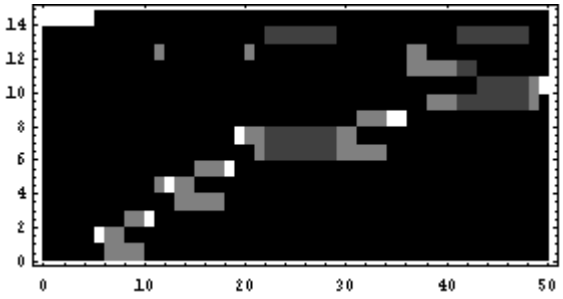


Abbildung 2.5-4: fuzzyfizierte Zugehörigkeit

Die Darstellung von →Mismatch-Treffern oder Hybridisierungs-Eigenschaften ist ebenfalls möglich und diese Visualisierung eignet sich recht gut für große „Set Cover“-Probleme bzw. Aufgabenstellungen mit vielen Oligonukleotid-Kandidaten und vielen Ziel-Sequenzen. In der Abbildung 6.3-2 ist ein recht großes Problem mit farbig kodierten Mismatch-Treffern dargestellt.

Sei X ein Raum, der soeben diskutierten und visualisierten Parameter, und sei $E: X \rightarrow \mathbb{R}$ eine differenzierbare Bewertungs- oder Fehlerfunktion. Der →Gradientenabstieg garantiert bei jeder Iteration eine Verbesserung $\Delta E \geq 0$ des zu optimierenden Parametersatzes. Die Abbildung 2.5-5 stellt das Prinzip des Gradientenabstiegs dar. Der Optimierungsprozess beginnt bei einem häufig beliebig gewählten Startpunkt $x \in X$. Von dort aus wird während jeder Iteration der Gradient ∇E (sprich: Nabla E) berechnet, der in die Richtung des steilsten Anstiegs zeigt. Mit $-\nabla E$, der entgegengesetzten Richtung, wird der Fehler schrittweise reduziert.

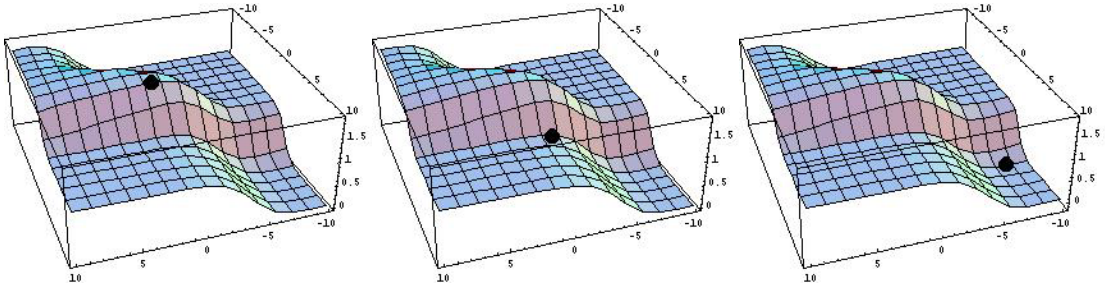


Abbildung 2.5-5: Prinzip des Gradientenabstiegs

Der Gradientenabstieg oder Methoden höherer Ordnung, wie die Levenberg-Marquardt Methode sind effiziente Methoden zur Fehlerminimierung. Bei Levenberg-Marquardt muss die Fehlerfunktion E zweimal differenzierbar sein. Der Startpunkt muss in der Nähe des Minimums liegen und die Fehlerfunktion muss hinreichend glatt sein. In der Praxis werden diese Voraussetzungen häufig nicht erfüllt. So manches Mal bleibt ein Gradientenabstieg in einem „lokalen Minimum" stecken, welches das Optimum nur unzureichend annähert.

Bei künstlichen Neuronalen Netzen wird ein Effekt ausgenutzt, den Gershenfield in [33] in Anspielung auf den bekannten Begriff „The Curse of Dimensionality" für die kombinatorische Explosionen zunächst recht widersprüchlich „The Blessing of Dimensionality" nennt. Er behauptet, dass der Rechenaufwand in Abhängigkeit von der Problemgröße, und damit ist die Anzahl der Freiheitsgrade gemeint, anfangs wie allgemein bekannt steigt, dann aber wieder sinkt. Die Abbildung 2.5-6/rechts soll die Problemtypen (a: geringe Problemgröße, geringer Rechenaufwand), (b: mittlere Problemgröße, hoher Rechenaufwand) und (c: hohe Problemgröße, geringer Rechenaufwand) veranschaulichen. Der erste dieser Problemtypen (a) ist allgemein bekannt; hier hilft häufig Gradientenabstieg weiter. Zum letzten Problemtyp (c) haben Untersuchungen hochdimensionaler Funktionen in der Festkörperphysik gezeigt, dass die Energiefunktionen, diese entsprechen den hier betrachteten Fehlerfunktionen, sehr viele „gute" lokale Minima haben. Daher eignen sich Probleme dieses Typs für künstliche Neuronale Netze oder „lokale Suche" [44] in Kombination mit „Simulated Annealing".

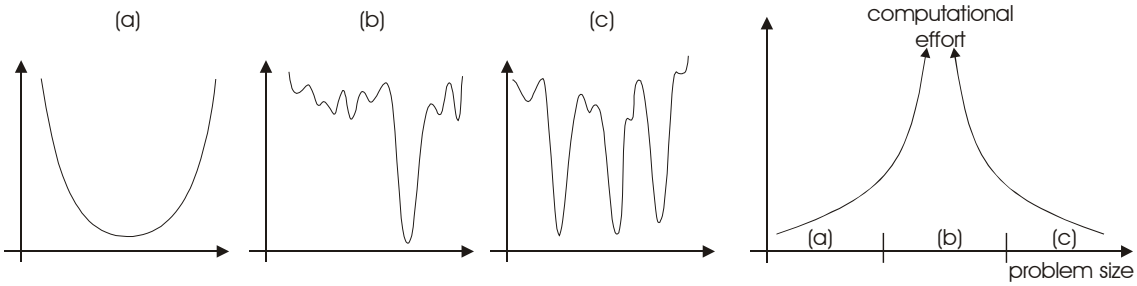


Abbildung 2.5-6: Charakteristik des Rechenaufwands als eine Funktion der Problemgröße (aus [33], Seite 167)

> "This figure also helps to explain the success of neural networks. If a model is not going to have a small number of meaningful parameters, then the best thing to do is to give it so many adjustable parameters that there's no trouble finding a good solution, and prevent overfitting by imposing priors." (Zitat [33], Seite 167)

Der Trainings-Prozess eines Neuronalen Netzwerks enthält häufig so viele Freiheitsgrade, dass die oben erwähnte Fehlerfunktion E sehr viele lokale Minima enthält, die das Optimum recht gut annähern. Somit wird bei künstlichen Neuronalen Netzen der Effekt „The Blessing of Dimensionality" gut ausgenutzt.

Der mittlere oben erwähnte Problemtyp (c) ist nach Gershenfield besonders schwierig, da hier simple globale Methoden, wie „Brute Force“ und Methoden, die den Such- oder Parameterraum X stark reduzieren (wie Gradientenabstieg als „lokale Suche“) scheitern. Es wäre sehr interessant zu wissen, mit welchem Problemtyp man gerade arbeitet. Eine Entscheidungshilfe liefert folgendes Zitat, das sich auf die Abbildung 2.5-6a-c bezieht:

> It’s not possible to decide which of these (types of search problems; Anm. des Autors) applies to a given nontrivial problem, because only a small part of the search space can ever be glimpsed. But a good clue is provided by the statistics of a number of local searches starting from random initial conditions. If the answer keeps being found the case on the left applies, if different answers are found but they have similar costs then the case on the right applies, and if there is a large range in the best solutions found then it’s the one in the middle. [33]

Sollte sich dabei der „Worst Case“ (Abbildung 2.5-6b) herausstellen, dann sind Genetische Algorithmen, die im folgenden Abschnitt vorgestellt werden, am besten geeignet, da sie weniger Rechenaufwand in die Optimierung eines Parametersatzes, d.h. eines Punktes im Suchraum X investieren, sondern möglichst global mit einem ganzen Ensemble von Punkten im Suchraum optimieren. Damit steht für den schwierigeren Fall der Probleme mit lokalen Minima durch Genetische Algorithmen ein angemessener Lösungsansatz zur Verfügung.

2.5.3. Ansatz mit Genetischen Algorithmen

Nach ersten Überlegungen in den 50er Jahren wurden Genetische Algorithmen [53], [33], [74] von John H. Holland 1975 in seiner Arbeit [43] mit dem Titel „Adaptation in Natural and Artificial Systems“ einer breiteren wissenschaftlichen Öffentlichkeit bekannt. Diese Algorithmen sind dem Prinzip der Evolution nachempfunden, bei dem sich eine Population von Individuen vorgegebenen Umweltbedingungen anpassen muss und sich weiterentwickelt. Abstraktionen und Modelle der soeben genannten Begriffe sorgen dafür, dass das Prinzip „Survival of the fittest“ für eine Realisierung als Algorithmus zugänglich gemacht wird. So werden z.B. die Umweltbedingungen abstrahiert zu einer Bewertungsfunktion auf der Menge der Individuen, und die Individuen selbst werden zu einer Kodierung von beliebigen aus Technik oder Naturwissenschaft stammenden Entitäten, die einer Optimierung bedürfen. Diese Optimierung ist gerade die Maximierung von Fitness. Die antreibenden Kräfte der Evolution sind die →Mutation, die Rekombination und das Zusammenspiel von Selektion und Reproduktion.

Genetische Algorithmen zeichnen sich dadurch aus, dass sie nicht so leicht in lokalen Minima stecken bleiben. Mit ihnen können diskrete Probleme, kontinuierliche Probleme und Kombinationen davon gelöst werden, und hinter dem eigentlichen Algorithmus (siehe Abbildung 2.5-7) verbirgt sich keine komplizierte Mathematik. Weiterhin kann der Genetische Algorithmus leicht auf neue Anforderungen angepasst werden und ist ohne Probleme parallelisierbar. Nachteile sind eine große Anzahl von Parametern sowie ein hoher Rechenaufwand. Die Konstruktion einer geeigneten Bewertungsfunktion, d.h. die Quantifizierung von Fitness, kann ein schwieriges Problem sein.

Die Kodierung von Individuen wird in Datenstrukturen wie Matrizen, Bäumen aber meistens in Zeichenketten bzw. Listen über einem Alphabet vorgenommen. Der →Code eines Individuums wird häufig als Chromosom bezeichnet und einzelne Merkmale auf dem Chromosom als Gen. Für diese Datenstrukturen wurden in der Literatur zahlreiche Versionen für die genetischen Operationen (Mutation und Rekombination) aufgeführt. Die Konstruktion dieser Operatoren hängt stark von der Optimierungsaufgabe und der gewählten Kodierung ab. So ist beispielsweise für einen nützlichen Rekombinations-Operator Voraussetzung, dass sich das Optimierungsproblem in Teilprobleme zerlegen lässt, oder dass zumindest die Gene auf

dem Chromosom weitgehend unabhängig voneinander sind. Andernfalls würde die Rekombination nach einigen Iterationen gefundene optimale Zusammenhänge unter den Genen zerstören. Es lohnt sich sehr eine gute Kodierung und eine nützliche Rekombination zu entwickeln, denn dieser Operator führt in den Suchprozess eine Form der Kollaboration innerhalb der Population ein. Dieser Informationsaustausch ermöglicht große Sprünge in dem Suchraum, die sonst nur durch eine lange Serie von Mutationen erzielt würden.

Mutationen sind im Allgemeinen zufällige Veränderungen auf dem Chromosom. Wenn möglich werden diese auch zielgerichtet implementiert, sodass der Suchprozess beschleunigt wird. Ein Genetischer Algorithmus besteht aus einer Schleife mit Abbruchkriterium. Die Schleifendurchläufe werden Generationen genannt. Die folgende Abbildung skizziert einen Genetischen Algorithmus.

1. Generierung einer zufälligen Anfangspopulation
2. Berechnung der Fitness jedes Individuums in der Population
3. In Abhängigkeit von der Fitness werden bestimmte Teile der Population für die Reproduktion selektiert und dabei den Operationen Mutation bzw. Rekombination unterzogen; andere Teile der Population werden durch die Reproduzierten ersetzt.
4. Die Schritte 2 und 3 werden wiederholt bis eine hinreichende Fitness erreicht oder eine maximale Anzahl von Generationen erzeugt wurde.

Abbildung 2.5-7: Genetischer Algorithmus

Es gibt zahlreiche in der Literatur beschriebene Selektions-, Reproduktions- und Ersetzungs-Schemata für den Schritt 3 aus Abbildung 2.5-7. Sie sind von der Fitness der Individuen abhängig, sorgen meistens für eine konstante Populationsgröße und dienen wie die anderen Parameter des Genetischen Algorithmus (Reproduktions-, Mutations- und Rekombinations-Rate) zur Einstellung der Balance zwischen „Exploration und Ausnutzung". Suchalgorithmen beinhalten generell ein sogenanntes *exploration-exploitation-dilemma*. Ein zu starkes Gewicht auf Exploration (z.B. durch große Mutations- und Rekombinations-Raten) führt zu einer unnötig großen Anzahl von Generationen. Andererseits wird ein zu starkes Gewicht auf Ausnutzung (*exploitation*) von Fitness (z.B. durch zu starke Vermehrung der besten Individuen) dazu führen, dass die Vielfalt in der Population verloren geht und der Suchraum nicht vollständig durchlaufen wird.

Bei der Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays muss zunächst eine Kodierung für Oligonukleotid-Bibliotheken gefunden werden. Diese besteht bei einem (M, P)-Problem[12] aus einer Liste bzw. Menge L von Elementen aus der Menge der Oligonukleotide K. Eine Population ist demnach eine Menge von Oligonukleotid-Bibliotheken und die einzelne Bibliothek ein Chromosom. In die Bewertungsfunktion gehen die →Sensitivität und die →Spezifität der gesamten Bibliothek ein.

Der Mutations-Operator für Oligonukleotid-Bibliotheken tauscht ein beliebiges oder zielgerichtet ein möglichst schlechtes Oligonukleotid der Bibliothek gegen ein beliebiges oder ein die „Restmenge" möglichst stark überdeckendes Oligonukleotid aus. Dieser Operator ist der Motor des Suchprozesses auf der Ebene der einzelnen Oligonukleotide. Zu der zielgerichteten Variante des Mutations-Operators heißt es in [30]: „*A recent and very promising approach for combinatorial optimization is to embed local search into the framework of evolutionary algo-*

[12] Vgl. Abschnitt 2.5

rithms“. Gradientenabstiegs-Verfahren oder Greedy-Strategien können in die Operatoren der Genetischen Algorithmen integriert werden und führen zu sogenannten „hybriden Algorithmen“. Auch der Rekombinations-Operator kann in einer zielgerichteten „*local search*“ Variante und in einer nicht-zielgerichteten Variante formuliert werden.

Der Rekombinations-Operator tauscht beliebige oder in der zielgerichteten Version möglichst komplementäre Oligonukleotid-Teilbibliotheken zwischen zwei Bibliotheken aus. Im Gegensatz zum Mutations-Operator macht es für den Rekombinations-Operator während der ersten Generationen des Genetischen Algorithmus kaum einen Unterschied, ob die zielgerichtete Version oder die nicht-zielgerichtete verwendet wird, da alle Oligonukleotid-Bibliotheken der Population noch nicht optimierte Eigenschaften gesammelt haben. Nach vielen Generationen jedoch könnte ein ungerichteter Rekombinations-Operator die guten Eigenschaften zweier Oligonukleotid-Bibliotheken zerstören. Eine sorgfältige Konstruktion beider Operatoren ist daher unerlässlich.

Bei der Konstruktion von Genetischen Algorithmen und deren Operatoren sind die Entwickler sehr phantasievoll und erfinderisch. Wie auch bei den künstlichen Neuronalen Netzen haben die Konzepte und Objekte zumeist ein Vorbild in der Natur. Der Bezug zu Genen, Chromosomen, Population, Mutation und Rekombination wurde bereits hergestellt. Auch Konzepte für die Populations-Dynamik haben ihre Entsprechung, wie zum Beispiel die Insel- (engl.: *Islanding*) und Eliten-Bildung (engl.: *Elitism*) [74], bei denen ein beliebiger oder als besonders gut bewerteter Teil der Population separat evolviert wird. Ein sehr interessanter Ansatz lässt sich von der folgenden Tatsache ableiten, die sich auf in der Natur beobachtete Mutations-Typen bezieht:

> „Bei der Deletion wird ein Teilstring gelöscht, bei der Duplikation wird ein Teilstring dupliziert und dem Erbgut hinzugefügt. Biologisch spielt die Duplikation eine große Rolle, da besonders häufig benötigte Stoffe dadurch noch schneller hergestellt werden können. Informationstechnisch dürfte der einzige Vorteil der Duplikation darin liegen, daß nach einer →Mutation die eine Kopie des Duplikats zerstört, noch die zweite funktionsfähige Kopie vorhanden ist. Auf diese Weise gibt es die Möglichkeit, daß Mutationen positive Entwicklungen initiieren können, ohne das zwangsläufig negative Entwicklungen zum Tragen kommen.“
> (Quelle: http://fachpublikation.de/dokumente/01/19/01007.html)

Das Prinzip der Gen-Duplikation angewendet auf die oben beschriebene Kodierung für Oligonukleotid-Bibliotheken, also deren Chromosomen, würde bedeuten, dass einzelne Oligonukleotide der Bibliothek, sprich die Gene des Chromosoms, mehrfach in der Kodierung auftauchen. Ein Oligonukleotid kann beispielsweise dann als zur Bibliothek gehörig definiert werden, wenn sich mindestens drei Duplikate in der Kodierung befinden. Sollten sich dann bei einem Individuum fünf Duplikate in der Kodierung befinden, dann wäre das Oligonukleotid sicher in der Bibliothek und eine Mutation auf einem der Duplikate würde daran nichts ändern. Wählt man nun n als maximale Anzahl von Duplikaten für ein Gen und skaliert die Anzahl von Duplikaten herunter auf Eins, dann wäre jede Mutation auf einem Duplikat eine graduelle Veränderung von $1/n$ und die Zugehörigkeit der Oligonukleotide zur Bibliothek wäre ebenfalls graduell bzw. „*fuzzy*“. Man könnte der Natur zugestehen, dass sie damit ein Prinzip erfunden hat, das man mit graduelle oder „*Fuzzy*-Mutation“ bezeichnen könnte. Die Natur löst auf diese Weise das oben erwähnte „*exploration-exploitation-dilemma*“. Die Mutations-Operatoren können neue Regionen des Suchraums erkunden, ohne auf die Ausnutzung bisher gefundener Information zu verzichten. Eine graduelle Zugehörigkeit der Oligonukleotide zur Bibliothek wird im Abschnitt 5.2 bei dem Ansatz über ein Gradientenabstiegs-Verfahren verwendet.

2.6. Auswertung einschließlich Interpretation

Mit der Auswertung von DNA-Mikroarrays [59] ist der gesamte Prozess nach der Durchführung des →Hybridisierungsprotokolls bis zur Formulierung des Analyse-Ergebnisses (z.B. „Die untersuchte Probe enthielt den Genotyp 1b des Hepatitis C-Virus“) gemeint. Sie umfasst für die Anwendungsgebiete →Genexpression [6], [7], [29] und Organismen-Identifikation (*„genotyping“*) [12], [60] die zwei Schritte „Quantifizierung der Hybridisierungssignale“ und „Interpretation“ der so gewonnenen Zahlenwerte zur Formulierung des Analyse-Ergebnisses.

Ein helles Hybridisierungssignal deutet auf die Anwesenheit von vielen Ziel-DNA-Molekülen hin. Ein Spot auf dem DNA-Mikroarray ist einem bestimmten Oligonukleotid zugeordnet, sodass über die Position auf die eingefangene Ziel-DNA geschlossen werden kann. Die Ziel-DNA wiederum wurde zuvor einem Virus, einem Bakterium oder einem beliebigen Organismus zugeordnet auf dessen Anwesenheit in der Probe in einem zweiten Schritt geschlossen werden kann.

Die Auswertung als letzter Schritt im Lebenszyklus von DNA-Mikroarrays gehört nicht zum zentralen Thema dieser Arbeit. Die →Konfigurierung von DNA-Mikroarrays hat jedoch viele Berührungspunkte zur Auswertung und Interpretation, da ein Teil der Aufgabenstellung die sichere Erzeugung von guten Hybridisierungssignalen ist. Das beabsichtigte mehrfache Treffen von →Ziel-Sequenzen zur Schaffung von →Redundanz auf einem DNA-Mikroarray und die Definition der Hybridisierungseigenschaften der Oligonukleotide in der Bibliothek gehören zu dieser Aufgabenstellung. Gerade bei dem Design von Oligonukleotiden, die im Verhältnis zu den →cDNAs relativ kurz sind, ist der Aspekt der Redundanz sehr wichtig:

> “gene expression levels are best analyzed with relatively long →probes; ... With long probes, it is possible to achieve good yields under stringent hybridization conditions. Hence it is possible to use a single spot of a →PCR product or clone to measure expression levels, whereas it has proved necessary to use sets of twenty 20-mers for each target to be sure that some would achieve levels of hybridization that are high enough.” [89]

In diesem Beispiel wird ein →Redundanz-Niveau (vgl. Abschnitt 4.2.2) von 20 empfohlen. Dabei wird eine möglichst große Sicherheit bei der Auswertung von Hybridisierungssignalen angestrebt, sodass die Interpretation aller Hybridisierungssignale zu sicheren Aussagen über die Anwesenheit von gesuchten Organismen oder Genen in der Probe führt.

Die Quantifizierung von Hybridisierungssignalen, wie sie in der Abbildung 1.6-2 dargestellt wurden, wird z.B. von Softwarepaketen wie Imagene oder GenePix durchgeführt. Ein Spot wird bei GenePix mit einem, bei Imagene mit mehreren Ringen umgeben (siehe Abbildung 2.6-1), welche Bereiche definieren, in denen das Hybridisierungssignal aufintegriert bzw. das lokale Hintergrundrauschen ermittelt wird. Da sich das Hintergrundrauschen auf dem DNA-Mikroarray großflächig ändert, ist es sehr wichtig ein lokales Hintergrundrauschen in die Quantifizierung von Hybridisierungssignalen einzubeziehen.

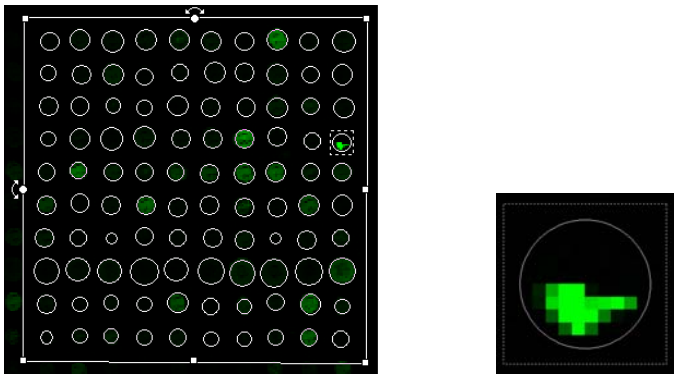


Abbildung 2.6-1: Quantifizierung von Hybridisierungssignalen[13]

Das Hintergrundrauschen entsteht durch Oberflächeninhomogenitäten und durch unspezifische Bindungen von Ziel-DNA auf der Mikroarray-Oberfläche, die bei dem Schritt „Waschen“ des →Hybridisierungsprotokolls nicht entfernt wurden. Das Hintergrundrauschen wird ebenfalls quantifiziert und mit dem eigentlichen Hybridisierungssignal verrechnet. Nach der Quantifizierung hat man ein Array von Zahlenwerten, und nach Anwendung der Zuordnung zwischen Spotposition und Oligonukleotid ergibt sich eine Tabelle der folgenden Form:

Oligonukleotid	Hybridisierungssignal
GCTACGTCGGCTTAGGATCGATCG	22
CGTTATCGGCTTAGTAGCCTGAG	87
...	...

Der logische Schluss von einem Zahlenwert eines Hybridisierungssignals eines Oligonukleotids zu der Anwesenheit einer Ziel-DNA kann nur unter Ausschluss von unspezifischen Hybridisierungen vorgenommen werden. Daher muss seitens des →Hybridisierungsprotokolls und der →Konfigurierung sichergestellt werden, dass keine oder möglichst wenige unspezifischen Hybridisierungen auftreten. Dies geschieht, indem vom Hybridisierungsprotokoll ein „Mismatch-Abstand“ (→Mismatch) oder ein durch eine thermodynamische Größe berechneter Abstand von Hybridisierungs-Stabilität (vgl. Abschnitt 4.2.1) vorgegeben wird, bei dem eine Diskriminierung von zwei Hybridisierungssignalen durchgeführt werden kann. Bei der Konfigurierung von Oligonukleotid-Bibliotheken darf anschließend dieser Abstand nicht unterschritten werden. Im Falle von schwer zu diskriminierenden Ziel-DNAs wird durch Einführung von →Redundanz auch bei (teilweise) Unterschreitung des Abstands der logische Schluss auf die Anwesenheit einer Ziel-DNA kalkulierbar. Die Interpretation von DNA-Mikroarray-Daten ist ein Anwendungsgebiet von Bayesschen Netzen, die bei der Verrechnung von mehreren möglichst redundanten Hybridisierungssignalen Evidenzen für die Analyse-Ergebnisse berechnen.

Das hier beschriebene Auswertungs-Szenario gehört zum Anwendungsgebiet „*genotyping*“ [60] der DNA-Analytik, auch „*genetic classification*“ [92] genannt. Andere Anwendungen, wie zum Beispiel die →Genexpression [6], [7], [29], haben ihren Schwerpunkt auf der quantitativen Analytik und arbeiten mit Vergleichen von zwei oder mehreren Hybridisierungssignalen aus zwei oder mehreren Mirkroarray-Analysen.

[13] Die zwei Grafiken aus der Abbildung 2.6-1 sind einer Visualisierung der Software GenePix 3.0 entnommen.

3. Stand der Technik

Zusammenfassung: Es wird der Stand der Technik bei der manuellen und softwareunterstützten Bestimmung von Primern, Fänger-Oligonukleotiden und →Oligonukleotid-Bibliotheken für DNA-Mikroarrays beschrieben. Es wird berichtet, welche Kriterien für die DNA-Analytik und ähnliche Technologien, z.B. das selektive Markieren von →mRNA, verwendet werden und welche Softwarelösungen und Methoden der hier vorgestellten Aufgabenstellung am nächsten kommen. Diese sind die „manuelle" Erstellung von Oligonukleotid-Bibliotheken, Verwendung von (Batch) →Primer Design Programmen und erste kommerzielle Systeme, die sich zum Ziel gesetzt haben, DNA-Mikroarrays zu unterstützen.

Es wird festgestellt, dass die manuelle →Konfigurierung von Oligonukleotid-Bibliotheken ein zeitaufwändiger und fehleranfälliger Prozess ist. "→*Chip design is a process that can take months*" [34]. Zu einer Reihe von Software-Systemen der genannten Ansätze für DNA-Mikroarrays und ähnlichen Technologien werden in einer Tabelle die Kriterien angegeben, die diese erfüllen. Dem Autor ist kein universitäres oder kommerzielles System bekannt, das den hier aufgeführten Umfang von Kriterien und algorithmischen Eigenschaften abdeckt.

Man kann sich dem Stand der Technik bei der →Konfigurierung von DNA-Mikroarrays von zwei Seiten nähern. Zum einen kann man die im Umfeld der →Molekularbiologie entwickelte Software [50], [53], [107], [106] und deren Anwendbarkeit auf die DNA-Mikroarrays betrachten, zum anderen kann für allgemeine Zwecke entwickelte Konfigurierungs-Software eingesetzt werden, die im universitären Umfeld [56], [36] wie auch kommerziell [37] zur Verfügung steht. Dieser Konfigurierungs-Software müsste dann in Form von Wissensbasen (*Knowledge-Based Configuration* [38]) die Kriterien der Konfigurierung von DNA-Mikroarrays beigebracht werden. Auf den ersten Blick ist der zweite Ansatz gar nicht abwegig, denn bei DNA-Mikroarrays handelt es sich, wie bei vielen anderen technischen Systemen, um ein variantenreiches Produkt, das sich dadurch auszeichnet, dass „in den meisten Fällen die Kundenanforderungen auf die dafür geeigneten Komponenten und deren Parametrierung und Auslegung abgebildet werden müssen. Dabei sind oftmals komplexe Abhängigkeiten gegeben" [37]. Zugunsten einer flexibleren Systemarchitektur und einer unabhängigeren Entwicklung von an die Aufgabenstellung angepassten Algorithmen wurde in dem FuE-Verbund Gensensorik sowie in dieser Arbeit auf den Ansatz mit der Konfigurierungs-Software verzichtet.

Im folgenden wird der Stand der Technik bei der manuellen und softwareunterstützten Bestimmung von Primern, →Oligonukleotiden und Oligonukleotid-Bibliotheken für DNA-Mikroarrays beschrieben. Es wird berichtet, welche Kriterien für die DNA-Analytik und ähnliche Technologien, z.B. das DNA-Computing, verwendet werden und welche Softwarelösungen und Methoden der hier vorgestellten Aufgabenstellung am nächsten kommen.

3.1. „manuelle" Erstellung von Oligonukleotid-Bibliotheken

Mit der „manuellen" Erstellung von Oligonukleotid-Bibliotheken ist der zeitaufwändige Einsatz von Programmen zur Schmelztemperaturberechnung, →Primer Design (siehe auch Abschnitt 3.2) und Berechnung von →Alignments zur Bewertung der Homologie sowie Laborarbeit zur Bewertung der Hybridisierungseigenschaften von Oligonukleotiden gemeint. Über einen Zeitraum von Wochen oder Monaten werden Oligonukleotide gesammelt und

teilweise im Labor auf ihre Verwendbarkeit überprüft. In dem Bioinformatik-Buch [34] heißt es auf Seite 314 "→Chip design is a process that can take months".

Neben dem Optimieren von einzelnen Oligonukleotiden auf gute Hybridisierungssignale im Labor kommen bei der Erstellung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays weiterhin die Kriterien der →Sensitivität (hier wie auch im Abschnitt 2.5 nicht als →Nachweisgrenze gemeint) und der Spezifität hinzu. Ohne Unterstützung durch Software arbeitet heute sicher kaum noch ein Biologe, jedoch gerade bei variantenreichen Virengenomen kann, selbst mit Unterstützung durch einen Computer und eines zumeist selbstentwickelten Systems von Dateien und Protokollen, das „Zählen von →Treffern" auf →Sequenzen innerhalb eines Genotyps oder bzgl. der ganzen Virenpopulation und unter Berücksichtigung von Hybridisierungseigenschaften und unerwünschten Treffern bei anderen Genotypen (oder →Nichtziel-Sequenzen) zu einer Tortur werden. Möglich ist auch, dass sich bei einem solchen Vorgehen Fehler einschleichen oder dass zugunsten eines geringeren Aufwands Abstriche an der Qualität der Oligonukleotid-Bibliothek gemacht werden. Die folgenden Abschnitte belegen, dass Teile der gerade vorgestellten Problematik heute durch Software unterstützt werden können.

Sollte man sich nach mehreren Wochen Arbeit für eine andere Oligonukleotid-Länge entscheiden, ist die bis dahin geleistete Arbeit größtenteils wertlos, da alle Oligonukleotide einer Bibliothek gleichförmige Hybridisierungs-Eigenschaften haben müssen und über die Definition der Sensitivität und der Spezifität abhängig voneinander sind. Auch die Entscheidung für eine andere Region in dem →Genom eines Organismus würde bedeuten, alle bis dahin gefundenen Oligonukleotide zu verwerfen und die Arbeit erneut zu beginnen. Eine automatisch und schnell erstellte Konfigurierung kann problemlos mit anderen Parametern auch ein zweites Mal gestartet werden. Bei genug Rechenkapazität ist es ebenfalls denkbar, einige Parameter-Bereiche zu durchlaufen und anschließend das beste Ergebnis auszuwählen. In der Abbildung 5.1-1 wird ein Ergebnis dieses Ansatzes mit dem Parameter Schmelztemperatur dargestellt. Über diesen Parameter wird zugleich die mittlere Oligonukleotid-Länge eingestellt und es wurde bei diesem Parameter-Durchlauf eine Grenze ermittelt, bei der die Spezifität der Oligonukleotid-Bibliotheken sprunghaft ansteigt.

Als manuelle →Konfigurierung von Oligonukleotid-Bibliotheken ist ebenfalls das Vorgehen von Lockhart [62] bei der Firma Affymetrix einzustufen. Dort werden sehr viele Oligonukleotide auf „*high density*"-Mikroarrays im Hybridisierungs-Experiment getestet. Die Oligonukleotide, die falsch-negative (also kein) oder falsch-positive (also unspezifische) Signale geben, werden entfernt und der Prozess wird wiederholt, bis eine gute Oligonukleotid-Bibliothek zusammengestellt wurde. Dieses Vorgehen führt sicher zu guten Bibliotheken, ist jedoch sehr zeit- und kostenaufwändig und an eine proprietäre Technologie gebunden. Auch in ähnlichen Technologien ist das experimentelle „Ausprobieren" stark vertreten. „In →antisense oligonucleotide experiments for example, the choice of the target is purely empirical; [...] Experimental success can only be achieved by a ‚brute force' approach. [...] Invariably, the majority of the oligonucleotides are ineffective. The synthesis, purification, and evaluation of several dozen candidate antisense effectors is laborious and expensive, and therefore ill-suited for high-throughput development." [102]

Die manuelle Erstellung von Oligonukleotid-Bibliotheken ist nicht etwa veraltet, sie wird in vielen Forschungseinrichtungen und Firmen praktiziert. Eine Stellenausschreibung in der Zeitschrift *transkript – BioTechnologie Nachrichten-Magazin* von 7/2001 belegt dies:

> Gesucht: Biochip-Designer (m/w)
>
> Erfahrene Wissenschaftler (Medizin/Molekularbiologie), die die biologischen Inhalte unserer DNA- und Proteinchips spezifizieren. Erfahrungen in

> Genomanalyse bzw. genetischer Ursachen komplexer Erkrankungen erforderlich. Fundiertes Wissen in PCR- und RNA-Amplifizierungsmethoden erwünscht.

3.2. Primer Design / Batch Primer Design

Stand der Wissenschaft und Technik ist es, bei der Erstellung von Oligonukleotid-Bibliotheken Programme zur Berechnung der Schmelztemperatur und der Sekundärstruktur der Fänger-Oligonukleotide zu verwenden. Programme zum Design von Primern, wie Oligo 5.0, Vector NTI, ARB oder zahlreiche Internet-Seiten, werden ebenfalls eingesetzt. Sie unterstützen zwar die Auswahl von Oligonukleotiden bezüglich der Schmelztemperatur, der (Fänger-)Sekundärstrukturen und der Länge, nicht berücksichtigt werden jedoch die Sensitivität und Spezifität bezüglich anderer in der zu untersuchenden Probe möglicherweise enthaltenen Sequenzen. Dadurch ist es sehr schwierig, falsch-positive bzw. falsch-negative Ergebnisse auszuschließen bzw. zu minimieren. Ferner ermöglichen die meisten Programme jeweils nur manuell, einzelne Oligonukleotide bezüglich bestimmter Kriterien zu optimieren. Es gibt nur wenige Programme [88], die automatisiert ein „Batch →Primer Design“, also die Bestimmung einer ganzen Menge von Oligonukleotiden, zulassen. Jedoch auch hier werden die Abhängigkeiten der Oligonukleotide untereinander nicht berücksichtigt und eine kombinatorische Optimierung, wie sie in Abschnitt 2.5 beschrieben wurde, kommt ebenfalls nicht zum Einsatz. In [20] und [21] hingegen werden Greedy-Algorithmen zum *Batch Primer Design* vorgestellt.

Primer Design Programme [50], [60] werden häufig als Unterstützung bei einer manuellen Konfigurierung einer Oligonukleotid-Bibliothek eingesetzt, um z.B. die Schmelztemperatur oder die Fänger-Sekundärstruktur zu berechnen. Dabei müssen die Benutzer die Primer Design Programme mühsam „austricksen“, z.B. dann, wenn nur ein Primer benötigt wird und trotzdem eine Länge des Amplifikates für die Bestimmung der Position des zweiten Primers angegeben werden muss.

Teilweise wird recht simpel aus einem →multiplen Alignment ein homologer Bereich herausgesucht und ein Oligonukleotid grob mit der →Wallace-Regel [113] und einem vorgegebenen Schmelztemperatur-Bereich ausgewählt. Häufig erzielen Molekularbiologen mit diesem Vorgehen recht gute Ergebnisse. Für diesen Ansatz gilt sicher, dass recht gute Homologien vorausgesetzt sind, dass man beim Design nur eines Primer-Paares noch viele Freiheitsgrade hat und die →PCR zudem wenig stringent (→Stringenz) angesetzt werden kann. Bei DNA-Mikroarrays hat man sich zum Ziel gesetzt, auch die Ziel-DNAs zu treffen, die nicht durch gute Homologien abgedeckt werden [81], das gleichzeitige spezifische Identifizieren von mehreren Genotypen schränkt häufig die Freiheitsgrade stark ein und für die Generierung von guten Hybridisierungssignalen, mit denen man einzelne →Mismatches diskriminieren möchte, ist ein stringentes →Hybridisierungsprotokoll unerlässlich. Weiterhin ist die flüssig-Phasen-Hybridisierung der PCR weniger problematisch als die fest-Phasen-Hybridisierung bei DNA-Mikroarrays. Ein Grund dafür sind sterische Probleme der zumeist langen Ziel-Sequenzen, die bei der →Hybridisierung an der Oberfläche der DNA-Mikroarrays auftreten:

> “We have found that the nature of the support, and especially the nature of the linkage between the support and the oligonucleotides, greatly effects performance. In particular, we have found that an optimal density and length of linker increases the hybridization yield substantially.” [89]

3.3. Primer für das selektive Markieren von mRNA

Bei der Literaturrecherche ist ein Programm für das selektive Markieren von →mRNA [107] dadurch aufgefallen, dass dort eine Menge von Oligonukleotiden durch einen „novel search-

ing algorithm“ berechnet wird. „The algorithm can be used to define the minimal number of oligonucleotides of a given length capable of priming all genes within any genome“ ([107], Seite 681). Zusammen mit [53], das weiter unten beschrieben wird, ist dieses Paper das einzige, in dem eine kombinatorische Optimierung zur Konstruktion einer Oligonukleotid-Bibliothek beschrieben wird. Für beide Veröffentlichungen wird im folgenden erklärt, warum diese nicht für das Design von Oligonukleotid-Bibliotheken für DNA-Mikroarrays verwendbar sind.

Für →Genexpressions-Experimente werden in [107] „genome-directed primers“ (GDPs) für das selektive Markieren der Ziel-RNA mit Fluoreszenzfarbstoffen berechnet. Dabei musste eine Menge von 3924 →ORFs vollständig überdeckt werden. Die GDPs sind nur 7 oder 8 Basen lang und sollen die sonst üblichen „Random Primer“ bei Prokaryoten[14] ersetzen. Die Reaktion „Reverse Transkription“ (das Umschreiben von →mRNA in →cDNA bei gleichzeitigem Markieren mit einem Fluoreszenz-Marker; →RT-PCR) mit den Random Primern wird in der Regel wenig stringent angesetzt, sodass alle RNA-Sequenzen fluoreszenzmarkiert werden, die Transkripte (→Transkription) eines zu detektierenden Pathogens wie auch die des Wirtes. Sinn macht der Ansatz mit GDPs bei der in vivo →Genexpression, da dort die Transkriptions-Aktivitäten verschiedener →Gene (z.B. auch humane mRNA) berücksichtigt werden müssen.

In [107] wurde über einen Vergleich zweier Genexpressions-Experimente nachgewiesen, dass mit GDPs die mRNA von Bakterien spezifischer transkribiert wird. Die Anzahl von unspezifischen Signalen bei einer anschließend durchgeführten →Hybridisierung auf einem DNA-Mikroarray lässt sich auf diese Weise reduzieren. Die unspezifischen Signale wurden über einen Vergleich zweier Genexpressions-Experimente definiert: „Genes with signal intensities generated from spiked mammalian →RNA probes that did not have corresponding true signals from the pure mycobacterial →probes were considered genes with nonspecific signals“ ([107], Seite 680). Nach einer Optimierung des →Hybridisierungsprotokolls erhielt die Arbeitsgruppe um Talaat nur 13.5% unspezifische Signale bei den durch GDPs erzeugten Ziel-Sequenzen gegenüber 32% unspezifischer Signale bei Random Primern.

Weil bei dieser reversen Transkription nur sehr kurze Primer verwendet werden, macht eine Berücksichtigung der Schmelztemperatur wenig Sinn und auch die Primer-Sekundärstruktur wird aus diesem Grund bei GDPs nicht berücksichtigt.

In [107] wird der seit 1978 theoretisch gut untersuchte „Greedy Set Cover“-Algorithmus in seiner ursprünglichen Version verwendet und es sind keine Prozeduren für spezifische Diskriminierungen erkennbar. In Abschnitt 2.5.1 wurde gezeigt, dass durch den „Greedy Set Cover“-Algorithmus nicht die minimale Anzahl von Oligonukleotiden gefunden wird. Daher sind einige Aussagen in dem folgenden Zitat nicht nachvollziehbar:

> We have developed a computer-based algorithm for prediction of the minimal number of primers to specifically anneal to all genes in a given genome. ([107], Seite 679)

Nicht richtig ist, dass der Algorithmus für die Spezifität gesorgt hat. Die in dem Paper experimentell nachgewiesene höhere Spezifität war von vornherein in den →Sequenzen der betrachteten humanen und bakteriellen mRNA enthalten. Die für die Spezifität notwendige Modifikation des Algorithmus wird in dem Paper selbst vorgeschlagen: „If necessary, the

[14] Bei Eukaryoten wird ein Poly-dT Primer verwendet, da hier das Stopcodon (→Codon) ein Poly-A-Schwanz ist.

assay could be improved by [...] altering the algorithm to eliminate primers recognizing mammalian transcripts“ ([107], Seite 681; siehe Tabelle 3.5-1).

3.4. Oligonukleotid-Bibliotheken für andere DNA-analytische Verfahren

Zur Dokumentation des Stands der Technik wurden oben bereits Programme zum →Primer Design für →PCR und „Reverse Transkription“ zitiert. Hier wird das Programm Oligo-Selector beschrieben. Es wurde für eine Technologie der DNA-Analytik entwickelt, die einzelne →Hybridisierungen verwendet. In der Arbeitsgruppe um Dr. Alexander Kel wird in [53] ein Genetischer Algorithmus eingesetzt, um familien-spezifische Oligonukleotid-Bibliotheken für die „G protein-coupled receptor protein superfamily“ zu bestimmen. Er nennt diese Anwendung der DNA-Analytik „identification by hybridization“ in Anlehnung an das „Sequencing by Hybridization“. Mittlerweile scheinen sich die Begriffe „genotyping“ sowie „Organismen-Identifikation“ durchzusetzen.

Da für einzelne →Hybridisierungen die Schmelztemperatur im Experiment jeweils optimal angepasst werden kann, wird in dem Programm Oligo-Selector nicht auf Hybridisierungs-Eigenschaften optimiert. Auch auf die Berücksichtigung der Stabilitäten der Sekundärstrukturen der Fänger- und Ziel-DNA wurde verzichtet. Mit der Entscheidung für einen Genetischen Algorithmus wurde ein mächtiges, wenn auch rechenaufwändiges, Werkzeug für die heuristische Lösung des in Abschnitt 2.5 beschriebenen „Set Cover“-Problems und für die Optimierung von Spezifität gewählt. In [53] wird in drei Schritten vorgegangen, in Schritt 1 und 2 kommen Genetische Algorithmen zum Einsatz.

1. Definition einer ersten „guten“ Liste von Oligonukleotiden, die aus allen möglichen Varianten von Oligonukleotiden erzeugt wird. Mit Hilfe mehrerer Techniken wird das „selective potential“ der einzelnen Oligonukleotide berücksichtigt.
2. Aus dieser Liste werden mehrere Oligonukleotid-Bibliotheken mit optimierten Eigenschaften (→Sensitivität und →Spezifität) konstruiert („Design the best cumulative sets of oligonucleotides“). Das „selective potential“ jeder Oligonukleotid-Bibliothek wird auf der Basis des kumulativen „selective potential“ aller zugehörigen Oligonukleotide bestimmt. Jedem Oligonukleotid wird ein Gewicht zugeordnet, das dessen Beitrag zum „selective potential“ der gesamten Oligonukleotid-Bibliothek entspricht.
3. Eine ausgewählte Oligonukleotid-Bibliothek wird auf Sensitivität und Spezifität auf einer Menge von Kontrolldaten untersucht.

Es wurden Bewertungsfunktionen für die Bewertung der Sensitivität und Spezifität einer ganzen Oligonukleotid-Bibliothek entwickelt. Die Abbildung 3.4-1 zeigt, wie stark das Ergebnis einer Optimierung mit einem Genetischen Algorithmus von dessen Parametrierung abhängt. Für die Durchläufe 1, 2, 3, 4 und 5 in der Abbildung 3.4-1 wurden die Populationsgröße N_p, die Anzahl der nach jedem Generations-Schritt selektierten Individuen N_0 (Reproduktions-Rate) und zwei Mutations-Raten p_m und p_g variiert. Die Rekombinations-Rate p_r wurde bei allen Durchläufen konstant auf 0,1 gesetzt.

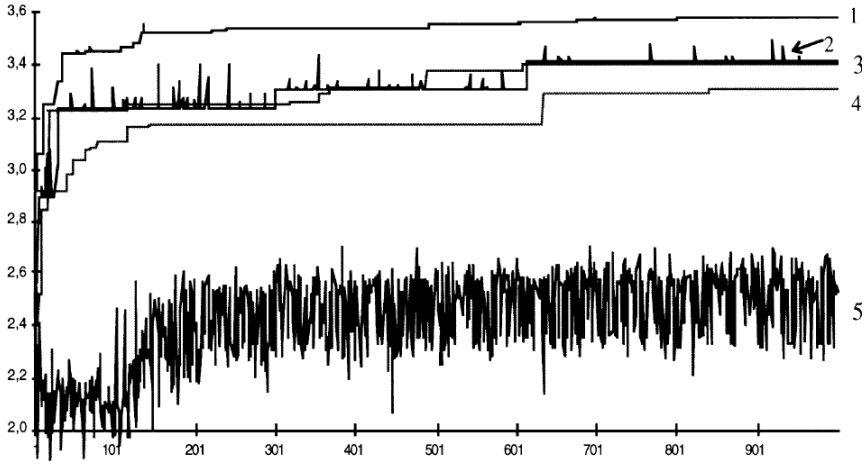


Abbildung 3.4-1: Aus [53] der Verlauf der Werte der Bewertungsfunktionen bei verschiedenen Parametrierungen (siehe Tabelle 3.4-1) des Genetischen Algorithmus

Weitgehend unabhängig von der Populationsgröße scheint die Güte des Ergebnisses zu sein. Die Geschwindigkeit, mit der gute Ergebnisse erzielt werden, ist hauptsächlich von den Mutations-Raten abhängig. Die Durchläufe 4 und 5 sind die schlechtesten und haben die größten Mutations-Raten (siehe Tabelle 3.4-1). Der Durchlauf 2 hat eine relativ große Reproduktions-Rate N_0, sodass häufiger gute Zwischenergebnisse durch Mutationen oder Rekombinationen zerstört werden und sich somit die „Spitzen" auf der Kurve zum Durchlauf 2 erklären. Man erkennt hier einen bereits in Abschnitt 2.5.3 „Ansatz mit Genetischen Algorithmen" erwähnten Nachteil der Genetischen Algorithmen, nämlich die große Anzahl von Parametern und die starke Abhängigkeit der Qualität des Ergebnisses von diesen.

Tabelle 3.4-1: Die Parametersätze aus [53] von 5 Durchläufen eines Genetischen Algorithmus; die Rekombinations-Rate p_r wurde jeweils auf 0,1 gesetzt

Durchlauf	1	2	3	4	5
N_p	10390	539	50	10390	250
N_0	199	199	7	199	37
p_g	4/5	4/5	4/5	19/20	1
p_m	1/5	1/5	1/5	6/10	6/10

Mit dem Programm Oligo-Selector wurde eine Oligonukleotid-Bibliothek für eine →Protein-Familie mit geringer Sequenz-Homologie konstruiert. Mit 15 Oligonukleotiden einer Länge zwischen 7 und 15 Basen konnten 70% der Ziel-Sequenzen der Protein-Familie korrekt identifiziert werden, bei nur 0,02% falsch-positiver Signale. Für das Design einer Oligonukleotid-Bibliothek für DNA-Mikroarrays wäre dieses Programm nicht einsetzbar, da eine Angleichung der Hybridisierungs-Eigenschaften der Oligonukleotide nicht durchgeführt wird, und da die für DNA-Mikroarrays so wichtige Optimierung der Hybridisierungseffizienz und Hybridisierungs-Spezifität (siehe Abschnitt 4.2.1) nicht berücksichtigt wird.

3.5. Stand der Technik - Zusammenfassung

Für einige an die DNA-Mikroarray Technologie angrenzende Gebiete, wie das →Primer Design für →PCR, →Antisense Oligonukleotide, „Reverse Transkription" oder für das „identification by hybridization" mit einzelnen Hybridisierungen, werden bereits Kriterien verwendet, die auch bei der Konfiguration von Oligonukleotid-Bibliotheken für DNA-Mikroarrays benötigt werden. Dazu gehören die Berücksichtigung der statistischen Sensitivität, vorausgesetzt, dass eine Eingabe mehrerer →Ziel-Sequenzen oder Motive mög-

lich ist, die Berücksichtigung der statistischen →Spezifität, die Schmelztemperatur, die Sekundärstrukturen der Fänger-Oligonukleotide und die sogenannte „GC-Clamp" oder „5´/3´ end stability". Kaum ein Programm berücksichtigt alle diese Kriterien (siehe Tabelle 3.5-1) und die meisten Programme wurden nicht für die Optimierung von Oligonukleotid-Bibliotheken für DNA-Mikroarrays entwickelt. Ausnahmen bilden ArrayDesigner, Hyb-Simulator, OligoLibrary und OligoPicker, die jedoch keine kombinatorische Optimierung durchführen und nicht die Sekundärstrukturen der Ziel-Sequenzen berücksichtigen. In [49] und [94] wurden ebenfalls nicht das „Set Cover"-Problem behandelt. In [49] wurde der Schwerpunkt auf die Entwicklung eines effizienten Algorithmus gesetzt, der die Anzahl der falsch-positiven Hybridisierungen minimiert und in [94] wurde BLAST für die Spezifität und Mfold für die Berücksichtigung der Sekundärstruktur der Oligonukleotide verwendet.

Tabelle 3.5-1: Kriterien, die bei der Auswahl von Primern und Oligonukleotiden berücksichtigt werden

Leistungsmerkmale einiger Software-Produkte für die Bestimmung von Oligonukleotiden	Oligo 6.0[15]	Vector NTI[16]	ARB[17]	Oligo-Selector[18]	Array-Designer[19]	Hyb-Simulator[20]	GDP	Oligo-Picker	Compugen[21]
Eingabe mehrerer Sequenzen / Motive	✓/✓	-/-	✓/-	✓/-	✓/-	?/-	✓	✓	✓
Schmelztemperatur Tm	✓	✓	✓	-	✓	✓	?	✓	✓
Sekundärstrukturen Fänger-DNA	✓	✓	✓	-	✓	✓	-	✓	✓
Sekundärstrukturen Ziel-DNA	-	-	-	-	-	-	-	-	-
Datenbank Sensitivität	✓	-	✓	✓	(✓)	✓	✓	?	?
Datenbank Spezifität / Kontrollrecherche	(✓)/- 6-7 mere	-/(✓)	✓/-	✓/-	-/-	✓/-	-	✓	✓
Optimierung mehrerer Oligonukleotide	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kombinatorische Optimierung einer Oligonukleotid-Bibliothek	-	-	-	✓	-	-	✓	-	-
Redundanz-/Toleranz-Niveau	(✓)	-		-			-	- ?	-
Hyb-Diskriminierung (DeltaG-Differenz oder MM-Anzahl)	-	✓		-			-	-	✓
GC-Clamp (5´/3´ end stability)	✓	✓		-			?	?	?
Hierarchische Beziehungen zwischen den Genotypen	-	-		-			-	-	-
Relative Lage der Oligonukleotide zueinander	-	-		-			-	-	-
Absolute Lage der Oligonukleotide auf dem Ziel	-	-	?	-	?	?	✓?	?	✓ 3´end
Mikroarray-Technologie	-	-	-	-	✓	✓	-	✓	✓

Nicht zum Stand der Technik gehören somit folgende Kriterien und Eigenschaften der Optimierungs-Algorithmen:

- Optimierung bezüglich eines →Redundanz-Niveaus (siehe Abschnitt 4.2.2). Das Redundanz-Niveau kann gerade zusammen mit der kombinatorischen Optimierung zur Verbesserung des Ergebnisses beitragen.
- Optimierung bezüglich eines →Toleranz-Niveaus (siehe Abschnitt 4.2.2).

[15] Oligo 6 : Molecular Biology Insights, Inc., 8685 US Highway 24, Cascade, CO 80809-1333, USA
[16] Vector NTI : InforMax, Inc., http://www.informaxinc.com/.
[17] ARB: Department of Microbiology, Technische Universität München, http://www.mikro.biologie.tu-muenchen.de/.
[18] Oligo-Selector: Arbeitsgruppe Dr. Alexander Kel, [53]
[19] ArrayDesigner: PREMIER Biosoft International, http://www.PremierBiosoft.com/
[20] ACGT – Advanced Gene Computing Technologies, Inc.
[21] Compugen: OligoLibraries™ http://www.labonweb.com/

- Berücksichtigung von Sekundärstrukturen der →Ziel-Sequenzen
- Berücksichtigung von vorhergesagten Hybridisierungseffizienzen (siehe Abschnitt 4.2.1 DeltaG-Differenz)
- relative Lage der →Oligonukleotide zueinander
- Berücksichtigung von hierarchischen Beziehungen zwischen den →Sequenzklassen (siehe Abschnitt 4.1.1)
- (neu im Bereich der DNA-Analytik mit DNA-Mikroarrays) kombinatorische Optimierung
- teilweise kombinierte Berücksichtigung mehrerer Kriterien in einem integrierten Algorithmus; teilweise Vorfilterung zur Reduzierung von Rechenzeit durch aufwändige Berechnungen (siehe Abschnitt 4.3)

Eines der Ergebnisse einer Fragebogenaktion, die zu einer Veröffentlichung mit dem Titel „classification of tasks in bioinformatics“ [103] führte, ist der Bedarf, Zwischenergebnisse von Bioinformatik-Werzeugen (→Alignments, Sequenzretrieval, →phylogenetische Analysen, usw.) softwarebasiert weiterverarbeiten zu können. Man benötigt demnach integrierte Systeme, die in der Lage sind, komplexe Aufgabenstellungen mit mehreren Bioinformatik-Werkzeugen automatisiert zu lösen. Eine Leistung dieser Arbeit besteht darin, die für das Design von Oligonukleotid-Bibliotheken für DNA-Mikroarrays notwendigen Kriterien zusammenzustellen und in ein integriertes System zu implementieren. Dem Autor ist kein universitäres oder kommerzielles System bekannt, das den hier aufgeführten Umfang von Kriterien und algorithmischen Eigenschaften abdeckt.

Die in [103] dokumentierte Nachfragen nach Integration von Bioinformatik-Werzeugen ist sicher ein Grund für den großen Erfolg der Skript-Sprache Perl in der Bioinformatik und ihrer Erweiterung zu BioPerl[22]. Es wurden bereits zahlreiche Schnittstellen zwischen Datenbanken und Bioinformatik-Werzeugen in Perl implementiert. Sogar einige für den manuellen Betrieb optimierte Web-Seiten können mit Perl automatisiert angesprochen und somit in einen größeren komplexen Prozess integriert werden. Teile des hier vorgestellten Systems zur Optimierung von Oligonukleotid-Bibliothek wurden in Perl entwickelt.

[22] die Internet-Seiten zu BioPerl: http://bioperl.org

4. Bewertungsfunktionen, Problemanalyse und Aufgabenspezifikation

Zusammenfassung: Es ist die Aufgabe der Bewertungsfunktionen, die zahlreichen Eigenschaften des Hybridisierungs-Prozesses von →Oligonukleotiden auf DNA-Mikroarrays zu quantifizieren und damit als Kriterium für den Optimierungs-Algorithmus zugänglich zu machen. Ebenfalls die Qualität einer ganzen Oligonukleotid-Bibliothek wird über die Berechnung der →Sensitivität und →Spezifität bewertet. Dazu ist es notwendig, Kriterien für die beiden Achsen der Vierfeldertafel zur Definition der Anzahlen von richtig-positiven, falsch-positiven, richtig-negativen und falsch-negativen Klassifikationen zu entwickeln.

Der Abschnitt 4.1 liefert die „Definition von →Ziel- und →Nichtziel-Sequenzen“ anhand einer hierarchischen Struktur zwischen den →Sequenzklassen und damit die x-Achse der Vierfeldertafel. Dabei wird unter Berücksichtigung dieser Strukturen zwischen den Sequenzklassen das Kriterium „maximale Spezifität“ erarbeitet und in das „Set Cover“-Problem integriert. Im Abschnitt 4.2 wird auf der Grundlage von Bewertungsfunktionen für die Hybridisierungseffizienz und Sekundärstrukturen sowie Parametern, wie dem →Redundanz- und →Toleranz-Niveau definiert, wann ein positives oder negatives Hybridisierungssignal zu erwarten ist. Nach der Bestimmung von fünf Parametern ist damit die Vierfeldertafel vollständig bestimmt und in einem letzten Abschnitt 4.3 wird eine formale Aufgabenspezifikation für die Algorithmen des folgenden Kapitels angegeben, bei der insgesamt 16 Parameter berücksichtigt werden.

Es ist die Aufgabe der Bewertungsfunktionen, die zahlreichen Eigenschaften des Hybridisierungs-Prozesses von →Oligonukleotiden auf DNA-Mikroarrays zu quantifizieren und damit als Kriterium für den Optimierungs-Algorithmus zugänglich zu machen. Das Kapitel über die Grundlagen der DNA-Analytik mit DNA-Mikroarrays hat gezeigt, dass Informationen über DNA-Sequenzen und deren Häufigkeiten in den Datenbanken, thermodynamische Modelle von Hybridisierung und Sekundärstruktur-Bildung sowie das Wissen über den Umgang mit DNA-Mikroarrays berücksichtigt werden müssen. Die Bewertungsfunktionen leisten einen Teil der Informationsverarbeitung, nämlich die Wandlung/Beschreibung der beteiligten Objekte und Eigenschaften in für Algorithmen greifbare Zahlenwerte, und gehen entweder als Parameter in die Berechnung der →Sensitivität und →Spezifität ein oder werden zusammen mit einem Kriterium in den Optimierungs-Algorithmen, zur Verbesserung der Qualität der Hybridisierungssignale und der Fähigkeit Signale zu diskriminieren, verwendet.

Im folgenden wird für die einzelnen Bewertungsfunktionen die Bedeutung, Berechnung und die mathematischen Eigenschaften analysiert. Daraus wird abgeleitet, in welcher Weise Kriterien auf die Werte der Bewertungsfunktionen angewendet werden, z.B. Anwendung eines scharfen oder fuzzy Grenzwertes oder Maximierung so weit möglich, und mit welcher Gewichtung die Ergebnisse in den Optimierungs-Algorithmus eingehen. Bei den Bewertungsfunktionen für Sekundärstrukturen war zusätzlich der Rechenaufwand ein wichtiger Aspekt für die Art und Weise in der das entsprechende Kriterium zum Einsatz kommt.

In den folgenden Abschnitten werden für die Berechnung der Sensitivität und Spezifität die Werte rp, fn, fp und rn (siehe Tabelle 4-1) definiert. Unter Berücksichtigung der ebenfalls im folgenden eingeführten Hierarchie zwischen →Sequenzklassen wird festgelegt, was →Ziel- und was →Nichtziel-Sequenzen sind, damit werden die beiden Spalten aus der Tabelle 4-1 bestimmt. Für die beiden Zeilen der Tabelle werden unter Berücksichtigung von Bewertungsfunktionen für die Hybridisierungseffizienz (Abschnitt 4.2.1), Sekundärstruktur-Bewertungs-

funktionen (Abschnitt 4.2.3) und →Redundanz- und →Toleranz-Niveau (Abschnitt 4.2.2) Kriterien definiert, die bestimmen, ob ein Oligonukleotid eine Sequenz „trifft“, also ein positives Signal gibt, bzw. ob eine Ziel-Sequenz korrekt – im Sinne des „Set Cover“-Problems – „abgedeckt“ wurde.

Tabelle 4-1: Kennzahlen eines Klassifikators; Vierfeldertafel

	Ziel-Klasse	Nichtziel-Klasse
Signal positiv	richtig-positive: rp	falsch-positive: fp
Signal negativ	falsch-negative: fn	richtig-negative: rn

In den beiden folgenden Abschnitten werden die notwendigen Kriterien für die Definition der Spalten und Zeilen der Tabelle 4-1 bearbeitet. Anschließend wird in Abschnitt 4.3 eine formale Aufgabenspezifikation angeben, die als Grundlage für die in Kapitel 5 vorgestellten Algorithmen dient.

4.1. Definition von Ziel- und Nichtziel-Sequenzen

Wie im Abschnitt 2.5 angekündigt, wird in diesem Abschnitt das „Set Cover“-Problem unter Berücksichtigung hierarchischer Strukturen zwischen den Sequenzklassen um das Kriterium „maximale Spezifität“ erweitert. Bei der Bearbeitung biologischer Fragestellungen, wie der Organismen-Identifikation, sind im Allgemeinen nicht nur zwei Klassen (Ziel-Klasse und Nichtziel-Klasse) vorgegeben, sondern mehr als zwei Gene oder Organismen, die im folgenden Sequenzklassen genannt werden. Für die Berechnung von Sensitivität und Spezifität ist es jedoch notwendig, diese Mehrklassen-Probleme auf ein Zwei-Klassen-Problem zurückzuführen. Die folgenden Abschnitte führen diesen Schritt durch und berücksichtigen dabei die für biologische Fragestellungen typische hierarchische Struktur zwischen den Sequenzklassen.

4.1.1. Hierarchische Struktur zwischen Sequenzklassen

Unter dem Begriff „Sequenzklasse“ werden die Mengen aller →Ziel- oder →Nichtziel-Sequenzen subsumiert. Bei dem bisher eingeführten „Set Cover“-Problem galt es nur eine Menge von Sequenzen zu überdecken, was im Wesentlichen mit der Maximierung von Sensitivität gleichzusetzen ist. Sobald die →Spezifität als Kriterium hinzukommt, gibt es eine Menge von Sequenzen, die möglichst wenig überdeckt werden soll, das heißt, dass möglichst wenig Sequenzen dieser Menge von Oligonukleotiden getroffen werden sollen. Diese Menge wird im folgenden die Menge der Nichtziel-Sequenzen genannt. Bei den von Biologen vorgegebenen Aufgabenstellungen der DNA-Analytik, wie zum Beispiel die Detektion von Genen, Organismen oder der Nachweis der einzelnen Genotypen eines Virus, kommt es nicht selten vor, dass eine Menge von →Sequenzklassen, also eine Menge von Sequenzmengen, vorgegeben wird. Diese hat häufig eine aus der →Phylogenie der betroffenen Organismen abgeleitete hierarchische Struktur. Wollte man eine DNA-Analytik für einige Tierarten, zum Beispiel für Säuger, Fische, Vögel, Spinnentiere und Insekten, entwickeln, so ergäbe sich die in Abbildung 4.1-1 dargestellte hierarchische Struktur unter den Sequenzklassen. Denn Säuger, Fische und Vögel können zu Wirbeltieren zusammengefasst werden, Spinnentiere und Insekten sind Gliederfüßer und Wirbeltiere, Würmer und Gliederfüßer werden dem Tierreich zugeordnet.

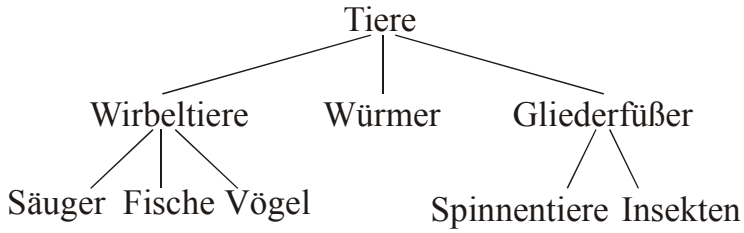


Abbildung 4.1-1: Beispiel einer hierarchischen Struktur zwischen Sequenzklassen

Die Abbildung 4.1-1 soll ausschließlich die biologisch motivierte bzw. aus einer →Phylogenie abgeleitete hierarchische Struktur unter →Sequenzklassen darstellen. Sie ist nur eine sehr grobe unvollständige Darstellung und vor allem selbst keine Phylogenie. Eine solche besteht nur aus Bifurkationen, wie in einem binären Baum, da sich aus einer Art niemals mehrere Arten gleichzeitig entwickelt haben.

Letztendlich werden verschiedene Arten auf der Grundlage einiger (willkürlich) ausgewählter Kriterien zu Gattungen zusammengefasst und diese zu Familien, Ordnungen, Unterklassen, Klassen, Unterstämme, Stämme und Reiche. In der Abbildung 4.1-1 ist der Knoten der „Wirbeltiere“ in dieser zoologischen Systematik der Unterstamm der Wirbeltiere, der Knoten „Säuger“ ist abgeleitet von der Klasse der Säuger, und die Klasse der Insekten gehört zum Stamm der Gliederfüßer und diese, wie alle anderen, zum Reich der Tiere. Man erkennt, dass die Ebenen der dargestellten Hierarchie nicht zwangsläufig mit den Ebenen der zoologischen Systematik übereinstimmen. Diese Übereinstimmung ist bei der Aufstellung einer Hierarchie von Sequenzklassen nicht gefordert.

In der Arbeitsgruppe "Klassifikation und Datenanalyse in den Biowissenschaften" der Gesellschaft für Klassifikation e.V.[23] wird in diesem Zusammenhang von „hierarchischer Klassifikation“ gesprochen:

> „Biologische Taxonomie (BT) und Systematik benötigen Verfahren zur hierarchischen Klassifikation. [...] Die derzeitige Verfügbarkeit umfangreicher molekularer Daten und die explosive Entwicklung entsprechender Datenbanken in den Biowissenschaften verhalf diesem Arbeitsgebiet zu großer aktueller Relevanz und etablierte es als interessanten Anwendungsbereich von Methoden aus Datenanalyse und numerischer Klassifikation.
>
> Die aktuellen Fortschritte auf dem Gebiet der Genomsequenzierung, insbesondere bei mikrobiellen Genomen, führen zu neuen Anwendungen für die Methoden der Datenanalyse und Bioinformatik. Verfahren zur hierarchischen Klassifikation von orthologen und paralogen Genfamilien sind essentiell für den Bereich der Genomanalyse, sowohl in Bezug auf die funktionelle Identifizierung neuer Gene als auch bei Untersuchungen zur Genomevolution.“

Die hierarchische Klassifikation ist zwar nicht Gegenstand dieser Arbeit, das Zitat zeigt jedoch, dass die Berücksichtigung von Hierarchien intrinsisch für biologische Fragestellungen in der DNA-Analytik ist. Eines der Ziele dieser Arbeit ist Oligonukleotid-Bibliotheken so zu optimieren, dass nach einem Hybridisierungs-Experiment aussagekräftige und für eine hierarchische Klassifikation gut separierbare Daten zur Verfügung stehen.

Im folgenden wird dargestellt, wie die zumeist aus einer →Phylogenie abgeleitete Hierarchie von Sequenzklassen bei der Bestimmung der Spezifität eingeht. In dem Kapitel 7 „Anwen-

[23] http://www.gfkl.de/ag_bt.html : AG „Klassifikation und Datenanalyse in den Biowissenschaften“

dungen und Ergebnisse“ wird eine Hierarchie von Sequenzklassen aufgestellt, die aus der phylogenetischen Struktur der Genotypen des Hepatitis C-Virus abgeleitet ist.

4.1.2. Problemanalyse für die Berücksichtigung der Hierarchie

Eine verbal formulierte Aufgabenspezifikation für eine biologische Fragestellungen könnte lauten „Konstruiere für eine vorgegebene Hierarchie von Sequenzklassen G_1, G_2, G_3, ... Oligonukleotid-Teilbibliotheken, die ihre Sequenzklasse möglichst vollständig mit guten Hybridisierungssignalen treffen und unter Berücksichtigung der Hierarchie die übrigen Sequenzklassen möglichst wenig treffen bzw. ein gut zu diskriminierendes negativ-Hybridisierungssignal geben. Weiterhin soll die Sequenzklasse G_0 möglichst wenig getroffen werden.“ Die Aspekte der Signalqualität und Diskriminierungsfähigkeit, mit der letztlich positive und negative Hybridisierungssignale definiert werden, sind Gegenstand des Abschnitts 4.2. In Abschnitt 4.3 wird ein einer formalen Aufgabenspezifikation die Berücksichtigung der Hierarchie, in einem, für alle drei in Kapitel 5 vorgestellten Optimierungs-Algorithmen, gültigen Rahmen-Algorithmus angegeben.

Die Abbildung 4.1-2 veranschaulicht ein Beispiel einer Hierarchie als Baumstruktur[24] und als Venn-Diagramm. In dem Venn-Diagramm ist die Beziehung „Kind-Knoten → Eltern-Knoten“ als Teilmengen-Beziehung dargestellt. Der Teilbaum „G_3 → G_2 → G_1“ wird somit auf $G_3 \subset G_2 \subset G_1$ abgebildet. Die „Kante →“ zwischen Kind-Knoten und Eltern-Knoten in der Baumstruktur stellt eine „ist ein“-Beziehung dar. D.h. der Teilbaum „Spinnentier → Gliederfüßer → Tier“ aus der Abbildung 4.1-1 steht für die Aussage „Ein Spinnentier *ist ein* Gliederfüßer *ist ein* Tier“. Die Mengen-Darstellung in dem Venn-Diagramm hingegen beruht darauf, dass die Elemente der Mengen G_0, ..., G_9 Teilsequenzen aus dem →Genom der entsprechenden Organismen sind. Solche Teilsequenzen, die nur G_1 oder z.B. den Spinnentieren zuzuordnen sind, sind deshalb nur Element der Menge G_1 bzw. „Spinnentiere“. Teilsequenzen, die hingegen allen Wirbeltieren gemein sind, sind Element der Menge „Wirbeltiere“.

Man erkennt in dem Venn-Diagramm, dass die zu einem Knoten nächsttieferen Hierarchie-Stufen nicht notwendigerweise eine Partition dieses Knotens sein müssen. D.h. $G_3 \cup G_4$ kann eine echte Teilmenge von G_2 sein bzw. $G_2 \setminus (G_3 \cup G_4)$ kann eine nicht-leere Menge sein. Ein Grund aus einer „realen biologischen Fragestellung“ dafür könnte sein, dass es Ziel-Sequenzen in G_2 gibt, die nicht weiter in G_3 oder G_4 eingeordnet waren. Es gibt zwei Gründe für die Existenz solcher Ziel-Sequenzen:

1. Diese Ziel-Sequenzen entsprechen zwar z.B. dem Genotyp eines Virus, bei der Annotation, der für den Eintrag in die internationalen Sequenzdatenbanken detaillierten Beschreibung aller Merkmale der Sequenz, wurde diese Ziel-Sequenz jedoch nicht eindeutig einem Subtyp zugeordnet.
2. Bei dem Aufbau eines phylogenetischen Baumes muss jedes Individuum bzw. jede Art einem Blatt des Baumes zugeordnet werden können, keines einem inneren Knoten. D. h. es gibt kein Individuum mit der Bezeichnung „Fisch“ oder „Eukaryot“. Ein kleinerer Teil eines Genoms kann jedoch z.B. allen Wirbeltieren oder allen Eukaryoten gemein sein und damit einem inneren Knoten in einem phylogenetischen Baum zugeordnet werden. Se-

[24] Da die in der Abbildung 4.1-2 dargestellte Hierarchie zusammen mit G_0 als Graph aus mehreren Zusammenhangskomponenten besteht und letztendlich neben der Wurzel G_1 auch weitere Wurzeln für weitere Bäume zugelassen sind, müsste an dieser Stelle streng genommen der Begriff „Wald“ [114], [41] oder der des „azyklischen Graphen“ verwendet werden. Ein Wald ist ein nichtzusammenhängender Graph, in dem jede Zusammenhangskomponente ein Baum ist. Ein Baum ist ein zusammenhängender Graph ohne Kreise.

quenzen, die diesem Teil des Genoms entnommen sind, sind demzufolge der Menge $G_2 \setminus (G_3 \cup G_4)$ zuzuordnen.

Weiterhin zeigt das Venn-Diagramm, dass es →Sequenzen geben kann, die nicht in G_0 oder G_1 enthalten sind. Zur Erinnerung sei gesagt, dass G_1 die größte Menge der →Ziel-Sequenzen ist und dass die Sequenzklasse G_0 möglichst wenig getroffen werden soll. Die Sequenzen außerhalb von G_0 und G_1 entsprechen somit den Sequenzen, die man nicht in der Probe einer DNA-Analyse vermutet (z.B. DNA-Sequenzen von Pflanzen in einer Blutprobe) und gegen die nicht notwendigerweise diskriminiert werden muss.

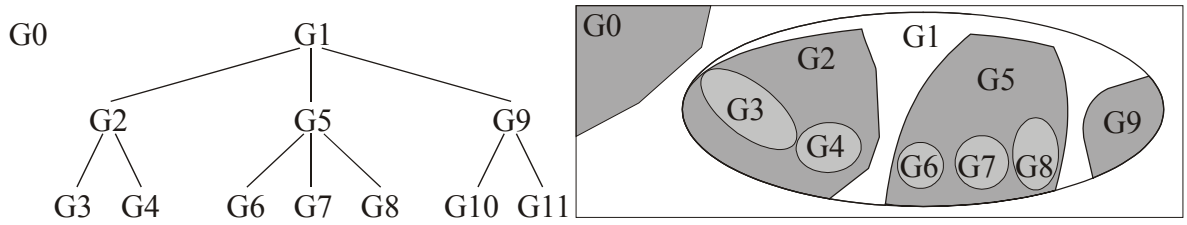


Abbildung 4.1-2: Hierarchie von Sequenzklassen G_1, G_2, G_3, ... und eine Menge G_0 von nicht zu treffenden Sequenzen

4.1.3. Formale Spezifikation für die Berücksichtigung der Hierarchie

Im folgenden wird die im letzten Abschnitt angegebene *verbale Aufgabenspezifikation* in einen formalen Rahmen gebracht. Dabei wird eine, die vorgegebene Hierarchie einer biologischen Problemstellung berücksichtigende, Berechnungsvorschrift für →Spezifität entwickelt und die gesamte Aufgabenstellung auf mehrere Teilprobleme reduziert.

Gegeben sei eine Menge von Sequenzklassen G_i zusammen mit einer Hierarchie. Die Hierarchie aus Abbildung 4.1-2 wird in Klammernotation mit (G_0, G_1 (G_2(G_3, G_4),G_5(G_6, G_7, G_8), G_9(G_{10}, G_{11}))) angegeben. Für eine simple Aufgabenstellung, bei der in einer Menge von Sequenzklassen „jeder gegen jeden" abgegrenzt werden soll, wäre eine einfache Liste (G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9, G_{10}, G_{11}), die man auch als flache Hierarchie bezeichnen könnte, die korrekte Darstellung. Mit g=11 wird die Anzahl der Sequenzklassen bezeichnet. Es werden g Oligonukleotid-Teilbibliotheken konfiguriert; die Sequenzklasse G_0 wird nicht mitgezählt, da diese eine Teilmenge der →Nichtziel-Sequenzen ist.

In Anlehnung an den Abschnitt 2.5 „Optimierung von Oligonukleotid-Bibliotheken" wird hier eine Menge von Ziel-Sequenzen M´ definiert, die alle Sequenzklassen G_i mit $i \neq 0$ umfasst. Die Menge M´ kann als Vereinigung aller G_i mit $i \neq 0$ gebildet werden, und ist damit identisch zur Vereinigung aller Wurzeln der Hierarchie:

$$M' := \bigcup_{i \neq 0} G_i = \bigcup_{i \neq 0, i \in W} G_i \quad ; \text{mit} \quad W = \text{<Menge aller Wurzeln der Hierarchie>}$$

Die Menge aller Oligonukleotide K bilde die Menge P als Teilmenge der Potenzmenge von M´ über das Bild der Menge K unter der Abbildung $Match: K \rightarrow \wp(M)$. Die Abbildung *Match* ordnet, wie auch schon im Abschnitt 2.5, jedem Oligonukleotid $x \in K$ die Teilmenge von M´ zu, die der Menge der →Treffer des Oligonukleotids auf den Ziel-Sequenzen entspricht:

$$P := Match(K) \subset \wp(M) \text{ , d.h. für ein Oligonukleotid } x \in K \text{ gilt } Match(x) \subset M'$$

Die Treffermenge $Match(x) \in P$ eines Oligonukleotids $x \in K$ wird also definiert als

$$Match(x) := \{ t \in M' \mid \text{„das Oligonukleotid x trifft die Ziel-Sequenz t"} \}$$

In Abschnitt 4.2 wird über die Sequenz-Differenz zwischen Fänger-Oligonukleotid x und Ziel-Sequenz t oder mit Hilfe von thermodynamischen Größen genau definiert, was ein Treffer ist. Hier kann zunächst die Übereinstimmung der Sequenz des Oligonukleotids x als Zeichenkette an einer Position auf der Ziel-Sequenz t als Treffer angenommen werden.

Hinter der verbalen Aufgabenspezifikation verbergen sich mehrere, nämlich g, „Set Cover"-Probleme. Im Abschnitt 2.5 wurden die „Set Cover"-Probleme mit dem Tupel (M, P) bezeichnet, was in dem Kontext der in Abbildung 4.1-2 dargestellten Hierarchie mit dem Teilproblem für die Wurzel $M=G_1$ übereinstimmt. Allgemein gilt für alle g Knoten G_i (i = 1, ..., g) der Hierarchie, dass sie die „Set Cover"-Probleme (G_i, P) bilden. Gesucht sind Oligonukleotid-Teilbibliotheken $L_i \subset P \subset \wp(M)$, sodass

$$\bigcup_{x \in L_i} x = G_i \quad \text{oder zumindest} \quad \bigcup_{x \in L_i} x \quad \text{möglichst viele Elemente aus } G_i \text{ enthält}$$

Bis zu diesem Punkt der Aufgabenspezifikation werden g Sensitivitäten maximiert. Zu der erwähnten Menge von „Set Cover"-Problemen kommt nun erschwerend das Kriterium der Spezifität hinzu. Die Spezifität ist definiert als das Verhältnis der richtig-negativen zu der Summe der richtig-negativen und falsch-positiven (siehe Abbildung 4.1-3/oben). Bei mehr als zwei Sequenzklassen, die sich z.B. in der flachen Hierarchie (G_1, G_2, G_3, G_4) alle gegeneinander abzugrenzen haben, wird für jedes $i \in \{1, 2, 3, 4\}$ die „Spezifität der Oligonukleotid-Teilbibliothek L_i für die Klasse G_i" definiert. In der Abbildung 4.1-3/unten ist dargestellt, wie in dem Fall *i=2* die richtig-negativen und falsch-positiven definiert sind. In diesem Fall ist jedes G_j mit $j \neq i$ eine Sequenzklasse, die nicht getroffen werden sollte. Das heißt, dass L_1 nicht G_2 ,G_3 und G_4 treffen darf, L_2 nicht G_1, G_3 und G_4 und so weiter.

	Ziel-Klasse	Nichtziel-Klasse
Signal positiv	richtig-positive	falsch-positive
Signal negativ	falsch-negative	richtig-negative

	G_1	G_2	G_3	G_4
Signal positiv	fp_1	rp_2	fp_3	fp_4
Signal negativ	rn_1	fn_2	rn_3	rn_4

Abbildung 4.1-3: Treffer-Tabellen für zwei und am Beispiel G_2 für mehr Klassen
rp = Anzahl der richtig-positiven; fn = Anzahl der falsch-negativen;
rn = Anzahl der richtig-negativen; fp = Anzahl der falsch-positiven

Beispiel für die Berechnung der Spezifität der Oligonukleotid-Teilbibliothek L_2 für die Sequenzklasse G_2 in der flachen Hierarchie (G_1, G_2, G_3, G_4):

$$\text{Spez}(L_2) = \frac{\sum rn}{\sum rn + \sum fp} = \frac{rn1 + rn3 + rn4}{rn1 + rn3 + rn4 \; + \; fp1 + fp3 + fp4}$$

Für nicht-flache Hierarchien, wie in der Abbildung 4.1-2 dargestellt, muss ebenfalls für jede Oligonukleotid-Teilbibliothek L_i definiert werden, welche Ziel-Sequenzen $G_i \subset M'$ nicht getroffen werden dürfen. Für G_2, G_5 und G_9, die erste Ebene der Baumstruktur, verhält es sich wie für eine flache Hierarchien und alle Sequenzklassen müssen sich gegeneinander abgrenzen. G_3 jedoch ist eine Teilmenge von G_2 und diese wiederum eine von G_1, sodass sich L_2 nicht gegen G_1 und L_3 nicht gegen $G_2 \cup G_1$ abgrenzen muss.

Umgekehrt muss sich L_2 nicht gegen $G_3 \cup G_4$ abgrenzen, da G_3 und G_4 Teilmengen von G_2 sind, sodass der gesamte „vertikale Pfad" des Baumes von einem Knoten ausgehend zu allen

Blättern und zur Wurzel nicht zur Nichtziel-Klasse gehört. Die Abbildung 4.1-4 veranschaulicht die vertikalen Pfade für die Sequenzklassen G_4 (links) und G_5 (rechts).

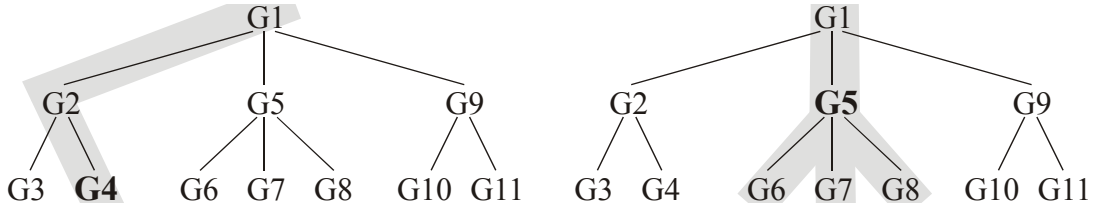


Abbildung 4.1-4: Die „vertikalen Pfade" für die Sequenzklassen G_4 und G_5

Demzufolge muss sich die Oligonukleotid-Teilbibliothek L_4 gegen $G_3 \cup G_5 \cup G_9$ abgrenzen und L_5 gegen $G_2 \cup G_9$. Die Sequenzklassen G_{10} und G_{11} tauchen hier nicht auf, da sie Teilmengen von G_9 sind. Somit kann für jede Oligonukleotid-Teilbibliothek L_i einer Hierarchie die Nichtziel-Klasse, als Vereinigung der G_i, die nicht getroffen werden dürfen, bestimmt werden. Damit ist die Anzahl der falsch-positiven und richtig-negativen bekannt, und die Spezifität der Oligonukleotid-Teilbibliothek L_i für die Klasse G_i kann nach der oben angegebenen Berechnungsvorschrift bestimmt werden.

Das „Set Cover"-Problem, welches im Abschnitt 2.5 für jede Menge M und jedes $P \subset \wp(M)$ mit dem Tupel (M, P) eingeführt wurde, kann nach der Definition einer Nichtziel-Klasse A_i für jede Sequenzklasse G_i (i=1, ..., g) formal zu einem „Set Cover"-Problem mit Spezifitäts-Nebenbedingung (M, P, A) erweitert werden. Dabei wird mit K_i gleich der Menge der aus G_i ableitbaren Oligonukleotide (M, P, A) := (G_i , *Match*(K_i), A_i) gesetzt. Die Nebenbedingung wird im folgenden Abschnitt in die Bewertungsfunktionen des Greedy-Algorithmus, des Genetischen Algorithmus und in den Ansatzes mit dem Gradientenabstiegs-Verfahrens integriert.

Es wäre möglich das (M, P, A)-Problem auf ein wie in Abschnitt 2.5 definiertes (M´´, P´´)-„Set Cover"-Problem zurückzuführen, indem man $M'' = M \cup A$ setzt und für die Menge A die Treffer der Oligonukleotide invertiert, d.h. jeder Treffer in A wird als nicht-Treffer gewertet und umgekehrt. Die Treffermenge *Match*(x) eines Oligonukleotids $x \in K$ wäre in dem Fall definiert als

Match(x) := { $t \in M'' = M \cup A$ | „$t \in M$ und das Oligonukleotid x trifft die Ziel-Sequenz t"
oder „$t \in A$ und das Oligonukleotid x trifft **nicht** die Sequenz t" }

Somit hätte man eine neu definierte Menge P´´ der für das „Set Cover"-Problem zugelassenen Teilmengen von M´´

$$P' := Match(K) \subset \wp(M'')$$

und ein spezifisches Oligonukleotid des (M, P, A)-Problems wäre damit ein sehr sensitives für die Teilmenge A aus M´´ in dem zugehörigen (M´´, P´´)-Problem. Dieser Ansatz wurde jedoch nicht weiter verfolgt, da der Greedy-Algorithmus nach mehreren Iterationen das wiederholte Treffen von Sequenzen in A nicht bestrafen würde. Zwar könnte dieses mit einem auf A großen →Redundanz-Niveau erreicht werden, der Nutzen einer flexiblen Gewichtung zwischen Sensitivität und Spezifität (vgl. Abschnitt 8.2 zu ROC-Curves) und die Einführung von →Redundanz- und →Toleranz-Niveau (in Abschnitt 4.2.2) wird in dieser Arbeit jedoch höher bewertet.

In diesem Abschnitt wird eine zuvor verbal spezifizierte Aufgabenstellung analysiert und auf eine Menge von „Set Cover"-Teilproblemen mit Spezifitäts-Nebenbedingung reduziert. In dem Kapitel zu den Optimierungs-Algorithmen ist daher, unter dem Gesichtspunkt Sensitivität für die Ziel-Sequenzen und Spezifität gegenüber den →Nichtziel-Sequenzen zu maximie-

ren, nur noch eine Oligonukleotid-Teilbibliothek für jedes dieser Teilprobleme zu erstellen. Die einzelnen Probleme dürfen jedoch nicht unabhängig voneinander betrachtet werden, da unter dem Gesichtspunkt der Signalqualität und Diskriminierungsfähigkeit, der im folgenden Abschnitt behandelt wird, nach wie vor Abhängigkeiten zwischen den Oligonukleotid-Teilbibliotheken - z.B. über die Schmelztemperatur - bestehen.

4.2. Definition und Vorhersage von „positiven und negativen Signalen“

Nach der Definition von →Ziel- und →Nichtziel-Sequenzen in Abschnitt 4.1 wird hier untersucht und definiert, wann ein Fänger-Oligonukleotid $x \in K$ eine Ziel-Sequenz $t \in M$ trifft, d.h. mit der Ziel-Sequenz hybridisiert und somit ein Hybridisierungssignal auf dem entsprechenden Spot auf dem DNA-Mikroarray detektiert werden kann. Damit ist die Abbildung $Match: K \rightarrow \wp(M)$ des vorigen Abschnitts vollständig definiert, und die Qualität bzw. der Nutzen eines Oligonukleotids bzgl. Sensitivität und Spezifität kann, unter Berücksichtigung der Hierarchie und der hier betrachteten Hybridisierungs-Eigenschaften, berechnet werden. Dafür benötigen wir die Nukleinsäuresequenzen der Oligonukleotide $x \in K$ und der Ziel-Sequenzen $t \in M$ und definieren dazu:

> $K, M \subset B^*$ mit $B = \{A, C, G, T\}$ mit B^* als der Menge der Zeichenketten über dem Alphabet über B.
>
> Weiterhin sei $| \ | : B^* \rightarrow \mathbb{N}$ die Abbildung, die einer Zeichenkette seine Länge zuordnet, dann ist z.B.: x = “GGTATGGCTATGCTAGG“ $\in$ K, $|x| = 17$ oder $|t| = 300$ für eine lange Ziel-Sequenz mit 300bp.

In den folgenden Abschnitten werden alle Eigenschaften der Oligonukleotide auf Sequenz-Ebene wie auch auf der Ebene der →thermodynamischen und →kinetischen Modelle der Hybridisierung und Sekundärstruktur-Bildung betrachtet. Diese Eigenschaften werden durch Bewertungsfunktionen quantifiziert und bilden so eine grobe Vorhersage der Hybridisierungseffizienz bzw. des Hybridisierungssignals. In [102] heißt es zum Thema „*hybridization prediction*“ im Zusammenhang mit dem Design von →Antisense-Oligonukleotiden „no way currently exists to know a priori which sites in the →mRNA molecule should be targeted“. Mit empirisch ermittelten Daten wird bestimmt, welche Intensität eines Hybridisierungssignals noch als ein positives Signal bewertet wird. Daraus werden für die in dem folgenden Vorhersage-Modell benutzten Grenzwerte für Bewertungsfunktionen abgeleitet. Diese definieren, was auf der Seite des Modells ein „vorhergesagt-positives“ oder ein „vorhergesagt-negatives“ Signal ist.

4.2.1. Bewertungsfunktionen für die Hybridisierungseffizienz

Der noch in Abschnitt 4.1.3 zugrundegelegte Begriff eines „Treffers eines Oligonukleotids $x \in K$ auf einer Ziel-Sequenz $t \in M$“, nämlich die Übereinstimmung der Sequenz des Oligonukleotids x als Zeichenkette an einer Position auf der Ziel-Sequenz t, ist ein sehr grobes Modell des Hybridisierungs-Prozesses. In Abhängigkeit von der Sequenzlänge, Schmelztemperatur und den Temperaturen, mit denen während Durchführung des →Hybridisierungsprotokolls hybridisiert und gewaschen wird, kommt es häufig zu Hybridisierungssignalen auch wenn das Fänger-Oligonukleotid die Ziel-Sequenz nicht vollständig, d.h. als Teilsequenz des *reverse-complement*, trifft (*engl.: perfect match*). Solche nicht vollständigen Hybridisierungen werden Mismatch-Hybridisierungen (→Mismatch) oder Hybridisierungen mit einem oder mehreren Basenaustauschen oder Basenfehlpaarungen genannt. Handelt es sich um eine

Hybridisierung mit einer →Nichtziel-Sequenz, so wird sie „unspezifische Hybridisierung“[25] genannt. D.h. in Abschnitt 4.1.3 wurde ein positives Signal mit der „*perfect match*“-Hybridisierung und ein negatives Signal mit der Mismatch-Hybridisierung gleichgesetzt. Dieser nur für die Zwecke einer vereinfachten Einführung verwendete Ansatz wird im folgenden schrittweise verfeinert und damit dem Hybridisierungs-Prozess angenähert.

In einer ersten Verfeinerung wird die Anzahl der Basenfehlpaarungen (engl.: *mismatches*) für jedes Paar $(x, t) \in K \times M$ bestimmt und es kann beispielsweise definiert werden, dass ein solches Paar mit keinem oder einem Basenaustausch einem positiven Hybridisierungssignal entspricht. Dieses Zählen von verschiedenen Komponenten zwischen zwei Zeichenketten oder Vektoren ist der Hamming-Distanz sehr ähnlich, die allerdings nur auf gleich lange Zeichenketten bzw. Sequenzen angewendet werden kann. Leicht definiert man sich mit Hilfe der Hamming-Distanz eine verallgemeinerte und für die hier benötigen Zwecke angemessene Abbildung für die Fälle unterschiedlich langer Sequenzen $|x| \neq |t|$:

$$H: B^* \times B^* \rightarrow \mathbb{N}$$

$$H(x, t) := H(t, x) := \min\{ \text{Hamming}_{|x|}(x, t') \mid t' \text{ ist Teilsequenz von t der Länge } |x| \}$$

mit $\text{Hamming}_n: B^n \times B^n \rightarrow \mathbb{N}$ als dem normalen Hamming-Abstands;
ohne Beschränkung der Allgemeinheit sei hier $|x| \leq |t|$, d.h. x kürzer als t.

Diese Abbildung ist zwar symmetrisch, aber die positive Definitheit und die Dreiecksungleichung gelten nicht, deshalb ist H keine Abstands-Funktion. Eingeführt wurde sie als Bewertungsfunktion für die potentielle Hybridisierungseffizienz des Oligonukleotids x bzgl. der Ziel-Sequenz t. Die Hybridisierungseffizienz bzw. die Intensität eines Hybridisierungssignals ist dabei größer für kleinere Werte H(x, t) und umgekehrt, da der Fall H(x, t) = 0 eine „*perfect match*“-Hybridisierung mit der größtmöglichen Hybridisierungseffizienz darstellt.

Mit einem Grenzwert g = 1 und dem Grenzwert-Kriterium $H(x, t) \leq g$ kann nun die oben bereits erwähnte Definition eines positiven Hybridisierungssignals formal angegeben werden. Eine mögliche Definition der Abbildung $Match: K \rightarrow \wp(M)$ aus Abschnitt 4.1.3 wäre jetzt:

$$Match(x) := \{ t \in M \mid H(x, t) \leq g \}$$

Die Tabelle 4.2-1 stellt das Grenzwert-Kriterium $H(x, t) \leq g$ in einer der Abbildung 4.1-3/oben ähnlichen Tabelle dar. Die zwei Zeilen, nämlich „Signal positiv“ und „Signal negativ“, aus Abbildung 4.1-3/oben wurden hier verfeinert zu den Klassen 0, 1, 2 und „≥ 3“, welche der Anzahl der Basenfehlpaarungen entsprechen. Oberhalb des fett/rot hervorgehobenen Balkens befinden sich die positiven Signale, darunter die negativen. Die Anzahl der „richtig-positiven“ ist damit rp0 + rp1, die Anzahl der „falsch-negativen“ ist fn2 + fn3 und so weiter.

Tabelle 4.2-1: Grenzwert-Kriterium für die Anzahl von Basenfehlpaarungen

H(x, t)	Ziel-Klasse	Nichtziel-Klasse
0	rp0	fp0
1	rp1	fp1
2	fn2	rn2
≥ 3	fn3	rn3

← Grenzwert-Kriterium: „$H(x, t) \leq g = 1$“

[25] Die „unspezifische Hybridisierung“ sollte nicht mit einer „unspezifischen Bindung“ oder einem „unspezifischen Hybridisierungssignal“ verwechselt werden, da diese ohne Hybridisierung auch durch Anlagerung auf der Oberfläche des DNA-Mikroarrays zustande kommen können.

Dieser Grenzwert bestimmt die „vorhergesagt-positiven“ bzw. „vorhergesagt-negativen“ Signale und hat damit als Kriterium für die Auswahl von Oligonukleotiden einen großen Einfluss auf den folgenden Entwicklungsprozess der Oligonukleotid-Bibliothek. Erstens stellt dieses Kriterium thermodynamische Realität dar, denn für sehr lange Fänger auf dem DNA-Mikroarray (z.B. bei Genexpressions-Experimenten mit →cDNA, der Ansatz in [52] mit 50-meren oder der von Operon mit 70meren) können durchaus auch 10 oder mehr Basenfehlpaarungen zu einem positiven Hybridisierungssignal führen. Zweitens bewirkt der Schritt von „$H(x, t) \leq 0$“ zu „$H(x, t) \leq 1$“, mit dem *fn*1 zu *rp*1 umdefiniert wird, dass es der Optimierungs-Algorithmus leichter hat, die Sensitivität zu maximieren, da dieser mehr Oligonukleotide zur Verfügung hat.

$$\{ x \in K \mid H(x, t) \leq 0 \} \subseteq \{ x \in K \mid H(x, t) \leq 1\}$$

Wie in der Tabelle 4.2-1 zu erkennen ist, erhält man auf diese Weise eine größere Sensitivität:

$$\frac{rp0 + rp1}{(rp0 + rp1) + (fn2 + fn3)} \geq \frac{rp0}{rp0 + (fn1 + fn2 + fn3)} \quad \text{, mit } fn1 = rp1$$

Auf der anderen Seite bewirkt das oben definierte Grenzwert-Kriterium mit dem gleichen Argument, dass sich die →Spezifität verringert, da *rn*1 zu *fp*1 umdefiniert wird. Dieses entspricht tatsächlich der Realität, wenn man längere Fänger-Oligonukleotide nimmt, z.B. von 30meren zu 70meren wechselt. Diese geben auch bei mehr Basenfehlpaarungen noch ein positives Hybridisierungssignal und sind damit weniger spezifisch. Dieser Zusammenhang und seine Auswirkungen auf das Design von Oligonukleotid-Bibliothek wird in Abschnitt 4.2.4.1 erneut aufgegriffen.

Wie oben bereits gesagt, ist die Vorhersage der Hybridisierungseffizienz bzw. des Hybridisierungssignals, auf jeden Fall mit dem hier betrachteten sehr groben Ansatz, aber auch noch mit den im folgenden weiter verfeinerten thermodynamischen Modellen, zur Zeit noch sehr schwierig und fehlerhaft (siehe oben [102]). Zumal noch weitere Faktoren, wie die Sekundärstruktur des Fängers und der Ziel-Sequenz, einen großen Einfluss haben, bedeutet dies, dass ein „vorhergesagt-positives“ Signal auf dem DNA-Mikroarray häufig nicht positiv ist bzw. „vorhergesagt-negatives“ Signal häufig nicht negativ. Dieser Abweichung zwischen Vorhersage und tatsächlichem Signal soll mit der in Tabelle 4.2-2 dargestellten Modifikation des Grenzwert-Kriteriums Rechnung getragen werden.

Tabelle 4.2-2: Grenzwert-Kriterium mit „Sicherheitsabstand“ $g_N - g_Z = 2$

	Ziel-Klasse	Nichtziel-Klasse
0 MM = PM	rp0	fp0
1 MM	fn1	fp1
2 MM	fn2	fp2
≥ 3 MM	fn3	rn3

Dazu werden für die linke Spalte, die Ziel-Klasse, und für die rechte Spalte, die Nichtziel-Klasse, verschiedene Grenzwerte g_Z und g_N eingeführt. Dabei soll g_Z oberhalb des bisher geschätzten Grenzwertes g zwischen positiven und negativen Signalen liegen und g_N unterhalb, also z.B. $g_Z = 0 \leq g = 1 \leq 2 = g_N$. Man erkennt, dass es der Optimierungs-Algorithmus gegenüber der Situation in Tabelle 4.2-1 schwieriger haben wird, die Sensitivität und Spezifität zu maximieren, denn für beide Quotienten wurde durch diese Modifikation der Zähler und damit auch wegen des unveränderten Nenners der Quotient verringert.

Dadurch wird zwar die Anzahl der verwendbaren Oligonukleotide reduziert, man gewinnt jedoch eine Art „Sicherheitsabstand“ zwischen den „vorhergesagt-positiven“ und den „vorhergesagt-negativen“ Signalen. Es wird die Fähigkeit verbessert, zwischen Ziel- und Nichtziel-Sequenzen zu diskriminieren, denn die auf dem DNA-Mikroarray gemessenen Hybridisierungssignale, der mit diese Ansatz gefundenen Oligonukleotide, werden sich stärker und mit größerer Sicherheit unterscheiden. Hauptsächlich die mittlere Länge der Fänger-Oligonukleotide und der „Abstand“ zwischen den Sequenzen der Ziel- und der Nichtziel-Klasse, also deren Separierbarkeit, werden darüber bestimmen, ob und mit welcher Differenz g_N - g_Z man es sich leisten kann dieses Grenzwert-Kriterium einsetzen zu können.

Bevor im folgenden der thermodynamische Ansatz vorgestellt wird, soll an dieser Stelle erwähnt werden, dass es zahlreiche weitere Verfeinerungen für Bewertungsfunktionen der erwarteten Hybridisierungssignale gibt, die ausschließlich mit der Information der Basensequenzen arbeiten. Die Hamming-Distanz wurde zur h-Distanz [31] modifiziert und es wird häufig auch die Position und Anordnung der Basenfehlpaarungen berücksichtigt. Dieser Ansatz wird als „mismatch geometries“ [39] beispielsweise beim „word design“ [68] für DNA-basierte Computer systematisch für eine möglichst große Menge maximal unterschiedlicher „Wörter“ auf einer vorgegebenen Sequenzlänge ausgenutzt.

Nicht nur die Position und Anordnung der Basenfehlpaarungen sondern auch welches Basenpaar A•T oder G•C in welche Basenfehlpaarung übergeht, kann berücksichtigt werden. Die sehr grobe „Wallace Regel“ [113] zur Berechnung der Schmelztemperatur $T_M = 2\,\#[AT] + 4\,\#[GC]$ deutet bereits an, dass der Wegfall einer G•C-Paarung die Schmelztemperatur und damit auch das Hybridisierungssignal mehr reduzieren wird als eine A•T-Paarung. Es ist denkbar mit diesen Informationen, zahlreiche Bewertungsfunktionen mit verschiedenen Rechenaufwänden und verschiedener Detailliertheit zu definieren. Mit etwas mehr Anstrengung ist man jedoch bereits bei den thermodynamischen Modellen zum Hybridisierungs-Prozess angelangt, von dem, nach dem Stand der Wissenschaft, die besten Vorhersagen zur Hybridisierungseffizienz zu erwarten sind. Seit Breslauer 1986 [16] wurden die, für diese Modelle benötigten, thermodynamischen Parametersätze häufig aktualisiert [22], [105], [121], [96], [95], und die Modelle selbst und die Parametersätze sind weiterhin Gegenstand intensiver Forschung[26].

Im folgenden werden die Zeilen aus Abbildung 4.1-3/oben und Tabelle 4.2-2, die letztendlich eine Art diskrete y-Achse darstellen, zu einer als reelle y-Achse dargestellten Differenz zweier thermodynamischer Größen verallgemeinert, die ebenfalls eine Bewertungsfunktion der Hybridisierungseffizienz zwischen einem Oligonukleotid $x \in K$ und einer Ziel-Sequenz $t \in M$ darstellt. Im Allgemeinen wird das Oligonukleotid x einige Basenfehlpaarungen zu einer Teilsequenz aus t haben. Es wird nun ein Oligonukleotid x´ konstruiert, das exakt komplementär zu der aus t ist. Eine mögliche Bewertungsfunktion für die Hybridisierungseffizienz ist dann:

$$\text{thdist: } B^* \times B^* \to \mathbb{R}^+ \text{ oder } K \times M \to \mathbb{R}^+$$

$$\text{thdist}(x, t) = \Delta G(x, t) - \Delta G(x', t)$$

Dabei ist ΔG: $B^* \times B^* \to \mathbb{R}$ eine Abbildung, die mit dem Funktionswert $\Delta G(x, t)$ die freie Energie ΔG der Hybridisierung von x an t an der bestmöglichen Position darstellt. Die freie Energie ΔG ist für sehr stabile Hybridisierungen eine betragsmäßig große negative Zahl. Da die „perfect match“-Hybridisierung von x´ an t immer stabiler ist als eine ähnliche Hybridisie-

[26] Professor John SantaLucia hat Anfang 2001 die Firma DNAsoftware gegründet. Dort wird das Programm HyTher [45] und die zugrundeliegenden thermodynamischen Parameter weiterentwickelt und vermarktet.

rung mit Basenfehlpaarungen, ist die Differenz thdist(x, t) = ΔG(x, t) - ΔG(x´, t) ≥ 0. Ist x selbst bereits ein exaktes Komplement zu einer Teilsequenz aus t, so ist x´ = x und thdist(x, t) = 0. Andere ähnlich konstruierte Bewertungsfunktionen sind denkbar, es kommt hier jedoch ausschließlich auf die Idee an, mit Hilfe von thermodynamischen Größen eine im Verhältnis zu H(x, t) exaktere Bewertungsfunktion konstruiert zu haben.

Die Werte von thdist(x, t) sind in der Abbildung 4.2-1 für alle betrachteten Oligonukleotide auf der mit „thdist(x, t)“ beschrifteten Achse aufgetragen. Die Abbildung ist soweit möglich an den Aufbau der Tabelle 4.2-2 angelehnt. Die beiden Spalten für die Ziel- und die Nichtziel-Klasse sind hier zwei auf der Seite liegende xy-Diagramme. Von oben nach unten, auf der Achse „thdist(x, t)“, nimmt der Abstand zwischen dem Oligonukleotid x und der Sequenz t zu, also die Anzahl der Basenfehlpaarungen oder hier die Differenz der oben betrachteten freien Energien. Von links nach recht, auf der mit „#Oligos“ beschrifteten Achse, ist die Anzahl der Oligonukleotide aufgetragen, die zu der Menge der Sequenzen in der Ziel- bzw. der Nichtziel-Klasse einen entsprechenden thdist-Wert annehmen.

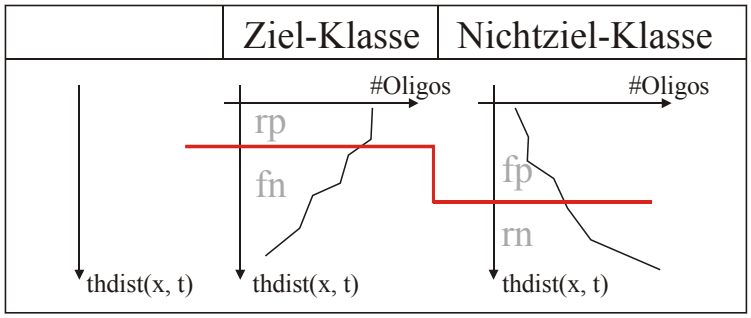


Abbildung 4.2-1: Grenzwert-Kriterium auf thdist(x, t)

Die zwei Funktionsverläufe, auf der Seite liegende schematisch dargestellte Histogramme, stellen recht optimistisch die Situation für eine Menge von Oligonukleotid-Kandidaten dar, die sich aus gut separierbaren →Sequenzklassen ergeben. Ebenfalls angelehnt an die Tabelle 4.2-2 ist hier das Grenzwert-Kriterium eingezeichnet und die sich daraus ergebenden Anzahlen von „richtig-positiven“ rp, „falsch-negativen“ fn, fp und rn. Sie sind als Fläche unter den Funktionsverläufen angedeutet. Mit zwei, ähnlich den oben eingeführten, Grenzwerten g_Z und g_N lautet das Grenzwert-Kriterium hier:

Für alle t aus der Ziel-Klasse gibt (x, t) ein positives Signal, wenn thdist(x, t) ≤ g_Z
Für alle t aus der Nichtziel-Klasse gibt (x, t) ein positives Signal, wenn thdist(x, t) ≤ g_N

In Abhängigkeit von der Komplexität/Detailliertheit der Berechnungsvorschrift von ΔG(x, t) werden bei diesem Ansatz neben den Parametersätzen für die „*nearest neighbor interactions*“ auch die Anzahl, Positionen, Anordnung und Typ von Basenfehlpaarungen berücksichtigt. Das Programm HyTher [45] lässt es beispielsweise zu „ΔG correction terms“ für den „bottum strand“, also dem Fänger auf dem DNA-Mikroarray, und den „top strand“, der Ziel-Sequenz, anzugeben. Diese werden z.B. mit mfold als freie Energien der Sekundärstruktur des Fängers und der Ziel-Sequenz ermittelt. Somit gehen Informationen über die Sekundärstruktur mit in die Berechnung der rp, fn, rn und fp und letztendlich in die Sensitivität und Spezifität mit ein.

Die Berücksichtigung der Sekundärstruktur auf so grundlegender Ebene im Optimierungs-Algorithmus ginge einigen Molekularbiologen sicherlich zu weit, da unter den Wissenschaftlern der Chemie und →Molekularbiologie bis heute Uneinigkeit über die Art und Weise des Einflusses von Sekundärstrukturen auf die Hybridisierungseffizienz herrscht [19], [23], [75], [77], [80], [100], [102]. Ein →Treffer auf Sequenz-Ebene eines unspezifischen Oligonukleotids gegenüber der Nichtziel-Sequenzklasse, die nicht mitgerechnet wird, weil die Bewer-

tungsfunktion aus Gründen der Sekundärstruktur ein negatives Hybridisierungssignal vorhersagt, würde bei diesem Ansatz nicht auffallen und das Oligonukleotid erscheint spezifischer als es ist. Ein zweiter Grund, die Sekundärstrukturen nicht auf diese Art und Weise zu berücksichtigen, ist der hohe Rechenaufwand, der diesem Ansatz zu Grunde liegt.

Ein Ansatz, der die Berücksichtigung des „ΔG correction terms" für den „top strand", also für die →Ziel-Sequenz, auf die Spitze treibt, ist in [23] beschrieben. Dort werden „*structure specific →probes*" erstellt für ein „*→mutation discrimination that target the regions of structural, rather than sequence, differences*". Dieser Ansatz wird jedoch durch einen beträchtlichen labortechnischen Aufwand unterstützt. In einem Experiment werden Enzyme verwendet, die bei Hairpin Strukturen mit einem Stem von mindestens 7 bp an spezifischen Positionen (*cleavage sites*) schneiden. Aus der fragmentierten DNA werden diese Positionen ermittelt und als Nebenbedingungen für mfold verwendet. Die auf diese Weise berechneten Sekundärstrukturen sind damit weniger fehlerhaft. Für ein Design von Oligonukleotid-Bibliotheken ist diese Methode nur in Ausnahmefällen verwendbar, da im Allgemeinen eine sehr große Anzahl von Ziel-Sequenzen berücksichtigt werden muss.

Bei der zumeist ebenfalls sehr großen Anzahl an Oligonukleotiden, die gegen jede Position auf allen Ziel- und Nichtziel-Sequenzen abgeglichen werden, ist auch schon das Grenzwert-Kriterium $H(x, t) \leq g$ sehr rechenaufwändig. Der Einsatz der exakteren aber aufwändigeren Methoden, wie z.B. die Verwendung von thdist(x, t) oder die intensive Berücksichtigung von Sekundärstrukturen, muss im Einzelfall geprüft werden.

In diesem Abschnitt wurden mehrere Ansätze vorgestellt, einen „→Treffer" eines Oligonukleotids auf einer →Ziel- oder →Nichtziel-Sequenz vorherzusagen bzw. für den Auswahlprozess zu definieren. Mit massivem Einsatz großer Rechenkapazitäten wäre der Ansatz über thdist(x, t) sicherlich in akzeptabler Zeit berechenbar und die Vorhersage der Hybridisierungseffizienz recht gut. Noch bessere und exaktere Bewertungen für die Hybridisierungseffizienz erhält man mit Hilfe von Programmen, die den Hybridisierungs-Prozess detailliert simulieren [10], [26], [40], [32], aus Effizienzgründen ist dieser Ansatz für die Optimierung von Oligonukleotid-Bibliotheken mit großen →Sequenzklassen und vielen Oligonukleotiden jedoch nicht empfehlenswert.

4.2.2. Redundanz- und Toleranz-Niveau

In dem vorangehenden Abschnitt wurden mehrere Definitionen angegeben, was ein Treffer bzw. ein positives Signal eines Oligonukleotids auf einer Ziel-Sequenz ist. Für die in Kapitel 2 beschriebene gesamte Aufgabenstellung leistet jedes Oligonukleotid seinen Beitrag für einen sensitiven (Abdeckung aller Varianten; „*set cover*"-Problem) und spezifischen Nachweis von Genen oder Organismen. Die Definition eines positiven Signals in dem vorangehenden Abschnitt bezog sich auf die Ebene einzelner Spots bzw. einzelner Oligonukleotide. In Bezug auf die gesamte Aufgabenstellung und bei der Auswertung der Hybridisierungssignale aller Spots eines DNA-Mikroarrays kommt häufig ein weiterer Aspekt hinzu. Die Schaffung von →Redundanz auf dem DNA-Mikroarray. Die einzelnen Spots auf einem DNA-Mikroarray oder auch die Fänger-Oligonukleotide können von einem informationstheoretischen Standpunkt als Informationskanäle zwischen einem Sender und einem Empfänger betrachtet werden. Technische Systeme, die mehrere nahezu identische solcher Informationskanäle verwenden, werden „redundant" genannt. Bei fehlerbehafteten Informationskanälen wird die Redundanz auf der Seite des Empfängers zur Fehlerkorrektur verwendet. Für DNA-Mikroarrays bedeutet die Einführung von Redundanz mehr Sicherheit bei der Auswertung der Hybridisierungssignale. Im folgenden werden zwei Formen der Redundanz auf einem DNA-Mikroarray

vorgestellt, jedoch nur eine davon geht als Kriterium in die Optimierung von Oligonukleotid-Bibliotheken ein.

DNA-Mikroarrays sind hochgradig parallele Nukleinsäure-Analyseverfahren. Es finden in bis zu 400.000 Spots[27] [89], [90] Hybridisierungen statt. Jeder einzelne Spot ist jedoch fehlerbehaftet und deshalb werden die Oligonukleotide zumeist mehrfach in Spots aufgebracht. Dadurch entsteht eine Redundanz, die →Spot-Redundanz genannt wird und Sicherheit gegenüber Fehlern auf der chemischen Oberfläche schafft oder auch Fehler bei dem Aufbringen der Oligonukleotide, z.B. mit einem Spotting-Roboter, abschwächt. Die Spot-Redundanz ist jedoch nicht Gegenstand der Optimierung einer Oligonukleotid-Bibliothek, sie hängt hauptsächlich von der Qualität der chemischen Oberfläche und des Spotting-Roboters ab. In diesem Abschnitt wird die sogenannte →Oligonukleotid-Redundanz betrachtet. Damit ist das mehrfache Treffen eines Gens oder Organismus bzw. der zugehörigen Ziel-Sequenzen mit mehreren Oligonukleotiden an verschiedenen Positionen gemeint. Ein Fänger-Oligonukleotid ist eine Komponente eines „Informationskanals" zwischen dem Ziel-Molekül in der Probe, also der Ziel-Sequenz, und der Signalerfassung. Eine mögliche Fehlerquelle auf diesem Informationskanal ist die im vorigen Abschnitt betrachtete Bewertung der Hybridisierungseffizienz. Haben mehrere Bewertungsfunktionen einem Oligonukleotid ein positives Hybridisierungssignal vorhergesagt, so kann dennoch wegen der Unsicherheit beim „Empfänger" die falsche „Nachricht" ankommen: „*Das Ziel-Molekül ist nicht in der Probe*". Das entspricht der Situation eines falsch-negativen Signals. Auch hier kann der Informationskanal mittels Redundanz durch mehrere Oligonukleotide an verschiedenen Positionen auf der Ziel-Sequenz verbreitert und damit abgesichert werden. Insbesondere die Probleme mit Sekundärstrukturen können auf diese Weise abgeschwächt werden.

Mit dem →Redundanz-Niveau $r \in \mathbb{N}$ wird die Anzahl der Oligonukleotide einer vorgegebenen Oligonukleotid-Redundanz bezeichnet. Fällt das positive Signal eine Oligonukleotids aus, so gibt es weitere r-1 Oligonukleotide, die diesen Fehler mildern können. Der umgekehrte Effekt, nämlich ein Hybridisierungssignal zu erhalten, wenn man keines erwartet hat, ist für die Bestimmung der →Spezifität besonders wichtig. Insbesondere bei sehr hohen Redundanz-Niveaus, z.B. $r = 20$, muss man sich fragen, ob ein einzelnes falsch-positives Hybridisierungssignal ein großes Gewicht bei der Optimierung einer Oligonukleotid-Bibliothek oder bei der Auswertung haben sollte. Daher wird das →Toleranz-Niveau $s \in \mathbb{N}$ eingeführt, das als Parameter für den Optimierungs-Algorithmus eine obere Grenze für die Anzahl von zugelassenen falsch-positiven Signalen darstellt.

Im vorangehenden Abschnitt wurden Grenzwerte für die Definition von positivem oder negativem Signal eines Oligonukleotids x auf einer Ziel-Sequenz eingeführt. Hier definiert das Redundanz-Niveau, wann eine Ziel-Sequenz für den Optimierungs-Algorithmus im Sinne des „*set cover*"-Problems als abgedeckt und damit als vollständig bearbeitet gilt, nämlich dann wenn sie r-mal ein positives Signal erzeugt hat. Die ungleiche Behandlung von Ziel- und Nichtziel-Sequenz im vorangehenden Abschnitt, durch zwei verschiedene Grenzwerte (siehe Tabelle 4.2-2 oder Abbildung 4.2-1) für positive und negative Signale, taucht hier ebenfalls mit dem Parameter-Paar Redundanz- und Toleranz-Niveau auf. Bei einem „*set cover*"-Problem mit Spezifitäts-Nebenbedingung dürfen die Nichtziel-Sequenzen $t \in A$ höchstens s-mal getroffen werden, erst bei mehr als s Treffern gehen sie als falsch-positives Signal in die Berechnung der Spezifität ein.

[27] In der „Affymetrix Technologie [...] lassen sich so ca. 400.000 verschiedene Gruppen von Oligonukleotiden auf einer Fläche von ca. 1,6 cm^2 plazieren. Jede Gruppe enthält dabei ca. 10^7 Oligomoleküle." [90]

Die Berechnungsvorschrift für die Sensitivität und Spezifität unter Berücksichtigung von Redundanz- und Toleranz-Niveau wird im folgenden angegeben. Dazu ist es notwendig eine neue Speicher-Struktur einzuführen. Für jede Ziel-Sequenz t aus $M \cup A$ und für eine Oligonukleotid-Teilbibliothek, $L_i \subset K$ wird ein Speicher $m(t, L_i)$ definiert, der die Anzahl der Treffer einer Oligonukleotid-Bibliothek auf dieser Sequenz zählt. Mit $m(t, L_i)$ werden neue Definitionen für die Anzahlen der richtig-positiven rp, falsch-negativen fn, fp und rn angegeben, die somit auf den vorangegangenen Definitionen für diese Zahlenwerte basieren:

$$m(t, L_i) := |\{x \in K \mid signal(x, t) \leq g_Z\}| \text{ , falls } t \in M$$

$$m(t, L_i) := |\{x \in K \mid signal(x, t) \leq g_N\}| \text{ , falls } t \in A$$

$$rp := |\{t \in M \mid m(t, L_i) \geq r\}| \; ; \qquad fp := |\{t \in A \mid m(t, L_i) > s\}|$$

$$fn := |\{t \in M \mid m(t, L_i) < r\}| \; ; \qquad rn := |\{t \in A \mid m(t, L_i) \leq s\}|$$

Die Abbildung signal(x, t) steht hier stellvertretend für entweder H(x, t) oder thdist(x, t). Mit den soeben definierten vier Zahlenwerten kann nun nach den bekannten Formeln die Sensitivität und Spezifität für eine ganze Oligonukleotid-Teilbibliothek unter Berücksichtigung des hier eingeführten Parameter-Paares Redundanz- und Toleranz-Niveau berechnet werden.

Mit der Treffer-Redundanz bzw. dem Redundanz-Niveau r und dem Toleranz-Niveau s wird vorgegeben, dass möglichst alle Ziel-Sequenzen r-mal getroffen werden sollten und die Nichtziel-Sequenzen höchstens s-mal. Die Treffer-Redundanz ist für das Erzeugen von Hybridisierungssignalen, mit den zugehörigen labortechnischen Problemen, von großer Bedeutung. Auf diese Weise wird eine aussagekräftige Auswertung eines DNA-Mikroarrays auch dann noch machbar sein, wenn einige wenige Spots aus labortechnischen oder thermodynamischen Gründen kein Hybridisierungssignal gegeben haben. Auf der anderen Seite laden „*high density microarrays*“ gerade dazu ein, mittels eines höheren Redundanz-Niveaus, Sicherheit bei der Auswertung zu schaffen.

4.2.3. Sekundärstruktur-Bewertungsfunktionen

Die Berechnung von Sekundärstrukturen der Ziel-Sequenzen und deren Bedeutung für die DNA-Analytik mit DNA-Mikroarrays wurde bereits intensiv im Abschnitt 2.3 erläutert. Sie können ganz beträchtlich das Zustandekommen von Hybridisierungssignalen behindern [19], [75], [100], und es wurde daher in Abschnitt 4.2.1 nicht zu unrecht in Betracht gezogen, die Sekundärstrukturen mit in die Vorhersage eines positiven oder negativen Hybridisierungssignals einzubeziehen [23]. Eine Sekundärstruktur mit einer großen Stabilität ist generell problematischer als eine mit einer geringen Stabilität. In diesem Abschnitt wird jedoch zusätzlich die Stabilität der Teile einer Sekundärstruktur betrachtet, die einer potentiellen Hybridisierung eines Oligonukleotids mit der Ziel-Sequenz im Wege stehen.

Dafür werden im folgenden zwei Bewertungsfunktionen angegeben, die die Zugänglichkeit des Oligonukleotids an die Ziel-Sequenz quantifizieren und somit dieses Kriterium für den Optimierungs-Algorithmus erschließen. Die erste Version der Bewertungsfunktion basiert auf der Berechnung der →mfe-Struktur, die „stabilste“ Sekundärstruktur mit der minimalen freien Energie (mfe: *minimal free energy*) und berücksichtigt damit nicht, dass es zu jeder Sequenz ein ganzes Ensemble von mehr oder weniger stabilen Sekundärstrukturen gibt, zwischen denen sich ein Gleichgewicht herausbildet. Diese Eigenschaft wird in der zweiten Version der Bewertungsfunktion berücksichtigt. Beiden Versionen gemein ist, dass sie neben der Sequenz der Ziel-Nukleinsäure zusätzlich das Fänger-Oligonukleotid und damit auch die zumeist eindeutig gegebene Position des Oligonukleotids auf der Ziel-Nukleinsäure berücksichtigen müs-

sen. Ein dritter Ansatz ist die in [19] vorgestellte Verwendung von „probability profiles“, die wie die zweite Version in dieser Arbeit mit Basenpaarwahrscheinlichkeiten arbeitet.

Bei variantenreichen Virengenomen haben einige Oligonukleotide x die Aufgabe eine große Menge $M' \subset M$ von Ziel-Sequenzen $t \in M'$ abzudecken bzw. nachzuweisen. Demzufolge müssten die in den folgenden Abschnitten definierten Bewertungsfunktionen $\Delta\Delta G(x, t)$ und sek(x, t) für jedes $t \in M'$ berechnet werden. Wegen des hohen Rechenaufwands beider Bewertungsfunktionen wurde in diesen Fällen für jede Sequenzklasse nur ein Repräsentant aus M´ bewertet.

4.2.3.1. Der ΔΔG-Ansatz zur Sekundärstruktur-Bewertung

Die Stabilität einer Sekundärstruktur wird in ΔG gemessen. Je größer das negative ΔG vom Betrag ist, umso stabiler ist die zugehörige Sekundärstruktur. Bei der Berechnung der →mfe-Struktur zu einer vorgegebenen Ziel-Sequenz $t \in M$ wird die Sekundärstruktur mit dem kleinsten ΔG berechnet. Sei nun weiterhin ein Oligonukleotid $x \in K$ gegeben, das nach einem der in Abschnitt 4.2.1 beschriebenen Kriterien die Sequenz t trifft. Erwartungsgemäß wird damit ebenfalls eine Position auf der Sequenz t eindeutig gegeben sein, andernfalls wird entweder ein $\Delta\Delta G(x, t)$ für mehrere Positionen berechnet oder es wird die Position mit dem kleinsten thdist(x, t) gewählt.

Mit dem Oligonukleotid x, der Ziel-Sequenz t und der Position von x auf t kann nun ein ΔG_1 zu der mfe-Struktur von t berechnet werden, unter der Bedingung, dass keine Basenpaarungen an der Position von x zugelassen sind. Berechnet man weiterhin ein ΔG_2 ohne dieser Bedingung, so ist $\Delta\Delta G(x, t) = \Delta G_1 - \Delta G_2$ proportional zu der Anzahl der Basenpaarungen, die sich ohne diese Bedingung an der betrachteten Position ausgebildet hätten. Da ΔG_2 mit möglicherweise mehreren Basenpaarungen stabiler als ΔG_1 ist, gilt $\Delta G_1 - \Delta G_2 \geq 0$.

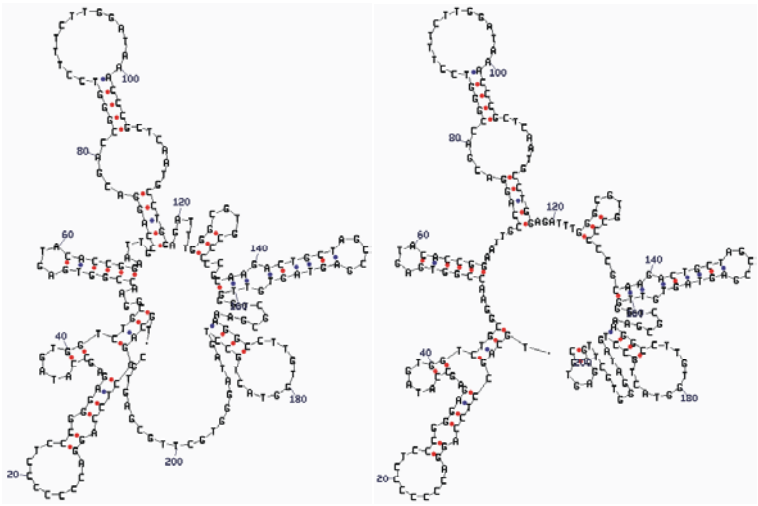


Abbildung 4.2-2: Sekundärstruktur mit und ohne Basenpaarungs-Bedingung

Die Abbildung 4.2-2 stellt zwei berechnete Sekundärstrukturen dar, die sich mit und ohne dieser Bedingung ergeben. Abgesehen von einem leicht unterschiedlichen „Layout“ hat die linke Sekundärstruktur, die mit der Bedingung berechnet wurde, am unteren Rand eine Sequenz mit 21 ungepaarten Basen. Da die rechte Sekundärstruktur in diesem Bereich nur zwei Basenpaarungen aufwies ist hier die Differenz zwischen $\Delta G_1 = -18{,}9$ und $\Delta G_2 = -19{,}0$ recht gering.

$\Delta\Delta G(x, t)$ könnte ebenfalls als Quotient $\Delta G_2 / \Delta G_1$ realisiert werden. Dann wäre $\Delta\Delta G(x, t) \geq 1$. In beiden Fällen werden die hier definierten Bewertungsfunktionen so eingesetzt, dass sol-

che Oligonukleotide mit einem möglichst kleinen Zahlenwert bevorzugt werden. Die Berechnung von $\Delta\Delta G(x, t)$ ist sehr aufwändig, daher wird dieses Kriterium, aus Gründen der Effizienz, auf eine möglichst kleine Menge von, durch andere Kriterien, vorgefilterten Oligonukleotiden angewendet.

4.2.3.2. Ansatz über die Matrix der Basenpaarwahrscheinlichkeiten

Die Matrix der Basenpaarwahrscheinlichkeiten enthält alle zu einer gegebenen Sequenz möglichen Sekundärstrukturen. Alle Strukturen dieses Ensembles bilden sich im Gleichgewicht tatsächlich aus, jedoch mit stark unterschiedlichen Häufigkeiten. Falls dieses Ensemble zwei oder mehr stabilste Strukturen mit einer geringen Differenz der freien Energien ΔG enthält, die sich zudem an der Position des betrachteten Oligonukleotids stark unterscheiden, so hat dieser Ansatz deutliche Vorteile gegenüber der soeben beschriebenen Berechnung von $\Delta\Delta G$ als Bewertungsfunktion für die Zugänglichkeit des Oligonukleotids x an der entsprechenden Position in der Sekundärstruktur der Ziel-Sequenz t.

Die hier definierte Bewertungsfunktion sek(x, t) berücksichtigt die Position des Oligonukleotids x und die Eigenschaft von t möglicherweise mehrere Sekundärstrukturen auszubilden. sek(x, t) wird als Summe der Basenpaarwahrscheinlichkeiten der Zeilen und Spalten der Dotplot-Matrix berechnet, die der Position des Oligonukleotids entsprechen. Die Abbildung 4.2-3 stellt das Prinzip der Berechnung von sek(x, t) dar. Die Abbildung 4.2-3/links enthält die aus Abschnitt 2.3 bekannte Sekundärstruktur mit einem Oligonukleotid an einer relativ ungünstigen Position (vgl. Abbildung 2.3-2). Hier gibt es neben der mfe-Struktur, die mit dem Oligonukleotid an den Positionen (1), (2) und (3) überlappt, nur vernachlässigbare konkurrierende weitere Sekundärstrukturen. Die Abbildung 4.2-3/rechts zeigt eine in [27] für die Problematik multistabiler Nukleinsäuremoleküle konstruierte Sequenz. Deutlich erkennt man neben der unverzweigten Abfolge von Stems die fünf Stems der zweiten Sekundärstruktur. Die rechts eingezeichnete Oligonukleotid-Position hat neben der Überlappung mit Teilen der mfe-Struktur auch einen Stem mit der zweiten Struktur gemein.

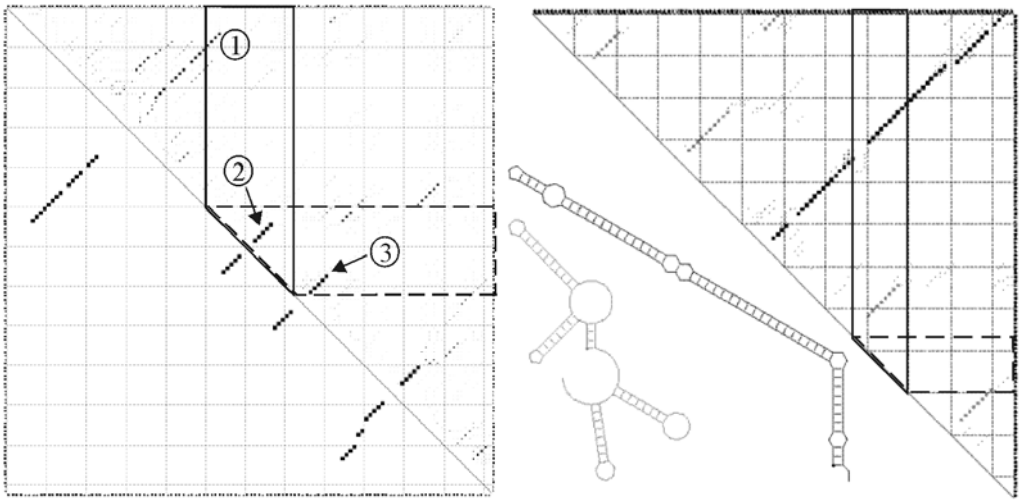


Abbildung 4.2-3: Prinzip der Berechnungsvorschrift von sek(x, t)
(rechts: Dreiecksmatrix und Sekundärstruktur aus [27])

In beiden Teilen der Abbildung ist jeweils ein Oligonukleotid durch umrahmte Bereiche von Zeilen und Spalten hervorgehoben. In diesen Bereichen werden die Komponenten der Matrix, die Basenpaarwahrscheinlichkeiten, aufaddiert. Ein zweifaches Aufaddieren der Komponenten in der Schnittmenge des gestrichelt umrandeten Zeilenbereiches und des durchgezogen umrandeten Spaltenbereiches oder wenigstens eine höhere Gewichtung dieser Komponenten ist durchaus begründbar, denn wie bereits in Abschnitt 2.3 erwähnt, sind diese Basen-

paarungen ebenfalls auf der Fänger-Sekundärstruktur zu finden. Zudem sind sie auf der Sekundärstruktur der Ziel-Sequenz →kinetisch begünstigte sehr lokale →Basenpaarungen.

Oligonukleotide x mit einem kleineren Zahlenwert werden durch die Optimierungs-Algorithmen bevorzugt in eine Oligonukleotid-Bibliothek aufgenommen. Nach erfolgter Berechnung der Dotplot-Matrix können für verschiedene Oligonukleotide die Werte sek(x, t) sehr effizient, ausschließlich durch Zugriffe auf Teile dieser Matrix, berechnet werden. Die vom einzelnen Oligonukleotid abhängige und deutlich aufwändigere Berechnung der →mfe-Struktur mit Basenpaarungs-Restriktion, bei dem Ansatz mit $\Delta\Delta G$, entfällt hier.

4.2.4. Eigenschaften der Fänger-Oligonukleotide

Die bisher betrachteten Bewertungsfunktionen hatten teilweise auch schon Eigenschaften der Oligonukleotide betrachtet. Diese Eigenschaften waren jedoch in einen größeren Kontext eingebettet, z.B. bei der Definition der Trefferanzahlen, oder sie waren zusammen mit einer Ziel-Sequenz t definiert (vgl. $\Delta\Delta G(x, t)$ und sek(x, t) in Abschnitt 4.2.3). In diesem Abschnitt werden die Eigenschaften des Oligonukleotids an sich aufgelistet. Es wird angegeben, wie sie berechnet werden und welchen Einfluss diese Bewertungen auf den Optimierungs-Algorithmus haben.

4.2.4.1. Schmelztemperatur, Oligonukleotid-Länge und GC-Gehalt

Diese drei Eigenschaften bilden eine Einheit, da sie jeweils voneinander abhängig sind. Die Wallace-Regel [113] zur Berechnung der Schmelztemperatur macht diese Aussage ganz deutlich:

$$\text{Schmelztemperatur:} \quad T_M = 2\,\#[AT] + 4\,\#[GC] \quad \text{Wallace-Regel}$$

$$\text{Oligonukleotid-Länge:} \quad |x| = \#[AT] + \#[GC] = \#[A] + \#[T] + \#[G] + \#[C]$$

$$\text{GC-Gehalt:} \quad \%GC = \#[GC] / |x|$$

Dabei bezeichnet #[GC] die Anzahl der Basen Guanin oder Cytosin und #[AT] die Anzahl der Basen Adenin und Thymin. Sei |x| die Länge des Oligonukleotids x. Der in der Literatur und in den meisten Primer-Design-Programmen häufig genannte GC-Gehalt „%GC“, der Anteil von G und C an der Gesamtanzahl der Basen, ist demzufolge #[GC] / |x|. Allgemeiner und informationshaltiger ist die Chromizität einer Nukleinsäuresequenz (#[A] / |x|, #[T] / |x|, #[G] / |x|, #[C] / |x|). Diese stellt den Anteil aller Basen einer Sequenz dar.

Die Wallace-Regel macht deutlich, dass die Schmelztemperatur im Allgemeinen mit der Länge des Oligonukleotids wächst. Diese Aussage wird exakt, wenn zudem ein konstanter GC-Gehalt vorausgesetzt wird. Umgekehrt wächst die Schmelztemperatur ebenfalls bei konstanter Länge und wachsendem GC-Gehalt. Weiterhin gilt, wie beim Design von Oligonukleotid-Bibliotheken, für vorgegebene Schmelztemperaturen, dass kürzere Oligonukleotide einen höheren GC-Gehalt haben müssen und umgekehrt.

In der Abbildung 4.2-4 ist unter Verwendung der Wallace-Regel die Schmelztemperatur in Abhängigkeit von %GC und der Oligonukleotid-Länge aufgetragen. Zusätzlich wurde in einem Grauwert die für die entsprechende (x,y)-Koordinate, in diesem Fall (%GC, Länge), die Anzahl der möglichen Oligonukleotide als Multinomialkoeffizient dargestellt. Ein dunklerer Grauwert steht für eine größere Anzahl. Bei einem %GC von 0 beispielsweise bestünde das Oligonukleotid nur aus den Basen A und T und bei einer Länge von 20 Basen würden sich damit „nur“ etwas über $2^{20} = 10^6$ mögliche Oligonukleotide ergeben. Bei der gleichen

Länge und einem GC-Gehalt von 0,5 ergäben sich, berechnet mit dem Multinomialkoeffizienten[28], mehr als 10^{10} mögliche Oligonukleotide.

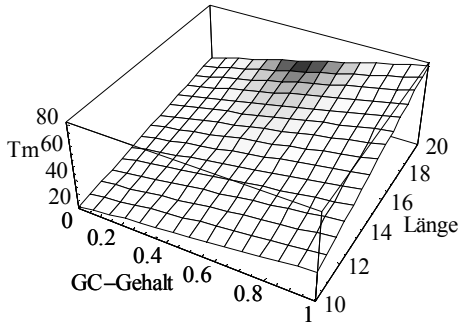


Abbildung 4.2-4: GC-Gehalt, Oligonukleotid-Länge und Schmelztemperatur

Oligonukleotide einer Bibliothek lägen, wegen der Beschränkung auf ein kleines Intervall von Schmelztemperaturen, in der Abbildung 4.2-4 auf einem Band von Höhenlinien dieser Schmelztemperaturen, und nicht zuletzt aus biologischen oder den soeben betrachteten kombinatorischen Gründen, ausschließlich in einem kleinen Intervall um einen GC-Gehalt von 0,5.

Die Bedeutung dieser drei Eigenschaften (GC-Gehalt, Oligonukleotid-Länge und Schmelztemperatur) für die Erstellung von Oligonukleotid-Bibliotheken ist fundamental. Häufig werden von den Molekularbiologen Intervalle zu zweien dieser Eigenschaften zur Verwendung als scharfes Grenzwert-Kriterium vorgegeben. Ein Intervall zur dritten Eigenschaft ergibt sich damit aus den ersten beiden, dieses kann durch eine Abfolge von Ebenenschnitten anhand der Abbildung 4.2-4 durchgespielt werden.

Der Aufwand zur Berechnung dieser Eigenschaften ist sehr gering, daher werden die soeben erwähnten scharfen Grenzwert-Kriterien, z.B. $58°C \le T_m \le 62°C$ und $15 \le |x| \le 35$, zur Vorfilterung, der zumeist sehr großen Menge aller aus den vorgegebenen →Sequenzklassen ableitbaren Oligonukleotide, verwendet. Die so reduzierte Menge von Oligonukleotiden bildet die Menge der Oligonukleotid-Kandidaten (engl.: *candidate →probes* [58], *candidate oligonucleotides* [53]). Neben der →Wallace-Regel [113], die nur für kurze Oligonukleotide bis 12 oder höchstens 20 Basenpaare angewendet werden sollte, gibt es weitere Formeln zur Berechnung der Schmelztemperatur, die ebenfalls mit geringem Rechenaufwand berechnet werden können.

$$(1) \qquad T_m = 81{,}5°C + 16{,}6 \log[c(Na^+)] + 0{,}41(\%GC) - 500/n$$

Die einzige Bewertungsfunktion, die noch gröber als die Wallace-Regel ist, ist der GC-Gehalt selber, der trotzdem noch Verwendung findet. Die Formel (1) aus [64] erhält neben der Salzkonzentration $c(Na^+)$ ausschließlich den GC-Gehalt %GC der Oligonukleotid-Sequenz und berechnet eine grobe Annäherung in °C. Die Formel (1) ist auch für Oligonukleotide mit mehr als 50 Basenpaaren gültig, und es gibt für diese Version einer Schmelztemperatur-Formel zwei weitere Versionen für →RNA-RNA und RNA-DNA Hybridisierungen.

[28] Der Multinomialkoeffizient $(N; n_1, n_2, ..., n_m) = (n_1 + n_2 + ...)! / (n_1! \, n_2 ! \, ...)$ mit $N = \Sigma \, n_i$ ist die Anzahl der Möglichkeiten eine Menge mit N verschiedenen Elementen in m Teilmengen mit den Kardinalitäten n_i zu zerlegen. Bezogen auf Nukleinsäure-Sequenzen bedeutet das, dass $(N; n_A, n_C, n_G, n_T)$ die Anzahl der möglichen Sequenzen der Länge N mit n_A As, n_C Cs, n_G Gs und n_T Ts ist. Für das hier betrachtete Oligonukleotid der Länge 20 gilt: $(N; 20, 0, 0, 0) = 1$ und $(N; 17, 1, 1, 1)$ ist bereits 6840 und $(N; 5, 5, 5, 5) = 11.732.745.024 > 10^{10}$. Dabei hätten die zu $(N; 5, 5, 5, 5)$ gehörigen Oligonukleotide einen GC-Gehalt von 0,5.

(2) $T_m = \Sigma(f_{ij} \cdot T_{ij})$ mit einem Parametersatz (T_{ij}) für T_m

(3) $T_m = \Delta H° / (\Delta S° + R \ln C_T)$ mit $\Delta H = \Sigma(f_{ij} \cdot H_{ij})$, ΔS analog

Die Formeln (2) [9] und (3) [16], [95], [96] arbeiten mit thermodynamischen Parametersätzen $((T_{ij})$, (H_{ij}) und $(S_{ij}))$, die durch aufwändige Messreihen ermittelt wurden [121], [22], [105]. Jeweils einem Paar (i, j) von zwei aufeinanderfolgenden Basen, den sogenannten „*nearest neighbors*“, wird einer der Parameter T_{ij}, H_{ij}, und S_{ij} zugeordnet. Die f_{ij} bezeichnen die Häufigkeit, mit der das Paar (i, j) in dem Oligonukleotid vorkommt. Damit geht, im Verhältnis zu (1), bereits deutlich mehr Information aus der Oligonukleotid-Sequenz in die Berechnung ein. ΔH ist die Enthalpie und ΔS die Entropie des Gleichgewichtszustands, der sich bei der Hybridisierung bildet. Diesem Ansatz liegt die Erkenntnis zugrunde, dass weniger die Wasserstoffbrückenbindungen zwischen den Einzelsträngen, sondern vielmehr längs der Achse der Helix wirkende Kräfte zwischen zwei aufeinanderfolgenden Basen die Stabilität der Hybridisierung bewirken. Diese Kräfte werden „*stacking forces*“ oder auch „*nearest neighbor interactions*“ genannt.

Aus Symmetriegründen gibt es nicht 16 sondern nur 10 Parameter für die „*nearest neighbor interactions*“, da es zu jedem solchen Paar auf dem Gegenstrang das Watson-Crick-Komplement (Umkehrung der Sequenz und Übergang von A→T, T→A, G→C und C→G) mit dem gleichen Zahlenwert gibt. Die 16 Möglichkeiten werden jedoch nicht auf 8 halbiert, da 4 Paare aufeinanderfolgender Basen selbstkomplementär sind. Die übrigen 12 sind ungleich ihrem Watson-Crick-Komplement und werden auf 6 reduziert, somit gibt es 4 + 6 = 10 Parameter.

Die Formeln (2) und (3) sind bereits deutlich besser als (1), der Ansatz (3) hat sich als Standard herausgebildet. HyTher [45] und MELTING [86] arbeiten mit dieser Formel. Bis heute wird mit neuester Messtechnik und möglichst großen Datenmengen versucht, die Parametersätze weiter zu verbessern. Es gibt bereits erste Versuche thermodynamische Parametersätze für drei aufeinanderfolgende Basen zu erstellen. Weitere Ansätze, die Berechnung der Schmelztemperatur T_m zu verbessern, sind spezielle Parameter für Hybridisierungen mit Basenfehlpaarungen (→*mismatches*) [1], [3], [2], [4], [70], [117] oder Korrekturterme für DNA-Mikroarrays und Sekundärstruktur, die in dem Programm HyTher [45] eingesetzt werden, oder die Berücksichtigung von verschiedenen Salzkonzentrationen $[Na^+]$:

(4) $T_m(\text{microarray}) = 1.2 \times T_m(\text{solution}) - 27.8°C$

(5) $T_m = 193.67 – (3.09\text{-}f_{(G+C)})(34.47\text{-}6.52 \log[Na^+])$

(6) $T_m = \Delta H° / (\Delta S° + R \ln C_T) + 16.6 \log\left(\frac{[Na^+]}{(1 + 0.7[Na^+])}\right)\text{-}269.3$

(4) aus [28] ist eine recht grobe Formel, die jedoch deutlich macht, dass die Hybridisierungen auf einem DNA-Mikroarray (Festphasenhybridisierungen) generell weniger stabil sind als solche in Lösung (Hybridisierungen in flüssiger Phase). Sicherlich gilt diese Faustformel nur für Oligonukleotide bis zu einer Länge von 40 Basenpaaren, da sonst der Faktor 1.2 den absoluten Term –27.8°C überwiegt und $T_m(\text{microarray}) < T_m(\text{solution})$ nicht mehr gilt. Wie auch schon die Formel (1) berücksichtigt (5) aus [9] verschiedene Salzkonzentrationen $[Na^+]$ und (6) aus [40], [118] ist eine für Salzkonzentrationen verallgemeinerte Version von (3).

Weiterhin gibt es neben DNA-DNA auch Parametersätze für RNA-RNA und RNA-DNA Hybride, dabei sind die RNA-RNA Hybridisierungen am stabilsten, dann kommen RNA-DNA und DNA-DNA Hybridisierungen. Mit chemisch modifizierten Basen, z.B. mit der Peptidnukleinsäure PNA, bekommt man noch mehr Stabilität als bei RNA-RNA. Es gibt Ansätze die

Problematik mit Sekundärstrukturen mit Hilfe von PNA zu lösen, die Peptidnukleinsäuren sind jedoch sehr teuer in der Synthetisierung.

Auf die Oligonukleotid-Länge wurde in diesem Abschnitt bisher wenig eingegangen. Sie hat jedoch einen wesentlichen Einfluss auf das Design von Oligonukleotid-Bibliotheken. Trivial ist, dass bezogen auf die Übereinstimmung von Zeichenketten eine längere Sequenz spezifischer als eine kurze und umgekehrt eine kurze Sequenz sensitiver als eine lange ist (d.h. mehr Ziel-Sequenzen trifft). Bezogen auf Hybridisierungs-Eigenschaften und das generieren von Hybridisierungssignalen gilt es jedoch einen Effekt zu berücksichtigen, der dem soeben genannten Zusammenhang entgegen wirkt. Bei längeren Fänger-Oligonukleotiden werden mehr →Mismatches benötigt, um →Nichtziel-Sequenzen diskriminieren zu können. Im UFT des FuE-Verbunds Gensensorik wurden mit Fänger-Oligonukleotiden um einer Länge von 20 bp, ein Mismatch diskriminiert. In [58] wird bei einer Länge von 50 bp von Kreuzhybridisierungen (falsch-positiven Signalen) mit bis zu 10 Mismatches ausgegangen; und bei 70 bp Länge 20 Mismatches. Dieser Zusammenhang muss bei dem Design von Oligonukleotid-Bibliotheken berücksichtigt werden und geht z.B. in die Wahl der in Abschnitt 4.2 eingeführten Grenzwerte g_Z und g_N ein.

4.2.4.2. Weitere Eigenschaften der Oligonukleotide

Neben Schmelztemperatur, Oligonukleotid-Länge und GC-Gehalt werden hier weitere Eigenschaften der Fänger-Oligonukleotide betrachtet, die nicht im Kontext einer Ziel-Sequenz zu sehen sind oder die Trefferanzahlen anbelangen. Die folgenden Eigenschaften beeinflussen alle die Qualität und Effizienz, mit der sich die Hybridisierungssignale ausbilden:

1) Sekundärstrukturen der Fänger-Oligonukleotide
2) GC-Clamp
3) Affymetrix-Regeln

Wie die Sekundärstruktur der Ziel-Sequenzen, so schwächen ebenfalls die kleinen Sekundärstrukturen der Fänger-Oligonukleotide, zumeist *Hairpin-Loops* mit kleinen *interior Loops*, die Bildung von Hybridisierungssignalen sehr. Häufig bildet sich diese Struktur auf dem Fänger und auf der Ziel-Sequenz, zumal sie als Struktur mit lokalen →Basenpaarungen durch die →Kinetik begünstigt ist. Das Kriterium GC-Clamp ist aus dem Primer-Design bekannt. Damit wird die Anzahl der Basen G und C an den Enden des Oligonukleotids bezeichnet. Diese begünstigen durch ihre hohe Stabilität die Effizienz und Initiierung einer Hybridisierung. Die Basen an den Enden im Allgemeinen und besonders A und T neigen dazu, ihre Basenpaarung aufzulösen. Der vielfach in Modellen verwendete „*helix initiation factor*“ [105] ist ein Resultat dieses Effekts.

Zu guter letzt gibt es einen ganzen Satz von Regeln [62], die in der Literatur [58] als Affymetrix-Regeln bzw. als „*Affymetrix pobe selection criteria*“ bekannt sind. Sie beschreiben die Eigenschaften für DNA-Mikroarrays geeigneter Oligonukleotide. Durch diese Regeln sollten im wesentlichen besonders „pathologische“ Sequenzen, wie z.B. AAAAATTTTTCCCCCGGGGG, ausgeschlossen werden[29]. Die direkte Analyse „*of →probe behavior as a function of certain sequence features*“ führte zu folgendem Satz von Regeln, die für Oligonukleotide mit 20 bp gelten. Hier wird die Übersetzung der Regeln aus [62], da sie missverstanden werden können, zusammen mit einer exakten und formalen Version, nach der oben eingeführten Nomenklatur, angegeben:

[29] Dieses Oligonukleotid ist nur beinahe ein Beispiel für eine Sequenz, die nicht die Regeln erfüllt, denn sie erfüllt gerade eben die Regel 5 und auch die im folgenden definierte Palindrom-Bewertungsfunktion der Regel 7 würde nur dann diese Sequenz ausschließen, wenn man sie zusätzlich auch „versetzt“ anwenden würde, dann ergäben sich 10 GC-Paarungen und damit mehr als 7. Alle übrigen Regeln werden ebenfalls erfüllt.

1) die Gesamtanzahl von As oder Ts ist kleiner als 10
   $\#A < 10$ und $\#T < 10$
2) die Gesamtanzahl von Cs oder Gs ist kleiner als 9
   $\#C < 9$ und $\#G < 9$
3) die Anzahl von As oder Ts in jedem Fenster von 8 Basen ist kleiner als 7
   Sei F_8 die Menge der 13 Fenster von 8 Basen eines 20mers;
   Für alle $x \in F_8$ gilt: $\#A < 7$ und $\#T < 7$
4) die Anzahl von Cs oder Gs in jedem Fenster von 8 Basen ist kleiner als 6
   Sei F_8 wie oben definiert: Für alle $x \in F_8$ gilt: $\#C < 6$ und $\#G < 6$
5) nicht mehr als 5 aufeinanderfolgende Cs oder Gs
   Sei F_6 die Menge der 15 Fenster von 6 Basen eines 20mers;
   Für alle $x \in F_6$ gilt: „CCCCCC“, „GGGGGG“ $\notin F_6$
6) nicht mehr als 6 aufeinanderfolgende As oder Ts
   Sei F_7 die Menge der 14 Fenster von 7 Basen eines 20mers;
   Für alle $x \in F_7$ gilt: „AAAAAAA“, „TTTTTTT“ $\notin F_7$
7) eine Palindrom-Bewertungsfunktion von kleiner als 7
   Sei $Hamming_{|x|}(x, x')$ der in Abschnitt 4.2.1 definierte Abstand zweier Sequenzen; sei weiterhin x' die Sequenz, die aus x ohne Umkehrung und nur durch Übergang zum Komplement der Basen (A→T, T→A, G→C und C→G) hervorgeht und sei F(x) die Menge aller Teilsequenzen von x. Dann ist $P(x) := \max\{|x_1| - Hamming_{|x1|}(x_1, x_2') \mid x_1, x_2 \in F(x)$ mit $|x_1| = |x_2|\}$ die Palindrom-Bewertungsfunktion und das Kriterium $P(x) < 7$

Die verbale Version der Regeln 1 und 2 könnte als „$\#A + \#T < 10$ *und* $\#C + \#G < 9$“ missverstanden werden. So ergäbe sich jedoch nur eine maximale Oligonukleotid-Länge |x| von 18 und das ist ein Widerspruch zu $|x| = 20$. Insgesamt sind diese Regeln, auch in ihrer z.B. in [58] grob für verschiedene Längen verallgemeinerten Version, wenig streng. Angewendet auf alle Oligonukleotide des HCV-Anwendungsbeispiels (siehe Abschnitt 7.1) wurden nur sehr wenige herausgefiltert.

4.3. formale Aufgabenspezifikation

Für eine Zusammenfassung, der bis hier eingeführten Bewertungsfunktionen und deren Kriterien, wird hier noch einmal die aus Abschnitt 4.1.2 bekannte verbale Aufgabenspezifikation wiederholt, um sie anschließend zu formalisieren. Sie lautet: „Konstruiere für eine vorgegebene Hierarchie von →Sequenzklassen G_1, G_2, G_3, ... Oligonukleotid-Teilbibliotheken L_1, L_2, L_3, ..., die ihre →Sequenzklasse möglichst vollständig mit guten Hybridisierungssignalen treffen (→Sensitivität) und unter Berücksichtigung der Hierarchie die übrigen Sequenzklassen möglichst wenig treffen (→Spezifität) bzw. ein gut zu diskriminierendes negativ-Hybridisierungssignal geben. Weiterhin soll die Sequenzklasse G_0 möglichst wenig getroffen werden.“ In der Informatik wird die Spezifikation eines Programms durch eine Anfangs- und eine Endebedingung angegeben. Im folgenden wird die Anfangsbedingung durch die notwendigen Eingaben, Optionen und Parameter beschrieben und die Endebedingung durch die geforderten Ausgaben und deren Eigenschaften. Das Programm selber wird durch einen für alle Ansätze von Optimierungs-Algorithmen identischen Rahmen-Algorithmus angegeben.

Optionen:

- O1 Eines von zwei möglichen Kriterien für die Definition von Treffern wird gewählt: H(x, t) oder thdist(x, t) ; vgl. Abschnitt 4.2.1
- O2 Wahl einer Sekundärstruktur-Bewertungsfunktion: ΔΔG(x, t) oder sek(x, t) ; vgl. Abschnitt 4.2.3

Parameter:

- P1 Parameter für die Eigenschaften der →Oligonukleotid-Sequenzen: minTm, maxTm, minLen, maxLen, minGC und maxGC mit den entsprechenden Kriterien für die Oligonukleotide $x \in K$:
  K1: $minTm \leq Tm(x) \leq maxTm$
  (mit den Parametern Salzgehalt, DNA/RNA, *microarray correction term*),
  K2: $minLen \leq |x| \leq maxLen$,
  K3: $minGC \leq \%GC(x) \leq maxGC$
- P2 Parameter für das →Treffer-Kriterium:
  Für O1=H ist $g_N, g_Z \in \mathbb{N}$; $g_Z \leq g_N$
  für O1=thdist ist $g_N, g_Z \in \mathbb{R}$; $g_Z \leq g_N$
  und Temperatur, Salzgehalt,
  DNA/RNA (Parametersätze der „nearest neighbor interactions“),
  „ΔG correction terms“ und *microarray correction term*
- P3 →Redundanz-Niveau $r \in \mathbb{N}$ und
  →Toleranz-Niveau $s \in \mathbb{N}$; vgl. Abschnitt 4.2.2
- P4 Parameter für das Sekundärstruktur-Kriterium:
  Für beide Fälle von O2 werden die Parameter Temperatur, Salzgehalt, DNA/RNA benötigt. Diese sollten im Falle von O1=thdist identisch zu den dort verwendeten Parametern sein.

Eingabe:

E1: eine Hierarchie von →Sequenzklassen,
z.B. (G0, G1 (G2(G3, G4),G5(G6, G7, G8), G9(G10, G11))) , g = 11.
Die Sequenzklassen enthalten die →Ziel-Sequenzen $t \in Gi \subset B^*$

Rahmen-Algorithmus:

- Reduktion der gesamten Aufgabenstellung auf g Teilprobleme $(M, A) = (G_i , A_i)$ von Ziel- und Nichtziel-Sequenzklassen, wie in Abschnitt 4.1.3 beschrieben. Zusammen mit dem weiter unten definierten P ergeben sich mit $(M, P, A) := (G_i , Match(K_i), A_i)$ die „*set cover*“-Probleme mit Spezifitäts-Nebenbedingung.
- Bestimme die Menge K´ aller aus M´ als Teilsequenzen ableitbaren Fänger-Oligonukleotide $K' \subset B^*$; M´ war im Abschnitt 4.1.3 als Vereinigung aller G_i mit $i \neq 0$ definiert.
- Filterung der Oligonukleotide x aus K´ mit mindestens zwei der Kriterien K1, K2, K3 und den optional auch mit den Affymetrix-Regeln R_1 bis R_7 (vgl. Abschnitt 4.2.4). Damit wird die Menge $K \subset K'$ der Oligonukleotid-Kandidaten gebildet.
- Kommentar: bis hier ist die Anzahl der Oligonukleotid-Kandidaten |P|=|K| bestimmt und für jedes $x \in K$ ist zusammen mit O1 und P2 *Match*(x) berechenbar.
- berechne P als *Match*(K) ; $P \subset \wp(M')$, d.h. für jedes Oligonukleotid $x \in K$ wird die Menge seiner →Treffer *Match*(x) $\subset$ M´ bestimmt.
- Kommentar: aus M, P und A kann nun für jedes $x \in K$ die Sensitivität sens(x) und die Spezifität spez(x) berechnet werden (vgl. Abschnitt 4.1.3). Weiterhin sind mit m(t, L), der Anzahl der Treffer einer Oligonukleotid-Bibliothek L auf einem $t \in M$, und mit Redundanz- und Toleranz-Niveau r, $s \in \mathbb{N}$ für jede Oligonukleotid-Teilbibliothek L die Zahlenwerte $spez_s(L)$ und $sens_r(L)$ (auch bereits ohne A) definiert (vgl. Abschnitt 4.2.2).

- finde für alle i=1, ..., g über kombinatorische Optimierung eine möglichst gute (sensitive) Überdeckung von G_i mit möglichst wenigen Treffern in A_i (Spezifität) und bevorzuge dabei soweit möglich die Oligonukleotide mit geringer Fänger-Sekundärstruktur, maximaler GC-Clamp (vgl. Abschnitt 4.2.4) und minimalem Zahlenwert für eine der, durch die Option O2 bestimmten, Sekundärstruktur-Bewertungsfunktionen $\Delta\Delta G(x, t)$ oder sek(x, t) (vgl. Abschnitt 4.2.3).

Ausgabe:

- A1 Eine Menge von g Oligonukleotid-Teilbibliotheken $L_i \subset K$; hier z.B.: L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}
- A2 Treffer-Tabellen (Beispiel: Tabelle 7.1-4) und Visualisierungen zur Darstellung der Positionen der Oligonukleotide auf den Sekundärstrukturen (Abbildung 7.2-6)

Endebedingung:

> Die für die Spezifikation eines Programms notwendige Endebedingung wurde im wesentlichen bereits im letzten Schritt des Rahmen-Algorithmus angegeben: Die Oligonukleotid-Teilbibliotheken L_i (i=1, ..., g) haben eine maximale Sensitivität und Spezifität und die Oligonukleotide $x \in L_i$ sind soweit möglich bezüglich ihrer Hybridisierungseigenschaften optimiert.

Insgesamt gehen 16 Parameter und Eingaben in den Algorithmus ein. Davon werden sechs (minTm, maxTm, minLen, maxLen, minGC und maxGC) hauptsächlich für die Bestimmung der Oligonukleotid-Kandidaten verwendet, fünf (Temperatur, Salzgehalt, DNA/RNA-Parametersätze der „nearest neighbor interactions“, „ΔG correction terms“ und der „microarray correction term“) für die Bewertung der Hybridisierungs-Effizienzen und ebenfalls fünf (g_N, g_Z, r, s und die Hierarchie der G_i) werden hauptsächlich für die Bestimmung der Anzahlen der richtig-positiven, falsch-positiven, richtig-negativen und falsch-negativen Klassifikationen in der Vierfeldertafel verwendet.

5. Optimierungs-Algorithmen

Zusammenfassung: In diesem Kapitel wird eine formale Beschreibung der verwendeten Optimierungs-Algorithmen angegeben, nämlich „Greedy Set Covering“, eine Kombination von Gradientenabstieg und Kompetition und Genetische Algorithmen. Es werden die Vorteile und Schwächen der drei sehr verschiedenen Ansätze deutlich. Das „Greedy Set Covering“ ist ein Verfahren zur „Lösungskonstruktion“, der Ansatz über Gradientenabstieg fällt in die Klasse der „lokalen Suche“ und der Genetische Algorithmus basiert auf evolutionären Prinzipien.

Das „Greedy Set Covering“ wird durch Verwendung einer verallgemeinerten Bewertungsfunktion modifiziert. Der Einfluss der →Sensitivität und →Spezifität einzelner Oligonukleotide kann auf diese Weise gewichtet werden. Es wird eine effiziente Berechnung von Sensitivität und Spezifität und eine den Rechenaufwand minimierende Integration der Kriterien für die Sekundärstruktur-Optimierung vorgestellt. Nach der Darstellung des Algorithmus für die Kombination von Gradientenabstieg und Kompetition wird ein konstruiertes Anwendungsbeispiel mit diesem Ansatz bearbeitet. Dabei wird deutlich, dass der Ansatz über Gradientenabstieg mit weniger Oligonukleotiden als das „Greedy Set Covering“ auskommt. Im Gegensatz zum Greedy-Algorithmus und zum Gradientenabstiegs-Verfahren wurde bei dem Ansatz über Genetische Algorithmen eine Oligonukleotid-Bibliothek als ganzes bewertet. Der Aspekt der kombinatorischen Optimierung wird dadurch deutlich besser berücksichtigt. Die Erstellung von „Genotyp 1a“-Teilbibliotheken für das Anwendungsbeispiel des Hepatitis C-Virus (vgl. Abschnitt 7.1) stellt exemplarisch die Leistungsfähigkeit des Genetischen Algorithmus dar.

In diesem Kapitel wird eine formale Beschreibung der verwendeten Optimierungs-Algorithmen angegeben. Es sollen Oligonukleotid-Bibliotheken für DNA-Mikroarrays optimal, d.h. nach den in den Kapiteln 2 und 4 beschriebenen und begründeten Kriterien, konfiguriert werden. In einer formalen Spezifikation der Aufgabenstellung wurde das „Set Cover“-Problem unter Berücksichtigung hierarchischer Strukturen zwischen den →Sequenzklassen um das Kriterium „maximale Spezifität“ erweitert. Im folgenden werden die Ansätze „Greedy Search / Greedy Set Covering“, Gradientenabstieg und Genetische Algorithmen, zu denen es im Abschnitt 2.5 bereits eine Einführung gab, detailliert vorgestellt.

Die Optimierungskriterien Schmelztemperatur, Oligonukleotid-Länge (der GC-Gehalt ergibt sich aus diesen), Bibliotheks-Größe, →Redundanz- bzw. →Toleranz-Niveau und Sekundärstrukturen werden, wenn sie nicht Teil des in Abschnitt 4.3 vorgestellten Rahmen-Algorithmus sind, in den genannten drei Ansätzen auf verschiedene Weise umgesetzt. Wie bei der Optimierung nach der Schmelztemperatur (siehe Abbildung 5.1-1) beim Greedy-Algorithmus, können durch mehrfache Aufrufe einige Vorgabeparameter (aus der Anfangsbedingung) ausgelassen und selbst ermittelt werden.

Ein exakter Vergleich der genannten drei Ansätze ist nur bei Beschränkung auf bestimmte Parameter möglich, die sich aus den grundlegenden Eigenschaften der Algorithmen ergeben. Auch die Bewertung der Qualität einer Oligonukleotid-Bibliothek darf bei dem Vergleich dieser Algorithmen nur unter Berücksichtigung der eingesetzten Rechenleistung und der Anzahl der Oligonukleotide durchgeführt werden. Nach der Beschreibung dieser drei Ansätze wird im Abschnitt 8.1 ausführlicher auf diese Problematik eingegangen.

Ein besonderes Problem ist die Optimierung nach der Sekundärstruktur, denn jede Berechnung ist eine sehr aufwändige Operation, die möglichst selten durchgeführt werden sollte. Bei dem Greedy-Algorithmus konnte dieses Problem recht einfach gelöst werden, da *nur eine* Oligonukleotid-Bibliothek konstruiert wird und die Aufnahme eines Oligonukleotids in die Bibliothek ein „seltenes Ereignis“ ist. Bei dem Genetischen Algorithmus hingegen existiert bereits bei der ersten Iteration eine ganze Population von vollständigen Oligonukleotid-Bibliotheken. In diesem Fall musste ein Verfahren entwickelt werden, das die Berechnung von Sekundärstrukturen dennoch weniger oft durchführt.

5.1. Greedy Search / Greedy Set Covering

Bei dem folgenden Greedy-Algorithmus wird nicht wie bei „lokaler Suche“ bzw. Gradientenabstieg (vgl. Abschnitt 2.5.2) oder dem Genetischen Algorithmus (vgl. Abschnitt 2.5.3) ein Suchraum, in diesem Fall der Raum aller Oligonukleotid-Bibliotheken[30], durchlaufen. Die Lösung, nämlich die Oligonukleotid-Bibliothek L, wird Schritt für Schritt konstruiert. Bei jedem Schritt wird ein Oligonukleotid der Bibliothek hinzugefügt. In Abschnitt 2.5.1 haben wir gesehen, dass das „Greedy Set Covering“ ein Kompromiss zwischen Komplexität und Qualität ist. Dabei liegt bei diesem Algorithmus das Gewicht dieses Kompromisses deutlich auf der Seite der durch Approximation bzw. Heuristik reduzierten Komplexität und für das möglicherweise suboptimale Ergebnis erhält man einen schnelleren Algorithmus. Dieser Geschwindigkeitsvorteil ergibt jedoch ein Potential für weitere Optimierungen (siehe Abschnitt 5.1.2).

5.1.1. Modifiziertes "Greedy Set Covering"

Für eine Menge von →Ziel-Sequenzen $M \subset M'$, einer Menge von Oligonukleotiden K, die $P = Match(K) \subset \wp(M')$ definieren und für eine Nichtziel-Klasse A, welche alle →Nichtziel-Sequenzen enthält wird im folgenden ein modifizierter Greedy-Algorithmus angegeben, der das „Set Cover“-Problem mit Spezifitäts-Nebenbedingung (M, P, A) approximativ löst und dabei die Kriterien Sekundärstruktur und Redundanz (bzw. →Oligonukleotid-Redundanz) berücksichtigt.

Teile des in Abschnitt 4.3 recht mathematisch formulierten Rahmen-Algorithmus überschneiden sich mit dem folgenden Algorithmus, da hier einige Bewertungsfunktionen bedingt durch die Struktur eines Greedy-Algorithmus effizienter implementiert werden können, z.B. die Berechnung von $spez_s(L)$ und $sens_r(L)$ geschieht indirekt.

Bei einem gegebenen (M, P)-Problem ist das zentrale Element, des in Abschnitt 2.5.1 angegebenen „Greedy Set Covering“-Algorithmus der eigentliche Greedy-Schritt. Bei diesem wird zu jedem Iterations-Schritt und zu der bis dahin gefundenen Lösung L, dasjenige $y \in P$ bestimmt, das am meisten Elemente aus M überdeckt, die bis dahin noch nicht überdeckt wurden. Diese Anzahl von Treffern wird hier durch eine Bewertungsfunktion ersetzt, die aus einer gewichteten Summe zwischen der Spezifität und der sogenannten „inkrementellen Sensitivität“ besteht.

$$score = w_1\, isens(y, L) + w_2\, spez(y)$$

Die inkrementelle Sensitivität $isens(x, L)$ eines Oligonukleotids x bei gegebenem Zwischenergebnis $L \subset K$ ist gerade die bei einem "Greedy Set Covering"-Algorithmus zu maximie-

[30] Bei nur |K|=1000 Oligonukleotid-Kandidaten und einer Größe der Oligonukleotid-Bibliothek von 623 gibt es $\binom{n}{k} = \binom{1000}{623}$ Möglichkeiten, das sind mehr als 10^{373} mögliche Oligonukleotid-Bibliotheken.

rende Anzahl von Treffern, die noch nicht durch L getroffen werden. Ist L = { } die leere Menge, so ist *isens*(x, L) = *sens*(x) für alle x ∈ K. Deckt L bereits die ganze Menge der Ziel-Sequenzen M ab, *Match*(L) = M, dann ist *isens*(x, L) = 0 für alle x ∈ K, und kein Oligonukleotid kann die Sensitivität von L verbessern. Sind *sens*(x) oder *isens*(x, L) ohne den Index r für das im Abschnitt 4.2.2 eingeführte →Redundanz-Niveau, geschrieben, dann beziehen sie sich auf das Redundanz-Niveau r = 1. Mit Index r bezeichnet $isens_r$(x, L) die inkrementelle Sensitivität bezogen auf die Anzahl der →Treffer, die noch nicht durch L r-mal getroffen wurden. Analog verhält es sich mit →$sens_r$(x).

Die übrigen zu optimierenden Kriterien, die Sekundärstruktur der Fänger-Oligonukleotide *sekOligo*(x) und die Sekundärstruktur der Ziel-Sequenzen t an der Bindungs-Position des betrachteten Oligonukleotids x *sekOligoZiel*(x, t), werden nach der Strategie „Optimierung soweit möglich" behandelt und sind daher nicht Bestandteil der zentralen Bewertungsfunktion „*score*". In dem Abschnitt 4.2.3 wurden für die Bewertungsfunktion *sekOligoZiel*(x, t) die zwei Ansätze ΔΔG(x, t) und sek(x, t) beschrieben. Der hier gewählte Ansatz für die soeben genannte Optimierungsstrategie erfordert, für *sekOligo*(x) und *sekOligoZiel*(x, t) jeweils ein Intervall von Zahlenwerten zu bestimmen. Diese müssen im Vorfeld durch eine Stichprobe ermittelt werden, da sie die gesamte Bandbreite der Zahlenwerte dieser Bewertungsfunktionen erfassen sollen: [*sekOligoMin*, *sekOligoMax*] und [*sekOligoZielMin*, *sekOligoZielMax*]. Diese Intervalle dienen dazu, jeweils eine Abfolge von Grenzwerten *sekOligoSchranke* und *sekOligoZielSchranke* zu definieren, die mit dem strengeren Grenzwerten beginnt und dann schrittweise relaxiert werden. Mit diesen Grenzwerten wird die Menge K der Oligonukleotid-Kandidaten auf eine Teilmenge reduziert, deren Oligonukleotide nach den Bewertungsfunktionen *sekOligo*(x) und *sekOligoZiel*(x, t) besonders gute Hybridisierungseigenschaften vorhergesagt werden. Die Abfolge der Grenzwerte definiert eine zunächst starke Reduktion, die nach und nach vermindert wird, sodass am Ende möglicherweise jeder Oligonukleotid-Kandidat, auch wenn schlechte Hybridisierungseigenschaften vorhergesagt werden, für die Maximierung von Sensitivität und Spezifität berücksichtigt wird. Damit wird letztendlich die, für einen Greedy-Algorithmus sehr wichtige, Reihenfolge verändert, in der die Oligonukleotide in die Oligonukleotid-Bibliothek L aufgenommen werden; denn die Aufnahme eines Oligonukleotids hängt neben seinen Eigenschaften ebenfalls stark von der Menge L, der bis zu diesem Schritt gewählten Oligonukleotide, ab. Die inkrementelle Sensitivität *isens*(x, L) macht das sehr deutlich.

Die Wirkung dieser Strategie wird an folgendem Beispiel erläutert. Sei L ein Zwischenergebnis eines Iterations-Schritts und $K_{red} \subset K$ eine durch die oben eingeführten Grenzwerte reduzierte Menge von Oligonukleotiden. Nun gilt zwar, dass das maximale *isens*(x_0, L) über alle x ∈ K größer-gleich dem maximalen Wert *isens*(x_r, L) über alle x ∈ K_{red} ist, ...

$$isens(x_0, L) = \max\{isens(x, L) \mid x \in K\} \geq \max\{isens(x, L) \mid x \in K_{red}\} = isens(x_{red}, L)$$

... und damit würde normalerweise x_0 als nächstes in L aufgenommen werden, jedoch die oben angegebene Strategie sorgt dafür, dass x_{red} in L aufgenommen wird. In der darauffolgenden Iteration ist die inkrementelle Sensitivität von x_0 bereits um einige Treffer von x_{red} reduziert und somit können viel mehr Oligonukleotid-Kandidaten mit x_0 konkurrieren. Diese größere Menge von Konkurrenten wird im Allgemeinen Oligonukleotide mit einer größeren Spezifität enthalten und eines davon könnte sich bei der Bewertung durch den $score = w_1\ isens(x, L) + w_2\ spez(y)$ gegen x_0 durchsetzen. Dieses Beispiel stellt recht optimistisch dar, dass die Abdeckung eines sehr sensitiven Oligonukleotids x_0 durch zwei Oligonukleotide mit besseren Eigenschaften erreicht wird. Im „*worst case*" bei Verwendung dieser Strategie jedoch enthält die Lösung L neben x_0 zusätzlich das zuvor gewählte x_{red}. Unter dem Aspekt der

Abdeckung, und vorausgesetzt das Redundanz-Niveau ist r = 1 oder durch x_0 bereits erreicht, wäre x_{red} überflüssig.

Die Erfahrung mit einigen Programmdurchläufen hat gezeigt, dass trotz einer starken Gewichtung der Spezifität in *score* = w_1 *isens*(y, L) + w_2 *spez*(y), z.B. w_1 = 0.2 und w_2 = 0.8 die Einführung einer Mindest-Spezifität ratsam ist. Weiterhin kann kurz vor Beendigung des Algorithmus, d.h. wenn M bis auf wenige Ziel-Sequenzen abgedeckt ist, eine Mindest-Trefferanzahl bzw. Mindest-Sensitivität sehr nützlich sein. Diese verhindert, das Sequenzierungsfehler (→Sequenzierung) oder Oligonukleotide mit wenig Aussagekraft für die nicht-sequenzierten Individuen einer Population mit in eine Oligonukleotid-Bibliothek aufgenommen werden.

Für die Berücksichtigung der Treffer-Redundanz wurde in Abschnitt 4.2.2 die Abbildung m(t, L_i) eingeführt. Für jede Ziel-Sequenz t aus M und für eine Oligonukleotid-Teilbibliothek, $L_i \subset$ K wird m(t, L_i) die Anzahl der Treffer der Oligonukleotid-Bibliothek L_i auf der →Sequenz t gezählt. Zusammen mit dem Redundanz-Niveau r gilt dann eine Menge M erst dann als 100% sensitiv abgedeckt, wenn m(t, L_i) = r für alle t ∈ M ist, d.h. wenn jedes t ∈ M r-mal getroffen wurde. Beispielsweise hätte ein L mit r-1 zu 100% sensitiven Oligonukleotiden x (*sens*(x) =1) eine, die Treffer-Redundanz berücksichtigende, Sensitivität von $sens_r(L) = 0$. Andererseits ist, zusammen mit dem Toleranz-Niveau s, ein L gegenüber der Nichtziel-Klasse A dann zu 100% ($spez_s(L) = 1$) spezifisch, wenn kein t ∈ A mehr als s-mal getroffen wurde, d.h. fn = 0 (vorausgesetzt fp ≠ 0). Der folgende Algorithmus führt für jedes t ∈ M´ eine Zählvariable m(t) ein, die immer dann inkrementiert wird, wenn ein Oligonukleotid in L aufgenommen wird, das die Ziel-Sequenz t trifft.

Die Elemente der Mengen P und K müssen hier sorgfältig unterschieden werden und sie gehen beide als Eingaben in den Algorithmus ein. P wird zwar über P = *Match*(K) aus K berechnet, es kann aber |P| < |K| sein, wenn zwei verschiedene Elemente x_1, x_2 ∈ K dieselbe Treffermenge y = *Match*(x_1) = *Match*(x_2) haben. In diesem Fall darf dennoch nicht alleine mit y, wie in der für die Einführung im Abschnitt 2.5.1 vereinfachten Version des Greedy-Algorithmus, gearbeitet werden, da hier die verschiedenen Sequenzen von x_1 und x_2 benötigt werden: z.B. *sekOligo*(x_1) ≠ *sekOligo*(x_2).

Der modifizierte "Greedy Set Covering"-Algorithmus für ein beliebiges (M, P, A) „Set Cover“-Problem mit Spezifitäts-Nebenbedingung, maximal möglicher Optimierung bezüglich *sekOligo*(x) und *sekOligoZiel*(x, t) und der Berücksichtigung der Treffer-Redundanz mit dem →Redundanz-Niveau r und dem →Toleranz-Niveau s wird kurz skizziert:

1. Eingaben und Parameter übernehmen: M, P, A, K, g_Z (wird hier ähnlich den Grenzwerten für die Sekundärstruktur-Bewertungsfunktionen von *gZmax* bis *gZmin* durchlaufen), g_N, r, s, (*sekOligoMin*, *sekOligoMax*), (*sekOligoZielMin*, *sekOligoZielMax*), *sekOligoSchrankeDelta*, *sekOligoZielSchrankeDelta*, w_1, w_2
   Kommentar: P = *Match*(K) und die übrigen (thermodynamischen) Parameter wurden hier weggelassen.
2. Initialisierung: L = { } , m(t) = 0 für alle t ∈ M´ = M ∪ A
   sekOligoSchranke= *sekOligoMin* , *sekOligoZielSchranke* = *sekOligoZielMin*, g_Z = *gZmax*, ...
3. Bestimme für jedes x ∈ K die Spezifität *spez*(x) aus A, g_N, und dem Toleranz-Niveau s
4. Schleife 1: Durchlaufe (*sekOligoSchranke*, *sekOligoZielSchranke*) bis (*sekOligoMax*, *sekOligoZielMax*)

5. Schleife 2: Durchlaufe g_Z von *gZmax* bis *gZmin*
   6. Schleife 3: Greedy-Iterationen
      7. Bestimme für das L dieser Iteration und für jedes $x \in K$ die „inkrementelle Sensitivität“ $isens_r(x, L)$ aus M, g_Z und dem Redundanz-Niveau r
      8. Wähle das $x' \in K$ sodass $w_1\ isens_r(x', L) + w_2\ spez_s(x')$ maximal ist
      9. Wenn $sekOligo(x) < sekOligoSchranke$ und $sekOligoZiel(x', t) < sekOligoZielSchranke$, dann setze $L = L \cup \{x'\} \subset K$
      10. inkrementiere m(t) für alle neu getroffenen $t \in Match(x') \subset M'$
   11. Schleife 3 Ende: Wiederhole ab Schritt 7 bis für alle $x \in K$ gilt: $isens_r(x, L) = 0$, d.h $\bigcup_{x \in L} Match(x) \cup x' = \bigcup_{x \in L} Match(x)$ oder bis sich kein $x' \in K$ finden lässt, das den Kriterien entspricht
   12. Schleife 2 Ende: relaxiert g_Z für den nächsten Schleifendurchlauf
13. Schleife 1 Ende: relaxiert *sekOligoSchranke* und *sekOligoZielSchranke* für den nächsten Schleifendurchlauf mit den Schrittweiten *sekOligoSchrankeDelta* und *sekOligoZielSchrankeDelta*

Die für die Anwendungsbeispiele (siehe Abschnitt 7) verwendete Implementierung in Perl berechnete für alle →Sequenzklassen einer vorgegebenen Hierarchie die Oligonukleotid-Teilbibliotheken L_i zugleich, um einige Rechenschritte zu sparen. Dazu gehören die in Abbildung 4.1-3/unten dargestellten Anzahlen von rp- und fp-Treffern und rn- und fn-Nichttreffern, die auf diese Weise für verschiedene G_i nur einmal berechnet werden müssen und anschließend mal als rp- und mal als fp-Klasse in die Berechnungen eingehen. Für die Nichttreffer-Anzahlen rn und fn gilt dieses Argument analog.

Die Klausel hinter dem „oder“ von Schritt 11 „bis sich kein $x \in K$ finden lässt, das den Kriterien entspricht“ bedeutet, dass der Algorithmus durchaus mit einem L abbricht, das z.B. nur 70% Abdeckung von M hat, oder das für eine der Oligonukleotid-Teilbibliotheken L_i das →Redundanz-Niveau erreicht wurde und für eine andere nicht.

5.1.2. Einsatz des Greedy-Algorithmus

Der Greedy-Algorithmus hat aufgrund seiner Heuristik verhältnismäßig wenig Rechenaufwand. Daher kann er problemlos eingesetzt werden, um z.B. solche Parameter, die in der Aufgabenspezifikation (Abschnitt 4.3) vorausgesetzt wurden, automatisch zu bestimmen. Für das Anwendungsbeispiel zum Hepatitis C-Virus (siehe Abschnitt 7.1) wurde eine Optimierung nach der Schmelztemperatur mit dem Greedy-Algorithmus durchgeführt. Für 23 Schmelztemperatur-Intervalle von 49 bis 75°C wurden jeweils eine Oligonukleotid-Bibliothek mit mehreren Teilbibliotheken einer kleinen Hierarchie erstellt.

Hier wurde das Kriterium Spezifität gewählt, bezüglich dessen eine optimale Schmelztemperatur gefunden werden sollte. Anstelle der Spezifität hätte ebenso bezüglich der Sekundärstruktur-Eigenschaften optimiert werden können. Der Greedy-Algorithmus hat zwar bereits nach der Spezifität oder den Sekundärstruktur-Eigenschaften optimiert, aber eben nur im Rahmen der Vorgaben, z.B. im Rahmen der Menge der Oligonukleotid-Kandidaten. Beim Übergang zu einem anderen Schmelztemperatur-Intervall [minTm, maxTm] erhält man andere Oligonukleotid-Sequenzen, verschiedene Längen und GC-Gehalte. Die Abbildung

5.1-1 zeigt die Ergebnisse von 23 Durchläufen des Greedy-Algorithmus. Jeweils zu einer Position auf der x-Achse, die einem Schmelztemperatur-Intervall der Länge 4°C entspricht, wurde eine Bibliothek erstellt. Die y-Achse zeigt die Summe aller falsch-positiven Treffer aller Teilbibliotheken. Damit entspricht der kleinste Wert auf der y-Achse der Bibliothek mit der größten Spezifität.

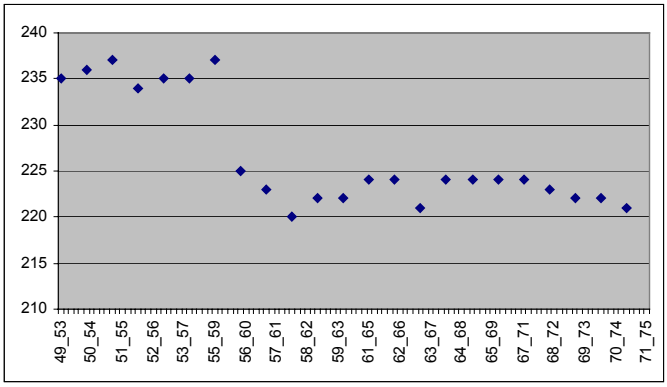


Abbildung 5.1-1: Summe aller falsch-positiven Treffer

Auf die für das Anwendungsbeispiel zum Hepatitis C-Virus bezogene Interpretation dieser Daten, z.B. der erhebliche Anstieg der Spezifität beim Überschreiten von 55°C, wird im Abschnitt 7.1 eingegangen. Als globalen Trend erkennt man erwartungsgemäß einen Anstieg der Spezifität von links nach rechts, da die mittlere Oligonukleotid-Länge der Bibliotheken von links nach rechts zunimmt. Auf der Ebene der →Sequenzen ist ein längeres Oligonukleotid stets spezifischer als ein kurzes. Dennoch darf die Schmelztemperatur oder die Oligonukleotid-Länge zur Optimierung der Spezifität nicht beliebig groß gewählt werden, da ein die Hybridisierungssignale betreffender Effekt die Spezifität des DNA-Mikroarrays mindert:

> "Sequence variation is best analyzed with the shortest oligonucleotides that will give specific hybridization to the target site. Lengths much shorter than 15-mer may find cross-hybridization with other sites. On the other hand, it is desirable to use short oligonucleotides for this purpose, to achieve good discrimination between variants, which, by definition, will be closely related in sequence. This may be difficult with →probes much longer than 15-mer. In this length region, it is necessary to carry out hybridization under nonstringent conditions of relatively high salt and low temperature." [89]

Mit zunehmender Oligonukleotid-Länge werden mehr Basenfehlpaarungen benötigt, um die Hybridisierungssignale zu diskriminieren. Das erzwingt die Wahl einer größeren Differenz $g_N - g_Z$ (vgl. Anschnitt 4.2.1) und mindert somit ebenfalls rechnerisch die Spezifität.

Bei der in dieser Arbeit durchgeführten Implementierung des Greedy-Algorithmus in Perl wurde zusätzlich die Möglichkeit geschaffen, die Menge L vorzugeben und einen Rest von fehlender Überdeckung durch weitere Oligonukleotide zu L hinzuzufügen. Dadurch kann eine Art manuelle Nachbearbeitung stattfinden, sollte sich ein Fänger-Oligonukleotid bei Hybridisierungs-Experimenten nicht wie vorhergesagt verhalten, oder sollte sich nach einer größeren →Kontrollrecherche (suche nach Treffern in G_0) ein Teil der Oligonukleotide als unbrauchbar erweisen. Dann können diese manuell aus L entfernt und die fehlende Abdeckung von M erneut und ohne größeren Rechenaufwand gewonnen werden.

5.2. Kombination von Gradientenabstieg und Kompetition

Bei der Optimierung mit →Gradientenabstiegs-Verfahren werden differenzierbare Kosten- bzw. Fehlerfunktionen $E: X \to \mathbb{R}$ definiert, mit Hilfe derer ein Suchraum X durchlaufen wird.

In dem Abschnitt 2.5.2 wurde dieser Ansatz bereits eingeführt und der Klasse der Verfahren für „lokale Suche“ [44] zugeordnet. Die Gradientenabstiegs-Verfahren sind ebenfalls iterative Verfahren und „lokale Suche“ bedeutet, dass bei jedem Iterations-Schritt nur ein Element des Suchraums X betrachtet wird. Die Elemente $\mathtt{L} \in X$ sind speziell kodierte Oligonukleotid-Bibliotheken. Ausgehend von einem Startzustand $\mathtt{L}_0$ springt das Verfahren gesteuert durch den Gradienten $-\nabla E$ von einer solchen Bibliothek zur nächsten bis ein nahezu optimales Ergebnis oder in einem „lokalen Minimum“ ein weniger optimaler Endzustand erreicht wird. Im letzteren Fall müsste das Verfahren mit einem anderen Startparameter wiederholt werden.

Die hier verwendete spezielle Kodierung für Oligonukleotid-Bibliotheken dient dazu, eine differenzierbare Fehlerfunktionen $E: X \rightarrow \mathbb{R}$ zu definieren. Die Zugehörigkeit eines Oligonukleotids zu einer Bibliothek wird graduell bzw. „fuzzy“ kodiert und berechnet. Dazu wird zunächst in einer Matrix der graduelle Beitrag eines Oligonukleotids $x \in K$ für das Treffen einer Ziel-Sequenz $t \in M$ dargestellt. Es wird für ein (M, P)-„*Set Cover*“-Problem mit $P = \mathit{Match}(K)$ eine Matrix $T \in \mathbb{R}^{|K|,|M|}$ definiert, d.h. die |M| Spalten entsprechen den Ziel-Sequenzen t und haben |K| Komponenten und die |K| Zeilen entsprechen den Oligonukleotiden x und haben |M| Komponenten. Die Komponenten der Matrix werden der einfacheren Lesbarkeit wegen nicht mit $T_{i,j}$ ($i = 1, ..., |K|$; $j = 1, ..., |M|$), sondern mit $T_{x,t}$ bezeichnet. Die (x, t) durchlaufen das kartesische Produkt $K \times M$. Ist $T_{x,t} = 0$, so trifft das Oligonukleotid x die Ziel-Sequenz t nicht oder x trifft, leistet jedoch keinen Beitrag die Ziel-Sequenz t effizient zu überdecken. Bei $T_{x,t} = 1$ ist der Beitrag dieses Treffers maximal und bei $T_{x,t} = 0.3$ gering, weil z.B. andere Oligonukleotide dieselbe Ziel-Sequenz treffen. Die Beiträge der Oligonukleotide in der Bibliothek für das gesamte (M, P)-Problem ergeben sich als Zeilensummen aus der Matrix T:

$$\text{Beitrag}(x) = \sum_{t \in M} T_{x,t} \quad ; \text{für alle } x \in K$$

$$\mathtt{x} = (\text{Beitrag}(x))_{x \in K} \in X = \mathbb{R}^{|K|}$$

Eine Visualisierung dieser Matrix mit weiß für 1 und schwarz für 0 ergibt eine Darstellung wie in Abbildung 2.5-3. Aus den Beiträgen resultieren die Zugehörigkeiten der Oligonukleotide zu der Bibliothek und somit das $\mathtt{x} \in X$. Anschließend wird zusammen mit einem Kriterium aus einem $\mathtt{x} \in X$ eine Oligonukleotid-Bibliothek $L \in \wp(K)$ berechnet, d.h. eine Funktion *defuzz*: $X \rightarrow \wp(K)$ angewendet.

Bei diesem Ansatz wurde kein reines Gradientenabstiegs-Verfahren angewendet, da gerade der Aspekt der kombinatorischen Optimierung nur schwer in einer Bewertungsfunktion zu modellieren ist. Zwar könnte die Bibliotheks-Größe durch einen Penalty-Term in die Funktion E eingehen, ein weiterer zu adjustierender Parameter und Probleme mit „lokalen Minima“ wären jedoch die Folge. Der hier vorgestellte Ansatz zielt gerade darauf ab, das in Abbildung 2.5-2 und Abbildung 2.5-3 dargestellte Problem, welches eine Art „Falle“ für den „Greedy Set Cover“-Algorithmus darstellt und zu einer suboptimalen Lösung führt, zu lösen. Dazu wird das Gradientenabstiegs-Verfahren mit einem die Kompetition zwischen Oligonukleotiden modellierenden Schritt kombiniert. Zwei x_1, x_2 oder mehr Oligonukleotide, die über ihre Treffer eine große Teilmenge von M zugleich abdecken, werden, zumindest bei einem →Redundanz-Niveau von $r = 1$, zu Konkurrenten. Auf der anderen Seite gibt es eine Kooperation zwischen Oligonukleotiden, deren Treffermengen sich wenig überschneiden:

$\mathit{Match}(x_1) \cap \mathit{Match}(x_2)$ groß $\Rightarrow$ x_1, x_2 konkurrieren
$\mathit{Match}(x_1) \cap \mathit{Match}(x_2)$ klein $\Rightarrow$ x_1, x_2 kooperieren

Der Aspekt der Kooperation und die Abdeckung von M durch L = *defuzz*($\mathfrak{X}$) wird durch die Fehlerfunktion E modelliert, und die Kompetition wird mittels eines während der Iterationen des Gradientenabstiegs durchgeführten Normierungsschritts bewirkt. Die Fehlerfunktion $E: X \to \mathbb{R}$ hat die Berechnungsvorschrift $E(\mathfrak{X}) = \sum_{x \in K} \sum_{t \in M} x_x \, T_{x,t}$. Diese Funktion ist zu maximieren. Sie nimmt ein Maximum bei $\mathfrak{X} = (1)_{|K|}$ (dem Vektor von |K| Einsen) an, was mit L = *defuzz*($\mathfrak{X}$) = K der Oligonukleotid-Bibliothek, die alle Kandidaten enthält, entspricht. Erst im Zusammenspiel mit der Kompetition unter den Oligonukleotiden ergibt sich eine sinnvolle Lösung. Die Abbildung zur Defuzzifizierung der Zugehörigkeitsgrade der Oligonukleotide *defuzz*: $X \to \wp(K)$ ist definiert als *defuzz*($\mathfrak{X}$) = $\{x \in K \mid \mathfrak{X}_x = 1\}$.

5.2.1. Algorithmus

Die folgende Skizze des Algorithmus für ein (M, P)-„*Set Cover*"-Problem stellt diesen Ansatz dar. Der Algorithmus ist zunächst ohne die Spezifitäts-Nebenbedingung, wie sie für ein (M, P, A)-Problem definiert ist (vgl. Abschnitt 4.1.3), angegeben. Auch die Sekundärstruktur-Optimierung und die Berücksichtigung der →Redundanz- und →Toleranz-Niveaus r und s wird hier zunächst ausgelassen. In Abschnitt 5.2.3 wird deren Berücksichtigung diskutiert. Die Grenzwerte g_Z und g_N (vgl. Abschnitt 4.2.1) gehen über die Definition der Menge *Match*(x) in die Matrix T ein.

1. Eingaben und Parameter übernehmen: M, P, K, Schrittweite η
   Kommentar: Es gilt P = *Match*(K).
2. Initialisierung:
   $T_{x,t} = 1$ falls $t \in$ *Match*(x), $T_{x,t} = 0$ falls $t \notin$ *Match*(x),
   $\mathfrak{X} = (0.5)_{|K|}$, $\Delta \mathfrak{X} = (0)_{|K|}$
3. Normierungsschritt: $(T_{x,t})_{x \in K} = (T_{x,t})_{x \in K} / \sum_{x \in K} T_{x,t} \in \mathbb{R}^{|K|}$ für alle $t \in M$
4. Schleife 1: Iterationen des Gradientenabstiegs
   5. Gradientenabstiegsschritt mit Schrittweite η:
      $\Delta \mathfrak{X}_x = \eta \sum_{t \in M} T_{x,t} = \frac{\partial E(x)}{\partial x_x} \in \mathbb{R}$ für alle $x \in K$
      $\mathfrak{X}_x = \mathfrak{X}_x + \Delta \mathfrak{X}_x$ für alle $x \in K$
      $\mathfrak{X}_x$ = Minimum(1, $\mathfrak{X}_x$) ; der Zugehörigkeitsgrad ist als maximal 1 definiert.
   6. Zugehörigkeits-Rückkopplung (damit wird die Information aus dem Gradientenabstieg der Matrix T zugeführt):
      $(T_{x,t})_{x \in K} = \mathfrak{X}_x \cdot (T_{x,t})_{x \in K}$
   7. für jedes Oligonukleotid das Maximum der entsprechenden Zeile aus T bestimmen:
      max_x = Maximum$\{ (T_{x,t})_{x \in K} \mid t \in M \}$
   8. Rückkopplung zur Verstärkung guter Kombinationen:
      Wenn $T_{x,t} \neq 0$ dann $T_{x,t} = max_x$
   9. Normierungsschritt: $(T_{x,t})_{x \in K} = (T_{x,t})_{x \in K} / \sum_{x \in K} T_{x,t} \in \mathbb{R}^{|K|}$ für alle $t \in M$

10. Abbruchbedingung: Anzahl maximaler Iterationen erreicht oder Sensitivität der Oligonukleotid-Bibliothek L = *defuzz*(ϰ) ist 100%.
11. Schleife 1 Ende: Falls Abbruchbedingung nicht erfüllt, zurück zu Schritt 5.
12. Ausgabe: L = *defuzz*(ϰ)

Dieser Algorithmus stellt einen verhältnismäßig wenig rechenaufwändigen Ansatz dar, der jedoch nicht, wie der heuristische Greedy-Algorithmus, auf die durch Abbildung 2.5-3 dargestellte Problematik „hereinfällt". Als Ansatz für „lokale Suche" ist dieser ebenfalls nicht so rechenaufwändig wie der Genetische Algorithmus, der mit einer ganzen Population von Oligonukleotid-Bibliotheken operiert.

5.2.2. Ein Anwendungsbeispiel

Das in Abbildung 5.2-1 dargestellte (M, P)-Problem besteht aus $|P| = |Match(K)| = |K| = 24$ Oligonukleotiden $K = \{x_1, x_2, ..., x_{24}\}$ und 50 Ziel-Sequenzen. Wie in der Abbildung 2.5-3 sind die Oligonukleotide auf der y-Achse aufgetragen und die Ziel-Sequenzen auf der x-Achse. Weiße Punkte stehen für Treffer der Oligonukleotide auf den Ziel-Sequenzen. Damit ergibt sich für jedes Oligonukleotid $x \in K$ durch die weißen Punkte auf der entsprechenden Zeile die Teilmenge *Match*(x), die einen Teil der Ziel-Sequenzmenge M überdeckt.

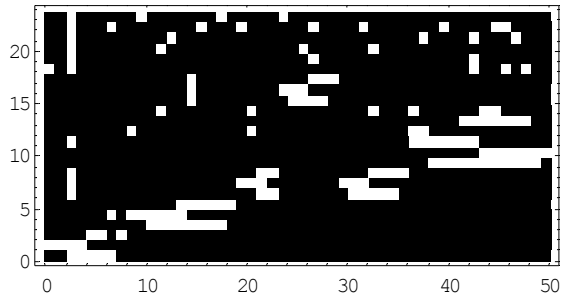


Abbildung 5.2-1: Anwendungsbeispiel Greedy vs. Gradientenabstieg

Das (M, P)-Problem enthält mehrere, nämlich fünf, Fallen, wegen der die Heuristik des Greedy-Algorithmus zu viele Oligonukleotide in die Bibliothek aufnehmen würde. Im unteren Teil erkennt man die fünf konstruierten Treffer-Muster und im oberen Teil wurden einige Treffer für ein leichtes Rauschen eingesetzt. Die Abbildung 5.2-1 kann nicht nur zur Darstellung eines „*Set Cover*"-Problems, sondern auch zur Visualisierung von dessen Lösung verwendet werden. Alle Oligonukleotide, die sich in der Lösungsmenge L befinden, werden wie oben beschrieben dargestellt, die anderen werden als schwarze Zeile eingefügt. Dieses Vorgehen entspricht exakt einer Visualisierung von T mit einem weißen Punkt für $T_{x,t} = 1$ und einem schwarzen Punkt für $T_{x,t} = 0$.

In der Abbildung 5.2-2/links ist das Ergebnis eines Greedy-Algorithmus, wie er in Abschnitt 2.5.1 angegeben wurde, dargestellt. Die Lösung $L_{Greedy} = \{x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{13}, x_{16}, x_{18}, x_{24}\}$ erreicht eine Sensitivität von 100% mit 14 Oligonukleotiden. Nur bei zweien der fünf konstruierten Fallen hat sich der Algorithmus wie erwartet verhalten (die Oligonukleotide mit den Indizes 1 bis 3 und 7 bis 9). Bei der Überdeckung der übrigen Sequenzen konnte auch der Greedy-Algorithmus überraschen und hat anstatt der erwarteten $15 = 3 \cdot 5$ Oligonukleotide nur 14 benötigt. In der Abbildung 5.2-2/rechts jedoch erkennt man mit $L_{GradDesc} = \{x_2, x_3, x_5, x_6, x_8, x_9, x_{11}, x_{12}, x_{17}, x_{18}\}$ und ebenfalls einer Sensitivität von 100% das bessere Abschneiden des Ansatzes mit der Kombination von Gradientenabstieg und Kompetition. Hier wurden nur 10 Oligonukleotide benötigt, was zugleich die optimale Lösung ist.

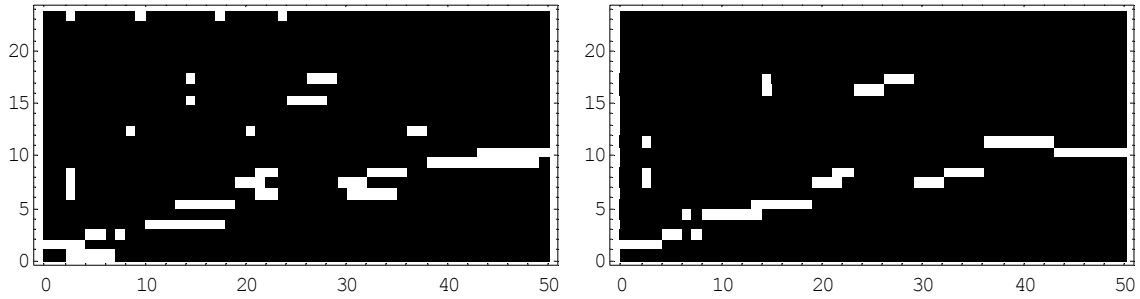


Abbildung 5.2-2: Vergleich der Ergebnisse: Greedy vs. Gradientenabstieg

5.2.3. Penalty-Terme für die übrigen Kriterien

In der oben angegebenen Skizze des Algorithmus wurden einige für die Erstellung von Oligonukleotid-Bibliotheken wichtige Nebenbedingungen weggelassen. Integriert in den, im Abschnitt 4.3 angegebenen, Rahmen-Algorithmus gehen jedoch bereits die Kriterien K1: $\text{minTm} \le T_m(x) \le \text{maxTm}$, K2: $\text{minLen} \le |x| \le \text{maxLen}$ und K3: $\text{minGC} \le \%GC(x) \le \text{maxGC}$ ein. Weiterhin kann über Option O1 ein Verfahren für die Definition von Treffern gewählt werden, das die Parameter $g_N, g_Z \in \mathbb{N}$ im Fall H(x, t) und $g_N, g_Z \in \mathbb{R}$ im Fall thdist(x, t) enthält. Die Möglichkeit zur Berücksichtigung einer Hierarchie zwischen Sequenzklassen ist, wie in Abschnitt 4.1.3 beschrieben, erst dann gegeben, wenn ein (M, P, A)-„*Set Cover*“-Problem mit Spezifitäts-Nebenbedingung gelöst werden kann. Auch die Sekundärstruktur-Optimierung und die Berücksichtigung der →Redundanz- und →Toleranz-Niveaus r und s wurden bisher ausgelassen.

Zur Integration einer Spezifitäts-Nebenbedingung kann die Fehlerfunktion $E: X \to \mathbb{R}$ um einen Penalty-Term für Treffer auf der Nichtziel-Klasse A erweitert werden. Die Berechnungsvorschrift lautet dann:

$$E(x) = \sum_{x \in K} \left(w_1 \sum_{t \in M} x_x \, T_{x,t} - w_2 \sum_{t \in A} x_x \, T_{x,t} \right) \text{ mit der Ableitung:}$$

$$\frac{\partial E(x)}{\partial x_x} = w_1 \sum_{t \in M} T_{x,t} - w_2 \sum_{t \in A} T_{x,t}$$

Die Parameter w_1 und w_2 sind Faktoren für die Gewichtung von Sensitivität und Spezifität. Wie auch schon bei dem Greedy-Algorithmus (vgl. Abschnitt 5.1.1) können die Oligonukleotid-Teilbibliotheken für die Sequenzklassen der Hierarchie zugleich berechnet werden, um Rechenaufwand zu sparen. $M' = M \cup A$ kann ebenfalls als disjunkte Vereinigung von Teilen der Sequenzklassen G_i dargestellt werden. Diese bilden in dem Gradientenschritt Teilsummen in der Ableitung der Fehlerfunktion, die mal positiv als rp-Klasse und mal negativ als fp-Klasse eingehen (siehe Abbildung 4.1-3/unten).

Für eine Sekundärstruktur-Optimierung sollte eine Strategie gewählt werden, die nur wenige Berechnungen von Sekundärstrukturen benötigt, da beide in Abschnitt 4.2.3 vorgestellten Versionen von Bewertungsfunktionen $\Delta\Delta G(x, t)$ und sek(x, t) sehr rechenaufwändig sind. Ein leicht zu realisierender Ansatz wäre eine erste Oligonukleotid-Bibliothek zu erstellen und diese mit einem harten Kriterium für die Sekundärstruktur zu filtern. Die Oligonukleotide mit einer guten Bewertung bleiben in der Bibliothek und werden in einer nächsten Iteration mit einem relaxierten Sekundärstruktur-Kriterium fest vorgegeben. Mit dem Ergebnis der zweiten Iteration wird genauso verfahren. Nach und nach verbleiben immer mehr Oligonukleotide in der Bibliothek. Bei maximal relaxiertem Kriterium wird sich eine Oligonukleotid-Bibliothek ergeben, die im Rahmen der vorgegebenen Daten eine maximale Sensitivität und Spezifität

hat und zudem aus möglichst vielen Oligonukleotiden mit einer bestmöglichen Sekundärstruktur-Bewertung besteht.

Für die Berücksichtigung der Redundanz- und Toleranz-Niveaus r und s kann ein zu den soeben beschriebenen Iterationen für die Sekundärstruktur-Optimierung ähnlicher Ansatz gewählt oder auch mit diesem kombiniert werden. Dann gehen r - 1 Treffer der vorgegebenen Oligonukleotide für die nächste Iteration nicht in die Kompetition und ebenfalls nicht in die Fehlerfunktion E ein. Weiterhin gehen s - 1 falsch-positive Treffer nicht in den für die Spezifitäts-Nebenbedingung hinzugefügten Penalty-Term ein.

5.3. Genetische Algorithmen

Aus dem Abschnitt 2.5.3 sind dem Leser noch die Begriffe zur Beschreibung des Modells der Genetischen Algorithmen [33], [74] bekannt. Eine Population besteht aus Individuen, deren Eigenschaften auf einem Chromosom kodiert sind. Die Individuen werden über Generationen (Iterationen) aufgrund ihrer Fitness (Bewertungsfunktion) selektiert und dadurch optimiert. Zufällige oder zielgerichtete Mutations- und Rekombinations-Operatoren verändern dabei die Eigenschaften der Individuen und sorgen somit für ein Durchlaufen des Suchraums. Eine Stärke der Genetischen Algorithmen, die bereits für molekularbiologische Aufgabenstellungen eingesetzt wurden [53], ist gerade die kombinatorische Optimierung. Einige der Schwächen sind ein hoher Rechenaufwand und die Notwendigkeit viele Modellparameter, wie die Populationsgröße und die Mutations- und Rekombinations-Rate anpassen zu müssen. Verwendung finden Genetische Algorithmen vor allem bei sehr großen (teils diskreten) Suchräumen mit vielen lokalen Minima auf der Bewertungsfunktion. Sollten typische Konfigurierungsprobleme für Oligonukleotid-Bibliotheken diesem Problemtyp entsprechen, dann müssten die Ergebnisse der Genetischen Algorithmen besser als die des Greedy- oder des Gradientenabstiegs-Verfahrens sein.

Da bei dem Genetischen Algorithmus mit einer Start-Population von Oligonukleotid-Bibliotheken begonnen wird, ist es ein Problem, initiale Bibliotheksgrößen festzulegen. Bei dem Greedy-Algorithmus ergibt sich die Anzahl der Oligonukleotide |L| mit dem Abbruchkriterium. Wenn es erfüllt ist, werden keine weiteren Oligonukleotide hinzugefügt. Und auch bei dem Ansatz über die Kombination von Gradientenabstieg und Kompetition ergibt sich die Bibliotheksgröße durch die Anzahl der Oligonukleotide x, deren Zugehörigkeitsgrad den Zahlenwert 1 erreicht: $|L| = |defuzz(\mathbb{X})|$. Ein naheliegender Ansatz für Genetische Algorithmen ist die Start-Population mit Bibliotheken verschiedener Oligonukleotid-Anzahlen zu konstruieren und die Mutations- und Rekombinations-Operatoren so zu definieren, dass sie die Größe der Bibliotheken ändern. Da eine größere Oligonukleotid-Bibliothek im Allgemeinen stets mehr Ziel-Sequenzen trifft als eine kleinere, ist von vornherein die Notwendigkeit gegeben, einen Penalty-Term für die Bibliotheksgröße in die Bewertungsfunktion der Individuen zu integrieren.

Bei dem Greedy-Algorithmus wurde mit der „inkrementellen Sensitivität“ *isens*(x, L) der Zugewinn eines potentiellen neuen Oligonukleotids $x \in K$ bei vorgegebenem Zwischenergebnis L bewertet. Bei dem Gradientenabstiegs-Verfahren wurde eine Fehlerfunktion für Vektoren von Zugehörigkeitsgraden zu einer „*fuzzy*“ Bibliothek verwendet. Neu an dem Ansatz über Genetische Algorithmen ist, das konkrete Oligonukleotid-Bibliotheken als ganzes bewertet werden. Dieses steht im Gegensatz zu dem Ansatz des Greedy-Algorithmus, die „Fitness“ einzelner Oligonukleotide zu betrachten. Die folgende „Formel“ stellt den Unterschied dieser beiden Ansätze einprägsam dar:

$$\text{Summe Fitness(oligo)} \neq \text{Fitness Summe(oligos)}$$

Die linke Seite steht für die Optimierung einzelner Oligonukleotide und damit für den Greedy-Algorithmus, auf der rechten Seite steht „Summe(oligos)“ für eine Oligonukleotid-Bibliothek. Mit einer die ganze Bibliothek berücksichtigenden Bewertungsfunktion (Quantifizierung von Fitness) ergibt sich ganz natürlich eine Kooperation und über einen Penalty-Term für die Bibliotheksgröße eine Kompetition zwischen den Oligonukleotiden. Fügt man ein „<“-Zeichen in die Formel ein, dann ergibt es das Sprichwort „Das Ganze ist mehr als die Summe seine Teile“.

5.3.1. Algorithmus

Die folgende Skizze des Algorithmus für ein (M, P, A)-„Set Cover“-Problem mit Spezifitäts-Nebenbedingung (vgl. Abschnitt 4.1.3) ist zunächst ohne die Optimierung bzgl. der Sekundärstrukturen angegeben, da diese den Kern des Algorithmus unnötig verkomplizieren. Die Berücksichtigung der Redundanz- und Toleranz-Niveaus r und s ist implementiert und die Grenzwerte g_Z und g_N (vgl. Abschnitt 4.2.1) gehen über die Definition der Abbildung Match(x) in die Berechnung der Sensitivität und Spezifität der gesamten Bibliothek ein.

Gegenüber den anderen beiden Ansätzen kommen bei dem Genetischen Algorithmus zahlreiche weitere Parameter hinzu: die Populationsgröße *nPop*, die Mutations-Rate *mutRate* und die Rekombinations-Rate *recombRate* und die maximale Anzahl der Generationen bzw. Iterationen *maxGen*. Häufig kommen auch Parameter für ein „simulated annealing“ zum Einsatz. Bei diesem „simulierten Abkühlen“ werden die Parameter vermindert, die eine starke Dynamik (=Temperatur) in der Population bewirken. Bei einem Genetischen Algorithmus sind die Mutations-Rate *mutRate* und die Rekombinations-Rate *recombRate* zwei solche Parameter, die bei der n-ten Iteration mit einem Funktionswert *mutRate*(n) und *recombRate*(n), definiert durch zwei monoton fallende Funktionen, eingehen könnten. Die Verwendung von verschiedenen Versionen von Mutations- und Rekombinations-Operatoren, etwa für die zielgerichteten Versionen, erhöhen ebenfalls die Anzahl der Parameter. In einer Version ohne automatischer Suche nach der Bibliotheksgröße *bibSize* muss diese zusätzlich als Parameter vorgegeben werden, sonst wird ein Intervall [*bibSizeMin*, *bibSizeMax*] von Größen für die Generierung der Startpopulation benötigt.

1. Eingaben und Parameter übernehmen: M, P, A, K, r, s,
   nPop, *mutRate*, *recombRate*, *maxGen*, *bibSize*.
   w_1, w_2 Faktoren für die Gewichtung von Sensitivität und Spezifität
   Kommentar: g_Z und g_N gehen über *Match*(x) in die Berechnung von P ein.
2. Initialisierung: *gen* = 0 die Iterationen-Zählvariable,
   Pop = <zufällige Anfangspopulation mit *nPop* Individuen L_i >
3. Schleife 1: ... über die Generationen: *gen* = *gen* + 1
   4. Bestimme für jedes L_i aus der Population die Fitness:
      Berechne nach den in Abschnitt 4.2.2 definierten Werten für rp, fn, rn und fp für eine ganze Oligonukleotid-Bibliothek die Sensitivität $sens_r(L_i)$ und die Spezifität $spez_s(L_i)$.
      Kommentar: Hier geht neben r und s auch g_Z und g_N ein.
      Fitness = $w_1\, sens_r(L_i) + w_2\, spez_s(L_i)$
   5. In Abhängigkeit von der Fitness werden bestimmte Teile der Population für die Reproduktion selektiert und dabei den Operationen Mutation (*mutRate*) und Rekombination (*recombRate*) unterzogen; andere Teile der Population werden durch die Reproduzierten ersetzt (Selektion, Mutation, Rekombination), sodass die Populationsgröße *nPop* konstant bleibt.

6. Abbruchbedingung: hinreichende Fitness erreicht oder eine maximale Anzahl von Generationen *maxGen* durchlaufen
7. Schleife 1 Ende: Falls Abbruchbedingung nicht erfüllt, zurück zu Schritt 4.
8. Ausgabe: $L = L_i$ mit $Fitness(L_i) \geq Fitness(L_j)$ für alle L_j aller über die Generationen erzeugten Populationen

Falls keine unveränderten Eliten (engl.: *Elitism*) [74] verwendet werden, wird im Schritt 8 das Individuum mit der größten Fitness, über alle durch die Generationen erzeugten Populationen, gewählt und nicht das beste Individuum der letzten Population. Denn durch die Mutations- und Rekombinations-Operatoren kann ein sehr „fittes“ Individuum jederzeit verschlechtert werden.

5.3.2. Anwendungsbeispiele

Zu dem in Abbildung 5.2-1 dargestellten Anwendungsbeispiel hat der Genetische Algorithmus auf Anhieb, d.h. ohne Ausprobieren mehrerer Parameter, ebenso wie die Kombination von Gradientenabstieg und Kompetition die optimale Lösung $L_{GenAlg} = \{x_2, x_3, x_5, x_6, x_8, x_9, x_{11}, x_{12}, x_{17}, x_{18}\}$ gefunden. Die Parameter waren: *nPop* = 50, *mutRate* = 3/4, *recombRate* = 1/2, *maxGen* = 100 und r = 1. Die Größe der Oligonukleotid-Bibliotheken war in einem ersten Testlauf mit *bibSize* = 10 vorgegeben. Der Parameter g_Z wird hier nicht benötigt, da dieses Anwendungsbeispiel bereits mit einer vorgegebenen Menge P vollständig definiert ist. Normalerweise würde P aus g_Z und g_N zusammen mit thdist(x, t) oder H(x, t) über P = *Match*(K) berechnet werden. Weiterhin werden die Parameter g_N, s, w_1 und w_2 nicht benötigt, da es sich um ein (M, P)-„Set Cover“-Problem ohne einer Menge A von →Nichtziel-Sequenzen handelt.

Bei der Anwendung des Genetischen Algorithmus auf ein (M, P, A)-Problem mit der Nebenbedingung „maximierte Spezifität“ bzgl. der Menge der Nichtziel-Sequenzen A wurde bei dem Durchlauf I mit 113 von 118 Ziel-Sequenzen des Genotyps 1a des Hepatitis C-Virus (vgl. Abschnitt 7.1) eine leicht geringere Sensitivität als bei einer mit dem Greedy-Algorithmus konstruierten „Genotyp 1a“-Teilbibliothek gefunden. Der Durchlauf II erzielte mit 115 Treffern eine etwas bessere Sensitivität. Die Spezifität war, gemessen durch die Summe der falsch-positiven Treffer, beim Durchlauf I besser und beim Durchlauf II schlechter. Die Tabelle 5.3-1 stellt die Anzahlen der falsch-positiven Treffer bezüglich der einzelnen Genotypen 1b, 4, 3 und 2 dar.

Tabelle 5.3-1: Anzahlen falsch-positiver Treffer einer „Genotyp 1a“-Teilbibliothek

Genotypen	1b	4	3	2
Greedy-Algorithmus	27	27	0	78
Genetischer Algorithmus I	44	25	0	2
Genetischer Algorithmus II	43	31	0	2
Genetischer Algorithmus III	37	27	0	77

Gegenüber Genotyp 1b ist die Spezifität der Oligonukleotid-Teilbibliothek bei drei Durchläufen des Genetischen Algorithmus vermindert, gegenüber Genotyp 4 nahezu und gegenüber Genotyp 3 unverändert. Eine deutlich bessere Spezifität gegenüber Genotyp 2 ergibt sich nur bei den Durchläufen I und II.

Die durch mehrere Testläufe optimierten Parameter des Durchlaufs I des Genetischen Algorithmus waren: *nPop* = 100, *mutRate* = 3/4, *recombRate* = 1/2, *maxGen* = 200, r = 1, s = 0, w_1 = 0.8 und w_2 = 0.2 sowie die Grenzwerte für H(x, t) g_Z = 0 und g_N = 1. Die Größe der Oli-

gonukleotid-Bibliotheken war mit *bibSize* = 9 vorgegeben und damit identisch zu der durch den Greedy-Algorithmus gefundenen Bibliotheksgröße. Der Durchlauf II unterschied sich zu I nur in dem Grenzwert $g_N = 0$. Damit ergeben sich definitionsgemäß weniger falsch-positive Treffer, und indirekt wird somit weniger Gewicht auf die Optimierung der Spezifität gelegt. Bei dem Durchlauf III wurde eine Bibliotheksgröße von *bibSize* = 10 vorgegeben.

5.3.3. Integration weiterer Kriterien

Die Optimierung nach der Sekundärstruktur ist wegen des hohen Rechenaufwands ein besonderes Problem. Die Berechnung von Sekundärstrukturen sollte möglichst selten durchgeführt werden. Bei dem Greedy-Algorithmus konnte dieses Problem recht einfach gelöst werden, da *nur eine* Oligonukleotid-Bibliothek konstruiert wird und die Aufnahme eines Oligonukleotids in die Bibliothek ein seltenes Ereignis ist. Bei dem Genetischen Algorithmus existiert bereits bei der ersten Iteration eine ganze Population von vollständigen Oligonukleotid-Bibliotheken. In diesem Fall musste ein Verfahren entwickelt werden, das die Berechnung von Sekundärstrukturen dennoch weniger oft durchführt.

Auf solche Individuen L_i der Population, die eine Mindest-Fitness erreicht haben, können ähnlich dem Ansatz in dem Greedy-Algorithmus die Grenzwerte *sekOligoSchranke* und *sekOligoZielSchranke* angewendet werden. So wird bei einer hoch angesetzten Mindest-Fitness auch hier die Anwendung dieses Kriteriums zu einem seltenen Ereignis. Die Oligonukleotide, die das Sekundärstruktur-Kriterium nicht erfüllen, werden aus der Menge der Oligonukleotid-Kandidaten K und aus jeder Oligonukleotid-Bibliothek L_i der Population entfernt. Die so verminderte Fitness der L_i muss anschließend durch weitere Generationen und zusammen mit anderen Oligonukleotiden wiedergewonnen werden. Kommt der Genetische Algorithmus bei einem stringenten Grenzwert nicht zu einem guten Ergebnis bezüglich Sensitivität und Spezifität, dann wird die Grenzwert schrittweise relaxiert.

So zahlreich wie die Parameter der Genetischen Algorithmen, so zahlreich sind auch Ansätze, die Performance des Algorithmus durch Varianten zu verbessern. Es gibt das Konzept der Inselbildung [74], bei der Teile der Population separat oder mit sehr geringen Rekombinations-Raten zu den anderen Populationsteilen evolvieren. Ein weiterer Ansatz die Performance zu verbessern, wäre mit einer „voroptimierten Startpopulation“ zu arbeiten. Dazu könnte der Greedy-Algorithmus und die Kombination von Gradientenabstieg und Kompetition verwendet werden. Letzterer würde initialisiert mit $\chi = (0.5 + \text{Random}[\delta])_{|K|}$ viele verschiedene Individuen für die Startpopulation generieren.

6. Das Optimierungs-Programm – optiNA „optimal Nucleic Acids“

Zusammenfassung: Das Optimierungs-Programm **optiNA** „optimal Nucleic Acids“ soll den Molekularbiolog(inn)en einen leichten Umgang mit dem Programm ermöglichen. Nach einer kurzen Darstellung der internetbasierten Systemarchitektur (Apache, MySQL, Perl, PHP und Bioinformatik-Tools) wird auf den Ablauf, die Bedienung und die Benutzungsoberfläche eingegangen. Es wird beschrieben, wie mit **optiNA** neue Versionen von Oligonukleotid-Bibliotheken erstellt werden können

In **optiNA** werden die Eigenschaften der berechneten Oligonukleotid-Bibliothek und der einzelnen Oligonukleotide mit Hilfe von Visualisierungen zur Treffer-Statistik und Sekundärstrukturen und Ausgaben von Tabellen von Trefferanzahlen und Zahlenwerten von Bewertungsfunktionen veranschaulicht. In dem Abschnitt 6.3 werden diese Visualisierungen und Tabellen sowie deren Interpretation beschrieben.

Das Optimierungs-Programm **optiNA** „optimal Nucleic Acids“ soll den Molekularbiolog(inn)en einen leichten Umgang mit dem Programm ermöglichen. Nicht zu unterschätzen ist der Wunsch verschiedene Parameter „ausprobieren“ zu können, um anschließend zu sehen, wie sich das Ergebnis ändert (vgl. dazu Abschnitt 8.2 zu ROC-Curves). **optiNA** wurde als internetbasierte Anwendung in PHP implementiert und ist damit ohne den Aufwand einer lokalen Installation des Programms von einem Rechner mit Internetanschluss nutzbar.

Eine Benutzerverwaltung mit Login und Passwort regelt den Zugriff auf den Server, der die Anwendung unter der Adresse http://home.zait.uni-bremen.de/~ellola/gensensorik/configTool/ bereitstellt. Ein Session-Management sorgt für eine ausgewogene Auslastung der Rechenkapazitäten des Servers. Der Benutzer wird unterstützt durch Datenbankfunktionalität, statistischen Auswertungen zu der berechneten Oligonukleotid-Bibliothek und Visualisierungen der Oligonukleotid-Positionen auf den Sekundärstrukturen. Als Arbeitspaket im Rahmen des FuE-Verbundes Gensensorik wurde zudem eine direkte Schnittstelle zum Mikropipettiersystem für die Erstellung der DNA-Mikroarrays erstellt, die jedoch bisher nicht in **optiNA** integriert wurde.

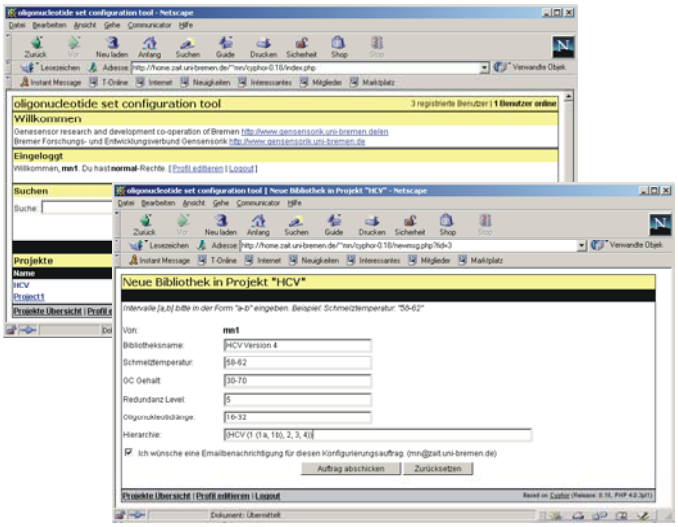


Abbildung 5.3-1: Screenshot des Internet-basierten Optimierungs-Programms

Die Oberfläche stellt ein „Front End“ zu dahinterliegenden Programmen dar. Einige Funktionalitäten sind nicht über die Oberfläche, sondern ausschließlich über den Aufruf auf der Unix-Kommandozeile verfügbar. Für den Greedy-Algorithmus gibt es beispielsweise teilautomatisiert die Möglichkeit, Oligonukleotid-Bibliotheken vorzugeben, um anschließend Oligonukleotide „hinzu zu konfigurieren“. Dieses Feature ist für eine manuelle Nachbearbeitung einer Oligonukleotid-Bibliothek sehr nützlich.

6.1. Systemarchitektur

Die Oberfläche wurde in PHP und die Optimierungs-Algorithmen in Perl kodiert. Web-Server, Compute-Server und der Server für die SQL-Datenbank können sich auf verschiedenen Rechnern befinden. Die Abbildung 6.1-1 stellt die Komponenten dieser Architektur und deren Zusammenspiel dar.

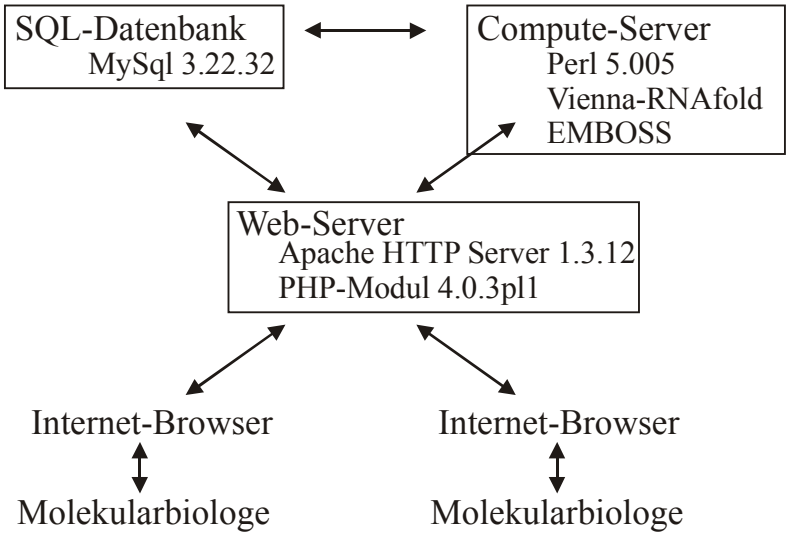


Abbildung 6.1-1: Systemarchitektur von optiNA

Auf dem Compute-Server ist Vienna RNAfold von Ivo Hofacker [42], [110], Perl und EMBOSS installiert. Auf diesem laufen die Optimierungs-Algorithmen in Perl. Zur Berechnung von sek(x, t) und $\Delta\Delta G(x, t)$ (vgl. Abschnitt 4.2.3), die Bewertungsfunktionen zu den Sekundärstrukturen, wurde die Perl-Bibliothek von Vienna RNAfold verwendet. Zur Berechnung der Schmelztemperaturen wurde ein Tool aus der „*European Molecular Biology Open Software Suite*“ EMBOSS benutzt.

Die Oberfläche läuft in einem PHP-Modul des Apache HTTP Servers auf dem Web-Server. Die PHP-Skripte auf dem Web-Server greifen für die Benutzerverwaltung und das Session-Management auf die SQL-Datenbank zu. Daten zu Ergebnissen der Optimierungs-Algorithmen werden im Dateisystem des Web-Servers gespeichert. In einer späteren Version wird dazu ebenfalls die SQL-Datenbank verwendet.

6.2. Ablauf, Bedienung und Benutzungsoberfläche

Möchte ein Benutzer von **optiNA** eine Oligonukleotid-Bibliothek erstellen, so muss er sich zunächst in dem System mit einem Login-Namen und einem Passwort anmelden. Auf der Startseite des Programms werden daraufhin die öffentlichen Projekte und die Projekte des Benutzers aufgelistet. In einem Projekt werden die Daten und mehrere Versionen von Oligonukleotid-Bibliotheken zusammengefasst. Arbeitet ein Benutzer gleichzeitig an der Erstellung von DNA-Mikroarrays für einen Virus, eine Bakteriengruppe und verschiedenen Algen, so wird er drei Projekte anlegen. Mit einem Mausklick auf ein Projekt in der Projekt-

Liste wird eine Seite mit der Liste der bisher erstellten Versionen von Oligonukleotid-Bibliotheken angezeigt.

Mit „*Neue Version erstellen*“ kann ein Auftrag zur Berechnung einer neuen Oligonukleotid-Bibliothek erstellt werden. Dazu erscheint ein Formular in dem ein Bibliotheksname und jeweils ein Intervall für Schmelztemperatur, GC-Gehalt und Oligonukleotid-Länge eingegeben wird. Weiterhin wird das →Redundanz- und →Toleranz-Niveau und die Hierarchie der →Sequenzklassen in Klammernotation eingegeben. Nach dem Absenden dieses Formulars wird ein zweites und letztes Formular angezeigt, in dem für jeden Knoten der Hierarchie eine Menge von →Ziel-Sequenzen eingegeben oder hochgeladen werden kann. Bei Betätigung der Schaltfläche „*Auftrag abschicken*“ wird dieser in eine Liste von Aufträgen mit dem Status „*pending*“ für anstehend/ unerledigt eingetragen. Der Benutzer bekommt die Annahme des Auftrags zusammen mit einer Identifikations-Nummer angezeigt und wird aufgefordert, nach einigen Stunden die Ergebnisse abzufragen.

Der Compute-Server startet, in Abhängigkeit von seiner Auslastung, einen oder mehrere Aufträge aus der Auftrags-Liste mit dem Status „*pending*“ und wechselt den Status zu „*running*“. Der Status „*running*“ wird dem Benutzer angezeigt, sollte er die Ergebnisse zu früh abfragen wollen. Nach Beendigung des Optimierungs-Algorithmus wird der Status zu „*finished*“ gewechselt und die neue Bibliotheks-Version in die Liste des zugehörigen Projekts eingetragen.

Zur Zeit der Abgabe dieser Arbeit befand sich die Oberfläche zu **optiNA** noch in der Entwicklung. Es ist jedoch geplant, mit einem Mausklick auf die Bibliotheks-Version eine HTML-Tabelle mit Oligonukleotiden anzuzeigen, die durch Verknüpfungen (*Hyper-Links*) einen leichten Zugriff auf verschiedene Informationen ermöglicht. Dazu gehören Verknüpfungen zur Treffer-Statistik, zu Sekundärstruktur-Visualisierungen und zu einer automatisch durchgeführten →Kontrollrecherche.

6.3. Visualisierungen und Tabellen

In **optiNA** werden die Eigenschaften der Oligonukleotid-Bibliothek und der einzelnen Oligonukleotide mit Hilfe von Visualisierungen zur →Treffer-Statistik und Sekundärstrukturen sowie Ausgaben von Tabellen von Trefferanzahlen und Zahlenwerten von Bewertungsfunktionen veranschaulicht. Die folgenden Abschnitte beschreiben diese Visualisierungen und Tabellen und deren Interpretation, die zum Zeitpunkt der Abgabe dieser Arbeit nicht vollständig in **optiNA** integriert waren.

6.3.1. Sensitivitäten und Spezifitäten

Für die Bewertung der Sensitivitäten und Spezifitäten einzelner Oligonukleotide wurden bisher die Abbildungen *sens*(x) und *spez*(x) eingeführt und für ganze Oligonukleotid-Bibliotheken $sens_r(L)$ und $spez_s(L)$. Die Berechnungen dieser Abbildungen basieren auf mehreren Definitionen der Zahlenwerte rp, fn, rn und fp. Neben der Verwendung als Bewertungsfunktionen in den Optimierungs-Algorithmen leisten diese Abbildungen eine Reduktion einer großen Menge von Eigenschaften auf wenige überschaubare Zahlenwerte.

So kann beispielsweise die manuell erstellte Oligonukleotid-Bibliothek für den Hepatitis C-Virus (siehe Abschnitt 7.1) und die mit **optiNA** erstellte Oligonukleotid-Bibliothek bei verschiedenen Niveaus von r und s verglichen werden. Dabei wird jedoch eine starke Datenreduktion durchgeführt und nur die Situation für jeweils einen Satz von Parametern (z.B. r=3 und s=0) betrachtet. Da die vier oben genannten Abbildungen von den Parametern r, s, g_Z und g_N abhängen, ist die Betrachtung einer ganzen Sequenz ($sens_1(L)$, $sens_2(L)$, $sens_3(L)$, ...) von Sensitivitäten bezüglich mehrerer Redundanz-Niveaus r =1, 2, 3, ... eine exaktere Bewertung der Eigenschaften einer Oligonukleotid-Bibliothek. Dieses Verfahren kann ebenfalls auf

$spez_s$(L) und auf die von g_Z und g_N abhängigen Abbildungen *sens*(x) und *spez*(x) angewendet werden. Dabei ergäbe sich wiederum eine große Menge von Zahlenwerten, und die Überschaubarkeit ist reduziert.

Ein Kompromiss zwischen Datenreduktion/ Überschaubarkeit und maximalem Informationsgehalt ist die Visualisierung der Treffer der Oligonukleotide auf den Ziel-Sequenzen als Matrix in der Abbildung 6.3-1. Ähnlich der Abbildung 2.5-3 und der Abbildung 5.2-1 entsprechen die Zeilen den Oligonukleotiden und die Spalten den Ziel-Sequenzen. Dies veranschaulicht der obere Teil der Abbildung, in dem ein Bereich von drei Genotyp-4-spezifischen Oligonukleotiden für eine Teilmenge der Ziel-Sequenzen vergrößert wurde. In dieser Grafik können Mehrfachtreffer abgelesen werden, aus denen sich die →Oligonukleotid-Redundanz ergibt. Weiterhin können direkt die Treffermengen *Match*(x_1) und *Match*(x_2) zweier Oligonukleotide x_1 und x_2 abgelesen werden; diese lassen erkennen, ob sie im Sinne der in Abschnitt 5.2 diskutierten Begriffe Kooperation und Kompetition gut kooperieren, konkurrieren oder ganz einfach für das Erreichen des →Redundanz-Niveaus notwendig sind.

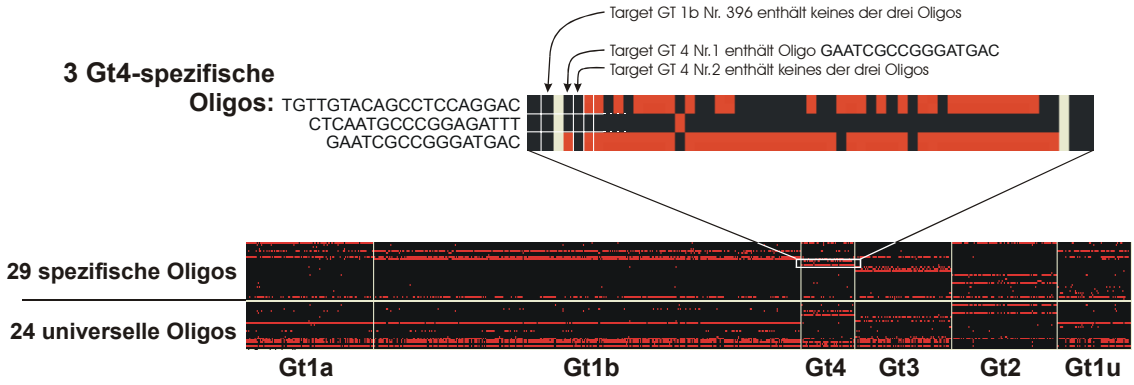


Abbildung 6.3-1: Visualisierung der Treffer

Durch Unterteilung der x-Achse der Matrix in der Abbildung 6.3-1 werden zusätzlich die Sequenzklassen dargestellt. Die horizontale Linie teilt die Zeilen bzw. Oligonukleotide in 29 genotypspezifische und 24 HCV-universelle Oligonukleotide. Während die Abbildung 6.3-1 nur „perfect match“-Treffer darstellt, sind in der Abbildung 6.3-2 neben „perfect match“-Treffern (weiß) ebenfalls 1-Mismatch-Treffer (→Mismatch) blau und 2-Mismatch-Treffer rot dargestellt.

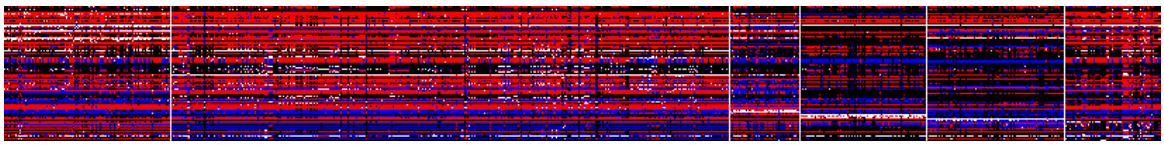


Abbildung 6.3-2: Visualisierung von Mismatch-Treffern

Für nicht zu große Mengen von Oligonukleotid-Kandidaten und Ziel-Sequenzen kann eine Darstellung, wie in Abbildung 6.3-2, auch vor der Erstellung einer Oligonukleotid-Bibliothek betrachtet werden, um Werte für die Grenzen g_Z und g_N im Fall von H(x, t) zu ermitteln (vgl. Abschnitt 4.2). Ebenfalls dargestellt werden Tabellen von richtig-positiven und falsch-positiven Trefferanzahlen. Diese werden im Kapitel 7 zusammen mit dem Anwendungsbeispiel zum Hepatitis C-Virus erläutert.

6.3.2. Visualisierung der Sekundärstrukturen

Die von **optiNA** berechnete Oligonukleotid-Bibliothek wird als HTML-Tabelle dargestellt. Diese enthält neben dem Oligonukleotid-Namen, dessen →Sequenz, GC-Gehalt, Schmelz-

temperatur und weiteren Zahlenwerten von den Bewertungsfunktionen auch eine HTML-Verknüpfung zu einer Postscript-Datei. Diese visualisiert die Sekundärstruktur und die Position des Oligonukleotids auf dieser durch farbliche Hervorhebung. Die Postscript-Datei wurde mit Vienna RNAfold [110], [42] erzeugt und anschließend für die farbliche Hervorhebung nachbearbeitet.

Die Abbildung 6.3-3 stellt die HTML-Tabelle und die Visualisierungen der Sekundärstrukturen dar. In die HTML-Tabelle können auf ähnlicher Weise HTML-Verbindungen zu einer detaillierteren Merkmalsbeschreibung der Oligonukleotide und zu Ergebnissen einer automatisch durchgeführten BLAST- oder FASTA-Recherche eingesetzt werden.

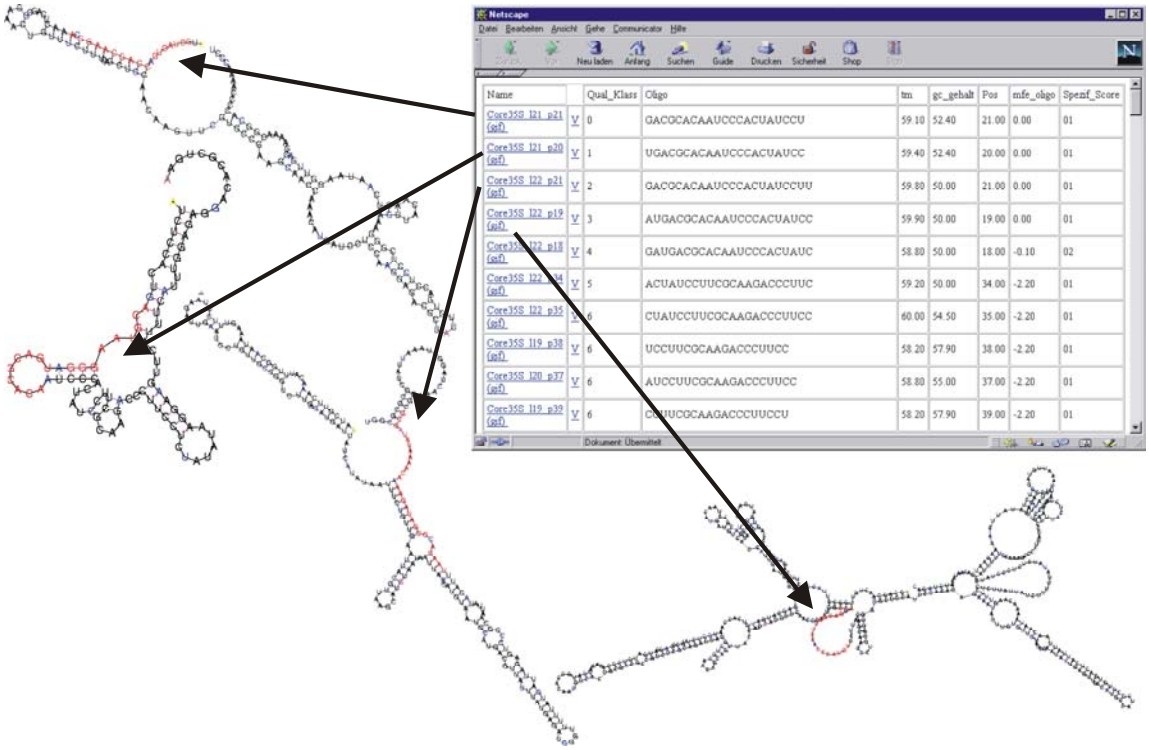


Abbildung 6.3-3: Visualisierung der Sekundärstrukturen in optiNA

7. Anwendungen und Ergebnisse

Zusammenfassung: Die hier vorgestellten Anwendungen, die Identifikation von Hepatitis C-Viren (HCV) mit DNA-Mikroarrays und ein Projekt zur Organismen-Identifikation „Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“, sind in Zusammenarbeit mit dem FuE-Verbund Gensensorik an der Universität Bremen durchgeführt worden. An den Ergebnissen des HCV-Projektes wird die Wirkungsweise der Bewertungsfunktionen des Abschnitts 4.1 zur Bewertung der theoretischen →Spezifität dargestellt. Am Projekt zur Organismen-Identifikation „Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“ kann die Bewertung der Spezifität als Differenz von Hybridisierungseffizienzen betrachtet werden. Die Funktion der in Abschnitt 4.2 vorgestellten Bewertungsfunktionen für die Hybridisierungseffizienz und Sekundärstrukturen wird damit verdeutlicht. Weiterhin können die unterschiedlichen Signalstärken innerhalb der Gruppe der richtig-positiven Signale anhand der Bewertungsfunktionen interpretiert werden.

Eine manuell erstellte Oligonukleotid-Bibliothek zur Identifikation von Hepatitis C-Viren wird mit zwei Versionen von automatisch mit **optiNA** erstellten Oligonukleotid-Bibliotheken verglichen. Dabei wurden für eine bessere Vergleichbarkeit Parameter, wie z.B. die Schmelztemperatur T_m konstant gelassen, obwohl ein softwarebasierter Ansatz das Potenzial hat, solche Parameter optimiert zu wählen (vgl. Abbildung 5.1-1).

Die manuell konfigurierte Oligonukleotid-Bibliothek wurde über einen Zeitraum von mehreren Monaten mit den Ende 1999 im Internet verfügbaren Diensten für Sequenzretrieval und Primer-Design erstellt. Mehrere Versionen der softwarebasiert konfigurierten Oligonukleotid-Bibliotheken wurden dagegen jeweils in wenigen Wochen erstellt. Zwei dieser Versionen werden im Abschnitt 7.1.2 vorgestellt. Trotz der Optimierung bezüglich mehrerer Kriterien ist die als Version 2 vorgestellte Oligonukleotid-Bibliothek insgesamt spezifischer und etwas sensitiver als die manuell konfigurierte Oligonukleotid-Bibliothek.

Eine Oligonukleotid-Bibliothek für die Organismen-Identifikation „Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“ wurde innerhalb weniger Tage erstellt. Mit einer kleinen Bibliothek von sechs Oligonukleotiden konnten zwei →Ziel-Sequenzen mit einem →Redundanz-Niveau von r = 3 und maximaler Spezifität nachgewiesen werden.

Die hier vorgestellten Anwendungen sind in Zusammenarbeit mit dem FuE-Verbund Gensensorik durchgeführt worden. Das Projekt zur Identifikation von Hepatitis C-Viren, anfangs bearbeitet mit Dr. Hildegard Gersdorf und später mit Dipl. Biol. Denja Drutschmann, begleitete von Beginn an die Arbeit an dem Thema „Optimierung von Oligonukleotid-Bibliotheken“. Parallel zur Weiterentwicklung des in dieser Arbeit entstandenen Optimierungsprogramms **optiNA** sind für das HCV-Projekt mehrere Oligonukleotid-Bibliotheken entstanden. Die einzelnen Versionen der Bibliotheken wurden somit nach und nach verbessert und mit einer manuell konfigurierten Bibliothek verglichen [82]. Da von den Molekularbiologen zugleich am →Hybridisierungsprotokoll, an Tests neuer von dem Chemie-Teilprojekt des FuE-Verbunds Gensensorik erstellten funktionalisierten Oberflächen für die kovalente Anbindung der Oligonukleotide und an mehreren Anwendungen gearbeitet wurde, gibt es nur wenige Ergebnisse zu Hybridisierungen mit den automatisch erstellten Oligonukleotid-Bibliotheken.

Die Mitte 2001 erstellte Oligonukleotid-Bibliothek zur „Organismen-Identifikation: Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“ wurde innerhalb weniger Tage erstellt. Dieses Projekt wurde in Zusammenarbeit mit Dr. Katja Kerkmann vom FuE-Verbund Gensensorik und der iSenseIt - Intelligente Sensorsoftware und Bioinformatik AG abgewickelt. Als Firmenausgründung des TZI und des FuE-Verbund Gensensorik der Universität Bremen ist der iSenseIt Anfang 2001 im Rahmen eines Kooperationsvertrages die Optimierungs-Software übergeben worden. Neben einigen Änderungen für Rechengeschwindigkeit und Installation in einen Rechner-Cluster basiert die von der iSenseIt verwendete Software auf dieser Arbeit.

An den Ergebnissen des HCV-Projektes wird die Wirkungsweise der Bewertungsfunktionen des Abschnitts 4.1 zur Bewertung der theoretischen →Spezifität dargestellt. Am Projekt zur Organismen-Identifikation „Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*“ kann die Bewertung der Spezifität als Differenz von Hybridisierungseffizienzen betrachtet werden. Die Funktion der in Abschnitt 4.2 vorgestellten Bewertungsfunktionen für die Hybridisierungseffizienz und Sekundärstrukturen wird damit verdeutlicht. Weiterhin können die unterschiedlichen Signalstärken innerhalb der Gruppe der richtig-positiven Signale anhand der Bewertungsfunktionen interpretiert werden.

7.1. Identifikation von Hepatitis C-Viren mittels DNA-Mikroarrays

Der Gesetzgeber in Deutschland hat Anfang 1999 für Tests auf Hepatitis C-Viren in Blutbanken DNA-analytische Verfahren vorgeschrieben. In einer Überarbeitung der Richtlinien aus dem Jahre 1996 von der Bundesärztekammer und dem Paul-Ehrlich-Institut heißt es: „Die Prüfung auf Hepatitis-C-Viren ist mit einer geeigneten Nukleinsäure-Amplifikationstechnik durchzuführen. Das Ergebnis muss negativ sein.“ [112] Die „Testung von Blutspenden auf Hepatitis-C-Virus mit Nukleinsäure-Nachweis-Techniken“ [112] wurde im Bundesgesundheitsblatt (1998; 11, Seite 512) vorgeschrieben.

Hepatitis C-Viren (HCV), die beim Menschen akute oder chronische Lebererkrankungen hervorrufen, besitzen ein hochvariables, 9,6 kb großes RNA-Genom (→Genom), das für die Identifikation des Erregers und für seine Genotypisierung, die für die medizinische Diagnose und Therapie von entscheidender Bedeutung ist, genutzt werden kann [98]. In [92] wird eine Einteilung in 6 →phylogenetisch verschiedene Gruppen, den sogenannten *clades* vorgeschlagen. Ein Zusammenhang zwischen den *clades* und den zuvor in der Literatur [66], [104] mit Genotyp 1 bis 11 bezeichneten Gruppen wird wie folgt angegeben:

Tabelle 7.1-1: Zusammenhang zwischen *clades* und Genotypen

Clades	1	2	3	4	5	6
Genotypen	1	2	3, 10	4	5	6, 7, 8, 9, 11

Die Genotypen wurden weiter unterteilt in Subtypen mit den Namen 1a, 1b, 1c, 2a, 2b, 2c, 3a, 3b usw. Von besonderer Bedeutung ist die Unterscheidung zwischen den Genotypen 1a und 1b, dessen Korrelation mit einer Resistenz gegen eine Interferontherapie allerdings kontrovers diskutiert wird [48], [51], [63], [66], [108], [122].

Derzeit sind etwas über 12000 HCV-Sequenzvarianten in dem Taxonomy-Browser des NCBI abrufbar. Zur Zeit des Sequenzretrievals für die manuell erstellte HCV-Bibliothek waren etwas mehr als 7000 HCV-Sequenzvarianten bekannt. Sie sind lediglich in der 5´-untranslatierten Region (5´UTR) stark konserviert, die häufig für den →RT-PCR Nachweis dieser Viren genutzt wird. Die Bindungsstellen der entsprechenden universellen bzw. genotypspezifischen Primer zeigen allerdings Mikroheterogenitäten, die dazu führen, dass in

speziellen Fällen nur etwa 70% der bekannten Varianten perfekt hybridisieren und als Ursache für falsch negative →PCR-Testergebnisse in Frage kommen. In dem FuE-Verbund Gensensorik wurden seit 1999 mehrere Oligonukleotid-Bibliotheken, die alle bekannten 5´UTR HCV-Sequenzen zu erfassen erlauben, dadurch erstellt, dass für alle Varianten separate Fänger-Oligonukleotide für die →Hybridisierung bereit gestellt werden [81], [25]. Die Qualität einer manuell konfigurierten und mehrerer automatisch mit Bioinformatik-Software konfigurierter Bibliotheken wurden bezüglich →Spezifität miteinander verglichen [82].

Für die manuelle als auch für die automatisch konfigurierten Bibliotheken bestand die Aufgabe darin, für die Genotypen 1a, 1b, 2, 3 und 4 des Hepatitis C-Virus Oligonukleotid-Teilbibliotheken zusammenzustellen. Weiterhin sollten 1-universelle[31], im folgenden mit 1u bezeichnet, und HCV-universelle[31] Teilbibliotheken erstellt werden. Die Abbildung 7.1-1 veranschaulicht dieses hierarchische Verwandtschaftsverhältnis zwischen den Genotypen.

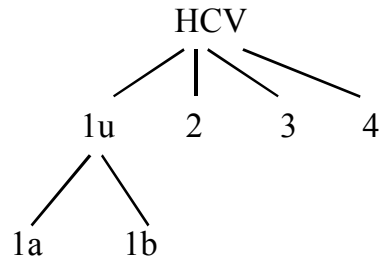


Abbildung 7.1-1: hierarchische Verwandtschaftsverhältnisse bei HCV

Nach einem Retrieval von über 7000 HCV-Sequenzen wurden 944 5´UTR-Sequenzen identifiziert. Davon wurden 749 Sequenzen den Genotypen 1a (118 Sequenzvarianten), 1b (396 Sequenzvarianten), 2 (97 Sequenzvarianten), 3 (89 Sequenzvarianten) und 4 (49 Sequenzvarianten) zugeordnet. Weitere 68 Sequenzen konnten nicht zweifelsfrei 1a oder 1b zugeordnet werden und wurden somit als 1u-Sequenzen für „1-universell“ verarbeitet. Die Knoten der Hierarchie in Abbildung 7.1-1 entsprechen den in Abschnitt 4.1.2 eingeführten →Sequenzklassen G_i.

7.1.1. Ergebnisse der manuell konfigurierten Oligonukleotid-Bibliothek

Die Tabelle 7.1-2 zeigt die →Treffer der im Oktober 1999 manuell konfigurierten Oligonukleotid-Bibliothek. Es sind für jede genotypspezifische Teilbibliothek L_i (in den Zeilen) die Trefferanzahlen $|G_i \cap \bigcup_{x \in L_i} Match(x)|$ für jede →Ziel-Sequenzmenge G_i (in den Spalten) aufgezeigt. Hier wurden mit $g_Z = 0$ und $g_N = 0$ nur die „*perfect match*“-Treffer als positives Signal gewertet. Die Oligonukleotide x, die zu einer Ziel-Sequenz t ein →Mismatch hatten, d.h. $H(x, t) = 1$, wurden als negatives Signal gewertet.

[31] Mit „1-universell“ bzw. 1u wird die Teilbibliotheken bezeichnet, die die Genotypen 1, 1a oder 1b detektiert. Entsprechend soll die HCV-universelle Teilbibliothek die Anwesenheit eines beliebigen HCV-Subtyps signalisieren.

Tabelle 7.1-2: Oktober 1999 manuell konfigurierte Oligonukleotid-Bibliothek

Treffer bei Genotyp: (Anz. Ziel-Sequenzen)	1a (118)	1b (396)	4 (49)	3 (89)	2 (97)
GT1a-Teilbibliothek	116	369	46	1	77
GT1b-Teilbibliothek	114	395	46	1	81
GT4-Teilbibliothek	0	0	43	0	0
GT3-Teilbibliothek	2	3	0	87	0
GT2-Teilbibliothek	1	0	2	0	77

Bei einem angestrebten →Redundanz-Niveau von r = 1 sind die Sensitivitäten $sens_r(L_i)$ der Oligonukleotid-Teilbibliotheken L_i der Zeile i als Quotient der Zahl auf der Hauptdiagonalen zu der geklammerten Gesamtanzahl $|G_j|$ aller Ziel-Sequenzen der entsprechenden Spalte j in der ersten Zeile abzulesen. Beispiele: Die Sensitivität der Genotyp-1a-Teilbibliothek beträgt: 116/118 ≈ 0,983. Die Genotyp-2-Teilbibliothek hat mit 77/97 ≈ 0,794 die schlechteste Sensitivität.

Die Zahlen, die nicht auf der Hauptdiagonalen liegen, sind die Anzahlen der falsch-positiven Signale. Für die Berechnung der Spezifität fehlt noch die Anzahl der richtig-negativen Signale, die sich als Summe der Differenzen „geklammerte Gesamtanzahl $|G_j|$ aller Ziel-Sequenzen der entsprechenden Spalte in der ersten Zeile minus Anzahl der falsch-positiven“ für alle Zahlen, außer derjenigen auf der Hauptdiagonalen, als Summanden ergeben:

$$\text{(Spezifität von Teilbibliothek } L_i) = \frac{\sum_{j \neq i} |G_j| - (\text{Treffer von } L_i \text{ auf } G_j)}{\sum_{j \neq i} |G_j|} = \frac{rn}{rn + fp}$$

$$= 1 - \frac{\sum_{j \neq i} (\text{Treffer von } L_i \text{ auf } G_j)}{\sum_{j \neq i} |G_j|} = 1 - \frac{fp}{rn + fp}$$

Beispiele: Die Spezifität der Genotyp-3-Teilbibliothek beträgt: (118-2 + 396-3 + 49-0 + 97-0) / (118 + 396 + 49 + 97) ≈ 0,992. Die Genotyp-4-Teilbibliothek hat trivialerweise die Spezifität 1 und mit (396-369 + 49-46 + 89-1 + 97-77) / (396 + 49 + 89 + 97) ≈ 0,219 hat die Genotyp-1a-Teilbibliothek die schlechteste Spezifität. Die Tabelle 7.1-3 enthält alle Sensitivitäten und Spezifitäten der Teilbibliotheken. Auffällig sind die geringen Spezifitäten der Teilbibliotheken für den Genotyp 1a und 1b und die geringe Sensitivität 77/97 ≈ 0,794 der Genotyp-2-Teilbibliothek.

Tabelle 7.1-3: Sensitivitäten und Spezifitäten der manuell konfigurierten Oligonukleotid-Bibliothek

	Sensitivität	Spezifität
GT1a oligos	0,983	0,219
GT1b oligos	0,997	0,314
GT4 oligos	0,878	1,000
GT3 oligos	0,978	0,992
GT2 oligos	0,794	0,995

Die in Tabelle 7.1-3 komprimiertere Darstellung der Eigenschaften der Teilbibliotheken enthält beispielsweise nicht mehr die Information, dass die Genotyp-1a- und Genotyp-1b-

Teilbibliotheken immerhin bezüglich Genotyp 3 sehr spezifisch sind. Deshalb ist eine Darstellung, wie in der Tabelle 7.1-2, vorzuziehen.

Die manuell konfigurierte Oligonukleotid-Bibliothek wurde über einen Zeitraum von mehreren Monaten mit den Ende 1999 im Internet verfügbaren Diensten für Sequenzretrieval und Primer-Design erstellt. Dabei wurden die Sekundärstrukturen der Fänger-Oligonukleotide nur stichprobenhaft überprüft. Die Menge der betrachteten Oligonukleotid-Kandidaten wurde als die Menge der Varianten von Oligonukleotiden in sieben Bindungsregionen innerhalb der 5´UTR-Sequenzen gewonnen. Diese Bindungsregionen wurden der Literatur entnommen. Die Abbildung 7.1-2 veranschaulicht die Bindungsregionen als rot hervorgehobene vertikale Balken in einem start verkleinerten multiplen →Alignment von 944 5´UTR-Sequenzen.

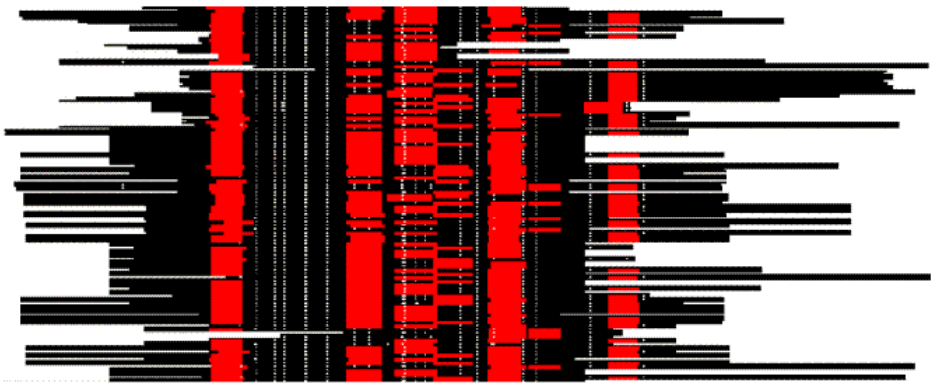


Abbildung 7.1-2: Beschränkung auf Bindungsregionen

7.1.2. Ergebnisse der mit **optiNA** konfigurierten Oligonukleotid-Bibliotheken

Für das HCV-Projekt sind mehrere Versionen von Oligonukleotid-Bibliotheken entstanden, die nach und nach verbessert wurden. Zwei von den erstellten Oligonukleotid-Bibliotheken werden im folgenden als Version 1 und 2 vorgestellt. In dem Optimierungsprogramm **optiNA** wurde der in Abschnitt 5.1.1 vorgestellte Greedy-Algorithmus verwendet. Von über 6000 potentiellen Oligonukleotiden wurden für die Version 1 der Oligonukleotid-Bibliotheken aus 14 ausgewählten Bindungsregionen, durch systematische Variation von Länge und Position sowie durch Filterung auf jeweils 4°C-T_M-Intervalle, Mengen definiert, die eine ausreichend große Anzahl von Oligonukleotiden enthalten, um noch alle HCV-Varianten erfassen zu können. Die Abbildung 7.1-3 zeigt die Anzahlen von Oligonukleotiden in den verschiedenen 4°C-T_M-Intervallen. In dem T_M-Intervall [64°C, 68°C] befanden sich mit einer Anzahl von knapp 1300 die meisten Oligonukleotide.

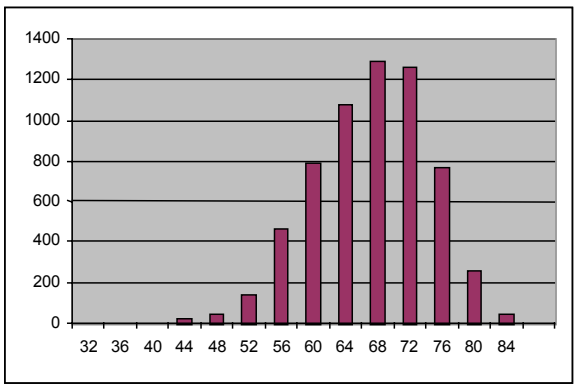


Abbildung 7.1-3: Anzahl der Oligonukleotide in 4°C-T_M-Intervallen

Um eine bessere Vergleichbarkeit zu gewährleisten, wurde für die Version 1 das gleiche T_M-Intervall [57°C, 61°C] wie bei der manuell erstellten Oligonukleotid-Bibliothek gewählt. Eine Auswahl von Teilmengen von Oligonukleotiden aus den knapp 1300 Kandidaten für die spezifische Überdeckung jeder oben angegebenen →Sequenzklasse entspricht dem „Set Cover"-Problem mit der Nebenbedingung maximaler Spezifität.

Die Tabelle 7.1-4 zeigt die →Treffer der Oktober 1999 automatisch konfigurierten Oligonukleotid-Bibliothek. Es sind für jede genotypspezifische Teilbibliothek (in den Zeilen) die Trefferanzahlen für jede Ziel-Sequenzmenge (in den Spalten) aufgezeigt. Von den 20 Zahlenwerten von falsch-positiven Treffern haben sich 8 verschlechtert, 5 verbessert und 7 sind gleich geblieben. Die Anzahl 295 der falsch-positiven Treffer der Version 1 ist jedoch drastisch gegenüber 743 falsch-positiven Treffern bei der manuell konfigurierten Oligonukleotid-Bibliothek reduziert.

Tabelle 7.1-4: Oktober 1999 automatisch konfigurierte Oligonukleotid-Bibliothek – Version 1

Treffer bei Genotyp:	1a	1b	4	3	2
(Anz. Ziel-Sequenzen)	(118)	(396)	(49)	(89)	(97)
GT1a-Teilbibliothek	113	11	26	0	69
GT1b-Teilbibliothek	95	396	23	59	1
GT4-Teilbibliothek	0	0	47	1	0
GT3-Teilbibliothek	2	2	0	89	0
GT2-Teilbibliothek	3	0	3	0	95

Durch einen Vergleich mit Tabelle 7.1-2 erkennt man bei der automatisch konfigurierten Oligonukleotid-Bibliothek eine deutlich verbesserte Spezifität und für Genotyp 2 ebenfalls eine verbesserte Sensitivität von $95/97 \approx 0{,}979$ gegenüber 0,794 bei der manuell erstellten Oligonukleotid-Bibliothek. Diese erste Version einer automatisch konfigurierten Oligonukleotid-Bibliothek wurde noch nicht bezüglich der Sekundärstrukturen der Ziel-Sequenzen optimiert. Die Weiterentwicklung des Bioinformatik-Systems **optiNA** umfasst die folgenden Änderungen:

- Bei der Erzeugung von Oligonukleotid-Kandidaten wurden nicht nur die Bindungsregionen berücksichtigt. Es wurden alle Oligonukleotide einer vorgegebenen Länge (hier 15 bis 30bp) aus den zu erwartenden →PCR-Produkten[32] ausgeschnitten. Dabei hat sich die Anzahl der Oligonukleotid-Kandidaten von 10839 auf 58589 erhöht.
- Es wurden zu 20 T_M-Intervallen von 49 bis 75°C mit einer Breite von 4°C Oligonukleotid-Bibliotheken berechnet. Die Abbildung 5.1-1 zeigt die Summe aller falsch-positiven Treffer. Mit einem T_M-Intervall von 57-61°C lag die erste manuell erstellte Oligonukleotid-Bibliothek nahezu optimal. Das Optimum liegt bei 58-62°C. Die Unterschiede ab 56-60°C sind jedoch unwesentlich. Ein allerdings erheblicher Sprung zeigt sich zwischen 55-59°C und 56-60°C.
- Optimierung bezüglich der Sekundärstruktur der Oligonukleotide
- Optimierung bezüglich der Sekundärstruktur der Ziel-Sequenzen an der Position des Oligonukleotids (vgl. Abbildung 2.3-2 und Abbildung 2.3-3)

[32] Die PCR-Produkte entsprechen den Sequenzen zwischen den beiden äußeren rot hervorgehobenen Bindungsregionen in der Abbildung 7.1-2. Dort binden die →Primer der, dem Hybridisierungs-Experiment vorgeschalteten, →Polymerasekettenreaktion.

- Optimierung des Redundanz-Niveaus r soweit möglich
- Bevorzugte Auswahl der Oligonukleotide mit einer größeren Anzahl von Basenfehlpaarungen zu den →Nichtziel-Sequenzen (d.h. $g_N \geq 2$ soweit möglich); $g_Z = 0$ wie auch bei der manuell konfigurierten Oligonukleotid-Bibliothek

Die Tabelle 7.1-5 zeigt die Anzahlen der Treffer für die Version 2 der mit **optiNA** erstellten Oligonukleotid-Bibliothek. Wegen der geänderten Optimierungskriterien und der vergrößerten Anzahl von Oligonukleotid-Kandidaten ist die Version 2 nicht direkt mit der Version 1 vergleichbar. Die Optimierung bezüglich der Sekundärstrukturen verringert die Anzahl der verwendbaren Oligonukleotide, und die Hinzunahme eines solchen Kriteriums läßt eine Verringerung der Sensitivität und der Spezifität erwarten. Der Wegfall der Beschränkung auf Bindungsregionen hebt diesen Effekt jedoch teilweise auf, sodass einige Zahlen auf den Nebendiagonalen gegenüber Version 1 verringert sind und andere vergrößert.

Die Sensitivitäten der Genotyp-1a-Teilbibliothek und der Genotyp-2-Teilbibliothek haben sich unwesentlich verbessert. Die volle Sensitivität von 1 haben die Teilbibliotheken zu Genotyp 1b und Genotyp 3 behalten, und die Sensitivität der Genotyp-4-Teilbibliothek blieb mit 47/49 ≈ 0,959 ebenfalls gleich. Bei größtenteils geringfügig geänderten Werten sind die stärksten Verschlechterungen die Unspezifitäten der Genotyp-1a- und der Genotyp-4-Teilbibliothek gegenüber den Genotyp-1b-Ziel-Sequenzen. Die deutlichste Verbesserung ist die um 55 falsch-positive Treffer (≈ 61,8%) verminderte Spezifität der Genotyp-1b-Teilbibliothek gegenüber den Genotyp-3-Ziel-Sequenzen. Insgesamt hat sich die Anzahl 295 der falsch-positiven Treffer der Version 1 auf 257 bei der Version 2 vermindert.

Tabelle 7.1-5: November 2000 automatisch konfigurierte Oligonukleotid-Bibliothek – Version 2

Treffer bei Genotyp: (Anz. Ziel-Sequenzen)	1a (118)	1b (396)	4 (49)	3 (89)	2 (97)
GT1a-Teilbibliothek	117	**27**	**27**	**0**	78
GT1b-Teilbibliothek	74	396	**24**	4	**7**
GT4-Teilbibliothek	1	10	47	1	0
GT3-Teilbibliothek	1	3	0	89	0
GT2-Teilbibliothek	0	0	0	0	96

7.2. Organismen-Identifikation: Cauliflower Mosaikvirus und *Agrobacterium tumefaciens*

Im Rahmen der Zusammenarbeit mit Dr. Katja Kerkmann (Abteilung BMG, Biotechnologie und Molekulare Genetik des UFT[33] des FuE-Verbunds Gensensorik der Universität Bremen) wurde eine Oligonukleotid-Bibliothek mit zwei Teilbibliotheken für Sequenz-Fragmente zweier Organismen erstellt, die spezifisch erkannt werden sollten. Es handelt sich um eine in einer frühen Phase befindlichen Vorstudie zur Organismen-Identifikation.

Die zwei Sequenzen sind: (1) ein Fragment des 35S-Promotors aus dem Cauliflower Mosaikvirus und (2) ein Fragment des NOS-Terminators des Gens für die Nopalinsynthase, das auf dem Ti-Plasmid von *Agrobacterium tumefaciens* kodiert ist. Das →Genom von *Agrobacterium tumefaciens* wurde kürzlich mit einer Größe von 5.674.062bp vervollständigt und am 14. Dezember 2001 veröffentlicht [35], [120]. Die Hybridisierungen wurden mit PCR-Produkten dieser zwei Sequenzen und einer Kontroll-Sequenz in der Arbeitsgruppe von

[33] UFT: Zentrum für Umweltforschung und Umwelttechnologie, http://www.uft.uni-bremen.de/

Professor Blohm (BMG) durchgeführt. Die Bioinformatik des FuE-Verbunds Gensensorik, in der diese Arbeit entstand, hat die Oligonukleotid-Bibliothek in Zusammenarbeit mit der iSenseIt AG konfiguriert.

Für zwei Hybridisierungs-Experimente wurden die berechneten Fänger-Oligonukleotide für das 35S-Promotor-Fragment und für das NOS-Terminator-Fragment (im folgenden als 35S und tNOS abgekürzt) zeilenweise mit hoher →Spot-Redundanz auf einen →Chip gespottet. Die Hybridisierungs-Lösung beider Experimente enthielt als Positivkontrolle das 169 bp-Fragment des M13mp18-Vektors [77] und auf dem DNA-Mikroarray wurde das Fänger-Oligonukleotid bcA (A=5´-TCC TGT GTG AAA TTG TTA TCC GCT-3´ → bcA=5´-AGC GGA TAA CAA TTT CAC ACA GGA-3´="reverse-complement" von A) immobilisiert. Das erste Hybridisierungs-Experiment enthielt zusätzlich die mit PCR amplifizierte 35S-Sequenz, und das zweite Hybridisierungs-Experiment enthielt neben der Positivkontrolle die ebenfalls PCR-amplifizierte tNOS-Sequenz.

Tabelle 7.2-1: Die Fänger-Oligonukleotide für diese Organismen-Identifikation

Name	Synonym	Sequenz
Core35S_l22_p18	35Sp18	GATGACGCACAATCCCACTATC
Core35S_l22_p34	35Sp34	ACTATCCTTCGCAAGACCCTTC
Core35S_l21_p77	35Sp77	TCATTTGGAGAGGACACGCTG
tNosCore_l27_p154	tNOSp154	GAGTCCCGCAATTATACATTTAATACG
tNosCore_l23_p191	tNOSp191	CAAAATATAGCGCGCAAACTAGG
tNosCore_l24_p203	tNOSp203	CGCAAACTAGGATAAATTATCGCG

Die Tabelle 7.2-1 zeigt die beiden Oligonukleotid-Teilbibliotheken dieses kleinen Projektes. In den ersten beiden Spalten stehen die in dem Optimierungs-Programm **optiNA** intern vergebenen Oligonukleotid-Namen und die von den Molekularbiologen des UFT vergebenen Namen. Die Namen der ersten Spalte wurden nach dem Muster

<*Ziel-Sequenz-Name*>_l<*Oligonukleotid-Länge*>_p<*Position auf der Ziel-Sequenz*>

erzeugt. Die dritte Spalte enthält die →Nukleotid-Sequenzen der Fänger-Oligonukleotide.

7.2.1. Hybridisierung mit einem PCR-Fragment des 35S-Promotors

Die Fänger-Oligonukleotide zum 35S-Promotor dienen dem Nachweis des Cauliflower Mosaikvirus. Die PCR-Produkte zu der 35S-Sequenz wurden mit einem DNA-Chip im UFT der Universität Bremen hybridisiert, der die in Tabelle 7.2-1 angegebenen konfigurierten Fänger-Oligonukleotide und als Positivkontrolle das Fänger-Oligonukleotid bcA = 5´-AGC GGA TAA CAA TTT CAC ACA GGA-3´ enthielt. Das →Hybridisierungsprotokoll sah keine Verwendung von Formamid vor, und es wurde ein hoher Salzgehalt verwendet. Damit war das Hybridisierungs-Experiment wenig →Stringent angesetzt.

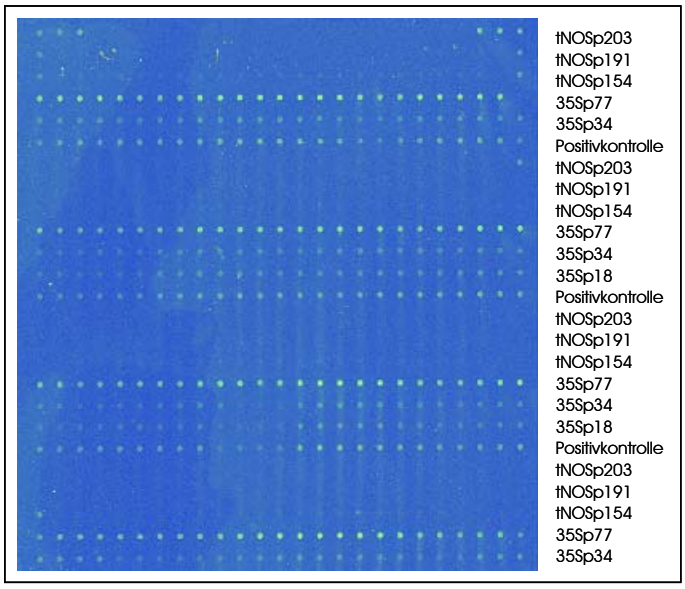


Abbildung 7.2-1: Hybridisierungssignale von dem Fragment des 35S-Promotors

Alle 7 Oligonukleotide (3 × 35s, 3 × tNOS und die Positivkontrolle) wurden mit größtmöglicher →Spot-Redundanz auf drei DNA-Mikroarrays mit 625 Spots immobilisiert (siehe Abbildung 7.2-1). Die Abbildung 7.2-2 stellt die 7 Mittelwerte über drei Arrays dar, die zugleich auf einem Träger in dem Hybridisierungs-Experiment verwendet wurden. Die 21 = 7 × 3 Werte, die dieser Mittelwertbildung zugeführt wurden, sind als Median aller zu den Fänger-Oligonukleotiden gehörigen Spots ermittelt worden. Die Signalintensität eines Spots wiederum wurde als Median der zugehörigen Pixel (Bildpunkte) aus dem Hybridisierungs-Bild quantifiziert.

Trotz der geringen →Stringenz des →Hybridisierungsprotokolls zeigen die folgenden Daten eine gute Hybridisierungs-Spezifität (Differenz der durchschnittlichen Hybridisierungseffizienzen zwischen den Oligonukleotiden der beiden →Sequenzklassen). Da in diesem Hybridisierungsansatz keine DNA des Nos-Terminators enthalten ist, fungieren die Fänger-Oligonukleotide des Nos-Terminators als Negativkontrolle für die Hybridisierung. Für die Auswertung und die Darstellung in Abbildung 7.2-2 wurde definiert, dass das Hybridisierungssignal der Positivkontrolle 100% beträgt und die übrigen Signale relativ dazu quantifiziert werden.

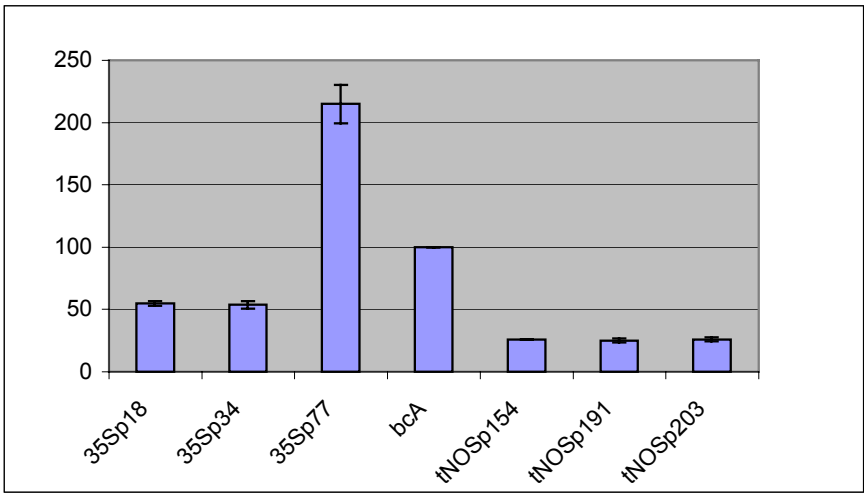


Abbildung 7.2-2: Hybridisierungssignale des Ansatzes mit dem Fragment des 35S-Promotors

In der Abbildung 7.2-2 ist zu sehen, dass die Hybridisierungssignale von zwei der berechneten Fänger-Oligonukleotide für die 35S-Sequenzen etwas mehr als 50% des Hybridisierungssignals der Positivkontrolle betragen, während das dritte berechnete Fänger-Oligonukleotid ein wesentlich stärkeres Signal als die Positivkontrolle hat. Die Signale der Negativkontrollen liegen deutlich unter den Hybridisierungssignalen der 35S-Fänger-Oligonukleotide.

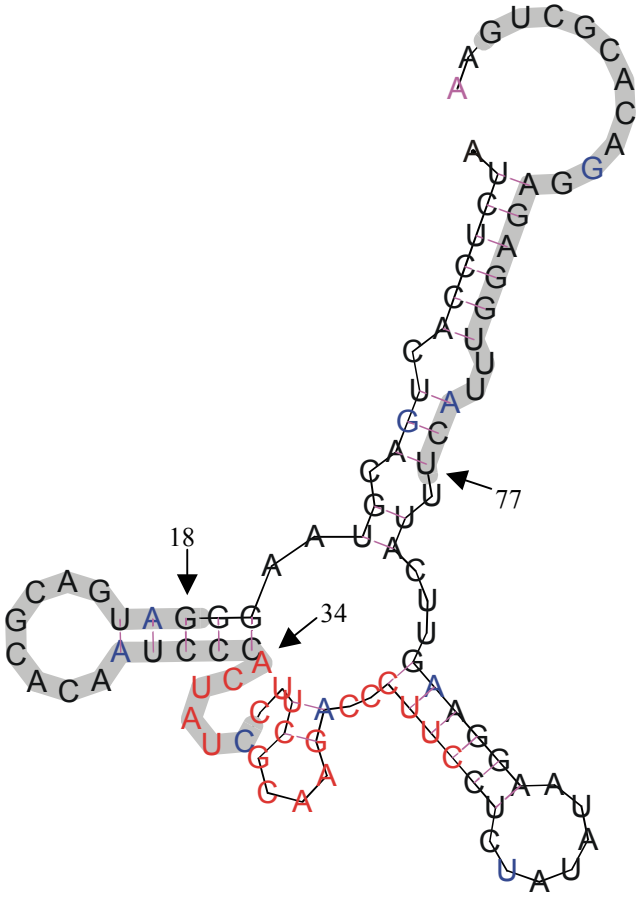


Abbildung 7.2-3: Positionen der drei Oligonukleotide für das 35S-Promotor-Fragment

Die Abbildung 7.2-3 stellt farblich und grau hinterlegt die Positionen der drei Oligonukleotide für das Fragment des 35S-Promotors dar. Das Oligonukleotid Core35S_l22_p18 ist am unteren Ende der Sekundärstruktur grau hinterlegt und Core35S_l22_p34 ist mit roten Buchstaben hervorgehoben. Die Intensität der Hybridisierungssignale dieser beiden Oligonukleotide waren fast identisch und betrugen etwas mehr als 50% des Hybridisierungssignals der Positivkontrolle. Das mit seinem besonders großen Hybridisierungssignal auffällige Oligonukleotid Core35S_l21_p77 fällt in dieser Darstellung ebenfalls durch seine Position auf. Es ist an einem Ende der →Ziel-Sequenz gelegen und möglicherweise daher besonders für eine effiziente Hybridisierung geeignet.

In der Anzahl von 12 (Core35S_l21_p77) und 14 (Core35S_l22_p18 und Core35S_l22_p34) ungepaarter Basen unterscheiden sich die drei Oligonukleotid-Positionen nur unwesentlich. Weiterhin ist die Anzahl der ungepaarten Gs und Cs jeweils 7. Diese Zahlenwerte wurden durch die im Abschnitt 4.2.3 vorgestellten Bewertungsfunktionen $\Delta\Delta G(x, t)$ und $sek(x, t)$ maximiert[34]; denn die durchschnittliche Anzahl ungepaarter Basen bezüglich aller Positionen eines 22-mers auf dieser Sekundärstruktur beträgt 11.3, und die durchschnittliche Anzahl ungepaarter Gs und Cs ist mit 4.8 um zwei geringer als bei allen drei oben betrachteten Oligonukleotiden.

7.2.2. Hybridisierung mit einem PCR-Fragment des NOS-Terminators (tNOS)

Die Fänger-Oligonukleotide zum NOS-Terminator dienen dem Nachweis des Ti-plasmids aus *Agrobacterium tumefaciens*. Die →PCR-Produkte zu der tNOS-Sequenz wurden ebenfalls mit einem DNA-Chip im UFT der Universität Bremen und den gleichen in Tabelle 7.2-1 angegebenen konfigurierten Fänger-Oligonukleotiden hybridisiert. Die Abbildung 7.2-4 ist bis auf die Positivkontrollen in den Ecken und in drei Zeilen invers zur Abbildung 7.2-1.

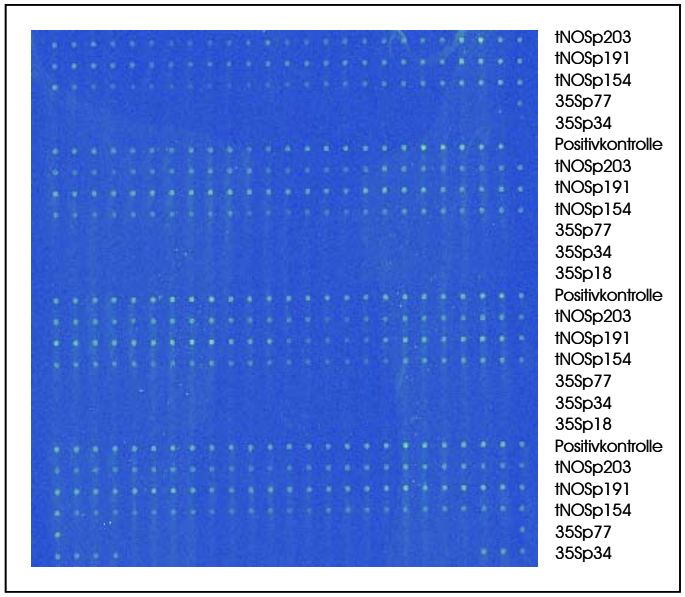


Abbildung 7.2-4: Hybridisierungssignale von dem Fragment des NOS-Terminators

Mit einem sonst identischen →Hybridisierungsprotokoll enthielt dieser Hybridisierungsansatz keine DNA des 35S-Promotors, sodass hier die 35S-Fänger-Oligonukleotide als Negativ-

[34] Die Minimierung der Funktionswerte der Bewertungsfunktionen $\Delta\Delta G(x, t)$ oder $sek(x, t)$ entspricht einer Maximierung der Anzahl ungepaarter Basen bzw. ungepaarter Gs und Cs.

kontrolle für die Hybridisierung fungieren. Wiederum wurde für die Auswertung und die Darstellung in Abbildung 7.2-5 das Hybridisierungssignal der Positivkontrolle auf 100% gesetzt und die übrigen Signale relativ dazu quantifiziert.

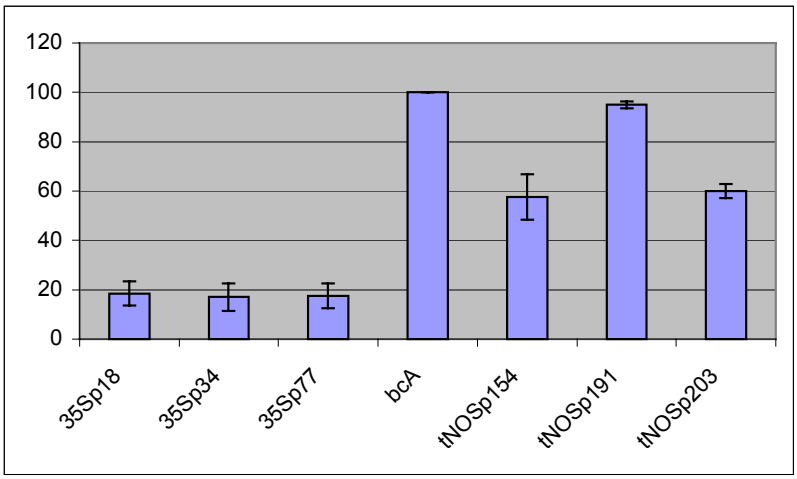


Abbildung 7.2-5: Hybridisierungssignale des Ansatzes mit dem Fragment des NOS-Terminators

Die Werte aus Abbildung 7.2-5 lassen eine bessere Diskriminierung als bei der 35S-Hybridisierung zu. Die Hybridisierungssignale der Negativkontrollen sind hier mit etwas unter 20% noch geringer, und mit ca. 60% sind die beiden schwächeren Signale von tNOSp154 und tNOSp203 immer noch 3-mal so groß wie die Signale der Negativkontrollen. Damit wird hier die NOS-Terminator-Sequenz erkannt und gegen die 35S-Sequenz abgegrenzt.

Die Abbildung 7.2-6 stellt farblich und grau hinterlegt die Positionen der drei Oligonukleotide für das Fragment des NOS-Terminators dar. Das Oligonukleotid tNosCore_l27_p154 ist am unteren durch Bulges unterbrochenen Stem-Loop grau hinterlegt. tNosCore_l23_p191 und tNosCore_l24_p203 überschneiden sich mit 11 Basen auf dem nach rechts-oben zeigenden Stem-Loop. tNosCore_l23_p191 ist mit roten Buchstaben hervorgehoben, tNosCore_l24-_p203 ist grau hinterlegt.

Die Oligonukleotide tNosCore_l27_p154 und tNosCore_l24_p203 haben mit 17 und 18 deutlich mehr ungepaarte Basen, als die entsprechenden durchschnittlichen Anzahlen ungepaarter Basen bezüglich aller Positionen eines Oligonukleotids der gleichen Länge auf dieser Sekundärstruktur. Die durchschnittliche Anzahl ungepaarter Basen eines 27-mers auf dieser Sekundärstruktur beträgt 10.5 und 9.4 für ein 24-mer.

Das Oligonukleotid tNosCore_l23_p191 hat interessanterweise mit nur 10 ungepaarten Basen das stärkste Hybridisierungssignal. Es ist jedoch bei allen dreien die Anzahl der ungepaarten Gs und Cs identisch, nämlich 4, und damit größer als die Durchschnitte (2.9, 2.6 und 2.5) ungepaarter Gs und Cs bezogen auf die Oligonukleotid-Längen 27, 24 und 23. Wie oben bereits erwähnt, führt die Minimierung der in Abschnitt 4.2.3 vorgestellten Bewertungsfunktionen $\Delta\Delta G(x, t)$ und $sek(x, t)$ zur Maximierung ungepaarter Basen an den Bindungsstellen.

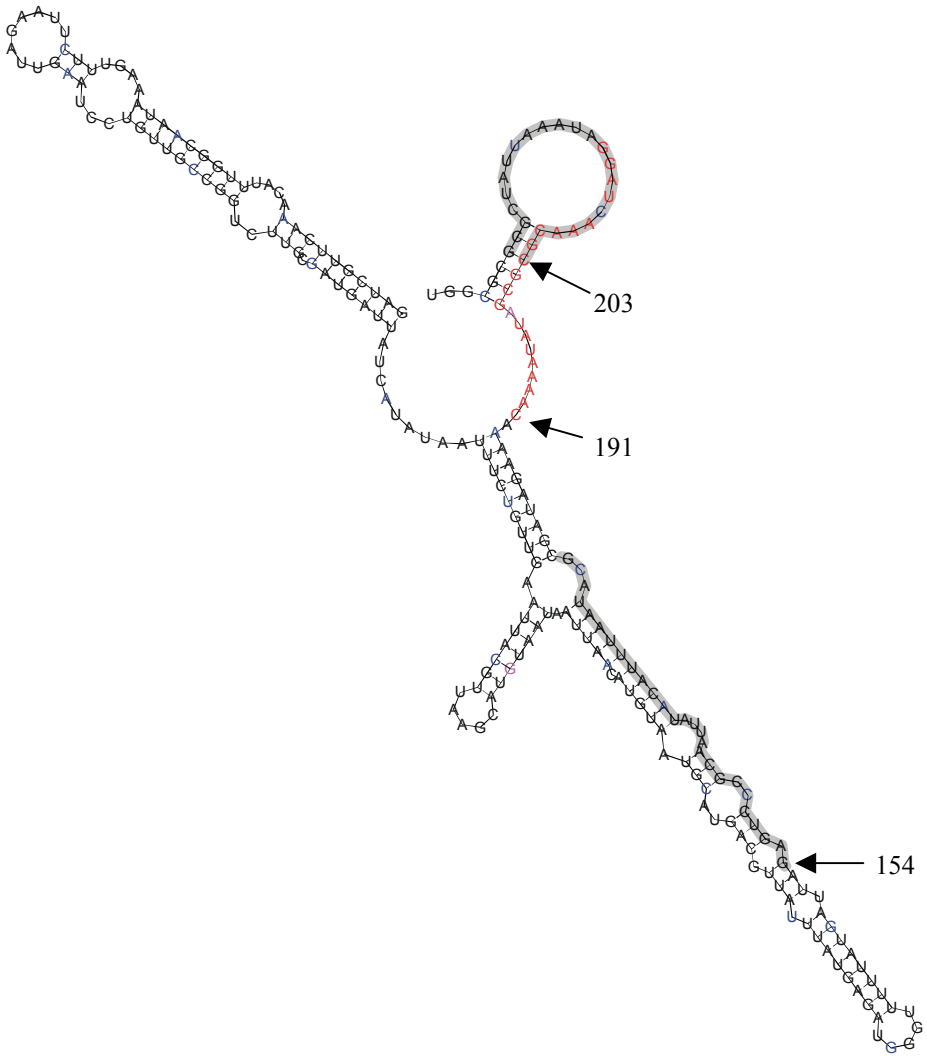


Abbildung 7.2-6: Positionen der drei Oligonukleotide für das NOS-Terminator-Fragment

Die relativ große Intensität des Hybridisierungssignals von tNosCore_l23_p191 könnte sich dadurch erklären lassen, dass sie die am wenigsten stabile Fänger-Sekundärstruktur hat. Diese wurde bei der Anwendung des Optimierungs-Algorithmus durch die Berücksichtigung der ΔG-Werte aller Fänger-Oligonukleotide minimiert (vgl. Abschnitt 4.2.4.2).

7.2.3. Diskussion der Ergebnisse der Organismen-Identifikation

In der Abteilung „Biotechnologie und Molekulare Genetik (BMG)“ des UFT der Universität Bremen wurden die ersten Ergebnisse eines Projektes zur Organismen-Identifikation zwischen dem Cauliflower Mosaikvirus und dem *Agrobacterium tumefaciens* vorgestellt. In Hybridisierungs-Experimenten mit der 35S- und der tNOS-Sequenz wurde in beiden Fällen die zu detektierende Sequenz mit dem maximalen →Redundanz-Niveau von $r = 3$ Hybridisierungssignalen pro Ziel-Sequenz erkannt. Dabei war das für die Diskriminierung wichtige minimale Verhältnis zwischen den richtig-positiven Signalen und den Negativkontrollen im Fall 35S 2:1 und bei der tNOS-Hybridisierung 3:1. Diese Ergebnisse wurden mit jeweils 2 weiteren Hybridisierungs-Experimenten reproduziert.

Eine Erklärung für diese gute Diskriminierung ist der erhebliche Unterschied zwischen den beiden Ausgangssequenzen 35S und tNOS. Deutlich problematischer war die Erstellung der Oligonukleotid-Bibliothek für den Hepatitis C-Virus, bei der zwei Oligonukleotide mit nur

wenigen Basenaustauschen in der Lage sein sollten, zwei Genotypen zu diskriminieren. Diese ersten Hybridisierungs-Experimente werden nach und nach in weitere Aufgabenstellungen mit weiteren zu diskriminierenden Ziel-Sequenzen eingebettet. Für nähere Informationen wende sich der Leser an die Abteilung „Biotechnologie und Molekulare Genetik (BMG)“ im UFT (Zentrum für Umweltforschung und Umwelttechnologie) der Universität Bremen.

8. Diskussion und Ausblick

Zusammenfassung: Das Kapitel 8 faßt die wichtigsten Eigenschaften des in dieser Arbeit entwickelten Systems zur Optimierung von Oligonukleotid-Bibliotheken zusammen und beschreibt den Einfluss, den dieses System auf die Arbeit mit DNA-Mikroarrays hat. DNA-Analytik mit DNA-Mikroarrays ist im Begriff, zu einem „high throughput"-Verfahren zu werden. In Zukunft wird es immer wichtiger, für diverse Anwendungen spezifische DNA-Mikroarrays zu entwickeln. Mit **optiNA** können Oligonukleotid-Bibliotheken qualitätsgesichert in kurzer Zeit erstellt werden.

In einem Ausblick werden Möglichkeiten vorgestellt, das in dieser Arbeit vorgestellte System zur Optimierung von Oligonukleotid-Bibliotheken weiter zu entwickln. Die Verallgemeinerung für ein wissensbasiertes Konfigurations-System, Kombination verschiedener algorithmischer Ansätze, ROC-Curves zur Visualisierung des Gegensatzes von Sensitivität und Spezifität und der Datensatz- und Ergebnis-Qualität und die Anwendung dieses Systems auf das Sequenz-Design für DNA-Computing sind interessante Möglichkeiten für eine weitere Beschäftigung mit dieser Thematik.

DNA-Analytik mit DNA-Mikroarrays ist im Begriff zu einem „high throughput"-Verfahren zu werden. In Zukunft wird es immer wichtiger, für diverse Anwendungen spezifische DNA-Mikroarrays zu entwickeln. Dabei sollte die Entwicklung, der Einsatz und die Auswertung der DNA-Mikroarrays schnell und qualitätsgesichert durchführbar sein. Bei der großen Menge von zu bearbeitenden Daten und der großen Anzahl von Qualitätskriterien ist diese Arbeit „manuell", d.h. ohne signifikante Unterstützung durch Bioinformatik-Systeme, nicht mehr zu leisten.

Mit **optiNA** wurde in dieser Arbeit ein Bioinformatik-System geschaffen, das den Entwicklungsprozess von DNA-Mikroarrays unterstützt. Entscheidet sich ein(e) Molekularbiologe/in während des Entwicklungsprozesses für eine andere Schmelztemperatur, eine andere Oligonukleotid-Länge oder eine andere Menge von →Ziel-Sequenzen, dann bedeutet das nicht mehr die Verschiebung der Fertigstellung des DNA-Mikroarrays um Wochen oder Monate.

Zu den genannten Qualitätskriterien, die zum Teil immer noch Gegenstand der Forschung sind, wurden Bewertungsfunktionen entwickelt oder aus dem Stand der Technik übernommen. Für das Bioinformatik-System **optiNA** und eine Sammlung von Perl- und Mathematica-Skripten wurden drei Ansätze für Optimierungs-Algorithmen entwickelt. Dabei wurde das System so angesetzt, dass die erwartungsgemäß fehlerbehafteten Bewertungsfunktionen beispielsweise durch ein erhöhtes Redundanz-Niveau berücksichtigt werden können bzw. Sicherheit bei der Auswertung der Ergebnisse eines Hybridisierungs-Experiments schaffen.

Die Optimierung von DNA-Mikroarrays wird sich noch Jahre weiterentwickeln. Mehrere Firmen weltweit haben sich auf dieses Ziel oder ähnlichen Aufgabenstellungen spezialisiert oder wurden eigens dafür gegründet. Die wichtigsten Aspekte der Weiterentwicklung sind eine verbesserte Vorhersage der Hybridisierungs-Effizienz, effiziente Algorithmen für eine →Kontroll-Recherche, die sich möglichst nahe an thermodynamischen Modellen orientiert und eine flexible Einsatzmöglichkeit der Optimierungs-Software für Spezialanwendungen. Denkbar ist ebenfalls die Weiterentwicklung bzw. die Integration des hier vorgestellten Systems in eine Konfigurierungs-Software, wie sie in [37], [36] und [55] vorgestellt werden oder die Extraktion und Verallgemeinerung der hier vorgestellten Ansätze für ein wissensbasiertes System bzw. für „*Knowledge-Based Configuration*" [38].

Die folgenden Abschnitte gehen auf einige Aspekte dieses Ausblicks näher ein und zeigen Möglichkeiten zur Weiterentwicklung auf bzw. nennen für themenverwandte Gebiete, wie das DNA-Computing, Einsatzmöglichkeiten des hier vorgestellten Systems.

8.1. Vergleich und Kombinationsmöglichkeiten der algorithmischen Ansätze

Ein Vergleich der drei Ansätze Greedy Set Covering, Kombination von Gradientenabstieg und Kompetition und Genetische Algorithmen ist sehr schwierig. Die Tabelle 8.1-1 stellt dar, dass die Ansätze bzgl. ihres Konzeptes, des Rechenaufwands und dem Potential zur kombinatorischen Optimierung sehr verschieden sind. Ein exakter Vergleich der genannten drei Ansätze ist nur bei Beschränkung auf bestimmte Parameter möglich, die sich aus den grundlegenden Eigenschaften der Algorithmen ergeben. Dennoch wird durch systemimmanente Eigenschaften eine identische Wahl beispielsweise von w_1 und w_2 beim Greedy-Algorithmus und beim Gradientenabstiegs-Algorithmus nicht dazu führen, dass die resultierenden Ergebnisse vergleichbarer sind. Diese Parameter beeinflussen grob die Gewichtung zwischen Sensitivität und Spezifität, wirken jedoch vollkommen verschieden.

Tabelle 8.1-1: Eigenschaften der drei Ansätze

	Greedy-Algorithmus	Kombination von Gradientenabstieg und Kompetition	Genetischer Algorithmus
Konzept	Lösungskonstruktion	lokale Suche	evolutionäres Prinzip
Rechenaufwand	gering	gering	groß
kombin. Optimierung	gering	mittel	gut

Auch die Bewertung der Qualität einer Oligonukleotid-Bibliothek darf bei dem Vergleich dieser Algorithmen nur unter Berücksichtigung der eingesetzten Rechenleistung und der Anzahl der resultierenden Oligonukleotide durchgeführt werden. In dieser Arbeit wurden alle drei Ansätze implementiert und an Optimierungsproblemen getestet. Der Genetische Algorithmus wurde anhand einer „Genotyp 1a“-Teilbibliothek getestet und am intensivsten wurde der Greedy-Algorithmus eingesetzt.

Die Verschiedenartigkeit der drei algorithmischen Ansätze ist jedoch zugleich ein Potential für bessere Algorithmen durch Kombination der einzelnen Verfahren. Am meisten bietet sich an, den Greedy-Algorithmus oder den Ansatz mit Gradientenabstieg in den Genetischen Algorithmus zu integrieren. In der Community um Genetische Algorithmen wird die Integration von zielgerichteten Mutations- und Rekombinations-Operatoren empfohlen. „For true optimization, hybrid methods such as a GA (Abk. für Genetischer Algorithmus) augmented by a hill climber or other kinds of gradient search have often been found to perform better than GA alone.“ [74] Mit diesem Ansatz ließe sich der Rechenaufwand durch eine Verringerung der benötigten Generationen (Iterationen) reduzieren. Erste Überlegungen dazu wurden bereits in den Abschnitten 2.5.3 und 5.3 angestellt.

8.2. ROC-Curves: Sensitivität vs. Spezifität

ROC-Curves oder ROC-Analysis (ROC: *receiver operating characteristic*) [65], [71] ist ein nützliches Werkzeug zur Bewertung von Tests und Prediktoren, die mit dem Wertepaar Sensitivität und Spezifität beschrieben werden. In einem Gespräch mit einer Molekularbiologin gab es auf die Frage „Hätte die automatisch erstellte Oligonukleotid-Bibliothek nicht spezifischer sein können?“ die Antwort „Ja, aber nur auf Kosten der Sensitivität“. Bei dem Design eines Prediktors für einen problematischen Datenbestand schließen sich Sensitivität und Spezifität gegenseitig aus. Die Abbildung 8.2-2 stellt diesen Zusammenhang anschaulich

dar. Der Punkt in der linken oberen Ecke stellt den perfekten Prediktor dar, der eine 100%-ige Sensitivität und Spezifität realisiert. Die Ecken links-unten und rechts-oben sind trivial realisierbar, indem ein Prediktor alle Fälle als positiv (rechts-oben: Sensitivität = 1; Spezifität = 0) oder als negativ (links-unten: Sensitivität = 0; Spezifität = 1) klassifiziert. Ein guter nicht-trivialer Prediktor wird demnach in dem linken-oberen Dreieck nahe der linken-oberen Ecke positioniert sein.

Eine ROC-Curve veranschaulicht man sich am besten über das Setzen eines Grenzwertes zwischen zwei Dichtefunktionen von zwei zugehörigen monomodalen Verteilungen. Die Position des Grenzwertes definiert die Anzahlen von richtig-positiven, falsch-negativen, richtig-negativen und falsch-positiven Fällen und damit die Sensitivität und Spezifität. Die Abbildung 8.2-1 stellt das Positionieren eines Grenzwertes dar. Damit wird festgelegt, ob die Sensitivität oder die Spezifität mehr gewichtet wird.

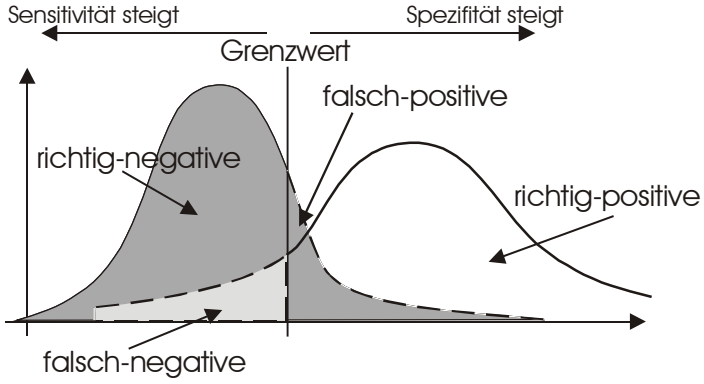


Abbildung 8.2-1: Positionierung eines Grenzwertes

Wird nun die Position des Grenzwertes variiert und die resultierenden Paare von Sensitivitäten und Spezifitäten in einem xy-Diagramm als Tupel (Sensitivität, 1- Spezifität) aufgetragen, dann erhält man eine ROC-Curve [65], [71], wie sie prototypisch in Abbildung 8.2-2 dargestellt ist. Diese Kurve gibt einen guten Überblick über den Zusammenhang von Sensitivität und Spezifität. Nicht immer ist es ratsam, den zur linken-oberen Ecke (optimaler Prediktor) nächstgelegenen Punkt auf der Kurve zu wählen. Die Entscheidung für mehr Sensitivität oder mehr Spezifität ist abhängig von dem Kontext, in dem der Prediktor eingesetzt wird. Die ROC-Curve gibt jedoch Entscheidungshilfen, ob z.B. durch die Aufgabe von etwas Spezifität mehr Sensitivität gewonnen werden sollte oder umgekehrt. Der Abstand der ROC-Curve zur Diagonalen bzw. die Nähe zur linken und oberen Kante des Quadrats gibt wieder, wie gut der Prediktor und der Datenbestand ist.

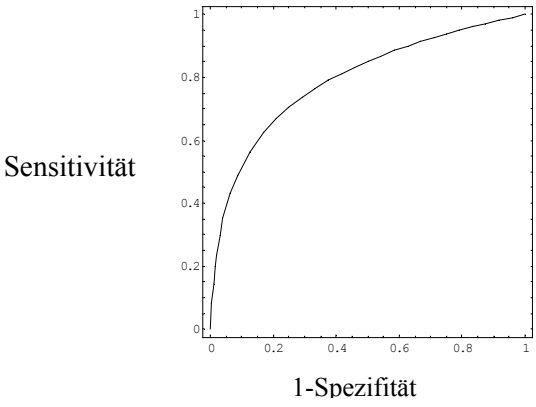


Abbildung 8.2-2: Eine typische „ROC Curve“

In den drei algorithmischen Ansätzen dieser Arbeit ist die Möglichkeit zur Gewichtung der Sensitivität und Spezifität bei dem Design von Oligonukleotid-Bibliotheken gegeben (vgl. Abschnitt 5.1.1). Variiert man diese Gewichtung, so ergeben sich auch hier ROC-Curves. Weiterhin kann auf diese Weise mit anderen Parametern gearbeitet werden, die indirekt die Sensitivität, Spezifität oder auch die Größe der Oligonukleotid-Bibliothek beeinflussen. Somit kann explorativ ermittelt werden, bei welcher Konstellation von Parametern und Eigenschaften der resultierenden Oligonukleotid-Bibliothek, der größte Gewinn bezüglich einer Menge von, mit großem Gewicht vorgegebenen, Eigenschaften erzielt werden kann.

Interessant wäre ebenfalls, die ROC-Curves für verschiedene Schmelztemperaturen oder für die einzelnen Ziel-Sequenzklassen einer Oligonukleotid-Bibliothek zu bestimmen. Auf diese Weise kann beurteilt werden, welche Ziel-Sequenzklasse bei welcher Schmelztemperatur bezüglich der Sensitivität oder Spezifität problematisch ist.

8.3. DNA-Computing

Ein zum Design von Oligonukleotid-Bibliotheken für DNA-Mikroarrays themenverwandtes Gebiet ist das DNA-Computing (auch *DNA based computation*) [61], [67], [68], [119]. Auch dort wird die Eigenschaft von Nukleinsäuren zur spezifischen Hybridisierung gezielt eingesetzt, hauptsächlich zur Nachbildung von Informationsverarbeitung.

Vom DNA-Computing erhofft man sich Ansätze zum Lösen von Problemen, dessen Rechenaufwand exponentiell mit der Problemgröße wächst. Beispiele solcher Probleme sind die Berechnung eines Hamiltonschen Graphen (engl. *hamiltonian path problem*) oder das „knacken" von kryptographischen →Codes, welche zu der Gruppe der NP-vollständigen Probleme gezählt werden. Das Grundprinzip des DNA-Computing besteht darin, dass die Eingangs-Information eines Problems in DNA-Moleküle kodiert wird und anschließend, befähigt durch eine große Anzahl von Molekülen und spezifischer Hybridisierung, eine kombinatorische Optimierung stattfindet. Die Hybridisierungen werden sowohl in flüssiger Phase wie auch auf DNA-Mikroarrays [61], [115] durchgeführt. Im Abschnitt 8.3.1 wird näher auf die Themenverwandtschaft zwischen dem Sequenz-Design für Oligonukleotid-Bibliotheken und für das DNA-Computing eingegangen.

8.3.1. Sequenz-Design für DNA-Computing

Bei der Organismen-Identifikation mit DNA-Mikroarrays haben die Oligonukleotide die Aufgabe, Ziel-Moleküle während des Hybridisierungs-Experiments einzufangen. Dazu müssen für eine gute Hybridisierungs-Effizienz optimale Hybridisierungs-Eigenschaften gegeben sein. Weiterhin ist eine maximale Spezifität gefordert, sodass das Oligonukleotid ausschließlich mit der →Ziel-Sequenz hybridisiert. Diese zwei Bedingungen tauchen in ähnlicher Form beim DNA-Computing auf. Der bei dem Design von Oligonukleotid-Bibliotheken, für z.B. variantenreiche Virengenome, wichtige Aspekt der Sensitivität („Set Cover"-Problematik) wird hier zunächst nicht betrachtet und am Ende dieses Abschnitts kurz aufgegriffen.

Die Hybridisierungs-Effizienz und Spezifität sind beim DNA-Computing wichtig für das sogenannte „*word design*" [68]. Damit werden die Sequenzen bezeichnet, die an den Information kodierenden Molekülen für die Hybridisierung mit weiteren Molekülen vorgesehen sind. Diese auch als „*sticky ends*" bezeichneten Sequenzen entsprechen damit den Fänger-Oligonukleotiden bei der Organismen-Identifikation mit DNA-Mikroarrays. Während für die Organismen-Identifikation die Oligonukleotide aus der Sequenz des Organismus ausgewählt werden müssen, gibt es beim „*word design*" für DNA-Computing die Möglichkeit, jede synthetisierbare DNA-Sequenz zu verwenden. In [68] wurde die theoretisch realisierbare Anzahl von spezifischen „*words*" für eine vorgegebene Länge berechnet. Die Bedingung für Spezi-

fität lautet in [68]: „for every pair of words *w*, *x* in a →code, there are at least *d* mismatches between *w* and *x* if $w \neq x$; and also between the reverse of *w* and the Watson-Crick complement of *x*“. Dieses Kriterium entspricht im wesentlichen der in Abschnitt 4.2.1 vorgestellten Hamming-Distanz; in [31] wird im selben Kontext die h-Distanz verwendet. Die Bedingung für Hybridisierungs-Effizienz der „*words*“ ist identisch zu der Bedingung für Oligonukleotide einer Oligonukleotid-Bibliothek: „the free energies and the enthalpies of the code words, and thus the melting temperatures, be similar“ [68]. Daher werden für das DNA-Computing ebenfalls die thermodynamischen Eigenschaften von Oligonukleotiden berücksichtigt [40] und ganze DNA Computer Designs in „*virtual test tubes*“ simuliert [32].

Damit können sich diese zwei Disziplinen, das DNA-Computing und das Design von Oligonukleotid-Bibliotheken für DNA-Mikroarrays, gegenseitig ergänzen. Es gibt auch Ansätze, DNA-Computing mit Hilfe von DNA-Mikroarrays „*on surfaces*“ durchzuführen [61], [115]. Das in dieser Arbeit vorgestellte System von Bewertungsfunktionen würde für die Berechnung von „*words*“ für das DNA-Computing zusätzlich die Sekundärstruktur des *words* selbst (vgl. Abschnitt 4.2.4.2), den thermodynamischen Abstand (vgl. thdist(x, t) in Abschnitt 4.2.1) zu anderen *words* und die Sekundärstrukturen der längeren aus den *words* zusammengesetzten Sequenzen betrachten, welche den Bewertungsfunktionen $\Delta\Delta G(x, t)$ und sek(x, t) aus dem Abschnitt 4.2.3 entsprechen. Auch das Konzept des einstellbaren →Redundanz-Niveaus könnte zur Vermehrung der Sicherheit bei besonders wichtigen Komponenten des DNA-Computing Einzug halten. Die Berücksichtigung von Sensitivität, d.h. das „Set Cover“-Problem oder auch das Treffen mindestens einer Ziel-Sequenz aus einer Menge von vorgegebenen Sequenzen, wäre, angewendet auf das DNA-Computing, nützlich für die Implementierung von Oder-Operatoren.

Literatur

[1] Allawi, H. T., and SantaLucia, J. Jr. (1998): Nearest Neighbor Thermodynamic Parameters for Internal G•A Mismatches in DNA, Biochemistry, 37, 2170-2179

[2] Allawi, H. T., and SantaLucia, J. Jr. (1998): Nearest-Neighbor Thermodynamics of Internal A•T mismatches in DNA: Sequence Dependence and pH Effects, Biochemistry, 37, 9435-9444

[3] Allawi, H. T., and SantaLucia, J. Jr. (1998): NMR solution structure of a DNA dodecamer containing single G•T mismatches, Nucleic Acids Research, Vol. 26, No. 21, 4925-4934

[4] Allawi, H. T., and SantaLucia, J. Jr. (1998): Thermodynamics of Internal C•T mismatches in DNA, Nucleic Acids Research, Vol. 26, No. 11, 2694-2701

[5] Baldi, P., Brunak, S. (2001): Bioinformatics – The Machine Learning Approach, The MIT Press.

[6] Bassett, D. E., Eisen, M. B., Boguski, M. S. (1999): Gene expression informatics – it's all in your mine, Nature Genetics Supplement, Vol. 21, 51-55

[7] Beißbarth, T., Fellenberg, K., Brors, B., Arribas-Prat, R., Boer, J. M., Hauser, N. C., Scheideler, M., Hoheisel, J. D., Schütz, G., Poutska, A., Vingron, M. (2000): Processing and quality control of DNA array hybridization data. Bioinformatics 16(11), 1014-1022.

[8] Bishop, C. M. (1995): Neural Networks for Pattern Recognition, Oxford University Press.

[9] Blake, R. D., and Delcourt, S. G. (1998): Thermal stability of DNA, Nucleic Acids Research, Vol. 26, No, 14, 3323-3332.

[10] Blake, R. D., Bizzaro, J. W., Blake, J. D., Day, G. R., Delcourt, S. G., Knowles, J., Marx, K. A., and SantLucia, J. Jr. (1999): Statistical mechanical simulation of polymeric DNA melting with MELTSIM, Bioinformatics, Vol. 15, No. 5, 370-375.

[11] Blohm, D. H. and Guiseppi-Elie, A. (2001): New developments in microarray technology. Current Opin. Biotechnol, **12**, 41-47.

[12] Bohnebeck, U., Nölte, M., Schäfer, T., Sirava, M., Waschulzik, T., Volkmann, G. (1999): An Approach to the Determination of Optimized Oligonucleotide Sets for DNA Chips. In: T. Lengauer, et. Al., Posters and Software Demonstrations - Seventh International Conference on Intelligent Systems for Molecular Biology ISMB 1999, p. 15, Heidelberg Germany.

[13] Boldt, L., Gersdorf, H., Niemeyer, C. M., Holtkamp, F., Bischoff, R., Sälter, W., Adler, M., Kayser, O., Wolf, M., Jüptner, W., und Blohm, D. A (1998): Nanotiterplate-based DNA Array applied for cDNA-Detection of Hepatitis C Virus, Poster auf dem Biosensor-Weltkongress, Berlin.

[14] Boldt, L. (1999): Methodische Untersuchungen im Vorfeld der Entwicklung miniaturisierter DNA-analytischer Verfahren, Dissertation, Universität Bremen.

[15] Brenig, W. (1975): Statistische Theorie der Wärme, Springer Verlag

[16] Breslauer, K. J., Frank, R., Blocker, H., Marky, L. A. (1986): Predicting DNA duplex stability from the base sequence, Proc. Natl. Acad. Sci. USA 83 , pages 3746-3750.

[17] Cormen, T.H., Leiserson, C. E., and Rivest, R. R. (1991): Introduction to Algorithms. McGraw Hill.

[18] Derr, T., Nölte, M., Castedello, T., Meyer, E., Lison, A. E., Leibfritz, D. (1997): Artificial Neural Network Classification of Renal Diseases Based on Data derived from Proton NMR Spectra of Human Blood Plasma. Proc. ISMRM'97, Vancouver.

[19] Ding Y., Lawrence, E. (2001): Statitstical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond, Nucleic Acids Research, Vol. 29, No. 5, 1034-1046.

[20] Doi, K., Imai H. (1997): Greedy Algorithms for Finding a Small Set of Primers Satisfying Cover and Length Resolution Conditions in PCR Experiments, Genome Informatics 1997, 43-52.
[21] Doi, K., Imai H. (1999): A Greedy Algorithm for Minimizing the Number of Primers in Multiple PCR Experiments, Genome Informatics 1999, 10, 73-82.
[22] Doktycz, M. J., Morris, M. D., Dormady, S. J., Beattie, K. L., and Jacobson, K. B. (1995): Optical Melting of 128 Octamer DNA Duplexes, The Journal of Biological Chemistry, Vol. 270, No. 15, pp. 8439-8445.
[23] Dong, F., Allawi, H. T., Anderson, T. Neri, B. P. and Lyamichev, V. I. (2001): Secondary structure prediction and structure-specific sequence analysis of single-stranded DNA, Nucleic Acids Research, Vol. 29, No. 15.
[24] Dorffner, G. (1991): Konnektionismus, B. G. Teubner.
[25] Drutschmann, D., Blohm, D. (2001): „DNA-Microarray zum Nachweis und zur Genotypisierung von Hepatitis C Viren“ in „Statusseminar Chiptechnologien: Vom Genom zum Proteom“, DECHEMA 2001.
[26] Flamm, C., Fontana, W., Hofacker, I. L. and Schuster, P. (2000): RNA folding at elementary step resolution, RNA, 6:325-338, Cambridge University Press.
[27] Flamm, C., Hofacker, I. L., Maurer-Stroh, S., Stadler, P. F., and Zehl, M. (2001): Design of Multi-Stable RNA Molecules, RNA, 7: 254-265.
[28] Fotin, A., Drobyshev, A. L., Proudnikov, D. Y., Perov, A. N., and Mirzabekov, D. (1998): Parallel thermodynamic analysis of duplexes on oligodeoxyribonucleotide mircochips, Nucleic Acids Research, Vol. 26, No. 6, 1515-1521.
[29] Furlong, E. E. M., Andersen, E. C., Null, B., White, K. P., Scott, M. P. (2001): Patterns of Gene Expression During Drosophila Mesoderm Development, Science, Vol. 293, 1629-1633.
[30] Galinier, P., Hao, J.-K. (1999): Hybrid Evolutionary Algorithms for Graph Coloring, Journal of Combinatorial Optimization 3, 379-397, Kluwer Academic Publishers.
[31] Garzon, M., Neathery, P., Deaton, R., Murphy, R. C., Franceschetti, D.R., and Stevens, S.E. Jr. (1997): A new metric for DNA computing. Koza, John R., Deb, K., Dorigo, M., Fegel, David B., Garzon, M., Iba, H., and Riolo, Rick L., (editors). Genetic Programming 1997: Proceedings the Second Annual Conference, The MIT Press. Pages 472-478.
[32] Garzon, M., Oehmen, C. (2001): Biomelecular Computation in Virtual Test Tubes in Jonoska, N., Seeman, N. C. (Editors) Proceedings of the 7th International Meeting on DNA Based Computers, University of South Florida, 2001.
[33] Gershenfield, N. (1999): The Nature of Mathematical Modeling, Cambrigde University Press.
[34] Gibas, C. and Jambeck, P. (April 2001): Developing Bioinformatics Computer Skills, O’Reilley.
[35] Goodner, B et al. (2001): Genome Sequence of the Plant Pathogen and Biotechnology Agent *Agrobacterium tumefaciencs* C58, Science, Vol. 294, 2323-2328.
[36] Günter, A. (1995): *KONWERK – ein modulares Konfigurierungswerkzeug* in: Maurer, F., Richter, M. M. (Hrsg.) *Expertensysteme ´95*, infix Verlag St. Augustin, Seite 1-18.
[37] Günter, A., Kreuz, I., Kühn, C. (1999): *Kommerzielle Software-Werkzeuge für die Konfigurierung von technischen Systemen* in KI - *Künstliche Intelligenz* Heft 3/99, Seiten 61-65, ISSN 0933-1875, arenDTaP Verlag Bremen.
[38] Günter, A., Kühn, C. (1999): *Knowledge-Based Configuration – Survey and Future Directions*. In Puppe, F. ed. *XPS-99: Knowledge Based Systems, Proceedings 5th Biannual German Conference on Knowledge Based Systems*, Springer Lecture Notes in Artificial Intelligence 1570, Germany.

[39] Hartemink, A. J., Gifford, D. K., Khodor, J. (1998): Automated Constrained-Based Nucleotide Sequence Selection for DNA Computation, Proceedings 4th Annual DIMACS Workshop on DNA Based Computers, Baltimore, Pennsilvania.
[40] Hartemink, A. J. and Gifford, D. K. (1997): Thermodynamic Simulation of Deoxyoligonucleotide Hybridization for DNA Computation, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns.
[41] Heun, V. (2000): Grundlegende Algorithmen - Einführung in den Entwurf und die Analyse effizienter Algorithmen, Vieweg.
[42] Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M., Schuster, P. (1994): Fast Folding and Comparison of RNA Secondary Structures. Monatshefte f. Chemie 125:167-188.
[43] Holland, J. H. (1975): Adaptation in Natural and Artificial Systems. University of Michigan Press. (Second Edition: MIT Press, 1992).
[44] Hunter, L. Editor (1993): Artificial Intelligence and Molecular Biology, MIT Press.
[45] HYTHER™ version 1.0, Nicolas Peyret and John SantaLucia, Jr., Wayne State University.
[46] Ibelgaufts, H (1993): Gentechnologie von A bis Z, VCH Verlagsgesellschaft.
[47] International human genome sequencing consortium (2001): Initial sequencing and analysis of the human genome, Nature, 409:860-921.
[48] Jaeckel, E., Cornberg, M., Wedemeyer, H., Santantonio, T., Mayer, J., Zankel, M., Pastore, G., Dietrich, M., Trautwein, C., Manns, M. (2001): Treatment of Acute Hepatitis C with Interferon Alfa-2b, New England Journal of Medicine, Nov. 15, 2001
[49] Kaderali, L. (2001): Selecting Target Specific Probes for DNA Arrays, Universität zu Köln.
[50] Kämpke, T., Kieninger, M. and Mecklenburg, M. (2001): Efficient primer design algorithms, Bioinformatics, 17(3):214-225; http://doprimer.interactiva.de.
[51] Kanai, K., Kako, M., Kumada, T., Tsubouchi, H., Aikawa, T., Kojima, M., Harada, H., Kawasaki, T., Nakashima, M., Okamoto, H., Mishiro, S. (1998): High-dose (9 MU) long-term (60 weeks) alfa-interferon therapy for chronic hepatitis patients infected with HCV genotype 1b, Archives of Virology, 143(8): 1545-1554.
[52] Kane M. D., Jatkoe T. A., Stumpf C. R., Lu J., Thomas J. D., Madore S. J. (November 2000): Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays, Nucleic Acids Research, Volume 28, Issue 22, Pages 4552-4557.
[53] Kel, A., Ptitsyn, A., Babenko, V., Meier-Ewert, S., Lehrach, H. (1998): A genetic algorithm for designing gene family-specific oligonucleotide sets used for hybridization: the G protein-coupled receptor protein superfamily. Bioinformatics, 14(3):259-270.
[54] Kosaraju, S. R., Schäfer, A. A., Biesecker, L. G. (1998): Approximation Algorithms for a Genetic Diagnostics Problem, Journal of Computational Biology, Vol. 5, No. 1, 9-26.
[55] Kreuz, I. (2000): Considering the Dynamic in Knowledge Based Configuration in Horn, W., ECAI 2000 Proceedings, IOS Press.
[56] Kühn, C. (2000): Modeling Structure and Behavior for Knowledge-Based Software Configuration in Horn, W., ECAI 2000 Proceedings, IOS Press.
[57] Lengauer T. (2001): "Computational Biology at the Beginning of the Post-genomic Era" in R. Wilhelm (Ed.): Informatics. 10Years Back. 10 Years Ahead, LNCS 2000, pp. 341-355, Springer-Verlag.
[58] Li, F., Stormo, G. D. (2001): Selection of Optimal DNA Oligos for Gene Expression Arrays, Bioinformatics (in press).
[59] Lin, S. M., Johnson, K. F. Eds. (2002): Methods of Microarray Data Analysis, Kluwer Academic Publishers.

[60] Lindblad-Toh, K., Winchester, E., Daly, M. J., Wang, D. G., Hirschhorn, J. N., Laviolette, J.-P., Ardlie, K., Reich, D. E., Robinson, E., Sklar, P., Shah, N., Thomas, D., Fan, J.-B., Gingeras, T., Warrington, J., Patil, N., Hudson, T. J., and Lander, E. S. (April 2000): Large-scale discovery and genotyping of single-nucleotide polymorphisms in the mouse, Nature Genetics, Vol. 24.

[61] Liu, Q., Wang, L., Frutos, A. G., Condon, A. E., Corn, R. M. and Smith, L. M. (Jan. 2000): DNA computing on surfaces, Nature, 175-179, Vol. 403.

[62] Lockhart, D., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and Brown, E. L. (1996): Expression monitoring by hybridization to high-density oligonucleotide arrays, Nature Biotechnology, Vol. 14, 1675-1680.

[63] Lopez-Labrador, F.-X., Ampurdanes, S., Giminez-Barcons, M., Guilera, M., Costa, J., Jiminez de Anta, M. T., Sanchez-Tapias, J. M., Rodes, J., Saiz, J.-C. (1999): Relationship of the Genomic Complexity of Hepatitis C Virus with Liver Disease Severaty and Response to Interferon in Patients with Chronic HCV Genotype 1b, Hepatology, p. 897-903, Vol. 29, No. 3.

[64] Lottspeich, F., Zorbas, H. Herausgeber (1998): Bioanalytik, Spektrum Akademischer Verlag.

[65] Lovell, D. R., Dance, C. R., Niranjan, M., Prager, R. W. and Dalton, K. J. (1996): Limits on the discrimination possible with discrete valued data, with application to medical risk prediction. Cambridge University Engineering Department.

[66] Maertens, G., Stuyver, L. (1997): Genotypes and Genetic Variation of Hepatitis C Virus in Harrison, T. J., Zuckermann, A. J. (1997): The Molecular Medicine of Viral Hepatitis, John Wiley & Sons Ltd.

[67] Mao C., LaBean T. H., Reif J. H., Seeman N. C. (2000): Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. Nature, 407: 493-496.

[68] Marathe, A., Codon, A. E., Corn, R. M. (2000): On Combinatorial DNA Word Design. DNA based Computers V, DIMACS Series, Winfree, E., Gifford, D. Eds., AMS Press, 75-89.

[69] Mathews, D.H., Sabina, J., Zuker, M. and Turner, D.H. (1999): Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure. J. Mol. Biol. 228, 910-940.

[70] Matson, R. S., Rampal, J., Pentoney, S. L., Anderson, P. D., and Coassin, P. (1995): Biopolymer Synthesis on Polypropylene Supports: Oligonucleotide Arrays, Analytical Biochemistry, 224, 110-116.

[71] Metz, C. E. (1978): Basic Principles of ROC Analysis, Seminars in Nuclear Medicine, Vol. VIII, No. 4.

[72] Meyer-Lüerßen, D. Herausgeber (August 1997): Diagnostik im Gespräch 3/1997 – Chancen und Risiken der Molekularen Diagnostik, VDGH Verband der Diagnostica-Industrie e.V.

[73] Meyer-Lüerßen, D. Herausgeber (November 2000): Gesundheitsfakten 4/2000 - Wenig spektakulär und dennoch gefährlich: Unterschätzte Virusinfektionen, VDGH Verband der Diagnostica-Industrie e.V.

[74] Mitchel, M. (1998): An Introduction to Genetic Algorithms, MIT Press.

[75] Mir, K. U., and Southern, E. M. (1999): Determining the influence of structure on hybridization using oligonucleotide arrays, Nature Biotechnology, Vol. 17, 788-792.

[76] Niemeyer, C. M., Blohm, D. (1999): DNA-Microarrays. Angew. Chem. Int. Ed., 38: 2865-2869.

[77] Niemeyer, C. M., Boldt, L., Ceyhan, B. and Blohm, D. (1999): Evaluation of Single-Stranded Nucleic Acids as Carriers in the DNA-Directed Assembly of Macromolecules, Journal of Biomolecular Structure & Dynamics, ISSN 0739-1102, Vol. 17.

[78] Niemeyer, C. M., Boldt, L., Ceyhan, B., Blohm, D. (1999): DNA-Directed Immobilization: Efficient, Reversible and Site-Selective Surface Binding of Proteins by Means of Covalent DNA-Streptavidin Conjugates. Anal. Biochem, 268: 54-63.

[79] Niemeyer, C. M., Bürger, W., Peplies, J. (1998): Covalent DNA-Streptavidin Conjugates as Building Blocks for the Fabrication of Novel Biometallic Nanostructures. Angew. Chem. Int. Ed., 37: 2265-2268.

[80] Niemeyer, C. M., Bürger, W. and Hoedemakers, R. M. J. (1998): Hybridization Characteristics of Biomolecular Adaptors, Covalent DNA-Streptavidin Conjugates, Bioconjugate Chemistry, 9, 168-175.

[81] Nölte, M., Gersdorf, H., Volkmann, G., Bischoff, R., Bohnebeck, U., Sirava, M., Schäfer, T., Waschulzik, T., Blohm, D. (2000): "Ein Bioinformatik-Prototyp zur Optimierung einer Oligonukleotidbibliothek für die Identifikation von Hepatitis C Viren mittels DNA-Mikorarrays" in "DNA-Chiptechnologie: Anwendung und Nutzung", Statusseminar DECHEMA.

[82] Nölte, M., Volkmann, G., Drutschmann, D., Waschulzik, T., Blohm, D. (2001): „Bioinformatik-System zur Optimierung von Oligonukleotidbibliotheken für DNA-Mikroarrays" in „Statusseminar Chiptechnologien: Vom Genom zum Proteom", DECHEMA.

[83] Nölte, M., Waschulzik, T., Bethke, M., Hoheisel, J., Blohm, D. (1999): Bestimmung von Hybridisierungseigenschaften von Oligonukleotiden mit Hilfe künstlicher Neuronaler Netzwerke in „Chiptechnologie für DNA-Diagnostik und Sequenzanalyse in Deutschland", Statusseminar DECHEMA.

[84] Nölte, M., Volkmann, G., Drutschmann, D., Blhom, D., Wischnewsky, M. B. (2001) Detektion von Hepatitis C Viren mit einer optimierten Oligonukleotid-Bibliothek in Medizinische Forschung und Gesundheitswissenschaften in Bremen (in press), Symposium 2001.

[85] Nölte, M., Volkmann, Wischnewsky, M. B. (2001) Software zur Konfigurierung und Auswertung von DNA-Mikroarrays in Medizinische Forschung und Gesundheitswissenschaften in Bremen (in press), Symposium 2001.

[86] Novère, N. Le (2001): MELTING, computing the melting temperature of nucleic acid duplex, Bioinformatics, Vol. 17 no. 12, Pages 1226-1227.

[87] Pevzner, P. A. (2000): Computational Molecular Biology – An Algorithmic Approach, MIT Press.

[88] Raddatz, G., Dehio, M., Meyer, T. F. and Dehio, C. (2001): PrimeArray: genome-scale primer design for DNA-microarray construction, Bioinformatics, Vol. 17 no. 1, Pages 98-99.

[89] Rampal, J. B. Editor (2001): DNA Arrays: Methods and Protocols, Humana Press, Vol. 170.

[90] Rauhut, R. (2001): Bioinformatik – Sequenz - Struktur - Funktion, Verlag Wiley-VCH.

[91] Ritter, H., Martinez, T., Schulten, K. (1990): Neuronale Netze – Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke, Addison-Wesley.

[92] Robertson, B., Myers, G., Howard, C., Brettin, T., Bukh, J., Gaschen, B., Gojobori, T., Maertens, G., Mizokami, M., Nainan, O., Netesov, S., Nishioka, K., Shin-i, T., Simmonds, P., Smith, D., Stuyver, L., and Weiner, A. (1998): Classification, nomenclature, and database for hepatitis C virus (HCV) and related viruses: proposals for standardization, VDN Virology Division News, Arch Virol 143/12.

[93] Rojas, R. (1993): Theorie der neuronalen Netzwerke, Springer-Verlag.

[94] Rouillard, J.-M., Herbert, C. J., and Zuker, M. (2002): OligoArray: genome-scale oligonucleotide design for microarrays, Bioinformatics, Vol. 18, no. 3, Pages 486-487.
[95] SantaLucia, J. Jr. (1998): A unified view of polymer, dumbell, and oligonucleotide DNA nearest-neighbor thermodynamics, Biochemistry, Proc. Natl. Acad. Sci. USA, Vol. 95, pp. 1460-1465.
[96] SantaLucia, J. Jr., Allawi, H. T., and Seneviratne, A. (1996): Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability, Biochemistry, 35, 3555-3562.
[97] Schena, M., Davis, R. W. (1999): Genes, genomes, and chips in Schena, M. (Editor), DNA Microarrays, Oxford University Press, 1-16.
[98] Simmonds, P. (2001): The origin and evolution of hepatitis viruses in humans, Journal of General Virology, 82, 693-712.
[99] Slavik, P. (1998): Approximation Algorithms for Set Cover and Related Problems. State University of New York at Buffalo.
[100] Southern, E., Mir, K., Shchepinov, M. (1999): Molecular interactions on microarray, Nature Genetics, Vol. 21(1), 5-9.
[101] Stamatiadis-Smidt, H., zur Hausen, H. (Hrsg.) und Eberhard-Metzger, C., Glomp, I., Hobom, B. (1998): Das Genom-Puzzle, Springer-Verlag.
[102] Stein, C. A. (1999): Hybridization prediction gets to first base, Nature Biotechnology, Vol. 17, 751-752.
[103] Stevens, R., Goble, C., Baker, P. and Brass, A. (2001): A classification of tasks in bioinformatics, Bioinformatics, 17(2):180-188.
[104] Stuyver, L., Rossau, R., Wyseur, A., Duhamel, M., Vanderborght, B., Van Heuverswyn H. and Maertens, G. (1993): Typing of hepatitis C virus isolates and characterization of new subtypes using a line probe assay, Journal of General Virology, 74, 1093-1102.
[105] Sugimoto, N., Nakano, S., Yoneyama, M. and Honda, K. (1996): Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes, Nucleic Acids Research, Vol. 24, No. 22, 4501-4505.
[106] Sugnet, C., Rice, E., Clark, T. (December 1999): Rational Selection of Oligonucleotide Probes for Microarray Construction.
[107] Talaat, A. M., Hunter, P., Johnston, S. A. (June 2000): Genome-directed primers for selective labeling of bacterial transcripts for DNA microarray analysis, Nature-Biotechnology; 18(6): 679-82.
[108] Tong, M. J., Reddy, K. R., Lee, W. M., Pockros, P. J., Hoefs, J. C., Keeffe, E. B., Hollinger, F. B., Heathcote, E. J., White, H., Foust, R. T., Jensen, D. M., Krawitt, E. L., Fromm, H., Black, M., Blatt, L. M., Klein, M., Lubina, J., and the Consensus Interferon Study Group (1997): Treatment of Chronic Hepatitis C With Consensus Interferon: A Multicenter, Randomized, Controlled Trial, Hepatology, 26, 747-754.
[109] Venter, J. C., et. al. (2001): The sequence of the human genome, Science, 291(5507), 1304-1351.
[110] Vienna RNA Package, http://www.tbi.univie.ac.at/~ivo/RNA.
[111] Vo-Dinh, T., Cullum, B. (2000): Biosensors and biochips: advances in biological and medical diagnostics, Fresenius Journal of Anal Chemistry, 366: 540-551.
[112] Vorstand und wissenschaftlicher Beirat der Bundesärztekammer, Hrsg. (2000): Richt-linien zur Gewinnung von Blut und Blutbestandteilen und zur Anwendung von Blutprodukten, Deutscher Ärzte-Verlag, Köln.
[113] Wallace, R. B., Shaffer, J., Murphy, R. F., Bonner, J., Hirose, T., Itakura, K. (1979): Hybridization of synthetic oligodeoxyribonucleotides to phi chi 174 DNA: the effect of single base pair mismatch, Nucleic Acids Research, 6, 3543-3557.
[114] Walz, G. (Redaktion) (2000): Lexikon der Mathematik, Spektrum Akademischer Verlag

[115] Wang, Liman, Liu, Qinghua, Corn, Rober M., Condon, Anne E. and Smith, Lloyd M. (2000): Multiple Word DNA Computing on Surfaces, J. Am. Chem. Soc.122:7435-7440.
[116] Wermter, S., Sun, R., Eds. (2000): Hybrid Neural Systems, Springer-Verlag.
[117] Werntges, H., Steger, G., Riesner, D. and Fritz, H.-J. (1986): Mismatches in DNA double strands: thermodynamic parameters and their correlation to repair efficiencies, Nucleic Acids Research, 3773-3790, Vol. 14.
[118] Wetmur, J. (1991): DNA Probes: Applications of the principles of nucleic acid hybridization, Crit. Rev. in Biochem. and Mol. Biol, 26, 227-259.
[119] Winfree, E., Furong, L., Wenzler, L. A., Seeman, N. C. (1998): Design and self-assembly of two-dimensional DNA crystals. Nature, 394: 539-544.
[120] Wood, D. W. et al. (2001): The Genome of the Natural Genetic Engineer *Agrobacterium tumefaciens* C58, Science, Vol. 294, 2317-2323.
[121] Xia, T., SantaLucia, J., Burkard, M. E., Kierzek, R., Schroeder, S. J., Jiao, X., Cox, C., Turner, D. H. (1998): Thermodynamic Parameters for an Expanded Nearest-Neighbor Modell for Formation of RNA Duplexes with Watson-Crick Base Pairs. Biochemistry, 37, p. 14719-14735.
[122] Zein, N. N., Persing, D. H., Czaja, A. J. (1999): Viral genotypes as determinants of autoimmune expression in chronic hepatitis C, Mayo Clin Proc, 74(5): 454-460.
[123] Zuker, M., Mathews, D.H. & Turner, D.H. (1999): Algorithms and Thermodynamics for RNA Secondary Structure Parameters: A Practical Guide. In RNA Biochemistry and Biotechnology, J. Barciszewski & B.F.C. Clark, eds., NATO ASI Series, Kluwer Academic Publishers.

Liste der verwendeten Symbole und Bezeichner

A	Nichtziel-Klasse bzw. Menge von →Nichtziel-Sequenzen bei einem (M, P, A)-„Set Cover"-Problem mit Spezifitäts-Nebenbedingung. Zumeist $A := \cup G_i$ für $i = 0$ und mehrere $i \neq 0$, $A \subset B^*$
B	die Menge bzw. das Alphabet der Basen {A, C, G, T}
B^*	Menge der Sequenzen/Wörter über dem Alphabet B
bp	Basenpaar
fn	Anzahl von falsch-negativen Signalen
fp	Anzahl von falsch-positiven Signalen
g	Anzahl der Sequenz-Klassen
g_N	Grenzwert für die →Nichtziel-Sequenzen zur Bestimmung von positiven und negativen Signalen
g_Z	Grenzwert für die →Ziel-Sequenzen zur Bestimmung von positiven und negativen Signalen
G_i	Sequenz-Klasse, $G_i \subset M' \subset B^*$
H(x, t)	...
isens(x, L)	Die während der Iterationen eines Greedy-Algorithmus verwendete „inkrementelle Sensitivität" eines Oligonukleotids $x \in K$ bei einem Zwischenergebnis L einer Oligonukleotid-Bibliothek.
$isens_r$(x, L)	Die „inkrementelle Sensitivität" eines Oligonukleotids $x \in K$, wie bei *isens*(x, L), unter zusätzlicher Berücksichtigung des →Redundanz-Niveaus r.
K	Menge aller Oligonukleotid-Kandidaten, $K \subset K' \subset B^*$
K´	Menge aller aus M´ ableitbaren Oligonukleotide (Teilsequenzen), $K' \subset B^*$
L_i	Oligonukleotid-Teilbibliothek, $L_i \subset L \subset K \subset K' \subset B^*$
L	Oligonukleotid-Bibliothek, $L \subset K \subset K' \subset B^*$
L´	nur in Abschnitt 2.5.1 verwendete Lösung des vereinfachten Greedy-Algorithmus mit $L' \subset P$. $L = Match^{-1}(L') \subset K$
L	speziell für das Gradientenabstiegs-Verfahren kodierte Oligonukleotid-Bibliotheken.
M	Menge von →Ziel-Sequenzen, zumeist $M := G_i$ für ein i, $M \subset M' \subset B^*$
M´	Menge aller →Ziel-Sequenzen einer Hierarchie, $M' \subset B^*$
Match(x)	Funktionswert von x unter der Abbildung $Match: K \rightarrow P$, die jedem Oligonukleotid x die Menge der „→Treffer" in M∪A zuordnet, $Match(x) \subset M \cup A$ und $Match(x) \in P$
m(t, L)	ordnet jedem $t \in M$ die Anzahl der →Treffer einer Oligonukleotid-Bibliothek L zu
$\mathbb{N}$	Menge der natürlichen Zahlen {0, 1, 2, 3, ...}
P	Menge aller verwendbaren Teilmengen der Grundmenge M bei einem (M, P)-„Set Cover"-Problem
$\wp(\bullet)$	Potenzmenge eine Menge; die Menge aller Teilmengen
$\mathbb{R}$	Menge der reellen Zahlen; z.B.: 0, 1, -½, $\sqrt{2}$, $\pi \in \mathbb{R}$
$\mathbb{R}^+$	Menge der positiven reellen Zahlen; $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$
r	Treffer-Redundanz bzw. →Redundanz-Niveau, $r \in \mathbb{N}$
rn	Anzahl von richtig-negativen Signalen
rp	Anzahl von richtig-positiven Signalen

s	→Toleranz-Niveau, $s \in \mathbb{N}$
$sens_r(L)$	Sensitivität einer ganzen Oligonukleotid-Bibliothek berechnet als rp / (rp + fn) durch die in Abschnitt 4.2.2 definierten Werte für rp, fn, rn und fp unter Berücksichtigung des →Redundanz-Niveaus r. Der Grenzwert g_Z (vgl. Abschnitt 4.2.1) geht ebenfalls in die Berechnung ein.
sens(x)	Sensitivität des Oligonukleotids $x \in K$
signal(x, t)	Die Abbildung signal(x, t) steht stellvertretend für entweder H(x, t) oder thdist(x, t).
$spez_s(L)$	Spezifität einer ganzen Oligonukleotid-Bibliothek berechnet als rn / (rn + fp) durch die in Abschnitt 4.2.2 definierten Werte für rp, fn, rn und fp unter Berücksichtigung des →Toleranz-Niveaus s. Der Grenzwert g_N (vgl. Abschnitt 4.2.1) geht ebenfalls in die Berechnung ein.
spez(x)	Spezifität des Oligonukleotids $x \in K$
t	→Ziel-Sequenz (engl. *target sequence*), $t \in M$
thdist(x, t)	...
T_M	Schmelztemperatur (engl. *melting temperature*)
x	Oligonukleotid, $x \in K \subset K' \subset B^*$; z.B. x = „GCTTAGGCTTAGGCGAT“
$\mathbf{x}$	eine fuzzy kodierte Oligonukleotid-Bibliothek, $\mathbf{x} \in X = \mathbb{R}^{\lvert K \rvert}$.

Glossar

Hier werden die, für diese Arbeit, zentralen Begriffe beschrieben und, unter dem Aspekt einer interdisziplinären Leserschaft, einige Grundbegriffe aus den beteiligten Disziplinen Mathematik, Informatik, (Molekular-)Biologie und →Thermodynamik erläutert. Glossare haben im Allgemeinen einen geringen wissenschaftlichen Anspruch und sind hauptsächlich als Hilfestellung für den Leser zu verstehen. Die Begriffe werden meistens sehr kontextbezogen ausgelegt. Die hier vermissten Begriffe können in den zahlreichen Glossaren im Internet nachgeschlagen werden:

- Bioinformatik: http://www.bioinf.org/molsys/glossary.html
- Molekularbiologie und forensische DNA-Analyse: http://www.rechtsmedizin.uni-mainz.de/Remedneu/molgen/dnaglos.htm
- Glossar Biochemie: http://www.biochemie.de/glossar-01.htm
- A Molecular Biology Glossary: University of Michigan DNA Sequencing Core: http://seqcore.brcf.med.umich.edu/doc/educ/dnapr/mbglossary/mbgloss.html
- Glossary of Genetic Terms: http://www.nhgri.nih.gov/DIR/VIP/Glossary
- Primer of Molecular Genetics: http://www.gdb.org/Dan/DOE/intro.html
- SGD Glossary Terms: http://genome-www.stanford.edu/Saccharomyces/help/glossary.html
- Human Genome Project Information: http://www.ornl.gov/hgmis/publicat/glossary.html
- Glossary of Biotechnology Terms: http://www.cs.washington.edu/homes/jbuhler/research/array/glossary.html
- LifeScience.de: http://www.lifescience.de
- NDI Foundation: http://www.ndif.org/t-a.html
- The On-line Medical Dictionary des CancerWeb: http://www.graylab.ac.uk/omd/
- Gensensorik: http://home.zait.uni-bremen.de/~mn/glossar/Glossar_Gensensorik.html
- Cancer glossary: http://www.pc-spes.com/Glossary/A.html-ssi
- http://www.sequenceanalysis.com/glossary.html
- Hypermedia Glossary of Genetic Terms: http://www.weihenstephan.de/~schlind/genglos.html

Algorithmus: „Eindeutiges, endlich beschreibbares und mechanisch durchführbares Verfahren zur Lösung einer bestimmten Problemklasse. Zu jedem Zeitpunkt des Verfahrens muss der Folgeschritt eindeutig durch den vorangegangenen Schritt festgelegt sein. Nach der Eingabe der jeweiligen Eingabedaten bricht das Verfahren nach endlich vielen Schritten ab und liefert das gesuchte Ergebnis“ [114]. Problemklassen sind beispielsweise das Suchen, Sortieren oder Arithmetik. Die in dieser Arbeit betrachteten Probleme gehören zur kombinatorischen Optimierung und werden als schwierige, sehr rechenintensive Probleme häufig approximativ (näherungsweise) gelöst.

Alignment: (engl. für „Ausrichtung“) Im Kontext der →Molekularbiologie ein →Algorithmus zum Vergleich zweier etwa gleichlanger →Sequenzen (globales Alignment; auch *Needleman-Wunsch algorithm*) bzw. zum Auffinden einer kürzeren Sequenz in einer längeren (lokales Alignment; „The version of the dynamic programming algorithm that performs local alignment of two sequences is known as the Smith-Waterman algorithm“ [34]).

antisense oligonucleotides (ASOs): Ein aus dem →antisense-Strang ausgeschnittenes →Oligonukleotid. „Antisense oligonucleotides are short DNA sequences, typically between 15-25 nucleotides, that can bind to a complementary →mRNA target by Watson-Crick base pairing and selectively inhibit the expression of the target gene from among the 80 000 or so estimated to be present in a typical mammalian cell. This in principle, makes possible the rational design of DNA-based therapeutic drugs for specific inhibition of any gene of known sequence. [...] Antisense oligonucleotides are also a useful

tool in biological studies of gene function“ [97]. Beim Design von ASOs müssen ähnliche Kriterien berücksichtigt werden wie bei dem Design von Fänger-Oligonukleotiden für DNA-Mikroarrays [19].

antisense-Strang: (auch Matrizenstrang, minus-Strang oder nicht-kodierender Strang) Der zur →mRNA reverse-komplementäre Strang. Ggs.: →sense-Strang.

Base: siehe bei →Nukleotid.

Basenpaarung: „Die Paarbildung zwischen zwei Basen in einem DNS-Molekül. Die Nukleinsäuren Adenin und Thymin sowie Guanin und Cytosin bilden jeweils ein charakteristisches Basenpaar. Die Paarbildung [auch Hybridisierung oder Assoziation] führt dazu, daß zwei DNS-Stränge sich zu einer Doppelhelix zusammenlagern“ [101].

bp: Abk. für Basenpaare. Siehe auch bei →kb.

cDNA: (Abk. für *complementary DNA*) „Die im Labor mit Hilfe des Enzyms reverse Transkriptase hergestellte Kopie einer →mRNA. Beliebt zum Klonieren von Genen [z.B. für →Genexpression], weil sie praktisch nur die Nettoinformation einer Erbanlage (ohne Introns) enthält.“ [101]. cDNA Klone sind zumeist mehrere tausend Basenpaare lange DNA-Stränge, die anhand von zellulärer mRNA transkribiert wurden und unter anderem als Fänger-Nukleinsäure für Genexpressions-Experimente auf DNA-Mikroarrays verwendet werden.

chip: engl. Synonym für ein Mikroarray mit immobilisierten Nukleinsäuren (überwiegend DNA-Mikroarrays) oder →Proteinen (auch Proteinchips).

Code: In der Codierungstheorie ist Code „die Bezeichnung für die endliche Menge nichtleerer Wörter [...], die das Bild der eineindeutigen Abbildung einer endlichen Menge von Nachrichten ist“ [114]. Der „Genetische Code“, als Abbildung von →Codons in der →mRNA zu einer von zwanzig Aminosäuren ist ein degenerierter Code, da die Abbildung nicht eineindeutig ist. Der resultierende Freiheitsgrad wird genutzt, um häufig auftretende Aminosäuren durch entsprechend mehr →Codons zu codieren. „Außerdem besteht eine Tendenz, ähnliche Aminosäuren (z.B. polare, hydrophobe, hydrophile etc.) durch ähnliche →Codons zu codieren“ [46].

Codon: Ein Codon ist eines von 64 möglichen Basentripletts aus der Menge $\{A, C, G, U\}^3$ = {AAA, AAC, ..., AUG (Startcodon), ..., GUU, UUU}. Ein Codon entspricht im „Genetischen →Code“ einer Aminosäure oder als Start-/ Stopcodon einem Steuerungssignal bei der →Genexpression.

Desoxyribonukleinsäure: (Abk.: DNA) Ein Kettenmolekül (Polymer) aus →Nukleotiden, deren →Basen in Form einer →Sequenz gerichtet, vom 5´- zum 3´-Ende, aufgeschrieben werden (Beispiel: 5´-ATCCGAAGCT-3´). Die D. „ist diejenige Substanz, in der in den meisten Organismen die Erbinformationen codiert (→Code) sind, die bei jeder Zellteilung an die Tochterzellen weitergegeben werden. [...] Chemische gesehen handelt es sich bei der DNA um ein unverzweigtes, hochmolekulares Polymer aus →Nucleotiden.“ [46].

DNA: international gebräuchliche Abk. für englisch *desoxyribonucleic acid*; deutsch: DNS für →Desoxyribonukleinsäure.

DNS: Abk. für →Desoxyribonukleinsäure.

EST: Abk. für *expressed sequence tag*; deutsch: exprimierte sequenzmarkierte Stelle. ESTs sind Teilsequenzen von →cDNA-Sequenzen. „ESTs are used for quick identification of genes and don’t cover the entire coding sequence of a gene“ [34]. EST ist weiterhin die

Bezeichnung für eine von mehr als fünf Sequenztypen (→mRNA, →cDNA, genomic --→DNA, EST, GSS) in der Sequenzdatenbank GenBank.

Exon: „Ein kodierender Abschnitt in einem aus Exons und →Introns bestehenden →Gen. Nur die Exons werden in eine Aminosäurekette übersetzt und werden zu einem Teil des →Proteins. Die als Introns bezeichneten Abschnitte erscheinen [in der Regel] nicht in dem Protein; sie werden bei der Reifung der →mRNA aus der RNA-Kopie eines Gens herausgeschnitten“ [101] (engl. *splicing*).

Functional Genomics: Das Studium der →Gene, ihrer resultierenden →Proteine und die Rolle dieser Proteine in den biochemischen Prozessen eines Organismus. [übersetzt aus: Human Genome Project Information; http://www.ornl.gov/hgmis/publicat/glossary.html]

Gelelektrophorese: Die Trennung von Bestandteilen von ionischen Lösungen in einem Gel bezüglich Unterschieden in der Geschwindigkeit ihrer Migration bei Anwendung eines elektrischen Feldes.

Gen: „Teil des Erbmaterials, der die genetische Information für einen bestimmten Zellbestandteil, eine →Ribonukleinsäure oder ein →Protein, enthält“ [101]. Dieser Teil ist der Abschnitt eines Chromosoms, der kodierende (→Exons) und nicht-kodierende (Introns; diese gibt es bei Bakterien nicht) Sequenzen umfasst und bei der →Genexpression zu RNA transkribiert wird. „There are three classes of genes. Protein coding genes [...], RNA specifying genes [... and silent/ inactive /] untranscribed genes“ [34], die „keinerlei Transcriptionsaktivität zeigen (auch Pseudogene). Diejenigen Gene, die in allen eukaryontischen Zellen, unabhängig von deren Spezialisierungsgrad, exprimiert werden, bezeichnet man gewöhnlich als Haushaltsgene [housekeeping gene]“ [46]. Gene werden häufig nach Funktion oder Zusammenhang bei der Ausbildung von Krankheiten benannt, wie das Onkogen k-RAS, das Tumorsuppressorgen DCC und für Prostatakrebs das Gen p53 [72].

Genetischer Code: →Code

Genexpression: [6], [7], [29] „Umsetzung der in einer Erbanlage [→Gen] gespeicherten Information in ein entsprechendes →Protein oder eine →Ribonukleinsäure“ [101]. Wird ein Gen in einem betrachteten Zellzustand exprimiert, so wird es „aktiv“ genannt. Die Aktivität der Gene wird über die Genregulation, mit Hilfe von regulatorischen Regionen, gesteuert. Durch Genexpressions-Experimente wird die Aktivität von Genen in verschiedenen Gewebetypen oder Krankheitsstadien analysiert. Bei der G. werden Teile der →DNA zunächst zu einer „unreifen“ RNA transcribiert (→Transcription), diese wird einem →splicing unterzogen und die so entstehende „reife“ →mRNA wird letztlich zu einer Aminosäuresequenz (→Protein) translatiert . Die →Translation realisiert den Genetischen →Code.

Genom: die Gesamtheit aller →Gene eines Organismus.

Gensonde: Synonym für →Oligonukleotid im Kontext der →Genexpression.

Gradientenabstiegs-Algorithmus: (engl. *gradient descent*, auch: Gradientenverfahren oder Verfahren des steilsten Abstiegs) Ein numerisches Lösungsverfahren für nicht-lineare Optimierungsprobleme, ein Verfahren zur Minimierung einer differenzierbaren Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Diese Funktionen werden häufig Kosten-, Fehler- oder allgemein Bewertungsfunktionen genannt. Beginnend mit einer zumeist zufällig gewählten Startposition $x_0 \in \mathbb{R}^n$ wird in einem iterativen →Algorithmus, zu jedem Schritt k, der Vektor $-\nabla f(x_k)$

des steilsten Abstiegs ermittelt und für die Berechnung von x_{k+1} verwendet: $x_{k+1} = x_k - \lambda \nabla f(x_k)$. Dabei ist $\lambda > 0$ die Schrittweite.

GVO: Abk. für „gentechnisch veränderter Organismus"; engl.: GMO für *genetically modified organism*.

Hybridisierung: 1. Der Übergang zweier Nukleinsäuren als Einzelstrang zu einem Doppelstrang. Zwischen den →Nukleotiden bzw. Basen der Nukleinsäuren kommt es dabei zur →Basenpaarung.
2. Die Durchführung eines Hybridisierungs-Experiments gemäß eines →Hybridisierungsprotokolls.

Hybridisierungsprotokoll: In dem H. werden die Bearbeitungsschritte (1. Spülung der Probe bzw. des Hybridisierungspuffers über das DNA-Mikroarray 2. Waschen zum Entfernen überschüssiger ungebundener Nukleinsäuren 3. Nachweis der Ziel-Nukleinsäuren) und Parameter eines Hybridisierungs-Experiments beschrieben. Zu den Parametern gehören der Salzgehalt, die Formamid- und Nukleinsäure-Konzentration des Hybridisierungspuffers, die Temperaturen beim Hybridisieren und Waschen und die Zeitdauer mit der hybridisiert und gewaschen wird. Mit dem H. werden für das Design von Oligonukleotid-Bibliotheken wichtige Parameter vorgegeben und die →Stringenz eines Hybridisierungs-Experiments bestimmt.

Intron: siehe bei →Exon.

kb: Abk. für Kilobase; Maßangabe für 1000 Basen bzw. 1000 **bp** für *base pairs*. kb ist nicht zu verwechseln mit der in der Informatik gebräuchlichen Maßangabe kB für 1024 Byte. So hat eine 10 kB große fasta-Datei mit 100 Byte für Titelzeile und Zeilenende-Zeichen 10,14 kb.

Kinetik: Die Lehre von der Bewegung durch Kräfte bzw. die Lehre von dem zeitlichen Verlauf thermodynamischer Prozesse. Bei der Hybridisierung und der Bildung von Sekundärstrukturen interessieren wir uns hauptsächlich für den Zustand dieser Systeme im Gleichgewicht und beschreiben diese mit der Schmelztemperatur T_m, der freien Enthalpie ΔG, der Enthalpie ΔH und der Entropie ΔS. Die Dynamik vor dem Erreichen des Gleichgewichtszustands wird durch die Kinetik beschrieben. Zwei in diesem Zusammenhang wichtigste Parameter des →Hybridisierungsprotokolls eines Experiments sind die Temperatur und die Dauer, mit der hybridisiert wird.

Konfigurierung: Zusammenstellung eines komplexen technischen Systems aus einzelnen Objekten zu einer Konfiguration. Beispiele für die K. sind neben der K. von DNA-Mikroarrays, die in dieser Arbeit synonym für Optimierung oder Design von DNA-Mikroarrays verwendet wird, die Erstellung von Konfigurationen für Computer oder auch die Anordnung von Einrichtungsgegenständen in einem Raum. Bei einer Konfigurierungs-Aufgabe ist folgendes gegeben: „eine Spezifikation [...]; eine Menge von Objekten und deren Eigenschaften; eine Menge von Relationen und Restriktionen zwischen den Objekten [...]; Wissen über die Vorgehensweise bei der Konfigurierung" [36]. Das Design von →Oligonukleotid-Bibliotheken ist ein typisches Konfigurierungsproblem, bei der eine Menge von →Oligonukleotiden (Objekte) bzgl. der Vorgaben eines →Hybridisierungsprotokolls und der biologischen Aufgabenstellung (Spezifikation) geeignet zusammengestellt wird.

Kontroll-Recherche: Bei der K. werden zu einer berechneten →Oligonukleotid-Bibliothek und einer gegebenen Menge von →Nichtziel-Sequenzen (im Abschnitt 4.1 definiert als die Menge G_0) die Anzahl der falsch-positiven →Treffer bestimmt. Zusammen mit der Anzahl der richtig-negativen Signale wird damit die →Spezifität berechnet.

Mb: Abk. für Megabase. Siehe auch bei →kb.

mfe-Struktur: Die Sekundärstruktur eines Ensembles von mehreren Strukturen mit der minimalen freien Energie.

Mismatch: engl. für Basenaustausch bzw. Basenfehlpaarung. Beispiel: Die zwei zueinander reverse-komplementären Sequenzen 5´-AGCTTCGGAT-3´ und 5´-ATCCGAAGCT-3´ haben keinen Mismatch zueinander. Wird nun in einer der Sequenzen eine Base ausgetauscht, dann spricht man von einem Mismatch an der entsprechenden Position. Die Mismatch-Anzahl ist ein einfaches Maß für die (Un-)Ähnlichkeit zweier Sequenzen. Ein allgemeineres Maß ist der Edit-Abstand.

Molekularbiologie: Die M. ist neben der Mikrobiologie und der Biochemie ein Teilgebiet der Biotechnologie. Sie gliedert sich auf in: →Functional Genomics, Genomics, Gentechnologie, Nukleinsäure-Analytik (DNA-Analytik, →Genexpression, →PCR usw.), Organismen-Identifikation, Phylogenetik (genotyping), Proteomics und →Sequenzierung.

mRNA: Abk. für englisch *messenger RNA*; Die Botenribonukleinsäure entsteht aus der →RNA durch das Entfernen (engl. *splicing*) der nichtkodierenden →Intron-Abschnitte.

multiples Alignment: Ein →Alignment von mehr als zwei →Sequenzen. Mit dem frei verfügbaren Programm CLUSTALW lassen sich multiple Sequenz Alignments erstellen.

Mutation: „Erbänderung durch Austausch eines Basenpaares gegen ein anderes (Punktmutation), Verlust von Basenpaaren (Deletion) oder Zufügen von Basenpaaren (Insertion)“ [101]. Eine durch eine Punktmutation entstandene Sequenzänderung wird →SNP genannt.

Nachweisgrenze: Grenze eines analytischen Verfahrens, bei der ein Analyt gerade noch nachgewiesen werden kann. Ein Beispiel aus dem Kontext der →Genexpressionsanalysen veranschaulicht eine Nachweisgrenze: “In all of these analyses, expression levels of high and medium abundance genes is not difficult to observe. This abundance refers to gene frequency of 1:10.000 to 1:50.000, and even 1:75:000 in the total →mRNA. Below these levels the signal to noise ratio becomes critical. Signal levels have been measured from spiked-in controls at 1:100.000 [...] This range of transcript abundance corresponds to about one to five copies per cell” [97]. Das Erfassen möglichst vieler Varianten z.B. hochvariabler Virengenome wird mit dem Begriff →Sensitivität beschrieben.

Neuronale Netzwerke: Künstliche Neuronale Netze sind ein Modell für Informationsverarbeitung, die an der Verarbeitung von Information der Nervenzellen in Gehirnen angelehnt ist. Massive Parallelität einer großen Anzahl einfacher Prozessoren, Fehlertoleranz gegenüber unsicheren und verrauschten Daten und das „Lernen“ aus einer vorgegebenen Mengen von Daten sind die wichtigsten Eigenschaften von künstlichen Neuronalen Netzen.

Nichtziel-Sequenz: (engl. *non-target sequence*) siehe bei →Ziel-Sequenz.

Nukleotid: „Untereinheit der DNS und RNS; besteht aus einer stickstoffhaltigen Base (Adenin, Guanin, Thymin oder Cytosin bei DNS; bei RNS anstelle von Thymin Uracil), einem Phosphatmolekül und einem Zuckerrest (Desoxyribose bei DNS; Ribose bei RNS). Im Laborjargon werden die Nukleotide bei Längenangaben von Nukleinsäuren einfachheitshalber als ‚Basen‘ bezeichnet“ [101].

Oligomer: Kettenmolekül mit wenigen Bausteinen. Im Kontext der →Molekularbiologie wird O. als Synonym für →Oligonukleotid verwendet.

Oligonukleotid-Bibliothek: Eine Menge von →Oligonukleotiden L, die bei Vorgabe von mehreren →Sequenzklassen G_i zusätzlich in Oligonukleotid-Teilbibliotheken L_i unterteilt ist.

Oligonukleotid: (auch Oligomer, Gensonde) Ein kurzes Kettenmolekül (Polymer) bestehend aus relativ wenigen (griechisch *„oligo": wenig*) →Nukleotiden bzw. Basen. Für das Design von DNA-Mikroarrays werden Oligonukleotide mit einer Länge von 15 bis 50 →Nukleotiden verwendet. Ein Oligonukleotid der Länge 50 wird auch als 50-mer bezeichnet.

Oligonukleotid-Redundanz: Eine →Ziel-Sequenz mit der Oligonukleotid-Redundanz r wird von einer →Oligonukleotid-Bibliothek r-mal durch Hybridisierungssignale, von möglichst an verschiedenen Positionen auf der Ziel-Sequenz hybridisierenden Oligonukleotiden, nachgewiesen; siehe auch bei →Spot-Redundanz.

ORF: Abk. für engl. *open reading frame*; deutsch: offenes Leseraster. Ein ORF ist eine Teilsequenz einer →DNA, die in ein →Protein translatiert werden könnte. Diese Teilsequenz entsteht durch Zerlegung der DNA-Sequenz in →Codons. Beispiel: ATGCATGGC →(ATG CAT GGC oder A TGC ATG GC oder AT GCA TGG C).

PCR: Abk. für englisch *polymerase chain reaction*; →Polymerasekettenreaktion

Phylogenie: Die Phylogenie klassifiziert durch das Studium evolutionärer Verwandtschaftsverhältnisse die Vielfalt biologischer Organismen. „Seit 1965 bedeutet Phylogenie molekulare Phylogenie. [...] Die auf molelularer Evolution beruhende Klassifizierung geht davon aus, dass die Geschichte eines Gens in seiner →Nukleotid-Sequenz aufgezeichnet ist, die Geschichte eines Organismus in der Summe seiner Gene" [90]. Die Verwandtschaftsverhältnisse werden in einer Baumstruktur (hierarchische Struktur) dargestellt.

Polymerasekettenreaktion: Abk. PCR; „Chemisch-enzymatisches Verfahren zum Vervielfältigen von DNS-Molekülen; vermag von einem einzigen Molekül in wenigen Stunden viele Millionen Kopien (die PCR-Produkte oder Amplifikate) herzustellen. Vielseitig anwendbar, z.B. um ein bestimmtes DNS-Fragment einer Genbibliothek soweit zu vermehren, daß man es genauer analysieren kann" [101].

Polymerisation: Mit P. wird die Bildung eines Kettenmoleküls (Polymer) bezeichnet, z.B. eine Nukleinsäure aus →Nukleotiden oder ein →Protein aus Aminosäuren.

Polymorphismus: „Individuelle Unterschiede in der Basensequenz [Anm.: oder Sequenzlänge]; auch in den nichtkodierenden Bereichen, also außerhalb der Gene; z.B. in den hochrepetitiven DNS-Abschnitten" [101]. Beispiele: →SNPs, Restriktionsfragment-Längenpolymorphismen.

Primer Design: Ein für die →Polymerasekettenreaktion notwendiges Design von →Oligonukleotiden als Startermoleküle.

probe: engl. für „Sonde", im Kontext von DNA-Mikroarrays auch „Fänger-Oligonukleotid" (→Oligonukleotid).

Protein: (auch Eiweiß) Ein P. ist eine Aminosäurekette (Polypeptid), die als Produkt der →Translation der →mRNA nach der →Genexpression entsteht. Die Reihenfolge der Bausteine dieser Kette (Aminosäurereste) werden durch den Genetischen →Code

bestimmt. Als Enzyme, Hormone und weitere funktionale Elemente gehören die P. zu den wichtigsten Bausteinen aller pflanzlichen und tierischen Zellen.

Redundanz: siehe bei →Spot-Redundanz und →Oligonukleotid-Redundanz.

Redundanz-Niveau: Das Redundanz-Niveau $r \in \mathbb{N}$ ist ein Parameter für den in Abschnitt 4.3 spezifizierten Optimierungs-Algorithmus, der vorgibt mit welcher →Oligonukleotid-Redundanz bzw. Treffer-Redundanz →Ziel-Sequenzen nachgewiesen werden sollen (vergleiche auch: →Toleranz-Niveau $s \in \mathbb{N}$).

Restriction mapping: Erstellung von Karten von Restriktionsfragmenten für eine gegebene DNA. Ohne Kenntnis der →Sequenz ist das *restriction mapping* ein aufwändiger labortechnischer Prozess. Siehe auch →Restriktionsenzyme.

Restriktionsenzyme: Restriktionsenzyme sind →Proteine, die eine DNA an bestimmten Positionen schneiden.

Reverse Transkriptase: Die Reverse Transkriptase ist ein Enzym, das zum Umschreiben von RNA in →cDNA verwendet wird. Es kommt in Retroviren vor, die damit ihr RNA-Genom (→Genom) in →DNA umschreiben, um es anschließend in ein Chromosom des Wirtes einzufügen. Die Reverse Transkriptase wird bei der →RT-PCR verwendet.

Ribonukleinsäure: (Abk. deutsch: RNS, international: RNA) „Nukleinsäure, die in der Regel als Kopie von DNS-Molekülen gebildet wird. In den Zellen höherer Organismen wird ein →Gen in eine identische RNS-Kopie ‚transkribiert' (umgeschrieben), das ‚Transkript' wird anschließend von den nichtkodierenden Introns befreit, die beiden Enden mit Schutzgruppen versehen und so eine mRNS hergestellt" [101].

RNA: international gebräuchliche Abk. für englisch *ribonucleic acid*; deutsch: RNS für →Ribonukleinsäure.

RT-PCR: Abk. für *reverse transcriptase polymerase chain reaction*; siehe auch →Reverse Transkriptase und →Polymerasekettenreaktion.

sense-Strang: (auch plus-Strang, codogener oder kodierender Strang) Derjenige DNA-Einzelstrang, der die gleiche Basensequenz wie die →mRNA besitzt. Ggs.: →antisense-Strang.

Sensitivität: Eine Bewertungsfunktion für die Güte eines zweiwertigen Klassifikators, die zusammen mit der →Spezifität betrachtet wird. Die S. beschreibt im Kontext dieser Arbeit die Vollständigkeit mit der ein Motiv, ein →Oligonukleotid oder eine Oligonukleotid-Bibliothek eine Menge von →Sequenzen „abdeckt". Ein hochsensitives Motiv beschreibt bzw. umfasst sehr viele Sequenzen einer vorgegebenen Menge von Sequenzen. Ein Oligonukleotid mit der Sensitivität 1 trifft (→Treffer) alle vorgegebenen →Ziel-Sequenzen. Berechnet wird die Sensitivität als Rate der *richtig-positiven* im Verhältnis zur Summe der *richtig-positiven* und *falsch-negativen*: $rp / (rp + fn)$. Siehe auch →Übereinstimmung und $sens(x)$ und $sens_r(L)$ in der „Liste der verwendeten Symbole". Gelegentlich wird die Sensitivität mit der →Nachweisgrenze einer Analysemethode verwechselt.

Sequenz: Im Kontext der →Molekularbiologie wird mit S. die Bausteinreihenfolge von Biopolymeren wie →DNA, →RNA oder →Proteinen bezeichnet. In großen, schnell wachsenden Sequenzdatenbanken werden diese Sequenzen, als Ergebnisse der →Sequenzierung, zusammen mit Annotationen gespeichert. Die Bioinformatik entwickelt zahlreiche →Algorithmen zur Verarbeitung von Sequenzen, wie Suche, →Alignment, Konstruktion und Design von Sequenzen mit vorgegebenen Eigenschaften. Für die Organis-

men-Identifikation und die Optimierung von →Oligonukleotid-Bibliotheken werden Mengen von vorgegebenen Sequenzen in →Ziel- und →Nichtziel-Sequenzen eingeteilt.

Sequenzierung: Man unterscheidet zwischen DNA- und →Protein-Sequenzierung. Bei der DNA-Sequenzierung wird die Abfolge der →Basen zu DNA-Molekülen einer vorgegebenen Probe bestimmt. Dieser Prozess ist weitgehend automatisiert und wird im "high-throughput"-Verfahren durchgeführt. Die Ergebnisse der Sequenzierung werden in Sequenzdatenbanken gespeichert, welche bei der Bestimmung der →Sensitivität und →Spezifität von →Oligonukleotiden eine wichtige Rolle spielen.

Sequenzklasse: Bei der Bearbeitung biologischer Fragestellungen werden mehrere Mengen von →Sequenzen betrachtet (z.B. bei der Detektion von Viren zusätzlich die Sequenzen des Wirtes). Eine solche Menge von Sequenzen wird allgemein als Sequenzklasse bezeichnet und in Abhängigkeit von der Aufgabenstellung als Menge von →Ziel- und →Nichtziel-Sequenzen.

SNP: (Abk. für englisch *single nucleotide polymorphism*; SNPs, sprich „snips“; auch Punktmutation) Spezielle Form eines →Polymorphismus; „Abweichung in nur einem einzigen Basenpaar in einem →Gen. Weit verbreitet, meist vermutlich ohne Konsequenz für die Funktion des entsprechenden →Proteins, zuweilen verantwortlich für eine Erbkrankheit. Wird als sehr genauer Orientierungspunkt bei der Suche nach Genen genutzt.“ [101]

Spezifität: Eine Bewertungsfunktion für die Güte eines zweiwertigen Klassifikators, die zusammen mit der →Sensitivität betrachtet wird. Eine große S. beschreibt im Kontext dieser Arbeit ein Motiv, ein Oligonukleotid oder eine Oligonukleotid-Bibliothek, die nur wenige →Treffer auf der Menge der →Nichtziel-Sequenzen hat. Berechnet wird die Spezifität als Rate der *richtig-negativen* im Verhältnis zur Summe der *richtig-negativen* und *falsch-positiven*: rn / (fp + rn). Siehe auch →Übereinstimmung und *spez*(x) und $spez_s(L)$ in der „Liste der verwendeten Symbole“.

Spot-Redundanz: Ein Oligonukleotid mit der Spot-Redundanz *n* wurde *n*-mal auf ein DNA-Mikroarray immobilisiert. Diese Form der Redundanz wird häufig zur Einschätzung der labortechnischen Reproduzierbarkeit von Hybridisierungssignalen eingesetzt. Siehe auch bei →Oligonukleotid-Redundanz.

Stringenz: →Hybridisierungsprotokolle werden stringent genannt, wenn sie so ausgelegt sind, dass Hybridisierungssignale mit möglichst wenigen unspezifischen Hybridisierungen zustande kommen. Eine hohe Stringenz hat eine Reduktion der Signalintensität und damit der →Nachweisgrenze zur Folge. Ein stringentes Hybridisierungsprotokoll hat beispielsweise einen hohen Formamid-Gehalt, eine geringe Salzkonzentration und eine relativ hohe Hybridisierungs-Temperatur.

STS: Abk. für engl.: *sequence tag site*; deutsch: sequenzmarkierte Stelle.

Thermodynamik: Die Lehre der Energieänderungen, die chemische und physikalische Vorgänge begleiten. Im Kontext dieser Arbeit wird hauptsächlich die freie Enthalpie ΔG von Hybridisierungen und Sekundärstrukturen betrachtet.

Tm: Abk. für engl. *melting temperature*; Schmelztemperatur.

Toleranz-Niveau: Das Toleranz-Niveau $s \in \mathbb{N}$ ist ein Parameter für den in Abschnitt 4.3 spezifizierten Optimierungs-Algorithmus, der für den Optimierungs-Algorithmus eine obere Grenze für die Anzahl von zugelassenen falsch-positiven Signalen darstellt (vergleiche auch: →Redundanz-Niveau $r \in \mathbb{N}$).

Transcription: Das zumeist im Zellkern stattfindende Umschreiben der →DNA in →mRNA. Die T. ist ein Teilschritt der →Genexpression.

Transcriptom: Die Menge aller Sequenztranskripte eines gegebenen Organismus [34]. Im Prozess der →Genexpression sind die Transkripte ein Zwischenprodukt auf dem Weg vom →Gen zum Genprodukt (→RNA oder →Protein).

Translation: Das an den Ribosomen stattfindende „Übersetzen" der →Codons der →mRNA in die Aminosäuresequenz der →Proteine. Diese Übersetzung realisiert den Genetischen →Code. Die T. ist ein Teilschritt der →Genexpression.

Treffer: Ein →Oligonukleotid $x \in K$ „trifft" eine →Ziel-Sequenz $t \in M$, wenn in Abhängigkeit des Kontextes eine der folgenden Bedingungen erfüllt ist:

- das Oligonukleotid x ist eine Teilsequenz der Ziel-Sequenz t; „*perfect match*"-Treffer
- bei der Hybridisierung zwischen x und t wird eine vorgegebene Anzahl von →Mismatches nicht überschritten; „*mismatch*"-Treffer
- bei der Hybridisierung zwischen x und t wird eine vorgegebene Schranke für thdist(x, t) nicht überschritten

Diese Kriterien sind Bedingung für die Generierung eines Hybridisierungssignals.

Übereinstimmung: Die Ü. ist wie die →Sensitivität und die →Spezifität eine Bewertungsfunktion für die Güte eines zweiwertigen Klassifikators. Die Ü. berechnet sich über (rp + rn) / (rp + fn + fp + rn). Der Nachteil dieser Bewertungsfunktion ist die Abhängigkeit von dem Quotienten (rp + fn) / (fp + rn), d.h. von Prevalenzen zwischen den zwei Klassen. [71]

Wallace Regel: [113] grobe Regel zur Berechnung der Schmelztemperatur:
$T_m = 2\ \#[AT] + 4\ \#[GC]$

Ziel-Sequenz: (engl. *target sequence*) Die, bei der Bearbeitung biologischer Fragestellungen, betroffenen →Sequenzklassen werden in →Ziel- und →Nichtziel-Sequenzen unterteilt. Die Ziel-Sequenzen entsprechen dem in einer Probe nachzuweisenden Organismus. Beispiel: Ziel-Sequenzen sind die Sequenzen des HCV-Genotyp-1b und Nichtziel-Sequenzen sind alle humanen und alle nicht-Genotyp-1b →Sequenzen. Die Definition der Ziel- und Nichtziel-Sequenzen ist eine Voraussetzung für die Berechnung der →Sensitivität und →Spezifität.