Algorithmen für das Design von DNA-Microarrays

Sven Rahmann

NWG Algorithmen und Stochastik für die Systembiologie,
AG Genominformatik, Technische Fakultät, Universität Bielefeld
Sven.Rahmann@cebitec.uni-bielefeld.de

Abstract: Wir formulieren und lösen algorithmische Probleme, die beim Entwurf und bei der Produktion von DNA-Microarrays, einem Hochdurchsatz-Hilfsmittel in der funktionalen Genomanalyse, auftreten; insbesondere wird die effiziente Auswahl Transkript-spezifischer Signatur-Oligonukleotide behandelt. Das Problem wird insbesondere dann komplex, wenn spezifische Oligonukleotide nicht existieren; für diesen Fall wird ein statistisches Gruppentest-Verfahren vorgeschlagen.

1 Übersicht

DNA-Microarrays oder DNA-Chips haben sich in den letzten Jahren zu einem wichtigen Werkzeug in der funktionalen Genomanalyse entwickelt. Verschiedene Technologie-Plattformen existieren; wir betrachten aber ausschließlich Oligonukleotid-Arrays hoher Dichte. Diese bestehen aus einer Glas- oder Quartzplatte, die an mehreren Tausend bis zu einer Million wohldefinierten und regelmäßig angeordneten Stellen (sog. *Spots*) mit einsträngigen DNA-Oligonukleotiden aus 8 bis 30 Nukleotiden beladen ist (typischerweise 25-mere). Auf einem Spot befinden sich mehrere Millionen DNA-Moleküle derselben Sequenz. Die Sequenz jedes Oligonukleotids ist so gewählt, dass sie als Signatur eines Gen-Transkripts dient, d.h. jeder Spot auf dem Chip repräsentiert durch seine Sequenz ein bestimmtes Transkript der zu untersuchenden Organismen.

In einem typischen Experiment werden aus einer Zell- oder Gewebeprobe die mRNA-Transkripte der exprimierten Gene extrahiert. Die zu den mRNA-Transkripten komplementären cDNAs oder cRNAs, die mit Fluoreszenz-Farbsoffen markiert sind, werden mit den Oligonukleotiden auf dem Chip hybridisiert; hierbei wird ausgenutzt, dass sich komplementäre DNA- und RNA-Basen spezifisch stabil aneinander binden. Anschließend kann an jedem Spot die Fluoreszenz-Intensität gemessen werden, die um so stärker ist, je mehr mRNA von dem ensprechenden Gen vorlag, d.h., je stärker das Gen in der Zellprobe exprimiert war. Auf diese Weise lassen sich Genexpressionsprofile von verschiedenen Zell- oder Gewebetypen erstellen und vergleichen. Beispielsweise erhofft man sich durch Betrachten der Unterschiede von gesunden Zellen und Tumorzellen, verschiedene Krebserkrankungen besser zu verstehen und gegebenenfalls den Therapieerfolg zu verbessern.

Die Genexpressionsanalyse mit DNA-Chips ist eine Technik, die Daten in großer Men-

ge und mit hohem Durchsatz liefert; sie ist allerdings nicht frei von Fehlerquellen: Jeder Schritt muss sorgfältig durchgeführt werden, damit die gemessenen Daten tatsächlich mit der Genaktivität korrelieren. Insbesondere muss der DNA-Chip sorgfältig entworfen und produziert werden. Mit den dabei auftretenden algorithmischen Problemen befasst sich diese Arbeit [Rah05]. Im einzelnen werden folgende Aspekte untersucht.

– Damit ein Spot genau ein bestimmtes Gen repräsentiert, müssen die dort aufgebrachten Oligonukleotide genspezifisch sein, d.h., unter den Hybridisierungsbedingungen ausschließlich an das Zieltranskript binden. Das Hauptproblem besteht darin, eine geeignete Menge von Oligonukleotiden für jedes Transkript auszuwählen.

– Ein grundsätzliches Problem dabei ist, den Zusammenhang zwischen dem Expressionsniveau eines Gens und der gemessenen Signalintensität an den Spots zu verstehen, insbesondere zunächst für ideal-spezifische Oligonukleotide. Von Sättigungseffekten abgesehen ist dieser Zusammenhang annähernd linear; die Proportionalitätskonstante, auch *Affinitätskoeffizient* genannt, hängt von vielen Faktoren ab. Ein wichtiger Faktor ist dabei die Bindungsstärke der Hybdridisierung, die mit Hilfe thermodynamischer Modelle (z.B. *nearest neighbor* Modelle) geschätzt wird.

– Während der Chipentwurfsphase sind die Affinitätskoeffizienten in der Regel unbekannt, da keine experimentellen Daten vorliegen, aus denen sie geschätzt werden könnten. Man muss daher auf Ersatzmaße zurückgreifen, die sich schnell durch Vergleich der Oligonukleotidsequenz mit den Transkriptsequenzen gewinnen lassen. Letzten Endes interessiert dabei nur die Entscheidung, ob ein Transkript stabil an ein Oligonukleotid bindet oder nicht. Wir treffen diese Entscheidung anhand des längsten gemeinsamen Teilworts (longest common factor; LCF) von Oligonukleotid und Transkript. Diese Wahl wird in Abschnitt 2 motiviert. Durch den Vektor der LCF-Längen eines Oligonukleotids mit allen Transkripten lässt sich das Kreuzhybridisierungs-Risiko approximieren.

– Abschnitt 3 bildet den algorithmischen Kern der Arbeit. Hier stellen wir einen effizienten Algorithmus zur Berechnung sogenannter *matching statistics* von allen Transkriptpaaren vor, der auf erweiterten Suffixarrays beruht. Aus den *matching statistics* lassen sich die gewünschten LCF-Vektoren für beliebige Oligonukleotide bestimmen. Da die Anzahl der zu berechnenden *matching statistics* in der Regel sehr groß ist, stellt sich das Problem der effizienten Repräsentierung dieser Werte. Wir nutzen die Struktur der Werte aus und führen den Begriff des Sprungs (*jump*) in *matching statistics* ein. Es genügt dann, die Position und das Niveau der Sprünge zu betrachten. In [Rah05] untersuchen wir weiter die statistische Häufigkeit von Sprüngen und geben ein Verfahren an, um aus den Sprüngen das Kreuzhybridisierungsrisiko von Oligonukleotiden zu schätzen. Weiterhin stellen wir Varianten der Algorithmen vor, die es erlauben, unterschiedliche Gewichtung auf schnellere Laufzeit oder genauere Ergebnisse zu legen. Wir illustrieren die Leistungsfähigkeit der entwickelten Verfahren am Beispiel der Oligonukleotidauswahl für *Saccharomyces cerevisiae* (Backhefe).

– Das LCF-Konzept aus Abschnitt 2 und die Algorithmen aus Abschnitt 3 erlauben es, einzelne Oligonukleotide effizient gemäß ihrer Spezifität zu klassifizieren. Das ursprüngliche Problem, nämlich eine passende *Menge* an Oligonukleotiden für jedes Transkript auszuwählen, ist damit noch nicht gelöst: Es sind zusätzliche Bedingungen einzuhalten. Bei-

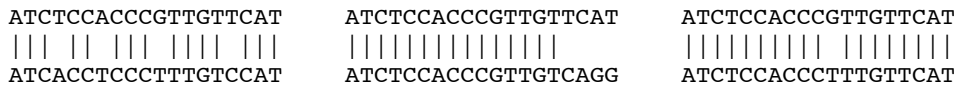
spielsweise möchte man ein Transkript gleichmäßig mit Oligonukleotiden abdecken; jedoch treten hochspezifische Oligonukleotide oft in Clustern auf, so dass man hier eine Wahl treffen muss. In der Praxis wird dieser Prozess teilweise manuell gehandhabt; in diesem Auszug gehen wir daher nicht näher darauf ein und verweisen auf [Rah05].

– Betrachtet man Gruppen von Transkripten (oder auch ganze Genome) mit hoher Sequenzähnlichkeit (z.B. alternative Spleißvarianten von Genen oder Subtypen desselben Virustyps), so stellt man fest, dass sich oft nicht genügend sequenzspezifische Oligonukleotide finden lassen. In diesem Fall lässt man zu, dass die Oligonukleotide mit mehreren der Sequenzen hybridisieren. Man beobachtet dann (theoretisch) die Summe der einzelnen Expressionswerte, oder bei qualitativer Anaylse mit binären Signalen das logische Oder der Einzelsignale. Die beobachteten Signale müssen also zunächst "dekodiert" werden. Auch das Oligonukleotid-Auswahlverfahren wird komplexer: Die Auswahl muss so erfolgen, dass robustes und effizientes Dekodieren möglich wird; andererseits sollte die Anzahl an Oligonukleotiden aus Wirtschaftilchkeitsgründen möglichst gering sein. Diese Forderungen werden im quantitativen Szenario als Konditionsminimierungsproblem einer (Sub-)Matrix formuliert, und im qualitativen Szenario als ein ganzzahliges lineares Programm [KRS+04, Rah05]. Zusätzlich werden schnelle Lösungsheuristiken angegeben. Abschnitt 4 geht kurz auf die grundlegenden Begriffe ein.

– Die diskutierten Oligonukleotid-Auswahlmethoden für die Genexpressionsanalyse setzen voraus, dass man das Transkriptom des zu untersuchenden Organismus kennt. Dieses ist im allgemeinen jedoch schwieriger zu bestimmen als das komplette Genom zu sequenzieren. Im Rahmen des Projekts ENCODE (ENCyclopedia Of Dna Elements[1]) wird versucht, die noch unbekannten Transkripte des Humangenoms innerhalb der genomischen Sequenz zu lokalisieren. Dazu wird das gesamte Genom mit Oligonukleotiden in geringem Abstand zueinander "gekachelt". Ein neues Transkript-Fragment ist entdeckt, wenn man für eine durchgehende Folge von Oligonukleotiden ein Signal sieht und dieses nicht durch bekannte Transkripte oder Kreuzhybridisierungen erklärt werden kann. Die Dissertation [Rah05] zeigt, wie die Methoden aus den vorangehenden Abschnitten effizient für diese Aufgabe eingesetzt werden können. Dabei ist insbesondere zu beachten, dass man die Oligonukleotide im allgemeinen nicht spezifisch auswählen kann.

– Zuletzt betrachten wir in [Rah05] ein Problem aus der Chipproduktion. Sind die Oligonukleotidsequenzen ausgewählt, so müssen sie auf dem Chip synthetisiert werden. Dies geschieht mit Hilfe photolithographischer Techniken und kombinatorischer Chemie. In einem Syntheseschritt werden ausgewählte Oligonukleotide um dasselbe eine Nukleotid verlängert. Aus Wirtschaftlichkeitsgründen und um die Fehlerzahl zu minimieren, soll die Anzahl der Syntheseschritte minimiert werden. Dies führt auf das NP-schwere Problem, die *kürzeste gemeinsame Supersequenz* von Tausenden sehr kurzer Sequenzen zu bestimmen. Wir stellen eine Heuristik für dieses Problem vor und berechnen die Verteilung von oberen und unteren Schranken für die Supersequenzlänge zufällig gezogener Oligonukleotide. Klassischerweise wird dasselbe Problem in der Informatik mit andereren Dimensionen betrachtet, nämlich für wenige lange Sequenzen mit großem Alphabet. Aus diesem Grund führen die aus der Literatur bekannten Heuristiken nicht zum Erfolg.

[1] `http://www.genome.gov/10005107`



(a) lcf $= 4$, lcf$^1 = 8$ matches $= 15$, len $= 19$

(b) lcf $= 15$, lcf$^1 = 16$, matches $= 15$, len $= 19$

(c) lcf $= 10$, lcf$^1 = 19$ matches $= 18$, len $= 19$

Abbildung 1: lcf-Werte, lcf^1-Werte, und Anzahl der matches für imperfekte probe-target Duplexe der Länge 19. Der mittlere Duplex ist deutlich stabiler als der linke und etwas stabiler als der rechte. Dies zeigt sich nicht in den match-Zahlen, lässt sich aber gut durch eine Kombination der lcf- und lcf^1-Werte modellieren.

Im Laufe der Zeit sind weitere Arbeiten zum Oligonukleotid-Auswahlproblem veröffentlicht worden (z.B. [LS01, KS02, RHZ02]); die diesem Beitrag zugrundeliegende Dissertation zeichnet sich dadurch aus, dass sie einen neuen Ansatz, den *Longest Common Factor* Ansatz, verfolgt und grundlegend thermodynamisch motiviert, und vor allem auch Methoden bereitstellt, um Fälle zu behandeln, in denen sich keine spezifischen Oligonukleotide bestimmen lassen.

2 Der *Longest Common Factor* Ansatz

Jede Oligonukleotid-Sonde (engl. *probe*) bindet unterschiedlich stark an ihr Ziel-Transkript (engl. *target*) und potenziell an weitere Transkripte; letzteren Effekt, die sogenannte Kreuz-Hybridisierung, möchte man durch geschickte Auswahl der Oligonukleotid-Sequenz vermeiden. Die exakte Vorhersage der Bindungsstärke in Form eines Affinitätskoeffizienten ist schwierig und bislang nicht genau untersucht, so dass wir ein sequenzbasiertes Ersatzmaß vorschlagen. Entscheidungen über das Risiko einer Kreuzhybridisierung können aufgrund dieses Ersatzmaßes getroffen werden.

Wir nutzen die Korrelation zwischen Bindungsstärke und dem längsten gemeinsamen Teilwort (*longest common factor*) zwischen Sonde p und Transkript t, die eine Folge der spezifischen Watson-Crick-Basenpaarung der DNA ist. Obwohl DNA physikalisch die Bindungen `A`-`T` und `C`-`G` eingeht, betrachten wir formal die probe-Sequenz als Teilwort der target-Sequenz. Wir schreiben $s \lhd t$, wenn s ein Teilwort von t ist; die Länge von s wird mit $|s|$ bezeichnet.

Definition 2.1 (Longest common factor). Ein gemeinsames Teilwort zweiter Wörter p und t ist ein Wort s mit $s \lhd p$ und $s \lhd t$. Ein gemeinsames Teilwort ist ein längstes gemeinsames Teilwort (*longest common factor*), wenn keine längeren gemeinsamen Teilwörter existieren. Wir schreiben $\text{lcf}(p, t) := \max\{|s| : s \lhd p \text{ und } s \lhd t\}$ für die *Länge* des longest common factor von p und t.

Die Wahl des lcf-Maßes lässt sich durch die folgenden physikalischen Umstände gut motivieren: Jede stabile Hybridisierung benötigt einen hinreichend stabilen Kern, mit dem die Hybridisierung beginnt. Dieser entspricht einem hinreichend langen gemeinsamen DNA-

Teilwort. Eine nicht perfekte Hybridisierung kann auch stabil sein; jedoch entsteht diese Stabilität durch mehrere perfekt übereinstimmede Teilwörter und wird durch die internen Fehler (Substitutionen oder Auslassungen) reduziert. Die Erfahrung zeigt, dass Duplexe mit derselben Zahl an übereinstimmenden Nukleotiden (*matches*) sehr unterschiedliche Stabilitäten aufweisen können, so dass die Gesamtzahl der matches kein gutes Maß darstellt (siehe auch Abbildung 1). In [PT02] und [ZMA03] wird gezeigt, dass der Stabilitätsverlust eines perfekten Duplexes durch Einfügen eines mismatches von der Position des mismatch abhängt; ein mismatch an zentraler Stelle führt zu einem stärkeren Stabilitätsverlust als ein mismatch in Randnähe. Diese positionale Abhängigkeit wird durch das lcf-Maß automatisch berücksichtigt. Eine genauere Quantifizierung kann durch die zusätzliche Berücksichtigung des längsten gemeinsamen Teilwortes mit maximal einem Fehler erreicht werden (lcf^1-Maß; siehe Abbildung 1).

Um target-spezifische Oligonukleotide für target t^* zu finden, bietet es sich an, $\mathrm{lcf}(p,t)$ für jeden Kandidaten $p \lhd t^*$ und jedes andere Transkript $t \neq t^*$ zu berechnen, um sicherzustellen, dass alle lcf-Werte klein sind und somit kein Kreuzhybridisierungs-Risiko besteht. Die Detailinformation, wie groß einzelne lcf-Werte genau sind, ist jedoch nicht notwendig; vielmehr lassen sich (a) hinreichend kleine lcf-Werte als äquivalent zu null behandeln, und (b) die Eigenschaften von p in Form der LCF-Statistik zusammenfassen.

Definition 2.2 (LCF-Statistik). Fixiere ein $\Delta > 0$. Die *LCF-Statistik* der Breite Δ für probe p gegen ein Transkriptom T (geordnete Menge von Transkripten) sei definiert als der Vektor $\mathrm{LCFS}(p \,|\, T; \Delta)$ mit

$$\mathrm{LCFS}(p \,|\, T; \Delta)_\delta := \#\{t \in T : \mathrm{lcf}(p,t) = |p| - \delta\} \qquad (\delta = 0, \ldots, \Delta - 1).$$

Die δ-Komponente gibt also an, wie viele Transkripte einen lcf-Wert mit p aufweisen, der um δ geringer ist als der Maximalwert $|p|$.

Erfahrungswerte zeigen, dass man gut mit $\Delta = 16$ arbeiten kann, wenn man an 25-mer probes (ein typischer Wert für Oligonukleotid-Chips) interessiert ist; dabei werden lcf-Werte von 9 oder darunter nicht mehr betrachtet.

Aus thermodynamischen Betrachtungen der sequenzabhängigen Hybridisierungsstabilität [San98, RG04] und Vernachlässigung anderer Effekte lässt sich das Kreuzhybridisierungsrisiko einer probe p approximieren als

$$U(p \,|\, T) := \sum_{\delta=0}^{\Delta-1} e^{-b\cdot\delta+\zeta} \cdot \mathrm{LCFS}'(p \,|\, T; \Delta)_\delta, \tag{1}$$

wobei wir $\mathrm{LCFS}'_\delta(\cdot) := \mathrm{LCFS}_\delta(\cdot)$ für $\delta > 0$ und $\mathrm{LCFS}'_0(\cdot) := \mathrm{LCFS}_0(\cdot) - 1$ setzen, um das beabsichtigte p-target nicht mitzuzählen. Kürzere lcf-Längen tragen also exponentiell weniger zum Kreuzhybridisierungsrisiko bei. Die Parameter b und ζ lassen sich z.B. aus thermodynamischen Modellen wie dem von SantaLucia [San98] bestimmen: In der diesem Beitrag zugrundeliegenden Dissertation [Rah05] werden unter realistischen Bedingungen die Werte $\zeta = 2.1972$ und $b = 1.4741$ hergeleitet.

Wir haben in diesem Abschnitt gezeigt, dass sich das Kreuzhybridisierungsrisiko eines probe-Kandidaten durch seine LCF-Statistik approximieren lässt; im nächsten Abschnitt betrachten wir das Problem der effizienten Berechnung der LCF-Statistik.

3 Berechnung der LCF-Statistik

Einfache Überlegungen zeigen, dass eine naive Berechnung der LCF-Statistiken für alle sinnvollen probe-Kandidaten auch für kleine Transkriptome ca. 10^{15} Zeichenvergleiche erfordern würde und damit impraktikabel ist. Die Verwendung von Indexstrukturen wie einem *enhanced suffix array* [AKO02, MM93], sowie die Ausnutzung weiterer Strukturen erlaubt eine beträchtliche Reduktion des Rechenaufwands bis hinunter in den Minutenbereich auf einer Einzelprozessormaschine.

Wir betrachten das DNA-Alphabet $\Sigma := \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ und zusätzliche Zeichen $\mathtt{X}$ (unbekanntes Nukleotid mit der Eigenschaft $\mathtt{X} \neq \mathtt{X}$ bei Vergleichen) und $\mathtt{\$}$ (Wort-Ende-Zeichen, ebenfalls mit $\mathtt{\$} \neq \mathtt{\$}$) und ordnen das erweiterte Alphabet, s.d. $\mathtt{A} < \mathtt{C} < \mathtt{G} < \mathtt{T} < \mathtt{X} < \mathtt{\$}$. Für ein Wort s sei $s_{(p)}$ das Suffix, das an Position p beginnt.

Aus dem gesamten Transkriptom $T = (t_1, \ldots, t_c)$ in der Form $T = t_1\mathtt{\$}\ldots\mathtt{\$}t_c$ wird ein enhanced suffix array erstellt.

Definition 3.1 (Enhanced suffix array). Sei $s' = (s_0, \ldots, s_{n-1})$ ein Wort der Länge n. Wir setzen $s := s'\mathtt{\$}$. Das erweiterte suffix array (*enhanced suffix array*) von s besteht aus mehreren Komponenten (Abb. 2):

- ein array $\mathtt{pos} = (\mathtt{pos}[0], \ldots, \mathtt{pos}[n])$ mit den Startpositionen der Suffixe von s in lexikographischer Ordnung, so dass $s_{(\mathtt{pos}[0])} < s_{(\mathtt{pos}[1])} < \cdots < s_{(\mathtt{pos}[n])}$; damit bilden alle Suffixe mit gemeinsamem Präfix ein Intervall in der durch $\mathtt{pos}$ gegebenen Sortierung,
- ein array $\mathtt{lcp}$ mit $\mathtt{lcp}[0] = 0$ und $\mathtt{lcp}[i]$:=Länge des längsten gemeinsamen Präfixes von $s_{(\mathtt{pos}[i-1])}$ und $s_{(\mathtt{pos}[i])}$,
- ein array $\mathtt{bck}$ (*bucket table*): Ein q-*bucket* von $\mathtt{pos}$ ist ein Intervall $[l, r]$ mit $\mathtt{lcp}[l] < q$, $\mathtt{lcp}[r+1] < q$ und $\mathtt{lcp}[i] \geq q$ für alle $i = l+1, \ldots, r$. Definiere einen Code $\langle a \rangle$ für jedes $a \in \Sigma$ wie folgt: $\langle \mathtt{A} \rangle := 0$, $\langle \mathtt{C} \rangle := 1$, $\langle \mathtt{G} \rangle := 2$, and $\langle \mathtt{T} \rangle := 3$. Für ein q-Wort $Q = (Q_0, \ldots, Q_{q-1}) \in \Sigma^q$ sei $\langle Q \rangle := \sum_{i=0}^{q-1} 4^{q-1-i} \langle Q_i \rangle$. Nun sei $\mathtt{bck}[\gamma]$ definiert als das linke Ende l des q-bucket $[l, r]$ des q-Wortes Q mit $\langle Q \rangle = \gamma$, falls $Q \lhd s$, und $\mathtt{bck}[\gamma] := \infty$ sonst ($\gamma = 0, \ldots, 4^q - 1$).
- ein *collection number* ($\mathtt{cl}$) array, in dem $\mathtt{cl}[i]$ der Index j des Transkripts t_j ist, an dem das Suffix $T_{(\mathtt{pos}[i])}$ von T beginnt.

Im Jahr 2003 wurde gezeigt, dass sich ein enhanced suffix array direkt durch einen ternären Divide-and-Conquer-Ansatz [KS03] in linearer Zeit berechnen lässt, auch ohne Hilfsmittel wie den Suffixbaum mit Suffixlinks.

Mit Hilfe eines enhanced suffix arrays lassen sich effizient sogenannte *matching statistics* für jedes target s gegen jedes Transkript t aus T berechnen. Aus diesen kann dann wiederum effizient $\mathrm{lcf}(p, t)$ für jedes Teilwort $p \lhd s$ gewisser Länge gewonnen werden.

Definition 3.2 (Matching statistics). Die *matching statistics* von s gegen t sind $\mathrm{ms}^{s|t} = \mathrm{ms} = (\mathrm{ms}_0, \ldots, \mathrm{ms}_{|s|-1})$, wobei ms_i die Länge des längsten Präfixes von $s_{(i)}$ ist, das

p	s_p
0	C
1	A
2	C
3	A
4	C
5	$
6	A
7	C
8	C
9	$

$\Rightarrow$

i	pos$[i]$	$s_{(\text{pos}[i])}$	lcp$[i]$	cl$[i]$
0	1	ACAC$...	0	103
1	6	ACC$	2	27
2	3	AC$...	2	103
3	0	CACAC$...	0	103
4	2	CAC$...	3	103
5	7	CC$	1	27
6	4	C$...	1	103
7	8	C$	1	27
8	5	$...	0	103
9	9	$	0	27

;

γ	bck$[\gamma]$
$\langle$AA$\rangle = 0$	∞
$\langle$AC$\rangle = 1$	0
$\langle$CA$\rangle = 2$	3
$\langle$CC$\rangle = 3$	5

Abbildung 2: Enhanced suffix array der Konkatenation s von CACAC$ und ACC$ mit hypothetischen collection numbers 103 (Positionen 0–5) und 27 (Positionen 6–9).

irgendwo in t vorkommt. Die Definition lässt sich erweitern, indem man eine Maximalzahl an Fehlern (Mismatches) zulässt.

Da kurze Matches (z.B. Längen unterhalb von $R^0_{\min} := 10$) für die Oligonukleotid-Auswahl uninteressant sind und lange Matches ebenfalls nur bis zur maximalen Oligonukleotidlänge interessieren (z.B. $R_{\max} := 25$), stellt sich folgendes Problem:

Für ein gegebenes target s und mehrere Transkripte t_c ($c = 1, \ldots, C$), berechne Approximationen MS$[i][c]$ ($i = 0, \ldots, |s| - 1$; $c = 1, \ldots, C$) der matching statistics $\text{ms}_i^{s|t_c}$ mit

$$\begin{array}{lcll} \texttt{MS}[i][c] & = & \text{ms}_i^{s|t_c}, & \text{falls} \quad \text{ms}_i^{s|t_c} \in [R^0_{\min}, R_{\max} - 1], \\ \texttt{MS}[i][c] & < & R^0_{\min}, & \text{falls} \quad \text{ms}_i^{s|t_c} < R^0_{\min}, \\ \texttt{MS}[i][c] & \geq & R_{\max}, & \text{falls} \quad \text{ms}_i^{s|t_c} \geq R_{\max}; \end{array} \tag{2}$$

Um das gestellte Problem zu lösen, setzen wir $q \leq R^0_{\min}$ voraus. Für jede Startposition i des targets s wird zunächst der Code $\gamma = \langle s_{i..(i+q-1)} \rangle$ des q-Wortes an Position i bestimmt und dann als Index für die bucket table verwendet. Es sei $r := \texttt{bck}[\gamma] \neq \infty$ (im Falle $r = \infty$ ist nichts zu tun). Der *bucket scan* Algorithmus (Abb. 3) wird sodann für den q-bucket, der bei Index r in pos beginnt, durchgeführt. Der Schlüssel zum Verständnis ist, einzusehen, dass für jeden Index r die Matchlänge $\mu := \text{lcp}(s_{(i)}, t_{(\text{pos}[r])})$ korrekt aktualisiert wird. Hierbei ist $\text{lcp}(x, y)$ die Länge des längsten gemeinsamen Präfixes von x und y, schon typographisch nicht zu verwechseln mit dem lcp-array.

Der Speicherverbrauch des MS-array kann relativ groß sein, insbesondere wenn s lang ist oder viele Transkripte vorliegen. Der benötigte Speicher lässt sich drastisch verkleinern, wenn man nur *Sprünge* in den MS-Werten abspeichert und folgende Struktur ausnutzt:

$$\text{ms}_i^{s|t} \geq \text{ms}_{i-1}^{s|t} - 1 \quad \text{für alle } i = 1, \ldots, |s| - 1.$$

Definition 3.3 (Sprünge in matching statistics). Ein *Sprung* liegt an Position $i > 0$ in $\text{ms}^{s|t}$ vor, genau dann wenn $\text{ms}_i^{s|t} \neq 0$ und $\text{ms}_i^{s|t} > \text{ms}_{i-1}^{s|t} - 1$. In diesem Fall nennen wir $J := \text{ms}_i^{s|t}$ das *Sprungniveau*.

Bucket Scan

Eingabe: Target s; Aktuelle Position i;
Transkriptom $T = (t_1, \ldots, t_C)$ und sein
Enhanced suffix array `pos`, `cl`, `lcp`;
Startposition r des q-buckets von $s_{i..i+q-1}$ ($r = \texttt{bck}[\langle s_{i..(i+q-1)}\rangle]$);
Maximal relevante Matchlänge $R_{\max}$;
Anfänglich $\texttt{MS}[i][c] = 0$ für alle $c = 1, \ldots, C$

Ausgabe: Approximative matching statistics $\texttt{MS}[i][c]$ gemäß (2) für alle $c = 1, \ldots, C$

1. $\mu \leftarrow q + \min(\text{lcp}(s_{(i+q)}, t_{(\texttt{pos}[r]+q)}),\ R_{\max} - q)$
2. $\texttt{MS}[i][\texttt{cl}[r]] \leftarrow \mu$
3. $r \leftarrow r + 1$
4. **while** $(\texttt{lcp}[r] \geq \mu)$
5. **if** $(\texttt{lcp}[r] = \mu)$ **then**
   $\mu \leftarrow \mu + \min(\text{lcp}(s_{(i+\mu)}, t_{(\texttt{pos}[r]+\mu)}),\ R_{\max} - \mu)$
6. $\texttt{MS}[i][\texttt{cl}[r]] \leftarrow \mu$
7. $r \leftarrow r + 1$
8. **while** $(\texttt{lcp}[r] \geq q)$
9. **if** $(\texttt{lcp}[r] < \mu)$ **then** $\mu \leftarrow \texttt{lcp}[r]$
10. **if** $(\texttt{MS}[i][\texttt{cl}[r]] < \mu)$ **then** $\texttt{MS}[i][\texttt{cl}[r]] \leftarrow \mu$
11. $r \leftarrow r + 1$

Abbildung 3: Der *bucket scan* stellt den Kern der matching statistics Berechnung dar.

Der Bucket Scan lässt sich leicht so modifizieren, dass nicht mehr alle Werte $\texttt{MS}[i][c]$ abgespeichert werden, sondern für jedes c nur noch die Sprunglisten aus Position und Niveau. Die theoretische Ersparnis für zufällige Wörter lässt sich durch stochastische Betrachtungen ermitteln [Rah05].

Durch geschicktes Auswerten der Sprunglisten von $\text{ms}^{s|t}$ lässt sich $\text{lcf}(p, t)$ für nach Startpositionen sortierte Listen von Teilwörtern $p \lhd s$ (Oligonukleotid-Kandidaten) effizient berechnen; daraus kann dann die LCF-Statistik für p gewonnen werden.

4 Behandlung nichtspezifischer Oligonukleotide

Die Erfahrung zeigt, dass es oftmals schwierig ist, für jedes target genügend spezifische probes zu finden, beispielsweise aufgrund hoher Sequenzähnlichkeit mehrerer Transkripte nach Genduplikationen oder bei der Betrachtung mehrerer alternativer Spleißformen von Genen. Da jedoch nichtspezifische probes den Nachteil haben, dass die gemessenen Signalintensitäten eine Summe aus mehreren (mit unbekannten Affinitätskoeffizienten gewichteten) Genexpressionslevels darstellen, ist eine Behandlung des Falls, in dem man beliebige probes zulässt, nicht anzuraten. Wir lassen daher zwar zu, dass mehrere targets s mit $p \lhd s$ existieren, verwerfen aber weiterhin probes p, für die Transkripte t existieren mit $\text{lcf}(p, t) = |p| - \delta$ für kleine Werte $\delta > 0$ (in der Praxis $\delta \in \{1, \ldots, 7\}$). Dies erhöht gera-

de im Falle mehrerer stark ähnlicher targets die Erfolgschancen, und grundsätzlich können die Verfahren zur probe-Auswahl der vorangehenden Abschnitte weiter benutzt werden.

Es stellt sich jedoch zusätzlich das Dekodierungs-Problem der gemessenen Signale. Aus Platzgründen verzichten wir hier auf eine quantitative Diskussion und betrachten nur den Fall binärer Signale, wobei der Spot von probe p ein Signal ("an") anzeige, wenn *mindestens eines* der targets von p "aktiv" (hochexprimiert) ist. Da diese logische Oder-Operation auf einem Spot nicht invertierbar ist, muss schon in der Design-Phase des Chips sichergestellt werden, dass zumindest verschiedene Kombinationen von probes verschiedene Kombinationen von hochexprimierten Targets detektieren können.

Definition 4.1 (d-Trennbarkeit von target-Mengen). Sei S eine Indexmenge von targets. Wir sagen, p hybridisiert mit S, wenn p an mindestens ein target in S hybridisiert ($p \lhd s_j$ für ein s_j mit $j \in S$). Es sei $P(j)$ die Menge aller probes, die an s_j hybridisieren und $P(S) := \bigcup_{j \in S} P(j)$. Nun seien S und T zwei verschiedene target-Mengen. Probe p_i *trennt* S und T, wenn $i \in P(S) \Delta P(T)$, d.h., falls p_i entweder mit S oder mit T hybridisiert. S und T sind *d-trennbar*, falls es mindestens d trennende probes gibt, d.h., falls $|P(S) \Delta P(T)| \geq d$.

Wir betrachten folgendes Design-Problem: Gegeben sei eine Menge von probe-Kandidaten und eine Hybridisierungsmatrix H mit $H_{ij} = 1$, wenn Kandidat p_i an target s_j hybridisiert, und $H_{ij} = 0$ sonst. Wähle eine möglichst kleine Teilmenge an probes aus, so dass jedes target an eine vorgegebene Mindestzahl von ausgewählten probes hybridisiert und je zwei target-Mengen (von geeignet beschränkter Kardinalität) d-trennbar sind für ein vorgegebenes d. In [KRS+04, Rah05] wird eine schnelle inkrementelle Heuristik und eine exakte Lösung, die auf ganzzahliger linearer Programmierung basiert, angegeben.

Die Dekodierung der ver-oder-ten Signale, die zusätzlich fehlerbehaftet sein können, erfolgt mit einem Markov chain Monte Carlo Verfahren, bei dem iterativ jeweils eine Erklärung für die beobachteten Signale vorgeschlagen, evaluiert und stochastisch modifiziert wird, so dass Stichproben aus der Menge aller Lösungen gemäß ihrer Plausibilität (Likelihood) oder a-posteriori-Wahrscheinlichkeit besucht werden. Im Rahmen der McMC-Prozedur lassen sich Verteilungen weiterer Eigenschaften der gesuchten target-Menge bestimmen, wie etwa ihre Kardinalität oder die Randverteilung für jedes einzelne j, dass target s_j aktiv ist, bei der gegebenden Datenlage. Die Ergebnisse zeigen, dass insbesondere bei Fehlerraten von 5–10%, die in der Praxis unweigerlich auftreten, auch in Simulationen das wahre target-Set nicht immer exakt rekonstruiert werden kann, die Erfolgsquote jedoch in praktisch relevanten Fällen oft über 95% beträgt.

Literatur

[AKO02] Mohamed Ibrahim Abouelhoda, Stefan Kurtz und Enno Ohlebusch. The Enhanced Suffix Array and Its Applications to Genome Analysis. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics (WABI 2002)*, Jgg. 2452 of *LNCS*, Seiten 449–463. Springer, 2002.

[KRS+04] Gunnar W. Klau, Sven Rahmann, Alexander Schliep, Martin Vingron und Knut Reinert.

Optimal robust non-unique probe selection using Integer Linear Programming. *Bioinformatics*, 20(Suppl.1):i186–i193, 2004.

[KS02] Lars Kaderali und Alexander Schliep. Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, 18(10):1340–1349, 2002.

[KS03] Juha Kärkkäinen und Peter Sanders. Simple linear work suffix array construction. In *Proceedings of the 13th International Conference on Automata, Languages and Programming (ICALP)*, Jgg. 2719 of *LNCS*, Seiten 943–955. Springer, 2003.

[LS01] Fungen Li und Gary Stormo. Selection of optimal DNA oligos for gene expression analysis. *Bioinformatics*, 17(11):1067–1076, 2001.

[MM93] Udi Manber und Gene W. Myers. Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[PT02] Alexander E. Pozhitkov und Diethard Tautz. An algorithm and program for finding sequence specific oligo-nucleotide probes for species identification. *BMC Bioinformatics*, 3(9), 2002.

[Rah05] Sven Rahmann. *Algorithms for probe selection and DNA microarray design*. Lecture Notes in Computer Science. Springer, Heidelberg, 2005. Zur Publikation angenommen.

[RG04] Sven Rahmann und Christine Gräfe. Mean and variance of the Gibbs free energy of oligonucleotides in the nearest neighbor model under varying conditions. *Bioinformatics*, 20(17):2928–2933, 2004.

[RHZ02] Jean-Marie Rouillard, Christopher J. Herbert und Michael Zuker. OligoArray: Genome-scale oligonucleotide design for microarrays. *Bioinformatics*, 18(3):486–487, 2002.

[San98] J. SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences of the U.S.A.*, 95:1460–1465, 1998.

[ZMA03] Li Zhang, Michael F. Miles und Kenneth D. Aldape. A model of molecular interactions on short oligonucleotide arrays. *Nature Biotechnology*, 21(7):818–821, 2003.

Sven Rahmann wurde am 1. August 1974 in Hamburg geboren. Sein Mathematik-Studium mit Nebenfach Informatik in Göttingen, Santa Cruz (Kalifornien) und Heidelberg wurde zunächst von der Studienstiftung des Deutschen Volkes gefördert. Im Rahmen seiner Diplomarbeit am Deutschen Krebsforschungszentrum wandte sich Herr Rahmann kombinatorischen und stochastischen Problemen zu, die bei der Sequenzanalyse in der Bioinformatik auftreten (“Word Statistics in Random Texts and Applications to Computational Molcecular Biology”). Er promovierte am Fachbereich Mathematik und Informatik der FU Berlin bei Professor Vingron mit der Arbeit “Algorithms for Probe Selection and DNA Microarray Design”, die am Max-Planck-Institut für Molekulare Genetik angefertigt wurde. Eine im Rahmen der Arbeit entstandene Veröffentlichung erhielt auf der IEEE Computer Society Bioinformatics Conference 2002 den Best Paper Award. Seit März 2004 ist Herr Rahmann Nachwuchsgruppenleiter für Algorithmen und Stochastik der Systembiologie an der Technischen Fakultät und am Institut für Bioinformatik des CeBiTec der Universität Bielefeld. Er ist Mitglied des Programmkomitees der IEEE CSB Konferenz 2004 und Co-Chair des ACM Symposium on Applied Computing Bioinformatics Track 2005.