# Automatisches Segmentieren von Mikroarraybildern

Dissertation zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

der

Technischen Fakultät der Universität Bielefeld

vorgelegt von

**Mathias Katzer**

Dipl.-Inform. Mathias Katzer
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: mkatzer@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 22. Oktober 2003 vorgelegt von Mathias Katzer.

Gutachter:

Prof. Dr. Franz Kummert
PD Dr. Anke Becker

Prüfungsausschuss:

Prof. Dr. Jens Stoye
Prof. Dr. Franz Kummert
PD Dr. Anke Becker
Dr. Christian Bauckhage

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

# Danksagung

Mein besonderer Dank gilt denjenigen, die mir durch Diskussionen, und Anregungen geholfen haben, diese Arbeit voranzubringen. Neben den Betreuern Prof. Dr. Kummert und PD Dr. Anke Becker möchte ich hier besonders Prof. Curtis Altmann (Florida State University, Tallahassee, USA) und Alexander Sczyrba (Arbeitsgruppe Praktische Informatik, Universität Bielefeld) nennen.

Auch die Unterstützung durch meine Familie hat nicht weniger zum Gelingen dieser Arbeit beigetragen. Ebenfalls danke ich den Mitgliedern der Arbeitsgruppe Angewandte Informatik und des Graduiertenkollegs Bioinformatik für die gute Arbeitsatmosphäre und Zusammenarbeit.

Darüberhinaus bin ich allen zu Dank verpflichtet, die ihre Bilddaten zur Verfügung gestellt haben:

A. Becker, H. Küster, Zentrum f. Genomforschung, Universität Bielefeld
M. Jones, Chugai Pharm., Tokyo
M. Cherry, T. Boussard, Stanford University, USA
J. Landgrebe, Universität Göttingen
A. Sporman, Stanford University, USA
S. Huang, Sloan Kettering Memorial Cancer Center, New York, USA
A. Viale, Rockefeller University, New York, USA
T. Speed, University of California at Berkeley, USA
D. Kreil, FlyChip, Cambridge, UK
M. Beyrouthy, Florida State University, Tallahassee, USA
V. Brendel, Iowa State University, USA

Göttingen, Januar 2004

## Abstract

Gene expression experiments using microarray hybridisation have become a widespread method in scientific as well as industrial research. Analysis of microarray images is a bottleneck of array data analysis pipelines, as it is usually performed using interactive computer programs. Apart from practical concerns, automation of microarray gridding and feature segmentation is most important to achieve constant data quality, which is a precondition to the integration of different expression data sets. Therefore, image processing methods that are applicable regardless of the employed array design and laboratory protocols are highly useful.

In this work a Markov random field (MRF) based approach to high level grid segmentation is proposed, which is robust to common problems encountered with array images and does not require calibration. The MRF framework allows to separate the heuristic modeling of spot grid layouts from the segmentation algorithm itself.

Also proposed is an active contour method for spot signal segmentation. Active contour models describe objects in images by local properties of their boundaries and thereby enable robust segmentation of irregularly shaped array spots. The traditional active contour model must be generalized for successful application to microarray spot segmentation.

The methods proposed in this work are implemented in the AIM (Automatic Image processing for Microarray experiments) system. The results of the system evaluation using a sample of 23 different types of microarray images show the usefulness of the MRF grid segmentation approach. The evaluation of quantitative image analysis is much more difficult since it seems hardly possible to produce authoritative as well as biologically relevant calibration data. The quantitative analysis of array spots using the active contour model reproduces the results of a fine tuned interactive image analysis with a commercial image processing tool (Imagene). Active contour segmentation is less sensitive to variations of grid segmentation than the well known Mann-Whitney segmentation.

# Inhaltsverzeichnis

**Inhaltsverzeichnis** v

**1 Einführung** 1

**2 Biologische Grundlagen** 3
2.1 Zellen . . . 3
2.2 Proteine . . . 3
2.3 Die Erbinformationsträger DNA und RNA . . . 5
2.3.1 Restriktionsenzyme . . . 8
2.3.2 Die Ribonukleinsäure . . . 9
2.4 Genexpression . . . 9
2.4.1 Gene und ihre Struktur . . . 9
2.4.2 Transkription . . . 10
2.4.3 Translation und Synthese . . . 12
2.5 Genregulation . . . 13
2.6 Zusammenfassung . . . 15

**3 Parallelisierte Hybridisierungsansätze** 17
3.1 Motivation . . . 17
3.2 Vorläufertechnologien der Mikroarray-Hybridisierung . . . 18
3.2.1 Southern-Blot . . . 18
3.2.2 Kolonie- und Membran-Hybridisierung . . . 20
3.2.3 Mikroarray-Hybridisierung . . . 20
3.3 Gedruckte DNA-Mikroarrays . . . 23
3.3.1 Herstellung von Mikroarrays . . . 23
3.3.2 Hybridisierung . . . 25
3.3.3 Bildaufnahme . . . 26
3.3.4 Differentielle Expressionsanalyse . . . 27
3.3.5 Typische Anwendungen von Mikroarray-Hybridisierung . . . 32
3.3.6 Expressionsdatenbanken . . . 33
3.4 Automatische Mikroarray-Bildverarbeitung . . . 34
3.4.1 Problembestimmung und Motivation . . . 34
3.4.2 Eigenschaften von Mikroarraybildern . . . 35
3.5 Zusammenfassung . . . 36

**4 Systemansätze** **39**
4.1 Ziele . . . 39
4.2 Andere Ansätze und Methoden . . . 39
4.2.1 Gittersegmentierung-Problemorientierte Ansätze . . . 40
4.2.2 Gittersegmentierung -Methodenorientierte Ansätze . . . 41
4.2.3 Signal-Segmentierung und Quantitative Auswertung . . . 44
4.3 Entwurf einer Systemstruktur . . . 46
4.3.1 Bildverarbeitungskette . . . 46
4.3.2 Ein Gesamtsystementwurf . . . 48

**5 Regionensegmentierung** **51**
5.1 Vorverarbeitung . . . 51
5.1.1 Histogrammbasierte Medianberechnung nach Huang . . . 51
5.1.2 Histogrammhierarchie . . . 52
5.1.3 Heap-Median . . . 53
5.2 Schwellwertverfahren . . . 54
5.2.1 Grundlagen . . . 54
5.2.2 Gewichtete Histogramme . . . 55
5.2.3 Verfahren zur Berechnung von Intensitätsschwellwerten . . . 56
5.3 Zusammenfassung . . . 57

**6 Gittersegmentierung** **59**
6.1 Nächste-Nachbar-Abstände und Regionenklassifikation . . . 59
6.1.1 Verteilungsmodelle für Regionenabstände . . . 59
6.1.2 Schätzung der Gitterabstände . . . 62
6.1.3 Klassifikation und Kanalkorrespondenz von Regionen . . . 62
6.1.4 Verkettung von Objekten . . . 64
6.1.5 Effiziente Nachbarsuche . . . 65
6.2 Achsenprojektionen und deren Interpretation . . . 66
6.2.1 Dynamische Programmierung . . . 67
6.3 Gitter-Hypothesen . . . 70
6.4 Markov-Zufallsfeld-Modell . . . 71
6.4.1 Grundlagen . . . 71
6.4.2 Markov-Zufallsfeld für die Gittersegmentierung . . . 74
6.4.3 Parameterwahl . . . 77
6.4.4 Energieminimierung . . . 77
6.5 Messpunktdetektion mit Eigenwertverfahren . . . 82
6.5.1 Motivation . . . 82
6.5.2 Eigenwerte und Eigenvektoren, Hauptkomponentenanalyse . . . 82
6.5.3 Eigenspots . . . 84
6.6 Messpunktdetektion mit aktiven Konturen . . . 86
6.6.1 Motivation . . . 86
6.6.2 Grundlagen . . . 86
6.6.3 Semikontinuierliches, skalenunabhängiges Konturmodell . . . 90
6.6.4 Optimierungsverfahren . . . 91
6.6.5 Merkmale . . . 93
6.6.6 Klassifikation mit $E_c$-Schwellwert . . . 96

6.7 Zusammenfassung . . . . . 99

**7 Quantitative Bildauswertung 101**
7.1 Modell der Bildintensität . . . . . 101
7.2 Signalsegmentierung . . . . . 103
7.2.1 Mann-Whitney-Segmentierung . . . . . 103
7.3 Hintergrundschätzung . . . . . 107
7.4 Verhältnisberechnung und Qualitätsmerkmale . . . . . 108

**8 Ergebnisse 109**
8.1 Methoden zur Systemevaluation . . . . . 109
8.1.1 Gittersegmentierung . . . . . 109
8.1.2 Quantitative Auswertung . . . . . 110
8.2 Stichproben . . . . . 113
8.3 Korrektheit der Gittersegmentierung . . . . . 117
8.3.1 Stabilität der Parameter des MZF-Modells . . . . . 120
8.3.2 Restenergieen des MZF-Modells . . . . . 122
8.4 Quantitative Bildauswertung . . . . . 122
8.4.1 Direkter Vergleich mit etablierten Systemen . . . . . 123
8.4.2 Konsistenz replizierter Messdaten . . . . . 127
8.4.3 Qualitativer Vergleich . . . . . 127
8.5 Laufzeitverhalten und Speicherbedarf . . . . . 129
8.5.1 Laufzeit . . . . . 129
8.5.2 Speicherbedarf . . . . . 133

**9 Diskussion 135**
9.1 Automatisierung der Gittersegmentierung . . . . . 135
9.2 Quantitative Bildauswertung . . . . . 136

**10 Zusammenfassung und Ausblicke 139**
10.1 Zusammenfassung . . . . . 139
10.2 Ausblicke . . . . . 141

**A Grundlagen biologischer Methoden 143**
A.1 Technische und biologische DNA-Replikation . . . . . 143
A.2 Reverse Transkription und Herstellung von cDNA . . . . . 144

**B Verteilungsmodelle für Abstände zum $k$-nächsten Nachbarn 145**
B.1 Zufällig verteilte Störregionen . . . . . 145
B.2 Messpunktregionen . . . . . 146

**C Auswertungen der *Medicago*-Nodulationsexperimente 151**

**D Datenschema des AIM-Systems 177**

**E Bildbeispiele 183**

**Literaturverzeichnis 195**

**Index** **202**

# 1 Einführung

Die Biologie unterteilt die Erbinformation der Organismen in einzelne *Gene*, die jeweils eine oder mehrere bestimmte Funktionen einnehmen. Während die Basensequenzen der Gene inzwischen fast routinemäßig „gelesen“ werden können, sind die oft vernetzten Funktionen der Gene viel mühsamer und schwieriger zu verstehen.

Die vorliegende Arbeit beschäftigt sich mit Methoden der Bildverarbeitung zur Auswertung von Mikroarray-Experimenten. Diese Methode erlaubt die Beobachtung der Aktivierung oder *Expression* vieler Gene gleichzeitig, und wird deshalb als wertvolles Hilfsmittel zur Untersuchung von Genfunktionen und deren Wechselwirkungen angesehen.

Mikroarrayexperimente werden oft zentralisiert betrieben, weil sie sich vergleichsweise gut automatisieren lassen, aber auch umfangreiche Infrastruktur erfordern. Dazu zählen technisches Gerät zur Durchführung der Experimente, spezifische Softwaresysteme für die Experimentplanung und -auswertung und deren Integration mit Systemen, in denen die gewonnenen Informationen genutzt, gespeichert und weitergegeben werden können.

Es wird seit längerer Zeit an Expressionsdatenbanken gearbeitet, die analog zu den Sequenzdatenbanken viele kleine Datensätze zusammenführen und sie dadurch optimal nutzbar machen. Die größten Hindernisse dabei sind bisher die mangelnde Standardisierung der Datengewinnung und -aufbereitung.

Die Mikroarray-Methoden liefern als Rohdaten Fluoreszenzaufnahmen, die tausende bis zehntausende von Messpunkten, die sogenannten *Spots* abbilden. Zur Experimentauswertung müssen die Spots zunächst segmentiert werden. In der Literatur wird die Segmentierung üblicherweise in zwei Schritte unterteilt, die als *Addressierung* oder *Gittersegmentierung* und als Spot- oder Signalsegmentierung bezeichnet werden. Die Bildsegmentierung wird in den meisten Fällen noch immer mit Hilfe interaktiver Programme in Handarbeit verrichtet. Dieser Zustand ist zunehmend unbefriedigend: Erstens passt die halbautomatische Segmentierung nicht in das Konzept einer automatisierten Experimentdurchführung und zweitens wirkt sich die globale Segmentierung auf die Signalsegmentierung und damit auf die quantitative Auswertung aus, hat also Einfluss auf Reproduzierbarkeit und Vergleichbarkeit von Messwerten.

Es gibt bisher nur wenige veröffentlichte Ergebnisse über automatische Gittersegmentierung und es sind noch keine Verfahren bekannt, die ohne Kalibrierung oder besondere Annahmen über den Entwurf der Mikroarrays auskommen. Während dies Teilproblem in spezialisierten industriellen Anwendungen gelöst ist, gibt es für die Gittersegmentierung unter weniger kontrollierten Bedingungen, wie sie bei der Integration von Daten aus verschiedenen Quellen herrschen, noch Bedarf für neue Bildverarbeitungsmethoden.

Die automatische Signalsegmentierung ist bereits intensiver bearbeitet worden, weil dieser Teil der Mikroarraybildsegmentierung bei der Arbeit mir interaktiven Systemen

noch wesentlich aufwändiger ist als die Gittersegmentierung. Auch hier gibt es noch Bedarf für robustere und leichter anwendbare Verfahren, die die automatische Signalsegmentierung weniger abhängig von der Gittersegmentierung machen und genauere Segmentierung der oft sehr unregelmäßig geformten Messpunkte ermöglichen.

Die vorliegende Arbeit beschreibt sowohl Methoden zur automatischen Gittersegmentierung als auch zur Signalsegmentierung. Um die Verfahren anwendungsnah evaluieren zu können, ist deren Einbindung in ein Gesamtsystem nötig, das die Handhabung und Speicherung der verschiedenen Daten in strukturierter Form ermöglicht. Der Entwurf für das Mikroarray-Bildverarbeitungssystem stellt die Verbindung zwischen den Einzelaspekten der Gitter- und Signalsegmentierung her.

Zur Systemevaluation ist eine heterogene Stichprobe von Mikroarraybildern nötig, damit die Eignung der Gittersegmentierung für Arrays verschiedener Herkunft untersucht werden kann. Die Evaluation der Signalsegmentierung erfordert idealerweise Kalibrierdaten mit bekanntem Auswertungsergebnis. Solche Daten stehen bisher nicht zur Verfügung, weil die Komplexität der biologischen Systeme, die mit Mikroarrayexperimenten untersucht werden, keine echten Kalibrierexperimente zulässt. Die biologische Relevanz künstlich erzeugter Kalibrierdaten ist fragwürdig.

Die Segmentierung von Mikroarraybildern ist auch für die Bildverarbeitung an sich ein interessantes Problem, weil diese Daten durch ihre von „gewöhnlichen“ Bildern verschiedenen Eigenschaften neue Sichtweisen auf Bildverarbeitungsmethoden motivieren.

Die ersten beiden Kapitel dieser Arbeit beschreiben die biologischen und technischen Grundlagen der Mikroarray-Methoden, aus denen sich Anforderungen an ein automatisches Segmentierungssystem ergeben. Es folgt ein Überblick über die in der Literatur beschriebenen Ergebnisse und Systeme, an die sich ein Kapitel zur Struktur des in dieser Arbeit entwickelten Systems anschließt. Die folgenden Abschnitte gehen auf die Details der verwendeten Bildverarbeitungsmethoden ein. An die Beschreibung der Bildverarbeitungsmethoden schließt sich das Kapitel über die Systemevaluation an. Die Arbeit endet mit der Diskussion der erreichten Ergebnisse und mit Ausblicken auf weitere Forschung.

# 2 Biologische Grundlagen

Der folgende Abschnitt fasst zum Verständnis der Mikroarray-Methoden relevante Grundlagen der Biologie zusammen. Die Ausführungen lehnen sich an das Kapitel KK des Lehrbuches „Molekulare Genetik“ von Rolf Knippers [66] sowie an das vierte Kapitel aus „Molecular Cell Biology“ von Harvey Lodish und anderen [34] an. Die thematische Auswahl und Gewichtung ist im Hinblick auf die später in Abschnitt 3 dargestellten typischen Anwendungen von Mikroarray-Methoden getroffen worden. Umfassende Darstellungen sind in den aufgeführten und weiteren Lehrbüchern, z.B. dem von Lewin [73] zu finden.

## 2.1 Zellen

Alle lebenden Wesen bestehen aus Zellen. Einfache Organismen bestehen aus nur einer Zelle, während Menschen und Säugetiere aus Billionen von Zellen aufgebaut sind. Zellen besitzen eine Hülle in der ein komplexes biochemisches System arbeitet, das sich durch die Kontrolle des Stoffwechsels, also des Stoffaustausches mit der Umwelt und des Stoffumsatzes im Inneren, selbst aufrechterhält.

Lebewesen werden nach dem inneren Aufbau ihrer Zellen eingeteilt. Die einfachste und evolutionshistorisch vermutlich älteste Klasse bilden die *Prokaryoten*, zu denen Bakterien, Archaebakterien und Blaualgen gehören. Prokaryoten besitzen im Gegensatz zu den komplizierteren *Eukaryoten* als wichtigstes Merkmal keinen abgeschlossenen Zellkern. Beispiele für einfache Eukaryoten sind Hefen und Amöben. Abb. 2.1 zeigt modellhaft die Bestandteile einer eukaryotischen Zelle. Die verschiedenen dargestellten Organellen sind von Membranen umschlossene Bereiche, in denen einzelne Funktionen der Zelle lokalisiert sind, z.B. die Erbgutspeicherung im Zellkern oder die Bereitstellung von Energie durch die Mitochondrien.

In höher entwickelten, vielzelligen Organismen gibt es verschieden spezialisierte (oder differenzierte) Zellen, wie z.B. Nerven-, Muskel-, Leber- und Blutzellen, die jeweils eine bestimmte Funktion und dementsprechende Eigenschaften besitzen. Innerhalb der erwähnten Zelltypen lassen sich noch feinere Unterscheidungen treffen, so dass die Anzahl verschiedener Zelltypen beim Menschen mit ca. 320 angegeben wird.

Trotz aller biologischen Vielfalt gibt es eine kleine Anzahl von Stoffklassen, die die biochemischen Abläufe wesentlich bestimmen. Sie werden in den nächsten Abschnitten charakterisiert.

## 2.2 Proteine

Proteine sind eine der wichtigsten in Lebewesen vorkommenden Stoffklassen, denn sie sind als Katalysatoren und Regulatoren an fast allen biochemischen Abläufen der

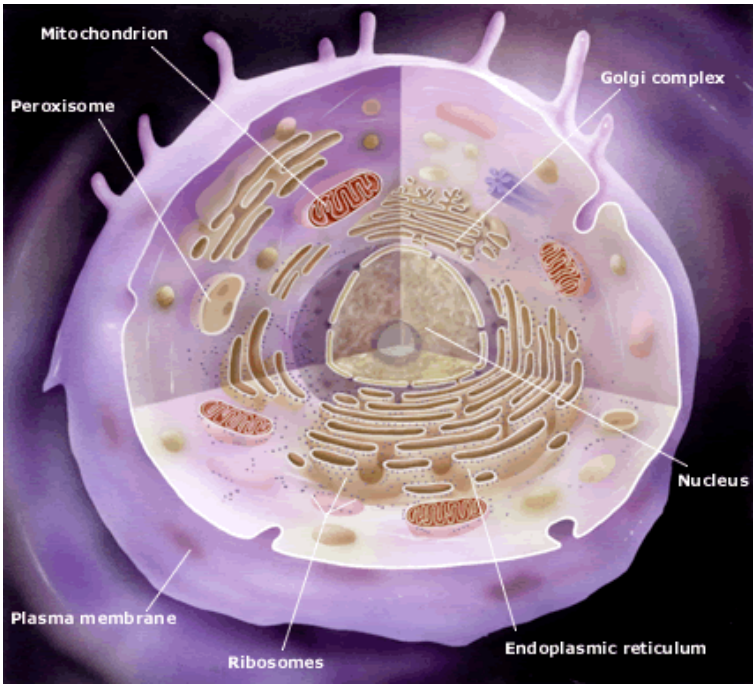



Abbildung 2.1: Aufbau einer Zelle eines höheren (eukaryotischen) Lebewesens (©Die Nobel Stiftung)

Zellen beteiligt. Man teilt Proteine nach ihrer Funktion in die folgenden Klassen ein:

1. Enzyme (Proteine mit Katalysatorfunktionen)
2. Strukturproteine (Gerüstfunktionen)
3. Regulatoren (DNA-bindende Proteine)
4. Transmembran- und Kanalproteine (Stofftransport durch Membranen)
5. Transportproteine (Stofftransport innerhalb von Membranen)

Alle Proteine sind Kettenmoleküle aus etwa zwanzig bis zu mehreren tausend Aminosäuren. Dies sind Moleküle mit einem Amino- und einem Carboxy-Ende, die bei der Proteinsynthese die sogenannte Peptidbindung (siehe Abb. 2.2) eingehen. Aminosäuren haben eine Seitenkette, die für ihre biochemische Funktion wesentlich ist.

In Proteinen treten 20 Aminosäuren mit verschiedenen Seitenketten auf, die sehr verschiedene Eigenschaften in Bezug auf ihre Ladungsverteilung, Hydrophilität, Lipophilität etc. besitzen. Die langen Kettenmoleküle sind keineswegs starr, denn in den Aminosäuren gibt es an dem zentralen C-Atom, dem $C_\alpha$, eine Rotationssymmetrie um die Achsen der Bindungen zu den Carboxy- und Aminogruppen. Die Aminosäureketten „falten“ sich daher. Die Seitenketten beeinflussen die Ausprägung der Rotationsfreiheitsgrade und bestimmen mit ihren Eigenschaften wesentlich die Wechselwirkung zwischen Teilen der gesamten Kette. Der Vorgang der Proteinfaltung ist sehr kompliziert und entzieht sich noch einer genauen Vorhersage.

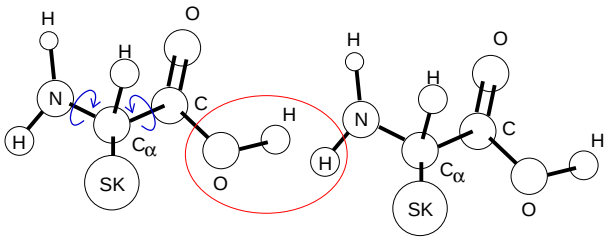



Abbildung 2.2: Die Struktur des Aminosäure-Grundgerüstes mit der Seitenkette (SK), nach der die 20 Aminosäuren unterschieden werden. Aminosäuren werden bei der Proteinsynthese unter Wasserabspaltung (rote Ellipse) durch sog. Peptidbindungen verkettet. Die Bindungen am $C_\alpha$-Atom haben Rotationsfreiheitsgrade (blau).

Die Katalysatorfunktion der Enzyme ist in bestimmten Bereichen auf der Molekülo-berfläche, den aktiven Zentren, lokalisiert. Durch Temperaturerhöhung oder die Verwendung von Lösungsmitteln lässt sich die Faltung aufheben; die Enzyme verlieren dann ihre Katalysatoreigenschaften. Man hat festgestellt, das bei vorsichtiger Wiederherstellung der Normalbedingungen die ursprünglichen Eigenschaften zurückkehren. Dies interpretiert man so, dass die energiegünstigste Faltung der Ketten nur von der Aminosäuresequenz vorherbestimmt ist. Wenn die Umweltbedingungen langsam verändert werden, bringt die Thermodynamik das Molekül in die energetisch optimale Faltung, während es bei rascher Abkühlung in einem anderen Zustand verbleibt. Die Aminosäuresequenz ist also die wesentliche Information über ein Protein. Abgesehen von Veränderungen, die die Faltungseigenschaften und die aktiven Zentren unbeeinflusst lassen, muss ein Organismus deshalb seine Aminosäuresequenzen erhalten und weitervererben.

## 2.3 Die Erbinformationsträger DNA und RNA

Zwischen 1940 und 1950 wurden mehrere grundlegende Experimente durchgeführt, die die Desoxyribonucleinsäure (Desoxyribonuleic Acid, DNA) als Träger der Erbinformation identifizierten. Das erste Experiment von *O.T. Avery et. al.* [7] wird „Transformation von Bakterien“ genannt. Zu diesem Experiment benötigt man einen Bakterienstamm, der eine Mutation, d.h. eine dauerhafte, erbliche Veränderung aufweist. Der von Avery benutzte Stamm konnte aufgrund einer Mutation eine bestimmte Aminosäure nicht selbst herstellen. Solche Bakterien können nur auf Nährböden wachsen, denen diese Aminosäure zugesetzt wurde, wogegen die nicht mutierte Form auch auf aminosäurefreien Nährböden gedeiht. Das Transformationsexperiment besteht darin, DNA aus nicht mutierten Bakterien zu extrahieren und zusammen mit den mutierten Bakterien auf einen aminosäurefreien Nährboden zu geben. Man beobachtet, dass diese Bakterien sich dann auch ohne die Zugabe der Aminosäure teilen. Offenbar haben die Bakterien durch die Wildform-DNA die Fähigkeit zur Synthese der Aminosäure wiedergewonnen. Dieses und andere Experimente hat man mit vielen verschiedenen Bakterienarten wiederholt und immer war das Ergebnis, dass mit der DNA Erbinformation transportiert wird. Schließlich haben 1953 Watson und Crick die Struktur

der DNA aufgeklärt. Im Zellkern kommt die DNA in Form der von ihnen entdeckten $\alpha$-Doppelhelix vor. Sie besteht aus zwei langen Kettenmolekülen, die sich spiralförmig ineinanderlegen (siehe Abb. 2.3). Die beiden einzelnen Ketten haben das gleiche Grundgerüst aus abwechselnd einer Phosphatgruppe und einem Zucker, der Desoxyribose (siehe Abb. 2.4). An dem Ring der Zuckermoleküle ist jeweils eine der Basen Adenin (A), Thymin (T), Guanin (G) oder Cytosin (C) gebunden. Einen Zucker mit Phosphat und Base bezeichnet man als Nucleotid. Die Doppelhelix-Struktur wird durch Wasserstoff-Brückenbindungen zwischen den nach innen zeigenden Basen aufrechterhalten (siehe Abb. 2.5), wobei die Paarungen Adenin - Thymin und Guanin - Cytosin der energiegünstigste Regelfall sind. Je weniger paarende Basen vorhanden sind, um so instabiler ist die Doppelhelix-Struktur.

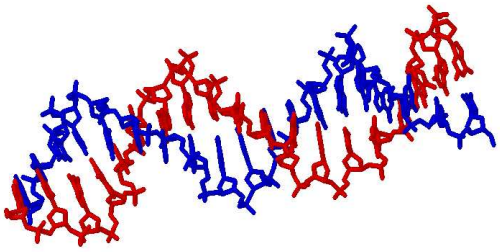


Abbildung 2.3: Die Doppelhelix-Struktur der DNA, innenliegend sind die Basen zu sehen (Quelle: PDB [10])

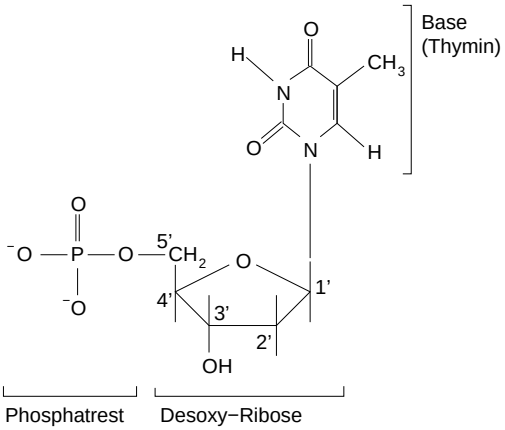



Abbildung 2.4: Das Nucleotid Desoxythymidinmonophosphat (dTMP)

Die Kohlenstoffatome der Desoxyribose werden, wie in der Abbildung gezeigt, von 1’ bis 5’ numeriert. Das Phosphat ist am 5’-Kohlenstoff gebunden und die Base be-

findet sich jeweils am 1'-Kohlenstoff. Bei der Kettenbildung (siehe Abbildung 2.6) bindet immer ein 3' Kohlenstoff an das Phosphat des nächsten Nukleotids. Die Enden von DNA-Einzelsträngen nennt man deswegen 3'- und 5'-Ende. In der $\alpha$-Doppelhelix laufen die beiden Stränge aufgrund der Eigenschaften des Zucker-Phosphat-Gerüstes in entgegengesetzter Richtung, an den Enden treffen also jeweils ein 5' und ein 3'-Einzelstrangende zusammen.

Die Basensequenzen der Einzelstränge sind wegen der Bindungseigenschaften der Basen untereinander und der Gegenläufigkeit *revers-komplementär*.

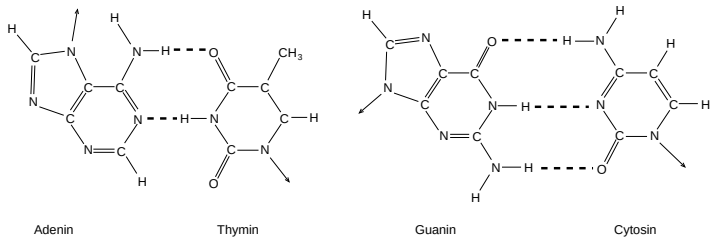



Abbildung 2.5: Struktur und Paarung der Basen

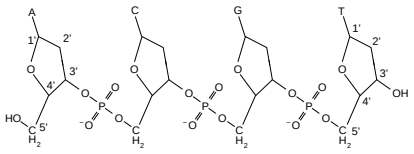



Abbildung 2.6: Eine kurze DNA mit der Basensequenz ACGT

Die beiden Stränge der Doppelhelix lassen sich durch äußere Energiezufuhr in Form von Temperatur- oder pH-Wert-Erhöhung trennen. Diesen Vorgang nennt man Denaturierung oder „Schmelzen". Die Schmelztemperatur bzw. die erforderliche Dissoziationsenergie einer DNA hängt von der Länge der Nukleotidkette und vom Verhältnis des Anteils von GC-Paaren zum Anteil von AT-Paaren ab, denn die GC-Paare mit drei Wasserstoffbrücken enthalten mehr Energie als AT-Paare mit nur zwei Bindungen. Kommen ungepaarte Basen vor, sinkt dadurch natürlich ebenso die Schmelztemperatur. Eine DNA mit vier Basenpaaren hat eine ausreichend hohe Dissoziationsenergie um bei Raumtemperatur eine stabile $\alpha$-Helix bilden zu können.

Die Denaturierung ist grundsätzlich umkehrbar. Bei richtiger Wahl von Temperatur und Salzgehalt des Lösungsmittels finden die komplementären Einzelstränge wieder zusammen (Reassoziation).

Der Rekombinationsvorgang wird auch als Hybridisierung bezeichnet, wenn die beiden Einzelstränge verschiedener Herkunft sind. Hybridisierungsexperimente ermöglichen einen (groben) Vergleich der Sequenzen zweier Organismen, ohne die Sequenz selbst zu kennen.

Die unerwünschte Hybridisierung von nur teilweise komplementären Nukleinsäuresequenzen nennt man Kreuzhybridisierung [1].

[1] *Cross-hybridization. The hydrogen bonding of a single- stranded DNA sequence that is partially but*

| Art | Anzahl Basenpaare | Anzahl Chromosomen |
|---|---|---|
| Fadenwurm *Caenorhabditis elegans* | $80 * 10^6$ | 4 |
| Ackerschmalwand *Arabidopsis thaliana* | $125 * 10^6$ | 5 |
| Hefe *Saccharomyces cerevisiae* | $1340 * 10^6$ | 16 |
| Krallenfrosch *Xenopus laevis* | $3000 * 10^6$ | 18 |
| Mensch *Homo sapiens* | $3000 * 10^6$ | 23 |
| Mais *Zea mays* | $5000 * 10^6$ | 10 |

Tabelle 2.1: Der Umfang der Basensequenzen einiger Organismen. Jedes Chromosom enthält eine lange Doppelhelix

### 2.3.1 Restriktionsenzyme

Für zahlreiche Untersuchungen an DNA ist es zweckmäßig und nötig, die in der Natur vorkommenden sehr langen DNA-Moleküle (siehe Tabelle 2.1) zu zerteilen. Dazu werden eine Reihe von Enzymen, die Restriktionsendonucleasen, benutzt. Sie kommen in jeder Zelle natürlich vor und dienen dort zum Abbau artfremder, in die Zelle eingedrungener DNA. Restriktionsendonucleasen erkennen eine spezifische DNA-Teilsequenz und schneiden dort das Molekül. Jeder Organismus hat seine eigenen, charakteristischen Restriktionsendonukleasen. Falls deren Erkennungssequenz in der eigenen DNA vorkommt, ist sie dort durch eine zusätzliche Methylgruppe an einer Adenin- oder Cytosinbase geschützt. Man spricht auch von Modifikation in Form von methylierten Basen. Zum systematischen Zerteilen der chromosomalen DNA kombiniert man verschiedene Restriktionsendonukleasen. Auf diese Weise legt man Bibliotheken von definierten DNA-Stücken an, die in Verbindung mit Hybridisierungsansätzen äußerst nützlich sind. Eine andere Klasse von Enzymen, die Exonukleasen, bauen DNA von den Enden her ab. Auch sie sind wichtige Hilfsmittel der Molekulargenetik.

#### Ein Nachweisverfahren: Der Southern-Blot

Ein klassisches Nachweisverfahren mit dem Hybridisierungsansatz ist der *Southern-Blot* [96]: Durch Gelelektrophorese werden DNA-Fragmente nach Länge und damit indirekt auch nach ihrer Sequenz aufgetrennt. Ein Abdruck („Blot“) des Gels, mit dem die DNA auf eine Membran übertragen wird, dient anschließend als Hybridisierungssubstrat für eine Probe von (z.B. radioaktiv) markierten DNA-Einzelsträngen. Wegen der Auftrennung der Sequenzen im Gel wird die DNA der Probe nur an bestimmten Stellen des Substrats geeignete Komplementärstränge finden und dort hybridisieren. Die gebildeten Doppelstränge lassen sich durch ihre Radioaktivität mit einem aufgelegten Röntgenfilm detektieren. Bei Verwendung geeigneter DNA-Fragmente können die in der Probe vorkommenden DNA-Sequenzen getrennt nachgewiesen werden.

*not entirely complementary to a singlestranded substrate. Often, this involves hybridizing a DNA probe for a specific DNA sequence to the homologous sequences of different species.* Aus: Susan Allender-Hagedorn und Charles Hagedorn, An Agricultural And Environmental Biotechnology Annotated Dictionary, http://filebox.vt.edu/cals/cses/chagedor/glossary.html

### 2.3.2 Die Ribonukleinsäure

In der Zelle kommt neben DNA noch eine weitere Nukleinsäure vor, die Ribonukleinsäure (Ribonucleic Acid, RNA). Das nächste Kapitel beschäftigt sich mit ihrer Bedeutung. RNA unterscheidet sich von DNA dadurch, dass das Kettengerüst nicht mit Desoxyribose, sondern mit Ribose aufgebaut und die Base Thymin (T) durch Uracil (U) ersetzt ist. RNA hat wegen des unterschiedlichen Gerüstes andere Flexibilitätseigenschaften als DNA und kommt in der Natur in der Regel einzelsträngig vor. RNA-Moleküle falten sich in sich selbst zu Schleifen, wenn entsprechend komplementäre Teilsequenzen vorhanden sind. Solche *Sekundärstrukturen* haben oftmals eine biologische Funktion als Bindestelle für Proteine.

RNA kann auch mit komplementären DNA-Einzelsträngen hybridisieren, was im *Northern Blot*-Verfahren zu Nachweiszwecken ausgenutzt wird. Man unterscheidet drei in der Zelle vorkommende Typen von RNA:

- mRNA
  Die DNA des Zellkerns wird nicht direkt zur Proteinsynthese herangezogen, sondern es werden Kopien der Gene in Form von RNA-Molekülen angefertigt. Diese RNA heißt Boten-RNA (Messenger-RNA) oder kurz mRNA.

- tRNA
  Die Transfer-RNA oder tRNA spielt eine zentrale Rolle bei der Übersetzung von Nukleinsäuresequenzen in Aminosäuresequenzen.

- rRNA
  Die Partikel an denen die Proteinsynthese abläuft, die Ribosomen, bestehen zum Teil aus RNA, der ribosomalen RNA (rRNA).

## 2.4 Genexpression

Unter dem Begriff Genexpression werden die Prozesse zusammengefasst, die ausgehend von der Basensequenz der DNA zur Aminosäuresequenz der Proteine führen. Die Messung der Genexpression oder *Expressionsanalyse* ist die häufigste Anwendung der Mikroarray-Technologie.

### 2.4.1 Gene und ihre Struktur

Die DNA liegt in Zellen gewöhnlich als Doppelstrang vor. Prokaryoten besitzen im häufigsten Fall einen Doppelstrang aus zwei ringförmigen DNA-Molekülen, der den größten Teil der DNA-Menge in der Zelle ausmacht. Neben diesem *Chromosom* gibt es die *Plasmide*, die nicht unbedingt fester Bestandteil des Erbgutes sind.

Protein- oder RNA-kodierende Sequenzen können grundsätzlich auf beiden Strängen der DNA liegen. Lange Zeit war die folgende Definition des Begriffs 'Gen' gültig:

> Als *Gen* bezeichnen wir einen DNA-Abschnitt, der die Information zur Herstellung eines Proteins trägt. Die Gesamtzahl der Gene eines Organismus nennen wir *Genom*.[66]

In prokaryotischen Genomen liegen die Gene nahezu lückenlos dicht und sind vielfach funktionsbezogen in sogenannten *Operons* gruppiert.

Im Gegensatz dazu sind die Chromosomen der eukaryotischen Organismen von der Kernmembran umschlossen und enthalten lineare DNA-Moleküle. Besonders bei höheren Tieren und Pflanzen sind umfangreiche nichtkodierende Bereiche zwischen den Genen vorhanden. Die Sequenzen in der Nähe der Gene haben zum Teil regulatorische Funktionen, wie die im nächsten Abschnitt beschriebenen Promotorsequenzen. Die eukaryotische DNA enthält auch lange Wiederholungssequenzen, die bei der DNA-Replikation während der Zellteilung eine Rolle spielen. Vielfach ist die Funktion der nichtkodierenden Bereiche unklar.

Durch (indirekten) Sequenzvergleich der Boten-RNA mit der genomischen DNA-Sequenz hat man festgestellt, dass die proteinkodierenden Sequenzen auf eukaryotischen Chromosomen häufig durch nichtkodierende Bereiche unterbrochen sind, die in der im Zellplasma beobachteten RNA nicht mehr vorkommen. Man bezeichnet die kodierenden Sequenzbereiche als *Exons* und die nichtkodierenden als *Introns*.

Der als Genexpression bezeichnete Weg von der DNA-Sequenz zum Protein gliedert sich grob in folgende Teile: die als Transkription oder RNA-Synthese bezeichnete Übersetzung der DNA-Sequenz in die komplementäre RNA-Sequenz, das anschließende Entfernen der Introns (*Spleißen*), die Übersetzung der RNA-Sequenz in eine Aminosäuresequenz und deren anschließende Verknüpfung zum Protein (*Translation*) sowie mögliche chemische Veränderungen am fertig verketteten Protein (*posttranslationale Modifikation*).

### 2.4.2 Transkription

Die Transkription ist der allgemeine Prozess zur Herstellung von RNA. Die daran wesentlich beteiligten Enzyme sind die RNA-Polymerasen. Unter geeigneten Bedingungen (Temperatur, Ionenkonzentration, ausreichend Nukleotide vorhanden) funktioniert die Transkription auch im Reagenzglas. Die RNA-Polymerasen verschiedener Organismen weisen einen komplizierten Aufbau aus etwa einem Dutzend (Prokaryoten) bis über 30 (Eukaryoten) Untereinheiten auf. Im folgenden wird die prokaryotische Transkription in vereinfachter Form dargestellt, an der wesentlich das aus drei Untereinheiten bestehende Minimal- oder Core-Enzym und die $\sigma$-Untereinheit beteiligt sind. Zunächst muss das Enzym an die Stelle des Genanfangs auf der DNA gebracht werden. Dabei sind die sogenannten Promotorsequenzen auf der DNA entscheidend. Diese sind nicht einheitlich; es gibt aber z.B. bei dem gut untersuchten Bakterium *Escherichia coli* einen Musterpromotor, dem alle Promotorsequenzen mehr oder weniger ähnlich sind. Er besteht aus einer Basenfolge 5'-TATAAT-3', der Pribnow- oder TATA-Box, zehn Basen vor dem Gen und einer AT-reichen Region 35 Nukleotide vor der kodierenden Sequenz, die idealerweise 5'-TTGACA-3' lautet. Diese Sequenz ist eine Konsensus-Sequenz, weil sie mit einem Alignment-Verfahren als Kompromiss der tatsächlich auftretenden Promotorsequenzen ermittelt wird.

Das Ausmaß der Ähnlichkeit des Promotors zum Musterpromotor beeinflusst die Häufigkeit, mit der dieser Prozess gestartet wird. Gemeinsam mit der Geschwindigkeit des Übergangs von der Initiations- zur Elongationsphase bestimmt dies die Häufigkeit der Transkription des Gens. Die als Regulation bezeichnete Kontrolle über die Transkription erfolgt durch verschiedene, von zelleigenen und äußeren Faktoren abhängige

Mechanismen (siehe Abschnitt 2.5). Die Beobachtung der Transkriptionstätigkeit unter gezielten Einflüssen kann deswegen Aufschluss über die Regulationsmechanismen geben. Man hat festgestellt, dass die mRNA in der Zelle mit großer Frequenz synthetisiert und schnell wieder abgebaut wird[2]. Die Konzentration der mRNA der Gene im Zellplasma ist also dazu geeignet, Einblicke in die Regulationsverhältnisse zu geben, weil sie die Transkriptionstätigkeit auch zeitlich aufgelöst wiedergibt. Für die Zelle ist der hohe Aufwand des dauernden Synthetisierens und Abbauens von mRNA nötig, weil er die lebenswichtige schnelle Reaktion auf Veränderungen der Umweltbedingungen ermöglicht (z.B. die Hitzeschockreaktion).

Die $\sigma$-Untereinheit der RNA-Polymerase reagiert sehr spezifisch auf das Promotor-Motiv. Sie bindet an die Promotorstelle der DNA woraufhin die lokale Entwindung der Doppelhelix einsetzt. Dann startet die eigentliche RNA-Synthese: in dem geöffneten

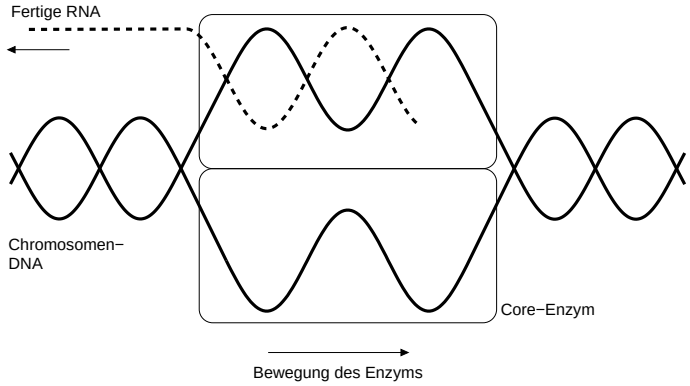



Abbildung 2.7: Schematische Darstellung des Transkriptionsvorgangs

Doppelstrangstück von ca. 12 Basenpaaren werden RNA-Nukleotide komplementär zur DNA aneinandergefügt. In dieser Initiationsphase kann die Synthese noch mehrmals abbrechen und neu beginnen. Danach löst sich die $\sigma$-Untereinheit aus dem Transkriptionskomplex aus DNA, Core-Enzym und RNA-Anfang und die Transkription geht in die Elongationsphase über. Hierbei wandert das Core-Enzym auf der DNA weiter, wobei immer ein Fenster im Doppelstrang geöffnet bleibt, in dem die RNA-Synthese fortgeführt wird (siehe Abbildung 2.7). Dabei kann es wiederum zu Fehlversuchen kommen, bevor sich die RNA von der DNA löst und das Enzym weiterrückt. Der Prozess endet, wenn eine Terminationssequenz erreicht wird, die die Bildung einer haarnadelartig geformten Schleife in der RNA verursacht. Man vermutet, dass diese Schleife das Enzym beeinflusst und so zum Abbrechen der Reaktion führt.

## Reifung der mRNA: Spleißen und Prozessieren

Das primäre Transkriptionsprodukt enthält noch alle Introns und kann im Fall prokaryotischer Transkription mehrere Gene eines Operons umfassen.

[2] Man hat radioaktiv markierte RNA-Bausteine in eine Zellkultur gegeben, kurz danach alle RNA aus den Zellen extrahiert und die mRNA von den anderen RNA-Typen durch Zentrifugation getrennt. Fast ausschliesslich die mRNA enthielt radioaktive Bausteine.

Introns können Längen von etwa 30 bis einige tausend Basen einnehmen, aber es gibt als gemeinsames Merkmal an den übergangsstellen zu den Exons die Sequenzmuster AGGU am 5'-Ende des Introns und CAGG am 3'-Ende, vor dem eine Folge von Pyrimidinbasen und ein Adenosin zu finden sind, die bei der Einleitung des Spleißvorgangs wichtig sind [66]. Das Entfernen des Introns läuft in Form von zwei Transester-Reaktionen ab, bei denen sich zunächst eine Schleife bildet, in der die Intron-Enden verbunden werden. Im zweiten Schritt wird die Schleife abgetrennt und gleichzeitig die Verbindung der angrenzenden Exons hergestellt. Bei manchen Transkripten einfacher Organismen läuft der Spleißprozeß spontan ab und kann im Reagenzglas ohne Zugabe von Proteinen nachvollzogen werden. Die Sekundärstruktur der RNA spielt hier offenbar die entscheidende Rolle. Man nennt diese Form *autokatalytisches Spleißen* oder *Selbstspleißen*.

In den meisten Fällen wird der Spleißprozeß jedoch durch das *Spleißosom* katalysiert. Dieser Komplex aus zahlreichen Proteinen und RNA-Molekülen findet sich nur im Zellkern von Eukaryoten. Er bildet sich mit Hilfe von RNA-Bindeproteinen jeweils an den Spleißstellen neu aus seinen Komponenten, den *small nuclear ribonucleoprotein (snRNP) - Partikeln*. Erst die „gereiften" mRNA-Moleküle verlassen den Zellkern, so dass die dort aufzufindenden Sequenzen nie die Intron-Sequenzen der genomischen DNA enthalten.

Durch den Spleißprozeß werden im Regelfall einfach alle Introns eines Transkripts entfernt. Es können aber auch ein oder mehrere Exons mit herausgeschnitten werden, da das beschriebene Sequenzmuster der Spleißstelle auch an den Enden einer Sequenz aus zwei Introns mit dazwischenliegendem Exon auftritt. Prozesse dieser Art sind in der Literatur unter dem Begriff *alternatives Spleißen* beschrieben. Sie ermöglichen die Kodierung verwandter, aber verschiedener Proteine durch eine gemeinsame Sequenz. Klassische Beispiele sind vier verschiedene Myelin-Proteine auf der Oberfläche von Neuronen, die durch alternatives Spleißen einer Primärsequenz entstehen und zwei verschiedene Spleißvarianten des Calcitonin-Gens, die in verschiedenen Zelltypen auftreten. Die Existenz des alternativen Spleißens motiviert einen neuen Genbegriff für Eukaryoten:

> Ein *Gen* ist eine Transkriptionseinheit, oft eine hintereinanderliegende Reihe von Exons und Introns, die gemeinsam transkribiert werden.

Die Details der Funktion des Spleißapparates sind noch nicht vollständig erforscht. Es gibt neuere Ansätze, mit Hilfe von Mikroarray-Techniken Einblick in die Funktionsweise des Spleißapparates zu gewinnen [32, 50, 110].

Spleißprozesse haben einen wesentlichen Anteil an der Genregulation; man schätzt, dass bei etwa 35% der menschlichen Gene alternatives Spleißen auftritt [50].

### 2.4.3 Translation und Synthese

Der Zellapparat übersetzt mit Hilfe der Ribosomen die Sequenz der mRNA in eine Proteinsequenz. Durch Experimente mit künstlicher mRNA (zunächst eine poly-U-Sequenz, dann kompliziertere aber regelmäßige Basenfolgen) wurde folgendes herausgefunden: In der Struktur der tRNA liegt die Kodierung von Aminosäuren durch Basenfolgen. Es gibt tRNAs für jede der 20 Aminosäuren. Die tRNA-Aminosäurepaare

binden jeweils spezifisch aneinander. Außerdem hybridisieren die tRNAs jeweils mit einem bestimmten Basentriplett, wobei verschiedene tRNAs die gleiche Aminosäure aber verschiedene Tripletts haben können. Wegen ihrer Kodierungseigenschaften heißen die Tripletts auch Codons, die komplementären Dreiersequenzen werden Anticodons genannt. Es gibt vier verschiedene Basen, somit können 64 Codons gebildet werden. Eine Zuordnungstabelle der Codons und Aminosäuren findet man in Lehrbüchern [66]. Nach der Beladung der tRNAs mit Aminosäuren findet die eigentliche Translation am Ribosom statt. Wie bei der Transkription gibt es auch hier einen Initiationsvorgang, der das richtige der drei möglichen Leseraster auf der mRNA findet. Man hat bei zahlreichen Proteinen an dem zuerst synthetisierten Ende die Aminosäure Methionin gefunden, die das Codon AUG hat. Diese Beobachtung führte zu der Erkenntnis, dass dieses Codon die Proteinsynthese einleitet. Tatsächlich beginnen 92% der proteincodierenden Sequenzen bei Bakterien und komplizierteren Organismen mit diesem Codon und 7% der übrigen mit GUG oder UUG. Am Komplex aus Ribosom und mRNA docken nun sukzessive beladene tRNAs mit passenden Anticodons zur gerade abgearbeiteten Stelle der mRNA an. Das Ribosom verknüpft die Aminosäuren und gibt anschließend die tRNAs zur erneuten Beladung frei. Spezielle Mechanismen stellen die richtige Richtung der Verkettung der Aminosäuren (durch vorübergehende Blockade der Aminogruppe des Initiations-Methionins) und die Vermeidung von „Lesefehlern“ sicher. Ein Korrekturmechanismus ist erforderlich, weil die Hybridisierung von nur drei Basen bei Raumtemperatur nicht mehr spezifisch genug ist, um zu verhindern, dass gelegentlich ein mRNA-Codon mit einer tRNA mit falschem Anticodon zusammenkommt. Die Korrektur wird durch vorübergehende Energiezufuhr erreicht, wodurch die schwächeren, fehlerhaften Hybridisierungen mit hoher Wahrscheinlichkeit aufgebrochen werden. Die Proteinsynthese endet mit dem Auftreten eines der drei Stop-Codons UAA, UAG und UGA, die keine Aminosäure kodieren. Der mit dem Start-Codon beginnende und mit einem Stop-Codon endende Sequenzabschnitt wird „Offenes Leseraster“ (open reading frame, ORF) genannt. Nach dem Stop-Codon ist die mRNA üblicherweise nicht zu Ende, die nachfolgende 3’-untranslatierte Region beeinflußt unter anderem die Haltbarkeit in der Zelle und damit die je mRNA-Molekül produzierte Proteinmenge. Die mRNA durchläuft die Translation und Proteinsynthese mehrfach. In Bakterienzellen, die keinen abgeschlossenen Zellkern haben, beginnt die Translation schon an der unfertigen mRNA, so dass die mRNA direkt im Zellplasma transkribiert wird.

## 2.5 Genregulation

Die vorangegangenen Abschnitte haben erläutert, wie die Nukleinsäuresequenzen der Chromosomen in Proteine übersetzt werden. In der Regel wird nur ein Bruchteil des gesamten Geninventars einer Zelle gleichzeitig exprimiert. Es gibt eine Anzahl von Genen, die nur bei der Zellteilung aktiv werden, andere werden in Abhängigkeit von Umweltbedingungen wie Sauerstoffversorgung oder Vorhandensein bestimmter Nährstoffe benötigt. Bei höheren Organismen mit differenzierten Zellen gibt es zelltypspezifische Genexpression, obwohl jede Zelle das gesamte Genom in sich trägt. Die Mechanismen, die die Expression kontrollieren, werden zusammenfassend als *Genregulation* bezeichnet.

Den *reversiblen* Regulationsvorgängen, also z. B. der Stoffwechselumstellung je nach Sauerstoffversorgung oder der Zellzyklusregulation liegt meist ein relativ gut untersuchter Prozess zugrunde, der am Beispiel der Regulation der *lac*-Gene von *E. coli* im folgenden Abschnitt näher erläutert wird.

Dagegen wirken bei den eukaryotischen, vielzelligen Organismen und insbesondere bei deren irreversiblen Differenzierungsvorgängen vielfältige, weniger gut erforschte Mechanismen, auf die hier nicht eingegangen wird.

## Regulation der Transkription: Das Modell von Jacob und Monod

J. Monod und F. Jacob haben 1961 ein Modell der Genregulation beschrieben, das in seinen Grundzügen bis heute verwendet wird [51].

Es gibt u.a. in *E. coli* eine gut untersuchte Gruppe von Genen (Das *lac*-Operon *lacZ, lacY, lacA*), die drei Enzyme zur Verdauung von Milchzucker (Lactose) kodieren. In gewöhnlichen *E. coli*-Zellen sind die Produkte dieser Gene nur in verschwindend geringer Menge vorhanden, solange keine Lactose zur Verfügung steht. Führt man Lactose zu, so werden die drei Gene nach kurzer Zeit stark exprimiert. Diesen Vorgang bezeichnet man als induzierte Expression.

Monod und Mitarbeiter fanden einen *E. coli*-Stamm ($i^-$) mit einer Mutation in dem *lacI*-Gen direkt vor dem *lac*-Operon, der die *lac*-Gene unabhängig von der Lactosekonzentration immer stark exprimiert.

Man brachte in *E. coli*-Zellen mit der ($i^-$)-Mutation ein Plasmid ein, das das *lacI*-Gen und das *lac*-Operon enthielt. In diesen Zellen wurde das *lac*-Operon wie in den Wildtypzellen reguliert. Kontrollversuche bei denen die *lacI*-Mutation auf dem Plasmid liegt bestätigen den Befund.

Aufgrund dieser Beobachtungen stellte man folgendes Modell auf[66]:

1. Das *lacI*-Gen stellt einen Repressor her, der sich im nichtinduzierten Zustand an einen DNA-Abschnitt vor der Gen-Folge *lacZ, lacY, lacA* (die Operatorsequenz) bindet und dadurch deren Transkription verhindert.

2. Induzierende Moleküle heften sich an den Repressor, der dadurch seine Bindungseigenschaften ändert und von der DNA abfällt. Die *lac*-Gene sind dann frei für die Transkription.

Später fand man heraus, dass der Repressor aus vier gleichen Untereinheiten (*lacI*-Produkt) besteht. Die DNA bindet an zwei Seiten des Repressorkomplexes und bildet dabei eine Schleife, die den Promotor des *lac*-Operons verdeckt. Die *lac*-Gene werden gemeinsam transkribiert, und somit wirkt der Repressor auf die ganze Gengruppe.

Neben dem *lacI*-Repressor wirkt noch ein weiteres Protein an der *lac*-Regulation mit, das sogenannte CAP-Protein. Seine Wirkung hängt indirekt vom Vorhandensein von Glukose ab. Der Glukosetransportmechanismus der Zelle erzeugt in seinem Ruhezustand, wenn also keine Glukose im Medium vorhanden ist, eine gewisse Menge zyklisches Adenosinmonophosphat (cAMP). Wenn Glukose transportiert wird, sinkt der cAMP-Spiegel. Zwei CAP-Proteine (cAMP receptor protein) bilden zusammen mit cAMP einen Komplex, der an eine Stelle der DNA nahe beim *lac*-Promotor bindet und dabei die DNA etwas biegt. Durch diese Biegung wird die Affinität der RNA-Polymerase zum *lac*-Promotor erhöht, die *lac*-Expression also verstärkt. Der Sinn die-

ses Mechanismus besteht darin, dass die Zelle Glukose mit geringerem Aufwand verwerten kann als Lactose. Wenn also Glukose vorhanden ist, sollte die Lactoseverdauung gedrosselt werden.

So wie hier dargestellt wirken also die Wechselwirkungen zwischen RNA-Polymerase und dem Promotor, zwischen Repressor und Operator und zwischen dem cAMP/-CAP-Komplex und dem Promotor auf die Stärke der Expression der *lac*-Gene.

Die *lac*-Regulation ist ein spezielles Beispiel, das aber als Modell für die Regulation vieler anderer Gene brauchbar ist. Unterschiede müssen natürlich in der Art der Induktion bestehen.

In der eukaryotischen Transkriptionsregulation spielen weitere Sequenzmotive und DNA-bindende Proteine (die Transkriptionsfaktoren) wichtige Rollen.

Ein anderes, bei Bakterien gefundenes bekanntes Prinzip ist die Genregulation durch Attenuation, die auf Gene bzw. Operons der Aminosäuresynthesewege wirkt. Sie basiert auf Sekundärstruktureffekten der mRNA und der Kopplung von Transkription und Translation, ist also ein rein prokaryotischer Mechanismus.

In Anhang A werden biologische Prozesse zur Vervielfachung von Nukleinsäuren beschrieben, an die sich im Zusammenhang mit der Mikroarraytechnologie genutzte biotechnische Verfahren anlehnen, nämlich die reverse Transkription und die Polymerase-Kettenreaktion (Polymerase Chain Reaction, PCR).

## 2.6 Zusammenfassung

Die vorangegangenen Abschnitte haben einen Überblick über die Grundlagen der Genexpression gegeben. Die Eigenschaften der wichtigsten beteiligten Stoffgruppen, der Proteine und der Nukleinsäuren wurden vorgestellt. In der Transkription wird von der genomischen DNA eine Kopie in Form von mRNA angelegt, die in der Translation in Aminosäuresequenzen übersetzt wird. Die Genexpression wird durch die Genregulation, deren wichtigste Ausprägung durch das Modell von Jacob und Monod beschrieben ist, koordiniert. Auch die beschriebenen Spleißvorgänge an der unreifen mRNA können als Regulationsmechanismen angesehen werden. Die Genregulationsprozesse sind sehr vielfältig und hier bei weitem nicht umfassend dargestellt.

# 3 Parallelisierte Hybridisierungsansätze

Dieser Abschnitt wird verschiedene auf Hybridisierungsansätzen beruhende Methoden beschreiben, die man in der Biologie verwendet, um Einblick in Funktion und Wechselwirkungen von Genen bzw. ihrer Produkte zu erlangen. Die Mikroarray-Hybridisierung, mit der sich der Hauptteil dieser Arbeit beschäftigt, wird ausführlicher dargestellt.

## 3.1 Motivation

Die ständige Verbesserung der Sequenzierverfahren hat dazu geführt, dass sich heute vergleichsweise schnell eine große Menge von Sequenzdaten für einen gegebenen Organismus beschaffen lässt, auch wenn eine vollständige Genomsequenzierung immer noch mit hohen Kosten verbunden ist. Die Methoden der experimentellen Untersuchung der biologischen Funktionen der Sequenzen haben sich noch nicht in gleicher Weise entwickelt. Das lässt sich aus der Tatsache schließen, dass nur ein kleiner Teil der Funktionsbeschreibungen (Annotationen) in den öffentlichen Sequenzdatenbanken [1] auf experimentellen Untersuchungen beruht. In den meisten Fällen existieren nur Funktionshypothesen aufgrund von Homologie (Sequenzähnlichkeit).

Die existierenden experimentellen Ansätze zur systematischen Untersuchung der Funktion vieler Sequenzen oder gar ganzer Genome stützen sich auf Expressionsanalyseverfahren. Die Funktion einer Sequenz wird dabei durch die Bedingungen charakterisiert, unter denen Expression stattfindet oder unterdrückt wird. Dies Prinzip unterliegt einigen Einschränkungen: Erstens kann offensichtlich nicht die Funktion von Genprodukten selbst untersucht werden, sondern nur die Bedingungen der Genexpression. Zweitens können nur wenige Bedingungen im gleichen Experiment variiert werden, wenn man verschiedene Effekte trennen können will. Je allgemeiner ein Experiment angelegt ist desto weniger detaillierte Aussagen werden sich aus den Ergebnissen ableiten lassen. Dennoch werden Expressionsanalysemethoden oft genutzt, denn sie lassen sich als Hybridisierungsansatz gut für viele Gene oder Sequenzen parallelisieren.

Es gibt grundsätzlich die Möglichkeit, Genexpression anhand der Genprodukte, der Proteine, direkt zu messen. In diesen Bereich gehören z.B. die 2D-Gelverfahren zur Trennung von Proteinen nach pH-Wert und isoelektrischem Punkt. Auf diesen Ansatz der Genexpressionsanalyse soll hier nicht weiter eingegangen werden. Ein anderer Ansatz besteht in der Beobachtung der Transkriptionstätigkeit der Zelle durch

[1] Nur 1.1% der Annotationen in der EMBL-Sequenzdatenbank sind mit dem entsprechenden Attribut versehen

Quantifizierung der mRNA, also der Zwischenstufe zwischen genomischer DNA und Aminosäuresequenz (siehe Abschnitt 2.4.2).

Im Folgenden wird nun die Entwicklung der heute zur Verfügung stehenden, auf Hybridisierungsansätzen beruhenden Expressionsanalyseverfahren skizziert.

In dieser Arbeit nicht behandelt werden Revers-Transkriptions-PCR-Verfahren. Diese Familie von Methoden eignet sich für genauere quantitative Messung der Expression eines Gens oder weniger Gene. RT-PCR wird zur Überprüfung der mit den weniger präzisen Arrayverfahren gefundenen Hypothesen eingesetzt.

## 3.2 Vorläufertechnologien der Mikroarray-Hybridisierung

### 3.2.1 Southern-Blot

Einer der ersten parallelen Hybridisierungsansätze war der sogenannte *Southern-Blot* [96]. Man zerteilt dabei zunächst genomische DNA mit Hilfe von Restriktionsenzymen in viele Fragmente unterschiedlicher Länge. Durch Agarose-Gel-Elektrophorese werden die Fragmente nach ihrer Länge aufgetrennt. Wenn die Restriktionsenzyme so gewählt werden, dass keine Fragmente mit gleicher Länge aber unterschiedlicher Sequenz entstehen, wird dadurch indirekt auch eine Aufteilung nach verschiedenen Sequenzen erreicht. Von dem Gel wird anschließend ein Abdruck auf einer Membran angefertigt, auf der sich die nach Länge getrennten Einzelstränge der Fragmente anlagern (siehe Abb. 3.1). Damit sind die Hybridisierungssonden vorbereitet.

Will man das Auftreten bestimmter mRNA-Sequenzen in der Zelle untersuchen, so muss die RNA extrahiert und in einem aufwändigen Verfahren gereinigt werden. Man benutzt dabei mit einem RNA-bindenen Substrat gefüllte Säulen, durch die der Zellextrakt hindurchsickert. Unerwünschte Zellbestandteile können bei niedriger Salzkonzentration aus der Säule ausgewaschen werden. Die RNA lässt sich anschließend bei erhöhter Salzkonzentration aus dem Substrat lösen. Im *Northern Blot*-Verfahren bringt man die so gewonnene RNA direkt auf dem Gelabdruck zur Hybridisierung (es bilden sich dann RNA-DNA-Hybride).

Häufiger wendet man jedoch die *reverse Transkription* (siehe auch Anhang A) an, bei der zur RNA basenkomplementäre DNA-Stränge *(complementary DNA, cDNA)* erzeugt werden. Dazu wird das Enzym 'reverse Transkriptase' benutzt, das ursprünglich bei Viren entdeckt wurde, für die es eine wichtige Rolle bei der Infektion von Wirtszellen spielt. Die reverse Transkription ermöglicht die Markierung der Probensequenzen, indem Nukleotide mit einem eingebauten Fluoreszenzfarbstoff verwendet werden. Die verketteten Nukleotide lassen sich dann mit entsprechenden optischen Verfahren leicht nachweisen.

Die reverse Transkription hat weiterhin den Vorteil, dass jedes RNA-Molekül mehrfach kopiert werden kann und somit ein Verstärkungseffekt auftritt. In vielen Fällen ist die überhaupt zur Verfügung stehende RNA-Menge zu klein, um sie sinnvoll in einem Hybridisierungsexperiment zu nutzen. Eine wirkungsvolle Möglichkeit die Menge zu vergrößern sind PCR-Verfahren mit Zufallsprimern (siehe Anhang A).

Beim Blot-Verfahren wird traditionell die Markierung mit dem radioaktiven Phosphor $^{32}$P oder besser dem weniger „harten“ Strahler $^{33}$P anstelle von Fluoreszenz-

markern benutzt. Die markierte cDNA wird auf den Gelabdruck gebracht, wo sie mit komplementären Sequenzen Doppelstränge bildet. Das Auftreten von Hybridisierungen wird bei einem gelungenen Blot auf eng begrenzte Abschnitte der Gelspuren begrenzt sein. Nach Abwaschen nicht hyridisierten Materials kann die Radioaktivität der cDNA in diesen 'Banden' durch einen Röntgenfilm empfindlich nachgewiesen werden. Je nach Zusammensetzung der Probe ergibt sich ein spezifisches Bandenmuster. In vielen Experimenten wird die Beeinflussung bestimmter Sequenzen durch Vergleiche der Bandenmuster mehrerer Blots abgelesen.

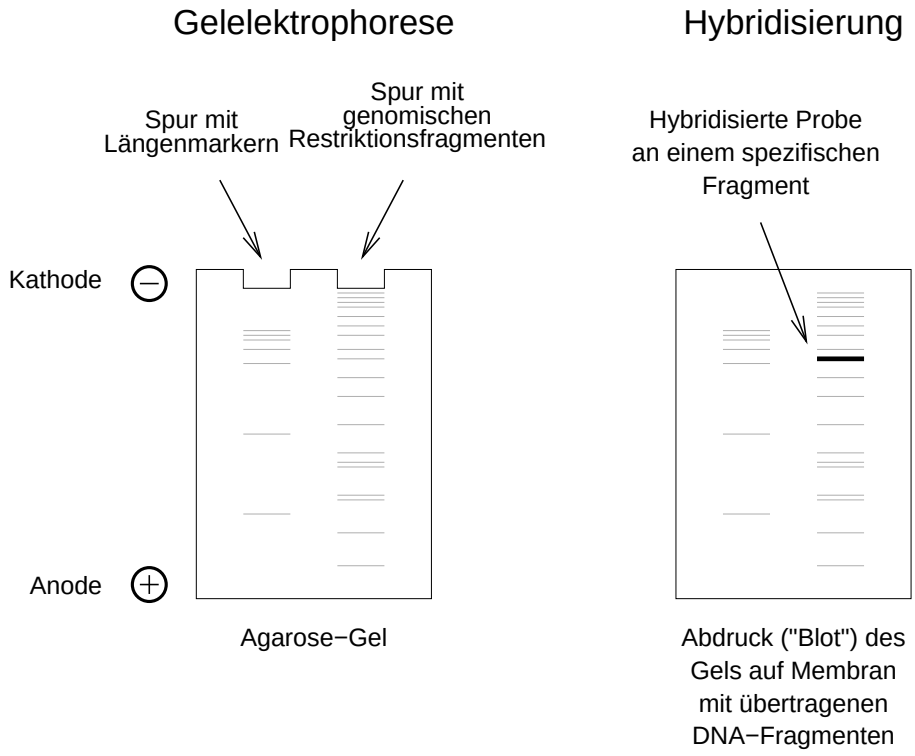



Abbildung 3.1: Zum *Southern-Blot*. Links: Auftrennen von DNA-Fragmenten durch Gelelektrophorese. Das Gel füllt einen schmalen Raum zwischen zwei Glasplatten, in die Vertiefungen am oberen Rand werden die Restriktionsfragmente eingebracht. Abhängig von ihrer Länge wandern die Fragmente im elektrischen Feld unterschiedlich schnell durch die poröse Feinstruktur des Gels. Der Nachweis der Fragmente kann z.B. durch Färbung mit Ethidiumbromid erfolgen. Rechts: Die längensortierten Fragmente werden durch Salzlösungen denaturiert und an eine Membran gekoppelt. Auf diese Membran werden radioaktiv markierte DNA-Proben zur Hybridisierung aufgebracht.

Das Southern-Blot-Verfahren ist relativ unkompliziert anzuwenden, wenn geeignete Restriktionsfragmente hergestellt werden können. Ein Nachteil ist, dass relativ wenig Kontrolle über die Fragmente besteht. Man kann nicht ohne weiteres ein interessantes Fragment isolieren und weiteren Untersuchungen unterziehen. Aus diesem Grund wird häufig mit sogenannten *DNA-Bibliotheken* gearbeitet, mit denen man DNA-Moleküle

isoliert aufbewahren und in größerer Menge replizieren kann. Ein klassisches Verfahren benutzt hierzu gentechnische Methoden (vgl. [34], Abschnitte 7.2 u. 7.3). Restriktionsfragmente oder cDNA-Gemische werden dabei in das Genom von Viren oder in Plasmide einzelliger Organismen wie *E. coli* eingeschleust. Durch starkes Verdünnen der so manipulierten Kultur werden die Träger der einzelnen Fragmente isoliert. Die Kulturen die aus den vereinzelten, modifizierten Organismen wachsen sind genetisch identisch und tragen somit alle die gleiche, eingeschleuste DNA in sich. Man bezeichnet sie auch als *Klone*; die gesamten Klone bilden die Bibliothek. Um in der Bibliothek vorhandene Sequenzen nachzuweisen (*Screenen* der Bibliothek), nutzt man analog zum Southern-Blot den Hybridisierungsansatz.

### 3.2.2 Kolonie- und Membran-Hybridisierung

Eine sehr rudimentäre Form des Screenings benutzt Abdrücke der Kulturplatten der Bibliothek auf Nylon- oder Nitrozellulosemembranen. Auf der Membran werden die zur Klonierung benutzten Zellen oder Viren zerstört, wodurch die DNA frei wird. Die Doppelstränge werden denaturiert (siehe Abschnitt 2.3) und durch Bestrahlung mit UV-Licht oder Hitzeeinwirkung an die Membran gebunden, womit ein Hybridisierungssubstrat für markierte cDNA-Proben hergestellt ist.

Der offensichtliche Vorteil des Verfahrens ist die große Einfachheit. Nachteilig wirkt sich aus, dass neben den ursprünglich zur Herstellung der Bibliothek verwendeten DNA-Sequenzen auch die DNA der Trägerorganismen auf das Hybridisierungssubstrat gelangt. Die Hybridisierung wird dadurch unspezifisch.

In einer verbesserten Variante des Verfahrens findet die DNA-Extraktion vor dem Aufbringen auf das Substrat statt. Die DNA der einzelnen Klone kann gereinigt und mit Hilfe von Stempelwerkzeugen in dichterer Packung auf Membranen aufgebracht werden. Abbildung 3.2 zeigt eine Radiographie einer solchen Membran nach der Hybridisierung. Bei genauerem Hinsehen erkennt man die bei der Membranhybridisierung üblicherweise verwendeten Replikatenmuster. Die Klone sind in Quadraten gedruckt, wobei jeweils an gegenüberliegenden Positionen der gleiche Klon aufgebracht wird. Dadurch wird bei der Auswertung die Erkennung von Fehlmessungen erleichtert. Weitere Verbesserungen können erreicht werden, indem durch PCR-Verfahren aus der Koloniezellen-DNA bessere Hybridisierungssonden isoliert werden. Dadurch sinkt die Gefahr der Kreuzhybridisierung. Die Stempeltechnik kann auch bei der Koloniehybridisierung zur Erreichung höherer Packungsdichten eingesetzt werden.

### 3.2.3 Mikroarray-Hybridisierung

Die bisher behandelten Verfahren erfordern einen relativ geringen Aufwand an Geräten und Material, unterliegen dafür aber verschiedenen Einschränkungen:

- Die Packungsdichte der Messpunkte auf Membranen reicht nicht aus, um die Genexpression eines Organismus global untersuchen zu können.
- Vergleichende Messungen erfordern mehrere Hybridisierungen.
- Automatisierung erscheint schwierig/unangemessen.

Es sind zwei Typen von Mikroarray-Verfahren entwickelt worden, die diese Einschränkungen weitgehend aufheben.

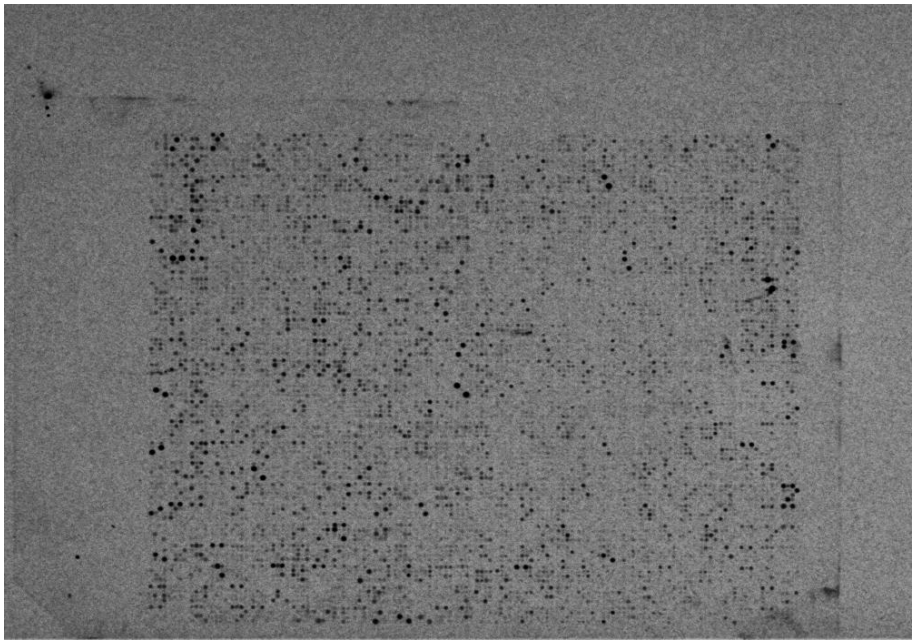


Abbildung 3.2: Eine Radiografie einer Hybridisierungsmembran. Der Kontrast ist gegenüber den Rohdaten verstärkt. (Bild zur Verfügung gestellt von Katja Manthey, Lehrstuhl Genetik, Universität Bielefeld)

### In-Situ synthetisierte Oligonukleotid-Arrays

Der technologisch fortgeschrittenste Typ sind die von Affymetrix entwickelten Oligonukleotid-Arrays [41]. Das Hybridisierungssubstrat wird hier mit synthetisch hergestellten kurzen DNA-Molekülen (20-60 Nukleotide) bestückt. Der Syntheseprozess basiert wie die Herstellung von integrierten Schaltungen auf einem photolithografischen Verfahren: Zunächst werden sog. Linker auf einer Trägeroberfläche aufgebracht, die an einem Ende chemisch an das Substrat binden und am anderen Ende eine zunächst blockierte 3'-Bindestelle für Nukleotide haben. Die Blockierung kann durch Belichtung aufgehoben werden. Man belichtet die mit Linkern versehene Oberfläche durch eine Maske, wodurch in frei definierbaren Zonen die Linker-Enden reaktiv werden. Anschließend wird die Oberfläche mit einem Nukleotid überspült, dessen 3'-Bindestelle wiederum photosensitiv blockiert ist. An den zuvor belichteten Stellen binden die Nukleotide an die Linker-Moleküle und bilden so den Anfang einer Nukleotidkette. Man belichtet sukzessiv mit weiteren Masken für die anderen drei Nukleotide und kann so an allen belichteten Stellen der Oberfläche das gewünschte Nukleotid an die Linker binden lassen. Der Vorgang wird mit weiteren Maskensätzen wiederholt, wodurch sich sukzessive beliebige Sequenzen aufbauen lassen.

Mit dieser Herstellungsmethode lassen sich Hybridisierungssonden für 64000 Oligonukleotidsequenzen auf einer Fläche von $11 \times 11mm$ unterbringen. Die Geometrie ist durch die Belichtungsmasken sehr genau definiert. Die Hybridisierung erfolgt mit fluoreszenzmarkierter cDNA, die mit Scannermikroskopen nachgewiesen werden kann (siehe Abschnitt 4). Radioaktive Marker können nicht gut in der erforderlichen

Auflösung abgebildet werden.

Da sich die Sonden nur bis zu einer Länge von 20-40 Nukleotiden zuverlässig synthetisieren lassen, sind besondere Verfahren beim Experimententwurf und bei der Auswertung nötig: Die Sondensequenzen müssen spezifisch für die Zielsequenzen sein, weshalb ggf. mehrere Sonden für eine längere Zielsequenz benötigt werden. Bei der Quantifizierung der Zielsequenz müssen daher Signale mehrerer Sonden verrechnet werden.

Nachteile dieser Technologie sind der hohe Stückpreis und die mangelnde Flexibilität. Änderungen der synthetisierten Sequenzen erfordern neue Masken und verursachen extrem hohe Kosten. Neuere Entwicklungen, die die starren Masken durch rekonfigurierbare Mikrospiegelanordnungen ersetzen, haben sich bisher noch nicht durchgesetzt [47, 94].

Der genaue Aufbau der Masken wird von den Herstellern nicht veröffentlicht, weshalb man bei der Auswertung auf Dienstleistungen oder mitgelieferte Software angewiesen ist.

## 3.3 Gedruckte DNA-Mikroarrays

Alternativ zur in-Situ-Synthese von DNA-Sonden auf dem Array selbst kann man auch anderweitig hergestellte DNA-Moleküle mechanisch auf Hybridisierungssubstrate aufdrucken [89]. Dabei können synthetische Oligonukleotide oder längere cDNA benutzt werden.

Man kann heute gedruckte Arrays mit einigen zehntausend Punkten herstellen. Die Packungsdichte ist geringer als bei den in-Situ-synthetisierten Arrays, weshalb für ein Hybridisierungsexperiment mit einem gedruckten Array mehr Probenmaterial gebraucht wird. Der wesentliche Vorteil des Verfahrens ist neben den geringeren Stückkosten seine größere Flexibiliät: Die Arrays können relativ unkompliziert mit verschiedenen Sequenzen bedruckt werden, ohne dass neue, aufwendige Masken hergestellt werden müssen.

Die klassische und nach wie vor häufigste Anwendung für gedruckte Mikroarrays ist die differentielle Expressionsanalyse, die am Ende des nächsten Abschnitts genauer beschrieben wird.

### 3.3.1 Herstellung von Mikroarrays

Dieser Abschnitt beschreibt die Herstellung gedruckter DNA-Arrays und die dadurch verursachten Eigenschaften der Bilddaten.

#### Hybridisierungssonden

Die Auswahl der auf das Array aufzubringenden Sondensequenzen ist wesentlicher Teil des Experimentdesigns und stellt ein eigenes Kapitel der Bioinformatik dar. Die Sequenz der Hybridisierungssonden braucht aber (wie auch beim Southern-Blot) nicht unbedingt bekannt sein, um die Mikroarray-Hybridisierung durchführen zu können.

Es kommen in aller Regel PCR-Verfahren zur Anwendung, um ausgehend von einer cDNA-Bibliothek oder synthetisierten Oligonukleotiden Material in ausreichender Menge herzustellen.

#### Substrate

Traditionell werden Mikroarrays auf Glasstreifen im Objektträgerformat gedruckt, denn Glas ist ein günstiges Material, weil es mechanisch stabil, nicht porös, kaum fluoreszierend und gut durchlässig für die bei der Bildaufnahme benutzten Wellenlängen ist. Außerdem kann DNA kovalent an die Oberfläche gebunden werden.

Die Glasoberfläche wird dazu (meist mit Poly-L-Lysin oder Silanen) beschichtet, wodurch neben der Bindefähigkeit auch eine erhöhte Oberflächenspannung erreicht wird, die zu gleichmäßiger geformten DNA-Punkten führt. Fertig beschichtete Gläser mit verschiedenen Beschichtungen und Eigenschaften werden kommerziell angeboten [47]. Die Blockierung der Zwischenräume zwischen den bedruckten Zonen der Oberfläche muss jeweils auf die verwendete Beschichtung abgestimmt sein. Eine von Holloway und anderen beschriebene Versuchsreihe [47] zeigt, dass die Wahl der Beschichtungen sehr großen Einfluss auf die Bildqualität bzw. die zu erwartende Hintergrundintensität hat.

### Drucken von Mikroarrays

Die Roboter, mit denen die PCR-Produkte aus der DNA-Bibliothek auf das Substrat aufgebracht werden, bezeichnet man als *Arrayer*. Das in den ersten Arbeiten beschriebene Prinzip ist im Wesentlichen bis heute unverändert geblieben [22, 39, 47, 89]: Es basiert auf einem karthesischen Roboter, der einen Druckkopf mit einer Anordnung von Spitzen zwischen den PCR-Platten, Glasstreifen und einer Wascheinrichtung transportiert (siehe auch Abb. 3.3).

Die Druckspitzen nehmen kleine Mengen DNA-Lösung aus den (i. a. genormten) PCR-Platten auf und werden auf jedem der Glasstreifen abgesetzt, wobei kleine Tropfen der DNA-Lösungen, die späteren Messpunkte oder *Spots*, zurückbleiben. In der Wascheinrichtung wird verbleibende DNA mit entionisiertem Wasser und Ultraschalleinwirkung aus den Spitzen ausgespült, bevor sie mit neuer DNA für den nächsten Durchlauf bestückt werden. Nach jedem Zyklus rückt die Druckposition auf den Gläsern spalten- und zeilenweise weiter, wodurch jede Druckspitze eine Gitteranordnung (*Grid*, manchmal auch als *Block* bezeichnet) von Messpunkten erzeugt. Üblich sind rechteckige Gitter, aber vereinzelt werden auch hexagonale Gitter verwendet, um eine höhere Packungsdichte zu erzielen.

Zahlreiche Detailverbesserungen haben seit den frühen Arbeiten zu größerer Genauigkeit, Geschwindigkeit und Zuverlässigkeit der Geräte geführt.

Dazu gehören z.B. die Verwendung von kleinen Tropfringen um jede Druckspitze. Die DNA-Lösungen werden nicht mit den Spitzen selbst aus den Platten aufgenommen, sondern mit den Ringen, in denen ein kleiner Tropfen hängen bleibt. Beim Drucken sticht dann die Spitze durch den Tropfen, wodurch die übertragene Flüssigkeitsmenge gleichmäßiger bleiben soll.

Bei anderen Geräten werden anstelle von Federspitzen Düsen in der bekannten Tintenstrahltechnologie eingesetzt [81]. Dadurch erreicht man eine sehr gleichmäßige Punktform, die Technik gilt aber als langsam und damit ungeeignet für große Arrays.

Die technisch ausgefeilteren Apparaturen sind teuer, weshalb immer noch einfache Geräte benutzt werden, wie der von Brown und Eisen [22] beschriebene, aus Standardbauteilen selbst zusammenzusetzende Arrayer.

Es gibt einige typische Eigenschaften der auf einem karthesischen Roboter basierenden Arrayer-Konstruktion, die für die Bildsegmentierung wichtig sind:

1. Die Kanten der Glasstreifen sind nicht immer nach den Koordinatenachsen des Roboters ausgerichtet.

2. Die Abstände der Druckspitzen bzw -düsen sind im Allgemeinen nicht Vielfache der Punktabstände auf dem Array.

3. Die Nadeln in den herkömmlichen Druckköpfen müssen vertikal beweglich sein, damit sie beim Aufsetzen auf das Glas möglichst geringe Kraft erfahren. Dadurch können sie schwingen und sich drehen.

Nach dem Trocknen kann es je nach verwendeter Beschichtung erforderlich sein, die Arrays noch einmal in Wasserdampf zu „rehydrieren“, um eine gleichmäßigere Form und Größe der DNA-Punkte zu erreichen. Anschließend muss die DNA (z.B. durch UV-Bestrahlung) kovalent an die Oberflächenbeschichtung gebunden werden.

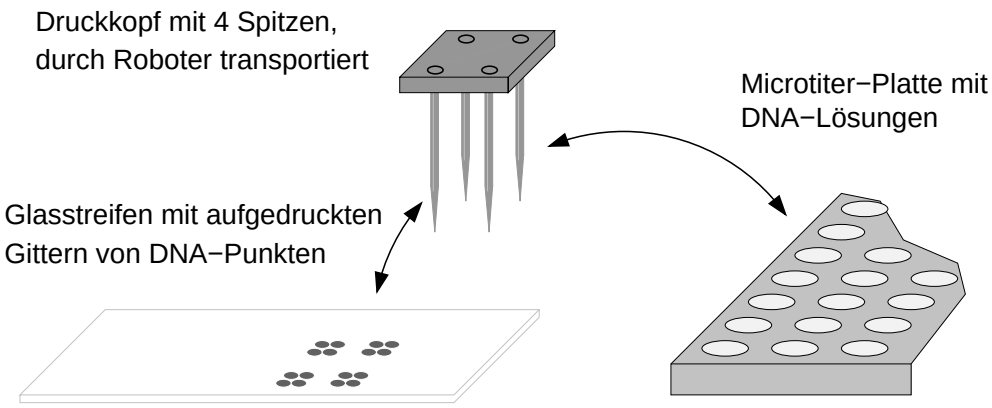



Abbildung 3.3: Prinzip des Druckens von Mikroarrays

Damit später keine weitere DNA an die unbedruckte Oberfläche bindet, muss sie blockiert werden. Die dazu verwendete Methode ist abhängig von der Beschichtung der Glasoberfläche. Durch die Blockierung können erhebliche Unterschiede der Oberflächeneigenschaften zwischen bedruckten und unbedruckten Bereichen auftreten.

### 3.3.2 Hybridisierung

Die Probenvorbereitung (RNA-Extraktion, reverse Transkription und Fluoreszenzmarkierung) in einem Mikroarrayexperiment verläuft analog zu dem in Abschnitt 3.2 skizzierten Southern-Blot-Verfahren. Bei auf Glas gedruckten Mikroarrays werden ausschließlich Fluoreszenzfarbstoffe zur Markierung der revers transkribierten RNA benutzt.

Die eigentliche Hybridisierung läuft in einem dünnen Flüssigkeitsfilm z.B. unter einem Deckgläschen ab. Stringente Hybridisierung wird durch Erhöhung von Salzgehalt und/oder Temperatur der Hybridisierungslösung erreicht. Diese Parameter müssen auf die Schmelztemperatur der Zielsequenzen abgestimmt werden, die durch ihre Länge und GC-Gehalte abzuschätzen ist.

Beim *Kammerverfahren* wird das Array während des mehrere Stunden dauernden Prozesses in einem Behälter mit hoher Luftfeuchtigkeit gelagert, um das Austrocknen von den Rändern des Deckglases her zu verhindern. Nach der Hybridisierung wird das Deckglas durch Abspülen vorsichtig entfernt, um Kratzer zu vermeiden. Reste der Hybridisierungsflüssigkeit und unspezifisch gebundene DNA werden in mehreren Waschgängen mit Salzlösungen entfernt. Für die Stärke der Waschlösungen ist ein Kompromiss zwischen tolerierter Kreuzhybridisierung und Schwächung des gewünschten Hybridisierungssignals zu treffen.

Aufwändigere Hybridisierungsautomaten vermeiden die Verwendung von Deckgläsern.

Nach dem Waschen werden die Arrays in einer Zentrifuge trockengeschleudert und können anschließend gescannt werden. Die Durchführung der Hybridisierung und des Waschens erfordert rasches Arbeiten, denn zu frühes Trocknen der Oberfläche führt zu einem starken Störsignal durch nicht mehr entfernbare Verunreinigungen.

Während der Hybridisierung bilden sich wegen der begrenzten Diffusionsgeschwindigkeit „Konzentrationstrichter“ der jeweiligen Zielsequenzen um die Spots, weshalb ein Mindestabstand insbesondere zwischen replizierten Spots der gleichen Sequenz sinnvoll ist. Die Größe des bedruckten Bereiches bestimmt allerdings auch die nötige RNA-Menge je Hybridisierung, die in manchen Experimenten ein begrenzender Faktor ist. Mögliche Auswege sind dann die dichtere Packung der Messpunkte oder die Vervielfachung der aus der RNA gewonnenen cDNA durch PCR mit Zufallsprimern.

### 3.3.3 Bildaufnahme

Bei der Hybridisierung bilden sich Doppelstränge aus den gedruckten Sonden und den mit Fluoreszenzmarkern versehenen cDNA-Molekülen, die aus der zu untersuchenden RNA hergestellt werden. Zum Nachweis der Hybridisierung auf den einzelnen Messpunkten werden daher Fluoreszenzbilder aufgenommen.

Für die Bildaufnahme werden zwei verschiedene Prinzipien angewandt, nämlich ein Abtastverfahren, das die Intensitäten der Bildpixel sequentiell misst und ein Verfahren, bei dem das ganze Bild simultan durch eine CCD-Kamera aufgenommen wird [19].

Fluoreszenz wird durch die Verteilung der quantenmechanischen Energiezustände der (Valenz-) Elektronen in Molekülen verursacht. Fluoreszierende Stoffe besitzen eine Lücke zwischen zwei dicht besetzten *Banden* im Spektrum. Bestrahlt man ein fluoreszentes Molekül mit Licht einer Energie im Band etwas über der Lücke, so nimmt das Molekül die Lichtenergie mit hoher Wahrscheinlichkeit auf, denn im Band liegen ja viele mögliche Energiezustände dicht beieinander.

Nach der Anregung wird die Energie zunächst in kleinen Schritten wieder abgegeben, bis der untere Rand des Bandes erreicht ist. Von dort kann die Energie nur in einem größeren Schritt, der bis ins nächste Band reicht, abgegeben werden. Wenn die Lücke hinreichend groß ist, reicht die Energie der dabei abgegebenen Strahlungsquanten (die *Fluoreszenzstrahlung*) aus, um sie gut nachweisen zu können. Wegen der Anregung *über* der Bandkante muss die emittierte Strahlung aber immer energieärmer sein als die anregende Strahlung (*Stokes-Verschiebung*).

Die Stokes-Verschiebung macht sehr selektive Fluoreszenzmessungen möglich: Das (optimal) anregende und das vom Fluoreszenzstoff emittierte Licht haben verschiedene Wellenlängen. Man regt die Fluoreszenz mit von Natur aus schmalbandigem Laserlicht an und verwendet schmalbandige Farbfilter für die Emissionswellenlänge, um die Anregungsstrahlung und das Tageslicht aus der Messung der Fluoreszenz herauszuhalten.

#### Konfokale Laserabtastmikroskopie

Die konfokale Laserabtastmikroskopie benutzt das oben beschriebene allgemeine Prinzip der Fluoreszenzmessung in einer Mikroskopoptik. Zusätzlich schränkt eine Lochblende im Okular-Brennpunkt des Strahlengangs (deswegen die Bezeichnung *konfokal*) die räumliche Empfindlichkeit ein. Die Abbildung 3.4 zeigt den Aufbau der Optik. Die Lochblende vor dem Detektor führt dazu, dass alle Strahlung, die nicht aus dem Fokuspunkt auf der Arrayoberfläche kommt, abgeschirmt wird (z.B. Fluoreszenzstrahlung eines Fingerabdrucks auf der Arrayunterseite). Die hohe Ortsauflösung erfordert präziseste Positionierung der Arrayoberfläche. Sie muss sich immer in der nur eini-

ge $\mu m$ tiefen Fokusebene befinden, sonst ergeben sich Schwankungen der Intensität. Deshalb sollten bei der Arrayherstellung besonders plan geschliffene Gläser verwendet werden. Der Träger für das Array muss ebenfalls sehr genau und verschleißarm gearbeitet sein. Zunehmend werden auch Geräte ohne die Lochblendenoptik angeboten, die nicht so hohe Anforderungen an die mechanische Präzision der Arrays stellen.

Zur Aufnahme der gesamten Arrayoberfläche wird der Glasträger zeilenweise durch den Fokuspunkt der beschriebenen Apparatur bewegt. Der Antrieb längs der Zeilen erfolgt direkt und in der anderen Richtung über eine Gewindespindel. Da bei der Abtastung Zeilenvor- und -rücklauf genutzt werden, kann besonders bei hohen Auflösungen ein Versatz von mehreren Pixeln zwischen aufeinanderfolgenden Bildzeilen entstehen.

Um verschiedene Fluoreszenzfarbstoffe messen zu können, müssen Laser und Emissionsfilter gewechselt werden. Dadurch erklärt sich vermutlich eine gelegentlich beobachtete Verschiebung zwischen den einzelnen Bildern, wie sie z.B. in Abb. 3.6 auf Seite 30 zu erkennen ist (rote und grüne Ränder an den gelben Punkten). Die Verschiebung kann auch über das Array variieren. Ein Erklärungsansatz dafür ist die Wärmeausdehnung der Transportspindel im Betrieb, denn theoretisch verlängert eine Temperaturerhöhung um 10K eine Stahlspindel der Länge eines Objektträgers um ca. 10 $\mu$, also eine typische Bildpixelausdehnung[2].

#### CCD-Kamera

Die Bildaufnahme mit Hilfe von *Charge Coupled Device*(CCD)-Kameras unterscheidet sich von der konfokalen Abtastmikroskopie im wesentlichen durch die zeitgleiche Detektion der Fluoreszenzstrahlung auf einer größeren Fläche. Zur Erzeugung der Anregungsstrahlung wird entweder wie bei der Abtastmikroskopie Laserlicht benutzt oder es werden Blitzlampen und Filter eingesetzt [19]. Die Arrayoberfläche wird großflächig mit dem Anregungslicht bestrahlt. Das Fluoreszenzlicht wird auch bei der CCD-Bildaufnahme durch Filter von der Anregungsstrahlung getrennt und von einer Optik auf die CCD-Oberfläche projiziert. Ein CCD besteht aus vielen Ladungsspeichern, die sich beleuchtungsabhängig aufladen. Nach einer gewissen Belichtungszeit wird die Beleuchtungsintensität durch Messung der Ladung in den Zellen bestimmt.

Übliche CCDs sind nicht groß genug, um ein ganzes Mikroarray bei der Standardauflösung von 10 $\mu$/Pixel abzubilden. Deshalb werden meistens mehrere Teilbilder aufgenommen und zusammengesetzt. Weil die Fluoreszenzstrahlung sehr schwach ist, können Belichtungszeiten von bis zu 10 Minuten nötig sein [19]. Es gibt wegen der großflächigen Anregung einen Überstrahlungseffekt, d.h. Streulicht von hellen Bildbereichen ist überall sichtbar. CCD-Mikroarraybilder erscheinen oft viel gleichmäßiger und rauschärmer als sequentiell abgetastete Bilder. Abbildung 3.5 zeigt zum Vergleich die Histogramme von einem CCD-Bild und einem Abtastmikroskop-Bild.

### 3.3.4 Differentielle Expressionsanalyse

Der folgende Abschnitt erläutert die quantitative Bildauswertung für den häufigsten Typ von Mikroarrayexperimenten, die differentielle Genexpressionsanalyse.

[2] In der in Abständen von etwa 8 min. eingescannten *wnt*-Arrayserie der Stichprobe (siehe Kap. 8) sinkt die Verschiebung von der ersten Zeile des ersten Bildes an über die nächsten Bilder erst schnell und dann immer langsamer ab. Vermutlich war das Gerät am Anfang kalt.

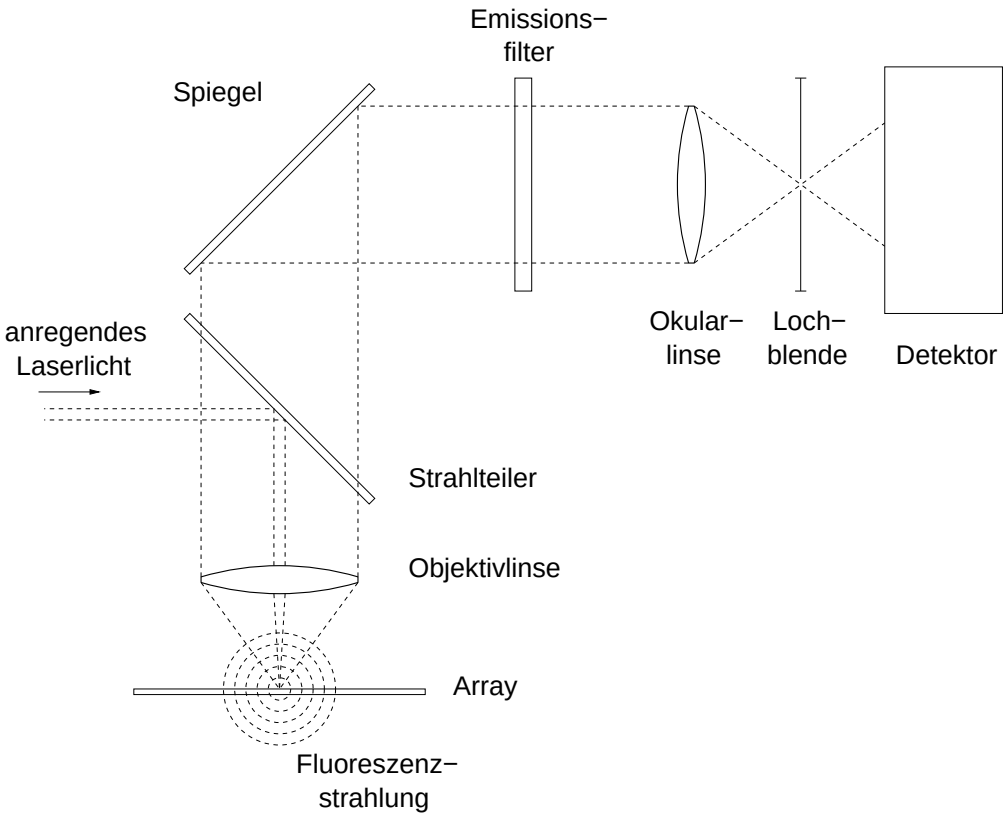



Abbildung 3.4: Die Optik eines Array-Scanners (nach Schena [88])

### Quantifizierung von Expressionsverhältnissen

Ein Großteil der bekannten Arbeiten, in denen gedruckte cDNA-Mikroarrays eingesetzt werden, beschreibt vergleichende Genexpressionsstudien [2, 35, 89, 98, 106], in denen Proben von Zellen immer paarweise verglichen werden. Um die Unterschiede der Genexpression zwischen den beiden Proben zu bestimmen, markiert man deren cDNA mit verschiedenen Fluoreszenzfarbstoffen, die selektiv nachweisbar sind. Die markierten cDNA-Proben werden *gemischt* zur Hybridisierung auf das Array gebracht (für dies Beispiel nehme man an, das Array trage für jedes Gen des untersuchten Organismus einen gedruckten DNA-Punkt). Man nennt dies Verfahren auch konkurrierende Hybridisierung, denn die Sonden-DNA bildet Doppelstränge mit cDNA aus beiden Proben. Nach der Hybridisierung sieht man in überlagerten Fluoreszenzaufnahmen der beiden Farbstoffe die Unterschiede der cDNA-Proben bzw. der Genexpression in den beiden Proben als Farbunterschiede der Messpunkte (siehe Abb. 3.6). Messpunkte, deren zugehörige Sequenzen nicht oder nur in geringer Menge in der RNA-Probe vorhanden waren, sind gar nicht bzw. nur sehr schwach sichtbar.

Solange keine Sättigung der Sonden-DNA auftritt, ergibt sich wegen der Hybridisierungskinetik [33] näherungsweise ein linearer Zusammenhang zwischen der Menge einer bestimmten cDNA-Sequenz in den Proben und der Menge der auf dem dazugehörigen Array-Punkt gebildeten Doppelstränge. Er gilt indirekt auch für die Menge

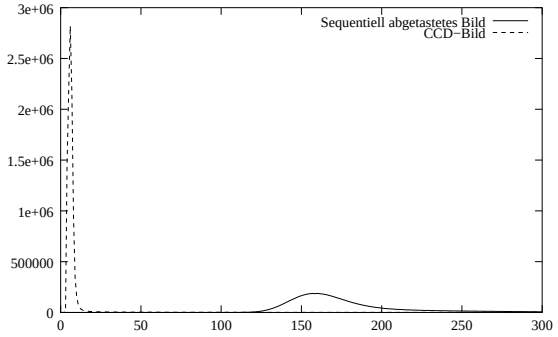



Abbildung 3.5: Ausschnitte von Histogrammen von Bildern der FlyChip-Stichprobe (siehe Seiten 189/190). Die Maxima gehören jeweils zur Hintergrundintensität. Die geringere Streuung des Hintergrundrauschens bei der CCD-Aufnahme und die verschiedene Kalibrierung sind deutlich sichtbar.

des Farbstoffs, der mit der cDNA auf dem Messpunkt gebunden wird.

Weil die Linearität für beide Proben gilt, gibt das Verhältnis der Fluoreszenzintensitäten der beiden Farbstoffe Änderungen der Zusammensetzung der cDNA-Proben wieder und damit indirekt auch das Verhältnis der mRNA-Mengen in den untersuchten Zellen. Zwischen der gesuchten Farbstoff-Fluoreszenzintensität $I^f$ und der Dichte $p$ des gedruckten DNA-Materials an einer Stelle $\vec{x}$ auf dem Substrat besteht also idealisiert der in Gl. (3.1) formulierte lineare Zusammenhang.

$$I(\vec{x}) = cp(\vec{x}) \tag{3.1}$$

In die Konstante $c$ gehen die Menge des durch die Hybridisierung auf dem Messpunkt abgelagerten Farbstoffs und auch die Gesamtmenge der verwendeten RNA, der Anteil farbstofftragender Nukleotide in der cDNA und die Detektionsempfindlichkeit der Bildaufnahme ein.

Die *gemessene* Intensität enthält neben der Farbstofffluoreszenz nach Gl. (3.1) zusätzlich Hintergrundanteile, die von Verunreinigungen und nicht perfekter Nullpunktkalibrierung der Bildaufnahme verursacht werden. Die Beschaffenheit des Hintergrundsignals hängt sehr von der Oberflächenbehandlung und -beschaffenheit des Mikroarrays ab. In den meisten Mikroarraybildern beobachtet man eine konstante Hintergrundkomponente mit überlagerten, scharfen Spitzen (Partikel), deren Dichte über die Gesamtfläche des Arrays variiert. Mögliche Ursachen für das Hintergrundsignal sind die Kalibrierfehler (Offset) des Bildaufnahmegerätes und Fluoreszenzstrahlung von nicht abgewaschener Hybridisierungslösung oder anderen Verunreinigungen der Substratoberfläche. Schwankungen von Konzentration und Schichtdicke der Hybridisierungslösung dürften zur Ortsabhängigkeit des Hintergrundsignals beitragen (siehe auch Anhang E).

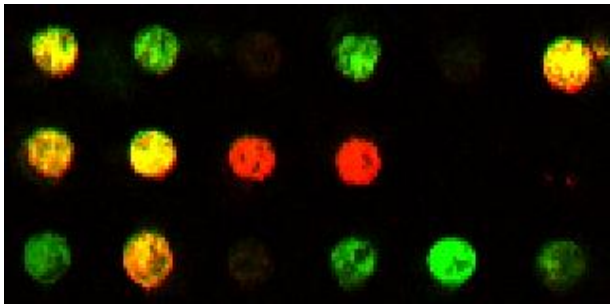


Abbildung 3.6: Ein Ausschnitt aus einem Mikroarraybild. Die Fluoreszenzaufnahmen sind als Rot- und Grünkomponente im RGB-Farbraum dargestellt, die benutzten Farbstoffe sind *Cy3* (grün, 523nm) und *Cy5* (rot, 635nm). Bei starken Veränderungen zwischen Referenz und manipulierter Probe erscheinen die Punkte rein rot oder rein grün, bei geringer Veränderung mischen sich die Farben zu Gelbtönen. Schwach exprimierte Gene liefern nur ein geringes Hybridisierungssignal, so dass die entsprechenden Punkte nur wenig Farbstoff tragen und dunkel erscheinen. Die gelben Punkte sind am unteren Rand leicht rot und oben leicht grün; hierbei handelt es sich um ein Artefakt der Bildaufnahme.

### Normalisierung

Wenn das Verhältnis der Intensitäten beider Kanäle gebildet wird, fällt die $p(\vec{x})$-Abhängigkeit wegen der konkurrierenden Hybridisierung weg. Man braucht daher keine absolute Kalibrierung zwischen Genexpression und Fluoreszenzintensität, jedoch müssen die je nach Kanal verschiedenen systematischen Störfaktoren in den Konstanten $c$ korrigiert werden. Diese Korrekturen werden unter dem Begriff *Normalisierung* zusammengefasst. Man nutzt dazu aus dem jeweiligen Experimententwurf folgende Annahmen über die Verteilung der Expressionsraten oder -verhältnisse, aufgrund derer die Daten von vielen Messpunkten gemeinsam korrigiert werden [11].

Gleichung (3.2) beschreibt den gebräuchlichsten Ansatz zur Verhältnisberechnung für einen Messpunkt:

$$R = \frac{I^f_{\text{Grün}}}{I^f_{\text{Rot}}} \approx \frac{I_{\text{Grün}} - I^{bg}_{\text{Grün}}}{I_{\text{Rot}} - I^{bg}_{\text{Rot}}} \tag{3.2}$$

$I_{\text{Grün}}$ und $I_{\text{Rot}}$ bezeichnen die gemessenen unkorrigierten Intensitäten der Bildkanäle, von denen geschätzte Hintergrundanteile $I^{\text{bg}}$ abgezogen werden. Diese Gleichung gilt punktweise an jeder Stelle des Messpunktes, wenn man die pixelweise Korrespondenz der zwei Bildkanäle voraussetzt. Dann können aus der Verteilung der pixelweise nach Gl. (3.2) berechneten Verhältnisse eine Gesamtschätzung für den Messpunkt und Merkmale für deren Zuverlässigkeit berechnet werden („Pixel-by-Pixel"-Ansatz). Messpunktbezogene Qualitätsmerkale, die sich auf pixelweise Korrespondenz der Kanäle stützen, sind z. B. der lineare Korrelationskoeffizient der Intensitäten der Messpunktpixel [59] oder die sog. „Spot Ratio Variability" von Brown und anderen [21].

Will man die pixelgenaue Korrespondenz zwischen den Kanälen nicht voraussetzen, bildet man das Verhältnis der Gesamtintensitäten in beiden Kanälen.

Die Hintergrundschätzungen werden meistens aus den Intensitäten der Pixel in der direkten Umgebung des Messpunktes berechnet (siehe auch Abschnitt 4.2.3). Das Auftreten von Messpunkten, die dunkler als ihre Umgebung sind (sog. „schwarze Löcher“), lässt diese Praxis fragwürdig erscheinen. Schwarze Löcher werden vermutlich durch die von der Blockierung bewirkten Unterschiede der Oberflächeneigenschaften innerhalb und außerhalb der Messpunkte verursacht.

Zur Bestimmung von Intensitäts- und Hintergrundwerten müssen zunächst die zu jedem Messpunkt gehörenden Signalpixel bestimmt werden. Der Abschnitt 4.2.3 geht näher auf Segmentierungsverfahren zu diesem Zweck ein.

Yang und andere beschreiben verbesserte Ansätze zur Normalisierung, die auch nichtlineare Verzerrungen der Intensität und Abhängigkeiten der Normalisierung von der Position auf dem Array berücksichtigen [108]. Dazu werden Kontroll-Messpunkte von Verdünnungsreihen von DNA-Sequenzen benutzt, die Hybridisierungssignale liefern, deren relative Größe bekannt ist. Aus den Messdaten dieser Kontrollpunkte wird mit parameterfreien Regressionsverfahren eine Normalisierungskennlinie geschätzt, nach der die übrigen Daten korrigiert werden. Bretz und andere [17] bemerken, dass die systematischen Störfaktoren sequenzabhängig sind. Sie untersuchen verschiedene replizierte Hybridisierungsstrategien auf ihre Eignung für eine verbesserte Normalisierung.

Im Kapitel 7 wird die Modellierung der konkurrierenden Hybridisierung noch eingehender behandelt.

### Replikate zur Qualitätssicherung

Die Zuordnung genau einer Sequenz zu einem Gen ist eine idealisierte Vorstellung. Tatsächlich gibt es in Sequenzdatenbanken meist mehrere ähnliche Einträge zu einem Gen (oder verschiedene Klone in DNA-Bibliotheken), die aus verschiedenen Quellen stammen und deshalb nicht genau gleich sind. Für die Messung der Expression müssen daher im Experimententwurf geeignete, spezifische Sonden bestimmt werden. Prinzipiell kann es also verschiedene Sondensequenzen für ein und dasselbe Gen geben. Ebenso werden manchmal die gleichen Sondensequenzen an mehreren Messpunkten auf ein Array aufgedruckt. Derartige replizierte Messpunkte sind zur Evaluation der in dieser Arbeit beschriebenen Bildverarbeitungsmethoden nützlich, weil ihre Konsistenz nur von ortsabhängigen Effekten auf dem Array selbst gestört wird.

Andere Formen von Replikaten sind die technischen Replikate, unter denen man Wiederholungen der Probenvorbereitung (reverse Transkription, Farbmarkierung) mit verschiedenen Arrays aber gleichem RNA-Material versteht und biologische Replikate, bei denen das gesamte Experiment vor der RNA-Gewinnung mit gleichen Bedingungen wiederholt wird. Wenn bei technischen Replikaten die Farbstoffe vertauscht werden, spricht man von „Dye-Swap“-Experimenten. Diese Methode deckt systematische Fehler durch Unterschiede der Detektionsempfindlichkeiten und der Effizienz des Farbstoffeinbaus in die cDNA auf [17, 31, 108]. Die Varianz von technischen und biologischen Replikaten dieser Arten ist wesentlich höher als die von mehrfach gedruckten Messpunkten, weil mehr Verfahrenschritte für die einzelnen Replikate getrennt durchgeführt werden. Biologische Replikate haben die höchste Varianz, weil z. B. die

Individuenstreuungen mit in die Messung eingehen. In der statistischen Auswertung können die unterschiedlichen Replikattypen sehr nützlich sein, um die verschiedenen Fehlerquellen zu analysieren [17]. Die meisten Störfaktoren können nicht ohne weiteres separat kalibriert werden.

### 3.3.5 Typische Anwendungen von Mikroarray-Hybridisierung

Bei komplexeren Fragestellungen kann es durchaus auch mehr als zwei Proben geben, die dann in mehreren Hybridisierungen untersucht werden müssen. Die Entscheidung darüber, welche Proben verglichen werden, wie die Normalisierung erfolgen kann und welche Auswertungsmethode benutzt wird, richtet sich nach der Fragestellung des Experiments. Die Arbeiten von Kerr und Churchill [63], von Yang und Speed [109] und von Bretz und anderen [17] behandeln Entwurfstechniken zur Ermittlung der besten Hybridisierungsstrategie bei gegebener Fragestellung.

Es gibt einige immer wieder benutzte Experimenttypen, die im Folgenden skizziert sind.

- *Screenen*: Die überblicksweise Untersuchung der Genexpression bzw. die Identifikation von Kandidaten für detailliertere Untersuchungen ist die einfachste und weit verbreitete Anwendung der Mikroarrayhybridisierung [3], [49]. Es wird zumeist eine Referenzprobe (Wildtyp, Normalbedingungen, etc.) gegen eine Probe unter experimentellem Einfluss (Mutante, variierte Umweltbedingung) verglichen. Eine typische Methode der Experimentauswertung ist die Identifikation der differentiell exprimierten Gene mit dem t-Test [95].

- *Typisieren von Expressionsmustern*: Alle Zellen der höheren Lebewesen tragen das vollständige Genom in sich, aber nur ein kleiner Teil davon kommt abhängig von der Differentiation zur Expression, wodurch sich charakteristische *Expressionsprofile* ergeben. Tumorgewebe verschiedener Typen hat ebenso charakteristische Expressionsprofile, wodurch sich medizinische Anwendungsmöglichkeiten ergeben haben [2, 90]. Als Referenz dienen Mischungen von RNA-Proben vieler Zelltypen oder RNA von gesundem Gewebe in der Tumorklassifikationsstudie. Eine verbreitete Auswertungsmethode für diesen Experimenttyp ist die Clusteranalyse, wobei die Datenvektoren den verschiedenen Zell- oder Gewebetypen zugeordnet sind. Man kann das Typisieren von Expressionsmustern auch als Verallgemeinerung des Screening-Ansatzes auf mehr als zwei Kategorien (oder Klassen) verstehen.

- *Zeitreihen*: Wenn die Interaktion verschiedener Gene untersucht werden soll, ist die Aufnahme von Zeitreihen der Genexpression sinnvoll. Die Referenz ist dabei eine an einem Startzeitpunkt genommene mRNA-Probe, die gegen mRNA-Proben von späteren Zeitpunkten verglichen wird [35, 98]. Die Datenauswertung erfolgt ebenfalls typischerweise durch Clusteranalyse, wobei die Datenvektoren hier aber Genen zugeordnet sind. Ihre Komponenten beschreiben den Expressionszustand des jeweiligen Gens zu verschiedenen Zeitpunkten.

#### Andere Anwendungen von Mikroarray-Hybridisierung

- *Unterscheiden von Spleißvarianten:*

  Ares und andere beschreiben ein Mikroarray, das Oligonukleotidsonden für die Exon-Exon-Übergangsstellen der reifen mRNAs vom Hefegenen enthält [32]. Mit entsprechenden Kontrollen und Normalisierungen weisen sie damit das Auftreten verschiedener Spleißvarianten vieler Gene gleichzeitig nach.

### 3.3.6 Expressionsdatenbanken

Es wird angestrebt, analog zu den Sequenzdatenbanken Expressionsdatenbanken aufzubauen, die die Ergebnisse vieler Experimente zusammenführen und für umfassende Analysen zugänglich machen [62].

Es gibt mehrere Systeme, mit denen solche Konzepte verfolgt werden:

- Die Stanford Microarray Database [44] ist die älteste und bislang umfangreichste Mikroarray-Datenbank. Sie wird von der Genetikabteilung der Stanford University, USA, betrieben, wo die gedruckten Mikroarrays erfunden worden sind.

- Der Gene Expression Omnibus [37] wird vom National Center for Biotechnology Information (NCBI), USA, betrieben und ist weniger auf Daten von gedruckten Glas-Mikroarrays spezialisiert.

- GeneX [75] wird vom National Center for Genome Resources (NCGR), USA, und IBM entwickelt und verfolgt ebenfalls einen technologieübergreifenden Ansatz.

- Die Expressionsdatenbank ArrayExpress [16] ist ein Projekt des Europäischen Bioinformatik-Institutes (EBI) in Cambridge.

Expressionsdatenbanken enthalten komplexe Datensätze, die neben den Expressionswerten selbst viele Nebeninformationen über das Experiment und verwendete Materialien, Methoden und Geräte umfassen.

Die MGED - Gesellschaft (Microarray Gene Expression Data Society) hat Empfehlungen über die mindestens erforderlichen Informationen über Mikroarrayexperimente als Richtlinie für die Veröffentlichung von Daten herausgegeben (Minimum Information About a Microarray Experiment, MIAME)[15]. Dazu ist das Datenmodell MAGE-OM (Microarray Gene Expression Object Model) für Mikroarray-Expressionsdaten vorgeschlagen worden, dessen Umsetzung in dem XML (eXtensible Markup Language) - Format MAGE-ML (MicroArray Gene Expression Markup Language) die Schnittstelle für den Datenaustausch zwischen Expressionsdatenbanken [101]. MAGE-OM enthält ein kontrolliertes Vokabular, das die universale Annotation von Mikroarray-Expressionsdatensätzen ermöglicht.

Die Standardisierung der Speicherung und des Austausches von Mikroarraydaten ist vergleichsweise weit vorangeschritten. Zur effektiven Nutzung von Expressionsdatenbanken müssen auch Standards geschaffen werden, die die Daten selbst vergleichbar machen. Darunter fallen robuste Methoden zur Extraktion von Intensitätsmesswerten und zur Normalisierung. Der Stand der Forschung in diesem Bereich ist noch weit von abschließenden Ergebnissen entfernt.

Neben den oben aufgelisteten Expressionsdatenbanksystemen gibt es zahlreiche Systeme, die mehr als Werkzeug zum Entwurf von Arrays und zur Bearbeitung und Auswertung von Expressionsmessungen beim Experimentator gedacht sind. Ein Beispiel ist das EMMA-System (EST Meets MicroArray)[8], das für die Evaluation der in dieser Arbeit beschriebenen Methode zur Messpunktsegmentierung benutzt wird. EMMA stellt Benutzerschnittstellen zur Dateneingabe, Normalisierung, Visualisierung und statistischen Auswertung bereit und enthält Datenbankfunktionalitäten zur Speicherung von Experimentdaten und Auswertungen.

## 3.4 Automatische Mikroarray-Bildverarbeitung

### 3.4.1 Problembestimmung und Motivation

Die Auswertung von Mikroarraybildern umfasst als ersten Schritt die *Adressierung* (*Gittersegmentierung*, *Gridding* ) der gedruckten Punkte, womit die eindeutige Zuordnung von logischen Koordinaten der Gitteranordnung (Gitter-Zeile-Spalte) zu Bildausschnitten gemeint ist, die jeweils genau einen Messpunkt umschließen. Der zweite Schritt ist die quantitative Auswertung der Fluoreszenzintensität einzelner Spots bzw. der jeweiligen Bildausschnitte. Dazu gehört die pixelgenaue Segmentierung signaltragender und nicht-signaltragender Bildbereiche, die Schätzung der Hintergrundintensität und die Berechnung von Intensitätsverhältnissen.

Die Aufteilung in Gitter- und Signalsegmentierung beschreibt die Struktur des Arbeitsablaufes der Bildauswertung mit interaktiven Systemen. Darin ist die Gittersegmentierung im Wesentlichen manuell auszuführen, während die anschließende Segmentierung und Quantifizierung der Spot-Intensitäten mit weitgehend automatischen, nicht interaktiven Verfahren erfolgt. Wegen der vielfältigen möglichen Störungen wird die Signalsegmentierung in vielen Fällen visuell überprüft und korrigiert und defekte oder verunreinigte Messpunkte werden markiert oder ausgesondert.

Die existierenden Verfahren zur Signalsegmentierung erfordern die Vorgabe von Näherungswerten für die Position und Größe des auszuwertenden Spots. Man hat gezeigt, dass die Ergebnisse der Intensitätsschätzung empfindlich von diesen Initialisierungen abhängen können [59, 107]. Im Hinblick auf die Reproduzierbarkeit und Integrierbarkeit von Daten aus verschiedenen Experimenten folgt daraus ein Bedarf sowohl für standardisierte, automatische Verfahren für die Gittersegmentierung als auch für robustere Spot-Segmentierung.

Zu diesen mehr theoretischen Überlegungen kommen praktische Aspekte: Beim Betrieb von Expressionsdatenbanken oder bei Einrichtungen, die viele Experimente als Dienstleistung durchführen, gibt es ein erhebliches Datenaufkommen, das eine manuelle Segmentierung nicht wünschenswert erscheinen lässt. Eine detaillierte visuelle Überprüfung von Spot-Segmentierungen ist in derartigen Anwendungsszenarien kaum zu leisten; automatische Bildverarbeitungsmethoden sind unbedingt erforderlich. Einige der in Abschnitt 3.3.6 aufgeführten Datenbanken nehmen mit beliebigen Werkzeugen quantifizierte Arraydaten an, obwohl es unter den gegebenen Voraussetzungen ratsam scheint, die Rohbilder zu speichern und alle Expressionsexperimente einer gemeinsamen Datenbasis mit den gleichen Methoden auszuwerten.

### 3.4.2 Eigenschaften von Mikroarraybildern

Im Abschnitt 3.3 sind die Array-Herstellung, die Hybridisierung und die Bildaufnahme und typischerweise dabei verwendete Methoden und Geräte beschrieben. Daraus ergeben sich verschiedene konkrete Anforderungen an Bildverarbeitungssysteme zur automatischen Mikroarray-Bildauswertung.

Zusammengefasst müssen folgende Punkte berücksichtigt werden:

1. Jede einzelne Druckspitze erzeugt ein periodisches Gitter von Messpunkten, aber nur ein Teil der Punkte gibt ein Signal, während andere dunkel bleiben.

2. Jedes einzelne der gedruckten Gitter ist näherungsweise periodisch, bis auf die durch Schwingungen etc. verursachten kleineren Störungen.

3. Das globale Muster der Messpunkte ist im Allgemeinen nicht periodisch. Die globale Periodizität hängt von der Anordnung der Druckspitzen im Arrayer ab.

4. Die Abstände von Zeilen und Spalten von Messpunkten in jedem Gitter sind im Allgemeinen nicht kalibriert. Gitterzeilen- und Gitterspaltenabstand können unterschiedlich sein.

5. Lücken zwischen den Gittern sind oft, aber nicht immer vorhanden.

6. Die Achsen aller Gitter in einem Array sind untereinander parallel, aber nicht unbedingt parallel zum Bildkoordinatensystem oder den Kanten des Glassubstrats.

7. Bis auf die Schwingungsstörungen und globale Rotationen ist die Anordnung der Messpunkte auf gemeinsam gedruckten Arrays gleich.

8. Die Messpunkte haben im Allgemeinen keine einheitliche Form und variieren in ihrer Größe.

9. Eingetrocknete Hybridisierungsflüssigkeit erzeugt ein großflächiges, sehr intensives Störsignal an den Bildrändern.

10. Reste der Waschlösungen und Artefakte des Hybridisierungsprozesses rufen ein über die Arrayoberfläche variables Hintergrundsignal hervor.

11. Die Gesamtintensität hängt von unkalibrierten Geräteeinstellungen ab (Laserintensität, Detektorverstärkung).

12. Bei der Bildaufnahme treten zwischen den Grauwertbildern der einzelnen Kanäle Verschiebungen bis zu einigen Pixeln auf. Die Verschiebung kann über das Gesamtbild variieren.

13. Die Bildaufnahme dauert je nach Gerät höchstens 10 bis 15 Minuten, während die Vorbereitung eines Mikroarrayexperiments Monate dauern kann.

Die ersten sieben Punkte sind für die Gittersegmentierung von Bedeutung, während die übrigen Punkte alle Aspekte der Verarbeitung von Mikroarraybildern betreffen.

Viele für die Segmentierung interessante Parameter wie z.B. die Meßpunktabstände sind prinzipiell bekannt oder können bei der Arrayherstellung vorgegeben werden. Gewöhnlich sind die im Entwurf vorgegebenen Messpunktkoordinaten, physikalische Gittergrößen und andere geometrische Parameter eines Mikroarrays in Dateien oder Datenbanken vorhanden. Unmittelbar ist davon für die Bildsegmentierung aber nur die Information über die logische Struktur der Messpunktanordnung verwertbar, weil die oben aufgelisteten Störfaktoren nicht erfasst werden.

Als Eingabeparameter der in dieser Arbeit beschriebenen Verfahren werden deshalb nur die Anzahlen von Spalten und Zeilen von Messpunkten je Gitter neben den Bildern selbst verlangt. Optional kann auch die Struktur des Druckkopfes, also die Zahl von Zeilen und Spalten von Druckspitzen, vorgegeben werden.

Die Bildaufnahmedauer gibt den in einem integrierten, automatisierten System für die Bildauswertung zur Verfügung stehenden Zeitrahmen vor. Der Zeit- und Arbeitsaufwand für das Experiment insgesamt ist wesentlich höher, weshalb die möglichst vollständige Auswertung auch von nicht optimal verlaufenen Experimenten notwendig ist. Unerkannte Fehler der Gittersegmentierung sind aus dem gleichen Grund weitgehend inakzeptabel, es sollte also möglichst eine automatische Rückweisung von unsicheren Ergebnissen geben.

Die Abbildung 3.7 läßt viele der oben aufgezählten Störfaktoren erkennen, und zahlreiche weitere Beispiele sind im Anhang E sowie im Abschnitt 8.2 zu finden.

## 3.5 Zusammenfassung

Dieses Kapitel enthält einen Überblick über Methoden zur Genexpressionsmessung in der Genomforschung. Die in dieser Arbeit vorwiegend behandelten gedruckten Mikroarrays ermöglichen durch konkurrierende Hybridisierung die differentielle Expressionsanalyse von zwei Proben auf einem einzigen Substrat. Dabei können mehrere Zehntausend Sequenzen aus einer Oligonukleotid- oder cDNA-Bibliothek gleichzeitig betrachtet werden.

Daneben gibt es das ältere Southern-Blot-Verfahren, bei dem gegen Restriktionsfragmente hybridisiert wird, Membranhybridisierungen und die technisch aufwändigen in-situ-Oligonukleotidarrays.

Neben den nötigen Nasslaborverfahren werden die Bildaufnahmetechnik und das gewöhnlich verwendete lineare Modell des Hybridisierungssignals skizziert und einige typische Klassen von Experimenten aufgeführt. Ein allgemeiner Überblick über statistische Methoden zur Analyse von Mikroarrayexpressionsdaten findet sich z. B. im Überblicksartikel von Smyth und anderen [95].

Um die Mikroarrayexpressionsdaten handhabbar zu machen und optimal zu analysieren sind Datenbanksysteme erforderlich. Die Quantifizierung der Rohdaten ist eines der noch nicht befriedigend gelösten Probleme der Experimentauswertung, da es noch keine vollautomatischen, standardisierten Methoden gibt, die für die Integration und Auswertung größerer Mengen von Expressionsdaten nötig sind.

Die Bildsegmentierung ist ein Teil dieses Problems. Spezifische Eigenschaften von Mikroarraybildern, die beim Entwurf eines Bildverarbeitungssystems zu berücksichtigen sind, werden im letzten Abschnitt aufgelistet.

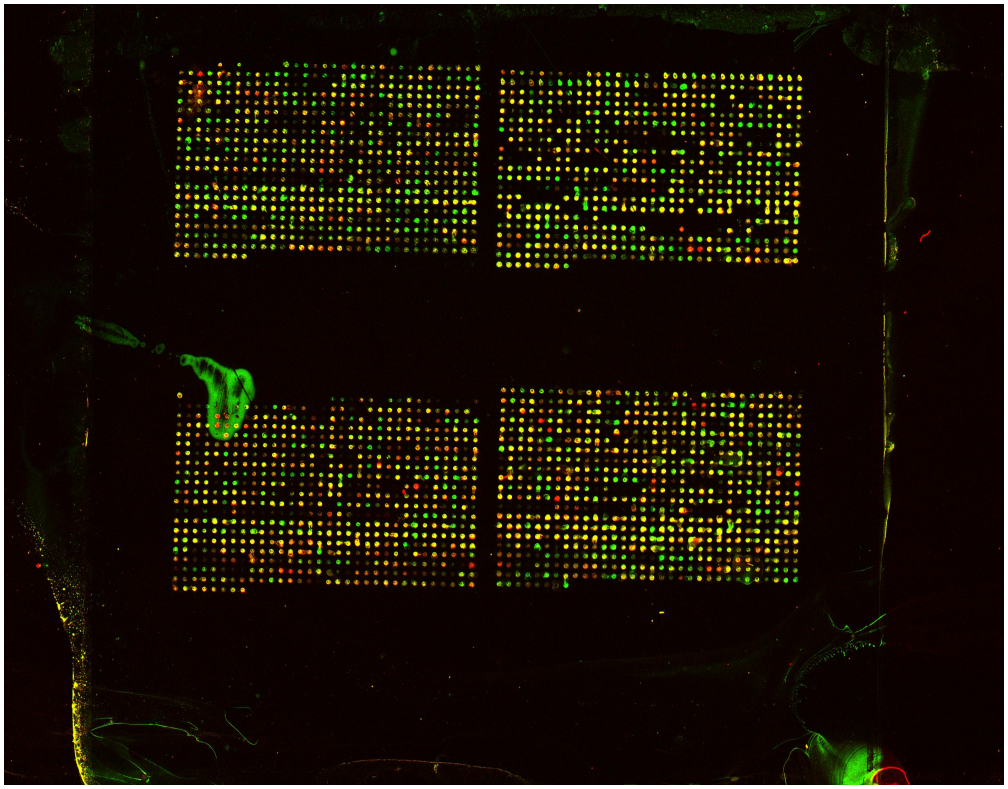


Abbildung 3.7: Ein Beispielbild aus der NMHy-Stichprobe (siehe 8.2). Neben der offensichtlichen Rotation ist erkennbar, dass die Messpunktgitter nicht aufeinander ausgerichtet sind, also kein global periodisches Raster vorhanden ist.

# 4 Systemansätze

## 4.1 Ziele

Das übergeordnete Ziel dieser Arbeit ist die Entwicklung von automatischen Bildverarbeitungsmethoden für Mikroarraybilder, die zu biologisch relevanten Ergebnissen beitragen. Ein erstes konkretes Ziel ist, die bisher mit viel Handarbeit verbundene Gittersegmentierung mit Bildverarbeitungsmethoden zu automatisieren, wodurch insbesondere für Hochdurchsatzexperimente und andere Anwendungen mit hohem Datenaufkommen Vereinfachungen zu erreichen sind. Dabei müssen die Eigenschaften der Bilddaten berücksichtigt werden, die aus den Eigenschaften der Mikroarrayverfahren und ihrer Anwendungsumfelder folgen. Die Gittersegmentierung initialisiert die quantitative Auswertung und ist deshalb indirekt auch für die spätere Auswertung des Gesamtexperiments bedeutsam.

Darüber hinaus wird untersucht, ob mit neuen Bildverarbeitungsmethoden die quantitative Bildauswertung direkt verbessert werden kann. Dies Ziel ist erheblich komplexer, weil die Definition von „verbessern“ auf das jeweilige Experiment und seine biologisch formulierte Fragestellung bezogen sein muss. Durch den stärkeren Bezug auf die eigentlich bearbeiteten *biologischen* Probleme ist es aber auch von besonderem Interesse.

Es ist naheliegend und erforderlich, Lösungsansätze als System mit offenen Schnittstellen zu implementieren, wenn der Erfolg im Hinblick auf die biologischen Fragestellungen gemessen werden soll, denn Bildsegmentierung und -auswertung müssen mit Werkzeugen für abstraktere, experimentbezogene Analysen integrierbar sein, um ihre Auswirkungen auf nachfolgende Analyseschritte untersuchen zu können.

Im Folgenden werden einige in der Literatur beschriebene Ansätze zur Gitter- und Signalsegmentierung vorgestellt, eingeordnet und anschließend der eigene Systementwurf motiviert und dargestellt.

## 4.2 Andere Ansätze und Methoden

Automatisches Segmentieren von Mikroarraybildern ist bisher nicht sehr umfassend als Bildverarbeitungsproblem bearbeitet worden. Anfänglich wurde die manuelle Gittersegmentierung unter anderem wegen der noch kleinen Stückzahlen von Arrays noch nicht als Problem angesehen (siehe z. B. Y. Chen [28]), so dass zunächst intensiver an Methoden zur Signalsegmentierung und -interpretation gearbeitet wurde.

### 4.2.1 Gittersegmentierung-Problemorientierte Ansätze

In diesem Abschnitt werden verschiedene Verfahren beschrieben, die direkt aus Anwendungszusammenhängen heraus entwickelt worden sind oder aus anderen Gründen vorrangig das Ziel der Bereitstellung eines Softwarewerkzeugs zur Mikroarraybildsegmentierung verfolgen.

#### Achsenprojektion der Intensität

Jain, Tokuyasu, Snijders und andere beschreiben ein System zur automatischen, quantitativen Mikroarray-Bildauswertung [53].

Das beschriebene Verfahren gliedert sich in folgende Schritte:

1. Schätzen von Spot- und Grid-Abständen aus Projektionen der Intensität auf die Bildkoordinatenachsen

2. Bestimmung einer Gitteranordnung aus den Projektionen

3. Neuberechnung von lokalen Achsenprojektionen für jedes Gitter

4. Maximierung der Gesamtintensität an den Gitterknoten durch spalten- und zeilenweises Verschieben der Gitter

5. Optimieren der Positionen einzelner Spots

6. Signalsegmentierung (Schwellwertsegmentierung mit lokalem Histogramm)

Das verwendete Schätzverfahren für Spot- und Gitterabstände ist nicht näher spezifizert. Die Autoren geben eine Erfolgsquote der automatischen Gittersegmentierung von 80-90% auf einer teilweise öffentlich zugänglichen Stichprobe an (siehe 8.2).

Das Verfahren ist nicht robust gegen Rotationen und helle Verunreinigungen.

#### Segmentierung von Membranradiografien

Steinfath [100] und Brändle [20] beschreiben Werkzeuge und Methoden vorwiegend zur Auswertung der Radiographien von Membranhybridisierungen. Steinfath benutzt zur Gittersegmentierung globale Achsenprojektionen und einen Algorithmus, der projektive Verzerrungen der Membran schätzt und korrigiert. Die Segmentierung stützt sich wesentlich auf ein detailliertes, nur für Hybridisierungsmembranen gültiges Intensitätsverteilungsmodell.

Brändle benutzt ebenfalls Achsenprojektionen, formuliert seinen Ansatz aber sehr allgemein über die Radontransformation. Dadurch wird Rotationsrobustheit erreicht, allerdings nur durch sehr aufwändige Grauwertprojektionen auf mehrere, nicht an das Bildkoordinatensystem gebundene Achsen. Brändle setzt Gaussfunktionen für die räumliche Intensitätsverteilung der Messpunkte an und beschreibt Schätzverfahren zum simultanen Anpassen der Parameter vieler Punkte.

#### Semiautomatische und andere Ansätze

Es gibt zahlreiche Softwarepakete zur interaktiven Mikroarray-Bildauswertung. Meistens ist darin die Segmentierung teilweise automatisiert.

- *ScanAlyze* von M. Eisen ist in den Anfangszeiten der Mikroarray-Methode entstanden [38] und wird noch immer verbreitet genutzt. Die Gittersegmentierung geschieht rein interaktiv.

- *Dapple* von J. Buhler ist für die halbautomatische Auswertung von Bildserien konzipiert. Das Programm verlangt die vollständige numerische Angabe der gedruckten Gitteranordnung sowie der Position des Spots oben links in jedem Bild. Die Spot-Positionen werden damit lokal für größte Intensität optimiert. Zur Signalsegmentierung werden anschließend Kreise an die Messpunkte angepasst.

  Als einziges der frei verfügbaren Programe bietet *Dapple* die Möglichkeit, auf einem handklassifizierten Teil der Quantifizierungsergebnisse einen Klassifikator zu trainieren, mit dem z. B. fehlerhafte Messpunkte in den übrigen Bildern markiert werden können.

- *Spot* von Yang et al. [107] benutzt ein Bild einer Serie als Template zur Gittersegmentierung. Auch hier müssen interaktiv die Ecken der Gitter angegeben werden.

  Zur Signalsegmentierung wird Seeded Region Growing [1] benutzt.

### 4.2.2 Gittersegmentierung -Methodenorientierte Ansätze

#### Verallgemeinerte Hit-or-Miss-Transformation

Vesanen, Tiainen und Yli-Harja beschreiben eine Verallgemeinerung der Hit-Miss-Transformation zur Segmentierung von Spots in Filter-Arraybildern [103]. Sie geben zusätzlich ein Verfahren an, mit dem aus einer klassifizierten Stichprobe ein strukturierendes Element gelernt werden kann und evaluieren die Methoden durch Vergleich mit der Mann-Whitney-Segmentierung (Siehe 7.2.1) auf zwei Beispielbildern.

Die Hit-Miss-Transformation wird als Rangordnungsoperation auf Grauwerten verallgemeinert, um Invarianz gegenüber Veränderungen der globalen Bildintensität zu erreichen. Zum Lernen der Masken wird das Problem als Lernen boolescher Funktionen formuliert und mit entsprechenden Methoden aus der Literatur bearbeitet.

Die in der Evaluation verwendeten Bilder zeigen kaum Größenvariation der Spots, weshalb unklar bleibt, wie allgemein das Verfahren anwendbar ist. Ein Vorteil gegenüber der Mann-Whitney-Segmentierung ist, dass die keine Näherungen der Spotpositionen bekannt sein müssen, dafür wird aber eine klassifizierte Stichprobe benötigt.

#### Morphologische Operationen

Mehrere Arbeiten beschreiben Segmentierungsverfahren, in denen morphologische Operationen zur Spotsegmentierung zum Einsatz kommen [5, 43, 46].

Angulo und Serra [5] sowie Hirata und andere [46] beschreiben sehr ähnliche Bildverarbeitungsketten zur Gitter- und Spot-Segmentierung, die fast ausschließlich aus morphologischen Operationen konstruiert sind. Das Griddingverfahren benutzt wie

der zuvor in Abschnitt 4.2.1 beschriebene Ansatz Achsenprojektionen. Die Rohbilder werden hier vor der Projektion mit einem *opening*-Operator behandelt, der benachbarte helle Spots zu geschlossenen, hellen Bereichen verbindet. Zur Schätzung der Spotgröße wird das sog. „morphologische Auslöschungsspektrum“ eines Bildes eingeführt, das mit Hilfe einer Folge von Erosionen mit quadratischen strukturierenden Elementen zunehmender Größe definiert wird. In dem Spektrum ist über der Größe des strukturierenden Elements der Anteil der lokalen Intensitätsmaxima des Ausgangsbildes aufgetragen, der bei der jeweiligen Elementgröße nicht mehr erhalten bleibt. Wenn die Spots annähernd gleich groß sind, ergibt sich bei der entsprechenden Elementgröße ein Maximum.

Auch in diesem Verfahren wird eine zweite, lokale Achsenprojektion berechnet, mit der die einzelnen Messpunktpositionen bestimmt werden. Für die Aufbereitung und Zerlegung der Projektion geben die Autoren einen heuristischen, aus mehreren eindimensionalen morphologischen Operationen, Mittel- und Schwellwertberechnungen aufgebauten Algorithmus an. Rotationen und eng benachbarte oder versetzte Gitter werden nicht behandelt.

Zur Spot-Segmentierung wird die Wasserscheiden-Transformation benutzt, die auf ein Differenzbild aus dem stark geglätteten Originalbild und dem dilatierten Originalbild angewendet wird.

Die Autoren schlagen eine modifizierte Wasserscheiden-Transformation vor, die das Problem der Übersegmentierung lösen soll: Zunächst werden ähnlich den Saatpixeln im Seeded Region Growing-Verfahren Marker an vermutete Positionen von Spots im Bild gesetzt. Das Vorgehen birgt offenbar technische Schwierigkeiten, wenn Marker auf dunkle Spotpositionen gesetzt werden, denn es wird folgende Heuristik zur Spot-Detektion verwendet:

1. Wenn in der Umgebung der geschätzten Spotposition kein lokales Intensitätsmaximum existiert, ist der Spot dunkel und es wird kein Marker gesetzt.

2. Wenn genau ein Maximum in der Umgebung existiert, ist der Spot sichtbar und es wird eine Marke auf das Maximum gesetzt.

3. Wenn mehrere Maxima vorhanden sind, wird eine Marke auf einen Zentroiden der Maximalstellen gesetzt.

Als Marker für den Hintergrund werden Linien benutzt, die zwischen den Reihen und Spalten der Messpunktgitter verlaufen. Die Wahl der Marker ist den von Yang et al. [107] vorgeschlagenen Saatpunkten für die Spotsegmentierung mit Seeded Region Growing sehr ähnlich.

Die Robustheitseigenschaften der beschriebenen Methoden bleiben weitgehend unklar, weil für die Evaluation zwar recht stark verrauschte, aber bezüglich der Spot- und Gitteranordnung fast perfekte Bilder verwendet werden.

Die Arbeit von Hirata und anderen [46] geht zumindest auf Probleme mit dunklen Randzeilen ein und schlägt verschiedene Heuristiken zur Abhilfe vor.

### k-Nächste-Nachbar-Graphen zur Gittersegmentierung

Jung und Cho beschreiben ein Verfahren zur Gittersegmentierung auf Basis von (modifizierten) k-nächste-Nachbarn-Graphen (kNN-Graphen bzw. mkNN-Graphen) [56].

Die Knoten beschreiben die Messpunktregionen eines Bildes, die in den kNN-Graphen genau dann durch eine Kante verbunden sind, wenn zwischen den Schwerpunkten der betreffenden Regionen eine Nächste-Nachbar-Beziehung gilt. Die Autoren stellen fest, dass die kNN-Graphen der Regionen eines Gitters häufig in viele Komponenten zerfallen und führen das auf sog. reziproke Paare von Regionen zurück, die wechselseitig nächste Nachbarn sind. Sie geben daher einen Algorithmus zur Konstruktion von mkNN an, in dem bei der Bestimmung des nächsten Nachbarn einer Region bereits verbundene Regionen unberücksichtigt bleiben. Dadurch kann es auch Kanten zu k-nächsten Nachbarn geben, was zu stärkerer Vernetzung führt.

Im nächsten Schritt des Verfahrens werden unverbundene Komponenten des mkNN-Graphen zusammengefasst, wenn ihre umschreibenden Rechtecke überlappen. Die Autoren geben eine statistische Abschätzung an, wie dicht die Gitterbesetzung mit Regionen und wie breit die Lücken zwischen den Gittern sein müssen, damit verschiedene Gitter sicher getrennt werden. Sie vernachlässigen jedoch die Existenz von Regionen, die nicht Messpunkte, sondern Rauschen oder Verunreinigungen beschreiben.

In den mkNN-Graphen können auch Regionen verbunden sein, die in einer rechteckigen Gitterzelle diagonal gegenüberliegen. Jung und Cho führen deshalb eine zweite Graphkonstruktion ein, die sie $\epsilon$-Graphen nennen. Die Knoten eines $\epsilon$-Graphen sind die Regionen in dem umschreibenden Rechteck eines Gitters. Von jedem Knoten können Kanten in die vier Richtungen entlang der Koordinatenachsen ausgehen. Kanten werden zur jeweils nächstliegenden Region entlang der vier Richtungen eingefügt. Eine kleine Abweichung $\epsilon$ von den Koordinatenachsen wird zugelassen, wodurch eine geringe Rotationsrobustheit erreicht wird.

Jung und Cho benutzen für die Nächste-Nachbar-Suche Algorithmen mit quadratischer Effizienz bezüglich der Regionenanzahl.

In den folgenden Kapiteln wird gezeigt, dass strukturelle Methoden wie die mkNN-Graphen prinzipiell sehr gut geeignet sind, um Rotationsrobustheit zu erreichen. Das Verfahren von Jung und Cho nutzt diese Möglichkeiten kaum aus.

### Markov Random Field für lokale Gittersegmentierung

Carstensen und Hartelius [25, 26] beschreiben ein Markov Random Field-Modell (siehe Kap. 6.4) für periodische Rechteckgittertexturen. Die Segmentierung der periodisch wiederholten Texturzellen erfolgt durch Template Matching und ist in das Modell integriert. Es ist aus zwei Komponenten aufgebaut, die verschiedene Arten von Störungen der regelmäßigen Rechteckgitterstruktur beschreiben. Die erste Komponente erfasst lokale, gaussverteilte Variationen der Gitterknoten-Orte. Für sie gilt die einfache, durch die Gitterstruktur vorgegebene Vierer-Nachbarschaft von Gitter*knoten*. Die zweite Komponente beschreibt (ebenfalls gaussverteilte) Längenvariationen der Gitter*kanten*. Durch Verwendung getrennter Nachbarschaftssysteme für horizontal und vertikal benachbarte Kanten können damit systematische Störungen der Gitterstruktur modelliert werden.

Zur Energieminimierung geben die Autoren einen Ensemble-Annealing-Algorithmus an, bei dem um einen Initialisierungspunkt im Parameterraum des Modells zufällige Startpunkte (das sog. Ensemble) initialisiert werden. Alle Zustands-Punkte werden stochastisch mit dem Metropolis-Algorithmus (siehe S. 78) über eine feste Zahl

von Iterationen bezüglich der MZF-Energie optimiert. Anhand der Verteilung der Restenergien im Ensemble werden 'gute', also global minimierte Ergebnisse identifiziert.

Die Autoren wenden ihr Modell auf Textilmaschen und Spots auf Hybridisierungsmembranen an. Zur Initialisierung der Ensemble-Minimierung wird die Gitterrotation durch eine Hough-Transformation geschätzt, so dass die Ränder danach durch Achsenprojektion bestimmt werden können. Als Template zur Segmentierung der Array-Spots dient eine Gaussmaske.

### 4.2.3 Signal-Segmentierung und Quantitative Auswertung

Die in der Literatur beschriebenen Methoden zur Segmentierung der Messpunktregionen lassen sich gut danach einordnen, ob sie die Zielregionen geometrisch oder durch statistische Eigenschaften von Bildintensitäten festlegen. Alle hier aufgeführten Verfahren benutzen die bei der Gittersegmentierung festgelegten Spot-Abstände.

#### Geometrische Verfahren

Die einfachste Methode zur Signalsegmentierung benutzt Kreise als Messpunktregionen. Diese Methode ist in fast allen Systemen als Referenzmethode implementiert. Mit „Scanalyze", dem ältesten Werkzeug zur Mikroarray-Bildauswertung mussten anfänglich Kreisradien manuell eingestellt werden. Inzwischen werden in den meisten Programmen Optimierungsverfahren benutzt, um die Radien optimal einzustellen (Dapple [23], Scanalyze [38]). Teilweise werden auch elliptische Regionen benutzt ([21], GenePix, ImaGene). Die Hintergrundregion wird bei Scanalyze von allen Nicht-Signalpixeln gebildet, während GenePix kleine Fenster diagonal zwischen den Messpunkten benutzt.

Die Programme von Steinfath und Brändle (siehe Abschnitt 4.2.1), die in erster Linie zur Quantifizierung von Membranhybridisierungen gedacht sind, benutzen ein parametrisches Gauss-Modell und eine Mischung von Gauss-Funktionen für die räumliche Intensitätsverteilung [100, 20].

Der praktische Vorteil der rigideren geometrischen Messpunktsegmentierung besteht darin, dass in jedem Fall eine Region pro Messpunkt bestimmt werden kann. Das ist bei den statistischen Verfahren nicht unbedingt der Fall.

#### Statistische Verfahren

Es gibt eine wesentlich größere Vielfalt an Verfahren, die statistische Methoden zur Messpunktsegmentierung einsetzen. Einige einfache Verfahren benutzen aus der Standardabweichung des lokalen Hintergrundes abgeleitete Schwellwerte [104] oder Quantile [53, 56, 87]. Die Hintergrundstichprobe bzw. die größtmögliche Signalregion werden jeweils durch Kreise mit festen Radien definiert.

Glasbey und Ghazal [43] haben die Eigenschaften der Kreissegmentierung, der Segmentierung mit Quantil-Schwellwert und eine Variante des Otsu-Verfahrens [1] (siehe Abschnitt 5.2.3) verglichen. Sie setzen ein Modell für die Abhängigkeit von Mittelwert und Varianz der Intensitäten der Signalpixel an und untersuchen, bei welcher

[1] k-Means-Clustering der Quadratwurzel der Intensitäten mit zwei Klassen

Methode der Signalsegmentierung das Modell die Intensitäten am besten erklärt. Die quantilbasierte Methode hatte hierbei die besten Eigenschaften.

Ein Verfahren von Y. Chen benutzt den parameterfreien Mann-Whitney-Test zur Schwellwertberechnung [28]. Die Mann-Whitney-Segmentierung wird in Abschnitt 7.2.1 eingehender behandelt.

Yang und andere haben das *Seeded Region Growing (SRG)* -Verfahren von Adams und Bischof mit der quantilbasierten Segmentierung und der Kreissegmentierung verglichen. Seeded Region Growing erweitert eine vorgegebene Zahl von Regionen ausgehend von Startstellen (den „Seeds“), indem Pixel vom aktuellen Regionenrand zu der Region hinzugefügt werden, die den ähnlichsten mittleren Grauwert besitzt. Es wird immer der Pixel zuerst bearbeitet, der die geringste Grauwertdifferenz zu einer der Regionen hat. Die Autoren geben eine heuristische Definition der Startstellen an, die gute Segmentierungen erzeugen soll, aber nicht garantiert, dass für jeden Messpunkt getrennte Regionen entstehen. Sie diskutieren die Eigenschaften der Verfahren anhand der *Apo A1*-Stichprobe (siehe Abschnitt 8.2) vor allem im Hinblick auf die Hintergrundkorrektur, in der sie die größte Fehlerquelle sehen. SRG liefert nach dieser Untersuchung die besten Signalintensitätsschätzungen, während das Kreisverfahren von Scanalyze die besten Eigenschaften bei der Hintergrundkorrektur besitzten soll.

Bozinov und Rahnenführer haben zwei Clusterverfahren (*k-Means* und *Partitioning Around Medoids*) auf die Pixel-Intensitäten in einem Fenster um vermutete Messpunktpositionen angewandt [14]. Das Verfahren hat gegenüber den bisher aufgezählten Ansätzen den Vorteil, dass keine Vorabfestlegung von möglichen Hintergrund- oder Signalbereichen nötig ist. Die Interpretation der Clusterstrukturen ist jedoch nicht immer leicht und die Clusterung sehr rechenintensiv.

## 4.3 Entwurf einer Systemstruktur

Im Folgenden wird eine Bildverarbeitungskette zur Gittersegmentierung skizziert, deren einzelne Komponenten in den anschließenden Kapiteln genauer behandelt werden. Die Abbildung 4.1 gibt einen Gesamtüberblick über den Entwurf. Abschnitt 4.3.2 erläutert die Einbettung der Bildverarbeitungskette in ein Gesamtsystem, das die Daten strukturiert speichert und Benutzerschnittstellen bereitstellt.

### 4.3.1 Bildverarbeitungskette

Ausgangspunkt der Gittersegmentierung ist das logische Modell der Anordnung von Messpunktgittern, die es im Bild zu finden gilt. Die Anzahlen von Zeilen und Spalten von Messpunktgittern werden vom Benutzer vorgegeben und optional auch die Anzahl von Gittern, d. h. die Struktur der Nadelanordnung des benutzten Druckkopfes.

Da das System kalibrationsfrei arbeiten soll, ist die Größe der Gitter im Bild zunächst unbekannt. Auch über die absolute Hintergrund- bzw. Messpunktintensität kann zunächst nichts ausgesagt werden, denn die Bildaufnahmeparameter können sich von Bild zu Bild willkürlich ändern.

Die auflösungs- und intensitätsinvariante Eigenschaft der Messpunktanordnung ist ihre Periodizität innerhalb der einzelnen Gitter, die durch die statistischen Eigenschaften der Abstände benachbarter Messpunktzentren charakterisiert wird. Für diese Abstände wird ein Verteilungsmodell entwickelt, damit die invariante Eigenschaft genutzt werden kann.

Mit Schwellwertverfahren, die im Kapitel 5 genauer dargestellt werden, bestimmt man die Regionen der Grauwertbilder, die zusammenhängende helle Bildbereiche beschreiben. Durch einen freien Parameter des Schwellwertverfahrens werden verschiedene Intensitätsbereiche abgetastet, so dass mehrere alternative Regionensegmentierungen der Grauwertbilder entstehen. Eine gute Segmentierung sollte die Periodizität der Gitterknoten möglichst deutlich wiedergeben. Man benutzt also diejenige Regionensegmentierung, deren Abstandsverteilung benachbarter Regionen möglichst gut dem Verteilungsmodell der Abstände im Modellgitter entspricht. Dadurch werden gleichzeitig die Abstände der Messpunktzeilen und -spalten geschätzt.

Man könnte nun mit der Kenntnis der Gittergröße direkt versuchen, das logische Gittermodell an das Bild anzupassen. Die Beschreibung der Eigenschaften von Mikroarraybildern lässt aber offen, wie die zu segmentierenden Gitter genau aussehen, d. h. welche Messpunkte sichtbar sind und welche dunkel bleiben. Template-Matching und ähnliche Verfahren sind nicht zuletzt wegen der großflächigen hellen Verunreinigungen weniger geeignet. Vielversprechender ist die Nutzung der verschiedenen Aussagen über Lagebeziehungen der Gitter im Bild oder mit anderen Worten des Kontextwissens. Es wird also ein (heuristisches) stochastisches Modell der Kontexte von Messpunktgittern im Gesamtbild formuliert, wodurch dem Mangel an Detailwissen Rechnung getragen wird. Die geeignete Modellklasse sind Markov-Zufallsfelder (Markov Random Fields , MZF), mit denen stochastische Eigenschaften der Nachbarschaftsbeziehungen von Objekten elegant durch Energiefunktionen formuliert werden. Die Minimierung der Energie liefert eine im Sinne des Modells optimale Segmentierung.

Das MZF-Modell der Gittersegmentierung enthält diskrete Zufallsvariablen, die die Platzierungen der einzelnen Gitter beschreiben. Die Diskretisierung ist möglich, weil

die Gitterperiodizität bekannt ist und erlaubt die Anwendung effizienter Algorithmen zur Energieminimierung. Im Gegensatz zu dem MZF-Modell von Karstensen und Hartelius, das die Anordnung von Messpunkten in kontinuierlicher Form beschreibt, wird hier die Struktur der Gitteranordnung diskret modelliert.

Die Rotation der Messpunktgitter gegen das Bildkoordinatensystem wird durch Verketten periodisch angeordneter Regionen geschätzt. Dadurch können in robuster Weise Achsenprojektionen zur Bestimmung der Anzahl (wenn sie nicht vorgegeben ist) und der ungefähren Platzierung der Gitter eingesetzt werden. Die Verkettung hat konzeptionelle Ähnlichkeit zu den mkNN-Graphen von Jung und Cho, ist aber im Gegensatz dazu rotationsrobust und wird mit geeigneten Datenstrukturen (dem kd-Baum) effizienter implementiert.

Die Achsenprojektionen der Verkettungsergebnisse zeigen die ungefähren Positionen der Gitter an, so dass Hypothesen für die Platzierung der Gitter zur Verfügung stehen, die die Werte der Zufallsvariablen des MZF bilden. Im Kapitel 6 wird die Erzeugung der Hypothesen für die Gitterplatzierung genauer behandelt.

Die Qualität der MZF-Gittersegmentierung hängt sehr stark von der korrekten Erkennung von Messpunkten an den hypothetisierten Gitterknotenorten ab, auf der die MZF-Energiefunktion aufbaut. Ein leistungsfähiges Verfahren für die Messpunktdetektion sind die aktiven Konturen [57, 105]. (Abschnitt 6.6). Wegen ihrer lokal adaptiven Eigenschaften ermöglichen sie die Segmentierung von verwischten, verlaufenen und sehr dunklen Messpunkten, für die die Regionensegmentierung der Vorverarbeitungskette meist keine oder viel zu große Regionen erzeugt.

Die in der Literatur beschriebenen aktiven Konturmodelle sind nicht ohne weiteres für die Segmentierung von Mikroarray-Spots einsetzbar, denn die gewöhnlich verwendeten Diskretisierungen sind wegen der teilweise geringen Größe der Zielregionen nicht zulässig. Ein weiteres Problem stellt das vergleichsweise starke Rauschen in Mikroarraybildern dar. Es wird daher ein teilweise kontinuierliches Konturmodell verwendet, dessen Energiefunktion mit geeigneten Nebenbedingungen robust minimiert werden kann.

Aus dem Konturmodell werden Merkmale abgeleitet, die die Klassifikation der Konturen zur Spot-Erkennung ermöglichen. Es können anhand von klassifizierten Beispielen trainierte Klassifikatoren oder ein an der Verteilung der Gradientenstärke des jeweiligen Bildes kalibriertes Schwellwertverfahren benutzt werden.

Zur Regionensegmentierung in der Vorverarbeitungskette sind die aktiven Konturen ungeeignet, da für die Initialisierung die Gitterkonstantenschätzung gebraucht wird. Zufällige Initialisierung führt zu unbrauchbaren Ergebnissen.

Alternativ wird zur Messpunkterkennung im Abschnitt 6.5 das Eigenwertverfahren untersucht, das wie die aktiven Konturen in vielen anderen Bildverarbeitungsanwendungen Anwendung gefunden hat. Anders als das aktive Konturmodell kann es jedoch nicht ohne eine klassifizierte Stichprobe von Bildern einzelner Messpunkte benutzt werden und es ist weniger robust gegen Skalierungen.

Die Messpunktsegmentierung mit dem aktiven Konturmodell wird auch zur quantitativen Auswertung angewendet. Dazu stehen noch zwei weitere, häufig verwendete Verfahren zur Auswahl, nämlich die Mann-Whitney-Segmentierung und die einfache Kreissegmentierung.

Die Abbildung 4.1 zeigt einen Überblick über das Bildverarbeitungssystem zur Gittersegmentierung. Die Schwellwertberechnungen und der Kantenoperator der akti-

ven Kontursegmentierung benötigen als Vorverarbeitung den Medianfilter zur Rauschunterdrückung. Die Gitterkonstantenschätzung beinhaltet die Anwendung des Abstandsverteilungsmodells.

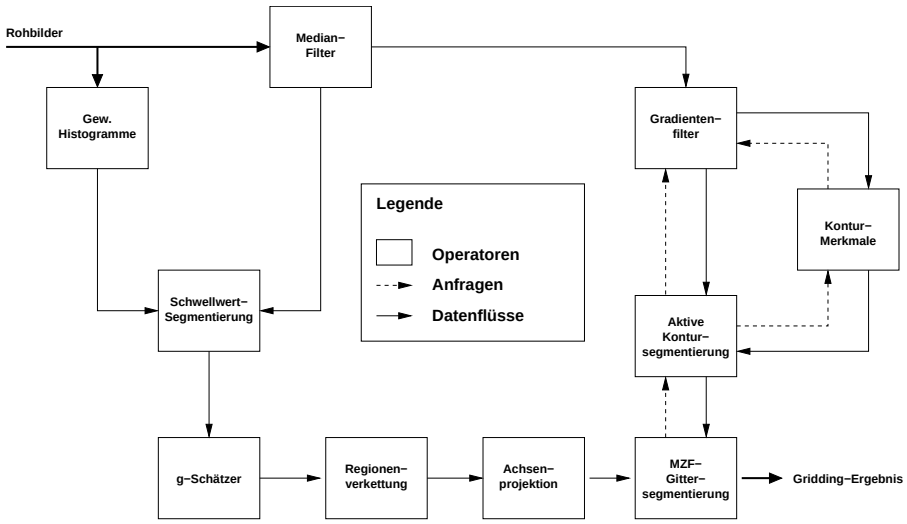



Abbildung 4.1: Ein Überblick über die Komponenten des Bildverarbeitungssystems zur Gittersegmentierung. Der rechte obere Zweig des Schemas, in dem sich die aktive Kontursegmentierung befindet, unterscheidet sich von dem Zweig links unten dadurch, dass die Kontursegmentierung nur nach Bedarf, gesteuert von der MZF-Segmentierung, berechnet werden. Der untere Zweig erzeugt die MZF-Hypothesen und arbeitet daher das gesamte Bild ab.

### 4.3.2 Ein Gesamtsystementwurf

Die Abb. 4.2 zeigt die Einbettung der Gittersegmentierung in das Gesamtsystem AIM (*Automatic Image processing system for Microarray experiments*). Das Datenschema des AIM-Systems umfasst *Projekte*, die Serien von gemeinsam gedruckten Mikroarrays enthalten. Jedes Projekt besitzt Attribute für die Struktur der Gitteranordnung, die Zahl der verwendeten Farbstoffe bzw. Bildkanäle und weitere Parameter wie z. B. die benutzte Anzahl von alternativen Schwellwertsegmentierungen. Die einzelnen Arrays sind als *Slides* abgebildet, für die jeweils Bilddateinamen, die geometrischen Daten der Segmentierung und quantitative Auswertungsergebnisse (Mittelwert und Median der Intensität auf der Messpunktregion, Mittelwert, Median und Modus der lokalen Hintergrundintensität) gespeichert werden.

Für die Eingabe und Bearbeitung von Projektdaten gibt es sowohl eine grafische als auch eine nicht-interaktive Schnittstelle, damit das System sowohl allein benutzt als auch in andere Systeme eingebunden werden kann. Die quantitative Auswertung findet in einem nicht-interaktiven Modul statt. Die Ergebnisse der automatischen Gittersegmentierung können mit einem interaktiven Werkzeug überprüft und bearbeitet werden, das auch Funktionen zur Inspektion der Rohbilder und der quantitativen Aus-

wertungsergebnisse besitzt. Innerhalb des AIM-Systems werden XML-Formate (eXtensible Markup Language) zur Datenspeicherung benutzt, deren Dokumenttypdefinitionen (DTDs) im Anhang D zu finden sind. Dadurch ist die Integration mit anderen Systemen, die ebenfalls XML-Schnittstellen wie MAGE-ML [97] oder GeneXML [75] besitzen, durch XSL-Transformationen (eXtensible Stylesheet Language, XSLT) technisch elegant möglich. XSLT ist eine funktionale Programmiersprache, mit der Übersetzungsregeln für die Komponenten der Hierarchien verschiedener XML-Formate formuliert werden können. Außerdem ist ein Export-Modul für das häufig verwendete GPR-Format vorgesehen, das von der kommerziellen Mikroarraybildauswertung GenePix (Axon Inc.)[2] stammt und auch in einigen Bioconductor-Paketen[3] verwendet wird. Dies ist eine frei verfügbare Sammlung von Programmen zur statistischen Auswertung von Mikroarrayexperimenten, die zunehmende Aufmerksamkeit erhält.

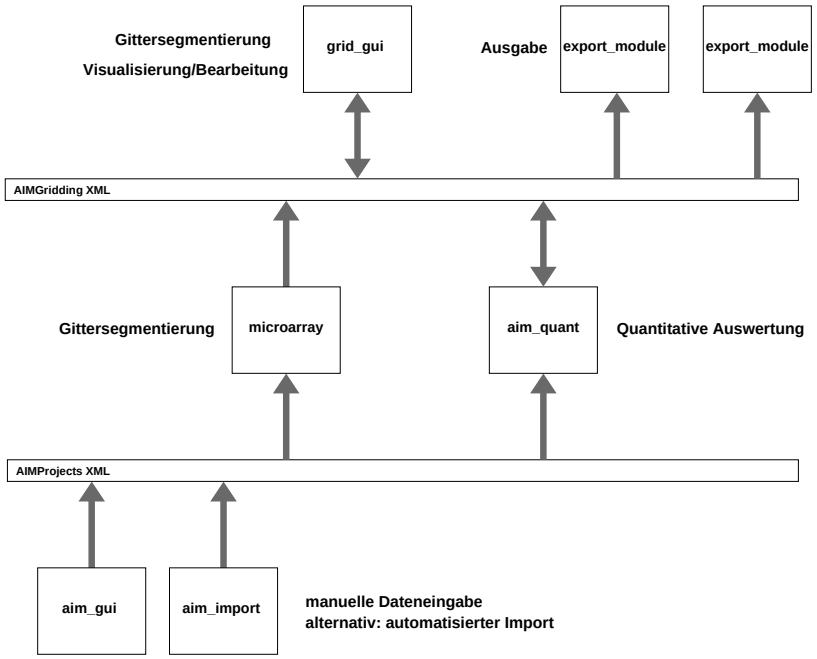



Abbildung 4.2: Überblick über die Struktur des AIM-Systems. Die Kästen stellen einzelne Programmkomponenten des Systems dar zwischen denen Daten mittels der beiden XML-Formate (lange Balken), deren Dokumenttypdefinitionen im Anhang D zu finden sind, ausgetauscht werden. Das Schema wird in der typischen Anwendung von links nach rechts durchlaufen. Der Prozess kann entweder über die grafische Benutzeroberfläche gesteuert oder auf Kommandozeilenebene abgewickelt werden.

Das AIM-System benutzt nicht direkt das zu MAGE-ML gehörende Datenmodell

[2] www.axon.com
[3] www.bioconductor.org

von MAGE-OM, weil darin viele hier benötigte Attribute für quantitative Auswertungsergebnisse fehlen und gleichzeitig sehr viele Entitäten und Attribute vorhanden sind, die für die Bildsegmentierung und -Auswertung nicht relevant sind. Es erscheint darüber hinaus zweckmäßiger, die Dynamik der Entwicklung von MAGE aus dem Bildverarbeitungssystem selbst herauszuhalten und stattdessen alle Abhängigkeiten von externen Standards in einzelnen Ein-/Ausgabemodulen zu isolieren.

# 5 Regionensegmentierung

Die Regionensegmentierung soll primär eine abstraktere Repräsentation der lokal periodischen Anordnung der Messpunkte in den Grauwertbildern liefern. Dabei brauchen die Form und die genaue Größe der Messpunkte zunächst keine Rolle spielen. Dies Kapitel behandelt einfache Schwellwertverfahren, mit denen zusammenhängende helle Bildbereiche (oder genauer: zusammenhängende Mengen von Pixeln mit Intensität über einem Schwellwert) als Regionen bestimmt werden. Das Ziel ist, zu den (sichtbaren) Messpunkten in jedem Kanal eine Region zu bestimmen. Das anschließende Kapitel behandelt die Analyse der Abstandsverteilungen der Regionenschwerpunkte, die mit der Regionensegmentierung selbst gekoppelt werden muss, um die robuste Segmentierung der Messpunkte zu erreichen.

## 5.1 Vorverarbeitung

Alle im Folgenden beschriebenen Verfahren liefern mit einer vorgeschalteten Rauschunterdrückung wesentlich bessere Ergebnisse, weshalb zunächst auf den Medianfilter eingegangen werden soll, der sich als Vorverarbeitungsoperator bewährt hat. Der Medianoperator ist schon seit langem bekannt und gehört zu den sog. Rangordnungsoperatoren. Er wird mit Hilfe einer quadratischen Maske von Pixeln mit ungerader Kantenlänge definiert. Der Operator gibt für jeden Pixel den Median der Intensitätswerte aus, die sich in der auf ihn zentrierten Maske befinden. Da der Median anders als z. B. der Mittelwert robust gegen Ausreißer ist, werden wenige Pixel große Artefakte sehr gut unterdrückt, wogegen größere Strukturen nicht gestört werden.

Dagegen verwischt eine Glättung von Mikroarraybildern durch Faltung mit Gauss- bzw. Binomialmasken die Messpunkte so sehr, dass sie nur noch schlecht als getrennte Regionen segmentierbar sind.

Medianfilterung von Mikroarraybildern bewirkt die Unterdrückung der hellen, kleinen Störungen durch verunreinigende Partikel auf dem Array. Eine Maskengröße von $5 \times 5$ Pixeln ergibt gute Rauschunterdrückung ohne Beeinträchtigung der Bildqualität.

Zur Berechnung des Medianfilters gibt es neben der trivialen Möglichkeit der Sortierung der Intensitätswerte verschiedene optimierte Verfahren. Welches davon optimal ist, hängt von der Maskengröße und der Intensitätswertemenge des zu filternden Bildes ab.

### 5.1.1 Histogrammbasierte Medianberechnung nach Huang

Huang [48] und andere haben einen Algorithmus beschrieben, der bei der Verschiebung der Maske über das Bild ein Histogramm der Pixelintensitäten unter der Maske und einen Zeiger auf das darin enthaltene Medianelement mitführt. Zur Initialisierung

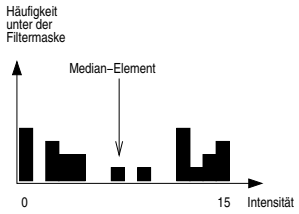



Abbildung 5.1: Medianberechnung mit Histogramm

z. B. in der linken oberen Bildecke werden einmal alle Intensitätswerte unter der Maske eingetragen. Der Median hat dann den Wert der kleinsten Intensität, von der aus mindestens die Hälfte der Histogrammeinträge links liegen (Siehe auch Abbildung 5.1). Die Berechnung des Medians des Nachbarpixels kann durch Aktualisierung des Histogramms und des vorhergehenden Medianwertes erfolgen: Die Maske wird „weitergerückt“, so dass je Zeile ein Histogrammeintrag entfernt bzw. ein Korb heruntergezählt und ein Eintrag neu registriert werden muss. Abhängig davon, ob die beiden ausgetauschten Einträge größer, kleiner oder gleich dem vorherigen Medianwert sind, ändert sich der neue Medianwert. In dem Beispiel der Abb. 5.1 müsste der neue Median auf den nächsttieferen besetzten Intensitätswert gesetzt werden, wenn ein Pixel mit Intensität 0 hinzugefügt und ein Pixel mit Intensität 15 herausgenommen würde. Sind die ausgetauschten Intensitätswerte beide kleiner oder beide größer als der alte Median, so ändert sich dieser nicht. Um Körbe mit mehreren Einträgen korrekt behandeln zu können, ist eine Hilfsvariable nötig, die die Position des Medianelements in einem gedachten „Stapel“ von Elementen im Korb angibt. Die Maske wird optimalerweise in Schlangenlinien Zeile für Zeile über das Bild geführt, so dass nur im allerersten Schritt das ganze Histogramm berechnet werden muss.

Das Verfahren ist insbesondere dann sehr schnell, wenn nur wenige Histogrammkörbe besucht werden müssen, um das neue Medianelement zu finden. Für die 16 Bit tiefen Mikroarraybilder hat das Histogramm $H = 65536$ Körbe, so dass bei kleinen Masken und besonders bei starkem Rauschen große Lücken zwischen den belegten Histogrammkörben liegen. Daher erhält man bei der Filterung solcher Bilder erheblich schlechtere Laufzeiten als bei der Filterung von gewöhnlichen, 8 Bit tiefen Bildern mit Histogrammlängen von $H = 256$. Die laufzeitbestimmende Größe ist die Länge $L$ der Bereiche leerer Histogrammkörbe. Im ungünstigsten Fall gleichverteilter Intensitätswerte ist die Lückenlänge $L$ poissonverteilt mit dem Erwartungswert $< L >= \frac{H}{M^2}$ bei Maskenkantenlänge $M$. Mit 50% Wahrscheinlichkeit muss der Median nach einer Austauschoperation verändert werden, so dass bei ungünstigen Bilddaten je Bildpixel etwa $\frac{H}{2M}$ Histogrammkörbe geprüft werden müssen.

### 5.1.2 Histogrammhierarchie

Der einfache Histogrammalgorithmus zur Medianberechnung verbraucht viel Zeit bei der Suche nach besetzten Histogrammkörben. Eine mögliche Abhilfe bietet eine Histogrammhierarchie wie sie in Abb. 5.2 skizziert ist. Die unterste Hierarchiebene (Tiefe $t = 0$) stellt das gewöhnliche Histogramm dar. Jede höhere Ebene besteht aus der halben Anzahl von Körben der darunterliegenden Ebene, die jeweils die Intensitäts-

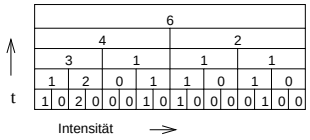



Abbildung 5.2: Hierarchie von Histogrammen

werte von zwei benachbarten Körben der tieferen Ebene umfassen. Das hierarchische Histogramm hat ld$H$ Ebenen, weshalb das Registrieren oder Entfernen von Intensitätswerten ebensoviele Schritte benötigt, weil die Körbe in allen Ebenen verändert werden müssen. In der Ebene der Tiefe $t$ gibt es $\frac{H}{2^t}$ Körbe.

Die Suche nach dem nächsten belegten Korb zur aktualisierung des Medians beginnt an einem Korb der unteren Ebene, von dem aus iterativ Nachbarkörbe geprüft werden. Wird eine Korbgrenze der darüberliegenden Ebenen erreicht, dann darf der Suchprozess in höheren Ebenen fortgesetzt werden, bis ein belegter Korb gefunden wird. Von dort erfolgt der Abstieg in die untere Ebene, wobei in jeder Ebene der am nächsten zum Start liegende, belegte Korb angesteuert wird. In jeder Ebene müssen dazu zwei Körbe geprüft werden. Um einen $L$ langen leeren Bereich in der untersten Ebene zu überschreiten, werden damit für Auf- und Abstieg jeweils höchstens $2\text{ld}L$ Schritte benötigt. Zusammen mit der Entfernungs- und Registrieroperation ergibt sich also ein Aufwand von $6\text{ld}L$ Schritten je Austauschoperation oder $3M\text{ld}\frac{H}{M^2}$ berührten Körben je Maskenverschiebung.

Das Diagramm in Abbildung 5.3 stellt die erwartete Zahl von Operationen im schlechtesten Fall für die Median-Algorithmen mit einfachem und hierarchischem Histogramm bei $H = 65536$ in Abhängigkeit von der Maskengröße dar. Bei der benutzten Maskengröße $5 \times 5$ arbeitet die hierarchische Variante mit erheblich weniger Schritten.

In der Praxis ist die Median-Berechnung mit dem hierarchischen Histogramm sogar etwas langsamer, was sicherlich zum Teil daran liegt, dass die Abschätzungen nur den für das Verfahren von Huang ungünstigsten Fall betrachten. Ein weiterer Grund besteht in den unterschiedlichen Korbsuchverfahren. Das hierarchische Verfahren braucht umständlichere Fallunterscheidungen, erzeugt kompliziertere, über größere Speicherbereiche gestreute Zugriffe und profitiert deshalb vermutlich weniger von Hardware-Optimierungen[1].

### 5.1.3 Heap-Median

Härdle und Steiger [45] beschreiben einen weiteren Ansatz zur Median-Berechnung, der zwei gleich große Heaps zur kompakten Speicherung der Intensitätswerte unter der Maske einsetzt. Einer der Heaps ist aufsteigend sortiert und enthält alle Intensitätswerte unter der Maske, die heller oder gleich der Median-Intensität sind, wogegen der andere Heap absteigend sortiert ist und die Intensitätswerte unter der Maske kleiner oder gleich der Median-Intensität enthält. Durch die Heap-Bedingung ist sichergestellt, dass die obersten Elemente der Heaps die Nachbarn des Median-Elementes in einer gedachten sortierten Liste der Intensitäten unter der Filtermaske sind.

[1] Der Unterschied ist bei Maschinen mit großem Cache-Speicher geringer

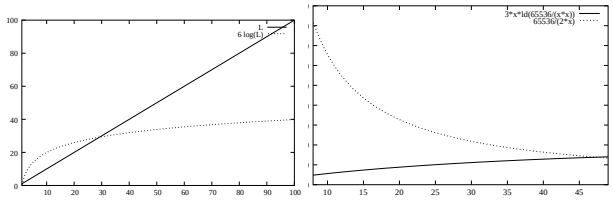



Abbildung 5.3: Effizienzvergleich der Aktualisierung des Median-Zeigers mit dem einfachen und dem hierarchischen Histogramm. Links: Aufwand abhängig von der Länge leerer Histogrammbereiche, rechts: Aufwand abhängig von der Maskengröße.

Beim Weiterschieben der Maske auf dem Bild werden ähnlich wie bei der histogrammbasierten Berechnung Austauschoperationen vorgenommen, die die Größe und Sortierbedingungen der Heaps erhalten. Dazu muss evtl. auch das Median-Element in einen der Heaps eingetragen und durch das obere Element des anderen Heaps ersetzt werden.

Der Aufwand für diese Operationen hängt nicht mehr von der Intensitätswertemenge sondern nur noch von der Heap- bzw. Maskengröße ab. Das Wiederherstellen der Sortierbedingung in einem Heap mit $N$ Elementen nach Austausch zweier Elemente erfordert $O(\log N)$ Operationen. Der Aufwand für die Median-Aktualisierung beim Verschieben einer $M \times M$-Maske beträgt also $O(2M \log M)$.

Auf einem Beispielbild erreicht man bereits mit einer einfachen Implementation dieses Verfahrens, die noch nicht alle Optimierungsmöglichkeiten nutzt eine Geschwindigkeitssteigerung um 34% gegenüber dem Verfahren mit dem einfachen Histogramm [85][2].

## 5.2 Schwellwertverfahren

Allgemein ist die Anwendung von Schwellwertverfahren angezeigt, wenn es ein einzelnes Merkmal gibt, das die Objektklassen eines Bildes (einschließlich Hintergrund) trennt (siehe z.B. [52]).

### 5.2.1 Grundlagen

Abgesehen von den großräumig variablen Hintergrundeigenschaften erfüllt die Intensität in den Mikroarray-Fluoreszenzbildern diese Voraussetzung. Man sucht also lokal für bestimmte Bildbereiche gültige Schwellwerte, die Hintergrund und Messpunkte trennen, und nicht einen globalen Schwellwert für das gesamte Bild. Die im Folgenden behandelten Methoden der Schwellwertberechnung stützen sich daher auf lokale Histogramme.

Die Intensitätsverteilung von Mikroarraybildern ist im Wesentlichen *unimodal*, weil der Bildhintergrund in der Regel den größten Teil der Bildfläche aber nur einen relativ kleinen Intensitätswertebereich einnimmt. Das Nutzsignal der Messpunkte ist über

[2] Dank an die Veranstalter und Teilnehmer des Seminars „Anwendungsorientierte Bildverarbeitung“

große Wertebereiche gestreut, in denen auch Störsignale von verunreinigenden Partikeln liegen.

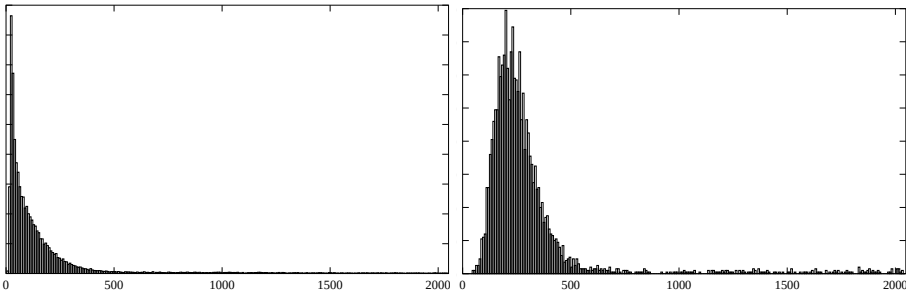



Abbildung 5.4: Beispiele von Histogrammen von Mikroarraybildern, berechnet auf Bildausschnitten von 200x200 Pixeln berechnet. Der rechte Verteilungsrand ist weggelassen. Links ist das Histogramm des Grünkanals gezeigt, rechts das Histogramm der relativen Intensitätsmaxima im Grünkanal (gewichtetes Histogramm)

### 5.2.2 Gewichtete Histogramme

Das wesentliche Hindernis bei der Schwellwertberechnung ist die wenig einheitliche Verteilung des Hintergrundsignals. Es treten verschiedene Effekte auf:

- Körnige und homogene Verunreinigungen auf der Arrayoberfläche geben Fluoreszenzlicht ab und erzeugen so ein strikt positives Störsignal.
- Bei der Bildaufnahme mit dem Scannermikroskop tritt ein Verwischungseffekt auf, der bei Vorhandensein von punktartigen Verunreinigungen zwangsweise eine unsymmetrische Intensitätsverteilung erzeugt [58].
- Wenn die Fluoreszenzintensität des Hintergrundes so gering ist, dass der Detektor des Bildaufnahmegerätes Photonen überwiegend als Einzelereignisse detektiert, ist die Intensität poissonverteilt. Die Poissonverteilung ist für sehr kleine Ereignisraten von Natur aus unsymmetrisch und geht bei höheren Photonenraten in die Binomial- bzw. Gaussverteilung über.

Durch Verwendung von *gewichteten Histogrammen* wird die Schwellwertberechnung robuster gegenüber den oben beschriebenen komplizierten Verhältnissen, denn zur Vermeidung von Störregionen müssen insbesondere lokale Intensitätsmaxima betrachtet werden.

Das gewichtete Histogramm $h$ eines $M \times N$ Pixel großen Bildes $f$ mit einer Intensitätswertemenge $C$ wird definiert durch

$$h(c) = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} w(i,j)\delta(f(i,j) - c) \quad \forall c \in C$$

Die Gewichtungsfunktion $w$ wird so definiert, dass nur die Intensitätswerte von Pixeln, die lokale Intensitätsmaxima darstellen, im Histogramm registriert werden:

$$w(i,j) = \begin{cases} 1 & : \quad f(i,j) \geq f(k,l) \forall (k,l) \in \mathcal{N}_8(i,j), (k,l) \neq (i,j) \\ 0 & : \quad sonst \end{cases}$$

Wegen der $\geq$-Relation werden auch Intensitätsplateaus registriert, damit keine komplizierte Sonderbehandlung von konstanten Bildbereichen nötig ist. Die Abbildung 5.4 zeigt an einem Beispiel die Wirkung der Gewichtung.

### 5.2.3 Verfahren zur Berechnung von Intensitätsschwellwerten

#### Quantile

Eine besonders einfache Möglichkeit zur Schwellwertberechnung aus Intensitätsverteilungen sind *Quantile*, also Intensitätswerte, unterhalb derer ein bestimmter Anteil der im Histogramm registrierten Ereignisse liegt. Der Median ist mit dem 50%-Quantil identisch.

Quantilbasierte Schwellwertberechnung hat sich als wenig robust gegen variablen Anteil von Störsignal erwiesen, d.h. es gibt keinen eindeutigen Quantilparameter, der immer eine brauchbare Segmentierung liefert.

#### Die Methode von Otsu

Das Verfahren von Otsu [82] nimmt eine Mischung von zwei Gaussverteilungen als Modell für das Histogramm an. Die beiden Gaussglocken beschreiben die Intensitätsverteilungen der Objekte von zwei zu trennenden Klassen, so dass sich leicht der optimal trennende Schwellwert berechnen lässt. Es brauchen keine weiteren Parameter eingestellt werden.

Mit dem Otsu-Verfahren berechnete Schwellwerte führen meistens zu sehr vielen Störregionen, weil die Verteilung der Hintergrundintensität oft nicht symmetrisch ist. Die typischen Histogramme von Mikroarraybildern entsprechen der Verteilungsannahme nicht sehr gut, so dass das Verfahren teilweise das Zentrum der zweiten Gaussglocke nicht in den Bereich der Signalintensitäten, sondern in den Fuß des Hintergrundmaximums setzt.

Robuste Schwellwerte ergibt das Otsu-Verfahren nur dann, wenn die Modellannahmen erfüllt sind, d. h. wenn die Intensitäten der beiden Objektklassen durch nicht zu stark überlappende Gaussverteilungen zu beschreiben sind.

#### Einseitige Varianzschätzung

Wenn man ein symmetrisch verteiltes Hintergrundsignal annimmt, kann links vom Histogrammaximum die Varianz der Hintergrundkomponente der Bildintensität geschätzt werden. Der Ort des Histogrammmaximums (oder der *Modus*) ist weitgehend vom Hintergrundanteil bestimmt, weil das Signal der Messpunkte über erheblich größere Wertebereiche gestreut ist. Anhand der geschätzen Breite des Hintergrundmaximums kann ein Schwellwert rechts vom Maximum angegeben werden, der Hintergrund und Messpunktsignal unter den Annahmen gut trennt. Nimmt man gaussverteilte

Hintergrundintensität an, wird die Varianz $\sigma^2$ des Hintergrundmaximums durch

$$\overline{\sigma^2} = \frac{1}{M} \sum_{i \leq i_{\text{modus}}} h(i)(i - i_{\text{modus}})^2 \tag{5.1}$$

$$M = \sum_{i \leq i_{\text{modus}}} h(i) \tag{5.2}$$

im Intensitätshistogramm $h(i)$ geschätzt. Der Schwellwert ist dann $i_{\text{modus}} + k\sigma$, wobei $k$ sinnvoll zwischen 1 und 3 zu wählen ist.

Für einige Arraybilder führt dieses Verfahren zu guten Ergebnissen, während es bei stärker unsymmetrischer Verteilung der Hintergrundintensität zu kleine Schwellwerte berechnet.

## 5.3 Zusammenfassung

Keines der aufgeführten Schwellwertverfahren erreicht eine robuste Segmentierung von Bildern unterschiedlich hergestellter Mikroarrays. Offenbar treffen die impliziten oder expliziten Verteilungsannahmen der Schwellwertberechnungsverfahren nicht allgemein zu. Für die Verfahren mit freien Parametern findet man aber so gut wie immer eine Einstellung, die zu einer guten Regionensegmentierung führt. Im nächsten Kapitel werden Methoden behandelt, die mit Hilfe des Kriteriums der Erhaltung der Gitterstruktur im Regionenbild ohne Benutzereingriff gute Parameter finden.

# 6 Gittersegmentierung

Dieses Kapitel zeigt, wie man durch Ausnutzung der Gitteranordnung der Messpunkte die robuste Regionensegmentierung erreicht. Aus den Regionenbildern werden Hypothesen für die Segmentierung der Messpunktgitter mit dem MZF-Modell erzeugt, mit dem sich der letzte Teil des Kapitels beschäftigt.

## 6.1 Nächste-Nachbar-Abstände und Regionenklassifikation

Gute Regionensegmentierungen zeichnen sich dadurch aus, dass sie viele periodisch angeordnete Messpunktregionen enthalten. Im folgenden Abschnitt wird ein Verteilungsmodell für die Abstände der Regionen eines Mikroarraybildes zu ihren $k$ nächsten Nachbarn entwickelt, mit dem Regionenbilder entsprechend bewertet werden können. Es müssen die Abstände in kleinen Nachbarschaften betrachtet werden, weil die Periodizität nur innerhalb der einzelnen Gitter, nicht aber im gesamten Bild gilt.

### 6.1.1 Verteilungsmodelle für Regionenabstände

Zunächst werden die Verteilungen der euklidischen Abstände der gesuchten Messpunktregionen betrachtet.

Im einfachsten, idealisierten Fall enthält das Bild ein ungestörtes, unendlich ausgedehntes und voll besetztes quadratisches Gitter von Regionen. Darin treten als Abstände zum $k$-nächsten Nachbarn der Zeilen- und Spaltenabstand $g$ (bei $k = 1, 2, 3, 4$) und die weiteren, für das Gitter charakteristischen Abstände $\sqrt{2}g$ (bei $k = 5, 6, 7, 8$), $2g$, $\sqrt{5}g$ usw. auf (Siehe Abbildung 6.1). Die Verteilungsdichten der Abstände zum $k$-nächsten Nachbarn in einem solchen idealen Gitter sind daher einzelne scharfe Peaks.

Rechteckgitter mit verschiedenen Zeilen- und Spaltenabständen $g_v$ und $g_h$ behandelt man durch Auftrennung des Nachbarschaftssystems des quadratischen Gitters in zwei Teile, die horizontalen und vertikalen Nachbarn. Die Abbildung 6.1 veranschaulicht die folgende Definition der Nachbarschaftssysteme: Die Entfernung eines Regionenschwerpunktes $\vec{c}$ zu seinen *horizontalen* Nachbarn ist entlang der horizontalen Achse größer oder gleich der Entfernung entlang der vertikalen Achse; die horizontalen Nachbarn einer Region im Ursprung liegen also zwischen den Diagonalen des Koordinatensystems. Entsprechendes gilt für die vertikalen Nachbarn. Die charakteristischen Peaks der getrennt für die horizontalen bzw. vertikalen Nachbarn berechneten Abstandshistogramme liegen dann bei $g_h, 2g_h$ (hor.) und bei $g_v, \sqrt{g_h^2 + g_v^2}, 2g_h$ (vert.) wenn $g_h < g_v$ gilt und andernfalls bei $g_h, \sqrt{g_h^2 + g_v^2}, 2g_h$ (hor.) und $g_v, 2g_h$ (vert.).

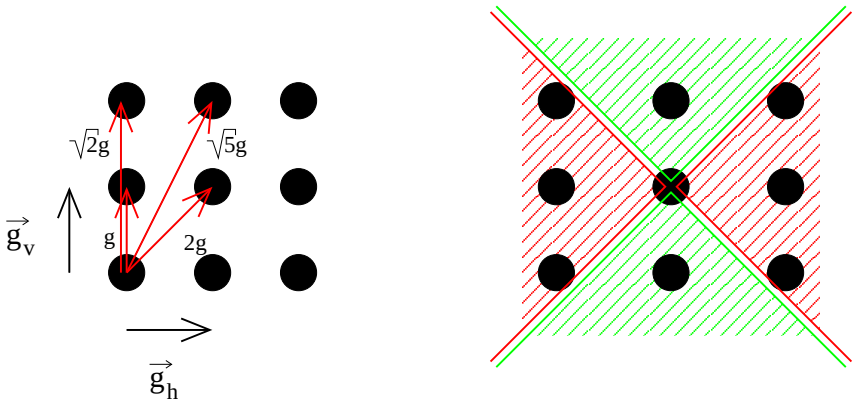



Abbildung 6.1: Die linke Zeichnung zeigt die Gittervektoren $\vec{g}_h$ und $\vec{g}_v$ sowie die charakteristischen Abstände der nächsten Nachbarn der Knoten eines quadratischen Gitters. Die Zeichnung rechts zeigt die Nachbarschaftssysteme für das Rechteckgitter. Die horizontalen Nachbarn des zentralen Gitterknotens liegen im rot schraffierten Bereich, die vertikalen Nachbarn im grünen.

Sind die Positionen der Gitterknoten mit unabhängigen, normalverteilten Störungen $\eta$ verrauscht, so werden auch die beobachteten Abstandswerte verrauscht. Die Verteilung des Abstandes einer Region zur nächsten horizontalen Nachbarregion

$$P(|d| < r)$$

ist näherungsweise normalverteilt mit den Parametern $(g-\sigma, \sigma)$ (siehe Anhang B.2). Die Abstände zum zweitnächsten Nachbarn sind ebenfalls ungefähr normalverteilt mit $(g + \sigma, \sigma)$.

Wenn die Gitterknoten nicht zu 100% mit Regionen belegt sind, treten schon bei $k \leq 2$ Abstände größer als eine Gitterzelle auf. In den Abstandsverteilungen der $k$-nächsten Nachbarn gibt es dann für quadratische Gitter mehrere Peaks bei $g$, $\sqrt{2}g$, $2g$ usw. gleichzeitig, während bei voller Gitterbelegung bei jedem $k$ nur ein einziger Peak vorkommen kann.

Hexagonale Gitter erzeugen andere charakteristische Peaks, können aber im Prinzip genau so wie Rechteckgitter behandelt werden. Mangels entsprechender Bilddaten wird dieser Fall nicht weiter behandelt.

Bis hierher werden noch die nicht zum Gitter gehörenden Störregionen vernachlässigt. Wenn sich solche Regionen zwischen den Gitterknoten befinden, nehmen diese die Position der ersten Nachbarn der umliegenden Gitterknoten ein, so dass die Ordnung $k$ der Abstände zwischen den erwünschten Regionen an den Gitterknoten größer wird. In den Verteilungen treten also mehrere charakteristische Peaks und die zufälligen Abstände der Störregionen auf. Die Abbildung 6.2 zeigt Histogramme der Abstände zum $k$-nächsten horizontalen Nachbarn für vier alternative Segmentierungen eines Mikroarraybildes, in denen die verschiedenen Effekte zu erkennen sind: Es gibt die charakteristischen Peaks des Gitters ungefähr bei den Abständen 27, 38, 54 und 60 (der Gitterspaltenabstand ist 27 Pixel). Man erkennt, dass der erste Peak im Histogramm der Abstände zum erstnächsten Nachbarn etwas nach unten verschoben ist, die

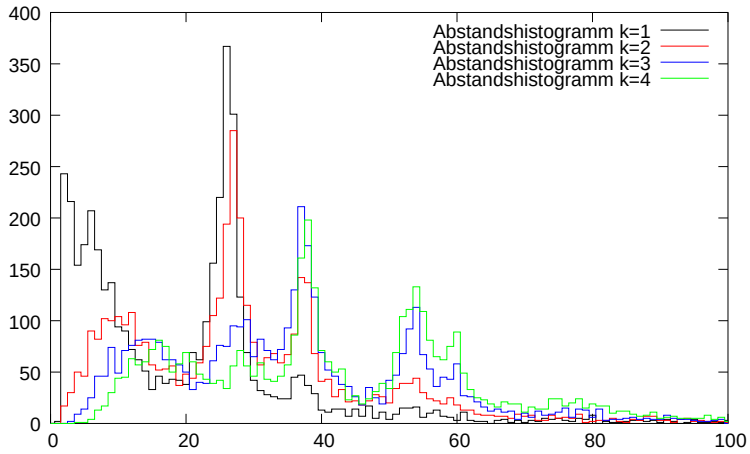



Abbildung 6.2: Histogramme der Abstände zu den $k$-nächsten Nachbarn der Regionen des Mikroarraybildes *nmhy1x103*

Verschiebung nach oben aber entgegen der Vorhersage des Modells nicht eintritt. Diese Eigenschaft lässt sich an den Abstandsverteilungen der Regionensegmentierungen der meisten Arraybilder beobachten und könnte durch Randeffekte begründet sein, die in der Herleitung des Modells nicht berücksichtigt sind.

Die breiten Maxima am unteren Ende der Abstandshistogramme in Abbildung 6.2 werden durch Störregionen außerhalb der Gitter verursacht. Unter den Annahmen, dass die Störregionen überall gleich wahrscheinlich und unabhängig voneinander auftreten, hat die Verteilung der Abstände zum $k$-nächsten Nachbarn die Form

$$p_{\lambda,k}(r) = 2\frac{(\lambda\pi)^k}{(k-1)!}r^{(2k-1)}e^{(-\lambda\pi r^2)} \tag{6.1}$$

(zur Herleitung siehe Anhang B.1). Die Parameter $k$ und $\lambda$ sind die Nachbarordnung und die Dichte der Störregionen, also die Zahl der Störregionen pro Fläche. Die Dichte ist in Wirklichkeit nicht konstant, weshalb die modellierte Verteilung geringere Varianz als die beobachteten Abstände hat. Das Modell umfasst auch nicht die Störregionen innerhalb des Gitters, deren Abstände zum nächsten Nachbarn durch die Regionen an den Gitterknoten nach oben begrenzt sind. Dennoch treten die charakteristischen Peaks in der Regel viel deutlicher hervor, wenn ein heuristischer Rauschanteil nach Gleichung (6.1) von den beobachteten Histogrammen abgezogen wird. Die Abbildung 6.3[1] zeigt die Wirksamkeit an einem extrem verrauschten Mikroarray-Bild.

Die Peaks der empirischen Abstandsverteilungen enthalten also wesentliche Information über die zunächst unbekannten Gitterabstände $g_h$ und $g_v$. Die Störregionen und Randeffekte bewirken, dass die gesuchten charakteristischen Abstände nicht bei den im Fall des idealen Gitters erwarteten $k$ auftreten, weshalb im Folgenden die addierten Abstandshistogramme aller $k$-nächsten Nachbarn betrachtet werden. Dadurch wird auch die Robustheit bei Verarbeitung von Arrays mit wenigen Messpunkten oder kleinen Gittern verbessert.

[1] Der freundliche Spender des Bildes möchte anonym bleiben.

### 6.1.2 Schätzung der Gitterabstände

Die summierten empirischen Abstandsverteilungen bis zur Ordnung $k$ werden durch eine Mischung $f_k$ von Gaussverteilungen modelliert, denn wie oben erwähnt können die Einzelpeaks näherungsweise durch Gaussfunktionen dargestellt werden. Die Orte der Peaks sind entsprechend der charakteristischen Vielfachen der Gitterabstände $h = (1, \sqrt{2}, 2, \sqrt{5}, \ldots)$ gekoppelt:

$$f_k(r) \quad = \quad \sum_{j=0}^{k} c_j e^{-\frac{(r-(gh_j+\frac{k-j}{k}s_j))^2}{s_j^2}} \tag{6.2}$$

Die Parameter $s_j$ geben die Breite der Peaks an. Der heuristische Term $\frac{k-j}{k}s_j$ im Argument der Exponentialfunktion berücksichtigt die kleinen Verschiebungen der Peaks bei verschiedenen Nachbarordnungen $k$, die man an Abbildung 6.2 beobachten kann.

Das Mischungsmodell $f_k$ wird mit dem Levenberg-Marquard-Algorithmus [72, 76], einem schnellen lokalen Verfahren zur Optimierung von Parametern unter der quadratischen Fehlerfunktion, an die Daten angepasst. Wegen der lokalen Optimierung wird die Parameterschätzung mehrfach mit jedem relativen Maximum des Histogramms als $g$-Startwert durchgeführt, um die global optimalen Parameter finden zu können. Das $g$ aus dem Parametersatz mit dem geringsten Restfehler bildet den Schätzwert für die Gitterabstände.

Die Anpassung wird getrennt für die Abstände im horizontalen und vertikalen Nachbarschaftssystem durchgeführt. Wenn ein Rechteckgitter vorliegt, passt die Kopplung der Peaks im Modell ($h$) nur bei den achsenparallelen charakteristischen Abständen von ganzzahligen Vielfachen der Gittervektorlängen. Die Parameteroptimierung stellt dann kleine Gewichtungsfaktoren $c_j$ für die Peaks nicht achsenparalleler Abstände ein.

Die oben beschriebene heuristische Korrektur für die Abstände von Störregionen verbessert die Robustheit der Parameteroptimierung, weil die Histogramme nach der Korrektur besser dem anzupassenden Modell entsprechen.

Die Höhe des ersten Verteilungspeaks dient als Bewertungskriterium für die Regionensegmentierung. Aus mehreren alternativen Segmentierungen mit dem Quantil-Schwellwertverfahren bei verschiedenen Parameterwerten kann damit ein Regionenbild ausgewählt werden, das besonders viele periodisch angeordnete, sehr wahrscheinlich Messpunkte darstellende Regionen enthält.

### 6.1.3 Klassifikation und Kanalkorrespondenz von Regionen

Die Regionensegmentierung erzeugt getrennte Regionenbilder für jeden Bildkanal. Nicht alle Regionen, die man in diesen Bildern findet, gehören zu (genau) einem Messpunkt, denn Verunreinigungen oder Kratzer auf den Arrays erzeugen oft sehr große Regionen oder Bildbereiche mit dicht gestreuten Störregionen. Durch Verschmelzen dicht benachbarter Regionen und anschließendes Entfernen derjenigen, die nicht in eine Gitterzelle passen, werden viele Störungen unterdrückt.

Vereinzelt gibt es auch stark deformierte Messpunkte, die in ein und dem selben Bildkanal mehrere Regionen erzeugen.

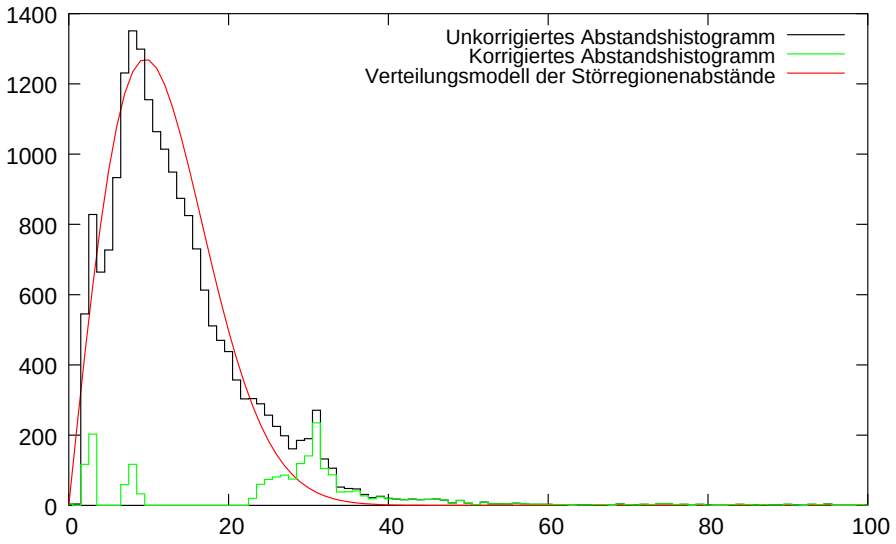



Abbildung 6.3: Die Korrektur des Abstandshistogramms mit dem Modell für Abstände zufällig verteilter Regionen (Gl. (6.1)) am Beispiel einer äußerst stark verrauschten Regionensegmentierung. Der Peak beim gesuchten Gitterabstand von 31 Pixeln hat hier nach der Korrektur den höchsten Wert im Histogramm.

Für die weitere Verarbeitung ist es zweckmäßig, alle Regionen die zum gleichen Messpunkt gehören zusammenzufassen, damit später die Zuordnung zu den logischen Gitterknoten leicht möglich ist.

Der untenstehende Algorithmus erzeugt *Objekte*, die mehrere Regionen zusammenfassen die zu einem einzigen Messpunkt gehören. Gleichzeitig werden dabei wie oben beschrieben zu große Regionen entfernt. Objekte können sowohl dicht benachbarte Regionen aus dem gleichen Kanal enthalten (z. B. wenn Messpunkte durch Kratzer geteilt sind) als auch Regionen aus verschiedenen Kanälen, die zum gleichen Messpunkt gehören. Man könnte das Vorgehen auch als mehrfache Clusterung von Regionen in den einzelnen Kanälen und Clusterzentren verschiedener Kanäle formulieren.

**Algorithmus 1** Erzeugung von Messpunktobjekten

- Erzeuge ein Objekt aus jeder Region.

**repeat**

- Markiere Objekte, die größer als eine Gitterzelle sind.
- Verschmelze Objekte aus dem gleichen Kanal, deren Regionenränder weniger als 30% der Gitterzellenausdehnung voneinander entfernt sind (Markierungen werden vererbt).

**until** keine neuen Markierungen oder Verschmelzungen

- Entferne alle markierten Objekte.
- Verschmelze Objekte aus verschiedenen Kanälen, deren Schwerpunkte weniger als 50% der Gitterzellenausdehnung entfernt sind.

An dem Beispiel in Abbildung 6.4 erkennt man, dass der Algorithmus auch einige zu Messpunkten gehörende Regionen entfernt, die dicht an Störregionen liegen.

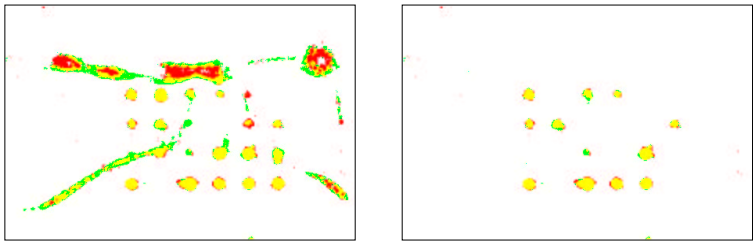


Abbildung 6.4: Die Erzeugung von Messpunktobjekten an einem Beispielbild. Links sind die überlagerten Regionenbilder zu sehen, rechts die nach Anwendung von Algorithmus 1 in Messpunktobjekten zusammengefassten Regionen.

Objekte, die Regionen aus mehreren Kanälen enthalten, dienen zur Schätzung der Verschiebung zwischen den einzelnen Fluoreszenzaufnahmen. Sie liefern eine Stichprobe von Verschiebungen zwischen den Regionenschwerpunkten der verschiedenen Kanäle, die später lokal je Gitter gemittelt zur Korrektur der Verschiebung zwischen den Grauwertkanälen benutzt werden.

### 6.1.4 Verkettung von Objekten

Der nächste Verarbeitungsschritt besteht in der zweidimensionalen Verkettung von Messpunktobjekten entsprechend der zuvor geschätzten Gitterabstände. Dadurch entsteht eine abstraktere Ordnung auf dem Bild der Messpunktobjekte, die die Rotation der Gitter erkennen läßt und ein diskretes Raster für die möglichen Gitterpositionierungen festlegt.

Die Anpassung des Verteilungsmodells an die kNN-Abstandshistogramme schätzt neben der Gitterkonstante selbst auch die Breite $\sqrt{s}$ des zugehörigen Peaks in der Abstandsverteilung. Zusätzlich liefert die Betrachtung der Abstands*vektoren* von Regionenpaaren, deren Abstand im 1. Modellpeak der kNN-Abstandsverteilung liegt, eine grobe Schätzung der Gitterachsen $\vec{g}_h$ und $\vec{g}_v$.

Der Algorithmus 2 verkettet mit diesen Voraussetzungen Objekte, die mit hoher Wahrscheinlichkeit zum Messpunktgitter gehören, in einer 4er-Nachbarschaft, beginnend bei einem Startobjekt mit Schwerpunkt $\vec{c}$:

**Algorithmus 2** Rekursive Objektverkettung

**for** $\vec{v}$ in $\vec{g}_h, -\vec{g}_h, \vec{g}_v, -\vec{g}_v$ **do**
  Suche nächsten Nachbarobjektschwerpunkt $\vec{n}$ zu $\vec{c} + \vec{v}$
  **if** $\|(\vec{c} + \vec{v}) - \vec{n}\| < 3\sqrt{s}$ der Gitterzellengröße UND $\vec{n}$ unverkettet **then**
    Verkette $\vec{c}$ und $\vec{n}$
    RekursiveObjektverkettung($\vec{n}$)
  **end if**
**end for**

Dieser Algorithmus wird an jeder (unverketteten) Region gestartet. Die Abbildung 6.5 zeigt an einem kleinen Beispiel den Ablauf und das Ergebnis des Verfahrens. In Abbildung 6.10 ist u.a. die Verkettung für ein größeres Bild zu sehen.

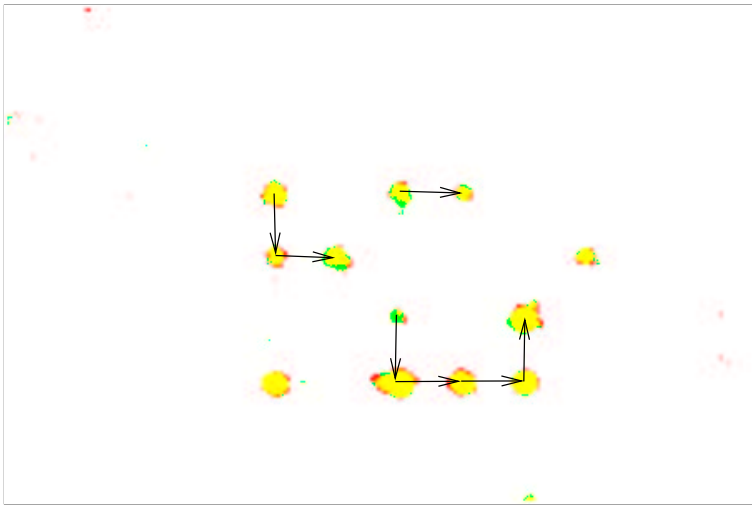


Abbildung 6.5: Die Abbildung zeigt den Ablauf der Verkettung der Messpunktobjekte eines kleinen Beispielbildes. Die Pfeile geben die Abfolge der Verkettung an.

Die so verketteten Objekte gehören mit hoher Wahrscheinlichkeit zum Messpunktgitter, weil bei der Fortsetzung des Gitters nur Regionen in einem kleinen Kreis um die extrapolierten Gitterpositionen akzeptiert werden. Weil die Regionensegmentierung so ausgewählt wird, dass möglichst viele periodisch angeordnete Regionen darin enthalten sind, liegen mit hoher Wahrscheinlichkeit ($P$) benachbarte Messpunktregionen in dem Kreis. Die verbleibenden Störregionen sind nicht mit dem Gitter korreliert und liegen daher mit kleinerer Wahrscheinlichkeit $p$ in dem Kreis. Grob abgeschätzt ist die Wahrscheinlichkeit, dass $n$ verkettete Objekte zum Messpunktgitter gehören $P^n$ und $p^n$ für den gegenteiligen Fall, d.h. die Zuverlässigkeit eines Verkettungsergebnisses steigt exponentiell mit seiner Größe.

Daher können aus den Abstandsvektoren der Objektschwerpunkte der größten Gitterfragmente (z. B. 50%) die Richtung der Gitterachsen sehr robust geschätzt werden.

### 6.1.5 Effiziente Nachbarsuche

Für die Suche nach den $k$-nächsten Nachbarn aller Punkte oder eines Anfragepunktes (z. B. dem extrapolierten Gitterort $\vec{c} + \vec{v}$ im Verkettungsalgorithmus) in einer Menge von Regionenschwerpunkten bietet sich die Speicherung der Schwerpunkte in kd-Bäumen an.

Die kd-Bäume sind sortierte Binärbäume, deren Knoten eine Menge $d$-dimensionaler Punkte alternierend nach den Dimensionen in möglichst gleiche Teile zerlegen [84]. Bei der Baumkonstruktion in zwei Dimensionen ist der Wurzelknoten ein Punkt, der in der Mitte der nach der x-Koordinate sortierten Punkte stehen würde. Die zwei Kindknoten des Wurzelknotens teilen die Punktmengen rechts und links von der Grenzlinie des Wurzelknotens nach den y-Koordinaten und deren Nachkommen teilen wiederum nach der x-Koordinate usw. Der kd-Baum stellt eine rekursive Teilung einer Punktmenge dar. Die Abbildung 6.6 zeigt ein Beispiel für die Baumkonstruktion auf einer kleinen Punktmenge

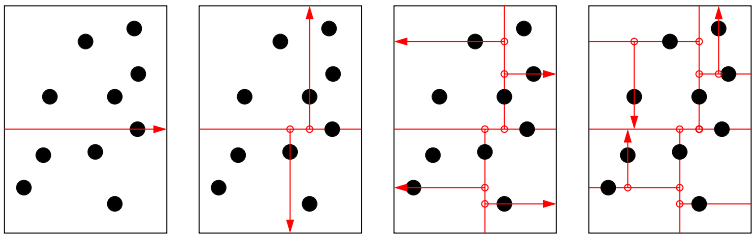


Abbildung 6.6: Die Konstruktion eines kd-Baums zur Repräsentation von 9 Punkten (schwarz) in der Ebene. Jedes Teilbild zeigt die Erzeugung einer neuen Hierarchieebene des Baumes. Beginnend vom Wurzelknoten, der die gesamte Punktmenge repräsentiert werden die Daten abwechselnd nach x- und y-Koordinate möglichst gleichmässig (zufällige Auswahl aus den beiden mittleren Elementen bei gerader Anzahl) geteilt.

Die Konstruktion eines kd-Baumes von $n$ Punkten benötigt $O(n \log n)$ Koordinatenvergleiche, weil in jeder der $\log n$ Ebenen des Baumes die Punktlisten neu geteilt werden müssen. Für die asymptotische Effizienz spielt dabei keine Rolle, ob die Punkte vorab sortiert werden oder tatsächlich erst während der Baumkonstruktion umgeordnet werden [9].

Zur Suche nach einem gegebenen Anfragepunkt steigt man rekursiv vom Wurzelknoten zu dem Kindknoten ab, dessen Unterbaum die Anfragekoordinaten enthält. Der Anfragepunkt kann so in $O(\log n)$ Schritten gefunden werden, wenn er im Baum enthalten ist.

Arya, Mount und andere haben eine effiziente Methode zur Suche der $k$ nächsten Nachbarn eines Anfragepunktes im kd-Baum beschrieben [6]. Der Algorithmus führt beim Traversieren des Baumes die bereits besuchten Punkte, die am nächsten am Anfragepunkt liegen, in einer Prioritätswarteschlange mit. Die Punkte in der Warteschlange definieren ein Suchfenster, in dem die Traversierung fokussiert wird. Die Abbildung 6.7 skizziert diesen Suchprozess für $k = 1$ an dem obigen Beispielbaum. Die Autoren zeigen, dass damit die Suche nach den $k$-nächsten Nachbarn des Anfragepunktes in zwei Dimensionen $O(k \log n)$ Schritte benötigt. Die $k$-nächsten Nachbarn aller Regionenschwerpunkte können also in $O(kn \log n)$ Schritten bestimmt werden.

Der Algorithmus von Arya und Mount kann auch mit den getrennten horizontalen und vertikalen Nachbarschaftssystemen umgehen, wenn nur solche Punkte in die Warteschlange eingetragen werden, die die Bedingungen für horizontale bzw. vertikale Nachbarn erfüllen. Auch Rechteckgitter können somit effizient behandelt werden.

## 6.2 Achsenprojektionen und deren Interpretation

Für die Segmentierung der Gesamtanordnung der Gitter werden Hypothesen für die Lage jedes einzelnen Gitters benötigt. Daher werden Projektionen der verketteten Objekte auf die Gitterachsen bestimmt. Die Störregionen werden durch die Verkettung gut unterdrückt, so dass die Projektionen Blockmuster bilden, die in das zweidimensionale Bild zurückprojiziert werden können. Dazu müssen die Blockränder in den auf den Achsen festgelegt werden, weshalb Sequenzen von Blöcken und Lücken bestimmt

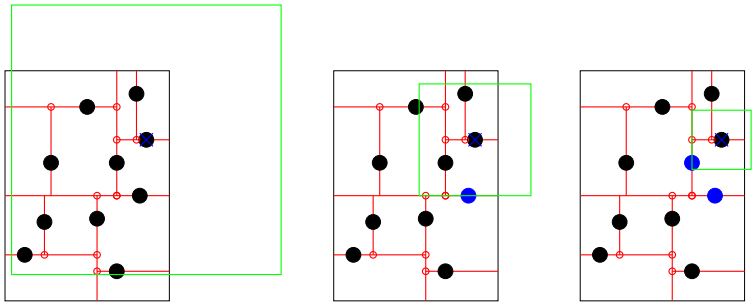


Abbildung 6.7: Die Suche des nächsten Nachbarn zu einem Anfragepunkt (blau durchkreuzt) im oben konstruierten Beispielbaum. Der bei fortschreitender Baumtraversierung schrumpfende Suchbereich ist durch das grüne Quadrat dargestellt. Die darin enthaltenen Punkte sind die jeweils verbleibenden Kandidaten für den nächsten Nachbarpunkt. Die bereits besuchten Datenpunkte sind blau eingefärbt.

werden, die die Projektion optimal zerlegen.

Bei vorgegebener Anzahl von Gittern ist die Sequenzlänge fest, so dass die Block-Segmentierung als gewöhnliches Alignment-Problem behandelt werden kann. Bei unbekannter Gitteranzahl stellt sich ein ähnliches Problem wie beim Alignment von genomischen Sequenzen mit ESTs (Expressed Sequence Tags, Repräsentanten der gereiften mRNA-Sequenzen von Genen) zur Bestimmung der Exon-Intron-Struktur von Genen („spliced alignment“ [83]), bei dem die optimale Zahl von Exons bzw. Blöcken implizit mitbestimmt werden muss. Beide Probleme werden effizient durch dynamische Programmierung gelöst.

Die Ausdehnung eines Gitters im Bild bzw. des zugehörigen Blocks in der Projektion ist bekannt, denn die Anzahlen von Spalten und Zeilen der Gitter gehören zu den Eingabeparametern des Systems. Mit den Schätzungen der Messpunktabstände und der Orientierung der Gitter aus den vorangegangenen Verfahrensschritten können daher die Größen der Gitter und somit auch ihre Länge in der Achsenprojektion ausgerechnet werden.

Die Abbildung 6.8 oben zeigt ein Beispiel einer Achsenprojektion von verketteten Messpunktobjekten. Um die Kostenfunktion für das Alignment übersichtlich zu halten, kappt man die Projektion bei dem rot eingezeichneten Schwellwert, der einer halben Gitterausdehnung senkrecht zur Projektionsachse entspricht und normiert die Projektion anschließend auf den Wertebereich $[0, 1]$.

### 6.2.1 Dynamische Programmierung

Zur Zerlegung der Projektion in eine optimale Abfolge von Blöcken und Lücken führt man Block-Hypothesen $H_i$ ein, die durch Intervalle $[H_i^a, H_i^e[$ der Länge $G$ (Gitterkantenlänge) dargestellt werden. Außerdem wird die Kostenfunktion des Alignments eingeführt , die innerhalb von Blöcken die Projektionswerte zählt, die kleiner als eine halbe Gitterausdehnung sind und zwischen Blöcken die Projektionswerte, die über diesem Wert liegen. Die Kostenfunktionen werden in den Gleichungen (6.5) und (6.4) mit der oben beschriebenen, normierten Achsenprojektion $h(j)$ ($j \in$ Zeilen- bzw. Spalten-

indizes der Bildpixel) formuliert.

Als Hypothesen betrachtet man entweder alle möglichen Blöcke auf der Projektionsachse oder man beschränkt (zur besseren Übersicht) die Auswahl auf „vielversprechende“ Hypothesen, die an steigenden Flanken der Projektion $h$ beginnen oder an fallenden Flanken enden. Die Abbildung 6.8 zeigt eine Projektion und eine solche Hypothesenmenge (blau).

In der optimalen Blocksequenz dürfen niemals überlappende Blöcke vorkommen, die Anfangskoordinaten der Blöcke sind immer aufsteigend. Mit der additiven Zerlegbarkeit der Kostenfunktion folgt daraus das Optimalitätsprinzip, das die Anwendung der dynamischen Programmierung erlaubt. Die Gleichung (6.6) gibt die Rekursionsformel für die Konstruktion einer optimalen Folge fester Länge an. Darin sind die $C_i^j$ die auf der Block-Hypothese $H_i$ endenden Blocksequenzen der Länge $j$, an die die jeweils besten weiteren Hypothesen angehängt werden. Die Fallunterscheidung in der Kostenfunktion (6.4) bewirkt, dass nie überlappende Blöcke in die Lösung aufgenommen werden.

$$K(C) = \sum_{H_i \in C} K_l(H_{i-1}^e, H_i^a) + K_b(H_i^a, H_i^e) \tag{6.3}$$

$$K_l(u,v) = \begin{cases} \sum_{j=u}^{v-1} h(j) & : \quad u < v \\ \infty & : \quad \text{sonst} \end{cases} \tag{6.4}$$

$$K_b(u,v) = \sum_{j=u}^{v-1} 1 - h(j) \tag{6.5}$$

Zur einfacheren Definition der Kostenfunktion werden zwei technische Blockhypothesen $H_{-1}$ und $H_0$ eingeführt, die außerhalb des Bildes liegen und die Länge $0$ haben.

$$\begin{aligned} C_i^j &= H_i \circ C_l^{j-1} \\ C_i^0 &= H_0 \\ l &= \arg \min_{0 \le k \le N} \left\{ K_b(H_i^a, H_i^e) + K_l(H_i^e, C_k^{j-1}) + K(C_k^{j-1}) \right\} \end{aligned} \tag{6.6}$$

Die Abbildung 6.8 zeigt unten den Ablauf der Rekursion als Zustandsdiagramm. Die Spalten in der Matrix gehören zu den Blockhypothesen, wogegen die Zeilen den Elementen der Zielsequenz zugeordnet sind. Die Knoten in dem Diagramm repräsentieren die $C_i^j$, also die optimalen, auf der Hypothese $H_i$ endenden Sequenzen der Länge $j$. Die gesuchte Lösung verfolgt man am Ende der Rekursion von der rechten unteren Ecke des Diagramms aus, die dem Zustand entspricht, in dem Ziel- und Eingabesequenz ganz abgearbeitet sind. Aus dem Diagramm liest man sofort ab, das die Speichereffizienz des Verfahrens $O(MN)$ und die Laufzeiteffizienz $O(MN^2)$ ist ($N$ Hypothesen, $M$ Blöcke in der Zielsequenz) .

Die Variante ohne vorgegebene Blockanzahl unterscheidet sich darin, dass beim Hinzufügen einer neuen Hypothese nicht nur Teillösungen mit genau einer Länge betrachtet werden dürfen, sondern alle Teillösungen, die links von der neu hinzuzufügenden Hypothese enden.

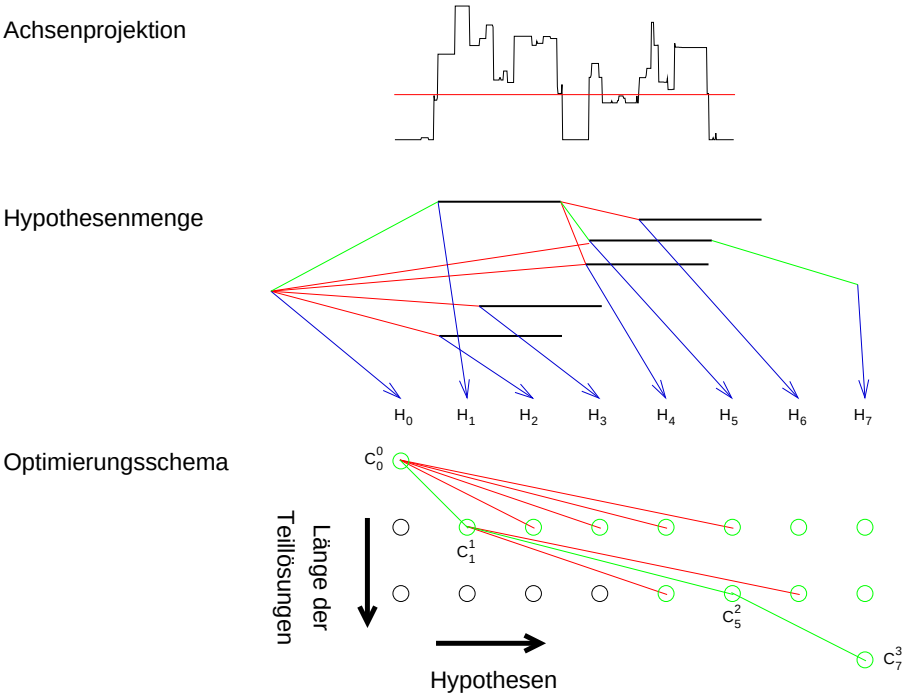



Abbildung 6.8: Achsenprojektion von verketteten Regionen, Gitterblock-Hypothesen und das Schema der Berechnung der optimalen Hypothesenfolge (grüne Linien) mit Gleichung (6.6). Im Schema unten sind die Teillösungen $C_i^j$ bezeichnet, die zur optimalen Lösung gehören. Die Teillösungen bestehen jeweils aus den Knoten entlang des Pfades zurück zur linken oberen Ecke des Diagramms. Die Hypothesen $H_0$ und $H_7$ haben die Länge 0 und dienen zur Vermeidung von Randproblemen. Sie gehören nicht zur eigentlichen Lösung des Alignmentproblems.

Man könnte sich vorstellen, diese Variante des Problems durch Berechnen aller optimalen Blocksequenzen bis zur Länge $N$ mit dem obigen Algorithmus zu lösen, aber dadurch würde erheblicher Speicher- und Rechenzeitaufwand verursacht.

Die bessere Lösung in Form von Gleichung (6.7) ersetzt die Rekursion über die Zielsequenzlänge durch eine Rekursion über die Eingabesequenz. Die Teillösungen $C_i$ sind hier die optimalen Folgen, die auf einer der schon abgearbeiteten Hypothesen enden. Dieses Verfahren ist möglich, weil die Hypothesen in der gesuchten Lösung überlappungsfrei und von links nach rechts geordnet erscheinen müssen. Bei anderen Alignment-Problemen in der Mustererkennung, die mit Algorithmen vom Typ der Gl. (6.6) gelöst werden wie z. B. die Suche der optimalen Zustandssequenz eines Hidden-Markov-Modells, gibt es diese Ordnung nicht. Dort können Elemente der Hypothesenmenge in beliebiger Reihenfolge und sogar mehrfach in der optimalen Lösungssequenz vorkommen. Im Diagramm 6.8 zeigt sich die Ordnung daran, dass die roten Linien, die das Anfügen einer Hypothese an eine Teillösung darstellen, immer von links oben nach rechts unten verlaufen.

$$\begin{aligned} C^j &= H_j \circ C^l \\ C^0 &= H_0 \\ l &= \arg \min_{0 \le k < j} \left\{ K_b(H_i^a, H_i^e) + K_l(H_j^e, C^k) + K(C^k) \right\} \end{aligned} \tag{6.7}$$

Die Abbildung 6.9 zeigt die rekursive optimale Verkettung der Blockhypothesen nach Gleichung (6.7). Dieses Verfahren hat die Laufzeiteffizienz $O(N^2)$[2] Es benötigt nur $O(N)$ Speicherplatz, weil anders als bei der Variante mit Längenvorgabe für jede Hypothese nur einmal eine optimale Verkettung berechnet wird. Daher wird je Hypothese nur ein Speicherplatz für die Rückverkettung gebraucht.

Die Kostenfunktionen $K_b$ und $K_l$ berechnet man am besten aus den kumulativen Projektionen $S_b(t) = \sum_{i=H_0^a}^{t} h(i)$ und $S_l(t) = \sum_{i=H_0^a}^{t} 1 - h(i)$, die vorab in Tabellen abgelegt werden. Die Kostenfunktionen für beliebige Intervalle sind dann mit konstantem Aufwand durch $K_b(u, v) = S_b(v) - S_b(u)$ und $K_l(u, v) = S_l(v) - S_l(u)$ zu berechnen.

## 6.3 Gitter-Hypothesen

Aus der Rückprojektion der optimalen Blockzerlegungen der beiden Achsenprojektionen bekommt man viereckige Bildbereiche als Schätzung der Gitterorte. In diesen Vierecken sucht man das größte Gitterfragment und richtet daran durch lineare Regression ein Modellgitter so aus, dass es das Viereck möglichst gut überdeckt. Gitterzeilen- und spaltenweise Verschiebungen dieses Modellgitters bilden die Hypothesenmenge für die Platzierung eines Gitters. Die Abbildung 6.10 zeigt ein Beispiel, in dem nur für eines der Gitter die Hypothesen eingezeichnet sind.

[2] Die Verwendung der folgenden einfache Eigenschaft erlaubt das schnelle Abbrechen der Suche nach der optimalen Teillösung: Wenn es einen Block gibt, dessen Einbau in die Lösung günstiger ist als die Überdeckung seines Intervalls durch eine Lücke, so können alle Lösungen die diesen Block durch eine Lücke ersetzen nicht optimal sein. Damit wird $O(N)$ Laufzeiteffizienz erreicht.

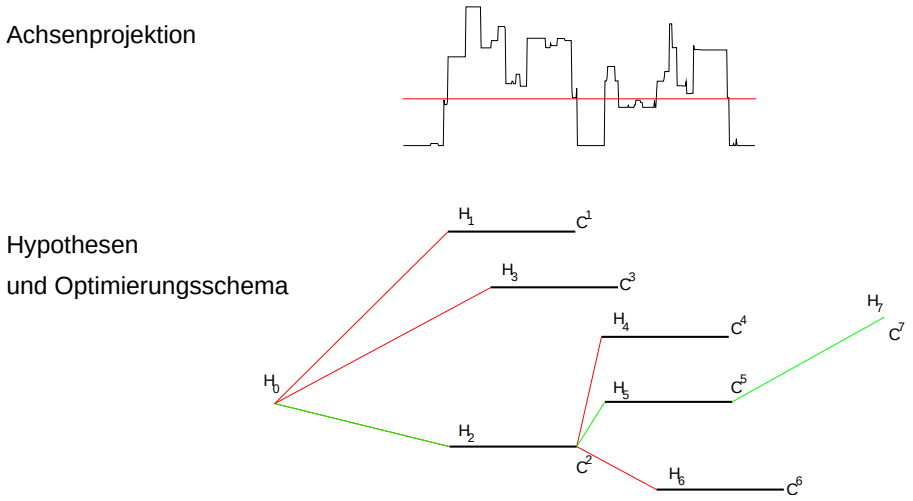



Abbildung 6.9: Optimale Blockzerlegung nach Gleichung (6.7) ohne vorgegebene Länge

## 6.4 Markov-Zufallsfeld-Modell

Für die Gittersegmentierung des Gesamtbildes muss eine geeignete Teilmenge aus der gesamten Hypothesenmenge bestimmt werden. Welche Hypothesen geeignet sind hängt zum einen von der Belegung ihrer Modellgitterknoten mit Messpunktobjekten und zum anderen von der Anordnung benachbarter Gitter ab. Es darf z. B. in aller Regel keine Überlappungen der Gitter geben. Ein geeigneter Ansatz zur Modellierung dieser Abhängigkeiten sind Markov-Zufallsfelder.

### 6.4.1 Grundlagen

Markov-Zufallsfelder beschreiben die Wahrscheinlichkeitsverteilung einer Menge abhängiger Zufallsvariablen. Sie werden in vielen Bildverarbeitungsanwendungen zur Modellierung von Kontextabhängigkeiten eingesetzt. Für die Zwecke dieser Arbeit werden MZFs mit diskreten Zufallsvariablen betrachtet. Die Darstellung der Grundlagen orientiert sich an den Büchern von Li [74] und Kindermann [64].

#### Definition

Sei $\mathcal{S} = \{1, \ldots, K\}$ die Menge der Knoten in einem (ungerichteten) Graphen mit dem Nachbarschaftssystem $\mathcal{N}$. $\mathcal{N}_i$ bezeichnet alle mit Knoten $i$ verbundenen Knoten des Graphen. Seien $F_i \in F, i \in \mathcal{S}$ diskrete Zufallsvariablen und $\mathcal{L}_i$ ihre Wertemengen.

Die Menge der Zufallsvariablen $F$ heißt genau dann *Markov-Zufallsfeld* (Markov Random Field, MZF) wenn die Bedingungen (6.8) und (6.9) erfüllt sind:

$$P(f_i|f_{\mathcal{S}\setminus\{i\}}) = P(f_i|f_{\mathcal{N}_i}) \tag{6.8}$$

$$P(f) > 0, \qquad \forall f \in \mathcal{L}_0 \times \ldots \times \mathcal{L}_K \tag{6.9}$$

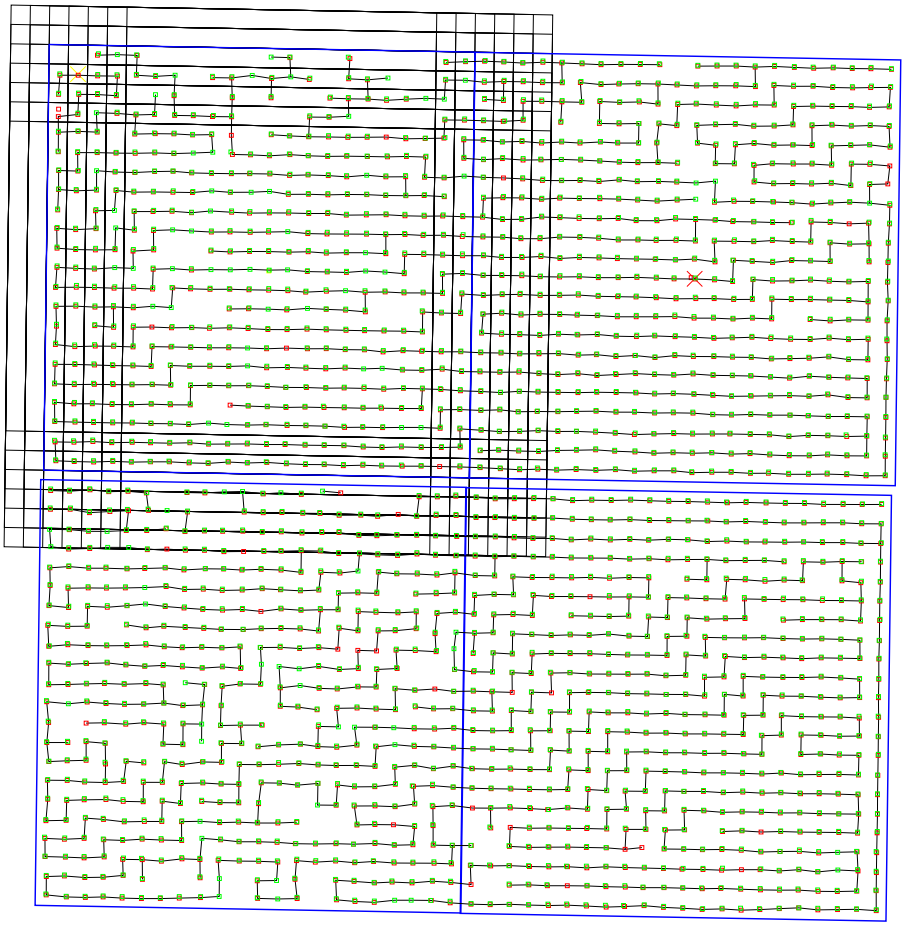


Abbildung 6.10: Gitterorthypothesen (schwarze Rechtecke) für eines von vier Gittern eines Arraybildes. Die Projektionsbereiche sind blau eingezeichnet, im Inneren sind die verketteten Regionen aus rotem und grünem Bildkanal zu sehen.

Vektoren von Instantiierungen aller Zufallsvariablen $f = (f_1, \ldots, f_K)$ heißen *Konfigurationen* des MZF.

Das Nachbarschaftssystem legt die Struktur der Abhängigkeiten der Zufallsvariablen fest. Man kann Markov-Zufallsfelder als Verallgemeinerung von Markov-Ketten verstehen, in denen die Zufallsvariablen nur von einer festen Zahl von Vorgängern abhängig sein dürfen. Anders als Markov-Ketten sind Markov-Zufallsfelder aber nicht gerichtet. Die Markov-Bedingung (6.8) drückt die lokalen Eigenschaften des Zufallsfeldes aus, die wesentlich zur praktischen Nützlichkeit des Ansatzes beiträgt. Die Positivitätsbedingung (6.9) bedeutet, dass es keine verbotenen oder unmöglichen Konfigurationen geben darf. Wenn die Wertemengen $\mathcal{L}_i$ aller Zufallsvariablen gleich sind, heißt das MZF *homogen*.

In den meisten Anwendungen interessiert man sich für eine global optimale Konfiguration $f^*$, möchte aber die lokalen Eigenschaften des MZFs für die Modellierung nutzen. Dank der Äquivalenz von Markov-Zufallsfeldern mit Gibbs-Zufallsfeldern, die Verteilungen der Form (6.10) besitzen, kann die Verbundwahrscheinlichkeit mit Hilfe der Cliquen[3] des Graphen vergleichsweise einfach ausgedrückt werden.

$$P(f) = \frac{e^{-\frac{U(f)}{kT}}}{Z} \tag{6.10}$$

Die Normierungskonstante $Z$ enthält eine Summe über den Konfigurationsraum $\mathbb{F}$ und heißt daher Zustandssumme oder Partitionierungsfunktion. Die Potentialfunktion $U(f)$ ist in Clique-Potentiale $V_c$ zerlegbar:

$$U(f) = \sum_{c \in \mathcal{C}_1} V_1(f) + \sum_{c \in \mathcal{C}_2} V_2(f) + \sum_{c \in \mathcal{C}_3} V_3(f) + \ldots \tag{6.11}$$

$\mathcal{C}$ ist die Menge aller Cliquen des Graphen und die $\mathcal{C}_k$ sind die Mengen der $k$-elementigen Cliquen. In inhomogenen Markov-Zufallsfeldern sind die Cliquepotentiale zusätzlich vom Knoten $i \in \mathcal{S}$ abhängig:

$$\begin{aligned} U(f) &= \sum_{\{i\} \in \mathcal{C}_1} V_1(i, f) \\ &+ \sum_{\{j,k\} \in \mathcal{C}_2} V_2(j, k, f) + \sum_{\{l,m,n\} \in \mathcal{C}_3} V_3(l, m, n, f) + \ldots \end{aligned} \tag{6.12}$$

Um die Verteilung $P(f)$ anzugeben, braucht man also nur die Cliquen des Graphen betrachten. In der Praxis wird die Summe (6.11) oft schon bei Cliquengröße zwei abgebrochen, sodass nur paarweise Kontexte modelliert werden.

Markov-Zufallsfelder sind ursprünglich in der theoretischen Physik zur Beschreibung von Wechselwirkungen der magnetischen Momente der Atome in Festkörpern entwickelt worden [64]. Inzwischen gibt es verschiedenste Anwendungen in Natur-, Wirtschafts- und Sozialwissenschaften und nicht zuletzt in der Bildverarbeitung.

[3] Vollständig verbundene Teilgraphen

Die typisch in der Bildverarbeitung eingesetzten Markov-Zufallsfeldmodelle unterscheiden sich von den ersten Modellen aus der Physik dadurch, dass neben den A-priori-Eigenschaften, die durch die Verteilung (6.10) ausgedrückt werden, auch (verrauschte) Beobachtungen $d$ der Konfiguration bekannt sind. In dieser Situation ist es zweckmäßig, einen Bayes-Ansatz zur Bestimmung der optimalen Konfiguration zu wählen [74]. Dazu benötigt man eine Kostenfunktion $C(f^*, f)$, mit der die Risikofunktion

$$R(f^*) = \sum_{f \in \mathbb{F}} C(f^*, f) P(f|d)$$

definiert wird. Für die quadratische Kostenfunktion folgt, dass die Konfiguration mit maximaler A-posteriori-Wahrscheinlichkeit das Risiko minimiert:

$$f^* = \arg\max_{f \in \mathbb{F}} P(f|d) \tag{6.13}$$

Nimmt man an, dass die Beobachtungsdaten $d_i$ der Knoten unabhängig und gleich verteilt sind, so folgt mit dem Satz von Bayes für die Energiefunktion der Gibbs-Verteilung mit gegebenen Daten der Zusammenhang (6.14) [30]:

$$U(f|d) = \sum_{c \in \mathcal{C}} V_c(f) - \sum_{s \in \mathcal{S}} \log P(d_s|f_s) \tag{6.14}$$

Daraus liest man ab, dass die Beobachtungen wie die Potentiale der einelementigen Cliquen in einem inhomogenen MZF wirken.

### 6.4.2 Markov-Zufallsfeld für die Gittersegmentierung

Durch die Methoden der vorangegangenen Abschnitte stehen für die Gittersegmentierung Hypothesen für die Lage jedes Gitters zur verfügung, aus denen eine geeignete Auswahl getroffen werden muss. Dazu wird ein Markov-Zufallsfeldmodell benutzt: Die Knotenmenge $\mathcal{S}$ bilden die gesuchten Gitter, denen jeweils durch die Zufallsvariablen $f_i$ eine der Hypothesen aus den Mengen $\mathcal{L}_i$ zugeordnet werden muss. Die Cliquepotentiale modellieren die Verträglichkeit der Hypothesen, die z.B. überlappungsfrei sein müssen.

Diese Formulierung der Gittersegmentierung hat Ähnlichkeit zur Objektklassifikation mit Hilfe des Szenenkontexts (siehe z.B. [74], Kap. 5.2.1). Der wesentliche Unterschied besteht darin, dass bei der Gittersegmentierung die Hypothesenmengen $\mathcal{L}_i$ abhängig von dem zu segmentierenden Bild sind, während im anderen Fall Objektklassen unabhängig von einzelnen Bildern existieren.

Die Hypothesenmengen werden aus verschiedenen Bildausschnitten bestimmt und sind daher nicht gleich; das Zufallsfeld der Gittersegmentierung ist also inhomogen.

In vielen MZF-Anwendungen werden die Cliquepotentiale ganz oder teilweise aus Beispieldaten gelernt, was hier aber durch die Inhomogenität und die relativ geringe verfügbare Datenmenge unpraktikabel wird. Daher werden heuristische Clique-Potentiale eingeführt, die einige allgemeine Annahmen über die gewünschten Segmentierungsergebnisse formalisieren. Es gibt ein Potential $V_1$ für einelementige Cliquen und drei Potentiale $V_{21}$ bis $V_{23}$ für zweielementige Cliquen, die im Folgenden motiviert und definiert werden.

1. Die Plausibilität einer einzelnen Gitterhypothese wird durch die Belegung ihrer Gitterknoten mit Regionen abgeschätzt. Es hat sich herausgestellt, dass es zweckmäßig ist, die Randzeilen stärker zu gewichten [58]. Eine leere Randzeile ist ein stärkerer Hinweis auf eine fehlerhafte Gitterpositionierung als leere Knoten im Zentrum eines Messpunktgitters.

   Seien $R_i$ bzw. $C_j$ die Anzahl von Regionen in der $i$-ten Zeile bzw. in der $j$-ten Spalte eines Gitters mit $H$ Zeilen und $B$ Spalten, $M = \lfloor min(H, B)/2 \rfloor$.

$$V_1 = S/N \tag{6.15}$$

$$S = \sum_{i=1}^{M} \frac{e^{-R_i^2} + e^{-R_{H-i-1}^2} + e^{-C_i^2} + e^{-C_{B-i-1}^2}}{1 + (M - i - 1)/M} \tag{6.16}$$

$$N = \sum_{i=1}^{M} \frac{1}{1 + (M - i - 1)/M} \tag{6.17}$$

   Die stärkere Gewichtung der Gitterränder wird durch den Nenner von S realisiert. Die Exponentialfunktionen bewirken, dass wenige fehlende Messpunkte nur geringen Effekt haben, leere Zeilen dagegen besonders schlecht bewertet werden. $N$ normiert den Potentialwert.

   Wie oben gezeigt wirken einelementige Cliquepotentiale analog zu Beobachtungen einzelner Knotenkonfigurationen.

2. Da Gitter im Normalfall nicht übereinander gedruckt werden, dürfen Überlappungen zwischen Gitterhypothesen nicht vorkommen. Um die Positivitätsbedingung (6.9) nicht zu verletzen, weist man Paaren von überlappenden Gitterhypothesen einen sehr hohen, aber nicht unendlichen Potentialwert zu. Überlappungen könnten prinzipiell mit allen umliegenden Gittern auftreten. Daher gilt für dieses Potential eine Achter-Nachbarschaft auf dem MZF-Graphen. In der Abbildung 6.13 ist diese Nachbarschaft rot dargestellt.

3. Regionen außerhalb von Gitterhypothesen, die in das Gitterraster passen, aber keiner der gewählten Hypothesen angehören, deuten auf falsche Segmentierung hin. Man sucht also die Regionen auf extrapolierten Gitterpositionen außerhalb der Gitterhypothesen und testet, ob sie auf Knoten benachbarter Hypothesen liegen. In Abb. 6.11 links sind an den zu testenden Positionen Kreise eingezeichnet. Natürlich können gelegentlich auch Störregionen an entsprechenden Stellen im Bild sein, weshalb die in Abb. 6.11 rechts gezeigte weiche Bewertung benutzt wird, die kleine Anzahlen von Regionen auf extrapolierten Gitterknoten eher toleriert. Zur Berechnung des Potentials wird der Anteil $n$ der extrapolierten Positionen bestimmt, der mit nirgendwo zugeordneten Regionen besetzt ist. Damit wird der Potentialwert durch $V_{22} = \frac{1-e^{-n^2}}{1-e^{-1}}$ berechnet.

   Diese Potentialkomponente wirkt auf zweielementige Cliquen von Gittern mit gegenüberliegenden Kanten, weil es vom Zustand der benachbarten Knoten abhängt, ob Regionen am Außenrand eines Gitters frei sind oder nicht. Die Nachbarschaft für dieses Potential ist in Abbildung 6.13 grün eingezeichnet.

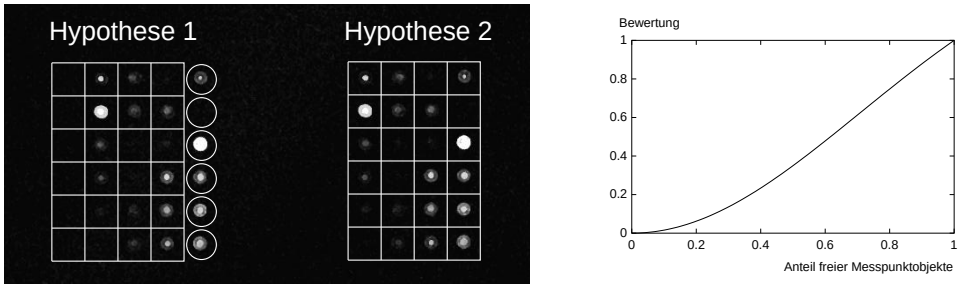



Abbildung 6.11: Zum Cliquepotential zur Bewertung von Messpunktobjekten ausserhalb von Gitterhypothesen. Links: Extrapolierte Gitterknoten (Kreise), an denen nach Messpunktobjekten gesucht wird. Rechts: Die Bewertungsfunktion für den Anteil von Extrapolierten Knoten mit Messpunktobjekten, die zu keinem Gitter gehören.

4. Die Gitter sind meistens in regelmäßigen Zeilen und Spalten angeordnet, wobei aber Abweichungen von ein bis zwei Messpunktzeilen- bzw. Messpunktspaltenabständen vorkommen. Daher wird in Form des Ausdrucks $V_{23} = \frac{h \sin \phi}{g}$ mit den in Abb. 6.12 gezeigten Längenbezeichnungen eine weitere Potentialkomponente für zweielementige Cliquen eingeführt. Anschaulich beschreibt dieser Ausdruck die Anzahl von Messpunktzeilen, um die die betrachtete Hypothese aus einer horizontalen Reihe herausragt. Diese Komponente muss geringer gewichtet werden als die datenabhängigen Terme, damit sie bei tatsächlich unregelmäßig angeordneten Gittern „überstimmt“ wird. Die Nachbarschaft der Gitter in Zeilen und Spalten ist in Abb. 6.13 mit blauen Pfeilen dargestellt.

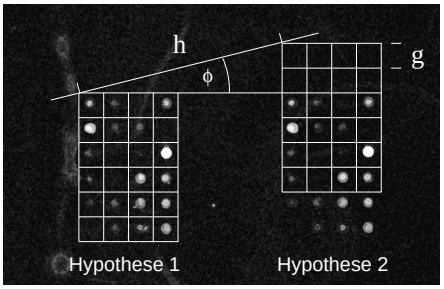



Abbildung 6.12: Das Cliquepotential zur Bewertung der Regelmäßigkeit der Gitteranordnung

Insgesamt lautet die Energiefunktion für das Markov-Zufallsfeldmodell der Gittersegmentierung damit

$$U(f) = \alpha \sum_{c \in \mathcal{C}_1} V_1(c) + \sum_{c \in \mathcal{C}_2} \{\beta_1 V_{21}(c) + \beta_2 V_{22}(c) + \beta_3 V_{23}(c)\} \tag{6.18}$$

Die wesentlichen Eigenschaften der gesuchten Gitteranordnung sind darin mit den Potentialfunktionen der ein- und zweielementigen Cliquen ausgedrückt, wodurch die Komplexität der Energieminimierung gering gehalten wird. Neben dem Rechenauf-

wand steigt durch die Verwendung von Cliquen höherer Ordnung auch die Zahl der zu schätzenden Parameter.

Diese Modellierung hat eine Eigenschaft, die sich nicht ganz genau in die oben dargestellte MZF-Theorie einfügen lässt: Die Potentiale $V_{21}$ bis $V_{23}$ für Zweiercliquen modellieren nicht nur A-priori-Wissen, sondern enthalten auch Beobachtungskomponenten. Beobachtungen auf Paaren oder Cliquen höherer Ordnung sind in dem Beobachtungsmodell Gl. 6.14 jedoch nicht vorgesehen und können auch nicht ohne weiteres formal korrekt eingeführt werden, weil eine Zerlegung der MZF-Verbundwahrscheinlichkeit nach Paaren fehlt. Für die Einzelclique-Beobachtungen (Gleichung (6.14)) folgt die Zerlegung aus der Unabhängigkeitsannahme. Die Beobachtungen auf Paaren werden im weiteren dennoch wie Beobachtungen auf einelementigen Cliquen behandelt.

### 6.4.3 Parameterwahl

Zur Vervollständigung des Modells fehlen noch die Werte der Modellkoeffizienten $\alpha$ und $\beta_1$ bis $\beta_3$. Die Anwendung von Schätzverfahren für diese Parameter ist nicht ohne weiteres möglich, weil die Hypothesenmengen jeweils nur in dem Bild Bedeutung haben, aus dessen Achsenprojektion sie definiert sind. Zudem sind die Hypothesenmengen der einzelnen Knoten unterschiedlich, so dass für das Schätzen der bedingten Wahrscheinlichkeiten durch Auszählen in den Nachbarschaften der Knoten nicht genügend Beispiele vorhanden sind.

Daher werden die Parameter heuristisch gesetzt und die Tauglichkeit und Robustheit der Parameterwahl, die Li auch als Stabilität bezeichnet [74], später im Kapitel 8 empirisch untersucht.

Wegen der Linearität des Modells und der gesamten Energiefunktion bezüglich der Parameter sind nur relative Unterschiede der Parameter wichtig. Daher wird $\alpha = 1$ gesetzt und die anderen Werte relativ dazu gewählt.

Der Parameter $\beta_{21}$ gewichtet das Überlappungs-Potential. Da überlappende Gitter praktisch nicht vorkommen, muss dieser Parameter einen großen Wert (z.B. 100) bekommen.

Der Parameter $\beta_{22}$ gewichtet die Potentialkomponente, die freie Regionen zwischen Gittern bewertet. Wegen des engen Zusammenhangs mit der Bewertung innerhalb der Gitter, die mit $\alpha$ gewichtet wird, wird $\beta_{22}$ wie $\alpha$ auf 1 gesetzt.

Der Parameter $\beta_{23}$ gewichtet die Potentialkomponente, die die regelmäßige Gitteranordnung bewertet. Da bei manchen Mikroarrays die Regelmäßigkeit durchbrochen ist, muss diese Komponente geringer als die datenabhängigen Potentialkomponenten gewichtet werden. Es wird daher $\beta_{23} = 0.33$ gesetzt.

### 6.4.4 Energieminimierung

Die Berechnung der optimalen Konfiguration in einem MZF ist ein höchst komplexes Problem, denn der Konfigurationsraum $\mathbb{F}$ ist sehr groß[4] und die Energiefunktionen lassen im Allgemeinen die Verwendung effizienter Algorithmen zur Bestimmung exakter Lösungen nicht zu. Das trifft auch auf die hier verwendete Energiefunktion zu, weshalb Näherungsverfahren erforderlich sind.

[4] Bei den kleinsten Arrays mit 4 Gittern hat er 5764801 und bei den größten etwa $2.4 * 10^{243}$ Elemente

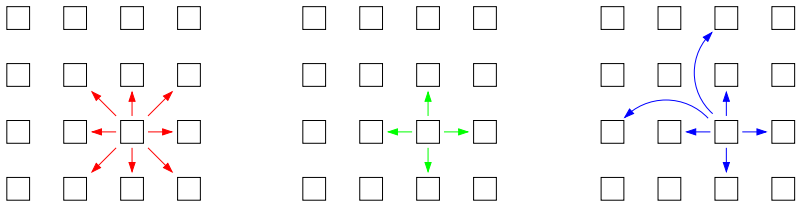


Abbildung 6.13: Die Nachbarschaftssysteme des MZF-Modells zur Gittersegmentierung. Links die Achternachbarschaft für das Überlappungspotential $V_{21}$, in der mitte die Vierernachbarschaft für das Potential $V_{22}$ zur Bewertung freier Regionen und rechts die Nachbarschaft über Zeilen und Spalten für das Potential $V_{23}$ der regelmäßigen Gitteranordnung

Ein klassisches Energieminimierungsverfahren ist „Simulated Annealing“ (simuliertes Erkalten) mit dem Metropolis-Algorithmus (Algorithmus 3), das wie die MZF-Modelle selbst in der statistischen Festkörperphysik entwickelt worden ist.

### Stochastische globale Optimierung

Dieses Verfahren berechnet zu einer vorgegebenen Konfiguration zufällige Änderungen, die entweder die Energie senken oder erhöhen können. Die erlaubten Änderungen werden durch die Nachbarschaft $\mathcal{N}(f)$ im Konfigurationsraum bestimmt. Das gewöhnlich verwendete, einfachste Nachbarschaftssystem erlaubt nur Änderungen von einzelnen Komponenten des Konfigurationsvektors $f$. Energiesenkende Änderungen werden immer akzeptiert, während energieerhöhende Änderungen nur mit einer gewissen, von dem sogenannten Temperaturparameter $T$ abhängigen Wahrscheinlichkeit akzeptiert werden. Bei hohen $T$-Werten sind so gut wie alle Änderungen möglich, während die Energie bei kleinen $T$-Werten praktisch nur sinken oder konstant bleiben kann. Der Simulated-Annealing-Algorithmus erzeugt eine Sequenz langsam sinkender $T$-Parameterwerte, mit denen jeweils Metropolis-Iterationen (Algorithmus 3) bis zu einem stationären Zustand, also z. B. bis sich die mittlere Energie über ein gewisses Zeitfenster nicht mehr wesentlich ändert, ausgeführt werden.

**Algorithmus 3** Metropolis-Algorithmus mit Temperaturparameter $T$

Initialisiere Konfiguration $f$
**repeat**
  generiere $f' \in \mathcal{N}(f)$ zufällig
  $\Delta U \leftarrow U(f') - U(f)$;
  $P \leftarrow \min(1, e^{-\Delta U/T})$;
  **if** (Zufallszahl $\in [0,1[) < P$ **then**
    $f \leftarrow f'$;
  **end if**
**until** ( $U(f)$ stationär)

Man kann zeigen, dass das globale Optimum erreicht wird, wenn für die Absenkung des Temperaturparameters über die Zeit $t$

$$\lim_{t\to\infty} T^t = 0$$

und

$$T^t \geq \frac{m\Delta}{\ln(1+t)} \tag{6.19}$$

gelten, wobei $m$ die Zahl der Knoten des MZF-Graphen und $\Delta$ der größtmögliche (von der Potentialdefinition abhängige) Energieunterschied zwischen zwei Konfigurationen ist [42].

Die Vorschrift (6.19) zur Temperaturabsenkung kann praktisch nicht eingehalten werden, weil sie zu viele Schritte benötigt. Stattdessen wird die Temperatur exponentiell abgesenkt:

$$T^{t+1} = \lambda T^t \quad \text{mit} \quad \lambda \sim 0.999 \ldots 0.9999 < 1$$

Die Konvergenzgeschwindigkeit des Verfahrens wird erheblich erhöht, wenn beim „Sampling“, also der zufälligen Auswahl neuer Konfigurationen die bedingten Wahrscheinlichkeiten des MZF berücksichtigt werden, anstatt beliebige Zustandsänderungen zu erzeugen (Gibbs-Sampling [42]). Dazu gehört auch die Betrachtung von Zustandsänderungen an mehr als einem Knoten gleichzeitig, wie das Beispiel in Abbildung 6.14 zeigt. Gibbs-Sampling wird für das MZF der Gittersegmentierung näherungsweise implementiert, indem zufällig zusammenhängende Abschnitte von Reihen oder Spalten von Gittern ausgewählt werden, die gemeinsam um eine exponentialverteilte Zahl von Messpunktzeilen oder -spalten verschoben werden.

Es gibt viele weitere verbesserte Varianten von Simulated-Annealing-Verfahren, aber die Konvergenzgeschwindigkeit dieser Algorithmenklasse ist für den praktischen Einsatz meistens zu gering. Mit dem einfachen Samplingverfahren war nach einigen Millionen Iterationen und mehreren Tagen Rechenzeit noch keine Konvergenz abzusehen; mit dem verbesserten Gibbs-Sampling-Verfahren ist die Minimierung für das Beispielbild der S. Meliloti-Stichprobe mit 144 Gittern (siehe Abb 6.14 und Abschnitt 8.2) nach ca. 500000 Iterationen bzw. 10 Stunden Rechenzeit abgeschlossen.

### Deterministische lokale Optimierung

Chou und andere haben den wesentlich effizienteren, heuristischen „Highest Confidence First“ (HCF)- Algorithmus vorgeschlagen [29], der eine Näherungslösung durch lokale, deterministische Optimierung berechnet. Damit die lokale Optimierung möglichst zum globalen Optimum führt, ist eine gute Startkonfiguration nötig. Die Autoren schlagen dazu vor, den Konfigurationsvektor $f$ entsprechend den einelementigen Cliquepotentialen optimal zu initialisieren. Die Anfangskonfiguration wird daher an jedem Knoten $i$ nach der Vorschrift

$$f_i = l^i_{min} = \arg\min_{l\in\mathcal{L}_i} V^i_1(l)$$

gesetzt, worin $V^i_1(l)$ das Potential der einelementigen Cliquen am Knoten $i$ ist.

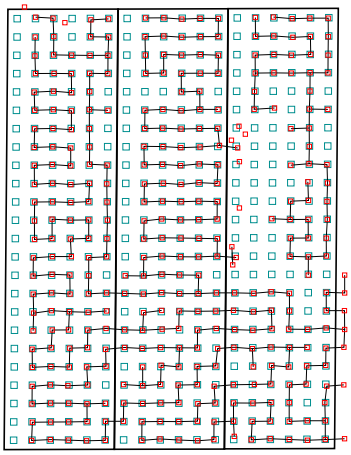


Abbildung 6.14: Drei dicht benachbarte, falsch positionierte Gitter von einem Array der S. meliloti-Serie (schwarz umrandet, Regionen in Rot dargestellt). Der dargestellte Zustand ist ein lokales Minimum, wenn nur einzelne Gitter verschoben werden, weil die Energie durch Überlappung oder frei werdende Messpunktregionen steigt.

Für jeden Knoten $i$ wird die „Stabilität“ $S_i$ definiert, die die mögliche Verbesserung durch eine lokale Konfigurationsänderung misst. Knoten die sich noch im Startzustand befinden müssen besonders behandelt werden, damit ihre Energie auf jeden Fall mindestens einmal mit dem vollständigen Potential berechnet wird. Daher werden die Mengen $\mathcal{L}_i$ um ein besonderes Element ’0’ erweitert, das Knoten im Startzustand kennzeichnet.

$$S_i = \begin{cases} - \min\limits_{l \in \mathcal{L}_i, l \neq l^i_{min}} \left[U_i(l) - U_i(l^i_{min})\right] & \text{falls} \quad f_i = 0 \\ \min\limits_{l \in \mathcal{L}_i, l \neq f_i} \left[U_i(l) - U_i(f_i)\right] & \text{sonst} \end{cases} \tag{6.20}$$

Der HCF-Algorithmus führt eine Folge von lokalen Konfigurationsänderungen an dem Knoten, der den jeweils größten Energiegewinn verspricht durch, bis ein lokales Minimum erreicht ist, d.h. keine Verbesserung durch lokale Konfigurationsänderungen mehr möglich ist. Das Verfahren ist effizient mit einem teilsortierten Binärbaum (Heap) implementierbar, der die nach Stabilität geordneten MZF-Knoten enthält (Algorithmus 4).

Chou und andere haben später eine parallelisierte Variante des Verfahrens beschrieben, die sie „Local HCF“ nennen [30]. Sie unterscheidet sich von der ersten Fassung darin, dass in jeder Iteration die Konfiguration aller Knoten gleichzeitig lokal optimal neu bestimmt wird. Dadurch soll vermieden werden, dass besonders starke Datenkomponenten der Potentiale an einigen wenigen Knoten die Konfiguration schnell in schlechte lokale Minima treiben.

Praktisch wird die Gleichzeitigkeit am einfachsten mit Hilfe von zwei Konfigurationsvektoren realisiert, die abwechselnd nur gelesen bzw. mit den neuen Konfigurationswerten beschrieben und nach jeder Iteration vertauscht werden. Der Rechenauf-

**Algorithmus 4** Highest Confidence First

```
Setze Konfigurationsvektor f = (0, ..., 0)
Berechne nach Gl. 6.20 Stabilität aller Knoten
Erzeuge Min-Heap H^f der Elemente von f geordnet nach Stabilität S
while ( Stabilität des Wurzelelements t in H^f < 0) do
  Wähle lokal optimale Konfiguration für Knotenvariable f_t
  for i ∈ {t} ∪ N_t do
    Berechne Stabilität von Knoten i
    Stelle Heap-Ordnung von H^f wieder her
  end for
end while
```

wand ist erheblich höher als beim einfachen HCF-Algorithmus, weil in jeder Iteration alle Knotenkonfigurationen neu berechnet werden müssen, in deren Nachbarschaft in der vorherigen Iteration eine Konfigurationsänderung stattgefunden hat. Beim einfachen HCF-Algorithmus wird nur die Konfiguration des Knotens mit dem kleinsten $S$-Wert geändert. Daraus ergibt sich aber für die Gittersegmentierung kein Problem, weil die hier betrachteten Zufallsfelder vergleichsweise wenige Knoten ($\leq\sim 10^2$) haben, während die Zufallsfelder in anderen Anwendungen einen Knoten pro Bildpixel besitzen ($\sim 10^5 - 10^6$). Der Local-HCF-Algorithmus stellt einen guten Kompromiss zwischen den Anforderungen möglichst geringer Laufzeit und der Approximation des globalen Energieminimums dar.

### Messpunktdetektion

Für die Berechnung der Potentialfunktionen des Zufallsfeldes müssen an den Gitterknoten der Hypothesen Messpunkte detektiert werden. Dazu kann einfach in den Regionenbildern aus der Hypothesengenerierung nach Regionenschwerpunkten gesucht werden, aber wegen der im Kapitel 5 erwähnten Schwächen der Schwellwertsegmentierung sind die Ergebnisse damit oft unbefriedigend. Besonders bei Arrays mit vielen schwach leuchtenden oder verschmierten Messpunkten, die gar nicht oder nicht getrennt segmentiert werden können, ist die Detektionsrate zu schlecht. In den beiden folgenden Abschnitten werden daher leistungsfähigere Methoden zur Messpunktsegmentierung und -detektion vorgestellt. Sie nutzen die Messpunkt-Abstandsschätzung für Normierungs- und Initialisierungszwecke und sind daher nicht für die Regionensegmentierung in der Hypothesengenerierung einsetzbar.

## 6.5 Messpunktdetektion mit Eigenwertverfahren

### 6.5.1 Motivation

Turk und Pentland [102] haben ein datengetriebenes Verfahren auf Basis der Hauptkomponentenanalyse zur Erkennung von Gesichtern beschrieben. Es erzeugt eine Repräsentation einer Menge von Beispielbildern, die zur Objekterkennung in neuen, unbekannten Bildern benutzt werden kann. Das Verfahren ist bei vielen Bildverarbeitungsproblemen erfolgreich eingesetzt worden, z. B. haben Nattkemper und andere mit diesem Ansatz Einzelzellen in Bildern von Gewebeschnitten detektiert [78]. In diesem Abschnitt wird die Eignung des Eigenwertverfahrens zur Erkennung und Segmentierung von Mikroarray-Messpunkten untersucht.

### 6.5.2 Eigenwerte und Eigenvektoren, Hauptkomponentenanalyse

Turk und Pentland gehen von einer Stichprobe von Bildern aus, die sie als Datenvektoren im hochdimensionalen Raum aller denkbaren Bilder ansehen. Der Ansatz beruht auf der Annahme, dass strukturell ähnliche Bilder der Zielobjekte (z. B. Gesichter oder Messpunkte) in einem niedrigdimensionalen Unterraum des gesamten Bild-Raumes liegen. Die Hauptkomponentenanalyse dient zur Bestimmung einer geeigneten Basis dieses Unterraumes, mit der wesentliche Eigenschaften der Bildstichprobe kompakt beschrieben werden.

Sei $G = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$ die Stichprobe der $M$ je $N^2$ Pixel großen Bilder. Zunächst wird die Stichprobe mittelwertfrei gemacht, d. h. das Mittel der Stichprobenelemente wird von allen Beispielen abgezogen, wodurch die transformierte Stichprobe $F$ entsteht:

$$F = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}, \qquad \mathbf{f}_i = \mathbf{g}_i - \frac{1}{M}\sum_j \mathbf{g}_j$$

Die Hauptkomponentenanalyse betrachtet die Kovarianzmatrix $K$ der transformierten Stichprobe, die Art und Stärke der linearen Abhängigkeiten der Komponenten der Beispielvektoren (der Pixelintensitäten der Bilder) beschreibt.

$$K = \frac{1}{M}\sum_{i=1}^{M} \mathbf{f}_i \mathbf{f}_i^T \tag{6.21}$$

$$= AA^T \quad \text{mit} \quad A = \frac{1}{\sqrt{M}}\left[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\right] \tag{6.22}$$

Gesucht ist ein Koordinatensystem, das die Variablen korrelationsfrei macht, d.h. in dem die Kovarianzmatrix diagonal ist und das gleichzeitig die Varianzen der Stichprobe entlang der Achsen maximiert. Ein solches Koordinatensystem wird von den Eigenvektoren $\mathbf{u}_i$ der Kovarianzmatrix $K$ gebildet. Die Eigenwerte $\lambda_i$ sind die Varianzen entlang der Achsen des von den Eigenvektoren aufgespannten Unterraumes (Vorausgesetzt, es gibt $N$ Eigenvektoren):

$$KU^T = U^T diag(\lambda_1, \ldots, \lambda_N) = diag(\lambda_1, \ldots, \lambda_N)U$$

Zur Darstellung des gesuchten Unterraumes werden die ersten $k$ Eigenvektoren mit den größten Varianzen benutzt, sodass die wesentliche Variation der Stichprobe erfasst wird. In der Regel besitzt die Kovarianzmatrix nur sehr viel weniger als die $N^2$ möglichen Eigenwerte, denn eine repräsentative Stichprobe von Bildern, auf die das Verfahren sinnvoll anwendbar ist, sollte gerade nicht in jeder prinzipiell möglichen Richtung variieren.

Turk und Pentland nennen die Hauptachsen der Stichprobe von Gesichtsbildern auch „Eigengesichter“, da sie als Bilder betrachtet Gesichtern ähnliche Strukturen zeigen.

Die Hauptachsentransformation beschreibt also eine Stichprobe durch orthogonale Eigenvektoren ihrer Kovarianzmatrix. Sie liefert eine Approximation der Stichprobenverteilung mit der Mächtigkeit einer quadratischen Form, denn die Kovarianzmatrix besteht aus Momenten zweiter Ordnung. Diese Eigenschaft ist wichtig für die Klassifikation von Bildern.

Es existieren grundsätzlich niemals mehr als $M$ (Stichprobengröße) Eigenvektoren. Turk und Portland betrachten daher die Eigenwertgleichung für die $M \times M$-Matrix $A^T A$ anstelle der $N^2 \times N^2$-Matrix $K$:

$$A^T A \mathbf{v}_i = \mu_i \mathbf{v}_i$$

Multipliziert man von links mit $A$, so erkennt man, dass die $A\mathbf{v}_i$ Eigenvektoren von $K$ sein müssen:

$$AA^T A \mathbf{v}_i = KA\mathbf{v}_i = \mu_i A\mathbf{v}_i$$

Damit ist für kleine Stichproben wegen der verkleinerten Dimension des Eigenwertproblems eine erheblich effizientere und numerisch stabilere Verarbeitung möglich. Das Eigenwertproblem wird mit der im ESMERALDA-Paket [40] implementierten Jakobi-Iteration [92] gelöst.

### Anwendung zur Objekterkennung

Neue Beispielbilder werden nach Turk und Pentland klassifiziert, indem man prüft, ob sie in dem von den ersten $k$ Eigenvektoren aufgespannten Unter-Bildraum liegen. Dazu projiziert man das Beispielbild $\mathbf{b}$ in den Unterraum und erhält den Koordinatenvektor $\omega$, der die Anteile der Eigenkomponenten an der Eigenvektordarstellung von $\mathbf{b}$ beschreibt:

$$\omega = [\mathbf{u}_1, \ldots, \mathbf{u}_k]^T (\mathbf{b} - \frac{1}{M} \sum_j \mathbf{g}_j) \tag{6.23}$$

Umgekehrt wird mit dem Koordinatenvektor eine Rekonstruktion $\tilde{\mathbf{b}}$ des ursprünglichen Bildes erzeugt:

$$\tilde{\mathbf{b}} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]\, \omega$$

Der euklidische Abstand von Original und Rekonstruktion $d = \|\tilde{\mathbf{b}} - \mathbf{b}\|$, also der Rekonstruktionsfehler, dient als (einziges) Klassifikationsmerkmal. Bilder der Zielklasse sollten durch Projektion und Rekonstruktion nur wenig verändert werden, wogegen die Projektion auf die Eigenvektoren andere Bilder stark verfremdet.

Diese Methode der Klassifikation, die als „Einklassenansatz“ bezeichnet werden könnte, liefert nur dann gute Ergebnisse, wenn das Zielklassengebiet im Bild-Raum gut durch eine quadratische Form beschrieben werden kann. Das ist in der Praxis nicht unbedingt der Fall, wie anhand der Beispiele im folgenden Abschnitt gezeigt wird[5].

Bessere Ergebnisse werden erreicht, indem der Koordinatenvektor als Merkmalsvektor eines gewöhnlichen Zweiklassenproblems verwendet wird. Zum Training des Klassifikators muss dann allerdings auch eine Stichprobe von Negativ-Beispielen, also Nicht-Gesichtern oder Nicht-Messpunkten vorhanden sein.

### 6.5.3 Eigenspots

Grundsätzlich sind zwei Anwendungen des Verfahrens für die Segmentierung von Mikroarraybildern interessant.

Erstens stellt sich bei der Berechnung der Gitterhypothesen-Bewertungsfunktionen der Potentialfunktion des Markov-Zufallsfeldes (siehe Abschnitt 6.4.2) das Problem, an gegebenen Stellen des Bildes über das Vorhandensein eines Spots zu entscheiden (Detektion). Zweitens können mit der von Turk und Pentland beschriebenen Kartentechnik („Face Map“) ganze Bilder segmentiert werden. Das Haupthindernis ist die schon von Turk und Pentland bemängelte geringe Robustheit des Eigenwertansatzes gegen Skalierungen. Die Abstände und mittleren Größen der Messpunkte in der zur Verfügung stehenden Stichprobe variieren um fast eine Größenordnung (von ca 8 bis zu 60 Pixeln), was die Größennormierung unumgänglich macht. Die Kartentechnik ist daher als Ersatz für die Schwellwertverfahren aus Kapitel 5 weitgehend ungeeignet, solange die Auflösung nicht fest vorgegeben oder extern kalibriert wird. Die Abtastung und Bewertung des Skalierungsparameters analog zur Abtastung des Schwellwertparameters in der Regionensegmentierung unter Bewertung der Segmentierungen mit dem Abstandsverteilungsmodell aus Kap. 6 ist prinzipiell möglich, aber wegen der komplizierteren Klassifikation jedes einzelnen Bildpunktes erheblich aufwändiger als bei den Schwellwertverfahren.

Bei der Detektionsaufgabe im MZF sind mit den Hypothesen die Abstände zwischen den Messpunkten schon vorgegeben, so dass zumindest eine näherungsweise Normierung erfolgen kann. Die Normierung über die Spalten- und Zeilenabstände ist nicht perfekt, weil die Messpunkte immer noch verschiedene Anteile der Spalten- und Zeilenzwischenräume ausfüllen können.

#### Klassifikationsproblem Messpunkt-Detektion

Zur Erkennung von Messpunkten ist in jedem Fall eine Stichprobe von Messpunktbildausschnitten nötig. Beim leistungsfähigeren Zweiklassenansatz muss die Stichprobe zusätzlich Bildausschnitte ohne Messpunkte enthalten. Die Stichproben sind am einfachsten anhand fertiger Gittersegmentierungen zu erstellen. Die Ausschnittgrößen entsprechen Gitterzellen des jeweiligen Bildes, die auf eine Standardgröße von $40\times40$ Pixeln skaliert werden. Leichtes Verrauschen der Positionen der Bildausschnitte führt zu verbesserten Ergebnissen bei der Anwendung im MZF zur Gittersegmentierung. Die Klassifikation der Stichprobe wird implizit durch die Gittersegmentierung vorgegeben; es gibt Ausschnitte an Messpunktpositionen und Bildbereiche außerhalb der

[5] Für Gesichtsbilder gibt es ähnliche, unveröffentlichte Ergebnisse von G. Fink und S. Lang

Messpunktgitter. Manuelle Klassifikation (z.B. bei dunklen Punkten innerhalb der Gitter) führt zu verbesserter Klassentrennung auf der Stichprobe von Bildausschnitten, verschlechtert aber die Ergebnisse der MZF-Gittersegmentierung.

Die Abbildung 6.15 zeigt die Mittelwerte und Eigenvektoren für Stichproben von Messpunkt-Bildausschnitten verschiedener Arraybilder (zu den Originaldaten siehe auch Abschnitt 8.2 und Anhang E), also positiv klassifizierten Elementen der Stichprobe, sortiert nach Größe der zugehörigen Eigenwerte. Der erste Eigenvektor hat bei fast allen Beispielen die gleiche Struktur wie der Mittelwert und beschreibt deshalb die Intensitätsvariation. Die zweiten und dritten Eigenvektoren beschreiben in den meisten Fällen die bei der Festlegung der Bildausschnitte künstlich verstärkten seitlichen Verschiebungen und der vierte Eigenvektor erfasst die Größenvariationen (ringförmiges Muster). Die weiteren Eigenvektoren sind nicht ohne weiteres anschaulich interpretierbar. Man stellt fest, dass sich nicht bei allen Teilstichproben die gleiche Art von Eigenwertzerlegung ergibt, weil offenbar die unterschiedlichen Arten von Variabilität bei verschieden hergestellten Arrays nicht in jedem Fall gleich gewichtet sind. Die Bilder aus der vcho-Stichprobe enthalten starke Hintergrundartefakte, die sich in den Eigenvektorzerlegungen niederschlagen. Zu der Kovarianzmatrix der vcho0-Stichprobe waren nur 6 Eigenvektoren numerisch bestimmbar.

Das Eigenwertspektrum zeigt, dass mit den ersten neun Eigenvektoren wesentliche Teile der Stichprobenvarianz erfasst werden.

Als Klassifikatoren kommen zum Beispiel Polynomklassifikatoren [70, 91] oder Supportvektormaschinen [24, 27] in Frage, da die als Merkmalsvektoren benutzten Koordinatenvektoren feste Länge haben.

Die Supportvektormaschine (SVM) hat prinzipiell bessere Generalisierungseigenschaften als der Polynomklassifikator, aber die Berechnungskomplexität der SVM-Unterscheidungsfunktion wächst mit der Stichprobengröße (genauer gesagt mit der Anzahl der nötigen Support-Vektoren), wogegen der Aufwand beim Polynomklassifikator nur vom Polynomgrad und der Anzahl der Merkmale abhängt. Praktisch ist der Berechnungsaufwand verglichen mit dem der anderen Systemkomponenten gering, daher sollte in der Regel die leistungsfähigere SVM-Klassifikation benutzt werden.

Mit dem Polynomklassifikator werden ca. 85% der Beispiele einer vom Trainingsdatensatz unabhängigen Testmenge richtig klassifiziert und mit der SVM 93%. Diese Ergebnisse schwanken um bis zu 5%, wenn Teilstichproben (Messpunkte verschiedener Arrays) zwischen Test- und Trainingsmengen ausgetauscht werden. Die Stichprobe umfasste nach dem Ausgleich der a-priori-Wahrscheinlichkeiten durch Zufallsauswahl insgesamt 5584 Beispiele. Es wurden die Implementationen des Polynomklassifikators von Kummert [70] und der SVM von Chang und Lin [27] verwendet.

### Exkurs zur Segmentierung von Gesamtbildern: Die Spot Map

Indem man das Detektionsverfahren nicht nur an einzelnen Stellen, sondern auf jeden möglichen Bildausschnitt anwendet, erhält man eine Karte der Messpunktähnlichkeit des Bildes. Die Abbildung 6.16 zeigt eine solche „Spot Map“[6]. Darin sind die lokalen Maxima der Messpunktähnlichkeit durch weiße Kreuze markiert. So gut wie alle Messpunkte des Beispiels werden detektiert, allerdings war für dieses Ergebnis die

[6] Analog zur „Face Map“ nach Turk und Pentland

manuelle Einstellung des Skalierungsparameters nötig.

## 6.6 Messpunktdetektion mit aktiven Konturen

Alternativ zum Eigenspot-Ansatz wird in diesem Abschnitt ein nicht primär datengetriebenes Verfahren untersucht.

### 6.6.1 Motivation

Die in Abschnitt 5.2 beschriebenen Schwellwertverfahren zur Regionensegmentierung besitzen nur begrenzte lokale Adaptivität, weil die Fenster bei der Histogrammberechnung nicht beliebig klein gewählt werden können. Deshalb eignen sie sich nicht gut zur Segmentierung von verschmierten oder nur schwach fluoreszierenden Messpunkten, wie das Beispiel in Abbildung 6.18 zeigt.

Eine Möglichkeit zur Verbesserung der lokalen Adaptivität sind kantenbasierte Verfahren. Die im Folgenden untersuchte Methode segmentiert Regionen für einzelne Messpunkte und ist auch für das Detektionsproblem anwendbar. Die aktive Konturesegmentierung ist im Gegensatz zum Eigenspotverfahren ein primär modellgetriebenes Verfahren.

### 6.6.2 Grundlagen

Kass, Witkin und Terzopoulos haben 1988 die „Snake“ als aktives Konturmodell vorgestellt, das sich in vielen Anwendungen bewährt hat [57].

Sie formulieren die Kontursegmentierung als Variationsproblem und beschreiben Objektgrenzen deshalb zunächst allgemein als parametrische Kurve $\vec{v}(s)$ der Länge $L$. Das zu minimierende Funktional $E^*_{Snake}(\vec{v})$ enthält sowohl Glattheitsterme, die Objektwissen modellieren, als auch Terme, die die Anpassung der Kontur an Kanten im Bild beschreiben. Die Glattheitskomponente oder „interne Energie“ $E_{int}$ setzt sich aus Streckung und Krümmung der Kurve zusammen. Man kann sie auch physikalisch als Streck- und Biegeenergie eines dünnen Stabes interpretieren. Weil die Snakes die Biegeenergie minimieren sind sie Spline-Kurven.

$$\begin{aligned} E^*_{Snake} &= \int_0^L E_{Snake}(\vec{v}(s))ds \\ &= \int_0^L E_{ext}(\vec{v}(s)) + E_{int}(\vec{v}(s))ds \\ &= \int_0^L E_{ext}(\vec{v}(s)) + \frac{1}{2}(\alpha(s)|\vec{v}_s(s)|^2 + \beta(s)|\vec{v}_{ss}(s)|^2)ds \end{aligned} \tag{6.24}$$

Die „externe Energie“ $E_{ext}(v)$ in Gl. (6.24) beschreibt die lokale (Objekt-) Kantenstärke des Bildes entlang der Kurve $\vec{v}$. Je nach Anwendung kann das die Bildintensität selbst oder eine mit Kantenoperatoren bestimmte Kantenstärke sein. Die Funktionen $\alpha(s)$ und $\beta(s)$ beschreiben die lokalen Steifheitseigenschaften der Kontur und ermöglichen die Modellierung von flexiblen Objekten und Knickstellen. Zur Segmentierung

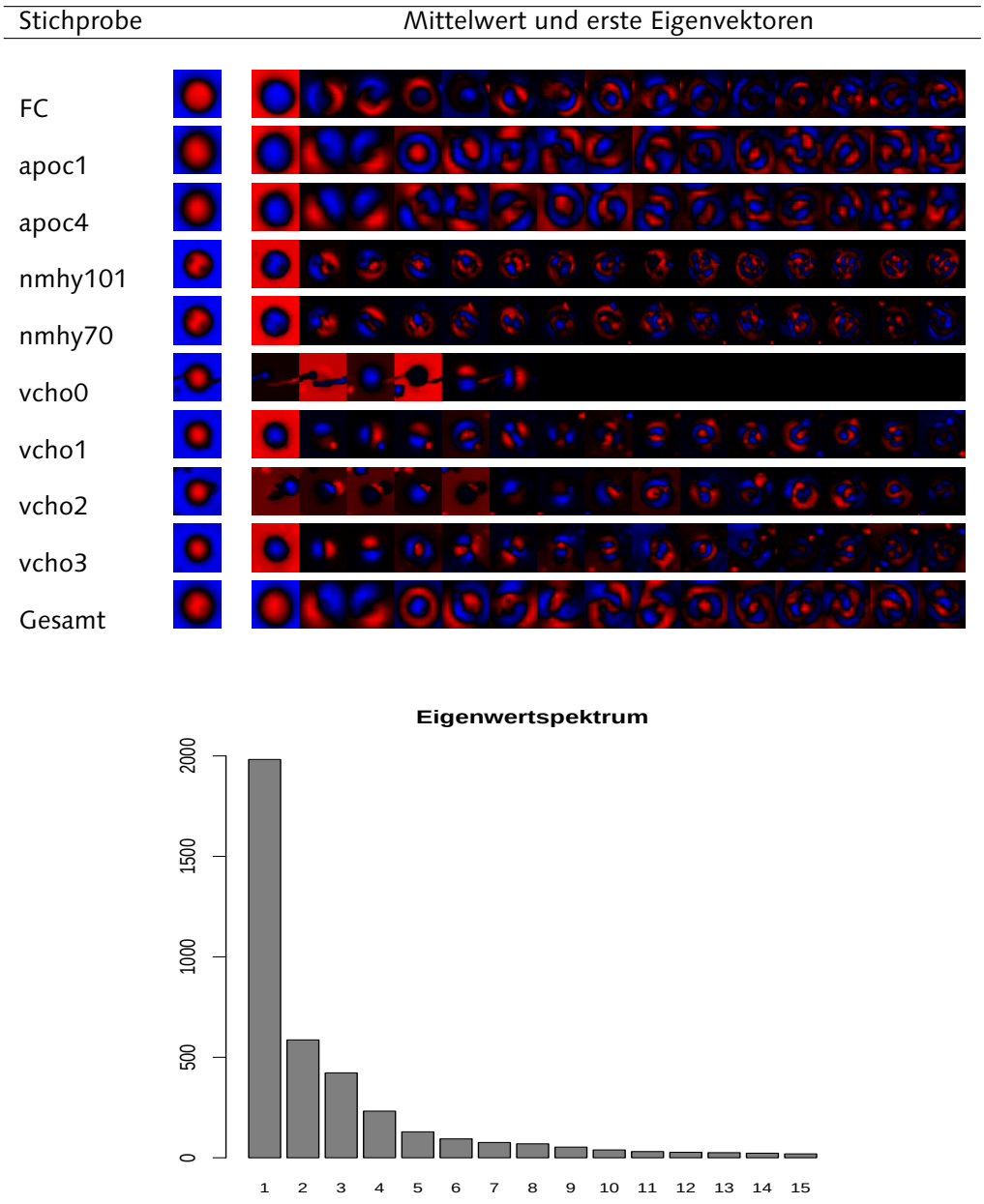



Abbildung 6.15: Hauptkomponenten (oben) und zugehörige Varianzen der Messpunktbild-Stichproben von Mikroarraybildern. Negative Komponenten der Eigenvektoren sind blau dargestellt, positive Komponenten rot. Die Vektoren sind ohne Nullpunktverschiebung linear für den maximalen Intensitätsbereich der Bilddarstellung normiert.

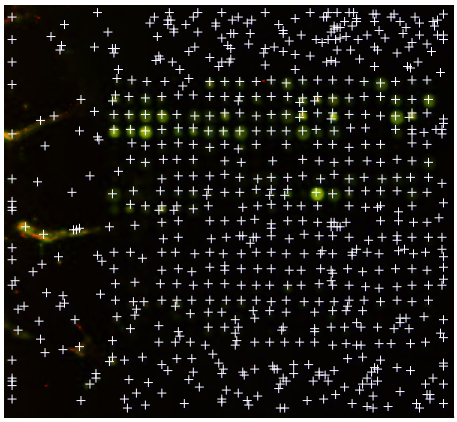


Abbildung 6.16: Ein Ausschnitt aus einem Mikroarraybild der Apo A1-Stichprobe, in dem lokale Maxima der Messpunktähnlichkeit mit weissen Kreuzen markiert sind. Die Grössenskalierung ist manuell eingestellt. Im Hintergrundrauschen gibt es hier wie auch bei anderen Beispielen viele lokale Maxima.

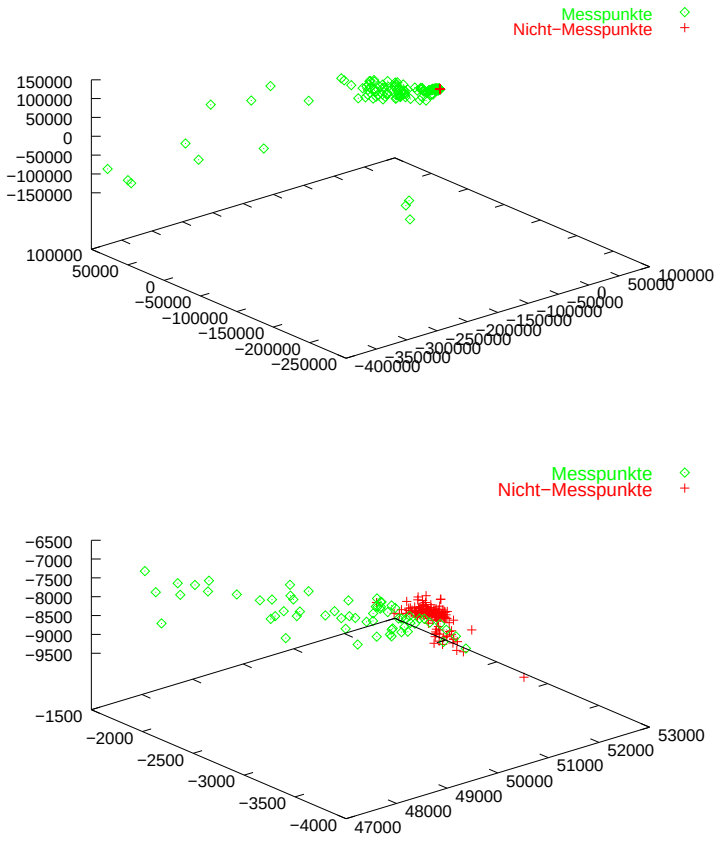



Abbildung 6.17: Projektion von Beispielbildern auf die ersten drei Hauptachsen

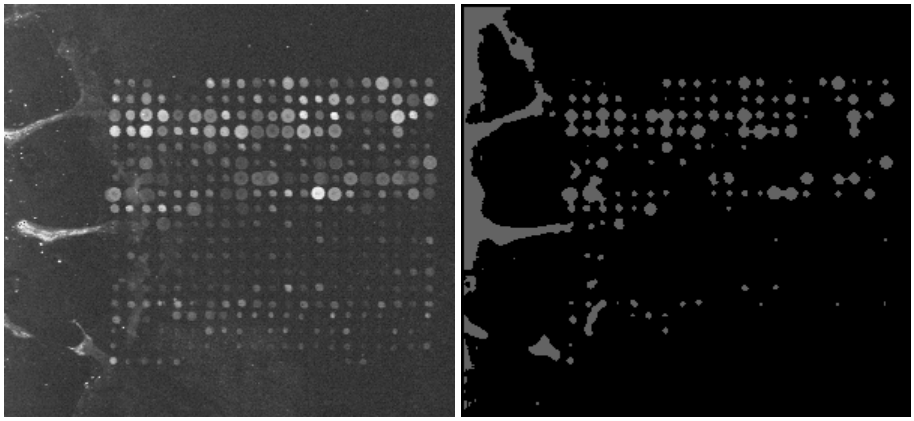


Abbildung 6.18: Links: Aufgehellter Ausschnitt aus einem Bild der ApoA1-Stichprobe ($\gamma$=3), Rechts: Schwellwertsegmentierung des Ausschnittes

von Objekten wird eine initiale Kontur benötigt, die man interaktiv oder mit Hilfe von Vorwissen festlegt und dann bezüglich des Energiefunktionals optimiert.

Für die praktische Durchführung der Energieminimierung benutzen die Autoren der Originalarbeit die Euler-Gleichungen des Variationsproblems, die sie diskretisieren und iterativ lösen. Die Kurve wird diskretisiert als Polygon dargestellt, dessen Knoten auf Bildpixelzentren liegen müssen. Die Ableitungen in den Euler-Gleichungen werden durch diskrete Differenzen approximiert.

Das iterative Lösen der Euler-Gleichungen ist mit $O(n^2)$ Operationen je Iteration ($n$ Knoten im Polygon) recht aufwändig, weshalb bald ein Dynamic-Programming-Algorithmus (Amini und andere [4]) und eine Näherungslösung in Form eines Greedy-Verfahrens (Shah und Williams [105]) entwickelt wurden. Alle Methoden benutzen ein Suchfenster fester Größe um die Polygonknoten, innerhalb dessen die Knoten in einer Iteration verschoben werden können.

Shah und Williams haben neben der schnelleren Minimierung Normierungen der einzelnen Terme des Energiefunktionals vorgeschlagen, die zu verbesserter Robustheit führen.

Aktive Konturen sind erfolgreich in medizinischen Anwendungen, zur Objektverfolgung in Bildsequenzen und zum Stereomatching eingesetzt worden. Man hat den Ansatz auch für die Segmentierung von Oberflächen im dreidimensionalen Raum verallgemeinert.

Mit neueren Ansätzen wie den B-Snakes versucht man die Zahl der zu optimierenden Parameter durch explizite Nutzung von Spline-Basisfunktionen zu reduzieren. Dadurch wird aber ein Rendering-Algorithmus zur Erzeugung der Kurve nötig [12, 18]. Mit traditionell verwendeten lokalen Minimierungsverfahren lassen sich konkave Konturen nur sehr schlecht segmentieren. Deshalb haben verschiedene Autoren stochastische, globale Optimierungsverfahren vorgeschlagen [54, 86].

Für die erfolgreiche Anwendung von aktiven Konturen zur Segmentierung von Mikroarray-Messpunkten sind einige Probleme zu lösen:

1. Die Größenvarianz der Messpunkte führt dazu, dass mehr Knoten im Konturpo-

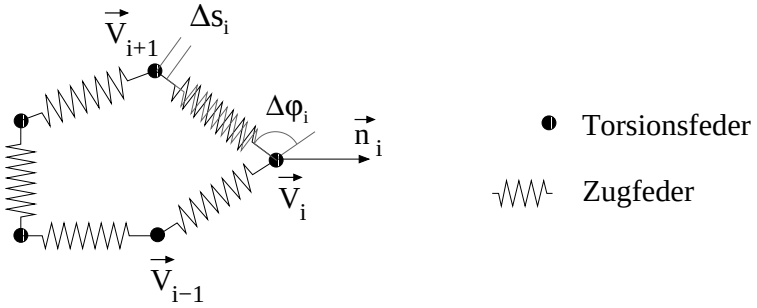



Abbildung 6.19: Das semikontinuierliche Konturmodell und seine Parameter. Die Normalenvektoren der Kontur zeigen an den Knoten aus dem Konturpolygon heraus und stehen im gleichen Winkel zu beiden angrenzenden Segmenten.

lygon als Pixel auf der Zielkontur vorhanden sind. Die Differenzenapproximationen der Ableitungen werden dann äußerst ungenau und die Minimierungsverfahren zerstören die Kurventopologie (es bilden sich Schleifen in dem Konturpolygon).

2. Das Konturmodell berücksichtigt die Kantenorientierung nicht. Eine um einen Messpunkt initialisierte Kontur wird daher auch von benachbarten Messpunkträndern „angezogen“.

3. Kleine, helle Verunreinigungen erzeugen unerwünschte lokale Energieminima, in denen die Minimierung vorzeitig enden kann. Die Polygonknoten liegen häufig sehr dicht um helle Störpartikel.

### 6.6.3 Semikontinuierliches, skalenunabhängiges Konturmodell

Die ersten beiden Probleme werden durch ein semikontinuierliches Konturmodell gelöst, das aus der physikalischen Interpretation des kontinuierlichen Ansatzes (6.24) motiviert ist.

Man setzt dazu die Kontur von vornherein als Polygon an, dessen Knoten und Kanten als Dreh- und Zugfederelemente gedacht sind (siehe Abb. 6.19).

Die Entsprechung des Energiefunktionals des kontinuierlichen Ansatzes ist hier die Energiefunktion (6.25), die von den Winkel- und Auslenkungsparametern der Federelemente abhängig formuliert ist. Die Auslenkungen hängen natürlich wieder von den Knotenkoordinaten $\vec{q} = [v_{0x}, v_{0y}, v_{1x}, v_{1y}, \ldots, v_{Nx}, v_{Ny}]^T \in \mathbb{R}^{2N}$ ab, aber die Ableitungen der Energiefunktion, die für die Minimierung notwendig sind, können damit ohne Differenzenapproximationen exakt angegeben werden.

$$E(\vec{q}) = \alpha E_a(\vec{q}) + \beta E_b(\vec{q}) + \gamma E_c(\vec{q}) \tag{6.25}$$

$$E_a = \sum_i \frac{1}{2}(\Delta s_i)^2 \tag{6.26}$$

$$E_b = \sum_i \frac{1}{2}(\Delta \varphi_i)^2 \tag{6.27}$$

$$E_c = -\sum_i \nabla f(\vec{v}_i) \bullet \vec{n}_i \tag{6.28}$$

Die Knotenkoordinaten werden nicht auf Pixelzentren festgelegt, so dass trotz der diskreten Polygonstruktur die Skalenunabhängigkeit erhalten bleibt. Gegenüber dem B-Spline-Ansatz, mit dem sich das Diskretisierungsproblem ebenfalls elegant lösen lässt, besteht der Vorteil, dass die Kontur nicht aus einer Basisdarstellung erzeugt werden muss, um die externe Energie berechnen zu können.

Der Biegeenenergieterm des Federmodells kann auch differentialgeometrisch motiviert werden [105].

Der Bildenergieterm (6.28) misst neben der Kantenstärke auch die Übereinstimmung der Kantenorientierung mit den nach außen gerichteten Konturnormalen $\vec{n}_i$ an jedem Knoten $i$:

$$\nabla f(\vec{v}_i) \bullet \vec{n}_i = \|\nabla f(\vec{v}_i)\| \cos \sphericalangle(\nabla f(\vec{v}_i), \vec{n}_i)$$

Dadurch wird die anziehende Wirkung benachbarter Objektkanten außerhalb des Konturpolygons vermieden, die bei Verwendung der einfachen Gradientenstärke als Bildenergie oft zu Fehlsegmentierungen führt. Als Gradientenoperator wird der Sobel-Operator mit Maskengröße $5 \times 5$ benutzt. Zur Berechnung des Bildenergieterms an den Polygonknoten muss zwischen den Pixelzentren des Gradientenbildes interpoliert werden.

### 6.6.4 Optimierungsverfahren

Der Parameterraum ($\mathbb{R}^{2N}$) des Federmodells ist kontinuierlich, weshalb ein kontinuierliches Energieminimierungsverfahren verwendet werden sollte.

#### Gradientenabstieg

Ein besonders einfaches, iteratives Verfahren ist der Gradientenabstieg nach Gleichung (6.29). Der Parametervektor $\vec{q_0}$ beschreibt die aus der Gitterkonstantenschätzung abgeleitete initiale Kontur.

$$\vec{q}_{t+1} = \vec{q}_t - s\vec{\nabla} E(\vec{q}_t) \tag{6.29}$$

Mit diesem Verfahren werden keine guten Ergebnisse erzielt, weil die Energiefunktion offenbar viele unerwünschte lokale Minima besitzt. Man findet sehr häufig, dass sich viele Polygonknoten an Stellen mit großer Kantenstärke dicht zusammendrängen. Auch kann mit der festen Schrittweite $s$ kaum ein stabiler Zustand erreicht werden. Die Beobachtungen legen nahe, dass durch Einführung der Nebenbedingung $E_a = 0$

(d. h. konstante Knotenabstände) sicherer eine Lösung nahe dem globalen Minimum der Energiefunktion (6.25) gefunden wird. Das Minimierungsproblem lautet damit

$$\min \beta E_b(\vec{q}) + \gamma E_c(\vec{q})|_{E_a=0}$$

### Sequentielle quadratische Programmierung

Für Optimierungsaufgaben dieses Typs ist *Sequentielles quadratisches Programmieren* (SQP) ein geeigneter Lösungsansatz [13]. Der SQP-Algorithmus minimiert iterativ die lokale Taylorapproximation 2. Ordnung der Energiefunktion unter der linearisierten Nebenbedingung. Sei $\vec{q}_0$ eine Startstelle im Parameterraum und seien $\mathbf{r}$ $\mathbf{B}$ und $\mathbf{h}$ Näherungen des Gradienten und der Hessematrix von $E_b + E_c$ sowie des Gradienten von $E_a$. Dann berechnet das SQP-Verfahren einen Schritt $\mathbf{d}$ durch

$$\min_{\mathbf{d}} \quad \mathbf{r}_t^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \mathbf{B}_t d \tag{6.30}$$

$$\text{unter} \quad \nabla \mathbf{h}(\vec{q}_t)^T \mathbf{d} + \mathbf{h}(\vec{q}_t) = 0 \tag{6.31}$$

Die Lösung von (6.30) existiert nur, wenn $\mathbf{B}$ positiv definit ist.

Die Minimierung von (6.30) führt auf ein lineares Gleichungssystem mit $n + 1$ Unbekannten, das je Iteration einmal zu lösen ist. Das Verfahren konvergiert mit quadratischer Geschwindigkeit in der Anzahl der Schritte.

Die Hessematrix hat Bandstruktur, d.h. nur die Hauptdiagonale und die ersten drei Nebendiagonalen sind besetzt, weil die Parameter $q_i$ (die Knotenkoordinaten) nur in den Federenergien von drei benachbarten Polygonknoten und dem Bildenergieterm eines Knotens vorkommen. Im Gleichungssystem zur Lösung von (6.30) wird die Bandstruktur aber durch die Gleichung für die linearisierte Nebenbedingung (6.31) gestört, sodass keine effizienten Lösungsverfahren für Bandmatrixgleichungen genutzt werden können.

Die Anwendung des SQP-Verfahrens scheitert in der Praxis, weil die Hessesche Matrix der Energiefunktion $E_b + E_c$ nur in einer sehr kleinen Umgebung um die gesuchte Minimalstelle positiv definit und das Verfahren dadurch numerisch sehr instabil ist. Durch Regularisierung könnte man eventuell bessere Stabilität erreichen, aber die quadratische Konvergenzgeschwindigkeit wäre vermutlich nicht zu halten. Der Aufwand für die Lösung des Gleichungssystems in jeder Iteration erscheint dann zu hoch.

Stattdessen wird das in Algorithmus 5 auf S. 93 gezeigte Verfahren verwendet, das Schritte im Parameterraum aus den Gradientenrichtungen

$$\frac{1}{|\nabla E_a|}\nabla E_a, \frac{1}{|\nabla E_b|}\nabla E_b, \frac{1}{|\nabla E_c|}\nabla E_c$$

der Energieterme $E_a$, $E_b$ und $E_c$ zusammensetzt. Die Gradientenrichtungen werden mit den Parametern $\alpha$, $\beta$ und $\gamma$ gewichtet. Wenn $\alpha$, der Parameter des Streckenergiegradienten, relativ zu den anderen Gewichten groß genug ist, bleibt die Nebenbedingung näherungsweise erfüllt.

**Algorithmus 5** Komponenten-Gradientenabstieg mit Schrittweitenadaption

$\lambda^0 \leftarrow 1.0$
**while** $\lambda^t > eps$ **do**
  $\vec{\delta} \leftarrow -\left(\frac{\alpha}{|\nabla E_a|}\nabla E_a + \frac{\beta}{|\nabla E_b|}\nabla E_b + \frac{\gamma}{|\nabla E_c|}\nabla E_c\right)$
  **if** $E(\vec{q^t} + 1.2\lambda^t\vec{\delta}) < E(\vec{q^t})$ **then**
    $\lambda^{t+1} \leftarrow 1.2\lambda^t$
  **else**
    **repeat**
      $\lambda^t \leftarrow 0.5\lambda^t$
    **until** $(E(\vec{q^t} + \lambda^t\vec{\delta}) < E(\vec{q^t}))||(\lambda < eps)$
    $\lambda^{t+1} \leftarrow \lambda^t$
  **end if**
  $\vec{q}_{t+1} \leftarrow \vec{q}_t + \vec{\delta}$
  $t \leftarrow t + 1$
**end while**

#### Konvergenz und Initialisierung

Das Verfahren adaptiert seine Schrittweite, damit in jeder Iteration die Energie sinkt. Wenn ein Schritt $\vec{\delta}$ im ersten Versuch die Energie verkleinert, wird die Schrittweite moderat vergrößert und im anderen Fall so lange verkleinert, bis eine Energiesenkung erreicht oder die Mindestschrittweite unterschritten ist. Da die Energiefunktion (6.25) nach unten beschränkt ist, muss das Verfahren konvergieren. Die Wahl der Startparameter $\vec{q}_0$ hat natürlich großen Einfluss auf die Zahl der nötigen Iterationen. Das Verfahren konvergiert besonders zuverlässig auf die gewünschte Objektkante, wenn die initiale Kontur die Zielkontur bereits berührt. Wenn der Abstand der initialen Kontur zur Zielkontur größer als der Radius der Gradientenoperator-Maske ist, erhält man mit größerer Wahrscheinlichkeit ein unerwünschtes Ergebnis. Zur Segmentierung der Mikroarray-Messpunkte sollte versucht werden, die initiale Kontur *von außen* zur Zielkontur laufen zu lassen, weil oft Intensitätsstufen im Inneren der Messpunkte unerwünschte lokale Minima bilden (siehe z. B. S. 187 unten). Praktisch wird das Konturpolygon daher so initialisiert, dass eine Hälfte der Knoten auf einem Halbkreis liegt und die andere auf einer glatt anschließenden Halbellipse, deren große Halbachse so lang wie der Durchmesser des Halbkreises ist. Geeignete Durchmesser sind durch die Messpunktabstände der Gitterhypothesen gegeben (siehe Abschnitt 6.3). Die Anzahl der Polygonknoten wird so bestimmt, dass die Segmente initial etwa so lang wie der Radius der Gradientenoperator-Maske sind.

### 6.6.5 Merkmale

Nach der Energieminimierung ist noch nicht klar, ob tatsächlich ein Objekt oder ein leerer Bildausschnitt segmentiert wurde. Für die Detektion von Messpunkten gibt es eine Reihe von Merkmalen:

1. Die Bildenergie $E_c$ zeigt die Kantenstärke und -orientierung auf der Konturlinie an und sollte daher gute Hinweise auf das Vorhandensein eines Messpunktes ge-

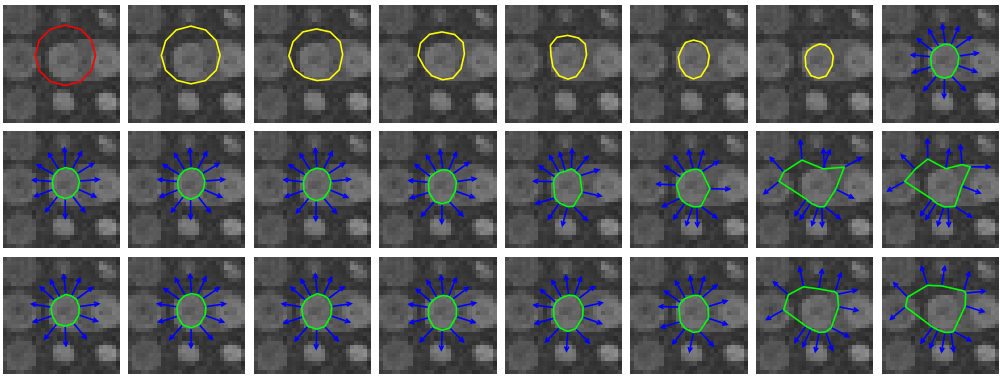


Abbildung 6.20: Erste Zeile: Initiale Kontur, Zwischenzustände und Minimierungsergebnis mit Normalenrichtungen ($\alpha = 0.2$,$\beta = 0.01$,$\gamma = 0.2$). Zweite Reihe: Minimierungsergebnisse unter Variation von $\alpha$ von $0.4$ (links) bis $0.01$ (rechts) bei $\beta = 0.01, \gamma = 0.2$. Letzte Zeile: wie zweite Zeile mit $\beta = 0.2$

ben. Zur Normierung kann die mittlere Kantenstärke im Hintergrund des Bildes benutzt werden.

2. Die Bildenergie $E_c$ sollte unter Variationen der Knotenpositionen senkrecht zur Kontur ansteigen, wenn sich die Kontur in einem gut ausgeprägten Minimum der Bildenergie, d. h. auf einer deutlichen Objektkante befindet. Dazu werden die Hessematrizen $H_i$ der Bildenergie $E_c$ bezüglich der Koordinaten des $i$-ten Polygonknotens betrachtet, die Teil der quadratischen Taylorapproximation der Energiefunktion sind. Die mit dem Normalenvektor $\vec{n}_i$ an diesem Knoten gebildeten quadratischen Formen

$$\vec{n}_i^T H_i \vec{n}_i$$

sollten daher Information über die Signifikanz des gefundenen Kantenzuges enthalten. Der Mittelwert der an allen Polygonknoten berechneten quadratischen Formen bildet ein Merkmal, das dank der Linearität der Ableitungen in der Hessematrix wieder mit der mittleren Hintergrundkantenstärke normiert werden kann.

3. Die Kantenrichtung im Bild und die Konturnormale sollten entlang einer Objektkante gut übereinstimmen. Der Mittelwert

$$\frac{1}{N} \sum_i \frac{\nabla f(\vec{v}_i)}{\|\nabla f(\vec{v}_i)\|} \bullet \vec{n}_i$$

des Skalarproduktes der beiden normierten Richtungsvektoren dient daher als Merkmal.

4. Grundsätzlich kann das Polygon bei der Energieminimierung mit Algorithmus 5 Schleifen bilden, was wesentlich häufiger bei Segmentierung von Hintergrundrauschen vorkommt. Mit dem Bentley-Ottman-Algorithmus [9] lässt sich effizi-

ent (mit $O(n \log n)$ Schritten bei $n$ Polygonknoten) feststellen, ob das Konturpolygon einfach, d. h. schleifenfrei ist, wodurch ein binäres Merkmal berechnet wird.

5. Die Intensitätsverteilung im Inneren des Polygonzuges sollte einen höheren Mittelwert haben als die Intensitätsverteilung um das Polygon herum. Daher werden die Verteilungen der Intensitäten zufällig ausgewählter Pixel innerhalb des Polygons und auf einem umschreibenden Kreis mit dem parameterfreien Mann-Whitney-Test (siehe Abschnitt 7.2.1 auf S.103) verglichen. Die Teststatistik $U$ dient als Merkmal.

In der Abbildung 6.22 sind die Merkmale für einen Teil der Stichprobe aufgetragen. Zur Lösung des Messpunktklassifikationsproblems sind die gleichen Bildstichproben und die gleichen Klassifikatortypen wie beim Eigenspot-Ansatz geeignet (siehe Abschnitt 6.5.3 S. 85). Die Robustheit der aktiven Kontursegmentierung gegen Verschiebungen der initialen Kontur ist erheblich besser als die Robustheit des Eigenspot-Verfahrens bezüglich der Wahl der Bildausschnitte, so dass das Verrauschen der Ausschnittpositionen der Trainingsstichprobe für die Klassifikationsleistung praktisch keinen Unterschied macht. Anders als beim Eigenspot-Ansatz ist die Klassifikationsleistung des Polynomklassifikators (88,1%) hier etwas besser als die der SVM (87,2%), wenn die automatisch aus Gittersegmentierungen erzeugten Stichproben benutzt werden.

Die Ursache dürfte darin liegen, dass die Klassen im Raum der Konturmerkmale schon wegen der geringeren Merkmalsdimension (9 bei Eigenspots, hier 5) stärker überlappen als beim Eigenspot-Ansatz. Das SVM-Training ist mit solchen Stichproben besonders schwierig, weil viele der sog. Slack-Variablen benutzt werden müssen, mit denen die Klassenüberlappung behandelt wird. Das Training konvergiert hier langsamer als bei der Eigenspot-Klassifikation. Auf der manuell klassifizierten Stichprobe mit geringerer Überlappung der Klassen im Merkmalsraum arbeitet dagegen die SVM genauer. Der Effekt bleibt erhalten, wenn Teilstichproben zwischen Trainings- und Testmenge ausgetauscht werden.

Es gibt zwei Erklärungsansätze für die schlechtere Leistung der SVM: Erstens ist die Stichprobe ausreichend groß um die klassenspezifischen Verteilungen der Merkmale auch an den Rändern der Klassengebiete recht genau wiederzugegeben. Damit sind für den Polynomklassifikator als Approximation des Bayesklassifikators optimale Voraussetzungen gegeben. Überlappungen der Klassen in den Trainingsdaten können sogar nützlich sein, wenn dadurch die wirklichen Verteilungen der Merkmale wiedergegeben werden, denn die Abtastung des Merkmalsraumes ist dann vollständiger als wenn Teile des Merkmalsraumes gar nicht in den Trainingsdaten repräsentiert sind, wodurch der Generalisierungsfehler geringer bleiben sollte. Der Vorteil der SVM, Klassengrenzen mit maximalem Abstand zu den Bespiel-Merkmalsvektoren festzulegen, kommt hier also gar nicht erst zum tragen. Zweitens könnten der verwendete Kern (Polynomkern) oder andere Parameter der verwendeten SVM-Implementation (libSVM [27]) für dieses Problem ungeeignet sein. Burges sieht die Notwendigkeit problemspezifisch geeignete Kerne zu finden als wesentliches Hindernis beim praktischen Einsatz der SVM an [24].

Mit der automatisch generierten Stichprobe, die mit dem Polynomklassifikator bisher besser bearbeitet werden kann, werden etwas bessere Ergebnisse der MZF-Gitter-

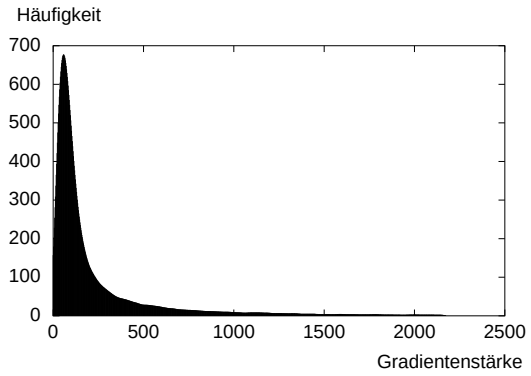



Abbildung 6.21: Ein typisches Histogramm der Gradientenstärke eines Mikroarraybildes.

segmentierung erreicht, weil vermutlich die subjektive, manuelle Stichprobenklassifikation im Sinne des MZF-Modells nicht optimal ist. Daher wird für die Klassifikation der aktiven Konturen der Polynomklassifikator benutzt.

### 6.6.6 Klassifikation mit $E_c$-Schwellwert

Alternativ zum Lernen von Klassifikatoren aus Stichproben kann auch ein heuristischer modellbasierter Ansatz gewählt werden. Hierzu betrachte man das Histogramm der Gradientenstärke in Abbildung 6.21. Die Form des Hintergrundpeaks in dieser Verteilung ist bei vielen verschiedenen Bildern sehr gut erhalten, seine absolute Lage ist aber von der Bildintensität abhängig. Die Gradientenstärken der Hintergrundpixel des Bildes ohne signifikante Kanten liegen zum großen Teil in dem Hintergrundpeak. Daher befindet sich ein Konturmodell sehr wahrscheinlich auf signifikanten Objekträndern, wenn die mittlere Gradientenstärke entlang des Polygons größer als der Wert am rechten Fuß des Hintergrundpeaks im Gradientenstärkehistogramm ist. Noch besser als die mittlere Gradientenstärke kann die Bildenergie $E_c$ betrachtet werden, weil dieser Wert kleiner als die mittlere Gradientenstärke ist, wenn die Kontur auf Rauschpixeln mit zufälliger Gradientenorientierung liegt. Der Schwellwert wird mit dem Verfahren „Einseitige Varianzschätzung“ (siehe S. 56) mit dem Parameter $k = 4$ bestimmt, d. h. der Schwellwert liegt etwa 2 Peakbreiten rechts vom Maximum des Histogramms.

Dieses Verfahren hat gegenüber den trainierten Klassifikatoren zwei große Vorteile: Es erfordert keine klassifizierte Stichprobe und es kalibriert anders als die statischen Klassifikatoren bei jedem einzelnen Bild die globale Intensität.

Die Abbildung 6.23 stellt die Ergebnisse der Messpunktdetektion mit aktiven Konturen und Eigenspot-Ansatz gegenüber. Bei diesem Beispiel ist der Eigenspot-Ansatz nicht so sensitiv wie die aktiven Konturen, produziert aber weniger falsch Positive. Beide Methoden erkennen deutlich mehr Messpunkte als die (hinterlegte) Schwellwertsegmentierung (man beachte mehrere Messpunkte umfassende Regionen).

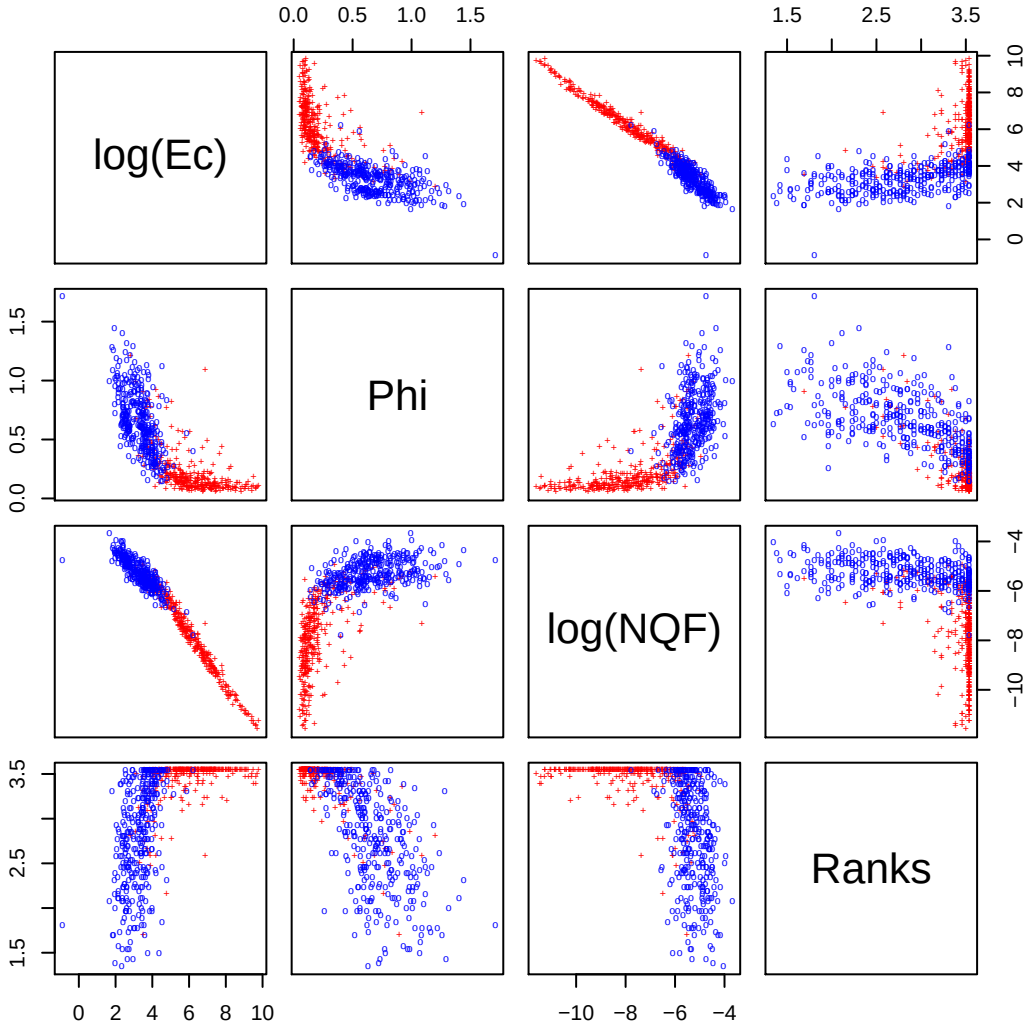



Abbildung 6.22: Paarweise gegeneinander aufgetragene Merkmale der Kontursegmentierungen von 200 Stichprobenelementen. Die Merkmalsvektoren von Messpunkten sind mit roten Kreuzen dargestellt, die der Gegenbeispiele mit blauen Kreisen.

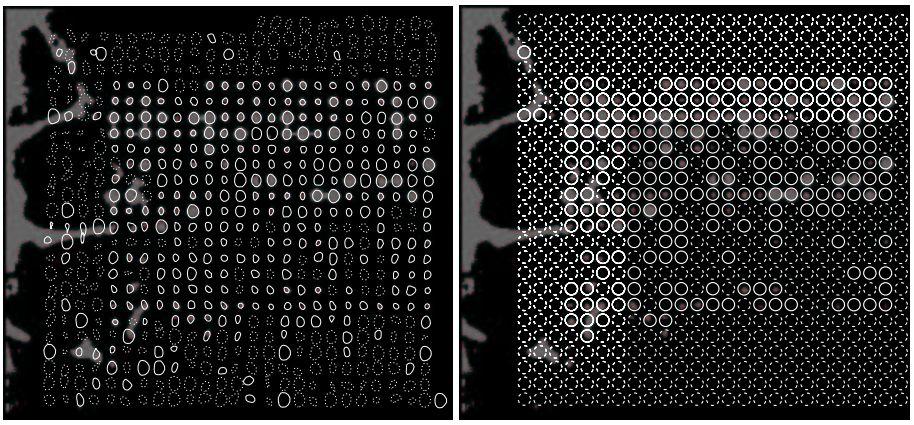


Abbildung 6.23: Links:Das Ergebnis von Kontursegmentierung und Klassifikation auf dem Beispielbildausschnitt aus der Apo-A1 Stichprobe. Als Messpunkte klassifizierte Konturen sind durchgehend dargestellt, Nicht-Messpunkte als unterbrochene Linien. Die rechte Grafik zeigt zum Vergleich die Klassifikationsergebnisse des Eigenspot-Verfahrens. Die Messpunkte sind darin als Kreise dargestellt, weil das Eigenspotverfahren keine Regionen berechnet. Hinterlegt ist jeweils das Binärbild der Schwellwertsegmentierung. Bei dessen Vergleich mit den neuen Detektionsverfahren muss noch berücksichtigt werden, dass die mehrere Messpunkte umfassenden Regionen beim Schwellwertverfahren wegfallen, weil sie größer als eine Gitterzelle sind.

## 6.7 Zusammenfassung

Dieses Kapitel hat die Komponenten der Gittersegmentierung behandelt. Das Kernstück ist das MZF-Modell, in dem typische Eigenschaften der Gitteranordnungen der Mikroarraybilder modelliert sind. Markov-Zufallsfelder beschreiben Wahrscheinlichkeiten abhängiger Zufallsvariablen, die mit ihren Abhängigkeiten die Knoten und Kanten eines ungerichteten Graphen bilden. Hier wird das MZF-Modell zur Formalisierung der in Abschnitt 3.4.2 aufgelisteten Eigenschaften der Gittersegmentierung eingesetzt. Die Gittersegmentierung wird dadurch als diskretes Optimierungsproblem formuliert, in dem aus Hypothesen für die Platzierung der Gitter eine global optimale Auswahl zu treffen ist.

Die Bestimmung der Hypothesen stützt sich wesentlich auf die lokale Periodizität der Gitteranordnung. Sie ermöglicht die automatische Kalibrierung der Bildauflösung und die Erkennung geeigneter Regionensegmentierungen. Verteilungsmodelle der Abstände zwischen benachbarten Regionenschwerpunkten bilden die Werkzeuge zur Nutzung dieser invarianten Eigenschaft.

Mit Hilfe der Verkettung von Messpunktobjekten in der Gitternachbarschaft wird die Gitterrotation geschätzt. Aus Achsenprojektionen der verketteten Objekte liest man die Hypothesen der Gitterplatzierung ab.

Die Cliquepotentiale des MZF-Modells, die die Wahrscheinlichkeit der Hypothesen bewerten, basieren auf der Detektion von Messpunkten an den Gitterknoten. Die einfachen Schwellwertverfahren zur Regionensegmentierung sind hierzu nicht immer Leistungsfähig genug, weshalb zwei verbesserte Ansätze untersucht werden: Der Eigenspot-Ansatz und die aktive Kontursegmentierung. Im Eigenspotverfahren wird durch Eigenwertzerlegung der Kovarianzmatrix einer Stichprobe von Messpunktbildern eine charakteristische Repräsentation erzeugt, die zur Merkmalsberechnung dient. Die aktive Kontursegmentierung setzt ein Flexibilitätsmodell der Messpunktränder ein, das an das Gradientenbild angepasst wird. Die Anpassung muss kontinuierlich formuliert werden, da Messpunkte sehr verschieden groß sein können. Aus den Modellparametern werden Merkmale berechnet, die wiederum zur Klassifikation mit überwacht gelerntem Klassifikator oder aber mit einem modellbasierten Verfahren, das ohne klassifizierte Stichprobe auskommt, eingesetzt werden. Am Beispiel in Abbildung 6.23 erkennt man die Überlegenheit der beiden neuen Messpunktdetektionsverfahren über das Schwellwertverfahren.

# 7 Quantitative Bildauswertung

Dieses Kapitel behandelt Methoden für die Auswertung zweifarbiger Bilder aus differentiellen Expressionsexperimenten mit konkurrierender Hybridisierung (siehe 3.3.4). Gedruckte Mikroarrays werden fast ausschließlich in diesem Typ von Experimenten verwendet. Die quantitative Bildauswertung hat das Ziel, die Intensitäten und die Verhältnisse der Intensitäten in den verschiedenen Bildkanälen jedes Messpunktes zu messen. Die Grundlage der weiteren Überlegungen hierzu bildet ein Modell des Fluoreszenzbildsignals der Mikroarraybilder.

## 7.1 Modell der Bildintensität

Folgendes Modell der Intensität der Fluoreszenzbilder $f^{\text{rot}}$ und $f^{\text{grün}}$ wird angenommen:

$$f^{\text{rot}} = f_F^{\text{rot}} + k^{\text{rot}} + s^{\text{rot}} \tag{7.1}$$

$$f^{\text{grün}} = f_F^{\text{grün}} + k^{\text{grün}} + s^{\text{grün}} \tag{7.2}$$

$f_F$ bezeichnet die Fluoreszenzintensität des jeweiligen Farbstoffs, also das Nutzsignal. $k$ ist positiv, variiert nur langsam über das Bild und erfasst Störfaktoren wie konstante additive Fehler (Offset) der Intensitätsmessung oder Glasfluoreszenz. $s$ ist ebenfalls ein positiver Term, der (körnige) Verunreinigungen der Arrayoberfläche beschreibt. $s$ kann wegen der unterschiedlichen Oberflächenbehandlung im Inneren und außerhalb der Messpunkte sehr verschiedene Eigenschaften besitzen. Die $f_F$ setzen sich aus der kanalunabhängigen Dichte $p$ der gedruckten Sonden-DNA und einem kanalspezifischen, pro Messpunkt konstanten Faktor $c$ zusammen, der die zu messende Transkriptmenge und weitere, später durch die Normalisierung zu korrigierende Faktoren enthält (vgl. S.28):

$$f_F^{\text{rot}}(\vec{x}) = p(\vec{x})c^{\text{rot}} \tag{7.3}$$

$$f_F^{\text{grün}}(\vec{x}) = p(\vec{x})c^{\text{grün}} \tag{7.4}$$

Die Gleichungen (7.3)und (7.4) beschreiben essentiell das Prinzip der konkurrierenden Hybridisierung. Für jeden Messpunkt ist das Verhältnis $R = c^{\text{grün}}/c^{\text{rot}}$ zu messen, aus dem durch die Normalisierung das Verhältnis der Transkriptmengen des Experiments bestimmt wird.

Aus den Bildern werden die mittleren Intensitäten $I^{\text{Rot}}$ und $I^{\text{Grün}}$ auf den Messpunktregionen $M$ geschätzt:

$$
\begin{aligned}
I^{\text{Rot}} &= \frac{1}{|M|} \sum_{(i,j)\in M} f_{ij}^{\text{rot}} \\
I^{\text{Grün}} &= \frac{1}{|M|} \sum_{(i,j)\in M} f_{ij}^{\text{grün}}
\end{aligned} \tag{7.5}
$$

$f_{ij}^{\text{rot}}$ und $f_{ij}^{\text{grün}}$ sind die Intensitäten der Pixel $(i, j)$ der nach Gl. (7.2) modellierten Fluoreszenzbilder. In den gemessenen Intensitäten sind noch die Hintergrundanteile $I_{bg}$ enthalten, die sich aus den Komponenten $k$ und $s$ des Intenitätsmodells zusammensetzen. Die unten noch einmal angegebene Gleichung (3.2) aus Abschnitt 3.3.4 über die differentielle Expressionsanalyse drückt die Schätzung des Verhältnisses $R$ eines Messpunktes durch dessen geschätzte Farbstoffluoreszenzintensitäten $I_F^{\text{Rot}}$ und $I_F^{\text{Grün}}$ in beiden Bildkanälen aus.

$$
R = \frac{I_F^{\text{Grün}}}{I_F^{\text{Rot}}} \approx \frac{I^{\text{Grün}} - I_{bg}^{\text{Grün}}}{I^{\text{Rot}} - I_{bg}^{\text{Rot}}} \tag{3.2}
$$

Dieser Ansatz ist sinnvoll, weil Nenner und Zähler von (3.2) linear bezüglich des jeweiligen $c$ aus den Gln. (7.3) und (7.4) sind.

Die Anwendung der Gleichung (3.2) setzt voraus, dass die Intensitätsmessung nach Gleichung (7.5) auf korrespondierenden Regionen $M$ der beiden Kanäle erfolgt, damit die in Gl. (7.3/7.4) postulierte Gleichheit von $p(\vec{x})$ in beiden Kanälen gegeben ist. Die Korrespondenz kann jedoch durch Verschiebungen der Kanäle gestört sein, die bei der Bildaufnahme entstehen (siehe S. 35). Die Verschiebung wird deshalb bei der Gittersegmentierung an Messpunktobjekten mit Regionen in beiden Kanälen geschätzt (siehe S. 63) und kann daher durch interpolierte Neuabtastung eines der beiden Fluoreszenzbilder näherungsweise korrigiert werden.[1]

Die Schätzung der Hintergrundintensitäten $I_{bg}^{\text{Grün}}$ und $I_{bg}^{\text{Rot}}$ ist sehr schwierig, denn, wie schon erwähnt, ist das Hintergrundsignal innerhalb und außerhalb der Messpunkte wegen der Behandlung der Arrayoberfläche nicht unbedingt gleich. Im Inneren der Messpunkte kann aber das Hintergrundsignal offensichtlich nicht isoliert gemessen werden. Schätzungen der Hintergrundintensität in der Umgebung der Messpunkte führen bei kleinem Nutzsignal leicht zu negativen Schätzwerten für die Farbstoffintensität. Eine häufige Ursache dafür sind „schwarze Löcher“ (siehe S. 31), also Messpunkte, die tatsächlich geringere Intensität als ihre Umgebung haben. Das Problem der schwarzen Löcher kann man mit Bildverarbeitungsmethoden allein nicht lösen. Eine andere Fehlerquelle ist die Segmentierung der Messpunkte und des Hintergrundes. Wenn sich in einer vermeintlichen Hintergrundregion Teile von Messpunkten befinden, können die Intensitätsschätzwerte leicht zu hoch ausfallen.

Fehler der Hintergrundkorrektur wirken sich auf die Schätzung der Gesamtintensität stärker aus, wenn die Signalregion größer als der tatsächliche Messpunkt ist. Die Region enthält dann mehr Pixel, deren Hintergrundanteil korrigiert werden muss, aber nicht zum Nutzsignal beitragen.

[1] Weglassen der Verschiebungskorrektur vergrößert den Varianzkoeffizienten in Replikatgruppen um bis zu 10% [59].

Dies sind zwei wichtige Gründe, weshalb die Signalsegmentierung für die quantitative Auswertung pixelgenau die Messpunkträndern finden sollte. Außerdem ist es entscheidend, die Hintergrundintensitätsschätzung gegen Fehlsegmentierungen robust zu machen.

## 7.2 Signalsegmentierung

Ein weit verbreitetes formadaptives Verfahren für die Messpunktsegmentierung ist die Mann-Whitney-Segmentierung [28], auf die der folgende Abschnitt näher eingeht. Die bereits im Abschnitt 6.6 vorgestellte aktive Konturesegmentierung scheint ebenfalls sehr geeignet, weil mit dem kontinuierlichen Konturmodell theoretisch sogar subpixelgenau segmentiert werden kann.

### 7.2.1 Mann-Whitney-Segmentierung

Der in diesem Abschnitt behandelte Algorithmus bestimmt mit dem parameterfreien Mann-Whitney-Verteilungstest aus den Intensitäten einer Menge vermuteter Signalpixel $S$ eine Teilmenge, deren Verteilung signifikant von der Intensitätsverteilung einer Menge von Hintergrundpixeln $H$ verschieden ist.

Damit das Verfahren angewendet werden kann, muss es also schon eine Vorsegmentierung der vermuteten Signalregion und einer Hintergrundregion geben. Dazu verwendet man die Ergebnisse der Gittersegmentierung.

Das Verfahren testet die Verteilung der Hintergrundstichprobe gegen Ausschnitte der sortierten Liste der Intensitäten der vermuteten Signalpixel $S$. Der Ausschnitt wird so lange von niedrigen zu höheren Intensitäten verschoben, bis der Test signifikant verschiedene Intensität anzeigt. Der Median der Intensitäten im zuletzt gewählten Ausschnitt liefert dann einen geeigneten Schwellwert zur Segmentierung der Signalpixel.

#### Mann-Whitney-Test

Der zum Vergleich der Intensitätsstichproben verwendete Mann-Whitney-Test beruht ausschließlich auf Betrachtung der Rangordnung der Elemente beider Stichproben. Allgemeine Darstellungen über diesen Test findet man z. B. bei Sprent [99] und in vielen anderen Lehrbüchern. Die Teststatistik $U$ wird in ihrer klassischen Form durch die Anzahl von Paaren aus Elementen der Stichproben $A$ und $B$ definiert, deren $A$-Element kleiner als das $B$-Element ist:

$$U = \#\{(a, b)|a < b, a \in A, b \in B\} \tag{7.6}$$

$U$ kann auch mit Hilfe von Rangsummen berechnet werden. Diese Variante heißt Wilcoxon-Version des Mann-Whitney-Tests. Es gibt eine effiziente Methode zur direkten Berechnung der Mann-Whitney-Teststatistik ohne den Umweg über Rangsummen, die hier im Algorithmus 6 verwendet wird.

Wenn die beiden Stichproben schon sortiert sind, kann man einfach abzählen, wieviele Elemente aus $A$ vor jedem Element aus $B$ liegen und dadurch die Anzahl der Paare ermitteln, die ein kleineres Element aus $A$ und ein größeres aus $B$ enthalten. $U$ kann damit bei schon vorsortierten Stichproben mit linearer Effizienz berechnet

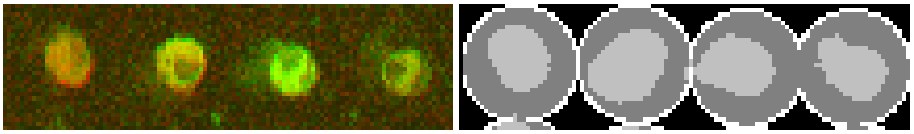


Abbildung 7.1: Links ein Ausschnitt aus einem Mikroarraybild der NMHy-Stichprobe und rechts die vermutete Signalregion (grau), die Hintergrundregion (weiss) und die segmentierte Signalregion (hellgrau) mit dem kritischen Wert $C = 5$, d.h. mit ca. 20% Signifikanz

werden. Das Beispiel in Tabelle 7.1 verdeutlicht dieses Prinzip. In der speziellen Anwendung des Tests zur Messpunktsegmentierung muss in der Regel mehrfach getestet werden, wobei $A$ konstant bleibt und $B$ eine Subsequenz einer längeren sortierten Liste ist.

Der Algorithmus 6 umfasst das gesamte Verfahren zur Schwellwertberechnung, also das mehrfache Testen gegen Teillisten aus den sortierten Intensitätswerten der vermuteten Signalpixel $I^S$ gegen die konstante Hintergrundmenge $I^H$, das in Abbildung 7.2 veranschaulicht ist. Die vermutete Signalregion wird darin mit einem Kreis um das Messpunktzentrum $(c_x, c_y)$ aus der Gittersegmentierung festgelegt, dessen Radius $R$ so gewählt wird, dass er gerade in eine Gitterzelle passt. Die Hintergrundstichprobe $I^H$ wird auf dem Kreisrand zufällig ausgewählt. In Abbildung 7.1 rechts ist die vermutete Signalregion grau und der Kreisrand aus dem $I^H$ gezogen wird weiß dargestellt.

Wenn bei einem dunklen Messpunkt die Intensität auf $S$ wie die Hintergrundintensität verteilt ist, existiert kein Intervall in $I^S$, das signifikant höhere Werte enthält als die Hintergrundstichprobe $I^H$. Für dunkle Messpunkte kann daher mit der Mann-Whitney-Segmentierung keine Region angegeben werden.

Die Segmentierung muss in beiden Bildkanälen getrennt durchgeführt und die größere Ergebnisregion benutzt werden, da sonst die besonders interessanten Messpunkte, die nur in einem der Kanäle hell sind, nicht richtig segmentiert werden könnten.

Chen schlägt für die Stichprobengröße $l$ den Wert 8 vor, da ab diesem Stichprobenumfang die Teststatistik $U$ in guter Näherung mit den Parametern $\mu = \frac{1}{2}l^2$ und $\sigma = \sqrt{l^2(l+1)/12}$ normalverteilt ist [28]. Erfahrungsgemäß sollte der kritische Wert so eingestellt werden, dass der Test nicht zu sensitiv auf Unterschiede der Stichproben reagiert (d. h. sog. $\beta$-Fehler oder Fehler 2. Art sollten vermieden werden, Nullhypothese ist die Gleichheit der Verteilung), weil dadurch zu kleine Schwellwerte berechnet und starke Fehlsegmentierungen verursacht werden. Mit dem Test bei 10%-Signifikanzniveau (kritischer Wert $C$=5 für l=8) bekommt man befriedigende Ergebnisse. Es gibt Modifikationen des Tests, die wertgleiche Elemente in den beiden Stichproben („Ties“) genauer behandeln. Sie erhöhen die Sensitivität und führen daher nicht zu robusterer Segmentierung.

Wenn auch „schwarze Löcher“ segmentiert werden sollen, muss der zweiseitige Test verwendet werden, der einem zusätzlichen einseitigen Test mit vertauschten Stichproben entspricht (man testet $A > B$ und $B > A$). Aus der Definintion der Teststatistik Gl. (7.6) folgt, dass die Teststatistik $U'$ bei vertauschten Stichproben $l^2 - U$ ist, denn es gibt insgesamt $l^2$ Paare. Dadurch kann der zweiseitige Test sehr einfach realisiert werden ($U^*$ im Algorithmus 6).

**Algorithmus 6** Mann-Whitney-Segmentierung

/* Parameter: $(c_x, c_y)$: Mittelpunkt der vermuteten Signalregion
$R$: Kreisradius der vermuteten Signalregion
$l$: Stichprobengröße des Tests
$C$: kritischer Wert des Tests */

$I^h \leftarrow \{f_{ij} | (R)^2 \leq (i - c_x)^2 + (j - c_y)^2 \leq (R)^2\}$
$I^H \leftarrow \{l$ zufällig gezogene Elemente aus $I^h\}$
$I^S \leftarrow \{f_{ij} | (i - c_x)^2 + (j - c_y)^2 \leq R^2\}$
Sortiere $I^S$ und $I^H$
$t \leftarrow 0$ // Anfang des zu testenden Intervalls in $I^S$
**repeat**
  $k \leftarrow t$
  $U \leftarrow 0, seenS \leftarrow 0$
  **for** $i \in \{1 \ldots l\}$ **do**
    **while** $(j \leq \#I^S) \wedge (I^S_k \leq I^H_i)$ **do**
      $seenS \leftarrow seenS + 1$
      $k \leftarrow k + 1$
    **end while**
    $U \leftarrow U + s$
  **end for**
  $U^* \leftarrow \min(U, l^2 - U)$
  $t \leftarrow t + 1$ // $I_H$ zu höheren Intensitäten verschieben (siehe Abb. 7.2)
**until** $(U^* \leq C) \vee (t \geq \#I^S - l)$

(Zur Übersicht ohne Fallunterscheidungen für die Segmentierung „schwarzer Löcher“)

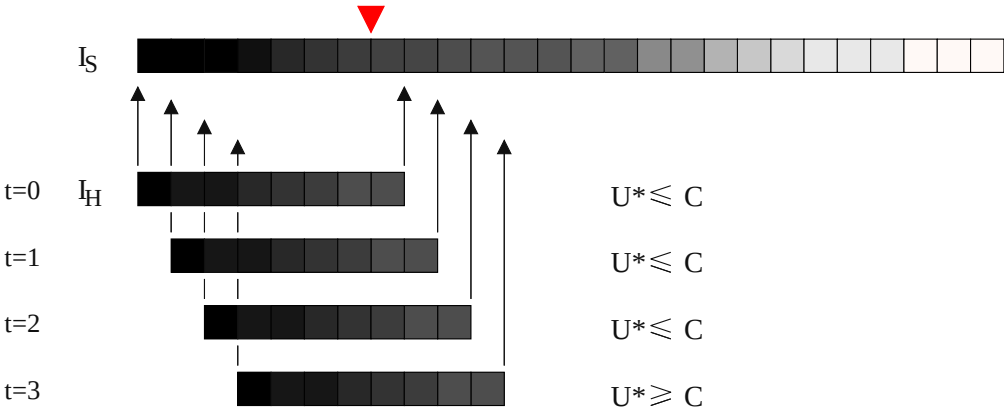



Abbildung 7.2: Die Schwellwertbestimmung mit dem Mann-Whitney-Test. Das Stichprobenintervall $I^S_k$ in der sortierten Liste der vermuteten Signalpixelintensitätswerte $I^S$ (schwarze Pfeile) von den dunkelsten Werten so weit zu helleren Werten verschoben, bis der Test signifikant höhere Intensität im Intervall als in der Hintergrundintensitätsstichprobe $I_H$ anzeigt. Der rote Pfeil markiert den berechneten Intensitätsschwellwert. Zu den Variablenbezeichnungen siehe auch Algorithmus 6.

| i | k | H | S | seenS | U |
|---|---|---|---|---|---|
| - | 0 | | | 0 | 0 |
| 1 | 0 | • | . | 0 | 0 |
| 2 | 0 | • | . | 0 | 0 |
| 3 | 0 | • | . | 0 | 0 |
| | 1 | . | ○ | 1 | |
| | 2 | . | ○ | 2 | |
| | 3 | . | ○ | 3 | |
| 4 | 3 | • | . | 3 | 3 |
| | 4 | . | ○ | 4 | |
| | 5 | . | ○ | 5 | |
| 5 | 5 | • | . | 5 | 8 |
| 6 | 5 | • | . | 5 | 13 |
| | 6 | . | ○ | 6 | |
| | 7 | . | ○ | 7 | |
| 7 | 7 | • | . | 7 | 20 |
| 8 | 7 | • | . | 7 | 27 |
| | 8 | . | ○ | 8 | |
| | 8 | | | 8 | 35 |

Tabelle 7.1: Ein Rechenbeispiel zum (einseitigen) Mann-Whitney-Test mit zwei achtelementigen Stichproben. Es ist der erste Ablauf der FOR-Schleife von Algorithmus 6 gezeigt, die die Teststatistik $U$ berechnet. Die Spalten H und S stellen die geordneten Elemente der beiden Stichproben dar. Der Schleifenzähler $i$ zählt über die Elemente von H. Immer wenn Elemente aus S kleiner als das nächste abzuarbeitende Element aus H sind, tritt der Algorithmus in die innere While-Schleife ein (grau unterlegte Zeilen der Tabelle) und die Variable seenS zählt mit, wieviele Elemente aus S passiert werden. Wenn dann das nächste H-Element abgearbeitet wird, weiß man, wieviele kleinere Elemente es in S gibt, also auch wie viele Paare es gibt, die aus dem aktuellen H-Element und einem kleineren S-Element bestehen. Man summiert in der Variable $U$ bei jedem besuchten H-Element die Anzahl der schon passierten S-Elemente auf und berechnet so die Teststatistik, ohne tatsächlich alle Paare zu bilden.

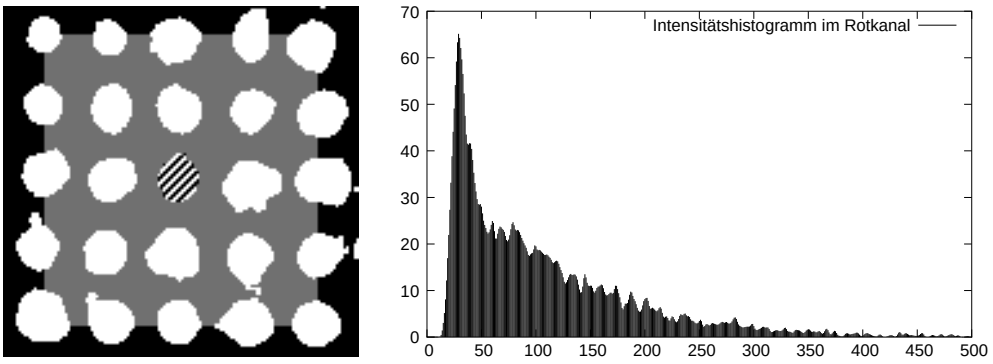



Abbildung 7.3: Links: Auf dem grauen Bildbereich wird die Hintergrundintensität zu der schraffierten Messpunktregion bestimmt. Rechts: Das Histogramm auf der Hintergrundregion. Der lang ausgezogene rechte Verteilungsschwanz, der auch Verunreinigungspartikel und verwischte Messpunkträndern enthält, hat wesentlich stärkeren Einfluss auf den Mittelwert oder Median als auf den Modus der Verteilung.

## 7.3 Hintergrundschätzung

Die zweite Komponente der quantitativen Bildauswertung neben der Signalsegmentierung ist das Schätzverfahren für die Hintergrundintensitäten $I_{\mathrm{bg}}$.

Das Verfahren muss robust gegen Segmentierfehler sein und soll möglichst wenig durch größere Verunreinigungen zwischen den Messpunkten gestört werden, denn wegen der verschiedenen Oberflächeneigenschaften innerhalb und außerhalb der Messpunkte sind Verunreinigungen außen meist nicht bedeutsam. „Überkorrektur“ des Hintergrundes muss vermieden werden, denn sie führt zu unbrauchbaren, negativen Messwerten.

Unter diesen Anforderungen scheint die Schätzung des Hintergrundanteils einer Messpunktintensität durch den häufigsten Intensitätswert (Histogrammmaximum oder Modus) in der Messpunktumgebung am besten geeignet, weil die konstanten Komponenten $k$ des Hintergrundsignals ausgeprägte Peaks im Histogramm erzeugen. Das Beispiel Abbildung 7.3 zeigt, dass der ansonsten häufig benutzte Median der Intensität in der Messpunktumgebung nicht so robust gegen großflächige Verunreinigungen ist wie der Modus. Bei dem Beispielmesspunkt haben Mittelwert, Median und Modus der Intensität auf der Hintergrundregion die Werte $106.8$, 81 und 29. Als lokale Hintergrundregion dient ein Fenster mit einer Kantenlänge von vier Gitterzellen um den jeweiligen Messpunkt ohne die darin enthaltenen Messpunktregionen, wie es in Abbildung 7.3 gezeigt ist.

Im Programm „Scanalyze“ wird eine sehr ähnlich definierte lokale Hintergrundregion benutzt, die Hintergrundintensität wird aber durch den Median der Intensitäten auf der Region geschätzt.

Plausibler als jede Hintergrundkorrektur anhand der Messpunktumgebung wäre ein Schätzverfahren, das nur Messpunktintensitäten benutzt. Dazu müssten z. B. Negativkontrollmesspunkte auf das Array gedruckt werden, deren Sequenz in den untersuchten Proben nicht vorkommt und die daher das geringstmögliche Signal zeigen sollten. Eine andere Möglichkeit sind Messpunkte von Verdünnungsreihen. Durch Regressi-

onsanalyse der Messwerte kann ebenfalls der Hintergrund geschätzt werden [108]. Solche Verfahren sind offensichtlich vom Arrayentwurf abhängig und daher im Allgemeinen nicht anwendbar.

Ein weiteres, allein mit Hilfe der Bilddaten kaum lösbares Problem stellt die Kreuzhybridisierung dar. Sie verursacht ein sequenzabhängiges Hybridisierungssignal, das auf andere Weise geschätzt werden muss.

## 7.4 Verhältnisberechnung und Qualitätsmerkmale

In Gleichung (3.2) werden zunächst die mittleren Intensitäten beider Kanäle bestimmt und dann deren Verhältnis gebildet. Weil aus dem Modell der konkurrierenden Hybridisierung die lineare Abhängigkeit der Intensitäten korrespondierender Messpunktpixel der beiden Bildkanäle folgt, kann das Verhältnis prinzipiell auch an jedem einzelnen Pixel bestimmt und dann eine Gesamtschätzung ermittelt werden. Im Idealfall sollte das Intensitätsverhältnis überall auf dem Messpunkt konstant sein, d. h. ein Messpunkt muss überall die gleiche Farbe haben.

Die Schätzung des Gesamtverhältnisses aus den Verhältnissen einzelner Pixelintensitäten heißt Pixel-by-Pixel-Ansatz. Diese Methode setzt eine sehr gute Verschiebungskorrektur voraus, damit die Korrespondenz der Kanäle sichergestellt ist. Sie hat aber den Vorteil, dass mit der Varianz der Pixelintensitätsverhältnisse auf dem Messpunkt ein Qualitätskriterium mitgeliefert wird. Ein Mittelweg zwischen Gesamtintensitätsverhältnis und Pixel-by-Pixel-Ansatz besteht in der Aufteilung der Signalregion in Sektoren, in denen je ein Verhältniswert berechnet wird. Die Sektoren sind größer als einzelne Pixel, so dass die Robustheit gegen Verschiebungen besser als beim reinen Pixel-by-Pixel-Ansatz ist. Trotzdem kann die Varianz des Verhältnisses geschätzt werden. Die Gleichung (7.8) beschreibt allgemein die Verhältnisberechnung. Je nach Definition der Sektoren $S_k$ (ganze Region $M$, Teile davon, einzelne Pixel) erhält man die verschiedenen Varianten.

$$R_k = \frac{\sum_{(i,j)\in S_k}\left(f^{\text{grün}}_{(i,j)} - I^{\text{grün}}_{\text{bg}}\right)}{\sum_{(i,j)\in S_k}\left(f^{\text{rot}}_{(i,j)} - I^{\text{rot}}_{\text{bg}}\right)} \tag{7.7}$$

$$R = \langle R\rangle_k \tag{7.8}$$

$$\text{mit} \quad M = \bigcup_k S_k, \quad S_i \cap S_j = \emptyset \Leftrightarrow i \neq j$$

Als weiteres Qualitätskriterium neben der Varianz der $R_k$ bietet sich der lineare Korrelationskoeffizient der Pixelintensitäten an.

Ähnliche Vorschläge für Qualitätskriterien finden sich in der Arbeit von Brown und anderen [21]. Sie führen die „Spot Ratio Variability“ (SRV) ein, die sie durch die Varianz der Pixelintensitätsverhältnisse dividiert durch das Gesamtverhältnis definieren. In dem Artikel wird gezeigt, dass durch gewichtetes Mitteln der Verhältnismessungen replizierter Messpunkte die Genauigkeit verbessert werden kann. Die Kehrwerte der SRV sind dazu geeignete Gewichtungsfaktoren.

Im Allgemeinen werden die verschiedenen Qualitätsmaße dazu verwendet, um nach der Bildauswertung mit benutzerdefinierten Auswahlregeln unbrauchbare Messpunkte auszufiltern.

# 8 Ergebnisse

Die beiden wesentlichen Ziele, die mit den vorgestellten Methoden erreicht werden sollen, sind die möglichst allgemeine Anwendbarkeit der Gittersegmentierung für die Auswertung von verschiedenen Mikroarrays und die Reproduktion oder Verbesserung der quantitativen Bildauswertung mit interaktiven Werkzeugen durch automatisierte Verfahren. Dieses Kapitel beginnt mit Überlegungen zu geeigneten Evaluationsmethoden, an die sich die Darstellung der Ergebnisse und des Ressourcenbedarfs des Gesamtsystems anschließen.

## 8.1 Methoden zur Systemevaluation

### 8.1.1 Gittersegmentierung

Die Gittersegmentierung wird durch den Vergleich der Ergebnisse mit manuell validierten Segmentierungen evaluiert. Für die Messpunktdetektion und die MZF-Energieminimierung gibt es verschiedene Verfahren, die prinzipiell beliebig miteinander kombiniert werden können. Als Energieminimierungsalgorithmen werden „Highest Confidence First“ und „Local Highest Confidence First“ untersucht. Das „Simulated Annealing“-Verfahren wird wegen der äußerst langen Rechenzeit getrennt betrachtet. Zur Messpunktdetektion werden das Schwellwertverfahren, die Segmentierung mit dem aktiven Konturmodell und die Eigenspot-Methode betrachtet.

Es ist zu erwarten, dass sich die Verfahrenskombinationen abhängig von den Eigenschaften der Eingabedaten unterschiedlich verhalten. Die Stichprobe wird daher in einige Kategorien eingeteilt, die spezifische Probleme der Bildsegmentierung betreffen.

Als Evaluationskriterium dient jeweils der Anteil der korrekt segmentierten Messpunktgitter je Bild. Die Zahl der korrekt segmentierten Gesamtbilder ist weniger aussagekräftig, weil sie den praktisch sehr relevanten Unterschied zwischen einer Gittersegmentierung mit nur wenigen falsch platzierten Gittern und einer vollständig falschen Segmentierung nicht erfasst. Zudem lässt der Stichprobenumfang teilweise noch zu wünschen übrig, so dass über den Anteil korrekt segmentierter Gitter besser abgesicherte Aussagen möglich sind als über den Anteil korrekt segmentierter Bilder, denn der größte Teil der Stichprobe besteht aus Bildern mit 16 oder mehr Gittern.

Ein Gitter ist korrekt segmentiert, wenn sein Ursprung weniger als 10% der Gitterzellenkantenlängen von der interaktiv validierten Position abweicht und die Gitterachsen ebenfalls nicht mehr als 10% von den korrekten Werten verschieden sind.

Neben der der Segmentierungsleistung selbst wird die Abhängigkeit von den MZF-Parametern untersucht, um die Eignung der (wie in Abschnitt 6.4.3 beschrieben) heuristisch gewählten Werte zu prüfen. Dazu wird die Segmentierungsleistung unter Variationen der gewählten Parameterwerte gemessen.

Außerdem wird die Tauglichkeit der Restenergie an den MZF-Knoten als Rückweisungskriterium untersucht. Dazu werden die empirischen Verteilungen der Restenergie an korrekt und falsch segmentierten Gittern verglichen.

### 8.1.2 Quantitative Auswertung

Die Evaluation der quantitativen Bildauswertung ist schwieriger als die Evaluation der Gittersegmentierung, weil keine Referenzdatensätze mit vorgegebenen, korrekten Auswertungsergebnissen bekannt sind. Daten mit exakt bekannten Soll-Messwerten können kaum durch reale Mikroarrayexperimente beschafft werden, weil die Genexpression der Lebewesen von zu vielen und zu schwer kontrollierbaren Faktoren abhängig ist. Um trotzdem zu einer aussagekräftigen Evaluation zu kommen, werden die folgenden drei Untersuchungen durchgeführt:

#### Vergleich mit etablierten Werkzeugen

Die Plausibilität der mit dem hier beschriebenen System bestimmten Messwerte wird durch den Vergleich mit Werten aus verbreiteten Systemen geprüft. Für Vergleiche dieser Art werden grafische Methoden benutzt, denn Ausreißer und systematische Abweichungen verbieten häufig die Anwendung üblicher Korrelationsmaße für Daten mit gaussverteilten Fehlern [53, 67, 71, 107].

Die mit verschiedenen Methoden bestimmten Messwerte werden also direkt gegeneinander aufgetragen, wobei die Datenpunkte ideal auf der Diagonale des Koordinatensystems liegen sollten.

Da die Messfehler in der Regel bei kleinen Intensitäten größer sind, ist es aufschlußreich, zusätzlich das Verhältnis der mit verschiedenen Methoden gemessenen Intensitätsverhältnisse über der (mittleren) Intensität aufzutragen.

Durch die direkten Vergleiche ist nicht zu entscheiden, welche Methoden genauer arbeiten, solange es keine Kalibrierdaten gibt, bei denen die absoluten Sollwerte jedes Messpunktes oder die relative Größe der Werte verschiedener Punkte vorgegeben sind.

#### Konsistenz der Messwerte replizierter Messpunkte

Die Varianz von Intensitätsmesswerten replizierter Messpunkte (gleiche Sondensequenzen auf einem Array) liefert ein Kriterium für die Genauigkeit der Messverfahren. Die Varianz der Replikat-Messwerte wird auch durch Unregelmäßigkeiten bei der Arrayherstellung verursacht, aber zumindest relative Aussagen über die Güte der verglichenen Auswertungsverfahren sind möglich.

Der Ansatz birgt einige Probleme: Die Varianz der Messwerte in einer Gruppe von Replikaten hängt in aller Regel auch von den Messwerten selbst ab. Daher sollte nicht die Varianz selbst sondern der Varianzkoeffizient $cv$ (Quotient aus Varianz und Mittelwert) der Messwerte in Replikatgruppen betrachtet werden.

Außerdem ist die Konsistenz allein nicht aussagekräftig, weil die ausschließliche Betrachtung der Replikat-Varianzen systematische Fehler der Einzelmessungen nicht berücksichtigt. Das Konsistenzkriterium $cv$ bevorzugt Messungen eines positiven Signals $s$ mit konstantem, ebenfalls positivem systematischen Fehler $b$:

$$cv = \frac{Var(s+b)}{E(s+b)} = \frac{Var(s)}{E(s)+E(b)} < \frac{Var(s)}{E(s)}$$

Derartige systematische Fehler der Intensitätsmessung entstehen z. B. durch zu große Signalregionen, die zu größeren Anteilen von Hintergrundintensität in den Messwerten führen.

Der Extremfall ist ein „Messverfahren“, das immer den gleichen konstanten Wert liefert: Es würde unter dem reinen Konsistenzkriterium optimal abschneiden. Daher sollte also zusätzlich die Verteilung der Messwerte insgesamt betrachtet werden. Wenn ein Verfahren bei dem Konsistenzkriterium besser abschneidet, aber gleichzeitig die Intensitätsmesswerte oder ihre Varianz insgesamt sinken, ist keine gesicherte Aussage möglich. Ein Messverfahren liefert dagegen wahrscheinlich bessere Ergebnisse als ein anderes, wenn es gleichzeitig geringere Variabilität innerhalb der Replikatgruppen und größere Variabilität auf allen Messwerten eines Arrays erzeugt.

Betrachtet man anstelle der Intensitäten das Intensitätsverhältnis der zwei Bildkanäle, das ja hauptsächlich gemessen werden soll, so müssen dabei die Eigenheiten der nichtlinearen Verhältnisberechnung berücksichtigt werden. Es ist festzustellen, dass systematische, positive Fehler $b$ der Einzelintensitäten $r$ und $g$ das Verhältnis näher zum Wert 1 verschieben, denn mit einigen trivialen Umformungen zeigt man die folgenden Ungleichungen:

$$\frac{g+b}{r+b} \leq \frac{g}{r} \quad \Leftrightarrow \tfrac{g}{r} \geq 1 \tag{8.1}$$

$$\frac{g+b}{r+b} \geq \frac{g}{r} \quad \Leftrightarrow \tfrac{g}{r} \leq 1 \tag{8.2}$$

Bei unterschiedlichen Fehlern im Rot- und Grünkanal ergibt sich im Wesentlichen der gleiche Effekt. Daraus folgt, dass bei systematisch falscher Hintergrundschätzung die Varianz der Verhältnismesswerte insgesamt sinkt und somit die Aussage des Konsistenzkriteriums weniger klar ist.

Ein weiterer wichtiger Unterschied zur einfachen Intensitätsmessung besteht darin, dass die Fehler hier nicht linear auf die Messgröße wirken. Das Fehlerfortpflanzungsgesetz (8.3) für das Intensitätsverhältnis beschreibt linearisiert den Einfluss der durch die Signalsegmentierung verursachten Intensitätsmessfehler $\sigma$ auf den Verhältniswert [21, 92].

$$\sigma_R = \sigma_{\text{grün}} \frac{I^2_{\text{fg\_rot}}}{I^4_{\text{fg\_grün}}} + \frac{\sigma_{\text{rot}}}{I^2_{\text{fg\_grün}}} - 2\text{cov}_{\text{rot,grün}} \frac{I_{\text{fg\_rot}}}{I^3_{\text{fg\_grün}}} \tag{8.3}$$

Die hohen Potenzen im Fehlerfortpflanzungsgesetz zeigen, dass die Verhältnisbildung unter ungünstigen Umständen als kräftiger „Varianzverstärker“ wirkt. Dieser Fall tritt insbesondere bei Messpunken mit starken Unterschieden der beiden Intensitäten ein, also gerade dann, wenn das Experiment einen Effekt gezeigt hat. Daher erscheint es weniger zweckmäßig, das Konsistenzkriterium auf die Intensitätsverhältnisse anzuwenden.

### Qualitative Evaluation

Es stehen zwar keine Kalibrierdaten für die signalnahe Mikroarrybildauswertung zur Verfügung, aber man kann abstrakteres Wissen über die untersuchten biologischen (Modell-)Systeme zur qualitativen Bewertung der Plausibilität von Auswertungsergebnissen nutzen. Man misst also die Bildauswertung daran, wie gut die Messwerte durch Modelle der Biologie erklärbar sind.

Man braucht zur Durchführung einer solchen Evaluation erstens ein Expressionsexperiment an einem geeigneten biologischen Modellsystem, über das ausreichend viele und mit Mikroarrayexperimenten nachvollziehbare Erkenntnisse vorliegen, und zweitens eine adäquate Prozedur zur Normalisierung und statistischen Auswertung der Messdaten, die die Grundlage der biologischen Interpretation des Experimentes liefert.

Die Evaluation wird mit den Daten aus Mikroarrayexperimenten zur Wurzelknötchenbildung (Nodulation) bei der Pflanze *Medicago truncatula* von H. Küster und anderen ( Universität Bielefeld, Fak. f. Biologie) [69] durchgeführt.

Wurzelknötchen (Noduln) werden in einer Symbiose der Pflanze mit Bodenbakterien (in diesem Experiment *Sinorhizobium meliloti*) gebildet. Die Noduln sind eigens ausgebildete Organe der Pflanze, an deren Bildung über 20 bekannte Gene beteiligt sind. Die Symbiose erlaubt der Pflanze die Aufnahme von Luftsauerstoff und verschafft ihr daher einen Standortvorteil auf nährstoffarmem Boden. Die Fähigkeit zur Bindung von Luftstickstoff wird landwirtschaftlich genutzt (Prinzip der Gründüngung) und ist nicht zuletzt deshalb ein interessanter Forschungsgegenstand.

Küster und andere beschreiben das Mt6kRIT-Mikroarray (*Medicago truncatula* 6k root interaction transcriptome) und damit durchgeführte Pilotexperimente zur *Sinorhizobium meliloti*-induzierten Nodulation, in denen die Genexpression in nicht-nodulierten *M. truncatula*-Wurzeln mit der Genexpression in Wurzelknötchen vier und zehn Tage nach dem ersten Kontakt mit den Symbionten verglichen wurde. Die beiden Experimente werden im Folgenden mit Nod4 und Nod10 bezeichnet.

Das Ziel beider Experimente ist die Identifikation der differentiell exprimierten Gene. Darunter sollten sich auf jeden Fall Nodulationsgene befinden, aber vor allem bei dem Nod10-Experiment ist wegen der längeren Dauer zwischen den Beobachtungszeitpunkten zu erwarten, dass auch andere Gene z. B. durch Entwicklungsprozesse ihre Expression verändern.

Die Mt6kRIT-Mikroarrays besitzen drei Replikate von jeder Sondensequenz und es gibt in den Nodulationsexperimenten sechs technische Replikate jeder Hybridisierung. Es gibt also die vergleichsweise große Anzahl von insgesamt 18 Replikaten für jeden Messpunkt. Die statistische Auswertung der Gesamtexperimente ist zuverlässiger als bei anderen Versuchen mit weniger Replikaten und macht so die Nod4- und Nod10-Experimente besonders für Evaluationszwecke geeignet.

Ursprünglich wurde das Experiment wie folgt ausgewertet [69]:

- Zur Bildauswertung diente die kommerzielle Software Imagene (interaktive Gittersegmentierung, Signalsegmentierung mit z. T. manuell optimierten Kreisen um die Messpunkte).

- Die Messwerte wurden mit dem LOWESS-Verfahren normalisiert, das lineare und nichtlineare Verzerrungen der Intensitätswerte korrigiert [36, 108]. Dem

Verfahren liegt die Annahme zugrunde, dass das mittlere Intensitätsverhältnis in allen Intensitätsbereichen konstant sei.

- Die Verhältniswerte wurden über alle vorhandenen Replikate gemittelt.

- Das primäre Ergebnis der statistischen Auswertung ist die nach den normalisierten M-Werten[1], also den gemessenen Änderungen der Transkriptmengen sortierte Liste der Sequenzen auf dem Array. Außerdem wurde der t-Test für die M-Werte jeder Replikatgruppe gegen (fast) die gesamten Daten durchgeführt. Die t-Werte liefern p-Werte (Wahrscheinlichkeiten für das nicht-vorliegen differentieller Expression) zu den mittleren M-Werten der Replikatgruppen.

Zur praktischen Durchführung der statistischen Auswertung ab der Normalisierung diente das EMMA-System (Uni Bielefeld, ZfG)[8].

Die Bilder aus diesem Experiment wurden mit den Methoden dieser Arbeit neu segmentiert und quantifiziert und der gleichen statistischen Auswertung unterzogen wie oben beschrieben.

Damit sind wiederum Vergleiche mit der etablierten Methode (der kommerziellen Bildauswertung) auf dem abstrakten Niveau der M-sortierten Sequenzliste möglich. Vergleiche der Rangplätze der einzelnen Sondensequenzen geben hierzu eine Übersicht.

Für die Segmentierung der Signalregionen wurde in diesem Teil der Evaluation die aktive Konturesegmentierung benutzt. Auf jeder Signalregion wurde der Mittelwert der Intensität bestimmt und die Hintergrundintensität (wie in Abschnitt 7.3 beschrieben) durch den Modus der Intensitätsverteilung geschätzt.

Bei der Bildauswertung mit Imagene benutzt das EMMA-System dagegen den Median der Intensität auf der Signalregion als Intensitätsmesswert und den Median der Intensität in der Umgebung des jeweiligen Messpunktes als Hintergrundschätzwert .

## 8.2 Stichproben

Zur Systemevaluation wird eine Stichprobe von insgesamt 387 Arraybildern verwendet. In der Tabelle 8.1 ist aufgelistet, wie sich die Gesamtstichprobe aus verschiedenen Typen von Arrays zusammensetzt. Die Teilstichproben unterscheiden sich primär in Größe, Anzahl und Anordnung der Gitter. Die Bezeichnungen der Arrayserien, die aus der gleichen Qelle stammen, sind in der ersten Spalte der Tabelle mit der gleichen Farbe unterlegt.

Die zwölf Arrays aus den *Medicago*-Nodulationsexperimenten, die zur Evaluation der quantitativen Bildauswertung dienen, sind in der Mt6kRIT-Serie enthalten. Für die direkten Vergleiche mit etablierten Auswertungswerkzeugen und die Konsistenzuntersuchung von Replikatgruppen werden ein Mt6kRIT-Array aus dem Nod4-Experiment und ein Array aus der ZmDB-606-Serie benutzt. Die ZmDB-606-Arrays besitzen wie die Mt6kRIT-Arrays ebenfalls dreifach replizierte Messpunkte und sind damit für die

[1] Im Zusammenhang mit der statistischen Auswertung werden häufig die Bezeichnungen „A-Wert“ für den Logarithmus zur Basis 2 des arithmetischen Mittelwertes der Intensitäten beider Kanäle eines Messpunktes und „M-Wert“ für den Logarithmus zur Basis 2 des Intensitätsverhältnisses benutzt.

| Name | Herkunft | Stichprobengröße | Gitteranordnung | Gittergröße | Dicht gepackte Gitter | Hohe Signaldynamik | Ungünstiges SNR | Dunkle Gitterränder | Verlaufene Messpunkte |
|---|---|---|---|---|---|---|---|---|---|
| Halle | A | 66 | 2×6 | 6×4 | | | | ● | |
| S.Meliloti Oligo | A | 60 | 4×12 | 21×20 | ● | | | ● | |
| S.Meliloti PCR | A | 23 | 12×12 | 6×24 | ● | | ● | | |
| Mt6kRIT | A | 35 | 4×12 | 18×24 | | ● | ● | ● | |
| Mt8kRIT | A | 24 | 4×12 | 21×24 | | ● | ● | ● | |
| Chugai | B | 40 | 4×4 | 22×22 | ● | | | | |
| CAMDA Contest | C | 29 | 2×2 | 44×44 | | | | | ● |
| SMD | C | 18 | 4×8 | 20×19 | | ● | ○ | ● | |
| TLG Mensch | D | 18 | 4×12 | 28×24 | ● | | | | |
| ApoAI | E | 16 | 4×4 | 21×19 | | ● | ● | ● | ● |
| Sporman | F | 11 | 4×4 | 19×19 | | | | | |
| WNT | G | 10 | 4×8 | 28×27 | | | | | |
| NMHy | H | 9 | 2×2 | 32×21 | | | | | |
| Swirl | I | 4 | 4×4 | 24×22 | | | | | |
| FlyChip 002 | J | 4 | 4×12 | 12×13 | | | | | |
| FlyChip 002 G | J | 1 | 4×12 | 12×13 | | | | | |
| Spot | K | 3 | 4×4 | 21×21 | | | | | |
| Maroun C | L | 4 | 4×8 | 30×30 | ● | | | ● | |
| Maroun B | L | 3 | 4×8 | 32×31 | ● | | | ● | ● |
| ZmDB Typ 606 | M | 3 | 2×4 | 45×45 | | | | ● | |
| ZmDB Typ 605 | M | 2 | 2×4 | 46×46 | | | | ● | |
| Pine | N | 2 | 2×2 | 24×16 | | | | | |
| MicroZip | O | 1 | 4×12 | 32×29 | ● | ● | | | ● |

Tabelle 8.1: Die Liste der Arrayserien, ihre Gitteranordnung und eine grobe Typisierung

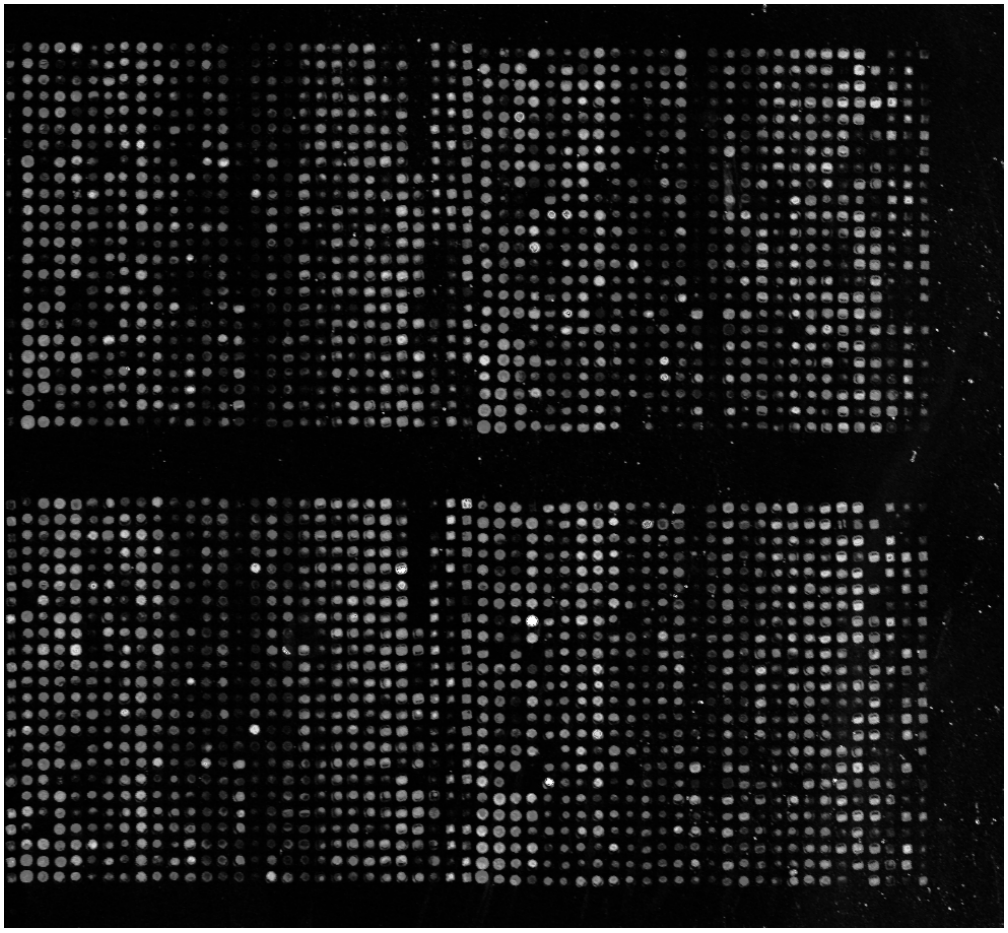


Abbildung 8.1: Ein Ausschnitt aus einem Mikroarraybild aus der Serie TLG Mensch. Es sind vier Gitter zu sehen, deren vertikale Ränder am Zeilenversatz etwa in der Bildmitte zu erkennen sind.

Konsistenzuntersuchung geeignet. Die Vergleichsauswertung für das ZmDB-Array erfolgte mit der frei erhältlichen, interaktiv zu benutzenden Software „Scanalyze“von M. Eisen [38].

Die Teilstichproben besitzen einige Eigenschaften, die unterschiedliche Probleme bei der Gittersegmentierung verursachen. Folgende Kategorien werden zur Beschreibung der Eigenschaften benutzt:

- Dicht gepackte Gitter

  Bilder mit dicht gepackten Gittern besitzen keine durchgehenden Lücken zwischen den einzelnen Messpunktgittern, so dass sie nicht ohne weiteres mit reinen Achsenprojektionsverfahren segmentierbar sind. Die Abbildung 8.1 zeigt als Beispiel einen Ausschnitt aus einem Bild der TLG Mensch-Arrayserie.

- Ungünstiges Signal-Rauschverhältnis

  Bei einigen Bildserien erreicht das Signal der meisten Messpunkte kaum größere Werte als das Hintergrundrauschen. Ursachen dafür können u. a. ungünstige

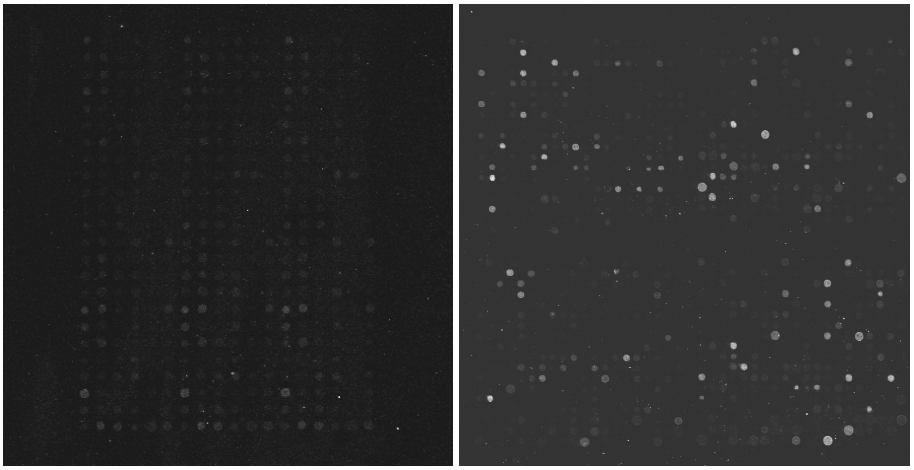


Abbildung 8.2: Zwei Ausschnitte von Arraybildern aus der S. Meliloti PCR- bzw. SMD-Stichprobe mit ungünstigen Signal-Rauschverhältnissen. Visuell wirkt sich der Effekt durch Kontrastarmut der Bilder aus.

Hybridisierungsbedingungen, vorzeitiger Zerfall des Flureszenzfarbstoffs (z. B. durch Ozon) oder zu vorsichtiges Waschen der Arrays nach der Hybridisierung sein. Da diese Einflüsse nicht direkt mit dem Drucken der Arrays zusammenhängen, sind meist nur einige Arrays einer Serie betroffen.

- Hohe Signaldynamik

  Der Farbraum der Mikroarraybilder ist mit 65535 Graustufen in jedem Kanal so fein diskretisiert, dass nicht mehr alle Helligkeitsabstufungen mit dem Auge wahrnehmbar sind. Bei einigen Arrayserien gibt es neben einigen sehr hellen Messpunkten zahlreiche solche, die sich nur minimal vom Hintergrund abheben und auch für menschliche Betrachter künstlich sichtbar gemacht werden müssen. Die ApoA1-Teilstichprobe zeigt diese Eigenschaft besonders deutlich (siehe Abb. 8.3). Der Unterschied zu den Bildern mit dem Merkmal „Ungünstiges Signal-Rauschverhältnis“ besteht darin, dass dort zusätzlich die Varianz des Hintergrundsignals so groß ist, dass sich die Intensitätsbereiche von Hintergrund und dunklen Messpunkten stark überlappen. Das Merkmal „Hohe Signaldynamik“ bezieht sich auf die Verteilung der Intensität der Messpunkte selbst, während das andere Merkmal Hintergrund und Messpunkte vergleicht.

- Dunkle Gitterränder

  Bei fast allen Bildern der Stichprobe stellt man fest, dass die Signalintensität an den oberen Gitterrändern höher als an den unteren Rändern ist. Das deutet auf einen Auswahleffekt bei der Zusammenstellung der Sondensequenzen hin, da die unteren Zeilen der Messpunktgitter zuletzt gedruckt werden. Eine weitere, triviale Ursache für dunkle Gitterzeilen ist die Beendigung des Druckvorgangs vor Erreichen des Zeilenendes oder allgemein unbenutzte Fächer in den Behälterplatten mit dem Sondenmaterial. Bei dem linken Arrayausschnitt

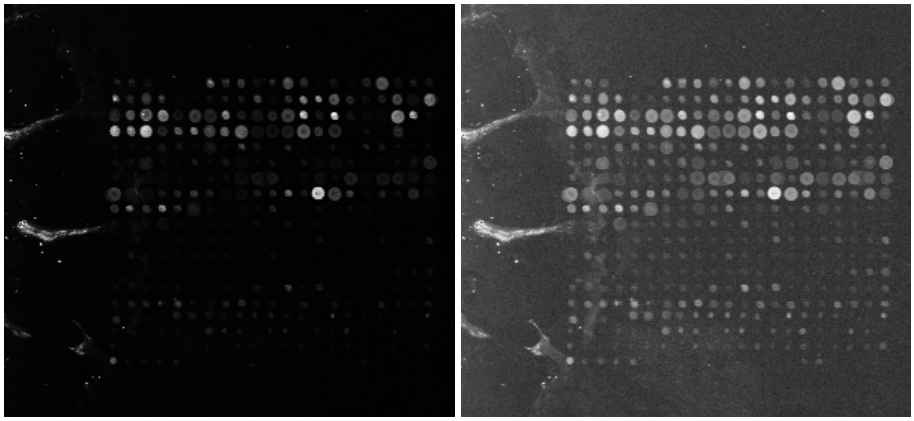


Abbildung 8.3: Ein Ausschnitt aus einem Bild der ApoA1-Serie. Links ist das Bild im Originalzustand zu sehen, rechts ist der gleiche Ausschnitt mit nichtlinear verstärkter Intensität gezeigt (Gammakorrektur mit $\gamma$=3). Es gibt Gruppen besonders heller und sehr schwach sichtbarer Messpunkte.

in Abb. 8.2 aus der S. meliloti-Serie ist dieser Effekt zu sehen, allerdings wurde spaltenweise gedruckt (?).

- Verlaufene Messpunkte

  Wenn die Messpunkte zu dicht oder unter zu hoher Luftfeuchtigkeit gedruckt werden, können sie ineinanderlaufen und sind nicht mehr leicht getrennt segmentierbar. Technische Schwierigkeiten mit dem Arrayer führen manchmal zu Unterbrechungen des Druckvorgangs und verursachen dabei verschmierte Messpunkte. In den Abbildungen 8.3 und 8.4 sind solche Arrayfehler zu sehen.

Bilder, die auch ein Mensch nur sehr mühsam oder gar nicht eindeutig segmentieren kann, sind in der Evaluationsstichprobe nicht enthalten. Darunter fallen z. B. Bilder von sehr stark verunreinigten oder fehlgeschlagenen Hybridisierungen, in denen fast keine Messpunkte zu sehen sind und Bilder in denen wegen Störungen des Arrayers die Gitterachsenrichtung nicht konstant ist.

In den folgenden Abschnitten werden nun die Ergebnisse selbst dargestellt.

## 8.3 Korrektheit der Gittersegmentierung

Die Tabelle 8.2 zeigt die durchschnittlichen Anteile korrekt segmentierter Gitter je Bild bei verschiedenen Kombinationen der MZF-Energieminimierungsverfahren HCF und Local HCF mit den Messpunktdetektionsverfahren (einfache Schwellwertregionen, Eigenspot-Ansatz und aktives Konturmodell mit dem trainierten Polynomklassifikator und mit dem am Hintergrundrauschen kalibrierten $E_c$-Schwellwert). Dazu sind jeweils die Konfidenzintervalle zum 5%-Niveau angegeben, die sich aus Stichprobengröße und den Anteilwerten selbst ergeben (siehe Krengel, Kap. 5 [68]).

Bis auf einen Fall liefert die MZF-Energieminimierung mit dem Local HCF-Algorithmus etwas bessere Ergebnisse. Die Unterschiede sind aber nicht sehr groß und nicht in jedem Fall signifikant. Dagegen sind erwartungsgemäß die Ergebnisse bei

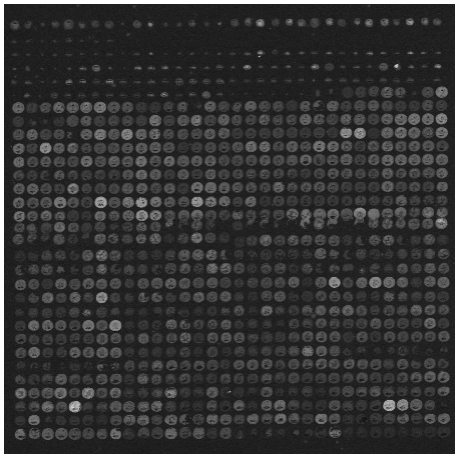


Abbildung 8.4: Ein Ausschnitt aus einem Bild der MarounB-Serie. Beim Drucken dieses Gitters hat sich vermutlich die Nadel im Druckkopf verschoben.

| | HCF | | Local HCF | |
|---|---|---|---|---|
| Schwellwert | 91.50 | [91.0,92.0] | 92.25 | [91.8,92.7] |
| Eigenspots | 94.91 | [94.5,95.3] | 95.29 | [94.9,95.6] |
| Aktive Konturen / Polynomklass. | 96.28 | [95.9,96.6] | 95.71 | [95.3,96.0] |
| Aktive Konturen / $E_c$-Schwellwert | 95.65 | [95.3,96.0] | 95.80 | [95.4,96.1] |

Tabelle 8.2: Die mittleren Prozentsätze korrekt segmentierter Gitter je Bild unter Verwendung verschiedener Energieminimierungsalgorithmen und Messpunktdetektionsmethoden. Dazu sind jeweils Konfidenzintervalle angegeben.

den drei aufwändigeren Messpunktdetektionsverfahren deutlich besser als mit dem Schwellwertverfahren der Regionensegmentierung. Der (signifikante) Unterschied beträgt etwa drei bis dreieinhalb Prozent, und die verbesserten Verfahren unterscheiden sich untereinander nur wenig. Mit dem kalibrierten $E_c$-Schwellwert zur Klassifikation der aktiven Konturen werden ebenso gute Ergebnisse erreicht wie mit den Verfahren, die klassifizierte Stichproben verwenden.

Mit der einfachen Messpunktdetektion werden 65% und mit den leistungsfähigeren Verfahren 75-77% der Bilder vollständig richtig segmentiert. Bei Verwendung der aktiven Kontursegmentierung werden 12 der 23 Bildserien vollständig richtig segmentiert.

Die Grafik in Abbildung 8.5 zeigt, wie die Ergebnisse innerhalb der Gesamt- und Teilstichproben variieren. Die Punkte zeigen die Mittelwerte der Anteile korrekt segmentierter Gitter je Bild für jede Teilstichprobe an, wobei die horizontalen Linien den Bereich der aufgetretenen Einzelwerte darstellen.

Man erkennt, dass die Bildserien mit guter Bildqualität, die sich überwiegend im unteren Teil der Liste befinden (siehe Tabelle 8.1), weitgehend fehlerfrei segmentiert werden. Einzelne Fehler in diesen Teilstichproben, wie z. B. bei den Fly-Chip-Arrays, werden durch lokale Kontaminationen der Arrayoberflächen verursacht. Die wichtigste Fehlerursache bei den übrigen Arrayserien sind dunkle Randzeilen oder -spalten, die keine eindeutige Gittersegmentierung zulassen. Die Komponente des MZF-Potentials,

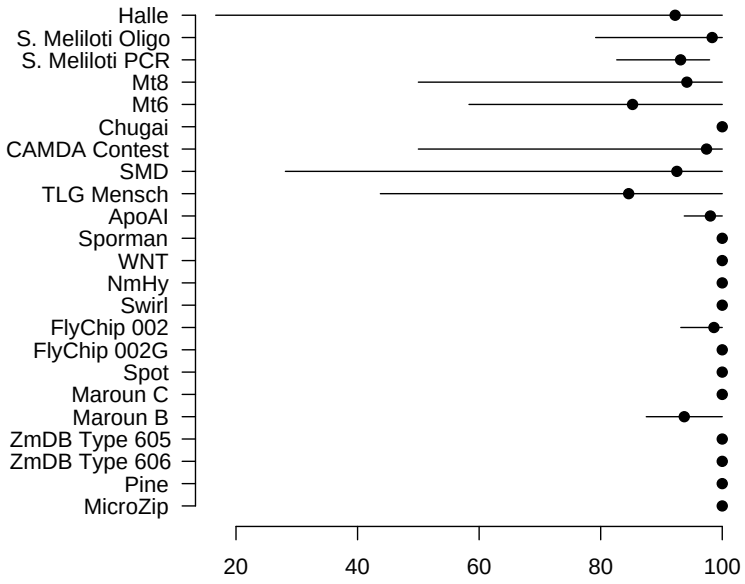



Abbildung 8.5: Die Punkte markieren die mittleren Prozentsätze korrekt segmentierter Gitter je Bild für alle Teilstichproben mit der aktiven Konturseg-mentierung und $E_c$-Schwellwertklassifikation. Die Balken zeigen den Bereich der aufgetretenen Werte.

die regelmäßige Gitteranordnungen bevorzugt, kann diese Probleme nicht immer beseitigen, aber die Ergebnisse werden erheblich schlechter, wenn man sie abschaltet. Die Segmentierfehler bestehen meist in Verschiebungen der Gitter um eine Zeile oder Spalte.

Die Arrays der „Halle“-Serie besitzen sehr kleine Gitter, deren Messpunkte z. T. überwiegend dunkel sind. Bei einigen dieser Bilder versagt die Gitterkonstantenschätzung, weil zu wenige Regionen an benachbarten Gitterknoten gefunden werden. Ein ähnliches Problem tritt bei einem Array in der Mt6kRIT-Serie auf.

Die Fehler bei der Gittersegmentierung der TLG Human-Serie sind zum Teil dadurch verursacht, dass hier wegen technischer Probleme beim Drucken einige Gitter ineinandergeschoben sind, wodurch keine sinnvolle Segmentierung mit dem MZF-Modell möglich ist, das überlappende Gitter verbietet. Dazu kommen Fehlleistungen der Messpunktdetektion.

Die Abbildung 8.6 stellt die Segmentierungsleistung mit den verschiedenen Messpunktdetektionsverfahren nach den Teilstichproben aufgeschlüsselt gegenüber. Die roten Balken gehören zum einfachen Schwellwertregionenverfahren, die orangefarbenen zur Eigenspot-Methode und die gelben zur aktiven Kontursegmentierung mit Polynomklassifikator. Die größten Einzelverbesserungen durch die aufwändigeren Verfahren sind bei Bildern zu finden, die verlaufene Messpunkte enthalten (CAMDA Contest, Apo A1, MicroZip).

Bei den Segmentierungsergebnissen der S. meliloti-Teilstichprobe stellt man häufig fest, dass die Energieminimierung mit Local HCF oder HCF das globale Optimum nicht erreicht, weil es Kopplungen über mehrere Knoten hinweg gibt, wie schon in Abschnitt 6.4.4 bzw. Abb 6.14 gezeigt. Für zwei dieser Bilder wurde die Energieminimierung mit dem Simulated-Annealing-Verfahren durchgeführt, wodurch sich die Energie noch etwas senken lies, die Zahl der Fehler (6 der 48 Gitter falsch positioniert) aber gleich blieb. Die Abbildung 8.7 zeigt die Energie im MZF während des Annealings. Die Tatsache dass die Fehlerzahl nicht sinkt lässt vermuten, dass in diesem Fall die Grenzen des heuristischen MZF-Modells erreicht sind.

### 8.3.1 Stabilität der Parameter des MZF-Modells

Die Abbildung 8.8 zeigt repräsentativ an dem Beispiel der Sporman-Teilstichprobe die Abhängigkeit der Segmentierungsleistung von den freien Parametern des MZF-Modells. Bei den meisten leicht zu segmentierenden Bildserien sind die Ergebnisse fast ganz unabhängig von der Parameterwahl, weil bei ausreichend großen Lücken zwischen den Gittern fast keine Interaktion zwischen den MZF-Knoten stattfindet. Bei einigen Bildserien, bei denen die Energieminimierung das globale Minimum nicht erreicht (S. meliloti PCR, TLG Human) ist die Abhängigkeit der Segmentierungsergebnisse von den Parametern sehr kompliziert (mehrere Maxima). Bei Bildserien, bei denen die Messpunktdetektion Probleme bereitet (z.B. Halle und Apo A1), sind die Maxima der Segmentierungsleistung bei höheren Werten des Parameters $\beta_3$ zu finden, der die regelmäßige Gitteranordnung gewichtet. Offensichtlich ist die a-priori-Information nützlicher, wenn die Beobachtungen unsicher sind. Weitere Beispiele hierzu sind in dem Artikel zur MZF-Gittersegmentierung diskutiert [60].

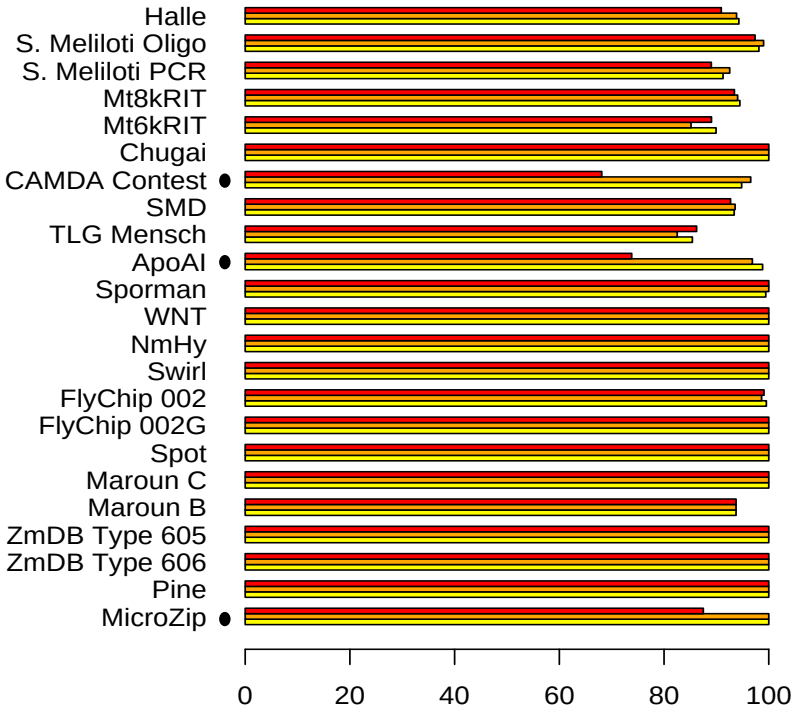



Abbildung 8.6: Vergleich der Prozentanteile korrekt segmentierter Gitter in jeder Teilstichprobe bei Messpunktdetektion durch Schwellwertsegmentierung (oberer Balken der Dreiergruppen), mit aktiven Konturen (mittlerer Balken) und mit dem Eigenspot-Ansatz (unterer Balken). Die Teilstichproben, die verlaufene Messpunkte enthalten, sind mit Punkten markiert. Bei diesen Bildserien bringt die verbesserte Messpunktdetektion deutlichen Gewinn.

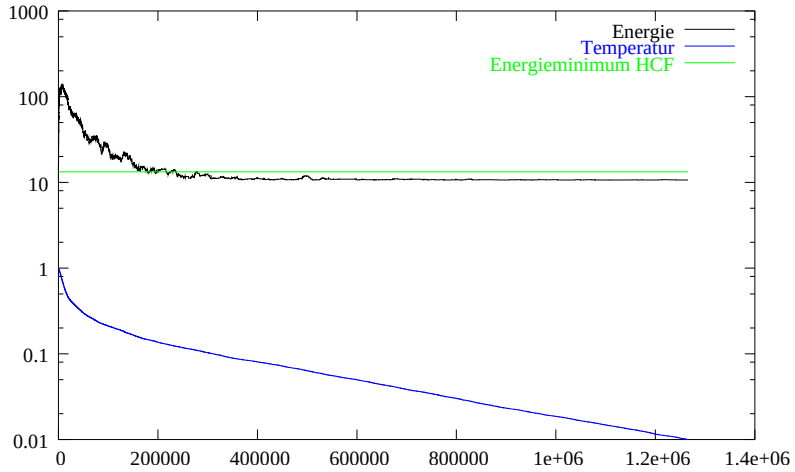



Abbildung 8.7: Die Energie der MZF-Konfiguration und der Temperaturparameter des Metropolis-Algorithmus bei der Gittersegmentierung mit dem Simulated-Annealing-Verfahren sowie die mit dem Local HCF-Verfahren erreichte Restenergie, aufgetragen über der Zahl der Iterationen (Konfigurationsänderungen)

### 8.3.2 Restenergieen des MZF-Modells

Die Abbildung 8.9 zeigt Histogramme der Restenergieen einzelner MZF-Knoten, die zu korrekten Segmentierungsergebnissen (grün) und falsch segmentierten Gittern gehören (rot). Die Energieverteilungen korrekt und fehlerhaft segmentierter Gitter überlappen sich stark, weshalb die Restenergie in dieser Form nicht als Rückweisungskriterium geeignet scheint.

## 8.4 Quantitative Bildauswertung

Die Evaluation der quantitativen Bildauswertung wird mit den Bildern aus dem *Medicago*-Nodulationsexperiment und einem Bild aus der ZmDB-Stichprobe durchgeführt, das ebenfalls viele replizierte Messpunkte enthält. Die Vergleichswerte für die Medicago-Bilder sind mit der kommerziellen Software Imagene bestimmt und für das ZmDB-Array mit dem frei erhältlichen Scanalyze.

Die Abbildung 8.10 zeigt die Regionenbilder, die mit den drei verschiedenen Verfahren zur Signalsegmentierung in einem Ausschnitt von einem der Medicago-Arraybilder berechnet werden. Die Radien der Kreissegmentierung werden aus der Regionensegmentierung der Gittersegmentierung übernommen. Bei dunklen Messpunkten, zu denen dort keine Regionen vorhanden sind, müssen daher (aus den Gitterkonstanten abgeleitete) Standardwerte eingesetzt werden. Am Ergebnis der Mann-Whitney-Segmentierung ist deutlich zu erkennen, dass dies Verfahren bei dunklen Messpunkten keine Regionen bestimmen kann. Bei sehr hellen Messpunkten erscheinen die Regionen dagegen oft zu groß. Zur quantitativen Auswertung dunkler Punkte ohne Mann-Whitney-Region wird die Kreissegmentierung benutzt. Die aktive Konturseg-mentierung erzeugt im Vergleich zur Mann-Whitney-Segmentierung kleinere Regio-

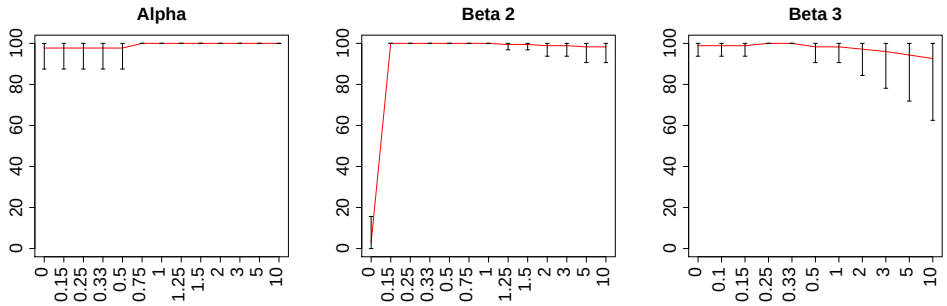



Abbildung 8.8: Untersuchung der Stabilität der MZF-Parameter an der Sporman-Stichprobe. In der Umgebung der gewählten Werte $\alpha = 1$ (Regionen in Gittern), $\beta_2 = 1$ (Regionen zwischen Gittern) und $\beta_3 = 0.33$ (Regelmäßige Gitteranordnung) ändert sich die Korrektheit der Segmentierung kaum. Die Variation des Parameters $\beta 1$, der das Überlappungs-Potential gewichtet, ist ohne Einfluss auf die Korrektheit der Segmentierung.

nen mit glatteren Rändern. Nur bei dunklen Messpunkten entstehen sehr unregelmäßig geformte Regionen, weil bei fehlenden Objektkanten Hintergrundrauschen oder Artefakte segmentiert werden. Verunreinigungen können auch bei diesem Verfahren Fehlsegmentierungen verursachen, aber sie sind wesentlich seltener als bei der Mann-Whitney-Segmentierung. Dazu muss allerdings gesagt werden, dass bei anderen Bildern (z. B. dem NMHy-Beispielbild in Abb. 7.1 auf S. 104) mit geringerem Rauschen (die mangels Replikaten nicht für die quantitative Evaluation geeignet sind) zumindest der visuelle Eindruck der Regionenbilder besser ist.

### 8.4.1 Direkter Vergleich mit etablierten Systemen

Die Abbildung 8.11 zeigt die Intensitätswerte, die man mit den drei verschiedenen Signalsegmentierungsmethoden (Kreise, Mann-Whitney und aktive Konturen) erhält, und die Vergleichswerte der Auswertung mit Imagene bzw. Scanalyze gegeneinander aufgetragen. Man erkennt, dass die aktive Konturseg mentierung zu höheren Intensitätswerten führt als die anderen Methoden (in der Zeile „Aktive Konturen“liegen praktisch alle Datenpunkte über den Diagonalen). An den zugehörigen Regionenbildern in Abbildung 8.10 erkennt man den Grund dafür: Die Signalregionen der aktiven Konturseg mentierung enthalten abgesehen von dunklen Messpunkten den kleinsten Hintergrundanteil, so dass sich die höchsten mittleren Intensitäten ergeben. Die Abweichungen zwischen den unterschiedlichen Methoden sind dennoch im Wesentlichen linear. Kreis- und aktive Konturseg mentierung machen die in den Bilddaten vorhandene Übersteuerung deutlicher sichtbar als die Mann-Whitney-Segmentierung und die Vergleichsmethoden.

In Abbildung 8.12 sieht man Vergleiche der mit verschiedenen Programmen bestimmten Intensitätsverhältnisse. Es ist jeweils die Differenz der M-Werte über dem mittleren A-Wert aufgetragen (also das Verhältnis der Verhältnisse über den gemittelten Intensitäten mit logarithmierten Achsen). Größere Abweichungen zeigen sich nur bei dunkleren Messpunkten. Dies ist besonders bei dem Vergleich mit Scanalyze auf

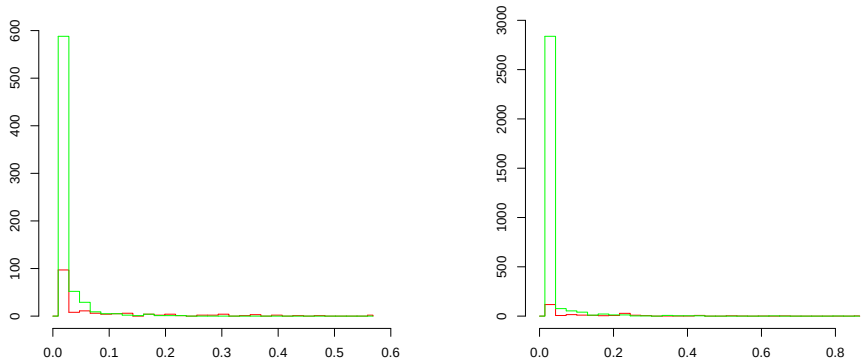



Abbildung 8.9: Histogramme der MZF-Knotenenergien richtig (grün) und falsch (rot) segmentierter Gitter für die Teilstichproben TLG Human (links) und S. meliloti PCR (rechts)

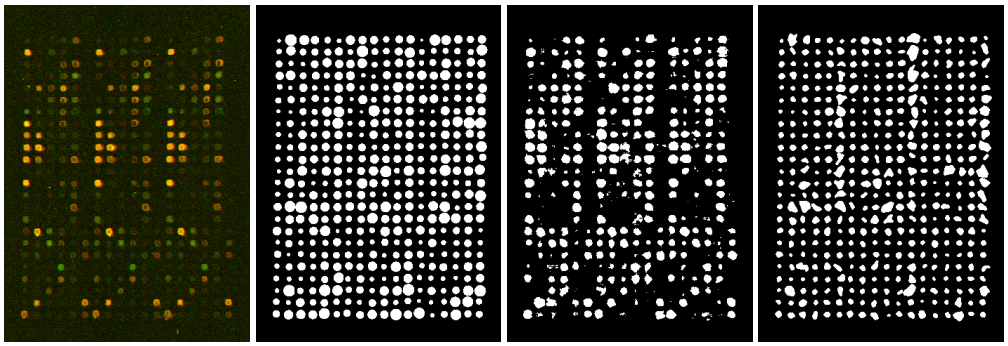


Abbildung 8.10: Ein Ausschnitt aus einem Mt6kRIT-Mikroarraybild und die mit Kreissegmentierung, Mann-Whitney-Segmentierung und aktiven Konturen bestimmten Messpunktregionen.

Unterschiede bei der Hintergrundkorrektur zurückzuführen, da die Abweichung offensichtlich systematisch ist (Bei den eigenen Ergebnissen und bei Imagene wurde der Modus der Hintergrundintensität benutzt, bei Scanalyze der Median der Hintergrundintensität).

### Robustheit bezüglich der Gittersegmentierung

Alle Signalsegmentierungsmethoden erfordern die Vorgabe einer ungefähren Messpunktposition durch die Gittersegmentierung. Die Messpunktsegmentierung sollte möglichst robust gegen kleine Variationen der Gittersegmentierung sein, damit die Reproduzierbarkeit auch bei manuell korrigierter Gittersegmentierung möglichst gut ist. Die Tabelle 8.3 stellt die Korrelationen der Messwerte bei verrauschten Messpunktpositionen (Standardabweichung 1 Pixel) mit den Messwerten bei unverrauschten Messpunktpositionen für die verschiedenen Segmentierungsverfahren gegenüber.

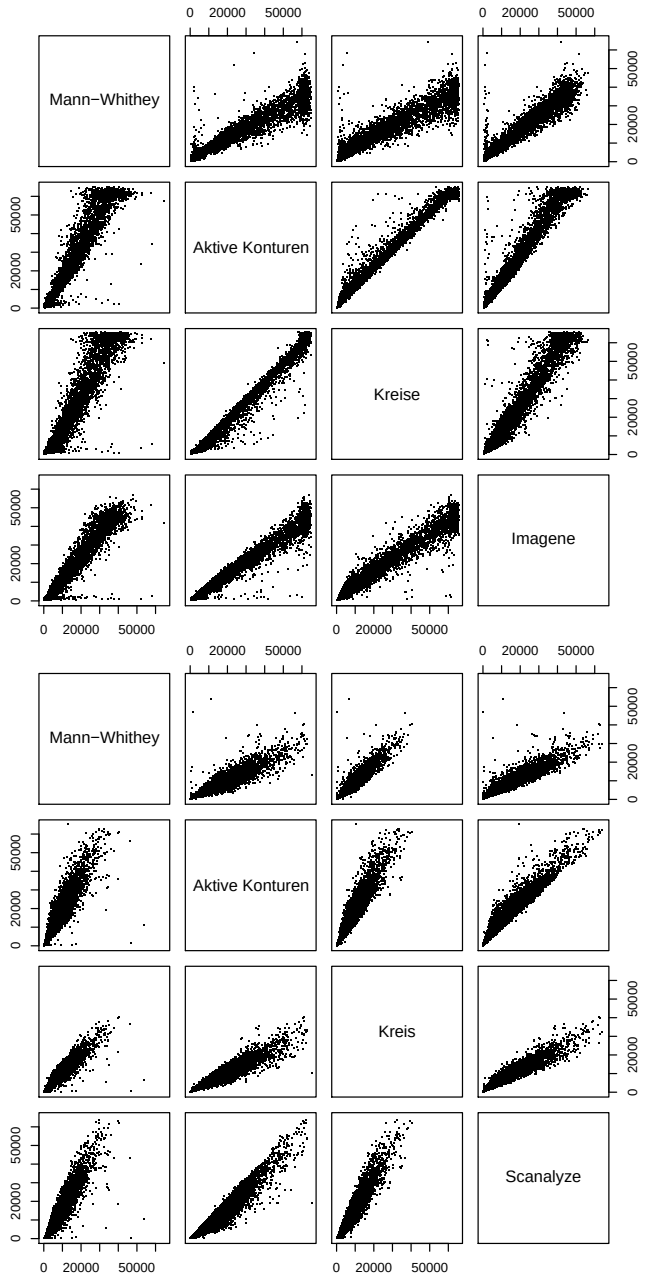



Abbildung 8.11: Oben sind die Intensitäten der Messpunkte im Rotkanal des Arrays Mt6kRIT-121S02 bei verschiedenen Methoden der Messpunktsegmentierung und die Vergleichsdaten der Imagene-Bildauswertung gegeneinander aufgetragen. Unten sind die Daten für das Array ZmDB-606-01-02-53 dargestellt, wobei die Vergleichsdaten mit Scanalyze bestimmt sind. Die Hintergrundanteile sind in beiden Fällen noch nicht korrigiert.

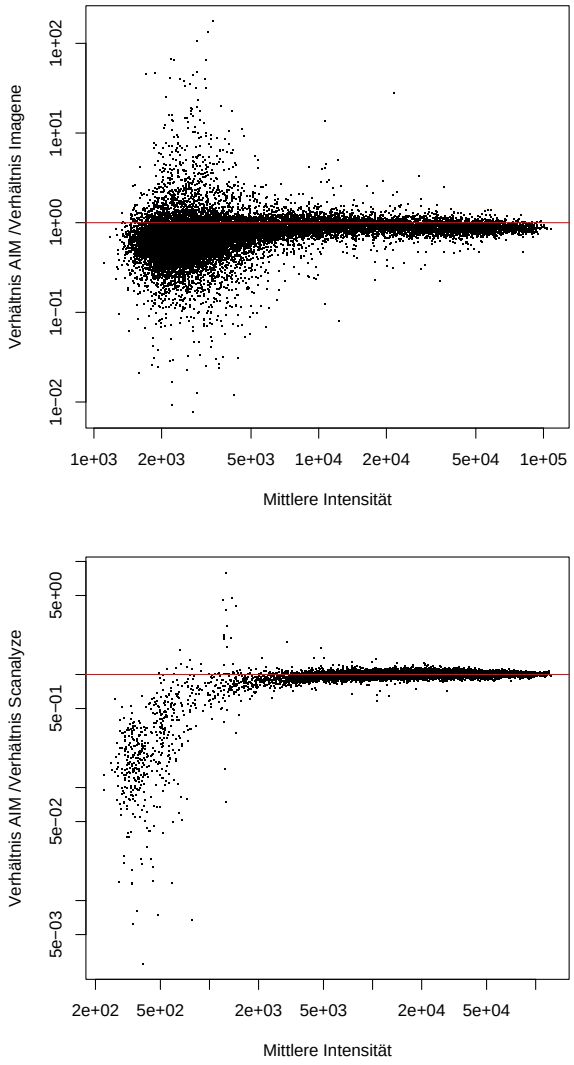



Abbildung 8.12: Oben sind die Differenzen der mit AIM und Imagene berechneten M-Werte der Messpunkte des Arrays Mt6kRIT-121S02 über den A-Werten aufgetragen (Verhältnis der Verhältnisse über der mittleren Intensität, logarithmische Achsen). Unten sind die mit Scanalyze und AIM bestimmten Messwerte des Arrays ZmDB-606-01-02-53 in gleicher Weise dargestellt.

Die Kreissegmentierung und die aktive Kontursegmentierung haben demnach die besten Robustheitseigenschaften. Das relativ schlechte Abschneiden der Mann-Whitney-Segmentierung ist vermutlich durch die empfindliche Abhängigkeit des Verfahrens von der lokalen Hintergrundstichprobe bedingt. Durch die Verrauschung der Gittersegmentierung geraten leicht Signalpixel benachbarter Messpunkte in die Hintergrundstichproben und stören die Segmentierung. Die Kreissegmentierung ist robust, weil deren Regionen meistens etwas Hintergrund mit einschließen, so dass bei kleinen Verschiebungen die tatsächlichen Messpunkte in der Signalregion bleiben.

| Array | Kreissegmentierung | Mann-Whitney | Aktive Konturen |
|---|---|---|---|
| ZmDB 606-01-02-53 | 0.9971 | 0.9463 | 0.9963 |
| Mt6kRIT-121-S02 | 0.9900 | 0.9541 | 0.9960 |

Tabelle 8.3: Die Korrelationen der Intensitätsmesswerte mit verrauschter und mit ungestörter Gittersegmentierung bei Verwendung verschiedener Signalsegmentierungsmethoden. Zum Vergleich dienten 15376 (ZmDB) bzw. 5453 (Mt6kRIT) Messpunkte mit Intensitäten von mindestens zwei Standardabweichungen über dem Hintergrundsignal.

### 8.4.2 Konsistenz replizierter Messdaten

Die Arrays der Zmdb-606- und Mt6kRIT-Teilstichproben besitzen drei (sequenzgleiche) Replikate von jedem Messpunkt und sind daher von den zur Verfügung stehenden Arrays am besten für die Konsistenzprüfung an Replikatgruppen geeignet. Auf den ZmDB-606-Arrays gibt es 5400 und auf den Mt6kRIT-Arrays 6144 Replikatgruppen. Davon wurden 4852 bzw. 2985 Gruppen verwendet, die frei von gesättigten Signalpixeln waren und deren Intensitätsmesswerte mindestens zwei Standardabweichungen über dem Hintergrundrauschen lagen. Mit diesen Anzahlen von Replikatgruppen ist trotz der recht kleinen Anzahl von drei Replikaten pro Gruppe ausreichende statistische Sicherheit gegeben. Die Abbildung 8.13 zeigt geschätzte Verteilungsdichten der Varianzkoeffizienten der Replikatgruppen und der Messwerte selbst. Bei den Daten beider Arraytypen erkennt man, dass Kreis- und Mann-Whitney-Segmentierung die größte (unerwünschte) Varianz in den Replikatgruppen erzeugen. Die aktive Kontursegmentierung führt zu etwa gleich großer Varianz in den Replikatgruppen wie die manuell unterstützte Auswertung mit Imagene und zu kleinerer Varianz als die Auswertung mit dem Programm Scanalyze. Dabei ist die Varianz der Intensitätsmesswerte insgesamt bei der aktiven Kontursegmentierung größer als bei den Vergleichsdaten, was zusammen mit der ersten Beobachtung positiv für die aktive Kontursegmentierung zu werten ist. Der Unterschied ist bei den Scanalyze-Daten deutlicher als bei den Imagene-Daten.

### 8.4.3 Qualitativer Vergleich

Nach der Betrachtung der bildnahen Messdaten werden nun die daraus abstrahierten Auswertungsergebnisse der *Medicago*-Nodulationsexperimente Nod 4 und Nod 10 verglichen, die man bei Verwendung des Bildauswertungssystems Imagene und dem hier beschriebenen Verfahren erhält. Zunächst fällt beim Vergleich der über alle

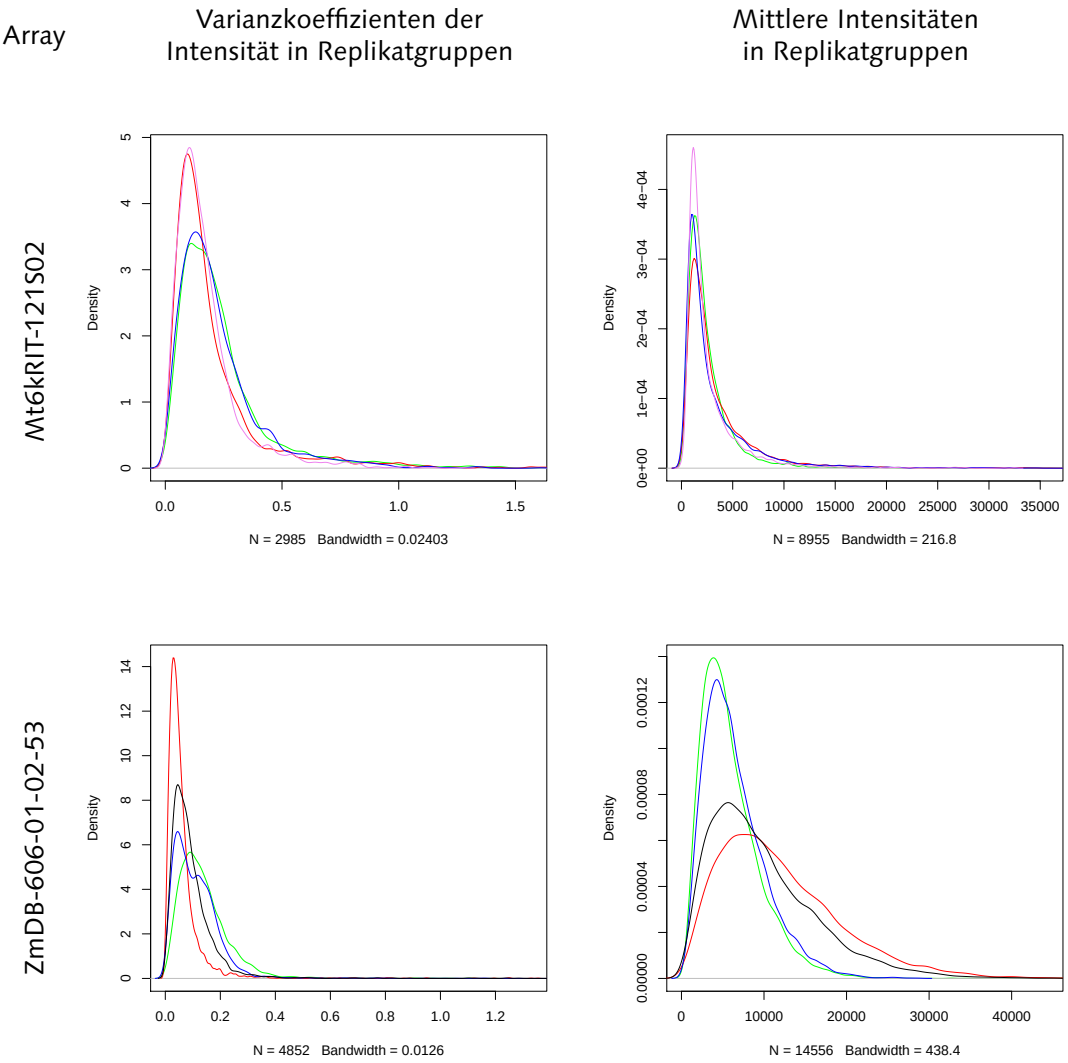



Abbildung 8.13: Verteilung der Varianzkoeffizienten der Intensitäten von Gruppen von replizierten Messpunkten (links) und Verteilung der mittleren Intensität der Gruppen bei Messpunktsegmentierung mit Kreisverfahren (blau), aktiven Konturen (rot), Mann-Whitney-Methode (grün) und mit den Programmen Scanalyze (schwarz) und Imagene (violett)

.

Replikate gemittelten M-Werte (logarithmierte Intensitätsverhältnisse) auf, dass AIM systematisch etwas kleinere Werte berechnet (siehe Abb. 8.14). Die Ursache ist vermutlich, dass das Auswertungssystem EMMA bei den Imagene-Daten den Median-Hintergrundschätzer und bei AIM den Modus-Hintergrundschätzer benutzt. Die Modus-Hintergrundschätzungen sind systematisch kleiner, weshalb der in Abschnitt 8.1.2 diskutierte Dämpfungseffekt eintritt. Die Korrelation der M-Werte beider Programme ist dennoch mit etwa 0.854 bei beiden Experimenten recht gut.

In der Originalauswertung des Experimentes sind die Sondensequenzen nach den M-Werten sortiert [69]. Daher sind in Abbildung 8.15 die Ränge der entsprechend geordneten Sondensequenzen bei den beiden Auswertungen gegeneinander aufgetragen. Man erkennt weitgehende Übereinstimmung, wobei es bei dem Nod 4-Experiment etwas häufigere Umordnungen gibt. Im Anhang C sind detaillierte Listen der Sondensequenzen mit den jeweils 250 größten M-Werten zu finden. Rangstatistiken zeigen, dass die bekannten Nodulin-Gene in der AIM-Auswertung etwas näher an den Enden der M-sortierten Listen der Sondensequenzen erscheinen. Daraus läßt sich aber nicht unbedingt schließen, dass diese Auswertung genauer ist, denn man muss davon ausgehen, dass in dem Experiment neben den Nodulin-Genen viele weitere Sequenzen unterschiedlich exprimiert werden.

Helge Küster, einer der Autoren der *Medicago*-Nodulationsexperimente, kommentiert die Ergebnisse der beiden Auswertungen so:

> Generell lässt sich also sagen, dass AIM zu sinnvollen Ergebnissen kommt und keinesfalls ein grundsätzliches Problem bei der Microarray-Auswertung mit diesem Programm besteht. Was im Vergleich zu kommerziellen Programmen „richtiger“ ist, ist subjektiven Kriterien unterworfen, da man z. B. auch in Imagene durch Änderungen der Größe der Radien der Kreise vermutlich sehr nahe an die AIM-Werte käme.

## 8.5 Laufzeitverhalten und Speicherbedarf

Für den Anwender des Systems ist in vor allem die Laufzeit der Gittersegmentierung interessant, da deren Ergebnisse in der Regel visuell überprüft werden müssen, bevor die automatische Verarbeitung fortgesetzt werden kann. Daher wird im Folgenden dieser Teil des Systems ausführlicher untersucht.

Als Testsystem diente ein gewöhnlicher PC mit Intel Pentium 4 - Prozessor, 2,4 GHz Takt, 256 Kilobyte Cache-Speicher und 512 Megabyte Hauptspeicher. Der Hersteller Intel gibt für diesen Prozessor die Benchmark-Werte SPEC SPECint*_base2000 819 und SPECfp*_base2000 806 an. Als Betriebssystem diente Linux der Kernelversion 2.4.21 und die Programme wurden mit der Gnu Compiler Collection der Version 3.3 bei der höchsten Optimierungsstufe und architekturspezifischer Codeerzeugung übersetzt.

### 8.5.1 Laufzeit

Die Abbildung 8.16 zeigt die Abhängigkeit der Gesamtlaufzeiten der Gittersegmentierung von Bildgröße und Regionenanzahl für alle Bilder der Stichprobe. In beiden

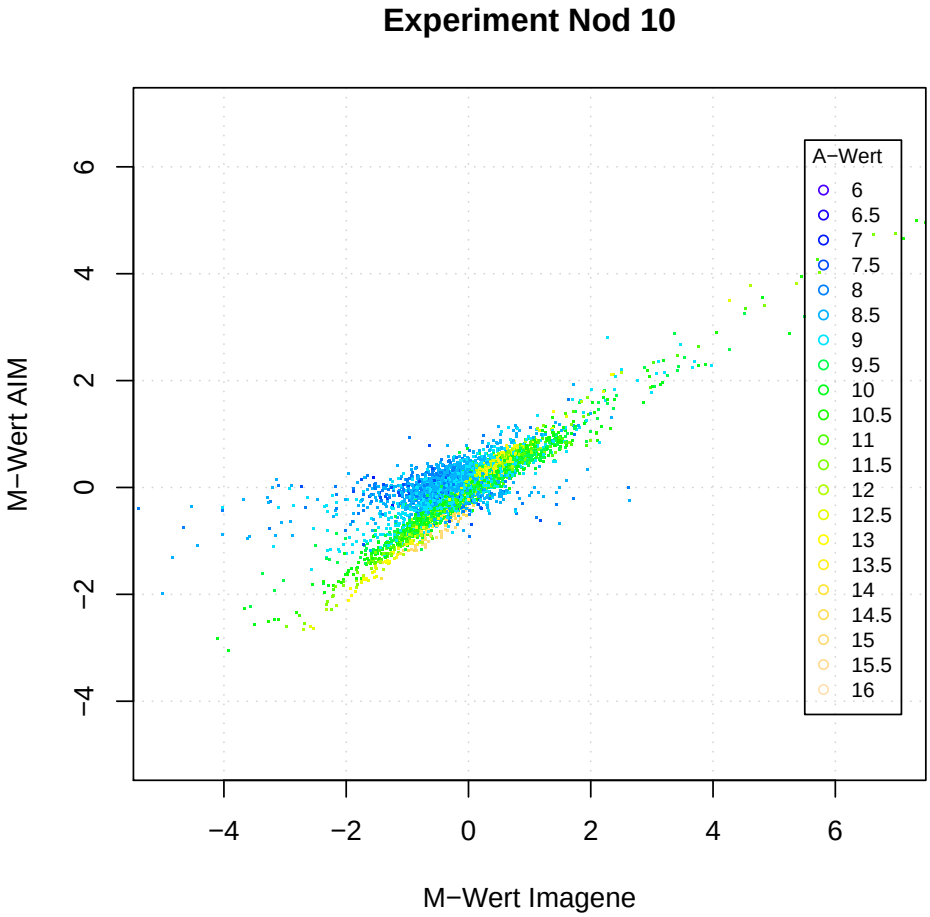



Abbildung 8.14: In dieser Grafik sind die mit Imagene und AIM berechneten M-Werte des Nod-10-Experiments gegeneinander aufgetragen. Die Farbe der Datenpunkte zeigt den A-Wert an (beachte Legende). Die (nicht gezeigten) Daten zum Nod-4-Experiment haben sehr ähnliche Eigenschaften.

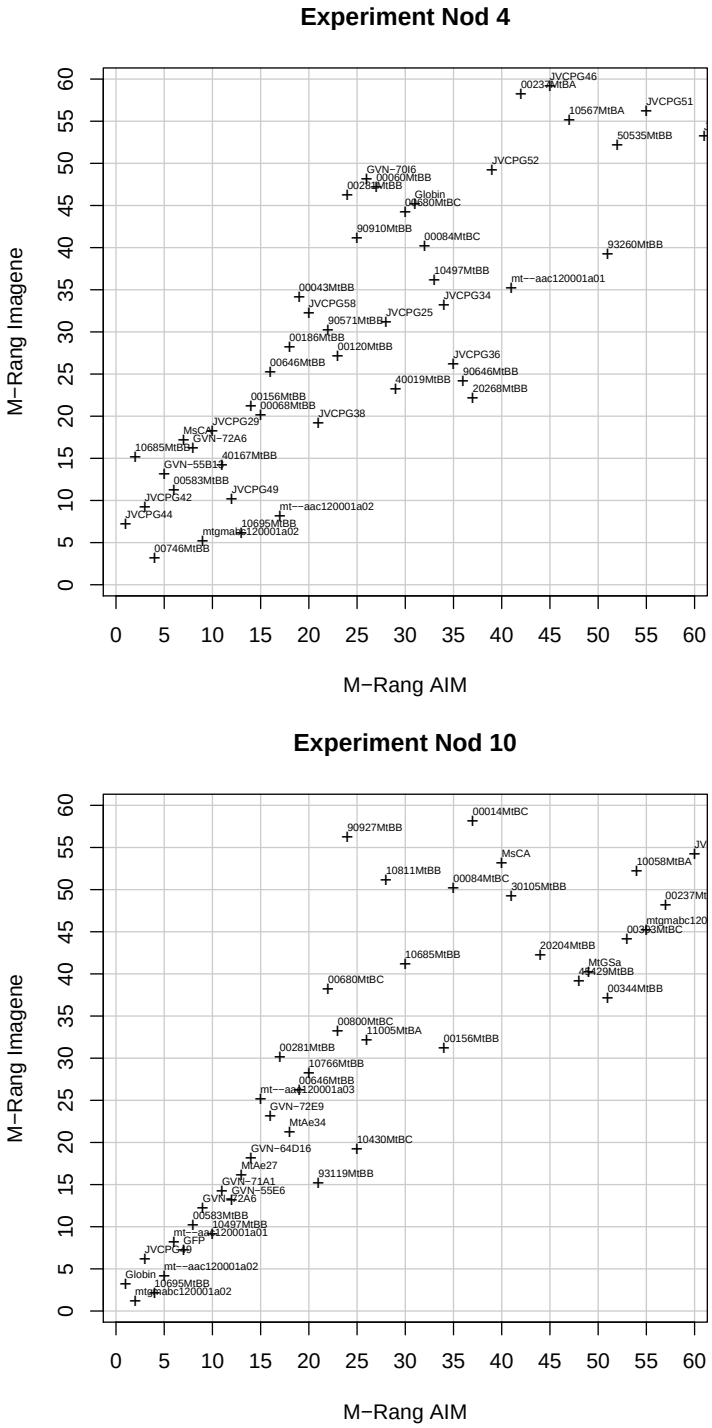



Abbildung 8.15: Vergleich der Rangordnungen der nach M-Werten sortierten Sondensequenzen der *Medicago*-Nodulationsexperimente bei Bildauswertung mit AIM und Imagene. Die Bezeichnungen der Sondensequenzen sind im Anhang C aufgeschlüsselt.

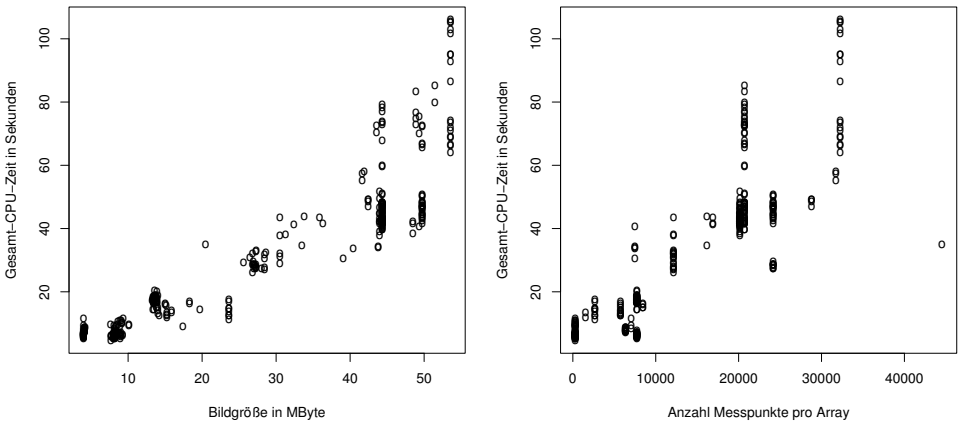



Abbildung 8.16: Die für die Gittersegmentierung benötigte Rechenzeit in Abhängigkeit von der Eingabebildgröße und der Anzahl gedruckter Messpunkte.

Fällen ist der Anstieg nur geringfügig überlinear. Die zu erwartende Wachstumsordnung ist $O(n \log n)$, die z. B. durch die Konstruktion der kd-Bäume zur Regionenspeicherung bedingt ist. Aus der in Abb. 8.17 dargestellten Verteilung der Laufzeit auf die Teile der Verarbeitungskette kann man allerdings ablesen, dass über 60% der Gesamtrechenzeit für die Vorverarbeitung und Regionensegmentierung verbraucht werden, deren Laufzeit linear von der Größe der Eingabebilder bzw. der Regionenanzahl abhängt. Die Laufzeit der kd-Baum-Algorithmen mit $O(n \log n)$-Effizienz ist in den Anteilen Gitterkonstantenschätzung, Verkettung und MZF-Hypothesengenerierung enthalten, die zusammen nur ca. 12% der gesamten Rechenzeit ausmachen. Daher ist der überlineare Anstieg in Abbildung 8.16 nicht allzu deutlich sichtbar.

Wenn die verbesserten Messpunktdetektionsverfahren (Eigenspots und aktive Konturen) benutzt werden, steigt die benötigte Rechenzeit etwa um den Faktor 5,5 an. Damit liegt die maximale Rechenzeit pro Bild noch unter 10 Minuten. Die aktive Kontursegmentierung dauert etwa 7.6 Millisekunden und die Messpunktdetektion mit dem Eigenspot-Ansatz etwa 7.1 Millisekunden pro Messpunkt. Die Energieminimierung der aktiven Kontursegmentierung konvergiert auf der Stichprobe im Mittel nach 24 Iterationen (mit Abbruchschrittweite $\lambda = 0.01$).

In den angegebenen Laufzeiten ist die Zeit für das Laden der Bilddaten und die Erzeugung der XML-formatierten Ausgabe nicht enthalten. Praktisch kann die Ein-Ausgabezeit wegen der relativ großen Datenmengen nicht vernachlässigt werden, sie ist aber nicht von den Bildverarbeitungsalgorithmen abhängig.

Die quantitative Bildauswertung erfordert wie die Gittersegmentierung den Medianfilter zur Vorverarbeitung. Von den drei betrachteten Signalsegmentierungsverfahren (Kreis, Mann-Whitney und aktive Konturen) braucht die aktive Kontursegmentierung die längste Rechenzeit, die einige Minuten pro Bild beträgt (wie oben angegeben). Dazu kommt die Schätzung des Maximums der Hintergrundverteilung, die z. Zt. noch nicht effizient implementiert ist und daher 15-20 Minuten je Bild dauert. Die Gesamt-

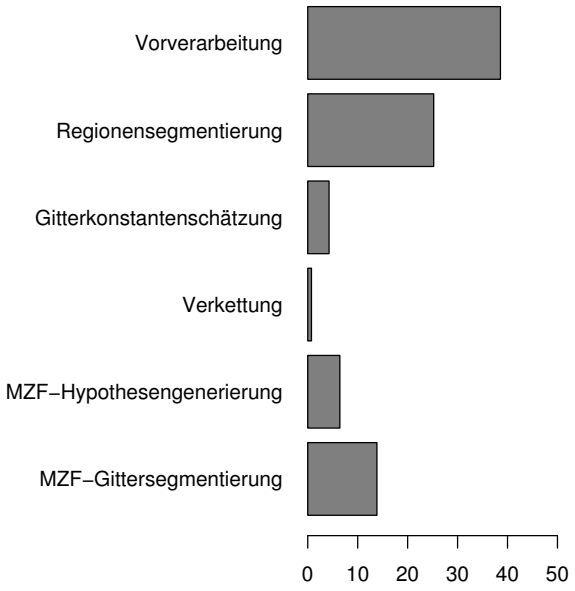



Abbildung 8.17: Die Anteile einzelner Teile der Bildverarbeitungskette an der Gesamtlaufzeit der Gittersegmentierung in Prozent

rechenzeit der quantitativen Auswertung je Bild beträgt daher bis zu ca. 25 Minuten.

### 8.5.2 Speicherbedarf

Der Speicherbedarf ist in der Abbildung 8.18 dargestellt. Es sind die mit Hilfe des Betriebssystems ermittelten, maximal aufgetretenen Prozessgrößen (tatsächlich genutzter Hauptspeicher) während der gesamten Gittersegmentierung gegen Eingabebildgröße und Regionenanzahl aufgetragen. Man erkennt darin wiederum im Wesentlichen lineare Abhängigkeiten. Bei den Ausreißern oben rechts in der Auftragung der Prozessgröße gegen die Bildgröße handelt es sich um Bilder mit vielen Regionen. Der wesentliche Teil des benutzten Speichers (zweifache Größe der Eingabebilder) wird für die Bilddaten selbst und das Medianbild benötigt. Dazu kommt noch der Speicherbedarf für Regionen, Objekte und MZF-Zustände, der nach Abbildung 8.18 mit etwa der ein- bis eineinhalbfachen Größe der Eingabebilder abzuschätzen ist. Bei Benutzung der aktiven Kontursegmentierung wird zusätzlich noch der vierfache Speicherplatz der Eingabebilder für die Gradientenbilder benötigt, so dass dann insgesamt Speicherplatz von der sechs- bis siebeneinhalbfachen Größe der Eingabedaten benutzt wird (auf der Stichprobe maximal etwa 440 Megabyte).

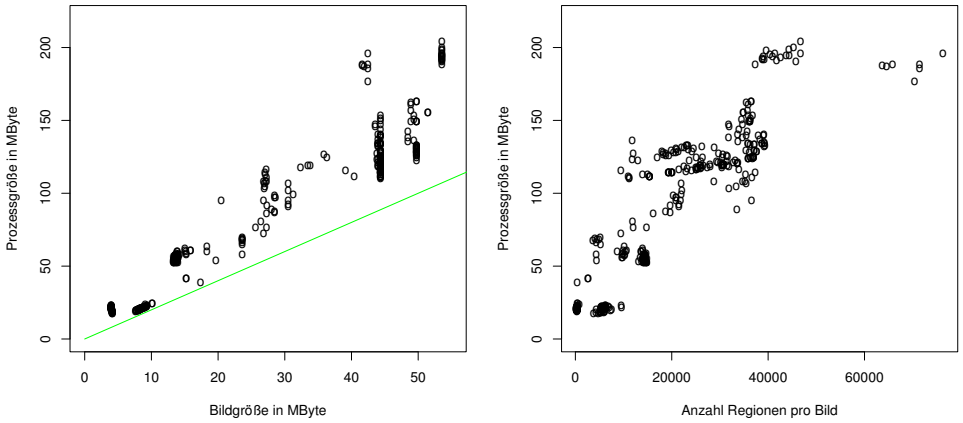



Abbildung 8.18: Die maximale Prozessgröße während der Gittersegmentierung in Abhängigkeit von der Eingabebildgröße und der Anzahl segmentierter Regionen. Die grüne Linie zeigt den Speicherbedarf für Roh- und Medianbilder an.

# 9 Diskussion

Vor der abschließenden Bewertung der Ergebnisse sollen noch einmal die wichtigsten Ziele dieser Arbeit zusammengestellt werden.

Die interaktiv durchgeführte Mikroarraybildauswertung ist ein Hindernis bei der Schaffung von Expressionsdatenbanken und bei der Automatisierung von Mikroarrayexperimenten, weil einerseits durch die manuelle Segmentierung viel Routinearbeit verursacht wird und andererseits der Benutzereingriff einen nicht gut reproduzierbaren Einfluss auf die Datenauswertung darstellt.

Daher sollten in dieser Arbeit erstens automatische Bildverarbeitungsverfahren zur Gittersegmentierung der Mikroarraybilder und zweitens robuste Verfahren zur Messpunktsegmentierung gefunden werden, um den Einfluss von Benutzerinteraktion in der Gittersegmentierung möglichst klein zu halten. Besonders bei dieser zweiten Fragestellung ist die Evaluation von Lösungsansätzen schwierig, weil es keine wirklichen Kalibrierdaten mit bekannten, korrekten oder optimalen Auswertungsergebnissen gibt.

## 9.1 Automatisierung der Gittersegmentierung

Die in dieser Arbeit vorgestellten Methoden zur Gittersegmentierung besitzen mehrere Vorteile gegenüber anderen Ansätzen. Es wird nur die Eingabe der Struktur der Messpunktanordnung verlangt, nicht aber die genauen Abmessungen. Die Bildauflösung und andere Parameter der verwendeten Geräte werden jeweils aus den Bildern geschätzt und müssen daher nicht extern kalibriert werden. Die Gittersegmentierung mit dem MZF-Modell arbeitet mit diskreten Hypothesenmengen, worin ein weiterer praktischer Vorteil besteht: Selbst wenn die automatische Segmentierung falsche Hypothesen auswählen sollte, ist das Ergebnis nicht unbedingt nutzlos, denn es ist für den Benutzer erheblich einfacher, aus einer kleinen, diskreten Menge von Verschiebungen die richtige auszuwählen als ein frei deformierbares Gitter pixelgenau im Bild zu positionieren.

Die zur Systemevaluation verwendete Stichprobe von Mikroarraybildern ist sehr heterogen im Hinblick auf Herkunft, verwendete Geräte und Schwierigkeitsgrad der Segmentierung. Die Daten stammen zum großen Teil aus dem Laboralltag und stellen keine „veröffentlichungsfähige“ Vorauswahl dar. Auch wenn nach wie vor visuelle Kontrolle und ggf. interaktive Korrektur der Gittersegmentierung notwendig sind, darf man die auf dieser Stichprobe erreichten 95% korrekt segmentierter Gitter als Fortschritt gegenüber den wenigen veröffentlichten Ergebnissen werten, denn in keiner der bekannten Arbeiten wird eine vergleichbar komplexe Stichprobe verwendet. Darüber hinaus sind (wie oben erwähnt) auch Teilergebnisse praktisch nützlich.

Die Heterogenität der Stichprobe spiegelt sich auch in den detaillierten Ergebnissen wieder. Die Serien von weniger verunreinigten Arrays werden meist vollständig richtig

segmentiert, wogegen bei stärkeren Kontaminationen und anderen Bildstörungen die Fehlerrate erwartungsgemäß höher ist.

Die Vollständig unüberwachte Gittersegmentierung und Rückweisung von unsicheren Segmentierungen ist bisher nicht möglich, weil das heuristische MZF-Modell die Bildinhalte nicht genau genug beschreibt. Daher sind die Restenergien des Modells im Allgmeinen nicht sinnvoll interpretierbar. Das Modell hat aber in der beschriebenen Form den Vorteil, auf verschiedenste Arten von Mikroarraybildern anwendbar zu sein.

Bemerkenswert ist, dass die Messpunktdetektion mit dem aktiven Konturmodell und dem modellbasierten Bildenergieschwellwert ähnlich gute Ergebnisse liefert wie die Messpunktdetektion mit gelernten Klassifikatoren. Es werden also auch ohne klassifizierte Stichproben von Bildern einzelner Messpunkte sehr gute Ergebnisse erreicht.

Die Laufzeit der Gittersegmentierung liegt mit maximal 10 Minuten selbst mit den aufwändigeren Messpunktdetektionsverfahren in der Größenordnung der Bildaufnahmedauer und die Speicheranforderungen sind ebenfalls leicht zu erfüllen. Der größte Teil der Rechenzeit wird für die Messpunktsegmentierung verbraucht, die ggf. auch gut parallel implementiert werden kann. Mit der einfachen Messpunktdetektion, die bei qualitativ guten Bildern völlig ausreicht um korrekte Gittersegmentierungen zu bekommen, sinkt die Laufzeit auf unter drei Minuten.

Das System ist also für den Einsatz in Hochdurchsatzanwendungen geeignet und die Gittersegmentierung mit den hier beschriebenen Methoden insgesamt auf jeden Fall von praktischem Nutzen.

## 9.2 Quantitative Bildauswertung

Den wichtigsten Beitrag dieser Arbeit zum Bereich der quantitativen Bildauswertung stellt die aktive Kontursegmentierung dar.

Im Vergleich zu den anderen untersuchten automatischen Verfahren werden hiermit die besten Resultate erzielt, und die Ergebnisse der aufwändig manuell bearbeiteten Auswertung mit dem kommerziellen System werden gut reproduziert. Das Evaluationskriterium der Konsistenz replizierter Messwerte lässt diese Schlüsse bei gleichzeitiger Beachtung der Verteilung der gesamten Messwerte zu, auch wenn keine absoluten Referenzdaten vorhanden sind.

Die Ergebnisse sind auch auf der Ebene der biologischen Interpretation der Messwerte sowohl im Vergleich mit dem kommerziellen System, als auch isoliert gesehen, plausibel. Der unabhängigen Beurteilung der verschiedenen Verfahren anhand der Erklärbarkeit der Daten an dem biologischen Modell sind durch die Komplexität der in den Evaluationsexperimenten untersuchten Lebewesen Grenzen gesetzt.

Beim Vergleich der Robustheit bezüglich kleiner Variationen der Gittersegmentierung, die z. B. durch Benutzereingriffe hervorgerufen werden, schneidet die aktive Kontursegmentierung ebenfalls sehr gut ab.

Die Ergebnisse der Mann-Whitney-Segmentierung sind weniger genau, weil die Voraussetzung für die Anwendung dieses Verfahrens, nämlich die vorherige Segmentierung der Hintergrundpixel der Vergleichsstichprobe, praktisch schwerer zu erbringen ist als die grobe Radiusschätzung der Messpunktregionen, die zur Initialisierung der aktiven Kontursegmentierung notwendig ist. Diese Tatsache unterstreicht die Eignung der aktiven Kontursegmentierung für die quantitative Mikroarrayauswertung.

Die Laufzeit liegt, abgesehen von der Hintergrundintensitätsschätzung, im gleichen Rahmen wie bei der Gittersegmentierung. Da die quantitative Auswertung unüberwacht ablaufen kann, ist die Laufzeit hier weniger kritisch und es besteht noch Optimierungspotential.

Die Reproduktion der Ergebnisse der manuellen Auswertung mit dem automatischen Verfahren stellt einen erheblichen Fortschritt dar.

Insgesamt zeigen die Ergebnisse, dass man den oben formulierten Zielen mit den in dieser Arbeit entwickelten Verfahren deutlich näher gekommen ist.

# 10 Zusammenfassung und Ausblicke

Abschließend werden die Arbeit und ihre Ergebnisse zusammengefasst und Ansatzpunkte für weitere Forschung diskutiert.

## 10.1 Zusammenfassung

Das Thema der Arbeit sind Bildverarbeitungsmethoden für die automatische Segmentierung und quantitative Auswertung von Bildern aus Mikroarray-Expressionsexperimenten.

Zu Beginn behandelt die Arbeit die biologischen Grundlagen der Genexpression, worunter man die Prozesse versteht, die die Synthese von Proteinen abhängig von der in der DNA kodierten Erbinformation steuern. Die Genexpression wird grob in die Stufen Transkription und Translation eingeteilt, die aus der genomischen DNA die Boten-RNA und daraus wiederum die Proteine synthetisieren.

Mit Mikroarrayexperimenten wird die Genexpression auf der Ebene der Boten-RNA untersucht. Die Methode stellt einen parallelisierten Hybridisierungsansatz dar, in dem die sequenzspezifische Doppelstrangbildung der DNA zur gleichzeitigen Messung der Mengen vieler (einige 1000 bis einige 10000) RNA-Sequenzen einer Probe genutzt wird. Die RNA wird dazu in DNA zurückübersetzt und dabei mit Fluoreszenzfarbstoffen markiert. Als Mikroarray bezeichnet man das Hybridisierungssubstrat, auf das Sonden-Punkte von DNA aller zu untersuchenden Sequenzen aufgedruckt sind. Bei der Hybridisierung bilden sich Doppelstränge von Sonden-DNA und der farbmarkierten Proben-DNA, die durch Aufnahme von Fluoreszenzbildern des Substrats nachgewiesen wird.

Mikroarrayexperimente eignen sich besonders zur Automatisierung, weshalb Bedarf für automatische Bildsegmentierung besteht. Die quantitative Auswertung erfordert genaue Segmentierung der gedruckten Punkte und verursacht großen Arbeitsaufwand, wenn sie interaktiv durchgeführt wird. Von automatischen Verfahren zur Messpunktsegmentierung darf man neben Zeitersparnis auch verbesserte Reproduzierbarkeit der Ergebnisse erwarten.

Expressionsdaten aus Mikroarrayexperimenten sollten sinnvollerweise analog zu Sequenzdaten in Datenbanken gesammelt werden, damit sie optimal genutzt werden können. Dazu müssen die Rohdaten mit standardisierten und allgemein anwendbaren Methoden ausgewertet werden.

Die Voraussetzungen der Anwendungsumfelder und die typischen Eigenschaften von Mikroarraybildern motivieren die in dieser Arbeit behandelten Bildverarbeitungsmethoden und den Systementwurf. Wesentliche Eigenschaften des Systems sind seine Anwendbarkeit auf verschiedene Typen von Mikroarrays und die Kalibrationsfreiheit.

Diese Eigenschaften sind wichtig, damit Daten aus verschiedenen Quellen leichter integriert werden können.

Die Kalibrationsfreiheit wird durch Nutzung der regelmäßigen Messpunktanordnung erreicht. Die Basis hierzu bilden Verteilungsmodelle der Abstände der nächsten Nachbarn der lokal periodisch angeordneten Messpunkte und der zufällig verteilten Partikel. Sie ermöglichen mit großer Robustheit gleichzeitig die Auswahl eines guten Regionenbildes aus mehreren alternativen Schwellwertsegmentierungen der Grauwertbilder und die Schätzung der Messpunktzeilen- und -spaltenabstände. Durch periodische Verkettung werden zu den Messpunktgittern gehörende Regionen erkannt.

Aus Achsenprojektionen der verketteten Regionen liest man Näherungen der Gitterpositionen ab und erhält durch Verschieben um einige Messpunktzeilen- und Spaltenabstände diskrete Mengen von Hypothesen für die Platzierung jedes Gitters.

Die Gittersegmentierung wird als global optimale Auswahl von Hypothesen für alle Gitter des Bildes gelöst. Die heuristische Bewertung einzelner Hypothesen und der Verträglichkeit von Hypothesen für benachbarte Gitter ist als Markov-Zufallsfeld (MZF) modelliert. Die Messpunktgitter des zu segmentierenden Arrays sind die Knoten des MZF, deren Zustandsmengen die lokal an den gefundenen Messpunktregionen ausgerichteten Verschiebungshypothesen bilden. Die Potentialfunktionen des MZF modellieren die Einzelhypothesenbewertung und die Abhängigkeiten der Auswahl von Hypothesen an verschiedenen Knoten. Die Gittersegmentierung erfolgt durch Minimierung der MZF-Energiefunktion. Aus Effizienzgründen muss dazu ein Näherungsverfahren benutzt werden, das aber praktisch in den meisten Fällen die optimale Lösung findet.

Die Berechnung der Potentialfunktionen des MZF erfordert die zuverlässige Erkennung von Messpunkten auf den Gitterplätzen der Hypothesen. Die Schwellwertregionensegmentierung ist hierfür nicht immer ausreichend. Leistungsfähigere Methoden für diesen Zweck sind der Eigenwertansatz und die aktive Kontursegmentierung. Das aktive Konturmodell muss kontinuierlich formuliert und mit einem gradientenrichtungsabhängigen Bildenergieterm verwendet werden, damit es auch bei eng benachbarten Messpunkten und sehr unterschiedlichen Messpunktgrößen robust anwendbar ist. Die Klassifikationsaufgabe in der Messpunkterkennung kann mit überwacht gelernten Klassifikatoren gelöst werden. Bei der aktiven Kontursegmentierung ist auch die Klassifikation mit einem Schwellwert für den Bildenergieterm des Modells möglich, der an der empirischen Verteilung der Kantenstärke des jeweiligen Bildes kalibrierbar ist und ohne klassifizierte Stichprobe auskommt.

Die Gittersegmentierung wird an einer Stichprobe von 387 Mikroarraybildern evaluiert, die 23 verschiedene Arten von Arraybildern enthält. Über 95 % der Gitter dieser Stichprobe werden von dem automatischen System mit moderatem Ressourceneinsatz korrekt segmentiert, wodurch die praktische Relevanz der Ergebnisse belegt wird.

Die quantitative Bildauswertung erfordert die möglichst genaue Segmentierung der Messpunkte. Zu diesem Zweck werden ein einfaches geometrisches Verfahren, die oft verwendete Mann-Whitney-Segmentierung und die aktive Kontursegmentierung untersucht.

Die quantitative Bildauswertung wird auf mehrere Arten evaluiert, weil keine absoluten Kalibrierdaten zur Verfügung stehen. Erstens werden die Messwerte aus dem hier beschriebenen System direkt untereinander und mit Messwerten von dem frei erhältlichen Programm Scanalyze und der kommerziellen Mikroarraybildauswertung

Imagene verglichen. Zweitens wird die Konsistenz der Messwerte replizierter Messpunkte bei Verwendung der verschiedenen Methoden untersucht, wodurch die absolute Genauigkeit besser beurteilt werden kann als durch Vergleiche der Methoden untereinander. Drittens werden die Rangfolgen differentiell exprimierter Sequenzen eines Referenzexperimentes bei Verwendung der aktiven Konturesegmentierung und der kommerziellen Bildauswertung gegenübergestellt, die man als Endergebnis einer typischen Experimentauswertung erhält.

Die aktive Konturesegmentierung erweist sich als sehr leistungsfähig, denn sie ermöglicht die automatisierte Reproduktion von interaktiv erstellten Auswertungen, die vergleichsweise großen Arbeitsaufwand verursachen.

## 10.2 Ausblicke

Die Gittersegmentierung ist aus der Sicht der Bildverarbeitung die am schwierigsten zu automatisierende Teilaufgabe der Mikroarrayauswertung. Der MZF-Ansatz hat sich hier zwar als recht leistungsfähig erwiesen, aber das Ziel vollautomatischer Verarbeitung konnte noch nicht erreicht werden.

Genauere Modelle für die Messpunktgitter könnten hier weiterhelfen. Man könnte Daten aus vorangegangenen Mikroarrayexperimenten, die Sondensequenzen (wenn bekannt) oder ihre Annotation nutzen, um grobe Vorhersagen über die zu erwartenden Messpunktsignale zu bekommen. Dazu ist neben Abfragen entsprechender Datenbanken oder sequenzbasierten Methoden zur Vorhersage stark exprimierter Gene (siehe z. B. McHardy und andere [77]) Detailwissen über das jeweilige Experiment erforderlich. Auch die Kenntnis von Kalibriermesspunkten oder Replikaten innerhalb der Gitter, die charakteristische Muster erzeugen, könnte zur Modellierung besserer MZF-Potentiale genutzt werden. Insgesamt muss hierzu die Bildverarbeitung viel umfassender mit anderen Bioinformatik-Werkzeugen integriert werden als bisher erforderlich. Standardisierte Schnittstellen wie MAGE-ML können in Zukunft helfen, die nötige Information generisch zugänglich zu machen. In speziellen Anwendungen, bei denen auf das Arraydesign Einfluss genommen werden kann, sind sicherlich auch mit geringerem technischen Aufwand noch deutliche Verbesserungen zu erreichen. Durch Aufdrucken von reinen Farbstoffpunkten an bekannten Positionen der Arrays und durch Verwendung entsprechender MZF-Potentiale könnte sehr wahrscheinlich eine zuverlässige Rückweisung unsicherer Segmentierungen erreicht werden.

Wenn umgekehrt richtig segmentierte Gitter mit hoher Wahrscheinlichkeit erkannt werden könnten, ist auch die Nutzung von Wissen über die Gesamtanordnung der Gitter vielversprechend. Die Geometrie der Gitteranordnung ändert sich bei gemeinsam gedruckten Arrays bis auf globale Rotationen und Translationen nicht und kann daher auch aus Beispielen geschätzt werden, wenn sie nicht vorgegeben ist.

Darüber hinaus könnte die gesamte Struktur des MZF-Modells überdacht werden. Denkbar wäre ein hierarchisches Modell, dessen untere Ebene wie das MZF-Modell von Carstensen und Hartelius [26] die Aufgabe des Suchprozesses zur Verkettung von Messpunktobjekten übernimmt. Dadurch würde die Trennung von Modellierung und Algorithmen verbessert, die mit dem MZF-Modell schon teilweise gelungen ist.

Ein Ansatzpunkt für mehr grundlagenorientierte Arbeit könnte das Problem des nur für Einzelbeobachtungen geeigneten Beobachtungsmodells für das MZF sein (siehe

S. 77). Beobachtungen auf Paaren von Knoten oder gar Cliquen höherer Ordnung dürften auch in anderen, nicht datennahen MZF-Anwendungen relevant sein, so dass ein allgemeineres Beobachtungsmodell von Interesse wäre.

Im Bereich der quantitativen Bildauswertung besteht vorrangig Bedarf für ein effizientes Verfahren zur Modus-Schätzung der Histogramme der Hintergrundintensität. Eine einfache Lösung hierfür könnte die getrennte Berechnung von Histogrammen in Fenstern zwischen Messpunktzentren sein, die dann nach Bedarf entsprechend der jeweiligen Hintergrundregion nach Abb. 7.3 addiert werden, ohne erneut auf das Bild zuzugreifen.

Alternativ dazu könnte ein leistungsfähigeres Schätzverfahren für den Modus der Verteilung benutzt werden. Die Artikel von Novak [80] und Jones, Marron und Sheather [55] diskutieren statistische Verfahren, die das Problem der Schätzung des häufigsten Stichprobenelements eleganter und auch mit einer kleineren Anzahl von Beispielen lösen. Die Autoren untersuchen dazu Verfahren zur Schätzung von Verteilungsdichten durch Glättung mit datenabhängig parametrisierten Operatoren. Mit einem solchen Verfahren könnte die Schätzung eventuell auch mit wenigen, zufällig ausgewählten Intensitätswerten aus den Hintergrundregionen durchgeführt und durch die kleinere Datenmenge der Rechenaufwand gesenkt werden.

Weitere Evaluation der quantitativen Bildauswertung mit Daten aus weiteren Quellen ist ebenfalls wünschenswert.

# A Grundlagen biologischer Methoden

## A.1 Technische und biologische DNA-Replikation

Bei der Zellteilung muß notwendigerweise die DNA der Chromosomen verdoppelt werden. Zu diesem Zweck gibt es wieder ein kompliziertes System von Enzymen, welches den Doppelstrang spaltet und jeweils die beiden Einzelstränge zu neuen $\alpha$-Helices ergänzt. Die Enzyme, die dabei zum Einsatz kommen, die DNA-Polymerasen, können nur in der 5'-3'-Richtung auf der Einzelstrang-DNA arbeiten, da sie Nukleotide nur an freie 3'-Enden anhängen können. Da die beiden Stränge der Doppelhelix gegenläufig orientiert sind, verkompliziert das die Rekonstruktion des Einzelstranges mit freiem 5'-Ende. In der Zelle wird dies Problem mit sog. RNA-Primern gelöst. Dies sind kurze RNA-Moleküle, die an den Einzelstrang hybridisieren, womit eine stückweise Rückwärts-Synthese des zweiten Halbstrangs vom 5'-Ende aus möglich wird. Die Primer werden durch spezielle Enzyme wieder abgelöst und die Lücken mit DNA-Nukleotiden aufgefüllt.

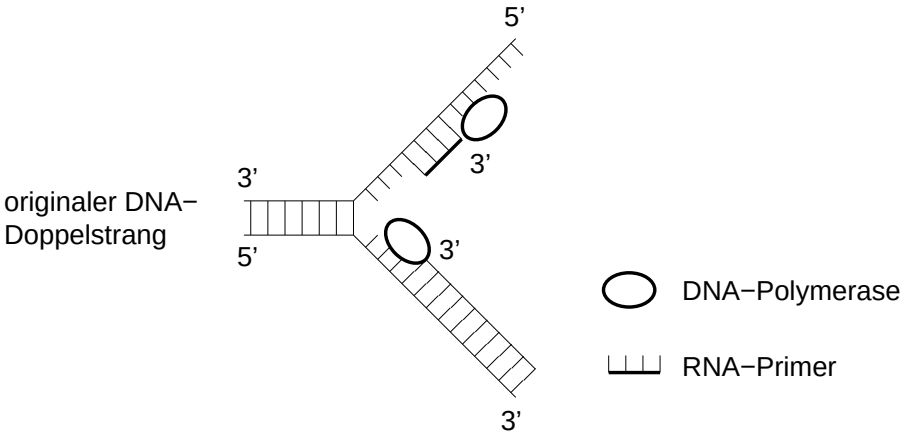



Abbildung A.1: Stark vereinfachte Darstellung der DNA-Replikation

Diesem Prinzip ist die Polymerase-Kettenreaktion (Polymerase Chain Reaction, kurz PCR) nachempfunden, die die sequenz-selektive Vervielfältigung von DNA im Reagenzglas ermöglicht. Ein PCR-Zyklus beginnt mit der Denaturierung eines DNA-Doppelstrangs durch hohe Temperatur. Dann werden zwei Sorten Primer zugegeben, deren Sequenz komplementär zu den 3'-Enden der beiden Einzelstränge ist. Sie binden deshalb dort an den beiden DNA-Strängen, wobei das Primer-3'-Ende jeweils zum Rest des Einzelstrangs zeigt. Damit ist die Voraussetzung für die Vervollständigung des Doppelstrangs durch DNA-Polymerase gegeben. Der Zyklus kann dann wieder-

holt werden, wobei die erzeugte DNA-Menge wegen der Verdopplung theoretisch exponentiell anwächst. Die Primer müssen künstlich in einem sog. Oligonukleotid-Syntheseverfahren hergestellt werden. Die Diplomarbeit von T. Nick [79] befaßt sich eingehender mit PCR-Verfahren im Zusammenhang mit Microarrays.

## A.2 Reverse Transkription und Herstellung von cDNA

Es gibt einen Vorgang, bei dem eine RNA-Sequenz wieder in eine DNA-Sequenz übersetzt wird, also genau die Umkehrung der gewöhnlichen Transkription. Er tritt im Infektionsvorgang sogenannter Retroviren auf. Retroviren tragen RNA als Erbmaterial. Die RNA ist sehr ähnlich einer mRNA des Wirtsorganismus aufgebaut. Bei der Infektion einer Zelle durch das Virus wird die Virus-RNA in das Zellplasma eingebracht. Der Proteinsyntheseapparat der Zelle verarbeitet die Virus-RNA, welche mehrere Proteine kodiert. Zunächst wird eine RNA-abhängige DNA-Polymerase hergestellt. Dies Enzym ist in der Lage, die erwähnte Übersetzung von RNA zu DNA durchzuführen und wird deshalb auch als reverse Transkriptase bezeichnet. Es dient dem Virus dazu, seine Erbinformation in das Genom des Wirtsorganismus, das ja in Form von DNA vorliegt, einzuschleusen. Die Virus-RNA kodiert noch weitere Proteine, z.B. die Strukturproteine für die Virushülle und Rezeptoren für die Wirtszelle. Retroviren können sehr lange im Genom des Wirts verbleiben und sogar mitvererbt werden, ohne aktiv zu werden („Latenz“ ). Wenn die Virus-Gene einmal exprimiert werden, bedeutet das die Herstellung von Virus-Partikeln durch die Zelle selbst. Dies endet mit der Auflösung der Zellwände und Freisetzung neuer Viruspartikel („Virulenz“ ).

Die reverse Transkription wird technisch genutzt, um in vitro aus mRNA komplementäre DNA (cDNA, complementary DNA) herzustellen.

Dazu müssen Primer verwendet werden, die am 3’-Ende der RNA hybridisieren und dadurch die Startstelle der reverse Transkription steuern. Bei der reversen Transkription eukaryotischer mRNA wird das charakteristische Poly-A-Ende als Primerbindestelle genutzt.

Weitere Voraussetzung ist die zugabe von Nukleotiden, aus denen die reverse Transkriptase den komplementären DNA-Strang aufbaut.

Allgemein ist das Verfahren nützlich, weil DNA stabiler und daher leichter zu handhaben ist als RNA. Bei der Probenvorbereitung der Mikroarrayhybridisierung wird die nötige Farbmarkierung der Proben durch Verwendung von Nukleotiden erreicht, in die Farbstoffmoleküle eingebaut sind.

# B Verteilungsmodelle für Abstände zum $k$-nächsten Nachbarn

## B.1 Zufällig verteilte Störregionen

In der Gitterkonstantenschätzung wird ein Verteilungsmodell der Abstände von „zufällig verteilten“ Regionen zu ihren nächsten Nachbarn benutzt. Es wird angenommen, dass die Störregionen einem *Poisson-Prozess* [65, 93] folgen.

**Definition 1.** Poissonverteilung
Eine diskrete Zufallsvariable $T \in \mathbb{N}$ heißt *poissonverteilt mit Parameter* $\lambda$, wenn ihre Verteilungsdichte

$$p(T = t) = \frac{\lambda^t}{t!} e^{-}\lambda$$

lautet. Erwartungswert und Varianz von $T$ sind gleich dem Verteilungsparameter $\lambda$.

**Definition 2.** 2D - Poisson-Prozess

Sei $D \subset \mathbb{R}^2$. Es wird ein Zufallsprozess betrachtet, der Punkte in $D$ erzeugt. Sei $N(A), A \subset D$ die Anzahl von zufälligen Punken in A. Die Menge der Zufallsvariablen $\{N(A) : A \subset D\}$ heißt *Poisson-Prozess* auf $D$ mit Dichteparameter $\rho$, wenn die folgenden Bedingungen gelten:

1. $N(A)$ ist poissonverteilt mit Parameter $rm(A)$, wobei $m(A)$ die Fläche (oder ein allgemeineres Lebesgue-Maß) von $A$ ist

2. Wenn $(A_1, A_2, \ldots)$ paarweise disjunkte Teilmengen von $D$ sind, dann ist $(N(A_1), N(A_2), \ldots)$ eine Sequenz von paarweise unabhängigen Zufallsvariablen

Es sei ein punkterzeugender Poissonprozess mit Dichte $\rho$ im $\mathbb{R}^2$ gegeben. $C_t$ sei die Kreisscheibe mit Radius $t$ um den Ursprung.

$$C_t = \left\{ \begin{pmatrix} \phi \\ r \end{pmatrix} \in \mathbb{R}^2 : r < t \right\}$$

$M_t$ sei die Anzahl der Punkte in $C_t$ und $Z_k$ sei der Abstand des $k$-nächsten Punktes zum Ursprung.
$M_t$ ist wegen der 1. Eigenschaft des Poissonprozesses poissonverteilt mit Parameter $\rho\pi t^2$.

Außerdem ist der Abstand des $k$-nächsten Punktes zum Ursprung genau dann kleiner oder gleich t, wenn $C_t$ mindestens $k$ Punkte enthält:

$$\begin{array}{rcl} Z_k \leq t & \Leftrightarrow & M_t \geq k \\ \Rightarrow \quad P(Z_k \leq t) & = & P(M_t \geq k) \end{array}$$

Es gilt also

$$\begin{array}{rcll} P(M_t = k) & = & e^{-2\pi t^2} \dfrac{\left(2\pi t^2\right)^k}{k!} & \text{(Poissonverteilung von } M_t\text{)} \\ \Rightarrow \quad P(M_t \geq k) & = & 1 - \displaystyle\sum_{j=0}^{k-1} e^{-2\pi t^2} \frac{\left(2\pi t^2\right)^j}{j!} & \text{(B.1)} \end{array}$$

Gleichung (B.1) liefert also die Verteilungsfunktion des Abstandes zum $k$-nächsten Nachbarn. Die Verteilungsdichte $p_{Z_k}$ muss nur noch durch Differenzieren ausgerechnet werden:

$$\begin{aligned} p_{Z_k}(t) &= \frac{d}{dt} P(Z_k < t) \\ &= \frac{d}{dt} P(Z_k \leq t) \\ &= \frac{d}{dt} P(M_t \geq k) \\ &= -\left( e^{-2\pi\rho t^2}(-2)\pi\rho t \sum_{j=0}^{k-1} \frac{(\rho\pi t^2)^j}{j!} + e^{-2\pi\rho t^2} \sum_{j=0}^{k-1} \frac{2j(\rho\pi t^2)^{j-1}\rho\pi t}{j!} \right) \\ &= e^{-2\pi\rho t^2} 2\pi\rho t \sum_{j=0}^{k-1} \frac{(\rho\pi t^2)^j}{j!} - \frac{j(\rho\pi t^2)^{j-1}}{j!} \\ &= e^{-2\pi\rho t^2} 2\pi\rho t \left( \frac{(\rho\pi t^2)^{k-1}}{(k-1)!} - \ldots + \ldots - 0 \right) \\ &= \frac{2(\pi\rho)^k t^{2k-1}}{(k-1)!} e^{-2\pi\rho t^2} \end{aligned}$$

(Siehe auch Übungsaufgabe 3, Kapitel 7 in [93])

Wegen der *Gedächtnislosigkeit* [93] des Poissonprozesses gilt dieses Verteilungsmodell nicht nur für die Nachbarabstände des Ursprungs, sondern auch innerhalb der Punktmenge.

## B.2 Messpunktregionen

In diesem Abschnitt wird das Verteilungsmodell für die Abstände von Messpunktregionenschwerpunkten auf benachbarten Gitterknoten mit gaussverteilten Störungen hergeleitet werden. Dabei werden die horizontalen und vertikalen Nachbarschaftssysteme aus Abschnitt 6.1.1 betrachtet. Primär sind daher Tripel von Messpunktregionen

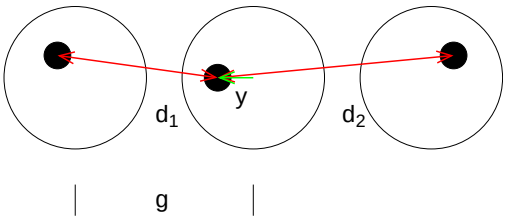



Abbildung B.1: Die Anordnung von drei benachbarten Regionenschwerpunkten (schwarze Punkte), die mit gaussverteilten Abweichungn (Standardabweichung angedeutet durch Kreise) um die idealen Gitterpositionen gestreut liegen

interessant, bei denen die Störungen der idealen Gitteranordnung entscheiden, welche der beiden äußeren Regionen nächster Nachbar der mittleren Region ist. Mehr als zwei Regionen können in den horizontalen bzw. vertikalen Nachbarschaftssystemen nicht als nächste Nachbarn in frage kommen, und das Regionentripel muss symmetrisch angeordnet sein.

Jeder der drei Regionenschwerpunkte sei radialsymmetrisch gaussverteilt mit Standardabweichung $\sigma$ um den jeweiligen idealen Gitterknoten. Damit eine geschlossene Lösung ermittelt werden kann, ist muss angenommen werden, dass $\sigma$ klein gegenüber dem Gitterabstand $g$ ist. Dann spielt nur die Störung entlang der Verbindungsachse der Gitterknoten eine Rolle und das Problem wird eindimensional. Die Verteilungsdichte der Störung des mittleren Gitterknotens entlang der Achse sei

$$p_y(x) = \mathcal{N}_{0,\sigma} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x}{2\sigma^2}}$$

Gesucht ist die Verteilung $F_d$ des Abstandes $d$ zum nächsten Nachbarn der mittleren Regionenschwerpunktes, also des Minimums der Abstände $d_1$ und $d_2$:

$$F_d(r) = P(d_1 < r \vee d_2 < r)$$

Die Verteilungsfunktion wird als Wahrscheinlichkeit des Ereignisses „$d_1$ oder $d_2$ sind kleiner als $r$“angesetzt. Die Verteilungen der Abstände $d_1$ und $d_2$ müssen aus Symmetriegründen gleich sein.

Zunächst betrachtet man die Verbundverteilung von $d_1$ und $d_2$. Wenn man y als gegeben ansieht, sind $d_1$ und $d_2$ unabhängig und man darf die Verteilungsfunktion multiplikativ zerlegen:

$$\begin{aligned} F_{d_1,d_2}(r_1, r_2) &= P(d_1 < r1, d_2 < r_2) && \text{(B.2)} \\ &= \int_{y=-\infty}^{y=\infty} P(d_1 < r_1, d_2 < r_2|y) p_y(y) dy && \text{(B.3)} \\ &= \int_{y=-\infty}^{y=\infty} F(d_1|y) F(d_2|y) * p_y(y) dy && \text{(B.4)} \end{aligned}$$

Die Dichte der bedingten Verteilungen $F(d_1|y)$ und $F(d_2|y)$ sind

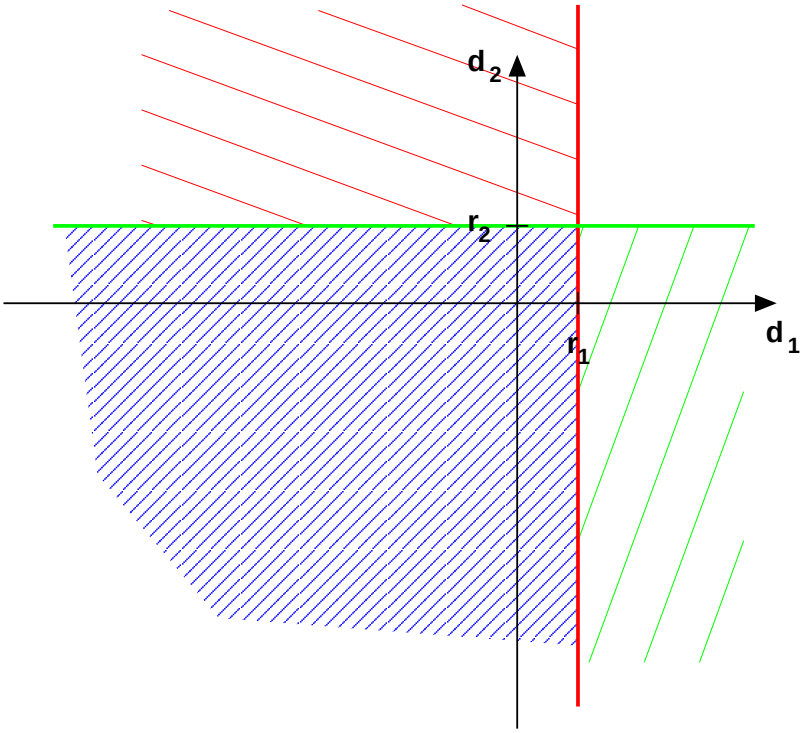



Abbildung B.2: Der gemeinsame Wahrscheinlichkeitsraum für die Zufallsvariablen $d_1$ und $d_2$. Die Flächen, die den disjunkten Ereignissen $(d_1 < r_1) \wedge (d_2 > r_2)$, $(d_2 < r_2) \wedge (d_1 > r_1)$ und $(d_1 < r_1) \wedge (d_2 < r_2)$ entsprechen sind rot,grün bzw. blau schraffiert. Das Ereignis $(d_1 < r \vee d_2 < r)$, mit dem die Verteilung des Abstands zum nächsten Nachbarn zusammenhängt, gehört zur gesamten schraffierten Fläche.

$$f_{d_1|y}(d_1|y) = N_{g,\sqrt{2}\sigma}(d_1 - y)$$
$$f_{d_2|y}(d_2|y) = N_{g,\sqrt{2}\sigma}(d_1 + y)$$

denn die Standardabweichungen der Störungen an allen drei Gitterknoten sind gleich, so dass ein Standardresultat über die Normalverteilung anwendbar ist (siehe z. B. [68]). Die Dichte der Verbundverteilung ist damit

$$f_{d_1,d_2}(r_1, r_2) = \int_{y=-\infty}^{y=\infty} f_{d_1|y}(r_1|y) f_{d_2|y}(r_2|y) f_y(y) dy$$

Für dies bestimmte Integral findet man wegen der Separierbarkeit der Exponentialfunktion und der Grenzwerteigenschaften der Errorfunktion (siehe unten) die folgende Lösung:

$$f_{d_1,d_2}(r_1, r_2) = \frac{1}{2\sqrt{3}s^2\pi} e^{-\frac{1}{3s^2}\left(r_1^2 - 3r_1 g + 3g^2 + r_2^2 - 3r_2 g + r_1 r_2\right)}$$

Aus der Abbildung B.2 liest man die Verteilung des Abstands zum nächsten Nachbarn ab:

$$
\begin{aligned}
P(d_1 < r \vee d_2 < r) &= P((d_1 < r) \wedge (d_2 > r)) && \text{(B.5)}\\
&+ P((d_2 < r) \wedge (d_1 > r)) && \text{(B.6)}\\
&+ P((d_1 < r) \wedge (d_2 < r)) && \text{(B.7)}\\
&= \int_{r_1=-\infty}^{r_1=r} \int_{r_2=r}^{r_2=\infty} f_{d_1,d_2}(r_1, r_2) dr_1 dr_2 && \text{(B.8)}\\
&+ \int_{r_1=r}^{r_1=\infty} \int_{r_2=-\infty}^{r_2=r} f_{d_1,d_2}(r_1, r_2) dr_1 dr_2 && \text{(B.9)}\\
&+ \int_{r_1=-\infty}^{r_1=r} \int_{r_2=-\infty}^{r_2=r} f_{d_1,d_2}(r_1, r_2) dr_1 dr_2 && \text{(B.10)}
\end{aligned}
$$

Die Verteilungsdichte des Abstands der mittleren Region zum nächsten Nachbarn ist daher

$$
\begin{aligned}
f_d(r) &= \frac{d}{dr} P(d_1 < r \vee d_2 < r) && \text{(B.11)}\\
&= \frac{1}{2\sigma\sqrt{\pi}} \left( 1 + \text{erf}\left( \frac{\sqrt{3}g - r}{2\sigma} \right) e^{-\frac{(g-r)^2}{4\sigma^2}} \right) && \text{(B.12)}
\end{aligned}
$$

Die nach der Differentiation verbleibenden Integrationen aus Gl. B.10 lassen sich mit längeren aber trivialen Umformungen auf die Definition der Errorfunktion

$$
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^{\pi} e^{-x'^2} dx'
$$

zurückführen. Diese Funktion geht für große positive Argumente gegen 1 und für negative Argumente mit großem Betrag gegen -1. Die Abbildung B.3 zeigt das Bild der Dichtefunktion $f_d(r)$ mit $\sigma = 0,5$ und $g = 5$.

Die Näherungsrechnungen in diesem Abschnitt werden in erster Linie dadurch gerechtfertigt, dass die Verwendung der Normalverteilungsapproximation zu guten Ergebnissen der Gitterkonstantenschätzung führt[1]. Das hier berechnete $f_d$ ist nicht mit der realen Verteilung der Abstände beim 2D-Gitter identisch, so dass die Normalverteilungsapproximation sozusagen eine Näherung zweiter Stufe ist. Numerische Experimente (siehe Abb. B.4) zeigen, dass die Abweichung zwischen Normalverteilungsapproximation mit den Parametern $(g-\sigma, \sigma)$ und der oben unter der Annahme $g >> \sigma$ hergeleiteten 1D-Näherung $f_d$ mit typischen Werten von $g$ und $\sigma$ unter 5% bleibt.

[1] Darum wurde auf die detaillierte Ausführung der Rechnungen verzichtet

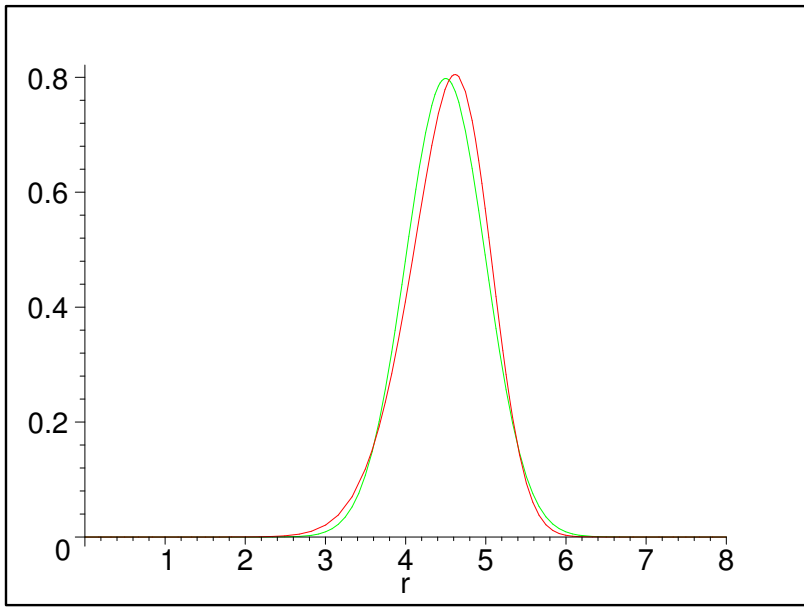



Abbildung B.3: Die Dichte $f_d(r)$ mit $\sigma = 0,5$ und $g = 5$ (rot) und zum Vergleich die Gaussverteilung mit Mittelwert $g - \sigma$ und Standardabweichung $\sigma$

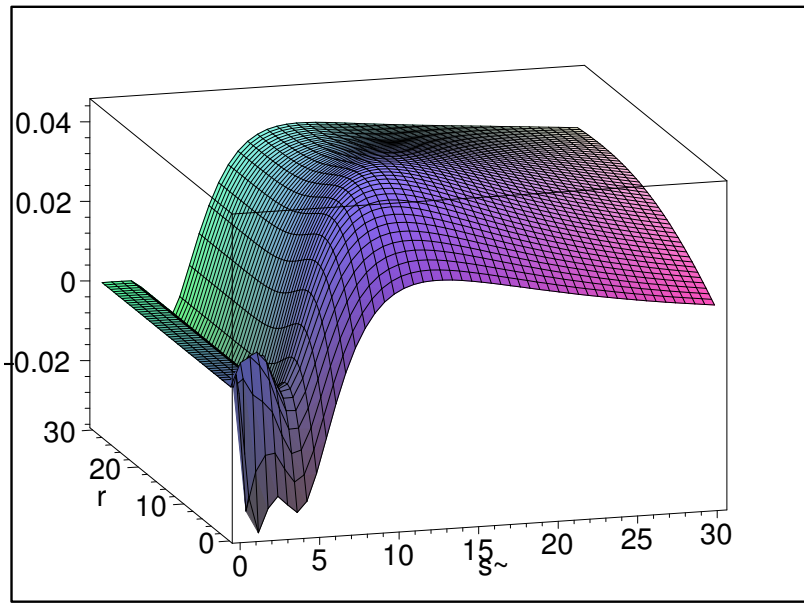



Abbildung B.4: Der Fehler der Approximation von $f_d$ durch die Gaussverteilung mit Mittelwert $g - \sigma$ und Standardabweichung $\sigma$

# C Auswertungen der *Medicago*-Nodulationsexperimente

Weitere Informationen über alle Klone des Mt6kRIT-Mikroarrays sind in der MENS-Datenbank (Medicago EST Navigation System) unter
`http://medicago.toulouse.inra.fr/EST` und unter
`http://www.Genetik.Uni-Bielefeld.DE/MolMyk/scinfo/analysis/cdna-arrays.shtml`
abrufbar [69].

Tabelle C.1: M-sortierte Liste der Sondensequenzen auf dem Mt6kRIT-Mikroarray des Nodulationsexperimentes Nod 4, bei Bildauswertung mit der Software Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 1 | 4 | 00746MtBB | XII.C. Unknown function [NODULIN] MTN11 PCR product | 3.3259e-10 | 9.97 | 6.59 |
| 2 | 125 | GVN-55E6 | nodulin 25 PCR product | 0.17922 | 11.39 | 6.58 |
| 3 | 9 | mtgmabc120001a02 | leghemoglobin, TC6726-homolog, 70% zu VfLb29 PCR product | 2.9629e-06 | 9.11 | 6.34 |
| 4 | 13 | 10695MtBB | XIII. No homology PCR product | 1.1673e-05 | 9.45 | 5.74 |
| 5 | 1 | JVCPG44 | MtN17 PCR product | 3.9672e-06 | 10.87 | 5.57 |
| 6 | 17 | mt–aac120001a02 | leghemoglobin, TC13379-homolog, 73% zu VfLb29 PCR product | 0.00013479 | 9.21 | 5.55 |
| 7 | 3 | JVCPG42 | MtN15 PCR product | 1.9891e-07 | 10.89 | 5.43 |
| 8 | 12 | JVCPG49 | MtN22 PCR product | 7.8993e-08 | 10.02 | 5.41 |
| 9 | 6 | 00583MtBB | V. Primary metabolism [NODULIN] LEGHEMOGLOBIN 1 PCR product | 8.8528e-06 | 10.87 | 5.27 |
| 10 | 5 | GVN-55B13 | MtN1 PCR product | 4.3875e-06 | 9.24 | 5.26 |
| 11 | 11 | 40167MtBB | XII.C. Unknown function MtN9-LIKE PROTEIN PCR product | 2.8304e-07 | 9.37 | 5.16 |
| 12 | 2 | 10685MtBB | XIII. No homology PCR product | 4.4835e-12 | 11.84 | 5.07 |
| 13 | 8 | GVN-72A6 | nodulin 25 PCR product | 0.00026334 | 10.59 | 4.95 |
| 14 | 7 | MsCA | carbonic anhydrase PCR product | 2.1917e-09 | 11.87 | 4.68 |
| 15 | 10 | JVCPG29 | MtN1 PCR product | 4.7209e-09 | 10.96 | 4.66 |
| 16 | 21 | JVCPG38 | MtN11 PCR product | 9.8385e-08 | 9.23 | 4.63 |
| 17 | 15 | 00068MtBB | XII.C. Unknown function [NODULIN] MTN1 PRECURSOR PCR product | 1.9969e-12 | 10.87 | 4.58 |
| 18 | 14 | 00156MtBB | V. Primary metabolism [NODULIN] CARBONIC ANHYDRASE PCR product | 7.4378e-12 | 10.90 | 4.44 |
| 19 | 35 | 20268MtBB | XII.C. Unknown function [NODULIN] MtN6 PCR product | 9.3673e-07 | 9.07 | 4.43 |
| 20 | 29 | 40019MtBB | IX. Protein synthesis and processing [NODULIN] MATRIX METALLOENDOPROTEINASE PCR product | 1.3284e-06 | 9.34 | 4.34 |
| 21 | 34 | 90646MtBB | XIII. No homology PCR product | 2.6306e-05 | 9.36 | 4.34 |
| 22 | 16 | 00646MtBB | XIII. No homology PCR product | 1.0082e-07 | 11.23 | 4.33 |
| 23 | 33 | JVCPG36 | MtN9 PCR product | 6.4541e-07 | 9.02 | 4.31 |
| 24 | 23 | 00120MtBB | XII.C. Unknown function [NODULIN] MTN29 PCR product | 1.2718e-05 | 9.50 | 4.20 |
| 25 | 18 | 00186MtBB | XII.C. Unknown function [NODULIN] MTN16 PCR product | 1.8126e-05 | 12.08 | 4.18 |
| 26 | 22 | 90571MtBB | XII.C. Unknown function [NODULIN] MtN19 PCR product | 1.1138e-06 | 10.30 | 4.08 |
| 27 | 28 | JVCPG25 | ENOD11 PCR product | 0.043359 | 11.57 | 3.88 |
| 28 | 20 | JVCPG58 | ENOD40 PCR product | 3.0044e-13 | 11.17 | 3.85 |
| 29 | 32 | JVCPG34 | MtN6 PCR product | 1.0848e-05 | 8.88 | 3.81 |
| 30 | 19 | 00043MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 4.4771e-13 | 11.59 | 3.77 |
| 31 | 38 | mt–aac120001a01 | leghemoglobin, TC10534-homolog, 77% zu VfLb29 PCR product | 0.0046476 | 9.20 | 3.76 |
| 32 | 31 | 10497MtBB | XIII. No homology PCR product | 0.00024226 | 9.60 | 3.76 |
| 33 | 67 | 10766MtBB | XIII. No homology PCR product | 0.018017 | 9.00 | 3.62 |
| 34 | 43 | 93260MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 1.1894e-05 | 10.20 | 3.57 |
| 35 | 25 | 90910MtBB | XII.C. Unknown function [NODULIN] MTN20 PCR product | 6.9675e-06 | 9.44 | 3.57 |
| 36 | 570 | 91284MtBC | IV. Vesicular trafficking secretion and protein sorting TRANSMEMBRANE TRANSMEMBRANE PROTEIN PCR product | 1 | 7.98 | 3.56 |
| 37 | 30 | Globin | Globin PCR product | 0.043753 | 9.25 | 3.45 |
| 38 | 24 | 00281MtBB | XII.C. Unknown function [NODULIN] MTN22 PCR product | 4.3263e-09 | 12.37 | 3.41 |
| 39 | 27 | 00060MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 1.6246e-10 | 11.19 | 3.38 |
| 40 | 26 | GVN-7016 | enod40 PCR product | 2.0835e-10 | 11.75 | 3.33 |
| 41 | 36 | JVCPG52 | MtN25 PCR product | 3.0308e-06 | 10.60 | 3.29 |
| 42 | 44 | 50535MtBB | XII.C. Unknown function PCR product | 0.0015516 | 9.05 | 3.25 |
| 43 | 48 | JVCPG48 | MtN21 PCR product | 0.00023998 | 8.74 | 3.23 |
| 44 | 45 | JVCPG51 | MtN24 PCR product | 5.3576e-07 | 8.84 | 3.13 |
| 45 | 52 | JVCPG31 | MtN3 PCR product | 2.1157e-05 | 9.02 | 3.12 |

Tabelle C.1: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 46 | 41 | JVCPG46 | MtN19 PCR product | 8.6432e-06 | 9.91 | 3.08 |
| 47 | 54 | 20131MtBA | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 9.5792e-07 | 9.53 | 3.07 |
| 48 | 82 | 91055MtBB | XIII. No homology PCR product | 0.8675 | 8.70 | 3.03 |
| 49 | 57 | 93119MtBB | XIII. No homology PCR product | 0.0011595 | 9.17 | 3.01 |
| 50 | 60 | 10582MtBB | VIII. Gene expression and RNA metabolism TRANSCRIPTION FACTOR PCR product | 1.5544e-05 | 8.64 | 2.98 |
| 51 | 5749 | 10888MtBA | X. Signal transduction EF-HAND CALCIUM-BINDING DOMAIN PCR product | 0 | 7.22 | 2.94 |
| 52 | 56 | 45657MtBA | X. Signal transduction and post-translational regulation ADENYL CYCLASE PCR product | 1.0448e-07 | 9.72 | 2.88 |
| 53 | 47 | mt–aac120001a03 | leghemoglobin, TC3089-homolog, 70% zu VfLb29 PCR product | 0.00021042 | 11.30 | 2.77 |
| 54 | 70 | 10430MtBC | III. Membrane transport MAJOR INTRINSIC PROTEIN (NODULIN26-LIKE ) PCR product | 0.00027967 | 8.84 | 2.71 |
| 55 | 49 | GVN-71A1 | nodulin 26 PCR product | 0.0091115 | 9.67 | 2.71 |
| 56 | 39 | 10811MtBB | X. Signal transduction REMORIN PCR product | 6.026e-05 | 9.26 | 2.71 |
| 57 | 50 | JVCPG27 | ENOD16 PCR product | 1.8062e-06 | 9.30 | 2.69 |
| 58 | 42 | JVCPG33 | MtN5 PCR product | 6.2795e-06 | 11.60 | 2.68 |
| 59 | 78 | 00198MtBB | V. Primary metabolism THIOREDOXIN M-TYPE CHLOROPLAST PRECURSOR PCR product | 0.007948 | 9.21 | 2.63 |
| 60 | 46 | GVN-64D16 | enod8 PCR product | 0.0019447 | 11.34 | 2.61 |
| 61 | 37 | GFP | Green Fluorescent Protein PCR product | 0.0027793 | 8.08 | 2.61 |
| 62 | 40 | JVCPG37 | MtN10 PCR product | 0.00029735 | 12.73 | 2.60 |
| 63 | 55 | 00344MtBB | XII. Miscellaneous lectin PCR product | 0.08463 | 8.85 | 2.60 |
| 64 | 68 | 91074MtBB | XIII. No homology PCR product | 7.2393e-05 | 9.84 | 2.52 |
| 65 | 58 | 00380MtBB | XII.C. Unknown function [NODULIN] MTN15 PCR product | 2.551e-05 | 9.22 | 2.52 |
| 66 | 51 | 30542MtBB | XIII. No homology PCR product | 3.1656e-05 | 9.30 | 2.45 |
| 67 | 3352 | 40202MtBB | XIII. No homology PCR product | 0 | 9.38 | 2.44 |
| 68 | 106 | 10968MtBB | VI. Secondary metabolism and hormone metabolism PROFUCOSIDASE PRECURSOR PCR product | 1 | 9.12 | 2.38 |
| 69 | 66 | JVCPG40 | MtN13 PCR product | 0.00088779 | 9.97 | 2.36 |
| 70 | 61 | 90842MtBB | XII.C. Unknown function PD156016 PCR product | 1.861e-07 | 9.20 | 2.36 |
| 71 | 76 | 50599MtBB | XIII. No homology PCR product | 5.7864e-05 | 10.21 | 2.36 |
| 72 | 221 | 40162MtBB | XII.C. Unknown function TRANSMEMBRANE MtN3 PCR product | 0.091299 | 9.65 | 2.34 |
| 73 | 1795 | 92019MtBC | XII.C. Unknown function PCR product | 0 | 10.01 | 2.34 |
| 74 | 63 | 00588MtBB | XII.C. Unknown function [NODULIN] MtN25 PCR product | 0.0002479 | 10.15 | 2.31 |
| 75 | 59 | 00240MtBB | XII.C. Unknown function [NODULIN] EARLY NODULIN 12 PRECURSOR PCR product | 8.4908e-06 | 12.58 | 2.28 |
| 76 | 72 | 00388MtBB | V. Primary metabolism THIOREDOXIN M-TYPE CHLOROPLAST PRECURSOR PCR product | 0.017441 | 9.15 | 2.26 |
| 77 | 93 | 00546MtBB | XII.C. Unknown function PCR product | 0.0023252 | 9.61 | 2.24 |
| 78 | 95 | 90141MtBA | VI. Secondary metabolism and hormone metabolism GLUCOSYLTRANSFERASE PCR product | 5.6774e-06 | 9.28 | 2.17 |
| 79 | 77 | GVN-74I6 | MtN21 PCR product | 0.00015766 | 9.21 | 2.17 |
| 80 | 74 | MtAe34 | Unknown function PCR product | 0.0023971 | 10.68 | 2.12 |
| 81 | 84 | 10203MtBB | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 7.4946e-05 | 9.97 | 2.12 |
| 82 | 81 | 93406MtBB | I. Cell Wall PROFUCOSIDASE PRECURSOR PCR product | 1 | 9.40 | 2.12 |
| 83 | 75 | 10307MtBB | VII. Chromatin and DNA metabolism histone H2A PCR product | 5.3122e-07 | 10.46 | 2.10 |
| 84 | 109 | 30497MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 0.21398 | 9.25 | 2.10 |
| 85 | 62 | JVCPG53 | MtN26 PCR product | 1.7881e-06 | 9.33 | 2.07 |
| 86 | 130 | 00127MtBC | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 0.016595 | 9.63 | 2.06 |
| 87 | 79 | 93045MtBA | V. Primary metabolism ALBUMIN PCR product | 2.6975e-06 | 10.73 | 2.04 |
| 88 | 83 | 91157MtBB | XIII. No homology PCR product | 0.001758 | 8.50 | 2.03 |
| 89 | 167 | 90621MtBB | XIII. No homology PCR product | 0.24288 | 9.90 | 2.02 |
| 90 | 88 | GVN-72E9 | nodulin 26 PCR product | 0.0012574 | 9.20 | 2.02 |
| 91 | 53 | 50612MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 0.00034696 | 12.97 | 2.00 |
| 92 | 105 | 90695MtBB | XIII. No homology PCR product | 0.00012666 | 9.89 | 1.99 |
| 93 | 92 | 00235MtBB | XII.B. Abiotic stimuli and development OSMOTIN/THAUMATIN-LIKE PROTEIN PRECURSOR PCR product | 3.981e-05 | 10.21 | 1.94 |
| 94 | 98 | 40082MtBB | XII.A. Defense and cell rescue ENDO-13-BETA-GLUCOSIDASE PRECURSOR PCR product | 0.0028808 | 9.16 | 1.94 |
| 95 | 101 | 20129MtBB | IV. Vesicular trafficking secretion and protein sorting ANNEXIN PCR product | 0.00047281 | 9.99 | 1.92 |

Tabelle C.1: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 96 | 80 | 10073MtBB | XII.C. Unknown function PCR product | 8.6857e-05 | 10.21 | 1.91 |
| 97 | 90 | 20267MtBB | XII.C. Unknown function [NODULIN] MTN26 PCR product | 0.065374 | 8.61 | 1.88 |
| 98 | 151 | 91023MtBB | XII.C. Unknown function [NODULIN] PCR product | 1 | 9.67 | 1.87 |
| 99 | 71 | 10489MtBB | IX. Protein synthesis and processing 30S RIBOSOMAL PROTEIN S20 PCR product | 1.5088e-05 | 12.88 | 1.86 |
| 100 | 3667 | 50830MtBC | XII.B. Abiotic stimuli and development OSMOTIN-LIKE PROTEIN PCR product | 1 | 8.02 | 1.85 |
| 101 | 85 | JVCPG56 | MtN29 chimeric PCR product | 3.0023e-07 | 12.39 | 1.85 |
| 102 | 103 | 30105MtBB | V. Primary metabolism BETA-AMYLASE PCR product | 0.0086124 | 9.44 | 1.83 |
| 103 | 94 | 45591MtBB | VI. Secondary metabolism and hormone metabolism ISOFLAVONE-O-METHYTRANSFERASE PCR product | 3.5553e-08 | 10.09 | 1.82 |
| 104 | 64 | 91068MtBB | XII.C. Unknown function PCR product | 4.8502e-05 | 13.12 | 1.78 |
| 105 | 120 | 00290MtBB | XII.C. Unknown function PD155440 PCR product | 0.092886 | 9.33 | 1.78 |
| 106 | 65 | MtAe27 | Unknown function PCR product | 1 | 9.12 | 1.76 |
| 107 | 136 | 00553MtBB | VIII. Gene expression and RNA metabolism TRANSCRIPTION FACTOR PCR product | 1 | 10.12 | 1.76 |
| 108 | 122 | 00515MtBB | XIII. No homology PCR product | 1 | 8.87 | 1.74 |
| 109 | 73 | GVN-65K1 | leghemoglobin 1 PCR product | 0.0012689 | 13.60 | 1.71 |
| 110 | 129 | 45429MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 1 | 8.60 | 1.67 |
| 111 | 139 | 90927MtBB | XIII. No homology PCR product | 0.70179 | 8.38 | 1.65 |
| 112 | 124 | 00153MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 3.7939e-06 | 9.91 | 1.64 |
| 113 | 91 | 90776MtBB | XIII. No homology PCR product | 0.050269 | 11.25 | 1.61 |
| 114 | 111 | 00643MtBB | XII.C. Unknown function PCR product | 0.0029688 | 9.16 | 1.61 |
| 115 | 96 | JVCPG43 | MtN16 PCR product | 4.6957e-09 | 11.09 | 1.60 |
| 116 | 102 | 50454MtBB | VI. Secondary metabolism and hormone metabolism TERPENE SYNTHASE PCR product | 0.020645 | 8.79 | 1.59 |
| 117 | 976 | 92093MtBC | XII.C. Unknown function PD102438 PCR product | 1 | 7.22 | 1.59 |
| 118 | 86 | 90689MtBB | III. Membrane transport CATIONIC AMINO ACID TRANSPORTER PCR product | 1 | 8.39 | 1.56 |
| 119 | 114 | 50550MtBB | V. Primary metabolism LIPASE PCR product | 0.0026236 | 9.66 | 1.55 |
| 120 | 87 | 10107MtBB | IX. Protein synthesis and processing PROTEASOME SUBUNIT BETA TYPE PCR product | 0.00010778 | 13.07 | 1.55 |
| 121 | 113 | JVCPG35 | MtN7 PCR product | 0.80142 | 8.70 | 1.51 |
| 122 | 89 | MsPEPC | PEP carboxylase PCR product | 0.00027241 | 9.61 | 1.50 |
| 123 | 115 | 30415MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE 23 KDA SUBUNIT PCR product | 0.0011788 | 10.66 | 1.49 |
| 124 | 116 | 30416MtBB | XII. Miscellaneous GLUCAN 14-ALPHA-GLUCOSIDASE PCR product | 0.00027051 | 10.94 | 1.48 |
| 125 | 100 | 10443MtBB | XII.C. Unknown function PCR product | 0.0019532 | 8.23 | 1.48 |
| 126 | 97 | 40035MtBA | I. Cell Wall CELL WALL INVERTASE BETA-FRUCTOFURANOSIDASE PCR product | 1.7573e-07 | 9.53 | 1.46 |
| 127 | 121 | 00383MtBB | XIII. No homology PCR product | 0.016855 | 8.67 | 1.45 |
| 128 | 291 | 90572MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 1 | 9.89 | 1.45 |
| 129 | 99 | 91045MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 0.00055108 | 12.36 | 1.45 |
| 130 | 126 | 00246MtBB | XII.C. Unknown function CYS RICH PROTEIN PCR product | 0.0019583 | 9.41 | 1.42 |
| 131 | 132 | 30530MtBB | XIII. No homology PCR product | 0.0014721 | 10.05 | 1.41 |
| 132 | 112 | 00436MtBB | XIII. No homology PCR product | 0.14478 | 10.71 | 1.39 |
| 133 | 107 | 90940MtBB | XII.C. Unknown function [NODULIN] MtN7 PCR product | 0.0032791 | 10.18 | 1.38 |
| 134 | 108 | 00093MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 0.00053775 | 11.47 | 1.38 |
| 135 | 171 | 00578MtBB | V. Primary metabolism [NODULIN] ADENINE PHOSPHORIBOSYLTRANSFERASE APRT PCR product | 0.0019689 | 8.94 | 1.32 |
| 136 | 146 | JVCPG55 | MtN28 PCR product | 1 | 8.07 | 1.31 |
| 137 | 118 | 10168MtBB | I. Cell Wall PECTIN METHYL-ESTERASE PCR product | 9.4134e-06 | 10.87 | 1.28 |
| 138 | 175 | 20111MtBA | V. Primary metabolism GLUTAMATE DEHYDROGENASE PCR product | 0.63246 | 9.54 | 1.28 |
| 139 | 141 | 30515MtBB | XIII. No homology PCR product | 6.4376e-06 | 10.41 | 1.27 |
| 140 | 176 | JVCPG41 | MtN14 PCR product | 1 | 9.05 | 1.26 |
| 141 | 255 | 10295MtBB | XII.C. Unknown function [NODULIN] MtN14 PCR product | 0.2933 | 9.45 | 1.25 |
| 142 | 150 | 90901MtBB | VIII. Gene expression and RNA metabolism RESPONSE REGULATOR PCR product | 0.64542 | 9.17 | 1.25 |
| 143 | 123 | 30270MtBB | XII.C. Unknown function PCR product | 1 | 12.61 | 1.24 |
| 144 | 119 | 40161MtBB | X. Signal transduction and post-translational regulation PROTEIN KINASE DOMAIN CONTAINING PROTEIN PCR product | 1 | 7.99 | 1.21 |

Tabelle C.1: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 145 | 197 | 93135MtBB | XII.C. Unknown function PCR product | 1 | 10.26 | 1.21 |
| 146 | 117 | 10261MtBB | IX. Protein synthesis and processing UBIQUITIN-CONJUGATING ENZYME E2-17 KDA PCR product | 0.030662 | 12.69 | 1.21 |
| 147 | 134 | 90971MtBB | XIII. No homology PCR product | 0.052503 | 9.59 | 1.20 |
| 148 | 165 | 50719MtBC | XIII. No homology PCR product | 1 | 11.17 | 1.20 |
| 149 | 195 | 10607MtBC | I. Cell Wall CAFFEIC ACID O-METHYLTRANSFERASE PCR product | 1 | 9.02 | 1.19 |
| 150 | 143 | 00457MtBB | XIII. No homology PCR product | 1 | 9.19 | 1.18 |
| 151 | 2291 | 91090MtBB | V. Primary metabolism 4-HYDROXYPHENYLPYRUVATE DIOXYGENASE PCR product | 1 | 9.09 | 1.18 |
| 152 | 186 | 20102MtBC | V. Primary metabolism L-ASCORBATE OXIDASE PRECURSOR PCR product | 0.0075711 | 9.62 | 1.17 |
| 153 | 155 | GVSN-24A24 | MtN13 PCR product | 2.1462e-07 | 12.51 | 1.15 |
| 154 | 104 | 50559MtBB | X. Signal transduction and post-translational regulation SER/THR PROTEIN KINASE PCR product | 0.11742 | 8.53 | 1.15 |
| 155 | 145 | 00617MtBB | XII.C. Unknown function PCR product | 1 | 11.71 | 1.15 |
| 156 | 133 | 20204MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 0.76068 | 8.05 | 1.14 |
| 157 | 198 | 30343MtBB | V. Primary metabolism ESTERASE PCR product | 1 | 8.69 | 1.14 |
| 158 | 144 | GVSN-9F12 | nodulin 75 PCR product | 0.020953 | 13.90 | 1.13 |
| 159 | 183 | KV3-21B21 | MtN21-like PCR product | 1 | 9.43 | 1.10 |
| 160 | 142 | 45014MtBA | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L32 RP49 PCR product | 0.49766 | 11.07 | 1.10 |
| 161 | 208 | 10312MtBC | XII.A. Defense and cell rescue CHITINASE PCR product | 0.12708 | 8.98 | 1.09 |
| 162 | 127 | 10021MtBC | I. Cell Wall CELL WALL PROTEIN PCR product | 7.4963e-08 | 13.20 | 1.08 |
| 163 | 275 | 50661MtBC | XII.C. Unknown function DEM PROTEIN PCR product | 1 | 9.37 | 1.07 |
| 164 | 147 | 00331MtBC | V. Primary metabolism ACID PHOSPHATASE PCR product | 0.0019897 | 10.90 | 1.07 |
| 165 | 138 | 10026MtBB | XII.C. Unknown function PCR product | 1 | 11.08 | 1.07 |
| 166 | 140 | 10690MtBB | XII.A. Defense and cell rescue MTN13 PCR product | 1 | 10.87 | 1.06 |
| 167 | 157 | 90870MtBB | XIII. No homology PCR product | 1 | 8.35 | 1.06 |
| 168 | 3827 | 91885MtBC | V. Primary metabolism GLYCOSYL TRANSFERASE HOMOLOG PCR product | 0 | 6.77 | 1.05 |
| 169 | 164 | 10403MtBB | IX. Protein synthesis and processing PROTEIN DISULFIDE ISOMERASE PRECURSOR PCR product | 0.15298 | 12.32 | 1.04 |
| 170 | 539 | 50648MtBB | XII.C. Unknown function ALLERGEN-LIKE PROTEIN PCR product | 1 | 8.30 | 1.03 |
| 171 | 1055 | 40134MtBB | V. Primary metabolism NITRILASES / CYANIDE HYDRATASE SIGNATURE CONTAINING PROTEIN PCR product | 1 | 8.97 | 1.03 |
| 172 | 156 | MtGSa | glutamine synthetase a PCR product | 0.021484 | 9.75 | 1.02 |
| 173 | 158 | 00203MtBB | XII.C. Unknown function PCR product | 1 | 10.77 | 1.01 |
| 174 | 249 | 93357MtBB | XIII. No homology PCR product | 1 | 9.30 | 1.01 |
| 175 | 137 | 91489MtBC | XIII. No homology PCR product | 1 | 13.77 | 1.00 |
| 176 | 152 | 90857MtBB | III. Membrane transport TRANSPORTER PCR product | 0.10384 | 11.17 | 1.00 |
| 177 | 232 | 90944MtBB | XII.C. Unknown function [PUTATIVE NODULIN] GLYCINE-RICH PROTEIN PRECURSOR PCR product | 1 | 8.54 | 0.99 |
| 178 | 281 | 00088MtBC | IX. Protein synthesis and processing PEPTIDYL-PROLYL CIS-TRANS ISOMERASE (CYCLOPHILIN) PCR product | 1.6207e-06 | 10.40 | 0.98 |
| 179 | 154 | 20062MtBB | V. Primary metabolism L-ASPARAGINASE PCR product | 0.02668 | 12.60 | 0.96 |
| 180 | 131 | 10472MtBB | XII.C. Unknown function SEVEN TRANSMEMBRANE DOMAIN PROTEIN PCR product | 1.6385e-06 | 12.66 | 0.96 |
| 181 | 153 | 40187MtBB | XIII. No homology PCR product | 1 | 9.26 | 0.96 |
| 182 | 188 | 30198MtBB | II. Cytoskeleton TUBULIN BETA CHAIN PCR product | 0.0088645 | 9.55 | 0.96 |
| 183 | 207 | 00187MtBB | V. Primary metabolism CELL WALL INVERTASE BETA-FRUCTOFURANOSIDASE PCR product | 1 | 9.10 | 0.94 |
| 184 | 159 | 93419MtBB | VI. Secondary metabolism and hormone metabolism CYTOCHROME P450 PCR product | 1 | 14.80 | 0.94 |
| 185 | 465 | 20154MtBC | V. Primary metabolism PYRUVATE DECARBOXYLASE ISOZYME PCR product | 0.3654 | 9.31 | 0.94 |
| 186 | 135 | 91067MtBB | XIII. No homology PCR product | 3.9563e-06 | 13.02 | 0.94 |
| 187 | 333 | 30097MtBA | V. Primary metabolism ASPARTATE AMINOTRANSFERASE PCR product | 1 | 9.33 | 0.94 |
| 188 | 250 | 00024MtBB | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S11 PCR product | 0.34306 | 9.98 | 0.93 |
| 189 | 4870 | 91892MtBC | XII.C. Unknown function PCR product | 0 | 10.03 | 0.93 |
| 190 | 3887 | 30370MtBC | XIII. No homology PCR product | 1 | 8.86 | 0.93 |
| 191 | 1057 | 50507MtBB | I. Cell Wall LACCASE PCR product | 1 | 9.05 | 0.92 |
| 192 | 230 | 00490MtBB | V. Primary metabolism LIPASE (CLASS 3) DOMAIN CONTAINING PROTEIN PCR product | 0.63888 | 9.85 | 0.92 |
| 193 | 166 | 10358MtBB | XII.C. Unknown function [NODULIN] PCR product | 0.13955 | 11.29 | 0.92 |
| 194 | 174 | 91038MtBB | XII.C. Unknown function PCR product | 0.28738 | 10.03 | 0.91 |

Tabelle C.1: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 195 | 148 | 50083MtBA | XII.C. Unknown function PCR product | 1 | 9.53 | 0.91 |
| 196 | 530 | 00736MtBB | XII.C. Unknown function PCR product | 1 | 9.24 | 0.91 |
| 197 | 226 | 91998MtBC | XIII. No homology PCR product | 0.0058235 | 9.93 | 0.91 |
| 198 | 203 | 90751MtBB | XIII. No homology PCR product | 1 | 9.29 | 0.91 |
| 199 | 2729 | 91464MtBC | XII.C. Unknown function PCR product | 1 | 8.16 | 0.90 |
| 200 | 2103 | 20042MtBA | XII. Miscellaneous LIPOXYGENASE Missing PCR product | 1 | 7.44 | 0.89 |
| 201 | 172 | MtSucS1_XbN61 | nodule-enhanced sucrose synthase MtSucS1 PCR product | 0.053383 | 10.63 | 0.89 |
| 202 | 329 | 90643MtBB | XII.C. Unknown function [NODULIN] MtN28 PCR product | 1 | 8.53 | 0.89 |
| 203 | 163 | HLG200 | HLG 1.0 well 200 PCR product | 0.0012553 | 12.06 | 0.89 |
| 204 | 239 | 45417MtBB | XIII. No homology PCR product | 0.0038328 | 10.47 | 0.89 |
| 205 | 415 | 20228MtBB | IX. Protein synthesis and processing PROTEIN DISULFIDE ISOMERASE PCR product | 0.25978 | 9.77 | 0.89 |
| 206 | 228 | 00437MtBB | XII. Miscellaneous NON-SPECIFIC LIPID TRANSFER-LIKE PROTEIN PCR product | 0.00062328 | 10.51 | 0.88 |
| 207 | 293 | 10058MtBA | XII.C. Unknown function PD001144 PCR product | 1 | 8.92 | 0.88 |
| 208 | 214 | 00740MtBB | III. Membrane transport MEMBRANE TRANSPORTER PCR product | 0.52122 | 10.25 | 0.88 |
| 209 | 216 | 90856MtBB | XII.C. Unknown function PD025544 PCR product | 0.085628 | 9.23 | 0.87 |
| 210 | 211 | 91662MtBC | XIII. No homology PCR product | 1 | 11.55 | 0.87 |
| 211 | 160 | 91319MtBC | XIII. No homology PCR product | 1 | 10.83 | 0.87 |
| 212 | 234 | 00328MtBB | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L11 PCR product | 0.35982 | 10.51 | 0.87 |
| 213 | 177 | 90648MtBB | XIII. No homology PCR product | 1 | 11.49 | 0.86 |
| 214 | 1369 | 91121MtBB | XIII. No homology no homology PCR product | 1 | 8.64 | 0.86 |
| 215 | 170 | 10733MtBB | XIII. No homology PCR product | 0.36936 | 11.95 | 0.86 |
| 216 | 3261 | 00697MtBB | III. Membrane transport AMINO ACID TRANSPORTER PCR product | 1 | 9.37 | 0.85 |
| 217 | 181 | 20373MtBB | I. Cell Wall ENDO-BETA-14-GLUCANASE PCR product | 1 | 8.15 | 0.85 |
| 218 | 686 | 50825MtBC | VII. Chromatin and DNA metabolism HISTONE H4 PCR product | 1 | 9.73 | 0.85 |
| 219 | 2256 | 10348MtBC | XII.C. Unknown function PCR product | 1 | 8.85 | 0.84 |
| 220 | 258 | 00254MtBB | VIII. Gene expression and RNA metabolism HIGH MOBILITY GROUP PROTEIN PCR product | 0.020584 | 10.67 | 0.84 |
| 221 | 270 | 30214MtBB | VII. Chromatin and DNA metabolism HISTONE H4 PCR product | 1 | 9.94 | 0.83 |
| 222 | 358 | 10987MtBC | XIII. No homology PCR product | 0.054207 | 9.66 | 0.83 |
| 223 | 276 | 10461MtBA | IV. Vesicular trafficking secretion and protein sorting COP-COATED VESICLE MEMBRANE PROTEIN P24 PRECURSOR PCR product | 0.76546 | 9.83 | 0.83 |
| 224 | 1544 | 91953MtBC | XIII. No homology PCR product | 0 | 9.39 | 0.83 |
| 225 | 370 | 91314MtBC | XII. Miscellaneous CALCIUM-BINDING PROTEIN PCR product | 0.80214 | 12.54 | 0.83 |
| 226 | 676 | 91119MtBB | XII.C. Unknown function PCR product | 1 | 8.97 | 0.81 |
| 227 | 248 | 90830MtBB | XIII. No homology PCR product | 0.028939 | 12.25 | 0.81 |
| 228 | 412 | 91717MtBC | PCR product | 1 | 9.74 | 0.80 |
| 229 | 69 | MtAe96 | Unknown function PCR product | 1 | 8.08 | 0.79 |
| 230 | 351 | 91127MtBB | XIII. No homology PCR product | 1 | 9.73 | 0.79 |
| 231 | 190 | 93062MtBA | V. Primary metabolism NADH-PLASTOQUINONE OXIDOREDUCTASE CHAIN 1 PCR product | 0.023125 | 10.82 | 0.79 |
| 232 | 951 | 91514MtBC | XIII. No homology PCR product | 1 | 9.71 | 0.79 |
| 233 | 589 | 10835MtBB | XII.C. Unknown function PD038106 PCR product | 1 | 8.80 | 0.79 |
| 234 | 179 | KV0-1A24 | chlorophyll a/b binding protein PCR product | 1 | 8.42 | 0.79 |
| 235 | 279 | 00439MtBC | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L38 PCR product | 0.37462 | 11.74 | 0.78 |
| 236 | 1111 | 50746MtBC | IV. Vesicular trafficking secretion and protein sorting TRANSLOCON-ASSOCIATED PROTEIN ALPHA SUBUNIT PRECURSOR PCR product | 1 | 9.64 | 0.78 |
| 237 | 278 | 50530MtBB | XII.C. Unknown function PCR product | 1 | 11.40 | 0.78 |
| 238 | 191 | 90604MtBB | XII.C. Unknown function PCR product | 1 | 12.71 | 0.78 |
| 239 | 222 | 90545MtBA | XIII. No homology PCR product | 1 | 11.83 | 0.77 |
| 240 | 271 | 00131MtBB | IX. Protein synthesis and processing INITIATION FACTOR 5A/MEMBRANE INTRINSIC PROTEIN PCR product | 1 | 10.79 | 0.77 |
| 241 | 1890 | 10498MtBB | XII.C. Unknown function PCR product | 1 | 9.12 | 0.77 |
| 242 | 1862 | 93052MtBC | XII. Miscellaneous LIPOXYGENASE Missing PCR product | 0 | 6.71 | 0.77 |

Tabelle C.1: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 243 | 243 | MtAMLs642 | Unknown PCR product | 0.26381 | 12.34 | 0.77 |
| 244 | 212 | 10310MtBB | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 1 | 10.11 | 0.76 |
| 245 | 257 | 10341MtBC | VII. Chromatin and DNA metabolism HISTONE H4 PCR product | 0.037829 | 9.81 | 0.76 |
| 246 | 352 | 10773MtBC | IV. Vesicular trafficking secretion and protein sorting SIGNAL RECOGNITION PARTICLE PCR product | 1 | 9.09 | 0.76 |
| 247 | 217 | 20302MtBC | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.00040606 | 11.76 | 0.76 |
| 248 | 818 | 91942MtBC | IX. Protein synthesis and processing TRANSAMIDASE PCR product | 1 | 8.66 | 0.76 |
| 249 | 213 | 10071MtBC | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S3A PCR product | 1 | 10.86 | 0.76 |
| 250 | 1398 | 50657MtBC | XII.C. Unknown function PCR product | 1 | 10.19 | 0.75 |

Tabelle C.2: M-sortierte Liste der Sondensequenzen auf dem Mt6kRIT-Mikroarray des Nodulationsexperimentes Nod 4, bei Bildauswertung mit dem hier beschriebenen System AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 1 | 5 | JVCPG44 | MtN17 PCR product | 1.4899e-09 | 12.30 | 4.53 |
| 2 | 12 | 10685MtBB | XIII. No homology PCR product | 4.7012e-14 | 12.84 | 4.29 |
| 3 | 7 | JVCPG42 | MtN15 PCR product | 3.1956e-11 | 12.38 | 4.27 |
| 4 | 1 | 00746MtBB | XII.C. Unknown function [NODULIN] MTN11 PCR product | 1.3579e-12 | 11.75 | 4.26 |
| 5 | 10 | GVN-55B13 | MtN1 PCR product | 1.4158e-10 | 11.85 | 4.26 |
| 6 | 9 | 00583MtBB | V. Primary metabolism [NODULIN] LEGHEMOGLOBIN 1 PCR product | 1.5595e-06 | 12.00 | 4.04 |
| 7 | 14 | MsCA | carbonic anhydrase PCR product | 1.3234e-10 | 13.07 | 3.97 |
| 8 | 13 | GVN-72A6 | nodulin 25 PCR product | 2.2385e-05 | 12.12 | 3.88 |
| 9 | 3 | mtgmabc120001a02 | leghemoglobin, TC6726-homolog, 70% zu VfLb29 PCR product | 1.0423e-05 | 10.99 | 3.86 |
| 10 | 15 | JVCPG29 | MtN1 PCR product | 4.2333e-12 | 11.98 | 3.81 |
| 11 | 11 | 40167MtBB | XII.C. Unknown function MtN9-LIKE PROTEIN PCR product | 5.9975e-11 | 11.12 | 3.72 |
| 12 | 8 | JVCPG49 | MtN22 PCR product | 1.3445e-11 | 11.38 | 3.70 |
| 13 | 4 | 10695MtBB | XIII. No homology PCR product | 0.040843 | 10.81 | 3.70 |
| 14 | 18 | 00156MtBB | V. Primary metabolism [NODULIN] CARBONIC ANHYDRASE PCR product | 2.9746e-12 | 11.81 | 3.67 |
| 15 | 17 | 00068MtBB | XII.C. Unknown function [NODULIN] MTN1 PRECURSOR PCR product | 5.1031e-13 | 11.85 | 3.66 |
| 16 | 22 | 00646MtBB | XIII. No homology PCR product | 5.7959e-09 | 12.13 | 3.59 |
| 17 | 6 | mt–aac120001a02 | leghemoglobin, TC13379-homolog, 73% zu VfLb29 PCR product | 5.661e-06 | 11.00 | 3.55 |
| 18 | 25 | 00186MtBB | XII.C. Unknown function [NODULIN] MTN16 PCR product | 9.3418e-07 | 13.16 | 3.52 |
| 19 | 30 | 00043MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 2.462e-14 | 12.40 | 3.42 |
| 20 | 28 | JVCPG58 | ENOD40 PCR product | 6.1282e-15 | 12.04 | 3.33 |
| 21 | 16 | JVCPG38 | MtN11 PCR product | 1.7362e-07 | 10.77 | 3.26 |
| 22 | 26 | 90571MtBB | XII.C. Unknown function [NODULIN] MtN19 PCR product | 2.4178e-09 | 11.51 | 3.06 |
| 23 | 24 | 00120MtBB | XII.C. Unknown function [NODULIN] MTN29 PCR product | 8.4671e-11 | 10.98 | 3.05 |
| 24 | 38 | 00281MtBB | XII.C. Unknown function [NODULIN] MTN22 PCR product | 1.2815e-09 | 13.11 | 3.03 |
| 25 | 35 | 90910MtBB | XII.C. Unknown function [NODULIN] MTN20 PCR product | 1.8041e-08 | 10.88 | 2.95 |
| 26 | 40 | GVN-70I6 | enod40 PCR product | 1.1574e-11 | 12.70 | 2.94 |
| 27 | 39 | 00060MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 3.0946e-11 | 11.96 | 2.93 |
| 28 | 27 | JVCPG25 | ENOD11 PCR product | 0.0014655 | 12.43 | 2.90 |
| 29 | 20 | 40019MtBB | IX. Protein synthesis and processing [NODULIN] MATRIX METALLOENDOPROTEINASE PCR product | 1.5187e-11 | 10.88 | 2.89 |
| 30 | 37 | Globin | Globin PCR product | 0.0077057 | 11.06 | 2.87 |
| 31 | 32 | 10497MtBB | XIII. No homology PCR product | 1.5613e-06 | 11.02 | 2.83 |
| 32 | 29 | JVCPG34 | MtN6 PCR product | 0.00074531 | 10.59 | 2.81 |
| 33 | 23 | JVCPG36 | MtN9 PCR product | 4.8565e-08 | 10.71 | 2.80 |
| 34 | 21 | 90646MtBB | XIII. No homology PCR product | 4.7397e-08 | 10.79 | 2.77 |
| 35 | 19 | 20268MtBB | XII.C. Unknown function [NODULIN] MtN6 PCR product | 0.00061568 | 10.61 | 2.69 |
| 36 | 41 | JVCPG52 | MtN25 PCR product | 2.1581e-06 | 11.59 | 2.63 |
| 37 | 61 | GFP | Green Fluorescent Protein PCR product | 0.0015758 | 10.38 | 2.60 |
| 38 | 31 | mt–aac120001a01 | leghemoglobin, TC10534-homolog, 77% zu VfLb29 PCR product | 0.00064479 | 10.78 | 2.53 |
| 39 | 56 | 10811MtBB | X. Signal transduction REMORIN PCR product | 2.508e-08 | 10.68 | 2.46 |
| 40 | 62 | JVCPG37 | MtN10 PCR product | 2.6232e-05 | 13.34 | 2.46 |
| 41 | 46 | JVCPG46 | MtN19 PCR product | 2.8434e-08 | 11.13 | 2.43 |
| 42 | 58 | JVCPG33 | MtN5 PCR product | 3.7933e-05 | 12.32 | 2.40 |
| 43 | 34 | 93260MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 6.2722e-09 | 11.27 | 2.33 |
| 44 | 42 | 50535MtBB | XII.C. Unknown function PCR product | 0.01066 | 10.58 | 2.24 |
| 45 | 44 | JVCPG51 | MtN24 PCR product | 1.5305e-08 | 10.56 | 2.21 |

Tabelle C.2: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 46 | 60 | GVN-64D16 | enod8 PCR product | 0.0017979 | 12.29 | 2.21 |
| 47 | 53 | mt–aac120001a03 | leghemoglobin, TC3089-homolog, 70% zu VfLb29 PCR product | 0.0019859 | 12.21 | 2.14 |
| 48 | 43 | JVCPG48 | MtN21 PCR product | 9.98e-09 | 10.68 | 2.13 |
| 49 | 55 | GVN-71A1 | nodulin 26 PCR product | 0.00056794 | 11.13 | 2.10 |
| 50 | 57 | JVCPG27 | ENOD16 PCR product | 9.5661e-05 | 10.89 | 2.05 |
| 51 | 66 | 30542MtBB | XIII. No homology PCR product | 0.046564 | 10.75 | 2.02 |
| 52 | 45 | JVCPG31 | MtN3 PCR product | 0.0011754 | 10.45 | 2.02 |
| 53 | 91 | 50612MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 3.2241e-05 | 13.47 | 2.01 |
| 54 | 47 | 20131MtBA | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 4.1542e-12 | 10.77 | 2.00 |
| 55 | 63 | 00344MtBB | XII. Miscellaneous lectin PCR product | 4.2178e-07 | 10.51 | 1.97 |
| 56 | 52 | 45657MtBA | X. Signal transduction and post-translational regulation ADENYL CYCLASE PCR product | 1.683e-06 | 10.67 | 1.94 |
| 57 | 49 | 93119MtBB | XIII. No homology PCR product | 5.7444e-05 | 10.26 | 1.93 |
| 58 | 65 | 00380MtBB | XII.C. Unknown function [NODULIN] MTN15 PCR product | 4.8798e-06 | 10.35 | 1.92 |
| 59 | 75 | 00240MtBB | XII.C. Unknown function [NODULIN] EARLY NODULIN 12 PRECURSOR PCR product | 1.4987e-05 | 13.34 | 1.91 |
| 60 | 50 | 10582MtBB | VIII. Gene expression and RNA metabolism TRANSCRIPTION FACTOR PCR product | 0.00031606 | 10.10 | 1.89 |
| 61 | 70 | 90842MtBB | XII.C. Unknown function PD156016 PCR product | 7.1745e-08 | 10.57 | 1.86 |
| 62 | 85 | JVCPG53 | MtN26 PCR product | 1.073e-08 | 10.77 | 1.86 |
| 63 | 74 | 00588MtBB | XII.C. Unknown function [NODULIN] MtN25 PCR product | 1.2874e-05 | 11.10 | 1.84 |
| 64 | 104 | 91068MtBB | XII.C. Unknown function PCR product | 2.4017e-07 | 13.64 | 1.79 |
| 65 | 106 | MtAe27 | Unknown function PCR product | 0.0088988 | 10.86 | 1.77 |
| 66 | 69 | JVCPG40 | MtN13 PCR product | 0.00021211 | 10.89 | 1.76 |
| 67 | 33 | 10766MtBB | XIII. No homology PCR product | 0.1108 | 10.23 | 1.76 |
| 68 | 64 | 91074MtBB | XIII. No homology PCR product | 6.0758e-07 | 10.91 | 1.75 |
| 69 | 229 | MtAe96 | Unknown function PCR product | 0.0010274 | 10.60 | 1.75 |
| 70 | 54 | 10430MtBC | III. Membrane transport MAJOR INTRINSIC PROTEIN (NODULIN26-LIKE ) PCR product | 0.0004174 | 10.34 | 1.74 |
| 71 | 99 | 10489MtBB | IX. Protein synthesis and processing 30S RIBOSOMAL PROTEIN S20 PCR product | 8.0594e-07 | 13.43 | 1.74 |
| 72 | 76 | 00388MtBB | V. Primary metabolism THIOREDOXIN M-TYPE CHLOROPLAST PRECURSOR PCR product | 1.3657e-10 | 10.40 | 1.73 |
| 73 | 109 | GVN-65K1 | leghemoglobin 1 PCR product | 0.00012409 | 14.43 | 1.73 |
| 74 | 80 | MtAe34 | Unknown function PCR product | 0.001385 | 12.16 | 1.71 |
| 75 | 83 | 10307MtBB | VII. Chromatin and DNA metabolism histone H2A PCR product | 2.8033e-09 | 11.31 | 1.70 |
| 76 | 71 | 50599MtBB | XIII. No homology PCR product | 1.7454e-05 | 11.11 | 1.67 |
| 77 | 79 | GVN-74I6 | MtN21 PCR product | 1.1678e-08 | 10.87 | 1.67 |
| 78 | 59 | 00198MtBB | V. Primary metabolism THIOREDOXIN M-TYPE CHLOROPLAST PRECURSOR PCR product | 1.4948e-05 | 10.53 | 1.64 |
| 79 | 87 | 93045MtBA | V. Primary metabolism ALBUMIN PCR product | 6.7711e-06 | 11.48 | 1.64 |
| 80 | 96 | 10073MtBB | XII.C. Unknown function PCR product | 2.461e-06 | 11.17 | 1.63 |
| 81 | 82 | 93406MtBB | I. Cell Wall PROFUCOSIDASE PRECURSOR PCR product | 0.0017885 | 10.73 | 1.62 |
| 82 | 48 | 91055MtBB | XIII. No homology PCR product | 1.6854e-05 | 9.93 | 1.61 |
| 83 | 88 | 91157MtBB | XIII. No homology PCR product | 0.00095042 | 10.21 | 1.60 |
| 84 | 81 | 10203MtBB | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 7.8627e-07 | 11.21 | 1.60 |
| 85 | 101 | JVCPG56 | MtN29 chimeric PCR product | 3.3877e-08 | 13.12 | 1.59 |
| 86 | 118 | 90689MtBB | III. Membrane transport CATIONIC AMINO ACID TRANSPORTER PCR product | 1 | 9.79 | 1.56 |
| 87 | 120 | 10107MtBB | IX. Protein synthesis and processing PROTEASOME SUBUNIT BETA TYPE PCR product | 3.4723e-07 | 13.70 | 1.52 |
| 88 | 90 | GVN-72E9 | nodulin 26 PCR product | 7.8888e-05 | 10.93 | 1.50 |
| 89 | 122 | MsPEPC | PEP carboxylase PCR product | 0.00052915 | 11.20 | 1.47 |
| 90 | 97 | 20267MtBB | XII.C. Unknown function [NODULIN] MTN26 PCR product | 2.4858e-06 | 10.12 | 1.46 |
| 91 | 113 | 90776MtBB | XIII. No homology PCR product | 0.0036777 | 11.80 | 1.45 |
| 92 | 93 | 00235MtBB | XII.B. Abiotic stimuli and development OSMOTIN/THAUMATIN-LIKE PROTEIN PRECURSOR PCR product | 1.8093e-05 | 11.26 | 1.44 |
| 93 | 77 | 00546MtBB | XII.C. Unknown function PCR product | 1.5567e-10 | 10.74 | 1.38 |
| 94 | 103 | 45591MtBB | VI. Secondary metabolism and hormone metabolism ISOFLAVONE-O-METHYTRANSFERASE PCR product | 4.329e-09 | 11.00 | 1.38 |
| 95 | 78 | 90141MtBA | VI. Secondary metabolism and hormone metabolism GLUCOSYLTRANSFERASE PCR product | 2.4336e-06 | 10.36 | 1.36 |

Tabelle C.2: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 96 | 115 | JVCPG43 | MtN16 PCR product | 6.0227e-10 | 11.72 | 1.36 |
| 97 | 126 | 40035MtBA | I. Cell Wall CELL WALL INVERTASE BETA-FRUCTOFURANOSIDASE PCR product | 1.1022e-05 | 10.73 | 1.35 |
| 98 | 94 | 40082MtBB | XII.A. Defense and cell rescue ENDO-13-BETA-GLUCOSIDASE PRECURSOR PCR product | 0.032502 | 10.52 | 1.32 |
| 99 | 129 | 91045MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 0.00069503 | 13.00 | 1.30 |
| 100 | 125 | 10443MtBB | XII.C. Unknown function PCR product | 0.005478 | 9.95 | 1.30 |
| 101 | 95 | 20129MtBB | IV. Vesicular trafficking secretion and protein sorting ANNEXIN PCR product | 4.4514e-06 | 11.12 | 1.30 |
| 102 | 116 | 50454MtBB | VI. Secondary metabolism and hormone metabolism TERPENE SYNTHASE PCR product | 0.0038171 | 10.06 | 1.29 |
| 103 | 102 | 30105MtBB | V. Primary metabolism BETA-AMYLASE PCR product | 0.00020082 | 10.76 | 1.29 |
| 104 | 154 | 50559MtBB | X. Signal transduction and post-translational regulation SER/THR PROTEIN KINASE PCR product | 6.3843e-05 | 10.31 | 1.28 |
| 105 | 92 | 90695MtBB | XIII. No homology PCR product | 0.0009932 | 10.84 | 1.28 |
| 106 | 68 | 10968MtBB | VI. Secondary metabolism and hormone metabolism PROFUCOSIDASE PRECURSOR PCR product | 0.10787 | 10.41 | 1.27 |
| 107 | 133 | 90940MtBB | XII.C. Unknown function [NODULIN] MtN7 PCR product | 0.0044339 | 11.04 | 1.27 |
| 108 | 134 | 00093MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 0.00063902 | 12.01 | 1.24 |
| 109 | 84 | 30497MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 0.00060792 | 10.34 | 1.24 |
| 110 | 258 | HLG100 | HLG 1.0 well 100 PCR product | 0.040957 | 11.05 | 1.22 |
| 111 | 114 | 00643MtBB | XII.C. Unknown function PCR product | 6.6333e-05 | 10.38 | 1.19 |
| 112 | 132 | 00436MtBB | XIII. No homology PCR product | 0.029324 | 11.44 | 1.19 |
| 113 | 121 | JVCPG35 | MtN7 PCR product | 0.00033087 | 10.18 | 1.17 |
| 114 | 119 | 50550MtBB | V. Primary metabolism LIPASE PCR product | 0.00035311 | 10.97 | 1.17 |
| 115 | 123 | 30415MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE 23 KDA SUBUNIT PCR product | 0.00013134 | 11.38 | 1.16 |
| 116 | 124 | 30416MtBB | XII. Miscellaneous GLUCAN 14-ALPHA-GLUCOSIDASE PCR product | 0.0062331 | 11.72 | 1.16 |
| 117 | 146 | 10261MtBB | IX. Protein synthesis and processing UBIQUITIN-CONJUGATING ENZYME E2-17 KDA PCR product | 0.009233 | 13.08 | 1.16 |
| 118 | 137 | 10168MtBB | I. Cell Wall PECTIN METHYL-ESTERASE PCR product | 4.2405e-06 | 11.53 | 1.15 |
| 119 | 144 | 40161MtBB | X. Signal transduction and post-translational regulation PROTEIN KINASE DOMAIN CONTAINING PROTEIN PCR product | 7.288e-05 | 9.71 | 1.15 |
| 120 | 105 | 00290MtBB | XII.C. Unknown function PD155440 PCR product | 1 | 10.44 | 1.14 |
| 121 | 127 | 00383MtBB | XIII. No homology PCR product | 0.00050542 | 9.96 | 1.13 |
| 122 | 108 | 00515MtBB | XIII. No homology PCR product | 0.23353 | 10.13 | 1.12 |
| 123 | 143 | 30270MtBB | XII.C. Unknown function PCR product | 0.27817 | 13.03 | 1.12 |
| 124 | 112 | 00153MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 4.2881e-06 | 10.86 | 1.11 |
| 125 | 2 | GVN-55E6 | nodulin 25 PCR product | 1 | 9.79 | 1.11 |
| 126 | 130 | 00246MtBB | XII.C. Unknown function CYS RICH PROTEIN PCR product | 0.28838 | 10.45 | 1.11 |
| 127 | 162 | 10021MtBC | I. Cell Wall CELL WALL PROTEIN PCR product | 3.8828e-09 | 13.70 | 1.10 |
| 128 | 463 | 10853MtBB | VI. Secondary metabolism and hormone metabolism IRON/ASCORBATE-DEPENDENT OXIDOREDUCTASE PCR product | 0.05227 | 9.69 | 1.10 |
| 129 | 110 | 45429MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 1 | 10.57 | 1.09 |
| 130 | 86 | 00127MtBC | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 6.0031e-07 | 10.86 | 1.09 |
| 131 | 180 | 10472MtBB | XII.C. Unknown function SEVEN TRANSMEMBRANE DOMAIN PROTEIN PCR product | 1.0206e-07 | 12.93 | 1.06 |
| 132 | 131 | 30530MtBB | XIII. No homology PCR product | 0.0015354 | 10.82 | 1.06 |
| 133 | 156 | 20204MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 0.00010907 | 9.65 | 1.04 |
| 134 | 147 | 90971MtBB | XIII. No homology PCR product | 0.37859 | 10.59 | 1.02 |
| 135 | 186 | 91067MtBB | XIII. No homology PCR product | 5.6031e-10 | 13.48 | 1.01 |
| 136 | 107 | 00553MtBB | VIII. Gene expression and RNA metabolism TRANSCRIPTION FACTOR PCR product | 0.019654 | 10.32 | 1.00 |
| 137 | 175 | 91489MtBC | XIII. No homology PCR product | 1 | 14.40 | 1.00 |
| 138 | 165 | 10026MtBB | XII.C. Unknown function PCR product | 1 | 11.63 | 0.99 |
| 139 | 111 | 90927MtBB | XIII. No homology PCR product | 0.31315 | 10.01 | 0.98 |
| 140 | 166 | 10690MtBB | XII.A. Defense and cell rescue MTN13 PCR product | 0.76747 | 11.43 | 0.97 |
| 141 | 139 | 30515MtBB | XIII. No homology PCR product | 1.452e-07 | 10.98 | 0.96 |
| 142 | 160 | 45014MtBA | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L32 RP49 PCR product | 0.3088 | 11.90 | 0.96 |
| 143 | 150 | 00457MtBB | XIII. No homology PCR product | 0.029055 | 10.63 | 0.96 |

Tabelle C.2: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 144 | 158 | GVSN-9F12 | nodulin 75 PCR product | 0.042855 | 15.06 | 0.95 |
| 145 | 155 | 00617MtBB | XII.C. Unknown function PCR product | 1 | 12.28 | 0.95 |
| 146 | 136 | JVCPG55 | MtN28 PCR product | 1 | 9.61 | 0.93 |
| 147 | 164 | 00331MtBC | V. Primary metabolism ACID PHOSPHATASE PCR product | 0.00019008 | 11.51 | 0.93 |
| 148 | 195 | 50083MtBA | XII.C. Unknown function PCR product | 0.00051901 | 10.67 | 0.93 |
| 149 | 1287 | 00441MtBB | XII.C. Unknown function PCR product | 1 | 10.59 | 0.91 |
| 150 | 142 | 90901MtBB | VIII. Gene expression and RNA metabolism RESPONSE REGULATOR PCR product | 0.022842 | 10.40 | 0.91 |
| 151 | 98 | 91023MtBB | XII.C. Unknown function [NODULIN] PCR product | 0.45283 | 10.41 | 0.89 |
| 152 | 176 | 90857MtBB | III. Membrane transport TRANSPORTER PCR product | 0.0029278 | 11.84 | 0.88 |
| 153 | 181 | 40187MtBB | XIII. No homology PCR product | 0.00010249 | 10.37 | 0.88 |
| 154 | 179 | 20062MtBB | V. Primary metabolism L-ASPARAGINASE PCR product | 0.001336 | 13.13 | 0.87 |
| 155 | 153 | GVSN-24A24 | MtN13 PCR product | 2.6906e-08 | 13.45 | 0.86 |
| 156 | 172 | MtGSa | glutamine synthetase a PCR product | 0.0012205 | 11.00 | 0.85 |
| 157 | 167 | 90870MtBB | XIII. No homology PCR product | 0.020098 | 9.77 | 0.85 |
| 158 | 173 | 00203MtBB | XII.C. Unknown function PCR product | 1 | 11.66 | 0.85 |
| 159 | 184 | 93419MtBB | VI. Secondary metabolism and hormone metabolism CYTOCHROME P450 PCR product | 1 | 15.11 | 0.84 |
| 160 | 211 | 91319MtBC | XIII. No homology PCR product | 1 | 11.46 | 0.82 |
| 161 | 1096 | MtAe5 | Unknown function PCR product | 1 | 11.49 | 0.82 |
| 162 | 275 | 30579MtBB | XII.C. Unknown function MEMBRANE PROTEIN PCR product | 0.0010101 | 10.52 | 0.82 |
| 163 | 203 | HLG200 | HLG 1.0 well 200 PCR product | 9.0669e-05 | 13.01 | 0.81 |
| 164 | 169 | 10403MtBB | IX. Protein synthesis and processing PROTEIN DISULFIDE ISOMERASE PRECURSOR PCR product | 0.84062 | 12.93 | 0.81 |
| 165 | 148 | 50719MtBC | XIII. No homology PCR product | 1 | 11.92 | 0.81 |
| 166 | 193 | 10358MtBB | XII.C. Unknown function [NODULIN] PCR product | 0.41561 | 12.17 | 0.80 |
| 167 | 89 | 90621MtBB | XIII. No homology PCR product | 1 | 10.33 | 0.80 |
| 168 | 379 | 90890MtBB | VI. Secondary metabolism and hormone metabolism LIPOXYGENASE PCR product | 0.68489 | 11.01 | 0.80 |
| 169 | 1085 | 40098MtBA | XII.C. Unknown function AP2 DOMAIN CONTAINING PROTEIN PCR product | 1 | 10.04 | 0.79 |
| 170 | 215 | 10733MtBB | XIII. No homology PCR product | 0.056746 | 12.32 | 0.79 |
| 171 | 135 | 00578MtBB | V. Primary metabolism [NODULIN] ADENINE PHOSPHORIBOSYLTRANSFERASE APRT PCR product | 0.118 | 10.14 | 0.79 |
| 172 | 201 | MtSucS1_XbN61 | nodule-enhanced sucrose synthase MtSucS1 PCR product | 0.20578 | 11.79 | 0.78 |
| 173 | 260 | MtSucS2 | second sucrose synthase MtSucS2, MtBC02H08 PCR product | 0.043967 | 10.50 | 0.78 |
| 174 | 194 | 91038MtBB | XII.C. Unknown function PCR product | 0.096181 | 10.79 | 0.77 |
| 175 | 138 | 20111MtBA | V. Primary metabolism GLUTAMATE DEHYDROGENASE PCR product | 0.00021518 | 10.60 | 0.77 |
| 176 | 140 | JVCPG41 | MtN14 PCR product | 0.0042415 | 10.21 | 0.76 |
| 177 | 213 | 90648MtBB | XIII. No homology PCR product | 1 | 12.12 | 0.76 |
| 178 | 2204 | mtgmabc120001a04 | putative chloride ion channel protein, TC10381-homolog, 89% zu VfCLIC1 PCR product | 1 | 9.74 | 0.75 |
| 179 | 234 | KV0-1A24 | chlorophyll a/b binding protein PCR product | 1 | 10.21 | 0.74 |
| 180 | 1456 | 90176MtBA | XIII. No homology PCR product | 1 | 9.78 | 0.74 |
| 181 | 217 | 20373MtBB | I. Cell Wall ENDO-BETA-14-GLUCANASE PCR product | 0.20034 | 9.74 | 0.73 |
| 182 | 305 | 00004MtBB | I. Cell Wall [NODULIN] MtN12 Prolin-rich protein PCR product | 0.0023984 | 14.34 | 0.73 |
| 183 | 159 | KV3-21B21 | MtN21-like PCR product | 1 | 10.61 | 0.73 |
| 184 | 607 | 91056MtBB | XIII. No homology PCR product | 0.80615 | 10.70 | 0.73 |
| 185 | 1220 | 30526MtBB | VI. Secondary metabolism and hormone metabolism AMINE OXIDASE [COPPER-CONTAINING] PCR product | 1 | 9.71 | 0.73 |
| 186 | 152 | 20102MtBC | V. Primary metabolism L-ASCORBATE OXIDASE PRECURSOR PCR product | 0.0024857 | 10.51 | 0.72 |
| 187 | 373 | 00340MtBB | XII.C. Unknown function PD155440 PCR product | 3.6166e-06 | 14.21 | 0.70 |
| 188 | 182 | 30198MtBB | II. Cytoskeleton TUBULIN BETA CHAIN PCR product | 0.19511 | 10.65 | 0.70 |
| 189 | 264 | 20187MtBB | VIII. Gene expression and RNA metabolism MADS TRANSCRIPTIONAL FACTOR PCR product | 1 | 10.04 | 0.70 |
| 190 | 231 | 93062MtBA | V. Primary metabolism NADH-PLASTOQUINONE OXIDOREDUCTASE CHAIN 1 PCR product | 0.026173 | 11.56 | 0.70 |
| 191 | 238 | 90604MtBB | XII.C. Unknown function PCR product | 0.0837 | 13.51 | 0.69 |
| 192 | 329 | 91072MtBB | XIII. No homology PCR product | 0.0021423 | 13.57 | 0.69 |
| 193 | 593 | 90866MtBB | XII.C. Unknown function PCR product | 0.34897 | 9.79 | 0.69 |

Tabelle C.2: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 194 | 252 | 20218MtBB | IV. Vesicular trafficking secretion and protein sorting ANNEXIN PCR product | 0.00083759 | 12.69 | 0.69 |
| 195 | 149 | 10607MtBC | I. Cell Wall CAFFEIC ACID O-METHYLTRANSFERASE PCR product | 0.032362 | 9.87 | 0.69 |
| 196 | 3048 | 90744MtBB | XIII. No homology PCR product | 1 | 9.60 | 0.68 |
| 197 | 145 | 93135MtBB | XII.C. Unknown function PCR product | 1 | 10.96 | 0.68 |
| 198 | 157 | 30343MtBB | V. Primary metabolism ESTERASE PCR product | 1 | 10.09 | 0.67 |
| 199 | 1944 | 90298MtBA | XIII. No homology PCR product | 1 | 10.18 | 0.67 |
| 200 | 268 | MtAMLs32 | Putative wound induced protein PCR product | 1 | 13.67 | 0.66 |
| 201 | 401 | 00042MtBC | V. Primary metabolism SUCROSE SYNTHASE PCR product | 0.21489 | 11.94 | 0.66 |
| 202 | 265 | 30373MtBB | XII.C. Unknown function PCR product | 0.001739 | 12.44 | 0.66 |
| 203 | 198 | 90751MtBB | XIII. No homology PCR product | 0.29927 | 10.43 | 0.66 |
| 204 | 310 | 91049MtBB | III. Membrane transport [PUTATIVE NODULIN] SYMBIOTIC AMMONIUM TRANSPORTER PCR product | 0.095449 | 10.39 | 0.66 |
| 205 | 263 | 50652MtBC | XIII. No homology PCR product | 1 | 12.38 | 0.66 |
| 206 | 374 | 10816MtBB | XII. Miscellaneous PEROXIDASE PCR product | 1 | 10.48 | 0.65 |
| 207 | 183 | 00187MtBB | V. Primary metabolism CELL WALL INVERTASE BETA-FRUCTOFURANOSIDASE PCR product | 0.012983 | 9.96 | 0.65 |
| 208 | 161 | 10312MtBC | XII.A. Defense and cell rescue CHITINASE PCR product | 0.00089566 | 10.26 | 0.65 |
| 209 | 1306 | 00006MtBB | XII.C. Unknown function TRANSLATIONALLY CONTROLLED TUMOR PROTEIN SIGNATURE CONTAINING PROTEIN PCR product | 1 | 9.96 | 0.64 |
| 210 | 413 | MtAe88 | Unknown function PCR product | 0.59181 | 14.47 | 0.64 |
| 211 | 210 | 91662MtBC | XIII. No homology PCR product | 1 | 12.14 | 0.64 |
| 212 | 244 | 10310MtBB | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 1 | 11.31 | 0.64 |
| 213 | 249 | 10071MtBC | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S3A PCR product | 1 | 11.42 | 0.64 |
| 214 | 208 | 00740MtBB | III. Membrane transport MEMBRANE TRANSPORTER PCR product | 1 | 11.06 | 0.64 |
| 215 | 772 | 93396MtBB | XIII. No homology PCR product | 0.49466 | 9.98 | 0.64 |
| 216 | 209 | 90856MtBB | XII.C. Unknown function PD025544 PCR product | 0.00022675 | 10.51 | 0.64 |
| 217 | 247 | 20302MtBC | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.00017334 | 12.68 | 0.63 |
| 218 | 319 | 10146MtBB | XII. Miscellaneous LIPOXYGENASE PCR product | 0.050122 | 12.16 | 0.63 |
| 219 | 1427 | 30473MtBA | XII.C. Unknown function PCR product | 1 | 10.34 | 0.63 |
| 220 | 3171 | 50540MtBB | VIII. Gene expression and RNA metabolism N2N2-DIMETHYLGUANOSINE TRNA METHYLTRANSFERASE PRECURSOR PCR product | 1 | 9.69 | 0.63 |
| 221 | 72 | 40162MtBB | XII.C. Unknown function TRANSMEMBRANE MtN3 PCR product | 1 | 9.83 | 0.63 |
| 222 | 239 | 90545MtBA | XIII. No homology PCR product | 1 | 12.20 | 0.62 |
| 223 | 298 | 90892MtBB | III. Membrane transport PEPTIDE TRANSPORTER PCR product | 1 | 12.71 | 0.62 |
| 224 | 885 | 10574MtBA | XII.C. Unknown function PCR product | 1 | 10.12 | 0.62 |
| 225 | 295 | 10806MtBB | XII.C. Unknown function PTS HPR COMPONENT SERINE PHOSPHORYLATION SITE CONTAINING PROTEIN PCR product | 0.12518 | 10.28 | 0.61 |
| 226 | 197 | 91998MtBC | XIII. No homology PCR product | 0.15264 | 11.00 | 0.61 |
| 227 | 317 | 50203MtBA | VI. Secondary metabolism and hormone metabolism CYTOCHROME P450 PCR product | 1 | 10.13 | 0.61 |
| 228 | 206 | 00437MtBB | XII. Miscellaneous NON-SPECIFIC LIPID TRANSFER-LIKE PROTEIN PCR product | 0.00031613 | 11.38 | 0.61 |
| 229 | 257 | 90844MtBB | IV. Vesicular trafficking secretion and protein sorting TSNARE/SYNTAXIN PCR product | 1 | 11.30 | 0.61 |
| 230 | 192 | 00490MtBB | V. Primary metabolism LIPASE (CLASS 3) DOMAIN CONTAINING PROTEIN PCR product | 1 | 11.15 | 0.61 |
| 231 | 544 | 90819MtBB | IX. Protein synthesis and processing UBIQUITIN PCR product | 0.46867 | 13.75 | 0.61 |
| 232 | 177 | 90944MtBB | XII.C. Unknown function [PUTATIVE NODULIN] GLYCINE-RICH PROTEIN PRECURSOR PCR product | 0.014242 | 9.62 | 0.61 |
| 233 | 2427 | 11004MtBB | XII.C. Unknown function MTN3-LIKE PROTEIN PCR product | 1 | 10.00 | 0.60 |
| 234 | 212 | 00328MtBB | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L11 PCR product | 0.11527 | 11.20 | 0.60 |
| 235 | 1742 | 90874MtBB | XII.C. Unknown function PCR product | 1 | 10.18 | 0.60 |
| 236 | 524 | 20106MtBA | V. Primary metabolism GLUCOSE-1-PHOSPHATE ADENYLYLTRANSFERASE LARGE SUBUNIT PRECURSOR PCR product | 0.046494 | 9.89 | 0.60 |
| 237 | 775 | 93341MtBB | III. Membrane transport PEPTIDE TRANSPORTER / NITRITE TRANSPORTER PCR product | 0.16019 | 10.08 | 0.60 |
| 238 | 350 | 50644MtBB | IX. Protein synthesis and processing GLUCOSAMINE–FRUCTOSE-6-PHOSPHATE AMINOTRANSFERASE ISOMERIZING 2 PCR product | 0.0012373 | 12.15 | 0.60 |

Tabelle C.2: Fortsetzung der M-Liste des Experiments Nod 4, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 239 | 204 | 45417MtBB | XIII. No homology PCR product | 0.00011046 | 11.18 | 0.60 |
| 240 | 642 | 50408MtBB | XII.C. Unknown function APETALA2 (AP2) DOMAIN CONTAINING PROTEIN PCR product | 0.045539 | 14.50 | 0.59 |
| 241 | 366 | 30494MtBA | XII.C. Unknown function GLU/LYS RICH PROTEIN / SER RICH PROTEIN PCR product | 1 | 12.00 | 0.59 |
| 242 | 262 | 20213MtBB | V. Primary metabolism TREHALOSE-PHOSPHATASE PCR product | 0.012765 | 10.15 | 0.59 |
| 243 | 243 | MtAMLs642 | Unknown PCR product | 1 | 14.37 | 0.59 |
| 244 | 472 | MtAMLs634 | Unknown PCR product | 1 | 13.89 | 0.59 |
| 245 | 2265 | 90774MtBB | IX. Protein synthesis and processing 60S RIBOSOMAL PROTEIN L18 PCR product | 1 | 9.84 | 0.59 |
| 246 | 271 | 00128MtBB | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S5 PCR product | 1 | 11.10 | 0.59 |
| 247 | 629 | 91006MtBB | XIII. No homology PCR product | 1 | 9.71 | 0.59 |
| 248 | 227 | 90830MtBB | XIII. No homology PCR product | 1 | 12.80 | 0.59 |
| 249 | 174 | 93357MtBB | XIII. No homology PCR product | 0.78144 | 10.45 | 0.59 |
| 250 | 188 | 00024MtBB | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S11 PCR product | 0.66436 | 10.74 | 0.59 |

Tabelle C.3: M-sortierte Liste der Sondensequenzen auf dem Mt6kRIT-Mikroarray des Nodulationsexperimentes Nod 10, bei Bildauswertung mit der Software Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 1 | 2 | mtgmabc120001a02 | leghemoglobin, TC6726-homolog, 70% zu VfLb29 PCR product | 1.3433e-08 | 10.66 | 8.20 |
| 2 | 4 | 10695MtBB | XIII. No homology PCR product | 7.3046e-08 | 10.30 | 7.99 |
| 3 | 1 | Globin | Globin PCR product | 8.9986e-08 | 10.38 | 7.98 |
| 4 | 5 | mt–aac120001a02 | leghemoglobin, TC13379-homolog, 73% zu VfLb29 PCR product | 7.3644e-09 | 10.82 | 7.80 |
| 5 | 3 | JVCPG49 | MtN22 PCR product | 4.6828e-07 | 10.91 | 7.50 |
| 6 | 7 | GFP | Green Fluorescent Protein PCR product | 2.6689e-09 | 9.87 | 7.48 |
| 7 | 6 | mt–aac120001a01 | leghemoglobin, TC10534-homolog, 77% zu VfLb29 PCR product | 5.2555e-09 | 10.53 | 7.33 |
| 8 | 10 | 10497MtBB | XIII. No homology PCR product | 7.3476e-09 | 10.23 | 7.12 |
| 9 | 8 | 00583MtBB | V. Primary metabolism [NODULIN] LEGHEMOGLOBIN 1 PCR product | 2.5408e-07 | 11.66 | 6.99 |
| 10 | 9 | GVN-72A6 | nodulin 25 PCR product | 2.8567e-07 | 11.46 | 6.63 |
| 11 | 12 | GVN-55E6 | nodulin 25 PCR product | 3.803e-08 | 12.07 | 5.74 |
| 12 | 11 | GVN-71A1 | nodulin 26 PCR product | 1.9661e-05 | 10.83 | 5.71 |
| 13 | 21 | 93119MtBB | XIII. No homology PCR product | 1.7594e-06 | 9.60 | 5.50 |
| 14 | 13 | MtAe27 | Unknown function PCR product | 4.8436e-07 | 10.33 | 5.44 |
| 15 | 14 | GVN-64D16 | enod8 PCR product | 8.5078e-05 | 11.96 | 5.36 |
| 16 | 24 | 10430MtBC | III. Membrane transport MAJOR INTRINSIC PROTEIN (NODULIN26-LIKE ) PCR product | 1.6168e-06 | 9.85 | 5.25 |
| 17 | 18 | MtAe34 | Unknown function PCR product | 5.8725e-05 | 11.62 | 4.85 |
| 18 | 16 | GVN-72E9 | nodulin 26 PCR product | 1.5509e-05 | 9.68 | 4.80 |
| 19 | 15 | mt–aac120001a03 | leghemoglobin, TC3089-homolog, 70% zu VfLb29 PCR product | 9.8347e-12 | 12.40 | 4.61 |
| 20 | 19 | 00646MtBB | XIII. No homology PCR product | 1.4427e-06 | 11.10 | 4.52 |
| 21 | 20 | 10766MtBB | XIII. No homology PCR product | 9.3739e-10 | 9.53 | 4.52 |
| 22 | 17 | 00281MtBB | XII.C. Unknown function [NODULIN] MTN22 PCR product | 6.1085e-11 | 12.69 | 4.27 |
| 23 | 28 | 00156MtBB | V. Primary metabolism [NODULIN] CARBONIC ANHYDRASE PCR product | 1.3323e-07 | 9.55 | 4.26 |
| 24 | 22 | 00800MtBC | V. Primary metabolism ASPARAGINE SYNTHETASE [GLUTAMINE-HYDROLYZING] PCR product | 3.6491e-08 | 10.58 | 4.06 |
| 25 | 37 | 00344MtBB | XII. Miscellaneous lectin PCR product | 2.9648e-08 | 8.99 | 3.97 |
| 26 | 35 | 45429MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 1.0315e-07 | 9.92 | 3.88 |
| 27 | 36 | MtGSa | glutamine synthetase a PCR product | 2.3664e-05 | 10.09 | 3.77 |
| 28 | 27 | 10685MtBB | XIII. No homology PCR product | 8.7845e-05 | 11.09 | 3.76 |
| 29 | 32 | 20204MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 3.7779e-07 | 8.89 | 3.70 |
| 30 | 39 | mtgmabc120001a04 | putative chloride ion channel protein, TC10381-homolog, 89% zu VfCLIC1 PCR product | 7.5094e-08 | 8.78 | 3.66 |
| 31 | 30 | 30105MtBB | V. Primary metabolism BETA-AMYLASE PCR product | 9.6107e-07 | 9.49 | 3.53 |
| 32 | 26 | 10811MtBB | X. Signal transduction REMORIN PCR product | 3.3753e-06 | 8.87 | 3.46 |
| 33 | 38 | 10058MtBA | XII.C. Unknown function PD001144 PCR product | 2.2242e-07 | 9.48 | 3.43 |
| 34 | 29 | MsCA | carbonic anhydrase PCR product | 4.8163e-06 | 10.99 | 3.41 |
| 35 | 42 | JVCPG44 | MtN17 PCR product | 8.232e-06 | 9.57 | 3.41 |
| 36 | 23 | 90927MtBB | XIII. No homology PCR product | 5.3008e-06 | 9.02 | 3.36 |
| 37 | 49 | 90646MtBB | XIII. No homology PCR product | 1.0803e-07 | 9.26 | 3.25 |
| 38 | 44 | 00120MtBB | XII.C. Unknown function [NODULIN] MTN29 PCR product | 2.7293e-06 | 8.79 | 3.20 |
| 39 | 31 | HLG100 | HLG 1.0 well 100 PCR product | 0.00018958 | 9.72 | 3.19 |
| 40 | 53 | JVCPG58 | ENOD40 PCR product | 1.9389e-05 | 10.07 | 3.17 |
| 41 | 54 | JVCPG42 | MtN15 PCR product | 1.2481e-05 | 9.83 | 3.14 |
| 42 | 33 | JVCPG51 | MtN24 PCR product | 6.313e-07 | 8.96 | 3.10 |
| 43 | 46 | 40167MtBB | XII.C. Unknown function MtN9-LIKE PROTEIN PCR product | 9.0203e-07 | 8.59 | 3.08 |
| 44 | 57 | JVCPG53 | MtN26 PCR product | 1.0625e-05 | 9.23 | 3.05 |
| 45 | 34 | GVN-70I6 | enod40 PCR product | 1.5778e-08 | 11.17 | 3.03 |

Tabelle C.3: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 46 | 56 | KV3-21B21 | MtN21-like PCR product | 3.2327e-05 | 10.20 | 3.01 |
| 47 | 63 | 00746MtBB | XII.C. Unknown function [NODULIN] MTN11 PCR product | 1.7377e-06 | 8.78 | 2.99 |
| 48 | 58 | 00186MtBB | XII.C. Unknown function [NODULIN] MTN16 PCR product | 0.026835 | 10.91 | 2.98 |
| 49 | 51 | 00203MtBB | XII.C. Unknown function PCR product | 0.0014309 | 10.95 | 2.96 |
| 50 | 43 | 00043MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 3.0682e-10 | 10.80 | 2.94 |
| 51 | 40 | JVCPG27 | ENOD16 PCR product | 1.8822e-08 | 9.70 | 2.88 |
| 52 | 75 | JVCPG29 | MtN1 PCR product | 1.6934e-05 | 9.91 | 2.86 |
| 53 | 52 | JVCPG48 | MtN21 PCR product | 5.1979e-07 | 8.78 | 2.77 |
| 54 | 4442 | 30554MtBC | V. Primary metabolism DEHYDROGENASE/REDUCTASE PCR product | 0 | 9.09 | 2.63 |
| 55 | 3158 | 93120MtBB | XII.C. Unknown function PCR product | 0 | 7.46 | 2.62 |
| 56 | 45 | JVCPG25 | ENOD11 PCR product | 0.00030131 | 12.44 | 2.51 |
| 57 | 41 | MtAMLs296 | DNA binding protein homolog PCR product | 5.7698e-06 | 8.71 | 2.50 |
| 58 | 64 | 20131MtBA | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 1.9086e-05 | 9.50 | 2.43 |
| 59 | 50 | 90870MtBB | XIII. No homology PCR product | 1.9217e-05 | 8.68 | 2.40 |
| 60 | 59 | 90571MtBB | XII.C. Unknown function [NODULIN] MtN19 PCR product | 0.00026255 | 8.97 | 2.40 |
| 61 | 70 | GVN-74J10 | MtN21-like PCR product | 0.0075337 | 9.84 | 2.39 |
| 62 | 48 | JVCPG37 | MtN10 PCR product | 5.427e-06 | 13.13 | 2.37 |
| 63 | 79 | 00068MtBB | XII.C. Unknown function [NODULIN] MTN1 PRECURSOR PCR product | 9.6215e-05 | 9.30 | 2.36 |
| 64 | 47 | GVN-65K1 | leghemoglobin 1 PCR product | 1.9257e-10 | 13.45 | 2.34 |
| 65 | 92 | 20267MtBB | XII.C. Unknown function [NODULIN] MTN26 PCR product | 9.85e-06 | 8.32 | 2.34 |
| 66 | 122 | 91668MtBC | XIII. No homology PCR product | 1 | 10.64 | 2.33 |
| 67 | 78 | 00651MtBC | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 0.012727 | 9.30 | 2.28 |
| 68 | 68 | 10073MtBB | XII.C. Unknown function PCR product | 9.8917e-06 | 10.74 | 2.27 |
| 69 | 25 | GVN-55B13 | MtN1 PCR product | 0.00015894 | 8.67 | 2.27 |
| 70 | 72 | 45203MtBA | V. Primary metabolism CTP SYNTHASE PCR product | 0.0087743 | 8.84 | 2.25 |
| 71 | 61 | 91525MtBC | XII.C. Unknown function PCR product | 2.251e-05 | 9.34 | 2.23 |
| 72 | 65 | 00060MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 2.4242e-05 | 10.52 | 2.21 |
| 73 | 62 | 00240MtBB | XII.C. Unknown function [NODULIN] EARLY NODULIN 12 PRECURSOR PCR product | 4.0776e-07 | 13.03 | 2.21 |
| 74 | 110 | 10690MtBB | XII.A. Defense and cell rescue MTN13 PCR product | 1 | 10.35 | 2.18 |
| 75 | 221 | 91090MtBB | V. Primary metabolism 4-HYDROXYPHENYLPYRUVATE DIOXYGENASE PCR product | 1 | 9.35 | 2.14 |
| 76 | 80 | 91023MtBB | XII.C. Unknown function [NODULIN] PCR product | 1.167e-05 | 9.38 | 2.13 |
| 77 | 76 | 10203MtBB | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.00033592 | 9.24 | 2.13 |
| 78 | 201 | 93226MtBB | II. Cytoskeleton F-ACTIN CAPPING PROTEIN ALPHA SUBUNIT PCR product | 1 | 10.47 | 2.12 |
| 79 | 94 | 91471MtBC | IX. Protein synthesis and processing CYSTEINE PROTEINASE PCR product | 0.01187 | 9.62 | 2.10 |
| 80 | 143 | 91628MtBC | VIII. Gene expression and RNA metabolism SPLICING FACTOR PCR product | 1 | 11.28 | 2.09 |
| 81 | 140 | 10850MtBC | XII.C. Unknown function PLASTOGLOBULE ASSOCIATED PROTEIN PCR product | 1 | 10.64 | 2.09 |
| 82 | 84 | JVCPG33 | MtN5 PCR product | 0.044837 | 10.99 | 2.06 |
| 83 | 74 | MtSucS2 | second sucrose synthase MtSucS2, MtBC02H08 PCR product | 2.3066e-07 | 9.82 | 2.03 |
| 84 | 60 | MHAM-47P10 | MtN93 PCR product | 8.981e-08 | 9.03 | 2.01 |
| 85 | 193 | 91485MtBC | IX. Protein synthesis and processing RIBOSOMAL PROTEIN PCR product | 1 | 11.57 | 1.99 |
| 86 | 136 | 91127MtBB | XIII. No homology PCR product | 1 | 9.55 | 1.99 |
| 87 | 86 | 90689MtBB | III. Membrane transport CATIONIC AMINO ACID TRANSPORTER PCR product | 0.0071155 | 8.36 | 1.99 |
| 88 | 88 | 92225MtBC | XIII. No homology PCR product | 1 | 11.39 | 1.97 |
| 89 | 66 | JVCPG1 | glutamine synthetase MtGSIa PCR product | 2.128e-09 | 11.70 | 1.95 |
| 90 | 1279 | 91585MtBC | XIII. No homology PCR product | 1 | 8.42 | 1.95 |
| 91 | 257 | 30475MtBB | IX. Protein synthesis and processing PROLYL 4-HYDROXYLASE ALPHA SUBUNIT PRECURSOR PCR product | 1 | 11.30 | 1.93 |
| 92 | 204 | 50501MtBB | XII.C. Unknown function PCR product | 1 | 9.51 | 1.93 |
| 93 | 95 | 00290MtBB | XII.C. Unknown function PD155440 PCR product | 0.00021058 | 8.88 | 1.93 |
| 94 | 97 | 30404MtBC | XII.C. Unknown function PCR product | 0.41594 | 9.56 | 1.92 |
| 95 | 99 | 45259MtBB | V. Primary metabolism 34-DIHYDROXY-2-BUTANONE-4-PHOSPHATE SYNTHASE PCR product | 0.023267 | 9.92 | 1.90 |

Tabelle C.3: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 96 | 85 | GVN-7416 | MtN21 PCR product | 0.00061014 | 8.62 | 1.88 |
| 97 | 90 | 10358MtBB | XII.C. Unknown function [NODULIN] PCR product | 7.2388e-05 | 12.02 | 1.87 |
| 98 | 73 | 20114MtBA | V. Primary metabolism GLUTAMINE SYNTHETASE PCR product | 6.6502e-08 | 12.30 | 1.85 |
| 99 | 71 | 00515MtBB | XIII. No homology PCR product | 0.00026037 | 8.79 | 1.83 |
| 100 | 77 | 91942MtBC | IX. Protein synthesis and processing TRANSAMIDASE PCR product | 1 | 9.22 | 1.82 |
| 101 | 114 | 91074MtBB | XIII. No homology PCR product | 0.00087607 | 9.02 | 1.82 |
| 102 | 103 | 90782MtBB | IX. Protein synthesis and processing [PROTEIN-PII] URIDYLYLTRANSFERASE PCR product | 0.00010637 | 8.75 | 1.81 |
| 103 | 125 | 00643MtBB | XII.C. Unknown function PCR product | 0.0010977 | 9.14 | 1.76 |
| 104 | 115 | 20258MtBC | X. Signal transduction and post-translational regulation PROTEIN KINASE PCR product | 1 | 9.69 | 1.76 |
| 105 | 82 | 10140MtBA | XII.B. Abiotic stimuli and development AUXIN INDUCED PROTEIN LIKE-PROTEIN PCR product | 3.1535e-06 | 9.73 | 1.74 |
| 106 | 107 | JVCPG52 | MtN25 PCR product | 0.012647 | 10.59 | 1.74 |
| 107 | 121 | 93260MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 0.0017264 | 9.54 | 1.74 |
| 108 | 150 | 90508MtBA | XII.C. Unknown function CONSERVED PROTEIN PCR product | 1 | 7.89 | 1.73 |
| 109 | 55 | 90408MtBA | XII.C. Unknown function PCR product | 0.056984 | 8.32 | 1.73 |
| 110 | 4206 | 93429MtBA | III. Membrane transport TRANSMEMBRANE TRANSPORT PROTEIN PCR product | 0 | 7.45 | 1.72 |
| 111 | 105 | JVCPG36 | MtN9 PCR product | 0.0018815 | 8.39 | 1.72 |
| 112 | 96 | 00372MtBC | XIII. No homology PCR product | 0.084156 | 10.99 | 1.71 |
| 113 | 81 | 00182MtBC | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 0.0027313 | 11.69 | 1.71 |
| 114 | 190 | 10495MtBB | X. Signal transduction and post-translational regulation PROTEIN PHOSPHATASE-2C PCR product | 1 | 10.39 | 1.70 |
| 115 | 69 | 90910MtBB | XII.C. Unknown function [NODULIN] MTN20 PCR product | 0.00010553 | 8.95 | 1.69 |
| 116 | 100 | JVCPG56 | MtN29 chimeric PCR product | 0.0047483 | 12.09 | 1.69 |
| 117 | 147 | 10008MtBC | V. Primary metabolism PROTEIN 2 PRECURSOR OEE2 23 KD SUBUNIT OF OXYGEN EVOLVING SYSTEM OF PHOTOSYSTEM II PCR product | 0.99032 | 10.12 | 1.68 |
| 118 | 253 | 40172MtBC | IX. Protein synthesis and processing EUKARYOTIC PEPTIDE CHAIN RELEASE FACTOR SUBUNIT 1 ERF1 PCR product | 1 | 11.10 | 1.68 |
| 119 | 231 | 11001MtBA | XII.C. Unknown function APETALA2 (AP2) DOMAIN CONTAINING PROTEIN PCR product | 1 | 10.67 | 1.67 |
| 120 | 223 | 91831MtBC | V. Primary metabolism OXIDOREDUCTASE PCR product | 1 | 12.08 | 1.66 |
| 121 | 216 | 00743MtBC | X. Signal transduction and post-translational regulation PHOSPHOLIPASE D PCR product | 1 | 9.49 | 1.63 |
| 122 | 67 | JVCPG55 | MtN28 PCR product | 0.010749 | 7.90 | 1.63 |
| 123 | 145 | 91397MtBC | XII.C. Unknown function PCR product | 1 | 9.78 | 1.62 |
| 124 | 2401 | 92040MtBC | V. Primary metabolism ACYL-COA-BINDING PROTEIN PCR product | 0 | 8.97 | 1.60 |
| 125 | 262 | 90690MtBB | XIII. No homology PCR product | 1 | 11.05 | 1.60 |
| 126 | 389 | 91746MtBC | IX. Protein synthesis and processing ASPARTIC PROTEINASE PCR product | 1 | 8.78 | 1.60 |
| 127 | 91 | HLG200 | HLG 1.0 well 200 PCR product | 0.00057331 | 12.70 | 1.60 |
| 128 | 202 | 91827MtBC | X. Signal transduction and post-translational regulation PHOSPHOLIPASE A2 PRECURSOR PCR product | 1 | 9.97 | 1.60 |
| 129 | 228 | 30538MtBC | XIII. No homology TRANSMEMBRANE PROTEIN PCR product | 1 | 10.49 | 1.59 |
| 130 | 219 | 90415MtBA | II. Cytoskeleton MICROTUBULE-ASSOCIATED PROTEIN PCR product | 1 | 11.25 | 1.59 |
| 131 | 117 | 10314MtBB | XII.C. Unknown function MO25 PROTEIN-LIKE PROTEIN PCR product | 0.00031472 | 9.08 | 1.59 |
| 132 | 160 | 91715MtBC | XII.C. Unknown function PCR product | 0.083583 | 9.38 | 1.59 |
| 133 | 307 | 91796MtBC | XIII. No homology PCR product | 1 | 9.99 | 1.58 |
| 134 | 132 | 30515MtBB | XIII. No homology PCR product | 0.0010393 | 10.48 | 1.58 |
| 135 | 87 | 00667MtBB | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 0.030452 | 11.67 | 1.58 |
| 136 | 118 | 92032MtBC | XIII. No homology PCR product | 1 | 8.95 | 1.58 |
| 137 | 210 | 92106MtBC | XIII. No homology PCR product | 0.43855 | 9.04 | 1.57 |
| 138 | 127 | 90829MtBB | XIII. No homology PCR product | 1.7114e-08 | 11.55 | 1.57 |
| 139 | 113 | MtSucS1_XbN61 | nodule-enhanced sucrose synthase MtSucS1 PCR product | 6.2089e-09 | 10.78 | 1.55 |
| 140 | 176 | 20352MtBC | XII.C. Unknown function PROTEIN 1-4 PCR product | 1 | 11.28 | 1.54 |
| 141 | 183 | 10648MtBA | VIII. Gene expression and RNA metabolism BASIC-HELIX-LOOP-HELIX DOMAIN CONTAINING PROTEIN PCR product | 1 | 10.45 | 1.53 |
| 142 | 220 | 50141MtBA | XIII. No homology PCR product | 1 | 10.81 | 1.52 |
| 143 | 203 | 90890MtBB | VI. Secondary metabolism and hormone metabolism LIPOXYGENASE PCR product | 0.72362 | 10.48 | 1.52 |

Tabelle C.3: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 144 | 128 | 50291MtBA | V. Primary metabolism CTP SYNTHASE PCR product | 1 | 8.73 | 1.51 |
| 145 | 291 | 00300MtBC | IX. Protein synthesis and processing PROTEASE INHIBITOR PCR product | 0.85941 | 10.08 | 1.51 |
| 146 | 214 | 30543MtBC | XIII. No homology PCR product | 1 | 11.01 | 1.51 |
| 147 | 330 | 50620MtBB | XII.C. Unknown function MtN7-LIKE PROTEIN PCR product | 1 | 11.61 | 1.51 |
| 148 | 161 | 10026MtBB | XII.C. Unknown function PCR product | 1 | 10.43 | 1.50 |
| 149 | 320 | 50368MtBC | VIII. Gene expression and RNA metabolism ETHYLENE RESPONSIVE ELEMENT BINDING FACTOR PCR product | 0.036536 | 9.53 | 1.50 |
| 150 | 3108 | 45516MtBA | XII.C. Unknown function Missing PCR product | 1 | 8.98 | 1.49 |
| 151 | 187 | 90844MtBB | IV. Vesicular trafficking secretion and protein sorting TSNARE/SYNTAXIN PCR product | 1 | 10.40 | 1.49 |
| 152 | 101 | 40019MtBB | IX. Protein synthesis and processing [NODULIN] MATRIX METALLOENDOPROTEINASE PCR product | 0.007704 | 8.12 | 1.49 |
| 153 | 169 | 91662MtBC | XIII. No homology PCR product | 1 | 11.54 | 1.49 |
| 154 | 339 | 91494MtBC | XIII. No homology PCR product | 1 | 10.17 | 1.49 |
| 155 | 5574 | 92043MtBC | XII.A. Defense and cell rescue [NODULIN] CHITOOLIGOSACCHARIDE DEACETYLASE Missing PCR product | 1 | 8.76 | 1.48 |
| 156 | 172 | 91712MtBC | V. Primary metabolism GLYCOSIDASE BETA-GLUCOSIDASE PCR product | 0.14058 | 10.86 | 1.48 |
| 157 | 168 | 00025MtBC | XII.B. Abiotic stimuli and development DORMANCY-ASSOCIATED / AUXIN-REPRESSED PROTEIN PCR product | 1 | 10.57 | 1.47 |
| 158 | 138 | 90776MtBB | XIII. No homology PCR product | 0.01342 | 11.06 | 1.47 |
| 159 | 212 | 90982MtBB | XIII. No homology PCR product | 1 | 11.09 | 1.46 |
| 160 | 249 | 50212MtBA | X. Signal transduction and post-translational regulation SER/THR PROTEIN KINASE PCR product | 1 | 9.54 | 1.45 |
| 161 | 151 | 20129MtBB | IV. Vesicular trafficking secretion and protein sorting ANNEXIN PCR product | 0.00057754 | 9.89 | 1.44 |
| 162 | 4718 | 50111MtBA | XII.C. Unknown function PCR product | 0 | 7.38 | 1.44 |
| 163 | 157 | 00588MtBB | XII.C. Unknown function [NODULIN] MtN25 PCR product | 0.00074588 | 9.14 | 1.44 |
| 164 | 3662 | 91788MtBC | IX. Protein synthesis and processing UBIQUITIN-LIKE PROTEIN SMT3 PCR product | 0 | 8.71 | 1.43 |
| 165 | 784 | 50845MtBC | XII.C. Unknown function ZINC FINGER CCCH SIGNATURE CONTAINING PROTEIN PCR product | 1 | 8.96 | 1.43 |
| 166 | 194 | 10759MtBA | XII.C. Unknown function SER-RICH PROTEIN PCR product | 0.045793 | 9.77 | 1.43 |
| 167 | 98 | 90643MtBB | XII.C. Unknown function [NODULIN] MtN28 PCR product | 0.0058105 | 8.54 | 1.43 |
| 168 | 167 | 91859MtBC | XIII. No homology PCR product | 1 | 10.87 | 1.43 |
| 169 | 274 | 50936MtBC | IX. Protein synthesis and processing 30S RIBOSOMAL PROTEIN S5 PCR product | 1 | 10.82 | 1.42 |
| 170 | 437 | 45394MtBC | XII.C. Unknown function PCR product | 1 | 11.83 | 1.42 |
| 171 | 739 | 10241MtBC | IX. Protein synthesis and processing 40S RIBOSOMAL PROTEIN S20 PCR product | 1 | 9.72 | 1.41 |
| 172 | 243 | 00740MtBB | III. Membrane transport MEMBRANE TRANSPORTER PCR product | 0.70675 | 9.84 | 1.41 |
| 173 | 180 | 91064MtBB | XIII. No homology PCR product | 0.36468 | 9.82 | 1.41 |
| 174 | 179 | 40104MtBA | III. Membrane transport HIGH-AFFINITY NITRATE TRANSPORTER PCR product | 0.78982 | 9.15 | 1.41 |
| 175 | 264 | 90733MtBB | XIII. No homology PCR product | 1 | 10.58 | 1.41 |
| 176 | 106 | 92155MtBC | II. Cytoskeleton NON INTERMEDIATE FILAMENT IFA BINDING PROTEIN PCR product | 0.00035939 | 11.04 | 1.40 |
| 177 | 83 | 10489MtBB | IX. Protein synthesis and processing 30S RIBOSOMAL PROTEIN S20 PCR product | 5.3674e-07 | 13.32 | 1.39 |
| 178 | 208 | 93300MtBB | XIII. No homology PCR product | 0.0028197 | 8.94 | 1.39 |
| 179 | 155 | 20357MtBC | III. Membrane transport PHOSPHATE/PHOSPHOENOLPYRUVATE TRANSLOCATOR PCR product | 1 | 10.25 | 1.39 |
| 180 | 761 | 20354MtBB | V. Primary metabolism ALDEHYDE DEHYDROGENASE PCR product | 1 | 9.06 | 1.39 |
| 181 | 142 | 90940MtBB | XII.C. Unknown function [NODULIN] MtN7 PCR product | 0.080665 | 10.06 | 1.38 |
| 182 | 137 | 90648MtBB | XIII. No homology PCR product | 0.18674 | 11.57 | 1.38 |
| 183 | 119 | 00483MtBB | XIII. No homology PCR product | 1.2194e-07 | 13.10 | 1.38 |
| 184 | 139 | 10128MtBC | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 0.060676 | 10.68 | 1.38 |
| 185 | 93 | 20333MtBC | XII.B. Abiotic stimuli and development AUXIN-RESPONSIVE PROTEIN-LIKE PROTEIN PCR product | 1 | 10.66 | 1.37 |
| 186 | 232 | 00397MtBC | XII.C. Unknown function PCR product | 1 | 10.10 | 1.37 |
| 187 | 519 | 91234MtBC | XII.C. Unknown function PCR product | 1 | 11.30 | 1.37 |
| 188 | 133 | 40187MtBB | XIII. No homology PCR product | 0.0040367 | 9.08 | 1.37 |
| 189 | 200 | MtAMLs204 | PR1 PCR product | 1 | 8.45 | 1.36 |
| 190 | 415 | 00246MtBB | XII.C. Unknown function CYS RICH PROTEIN PCR product | 1 | 9.11 | 1.35 |
| 191 | 124 | 40043MtBA | X. Signal transduction and post-translational regulation SHAGGY-RELATED PROTEIN KINASE PCR product | 0.00010592 | 12.06 | 1.35 |
| 192 | 207 | 00474MtBC | XII.A. Defense and cell rescue MYROSINASE BINDING PROTEIN PCR product | 0.92291 | 10.93 | 1.35 |
| 193 | 286 | 92180MtBC | XII.C. Unknown function PCR product | 1 | 11.46 | 1.35 |

Tabelle C.3: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 194 | 424 | 50814MtBC | XII.C. Unknown function PCR product | 1 | 9.71 | 1.34 |
| 195 | 369 | 93262MtBB | IX. Protein synthesis and processing PROTEIN RIBOSOMAL L32 PCR product | 1 | 9.76 | 1.34 |
| 196 | 256 | JVCPG40 | MtN13 PCR product | 0.010194 | 9.20 | 1.34 |
| 197 | 218 | 92039MtBC | XII.C. Unknown function PCR product | 0.091234 | 9.54 | 1.33 |
| 198 | 178 | 91473MtBC | V. Primary metabolism OXIDOREDUCTASE/DEHYDROGENASE PCR product | 1 | 10.82 | 1.33 |
| 199 | 164 | 45365MtBA | IV. Vesicular trafficking secretion and protein sorting VACUOLAR PROTEIN SORTING PROTEIN PCR product | 1 | 9.89 | 1.32 |
| 200 | 289 | 00670MtBA | IV. Vesicular trafficking secretion and protein sorting CLATHRIN COAT ASSEMBLY PROTEIN PCR product | 1 | 11.22 | 1.32 |
| 201 | 277 | 90892MtBB | III. Membrane transport PEPTIDE TRANSPORTER PCR product | 1 | 12.13 | 1.31 |
| 202 | 515 | 91534MtBC | IV. Vesicular trafficking secretion and protein sorting SYNTAXIN PCR product | 1 | 10.22 | 1.31 |
| 203 | 166 | 90545MtBA | XIII. No homology PCR product | 1 | 11.77 | 1.31 |
| 204 | 280 | 00127MtBC | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 0.79589 | 9.76 | 1.30 |
| 205 | 197 | 00436MtBB | XIII. No homology PCR product | 0.035249 | 11.09 | 1.29 |
| 206 | 304 | 50657MtBC | XII.C. Unknown function PCR product | 0.54567 | 9.97 | 1.29 |
| 207 | 312 | 00392MtBC | XII.C. Unknown function PCR product | 0.93156 | 10.06 | 1.28 |
| 208 | 353 | 40081MtBA | V. Primary metabolism NADH-UBIQUINONE OXIDOREDUCTASE 75 KDA SUBUNIT PCR product | 1 | 10.02 | 1.28 |
| 209 | 189 | 30198MtBB | II. Cytoskeleton TUBULIN BETA CHAIN PCR product | 0.0071203 | 9.74 | 1.28 |
| 210 | 177 | 10972MtBC | X. Signal transduction and post-translational regulation PROTEIN KINASE DOMAIN CONTAINING PROTEIN PCR product | 1 | 10.74 | 1.28 |
| 211 | 308 | 90632MtBB | XIII. No homology PCR product | 1 | 9.53 | 1.27 |
| 212 | 225 | 50863MtBC | XIII. No homology PCR product | 0.27913 | 11.27 | 1.26 |
| 213 | 359 | 10987MtBC | XIII. No homology PCR product | 0.029239 | 9.68 | 1.26 |
| 214 | 141 | 00093MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 0.0061672 | 10.92 | 1.26 |
| 215 | 149 | 10392MtBB | XII.C. Unknown function PCR product | 1 | 8.90 | 1.26 |
| 216 | 247 | 10348MtBC | XII.C. Unknown function PCR product | 0.70544 | 9.26 | 1.26 |
| 217 | 109 | 01434MtBB | IX. Protein synthesis and processing HEAT SHOCK 70 KDA PROTEIN MITOCHONDRIAL PRECURSOR PCR product | 0.00060947 | 10.07 | 1.26 |
| 218 | 324 | 00697MtBB | III. Membrane transport AMINO ACID TRANSPORTER PCR product | 1 | 9.16 | 1.26 |
| 219 | 148 | JVCPG46 | MtN19 PCR product | 0.00026812 | 8.87 | 1.25 |
| 220 | 430 | 40166MtBA | III. Membrane transport SUGAR TRANSPORTER PCR product | 1 | 12.01 | 1.25 |
| 221 | 306 | 40163MtBA | V. Primary metabolism PHOSPHOETHANOLAMINE N-METHYLTRANSFERASE PCR product | 0.074637 | 10.04 | 1.24 |
| 222 | 440 | 91793MtBC | XIII. No homology PCR product | 1 | 10.23 | 1.24 |
| 223 | 371 | 00316MtBC | I. Cell Wall ARABINOGALACTAN-PROTEIN PRECURSOR PCR product | 0.066898 | 9.82 | 1.24 |
| 224 | 613 | 10763MtBC | II. Cytoskeleton ANNEXIN PCR product | 0.22304 | 10.37 | 1.24 |
| 225 | 341 | 00312MtBB | IX. Protein synthesis and processing MULTICATALYTIC ENDOPEPTIDASE COMPLEX PROTEASOME COMPONENT BETA SUBUNIT PCR product | 1 | 9.80 | 1.24 |
| 226 | 300 | 91390MtBC | IX. Protein synthesis and processing AMINOTRANSFERASE PCR product | 0.45795 | 9.38 | 1.23 |
| 227 | 491 | 91417MtBC | XIII. No homology PCR product | 1 | 9.27 | 1.23 |
| 228 | 104 | 00514MtBB | V. Primary metabolism HISTIDINE DECARBOXYLASE PCR product | 3.6015e-06 | 11.79 | 1.23 |
| 229 | 334 | 90468MtBA | XIII. No homology PCR product | 1 | 11.76 | 1.23 |
| 230 | 610 | 91877MtBC | XIII. No homology PCR product | 0.82412 | 9.69 | 1.23 |
| 231 | 801 | 91747MtBC | V. Primary metabolism GLUTAMINE SYNTHETASE PCR product | 1 | 10.20 | 1.23 |
| 232 | 4049 | 11008MtBC | XIII. No homology AMINOTRANSFERASE PCR product | 0 | 8.62 | 1.22 |
| 233 | 171 | 10307MtBB | VII. Chromatin and DNA metabolism histone H2A PCR product | 0.0021876 | 9.28 | 1.21 |
| 234 | 131 | 10310MtBB | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 1 | 9.98 | 1.21 |
| 235 | 174 | 10279MtBC | V. Primary metabolism HEME OXYGENASE PCR product | 5.1576e-06 | 11.50 | 1.21 |
| 236 | 162 | 00666MtBC | XII.C. Unknown function WOUND-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.27408 | 8.74 | 1.21 |
| 237 | 331 | 91270MtBC | XIII. No homology PCR product | 0.03231 | 10.03 | 1.20 |
| 238 | 153 | 93362MtBC | XII.C. Unknown function PCR product | 0.067509 | 8.87 | 1.20 |
| 239 | 454 | 30466MtBC | XII.C. Unknown function PCR product | 1 | 10.52 | 1.20 |
| 240 | 514 | 50144MtBA | X. Signal transduction and post-translational regulation RAS-GTPASE-ACTIVATING PROTEIN SH3-DOMAIN BINDING PROTEIN PCR product | 1 | 11.95 | 1.20 |

Tabelle C.3: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung Imagene

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 241 | 372 | 91217MtBC | V. Primary metabolism VACUOLAR ATP SYNTHASE SUBUNIT PCR product | 1 | 9.46 | 1.19 |
| 242 | 399 | 30464MtBC | XII.C. Unknown function PCR product | 1 | 11.68 | 1.19 |
| 243 | 401 | 10178MtBB | I. Cell Wall CINNAMOYL-COA REDUCTASE PCR product | 0.41449 | 10.35 | 1.19 |
| 244 | 152 | 20236MtBB | V. Primary metabolism ALKALINE/NEUTRAL INVERTASE PCR product | 0.014604 | 8.55 | 1.19 |
| 245 | 650 | 91674MtBC | XIII. No homology CARBOXYPEPTIDASE PRECURSOR PCR product | 0.27756 | 9.97 | 1.18 |
| 246 | 309 | GVSN-24A24 | MtN13 PCR product | 0.014203 | 12.11 | 1.18 |
| 247 | 516 | 00653MtBC | V. Primary metabolism NADPH-CYTOCHROME P450 REDUCTASE PCR product | 1 | 10.17 | 1.18 |
| 248 | 711 | 91922MtBC | X. Signal transduction and post-translational regulation BETA-GLUCAN-ELICITOR RECEPTOR PCR product | 1 | 10.35 | 1.18 |
| 249 | 5479 | 45101MtBC | VII. Chromatin and DNA metabolism HISTONE H2A Missing PCR product | 0 | 7.08 | 1.18 |
| 250 | 196 | 30103MtBC | V. Primary metabolism FERREDOXIN–NITRITE REDUCTASE PCR product | 0.053192 | 8.75 | 1.17 |

Tabelle C.4: M-sortierte Liste der Sondensequenzen auf dem Mt6kRIT-Mikroarray des Nodulationsexperimentes Nod 10, bei Bildauswertung mit dem hier beschriebenen System AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 1 | 3 | Globin | Globin PCR product | 1.0876e-11 | 12.55 | 5.36 |
| 2 | 1 | mtgmabc120001a02 | leghemoglobin, TC6726-homolog, 70% zu VfLb29 PCR product | 4.3717e-16 | 12.52 | 5.30 |
| 3 | 5 | JVCPG49 | MtN22 PCR product | 2.1354e-07 | 12.39 | 5.22 |
| 4 | 2 | 10695MtBB | XIII. No homology PCR product | 1.541e-12 | 12.32 | 5.19 |
| 5 | 4 | mt–aac120001a02 | leghemoglobin, TC13379-homolog, 73% zu VfLb29 PCR product | 6.1553e-17 | 12.60 | 5.17 |
| 6 | 7 | mt–aac120001a01 | leghemoglobin, TC10534-homolog, 77% zu VfLb29 PCR product | 5.2973e-14 | 12.40 | 5.00 |
| 7 | 6 | GFP | Green Fluorescent Protein PCR product | 3.4449e-13 | 12.44 | 4.96 |
| 8 | 9 | 00583MtBB | V. Primary metabolism [NODULIN] LEGHEMOGLOBIN 1 PCR product | 5.9474e-17 | 12.81 | 4.76 |
| 9 | 10 | GVN-72A6 | nodulin 25 PCR product | 1.5566e-16 | 12.82 | 4.73 |
| 10 | 8 | 10497MtBB | XIII. No homology PCR product | 4.0155e-14 | 12.11 | 4.65 |
| 11 | 12 | GVN-71A1 | nodulin 26 PCR product | 5.6491e-07 | 12.31 | 4.26 |
| 12 | 11 | GVN-55E6 | nodulin 25 PCR product | 0.00011414 | 12.35 | 4.03 |
| 13 | 14 | MtAe27 | Unknown function PCR product | 1.5183e-10 | 12.27 | 3.94 |
| 14 | 15 | GVN-64D16 | enod8 PCR product | 2.4018e-15 | 13.24 | 3.82 |
| 15 | 19 | mt–aac120001a03 | leghemoglobin, TC3089-homolog, 70% zu VfLb29 PCR product | 4.0239e-19 | 13.25 | 3.77 |
| 16 | 18 | GVN-72E9 | nodulin 26 PCR product | 0.00020799 | 11.46 | 3.55 |
| 17 | 22 | 00281MtBB | XII.C. Unknown function [NODULIN] MTN22 PCR product | 1.398e-10 | 13.38 | 3.50 |
| 18 | 17 | MtAe34 | Unknown function PCR product | 4.003e-08 | 13.18 | 3.40 |
| 19 | 20 | 00646MtBB | XIII. No homology PCR product | 1.4803e-13 | 12.30 | 3.35 |
| 20 | 21 | 10766MtBB | XIII. No homology PCR product | 4.8859e-10 | 11.20 | 3.25 |
| 21 | 13 | 93119MtBB | XIII. No homology PCR product | 1.3821e-09 | 11.39 | 3.19 |
| 22 | 24 | 00800MtBC | V. Primary metabolism ASPARAGINE SYNTHETASE [GLUTAMINE-HYDROLYZING] PCR product | 8.2673e-13 | 11.55 | 2.90 |
| 23 | 36 | 90927MtBB | XIII. No homology PCR product | 2.8369e-06 | 11.26 | 2.88 |
| 24 | 16 | 10430MtBC | III. Membrane transport MAJOR INTRINSIC PROTEIN (NODULIN26-LIKE ) PCR product | 4.3808e-05 | 11.27 | 2.88 |
| 25 | 69 | GVN-55B13 | MtN1 PCR product | 0.0038855 | 10.75 | 2.82 |
| 26 | 32 | 10811MtBB | X. Signal transduction REMORIN PCR product | 1.9461e-08 | 10.83 | 2.67 |
| 27 | 28 | 10685MtBB | XIII. No homology PCR product | 6.4725e-09 | 12.15 | 2.64 |
| 28 | 23 | 00156MtBB | V. Primary metabolism [NODULIN] CARBONIC ANHYDRASE PCR product | 9.2294e-10 | 10.95 | 2.58 |
| 29 | 34 | MsCA | carbonic anhydrase PCR product | 3.1047e-07 | 12.27 | 2.46 |
| 30 | 31 | 30105MtBB | V. Primary metabolism BETA-AMYLASE PCR product | 1.1316e-08 | 11.36 | 2.43 |
| 31 | 39 | HLG100 | HLG 1.0 well 100 PCR product | 5.8351e-09 | 12.14 | 2.38 |
| 32 | 29 | 20204MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 4.427e-06 | 10.65 | 2.35 |
| 33 | 42 | JVCPG51 | MtN24 PCR product | 0.005759 | 10.43 | 2.35 |
| 34 | 45 | GVN-7016 | enod40 PCR product | 9.903e-11 | 12.24 | 2.34 |
| 35 | 26 | 45429MtBB | III. Membrane transport HEXOSE TRANSPORTER PCR product | 0.011491 | 11.40 | 2.30 |
| 36 | 27 | MtGSa | glutamine synthetase a PCR product | 2.5182e-08 | 11.72 | 2.29 |
| 37 | 25 | 00344MtBB | XII. Miscellaneous lectin PCR product | 5.8038e-08 | 10.87 | 2.28 |
| 38 | 33 | 10058MtBA | XII.C. Unknown function PD001144 PCR product | 4.7065e-09 | 10.86 | 2.27 |
| 39 | 30 | mtgmabc120001a04 | putative chloride ion channel protein, TC10381-homolog, 89% zu VfCLIC1 PCR product | 5.5009e-12 | 10.76 | 2.25 |
| 40 | 51 | JVCPG27 | ENOD16 PCR product | 4.3133e-07 | 11.28 | 2.24 |
| 41 | 57 | MtAMLs296 | DNA binding protein homolog PCR product | 9.1245e-05 | 10.60 | 2.20 |
| 42 | 35 | JVCPG44 | MtN17 PCR product | 3.4335e-08 | 11.29 | 2.20 |
| 43 | 50 | 00043MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 4.1595e-10 | 11.71 | 2.19 |
| 44 | 38 | 00120MtBB | XII.C. Unknown function [NODULIN] MTN29 PCR product | 8.7756e-08 | 10.66 | 2.16 |
| 45 | 56 | JVCPG25 | ENOD11 PCR product | 7.9706e-11 | 13.07 | 2.14 |

Tabelle C.4: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 46 | 43 | 40167MtBB | XII.C. Unknown function MtN9-LIKE PROTEIN PCR product | 7.8959e-08 | 10.74 | 2.13 |
| 47 | 64 | GVN-65K1 | leghemoglobin 1 PCR product | 1.5266e-14 | 14.10 | 2.12 |
| 48 | 62 | JVCPG37 | MtN10 PCR product | 5.9504e-10 | 13.79 | 2.11 |
| 49 | 37 | 90646MtBB | XIII. No homology PCR product | 9.1343e-08 | 10.76 | 2.10 |
| 50 | 59 | 90870MtBB | XIII. No homology PCR product | 2.0065e-08 | 10.50 | 2.08 |
| 51 | 49 | 00203MtBB | XII.C. Unknown function PCR product | 0.032091 | 11.98 | 2.07 |
| 52 | 53 | JVCPG48 | MtN21 PCR product | 6.8855e-09 | 10.72 | 2.01 |
| 53 | 40 | JVCPG58 | ENOD40 PCR product | 3.4612e-09 | 11.30 | 1.97 |
| 54 | 41 | JVCPG42 | MtN15 PCR product | 4.7108e-07 | 11.35 | 1.93 |
| 55 | 109 | 90408MtBA | XII.C. Unknown function PCR product | 0.0029453 | 10.17 | 1.92 |
| 56 | 46 | KV3-21B21 | MtN21-like PCR product | 2.3377e-06 | 11.37 | 1.90 |
| 57 | 44 | JVCPG53 | MtN26 PCR product | 3.7375e-05 | 10.81 | 1.88 |
| 58 | 48 | 00186MtBB | XII.C. Unknown function [NODULIN] MTN16 PCR product | 0.024741 | 12.34 | 1.86 |
| 59 | 60 | 90571MtBB | XII.C. Unknown function [NODULIN] MtN19 PCR product | 0.013382 | 10.47 | 1.84 |
| 60 | 84 | MHAM-47P10 | MtN93 PCR product | 8.5362e-05 | 10.82 | 1.83 |
| 61 | 71 | 91525MtBC | XII.C. Unknown function PCR product | 1.4886e-06 | 10.95 | 1.82 |
| 62 | 73 | 00240MtBB | XII.C. Unknown function [NODULIN] EARLY NODULIN 12 PRECURSOR PCR product | 6.7625e-13 | 13.81 | 1.80 |
| 63 | 47 | 00746MtBB | XII.C. Unknown function [NODULIN] MTN11 PCR product | 2.4563e-06 | 10.61 | 1.77 |
| 64 | 58 | 20131MtBA | VI. Secondary metabolism and hormone metabolism FERRITIN PCR product | 2.8341e-06 | 10.65 | 1.72 |
| 65 | 72 | 00060MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 3.3559e-05 | 11.35 | 1.69 |
| 66 | 89 | JVCPG1 | glutamine synthetase MtGSIa PCR product | 1.3295e-10 | 12.34 | 1.67 |
| 67 | 122 | JVCPG55 | MtN28 PCR product | 0.13725 | 10.09 | 1.65 |
| 68 | 68 | 10073MtBB | XII.C. Unknown function PCR product | 2.6899e-07 | 11.55 | 1.65 |
| 69 | 115 | 90910MtBB | XII.C. Unknown function [NODULIN] MTN20 PCR product | 2.153e-06 | 10.56 | 1.65 |
| 70 | 61 | GVN-74J10 | MtN21-like PCR product | 0.00016721 | 11.65 | 1.64 |
| 71 | 99 | 00515MtBB | XIII. No homology PCR product | 0.016936 | 10.52 | 1.63 |
| 72 | 70 | 45203MtBA | V. Primary metabolism CTP SYNTHASE PCR product | 5.4002e-07 | 10.71 | 1.62 |
| 73 | 98 | 20114MtBA | V. Primary metabolism GLUTAMINE SYNTHETASE PCR product | 7.9515e-10 | 12.94 | 1.61 |
| 74 | 83 | MtSucS2 | second sucrose synthase MtSucS2, MtBC02H08 PCR product | 1.299e-07 | 11.27 | 1.60 |
| 75 | 52 | JVCPG29 | MtN1 PCR product | 0.035375 | 11.30 | 1.59 |
| 76 | 77 | 10203MtBB | XII.C. Unknown function PHOSPHATE-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.028145 | 10.78 | 1.58 |
| 77 | 100 | 91942MtBC | IX. Protein synthesis and processing TRANSAMIDASE PCR product | 0.00028055 | 10.89 | 1.58 |
| 78 | 67 | 00651MtBC | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 9.4971e-07 | 10.71 | 1.58 |
| 79 | 63 | 00068MtBB | XII.C. Unknown function [NODULIN] MTN1 PRECURSOR PCR product | 5.0782e-07 | 10.85 | 1.49 |
| 80 | 76 | 91023MtBB | XII.C. Unknown function [NODULIN] PCR product | 9.9322e-08 | 10.75 | 1.46 |
| 81 | 113 | 00182MtBC | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 0.0049253 | 12.26 | 1.43 |
| 82 | 105 | 10140MtBA | XII.B. Abiotic stimuli and development AUXIN INDUCED PROTEIN LIKE-PROTEIN PCR product | 1.9496e-05 | 10.94 | 1.42 |
| 83 | 177 | 10489MtBB | IX. Protein synthesis and processing 30S RIBOSOMAL PROTEIN S20 PCR product | 1.4736e-06 | 13.86 | 1.42 |
| 84 | 82 | JVCPG33 | MtN5 PCR product | 0.00065479 | 11.93 | 1.40 |
| 85 | 96 | GVN-74I6 | MtN21 PCR product | 0.00017572 | 10.83 | 1.39 |
| 86 | 87 | 90689MtBB | III. Membrane transport CATIONIC AMINO ACID TRANSPORTER PCR product | 0.0030094 | 10.22 | 1.39 |
| 87 | 135 | 00667MtBB | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 9.8331e-05 | 12.13 | 1.39 |
| 88 | 88 | 92225MtBC | XIII. No homology PCR product | 1 | 12.04 | 1.37 |
| 89 | 268 | MsPEPC | PEP carboxylase PCR product | 0.015217 | 10.88 | 1.36 |
| 90 | 97 | 10358MtBB | XII.C. Unknown function [NODULIN] PCR product | 2.8135e-06 | 12.92 | 1.33 |
| 91 | 127 | HLG200 | HLG 1.0 well 200 PCR product | 4.4861e-05 | 13.57 | 1.33 |
| 92 | 65 | 20267MtBB | XII.C. Unknown function [NODULIN] MTN26 PCR product | 0.052268 | 10.22 | 1.31 |
| 93 | 185 | 20333MtBC | XII.B. Abiotic stimuli and development AUXIN-RESPONSIVE PROTEIN-LIKE PROTEIN PCR product | 0.18991 | 11.45 | 1.30 |
| 94 | 79 | 91471MtBC | IX. Protein synthesis and processing CYSTEINE PROTEINASE PCR product | 0.0031038 | 10.59 | 1.30 |
| 95 | 93 | 00290MtBB | XII.C. Unknown function PD155440 PCR product | 1.9318e-05 | 10.51 | 1.25 |

Tabelle C.4: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 96 | 112 | 00372MtBC | XIII. No homology PCR product | 3.0731e-06 | 11.71 | 1.24 |
| 97 | 94 | 30404MtBC | XII.C. Unknown function PCR product | 1 | 10.88 | 1.23 |
| 98 | 167 | 90643MtBB | XII.C. Unknown function [NODULIN] MtN28 PCR product | 0.0010965 | 10.13 | 1.21 |
| 99 | 95 | 45259MtBB | V. Primary metabolism 34-DIHYDROXY-2-BUTANONE-4-PHOSPHATE SYNTHASE PCR product | 0.0055044 | 11.13 | 1.20 |
| 100 | 116 | JVCPG56 | MtN29 chimeric PCR product | 0.0004157 | 12.82 | 1.19 |
| 101 | 152 | 40019MtBB | IX. Protein synthesis and processing [NODULIN] MATRIX METALLOENDOPROTEINASE PCR product | 1 | 10.22 | 1.18 |
| 102 | 516 | 30378MtBC | XII. Miscellaneous BLUE COPPER PROTEIN PCR product | 1 | 9.99 | 1.16 |
| 103 | 102 | 90782MtBB | IX. Protein synthesis and processing [PROTEIN-PII] URIDYLYLTRANSFERASE PCR product | 0.0027808 | 10.24 | 1.16 |
| 104 | 228 | 00514MtBB | V. Primary metabolism HISTIDINE DECARBOXYLASE PCR product | 5.0527e-07 | 12.21 | 1.16 |
| 105 | 111 | JVCPG36 | MtN9 PCR product | 0.044158 | 10.24 | 1.15 |
| 106 | 176 | 92155MtBC | II. Cytoskeleton NON INTERMEDIATE FILAMENT IFA BINDING PROTEIN PCR product | 0.00013904 | 11.66 | 1.15 |
| 107 | 106 | JVCPG52 | MtN25 PCR product | 0.00029002 | 11.52 | 1.14 |
| 108 | 972 | 93093MtBA | XII.B. Abiotic stimuli and development WOUND INDUCIVE PROTEIN Missing PCR product | 1 | 9.95 | 1.14 |
| 109 | 217 | 01434MtBB | IX. Protein synthesis and processing HEAT SHOCK 70 KDA PROTEIN MITOCHONDRIAL PRECURSOR PCR product | 2.5181e-08 | 11.56 | 1.14 |
| 110 | 74 | 10690MtBB | XII.A. Defense and cell rescue MTN13 PCR product | 0.53617 | 11.30 | 1.13 |
| 111 | 689 | 91361MtBC | XII.C. Unknown function PCR product | 1 | 10.46 | 1.13 |
| 112 | 509 | 93406MtBB | I. Cell Wall PROFUCOSIDASE PRECURSOR PCR product | 0.030281 | 10.27 | 1.13 |
| 113 | 139 | MtSucS1_XbN61 | nodule-enhanced sucrose synthase MtSucS1 PCR product | 9.8622e-09 | 11.94 | 1.13 |
| 114 | 101 | 91074MtBB | XIII. No homology PCR product | 2.4338e-05 | 10.53 | 1.12 |
| 115 | 104 | 20258MtBC | X. Signal transduction and post-translational regulation PROTEIN KINASE PCR product | 1 | 10.89 | 1.12 |
| 116 | 327 | 91055MtBB | XIII. No homology PCR product | 0.5169 | 9.81 | 1.11 |
| 117 | 131 | 10314MtBB | XII.C. Unknown function MO25 PROTEIN-LIKE PROTEIN PCR product | 2.9647e-07 | 10.86 | 1.11 |
| 118 | 136 | 92032MtBC | XIII. No homology PCR product | 0.22872 | 10.33 | 1.11 |
| 119 | 183 | 00483MtBB | XIII. No homology PCR product | 1.2711e-07 | 13.93 | 1.11 |
| 120 | 262 | 50612MtBB | XII.C. Unknown function [NODULIN] ENOD40 PCR product | 7.4215e-06 | 13.20 | 1.10 |
| 121 | 107 | 93260MtBB | XII. Miscellaneous [NODULIN] MTN5 (NON SPECIFIC LIPID TRANSFER PROTEIN) PCR product | 3.1303e-05 | 10.54 | 1.10 |
| 122 | 66 | 91668MtBC | XIII. No homology PCR product | 0.91009 | 11.77 | 1.10 |
| 123 | 491 | 90681MtBB | V. Primary metabolism ALCOHOL DEHYDROGENASE PCR product | 0.0012042 | 10.51 | 1.10 |
| 124 | 191 | 40043MtBA | X. Signal transduction and post-translational regulation SHAGGY-RELATED PROTEIN KINASE PCR product | 0.0016769 | 12.98 | 1.08 |
| 125 | 103 | 00643MtBB | XII.C. Unknown function PCR product | 0.0024253 | 10.69 | 1.08 |
| 126 | 277 | 30497MtBB | XII.C. Unknown function [PUTATIVE NODULIN] PCR product | 0.00032648 | 10.36 | 1.08 |
| 127 | 138 | 90829MtBB | XIII. No homology PCR product | 3.2299e-09 | 12.47 | 1.08 |
| 128 | 144 | 50291MtBA | V. Primary metabolism CTP SYNTHASE PCR product | 1 | 10.08 | 1.07 |
| 129 | 459 | JVCPG38 | MtN11 PCR product | 0.00070604 | 10.00 | 1.07 |
| 130 | 563 | 30582MtBC | XII.A. Defense and cell rescue CATHEPSIN D INHIBITOR PCR product | 0.0009151 | 10.00 | 1.07 |
| 131 | 234 | 10310MtBB | VIII. Gene expression and RNA metabolism ZINC FINGER PROTEIN PCR product | 1 | 11.35 | 1.06 |
| 132 | 134 | 30515MtBB | XIII. No homology PCR product | 4.0202e-07 | 11.27 | 1.06 |
| 133 | 188 | 40187MtBB | XIII. No homology PCR product | 1.733e-05 | 10.67 | 1.06 |
| 134 | 253 | 10107MtBB | IX. Protein synthesis and processing PROTEASOME SUBUNIT BETA TYPE PCR product | 6.0679e-07 | 13.62 | 1.06 |
| 135 | 893 | 00441MtBB | XII.C. Unknown function PCR product | 1 | 10.49 | 1.05 |
| 136 | 86 | 91127MtBB | XIII. No homology PCR product | 1 | 10.96 | 1.05 |
| 137 | 182 | 90648MtBB | XIII. No homology PCR product | 0.29656 | 12.35 | 1.04 |
| 138 | 158 | 90776MtBB | XIII. No homology PCR product | 0.020228 | 11.84 | 1.03 |
| 139 | 184 | 10128MtBC | XII.B. Abiotic stimuli and development ALUMINUM-INDUCED AUXIN-REPRESSED PROTEIN PCR product | 0.0084173 | 11.47 | 1.03 |
| 140 | 81 | 10850MtBC | XII.C. Unknown function PLASTOGLOBULE ASSOCIATED PROTEIN PCR product | 0.37303 | 11.63 | 1.03 |
| 141 | 214 | 00093MtBB | IX. Protein synthesis and processing MICROSOMAL SIGNAL PEPTIDASE SUBUNIT PCR product | 0.0044937 | 11.56 | 1.02 |
| 142 | 181 | 90940MtBB | XII.C. Unknown function [NODULIN] MtN7 PCR product | 0.011177 | 11.10 | 1.01 |
| 143 | 80 | 91628MtBC | VIII. Gene expression and RNA metabolism SPLICING FACTOR PCR product | 0.31339 | 12.29 | 1.01 |
| 144 | 276 | 91737MtBC | XII.C. Unknown function PCR product | 1 | 10.03 | 1.01 |
| 145 | 123 | 91397MtBC | XII.C. Unknown function PCR product | 1 | 11.01 | 1.00 |

Tabelle C.4: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 146 | 507 | 91140MtBB | XIII. No homology PCR product | 0.0093484 | 10.10 | 1.00 |
| 147 | 117 | 10008MtBC | V. Primary metabolism PROTEIN 2 PRECURSOR OEE2 23 KD SUBUNIT OF OXYGEN EVOLVING SYSTEM OF PHOTOSYSTEM II PCR product | 0.46967 | 11.17 | 1.00 |
| 148 | 219 | JVCPG46 | MtN19 PCR product | 0.00011479 | 10.47 | 1.00 |
| 149 | 215 | 10392MtBB | XII.C. Unknown function PCR product | 0.028372 | 10.06 | 1.00 |
| 150 | 108 | 90508MtBA | XII.C. Unknown function CONSERVED PROTEIN PCR product | 0.0675 | 10.00 | 1.00 |
| 151 | 161 | 20129MtBB | IV. Vesicular trafficking secretion and protein sorting ANNEXIN PCR product | 1.0891e-05 | 11.07 | 0.99 |
| 152 | 244 | 20236MtBB | V. Primary metabolism ALKALINE/NEUTRAL INVERTASE PCR product | 0.0014955 | 10.26 | 0.99 |
| 153 | 238 | 93362MtBC | XII.C. Unknown function PCR product | 0.080269 | 10.36 | 0.99 |
| 154 | 899 | 92033MtBC | V. Primary metabolism PHOSPHOENOLPYRUVATE CARBOXYKINASE [ATP] PCR product | 1 | 10.13 | 0.99 |
| 155 | 179 | 20357MtBC | III. Membrane transport PHOSPHATE/PHOSPHOENOLPYRUVATE TRANSLOCATOR PCR product | 1 | 11.30 | 0.97 |
| 156 | 1631 | 20268MtBB | XII.C. Unknown function [NODULIN] MtN6 PCR product | 0.097929 | 9.57 | 0.97 |
| 157 | 163 | 00588MtBB | XII.C. Unknown function [NODULIN] MtN25 PCR product | 5.6094e-05 | 10.56 | 0.97 |
| 158 | 342 | 45590MtBA | IX. Protein synthesis and processing PHENYLALANYL TRNA SYNTHETASE PCR product | 4.1328e-06 | 10.62 | 0.96 |
| 159 | 510 | 10364MtBC | XII.C. Unknown function PCR product | 0.0058872 | 9.89 | 0.96 |
| 160 | 132 | 91715MtBC | XII.C. Unknown function PCR product | 0.57878 | 10.83 | 0.95 |
| 161 | 148 | 10026MtBB | XII.C. Unknown function PCR product | 1 | 11.40 | 0.95 |
| 162 | 236 | 00666MtBC | XII.C. Unknown function WOUND-INDUCED PROTEIN-LIKE PROTEIN PCR product | 0.14531 | 10.42 | 0.94 |
| 163 | 301 | 45510MtBC | XII.C. Unknown function PCR product | 0.0096719 | 10.12 | 0.94 |
| 164 | 199 | 45365MtBA | IV. Vesicular trafficking secretion and protein sorting VACUOLAR PROTEIN SORTING PROTEIN PCR product | 1 | 11.23 | 0.94 |
| 165 | 5085 | MtAe96 | Unknown function PCR product | 0.0059333 | 9.88 | 0.94 |
| 166 | 203 | 90545MtBA | XIII. No homology PCR product | 0.06364 | 12.32 | 0.94 |
| 167 | 168 | 91859MtBC | XIII. No homology PCR product | 0.2351 | 11.95 | 0.94 |
| 168 | 157 | 00025MtBC | XII.B. Abiotic stimuli and development DORMANCY-ASSOCIATED / AUXIN-REPRESSED PROTEIN PCR product | 1 | 11.43 | 0.94 |
| 169 | 153 | 91662MtBC | XIII. No homology PCR product | 1 | 12.26 | 0.93 |
| 170 | 288 | 91599MtBC | XIII. No homology PCR product | 0.24131 | 11.78 | 0.93 |
| 171 | 233 | 10307MtBB | VII. Chromatin and DNA metabolism histone H2A PCR product | 9.4025e-07 | 10.60 | 0.93 |
| 172 | 156 | 91712MtBC | V. Primary metabolism GLYCOSIDASE BETA-GLUCOSIDASE PCR product | 0.073874 | 11.67 | 0.92 |
| 173 | 349 | 10329MtBA | XIII. No homology PCR product | 0.020338 | 11.78 | 0.92 |
| 174 | 235 | 10279MtBC | V. Primary metabolism HEME OXYGENASE PCR product | 0.00041754 | 12.15 | 0.91 |
| 175 | 322 | 90896MtBB | XIII. No homology PCR product | 1 | 10.41 | 0.91 |
| 176 | 140 | 20352MtBC | XII.C. Unknown function PROTEIN 1-4 PCR product | 0.49456 | 12.03 | 0.91 |
| 177 | 210 | 10972MtBC | X. Signal transduction and post-translational regulation PROTEIN KINASE DOMAIN CONTAINING PROTEIN PCR product | 0.47418 | 11.38 | 0.91 |
| 178 | 198 | 91473MtBC | V. Primary metabolism OXIDOREDUCTASE/DEHYDROGENASE PCR product | 0.30174 | 11.52 | 0.91 |
| 179 | 174 | 40104MtBA | III. Membrane transport HIGH-AFFINITY NITRATE TRANSPORTER PCR product | 0.85568 | 10.95 | 0.90 |
| 180 | 173 | 91064MtBB | XIII. No homology PCR product | 1 | 10.89 | 0.90 |
| 181 | 363 | 91665MtBC | XII.A. Defense and cell rescue GLUTATHIONE S-TRANSFERASE PCR product | 1 | 10.18 | 0.90 |
| 182 | 627 | 90874MtBB | XII.C. Unknown function PCR product | 1 | 10.30 | 0.90 |
| 183 | 141 | 10648MtBA | VIII. Gene expression and RNA metabolism BASIC-HELIX-LOOP-HELIX DOMAIN CONTAINING PROTEIN PCR product | 1 | 11.36 | 0.90 |
| 184 | 380 | 10532MtBB | XII.C. Unknown function PCR product | 0.045462 | 12.07 | 0.89 |
| 185 | 320 | 10683MtBA | VII. Chromatin and DNA metabolism REVERSE TRANSCRIPTASE PCR product | 0.10645 | 11.86 | 0.89 |
| 186 | 296 | 91380MtBC | XIII. No homology PCR product | 0.098706 | 11.32 | 0.89 |
| 187 | 151 | 90844MtBB | IV. Vesicular trafficking secretion and protein sorting TSNARE/SYNTAXIN PCR product | 1 | 11.51 | 0.89 |
| 188 | 281 | MtAMLs32 | Putative wound induced protein PCR product | 0.0059264 | 14.34 | 0.89 |
| 189 | 209 | 30198MtBB | II. Cytoskeleton TUBULIN BETA CHAIN PCR product | 0.0014842 | 10.83 | 0.89 |
| 190 | 114 | 10495MtBB | X. Signal transduction and post-translational regulation PROTEIN PHOSPHATASE-2C PCR product | 1 | 11.34 | 0.88 |
| 191 | 255 | 90354MtBA | VIII. Gene expression and RNA metabolism POLY(A) BINDING PROTEIN II PCR product | 1 | 10.93 | 0.88 |
| 192 | 1144 | 91442MtBC | PCR product | 1 | 10.03 | 0.88 |

Tabelle C.4: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 193 | 85 | 91485MtBC | IX. Protein synthesis and processing RIBOSOMAL PROTEIN PCR product | 0.5533 | 12.64 | 0.88 |
| 194 | 166 | 10759MtBA | XII.C. Unknown function SER-RICH PROTEIN PCR product | 0.10399 | 10.80 | 0.88 |
| 195 | 559 | 10968MtBB | VI. Secondary metabolism and hormone metabolism PROFUCOSIDASE PRECURSOR PCR product | 8.0548e-05 | 9.97 | 0.88 |
| 196 | 250 | 30103MtBC | V. Primary metabolism FERREDOXIN–NITRITE REDUCTASE PCR product | 1 | 10.57 | 0.88 |
| 197 | 205 | 00436MtBB | XIII. No homology PCR product | 0.01977 | 11.79 | 0.88 |
| 198 | 259 | 91315MtBC | XIII. No homology PCR product | 1 | 12.46 | 0.88 |
| 199 | 900 | 90815MtBB | X. Signal transduction B' REGULATORY SUBUNIT OF PROTEIN PHOSPHATASE PP2A PCR product | 1 | 10.05 | 0.88 |
| 200 | 189 | MtAMLs204 | PR1 PCR product | 0.00082414 | 10.16 | 0.88 |
| 201 | 78 | 93226MtBB | II. Cytoskeleton F-ACTIN CAPPING PROTEIN ALPHA SUBUNIT PCR product | 1 | 11.77 | 0.87 |
| 202 | 128 | 91827MtBC | X. Signal transduction and post-translational regulation PHOSPHOLIPASE A2 PRECURSOR PCR product | 1 | 11.03 | 0.87 |
| 203 | 143 | 90890MtBB | VI. Secondary metabolism and hormone metabolism LIPOXYGENASE PCR product | 0.20908 | 11.46 | 0.86 |
| 204 | 92 | 50501MtBB | XII.C. Unknown function PCR product | 1 | 11.27 | 0.86 |
| 205 | 376 | 00195MtBC | IV. Vesicular trafficking secretion and protein sorting DNAJ DOMAIN CONTAINING PROTEIN PCR product | 1 | 12.99 | 0.86 |
| 206 | 914 | 20253MtBC | V. Primary metabolism NITRATE REDUCTASE PCR product | 1 | 10.24 | 0.86 |
| 207 | 192 | 00474MtBC | XII.A. Defense and cell rescue MYROSINASE BINDING PROTEIN PCR product | 0.095982 | 11.60 | 0.86 |
| 208 | 178 | 93300MtBB | XIII. No homology PCR product | 5.5921e-06 | 10.29 | 0.86 |
| 209 | 556 | 40184MtBA | XII.C. Unknown function RING-H2 FINGER PROTEIN PCR product | 0.0044022 | 13.78 | 0.86 |
| 210 | 137 | 92106MtBC | XIII. No homology PCR product | 0.063154 | 10.44 | 0.86 |
| 211 | 1689 | 91707MtBC | XIII. No homology PCR product | 0.01798 | 10.01 | 0.85 |
| 212 | 159 | 90982MtBB | XIII. No homology PCR product | 1 | 11.84 | 0.85 |
| 213 | 605 | 91398MtBC | XII.C. Unknown function PCR product | 0.00041719 | 10.27 | 0.85 |
| 214 | 146 | 30543MtBC | XIII. No homology PCR product | 0.44613 | 11.98 | 0.85 |
| 215 | 257 | 91945MtBC | XII.C. Unknown function PCR product | 1 | 12.00 | 0.85 |
| 216 | 121 | 00743MtBC | X. Signal transduction and post-translational regulation PHOSPHOLIPASE D PCR product | 1 | 10.68 | 0.85 |
| 217 | 278 | 10584MtBB | X. Signal transduction and post-translational regulation PROTEIN KINASE DOMAIN CONTAINING PROTEIN PCR product | 0.46103 | 11.83 | 0.85 |
| 218 | 197 | 92039MtBC | XII.C. Unknown function PCR product | 0.47922 | 10.90 | 0.85 |
| 219 | 130 | 90415MtBA | II. Cytoskeleton MICROTUBULE-ASSOCIATED PROTEIN PCR product | 0.47457 | 12.11 | 0.84 |
| 220 | 142 | 50141MtBA | XIII. No homology PCR product | 1 | 12.00 | 0.84 |
| 221 | 75 | 91090MtBB | V. Primary metabolism 4-HYDROXYPHENYLPYRUVATE DIOXYGENASE PCR product | 1 | 10.41 | 0.84 |
| 222 | 307 | 90017MtBA | XIII. No homology PCR product | 0.4483 | 13.58 | 0.84 |
| 223 | 120 | 91831MtBC | V. Primary metabolism OXIDOREDUCTASE PCR product | 0.66721 | 12.86 | 0.84 |
| 224 | 1699 | 30577MtBC | XIII. No homology PCR product | 1 | 9.95 | 0.84 |
| 225 | 212 | 50863MtBC | XIII. No homology PCR product | 0.1618 | 11.96 | 0.84 |
| 226 | 323 | 91519MtBC | V. Primary metabolism ALCOHOL DEHYDROGENASE PCR product | 1 | 12.09 | 0.84 |
| 227 | 1049 | 10439MtBB | XII.C. Unknown function PCR product | 0.19494 | 10.12 | 0.83 |
| 228 | 129 | 30538MtBC | XIII. No homology TRANSMEMBRANE PROTEIN PCR product | 0.95834 | 11.47 | 0.83 |
| 229 | 920 | 90944MtBB | XII.C. Unknown function [PUTATIVE NODULIN] GLYCINE-RICH PROTEIN PRECURSOR PCR product | 0.0036453 | 10.02 | 0.83 |
| 230 | 275 | 30367MtBC | V. Primary metabolism GALACTINOL-RAFFINOSE GALACTOSYLTRANSFERASE PCR product | 0.00049316 | 10.98 | 0.83 |
| 231 | 119 | 11001MtBA | XII.C. Unknown function APETALA2 (AP2) DOMAIN CONTAINING PROTEIN PCR product | 1 | 11.55 | 0.83 |
| 232 | 186 | 00397MtBC | XII.C. Unknown function PCR product | 0.094587 | 11.03 | 0.83 |
| 233 | 1035 | 30542MtBB | XIII. No homology PCR product | 0.012966 | 10.29 | 0.83 |
| 234 | 319 | 90035MtBA | XIII. No homology PCR product | 0.32996 | 11.77 | 0.83 |
| 235 | 1019 | 91451MtBC | XIII. No homology PCR product | 1 | 10.40 | 0.83 |
| 236 | 264 | 90511MtBA | V. Primary metabolism ACID PHOSPHATASE PCR product | 1 | 12.35 | 0.83 |
| 237 | 1386 | MtAe84 | similar to RING zinc finger proteins PCR product | 0.0018796 | 10.85 | 0.83 |
| 238 | 274 | 50445MtBB | XII.C. Unknown function BOLA MORPHOGEN PCR product | 0.090645 | 11.31 | 0.83 |
| 239 | 549 | 90621MtBB | XIII. No homology PCR product | 7.6737e-07 | 10.58 | 0.82 |
| 240 | 384 | 93022MtBC | XIII. No homology DNAJ CONTAINING PROTEIN FRAGMENT PCR product | 0.1441 | 9.87 | 0.82 |
| 241 | 270 | 45048MtBB | XII.C. Unknown function P-LOOP CONTAINING PROTEIN PCR product | 0.00012323 | 11.44 | 0.81 |

Tabelle C.4: Fortsetzung der M-Liste des Experiments Nod 10, Bildauswertung AIM

| Rang | Rang in AIM-Liste | Identifikation | Funktionsannotation | p-Wert | A-Wert | M-Wert |
|---|---|---|---|---|---|---|
| 242 | 392 | 90503MtBA | VI. Secondary metabolism and hormone metabolism PHOSPHORIBOSYLANTHRANILATE TRANSFERASE PCR product | 1 | 10.47 | 0.81 |
| 243 | 172 | 00740MtBB | III. Membrane transport MEMBRANE TRANSPORTER PCR product | 0.99352 | 10.88 | 0.81 |
| 244 | 444 | 91469MtBC | XIII. No homology PCR product | 1 | 12.03 | 0.81 |
| 245 | 304 | 10396MtBC | XIII. No homology PCR product | 0.65383 | 11.42 | 0.81 |
| 246 | 297 | 92134MtBC | X. Signal transduction and post-translational regulation TRANSMEMBRANE PHOSPHOLIPASE C2 DOMAIN PCR product | 0.43672 | 11.77 | 0.81 |
| 247 | 216 | 10348MtBC | XII.C. Unknown function PCR product | 0.00080177 | 10.40 | 0.81 |
| 248 | 316 | 91382MtBC | XIII. No homology PCR product | 0.0058764 | 11.05 | 0.81 |
| 249 | 160 | 50212MtBA | X. Signal transduction and post-translational regulation SER/THR PROTEIN KINASE PCR product | 1 | 10.92 | 0.81 |
| 250 | 298 | 91281MtBC | XIII. No homology PCR product | 0.2107 | 12.49 | 0.80 |

# D Datenschema des AIM-Systems

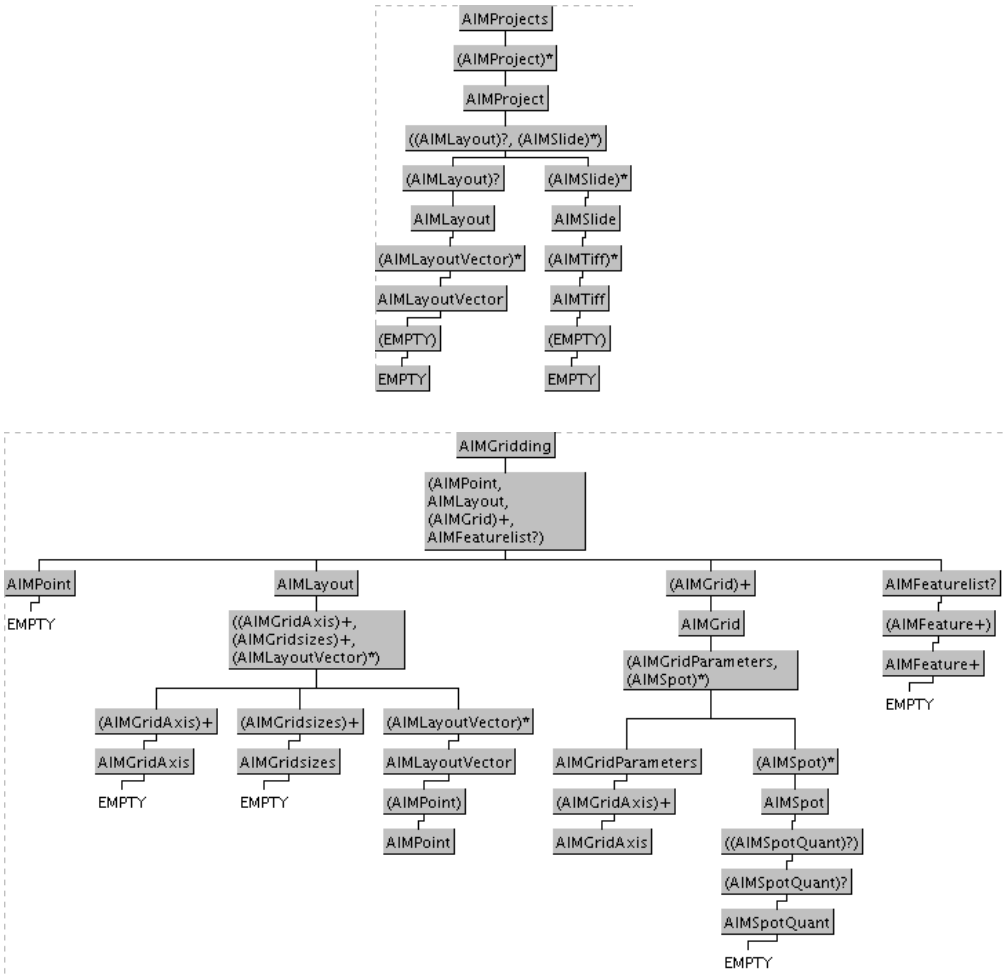



Abbildung D.1: Die Struktur der *Document Type Definitions* (DTDs), die das Datenschema des AIM-Systems definieren.

```
<!-- AIMGridding.dtd -->

<!ELEMENT AIMGridding ( AIMPoint,AIMLayout, (AIMGrid)+,
                        AIMFeaturelist? ) >
```

```
<!ATTLIST AIMGridding
slideid CDATA #REQUIRED
quantcmdline CDATA #IMPLIED
gridcmdline CDATA #IMPLIED
>

<!ELEMENT AIMLayout ( (AIMGridAxis)+, (AIMGridsizes)+,
              (AIMLayoutVector)* ) >
<!ATTLIST AIMLayout
projectid CDATA #REQUIRED
metagridcols CDATA #REQUIRED
metagridrows CDATA #REQUIRED
muperunit CDATA #REQUIRED
>

<!ELEMENT AIMLayoutVector (AIMPoint)>
<!ATTLIST AIMLayoutVector
gridfrom CDATA #REQUIRED
gridto CDATA #REQUIRED
>

<!ELEMENT AIMGrid (AIMGridParameters , (AIMSpot)*) >
<!ATTLIST AIMGrid
metarow CDATA #REQUIRED
metacol CDATA #REQUIRED
block CDATA #REQUIRED
oscore CDATA #REQUIRED
lscore CDATA #REQUIRED
gscore CDATA #REQUIRED
score CDATA #REQUIRED
flags CDATA #REQUIRED
>

<!ELEMENT AIMGridParameters ( AIMGridAxis )+ >
<!ATTLIST AIMGridParameters
columns CDATA #REQUIRED
oddcolumns CDATA #REQUIRED
rows CDATA #REQUIRED
oddrows CDATA #REQUIRED
originx CDATA #REQUIRED
originy CDATA #REQUIRED
naxes CDATA #REQUIRED
deltax CDATA #REQUIRED
deltay CDATA #REQUIRED
deltan CDATA #REQUIRED
>
```

```
<!ELEMENT AIMGridAxis EMPTY >
<!ATTLIST AIMGridAxis
axnum  CDATA #REQUIRED
axx CDATA #REQUIRED
axy CDATA #REQUIRED
>
<!ELEMENT AIMSpot ((AIMSpotQuant)?)>
<!ATTLIST AIMSpot
address CDATA #REQUIRED
x CDATA #REQUIRED
y CDATA #REQUIRED
r CDATA #IMPLIED
f CDATA #REQUIRED
>
<!ELEMENT AIMSpotQuant EMPTY>
<!ATTLIST AIMSpotQuant
FGMEAN0 CDATA #REQUIRED
FGMEAN1 CDATA #REQUIRED
FGMEDIAN0 CDATA #REQUIRED
FGMEDIAN1 CDATA #REQUIRED
BGMEAN0 CDATA #REQUIRED
BGMEAN1 CDATA #REQUIRED
BGMEDIAN0 CDATA #REQUIRED
BGMEDIAN1 CDATA #REQUIRED
MEANRATIO CDATA #REQUIRED
>
<!ELEMENT AIMPoint EMPTY>
<!ATTLIST AIMPoint
x CDATA #REQUIRED
y CDATA #REQUIRED
>

<!ELEMENT AIMGridsizes EMPTY>
<!ATTLIST AIMGridsizes
block CDATA #REQUIRED
rowscolumnsflags CDATA #REQUIRED
>

<!ELEMENT AIMFeaturelist (AIMFeature+)>
<!ELEMENT AIMFeature EMPTY>
<!ATTLIST AIMFeature
name CDATA #REQUIRED
geo CDATA #REQUIRED
datatype CDATA #REQUIRED
>
```

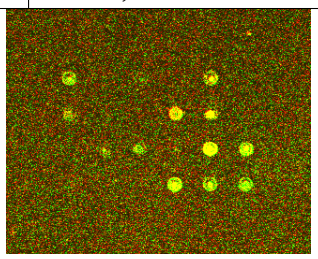
```
<!ELEMENT AIMtime     (EMPTY)>
<!ATTLIST AIMtime     year   CDATA #REQUIRED
                      month  CDATA #REQUIRED
                      day    CDATA #REQUIRED
                      hour   CDATA #REQUIRED
                      minute CDATA #REQUIRED
                      second CDATA #REQUIRED >



################

<!ELEMENT AIMProjects (AIMProject)*>
<!ELEMENT AIMProject (AIMLayout)? (AIMSlide)* >
<!ELEMENT AIMSlide (AIMTiff)*>
<!ELEMENT AIMTiff (EMPTY)>
<!ELEMENT AIMLayout (AIMLayoutVector)* >
<!ELEMENT AIMLayoutVector (EMPTY)>
<!ATTLIST AIMProject
  name CDATA #REQUIRED
  workdir CDATA #REQUIRED
  imagepath CDATA #REQUIRED
  n_channels CDATA #REQUIRED
  muperpixel CDATA #REQUIRED
  gridtype CDATA #REQUIRED
  columns CDATA #REQUIRED
  rows CDATA #REQUIRED
  columns_odd  CDATA #IMPLIED
  rows_odd CDATA #IMPLIED
  output_format CDATA #REQUIRED
  output_attribs CDATA #IMPLIED
  gld_options CDATA #REQUIRED
  grids_aligned CDATA #REQUIRED
  dark_edge CDATA #REQUIRED
  snrquantil CDATA #IMPLIED
>
<!ATTLIST AIMSlide
 id cdata #REQUIRED
>
<!ATTLIST AIMTiff
filename CDATA #REQUIRED
channel CDATA #REQUIRED
>
<!ATTLIST AIMLayout
metagridcols CDATA #REQUIRED
metagridrows CDATA #REQUIRED
muperunit CDATA #REQUIRED
```

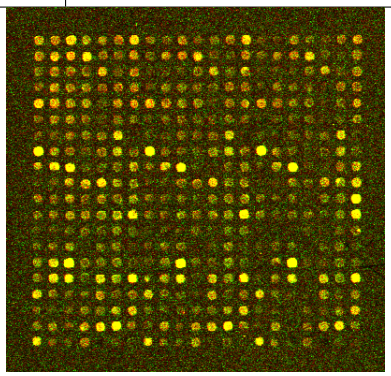
```
>
<!ATTLIST AIMLayoutVector
metagridcol_from CDATA #REQUIRED
metagridrow_from CDATA #REQUIRED
metagridcol_to CDATA #REQUIRED
metagridrow_to CDATA #REQUIRED
relx CDATA #REQUIRED
rely CDATA #REQUIRED
>
```

# E Bildbeispiele

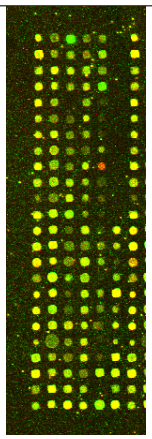
Auf den folgenden Seiten sind Bildausschnitte von je einem Messpunktgitter aus allen Teilstichproben zu finden. Dazu ist jeweils der untersuchte Organismus, die Quelle und das verwendete Bildaufnahmeprinzip angegeben. Für den Abdruck muss die Intensität der Bilder normiert werden. Zu jedem Beispielbild ist angegeben, welchen Anteil der Intensitätsskala von 65535 Grauwerten die dunkelsten 95% der Bildpixel in jedem Kanal einnehmen. Die Intensität ist so skaliert, dass eben diese 95% den gesamten Intensitätsbereich abdecken. Die restlichen 5% der Bildpixel haben gesättigte Intensität.

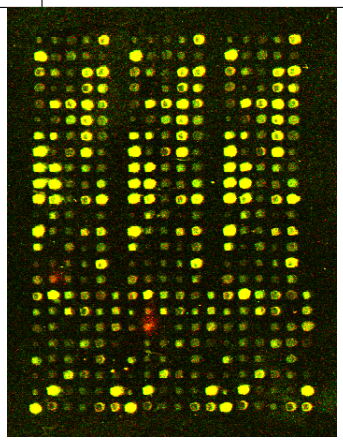
Die Beispielbilder sind repräsentativ für die geometrischen Eigenschaften der Messpunkte und ihrer Anordung in den jeweiligen Bildserien. Andere Eigenschaften wie Gesamthelligkeit und Verunreinigungsgrad sind von der einzelnen Hybridisierung und den Bildaufnahmeparametern abhängig und können daher innerhalb einer Serie variieren. Die Angaben zum Intensitätsbereich sind also ein grober Anhaltspunkt.

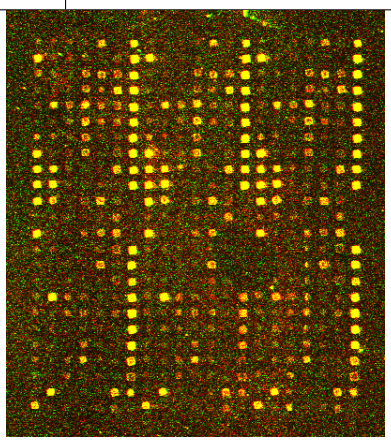
| Name | Organismus | Quelle | Aufn.[1] |
|---|---|---|---|
| Halle | ? | Anke Becker, Universität Bielefeld, Y. Gäbler und T. Nürnberger, Institut fuer Pflanzenbiochemie,Halle/Saale | AM |
|  | | | |
| Intensitätsbereiche: 2.80%,1.80% | | | |

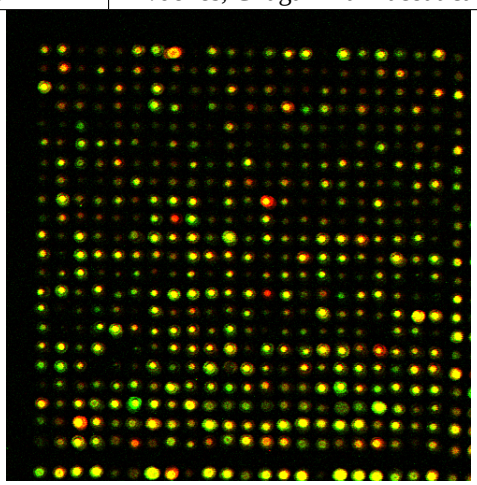
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| S. Meliloti Oligo | *Sinorhizobium meliloti* | Anke Becker, Universität Bielefeld | AM |
|  | | | |
| Intensitätsbereiche: 13.44%,10.35% | | | |

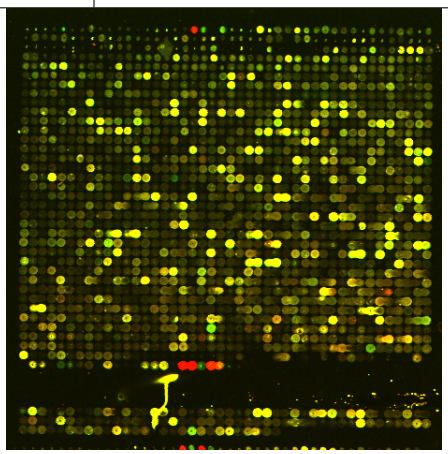
[1] Art der Bildaufnahme: AM=Abtastmikroskop, siehe auch 3.3.3

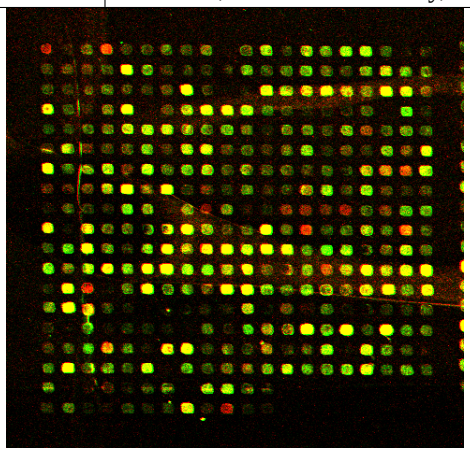
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| S. Meliloti PCR | *Sinorhizobium meliloti* | Anke Becker, Universität Bielefeld | AM |
|  | | | |
| Intensitätsbereiche: 10.08%,6.98% | | | |

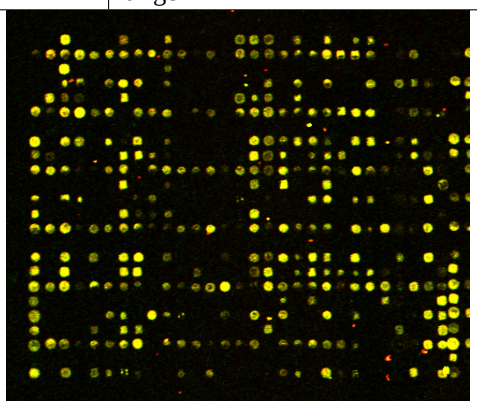
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Mt6kRIT | *Medicago truncatula* | Helge Küster, Universität Bielefeld | AM |
|  | | | |
| Intensitätsbereiche: 12.17%,10.46% | | | |

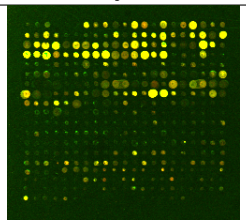
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Mt8kRIT | *Medicago truncatula* | Helge Küster, Universität Bielefeld | AM |
|  | | | |
| Intensitätsbereiche: 4.75%,4.04% | | | |

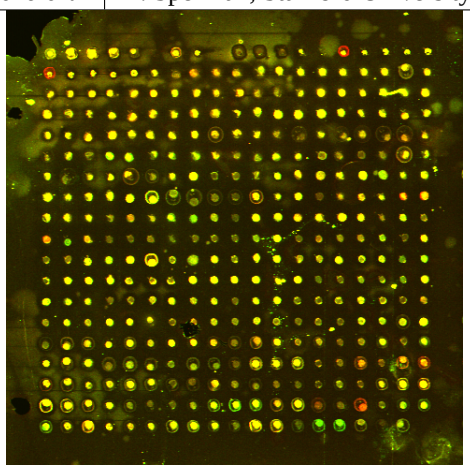
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Chugai | Mensch | M. Jones, Chugai Pharmaceuticals, Tokyo | AM |
|  | | | |
| Intensitätsbereiche: 14.45%,9.84% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Contest | *S. cerevisiae* | CAMDA 2000 Contest Data Set `http://www.camda.duke.edu` | AM |
|  | | | |
| Intensitätsbereiche: 7.59%,6.05% | | | |

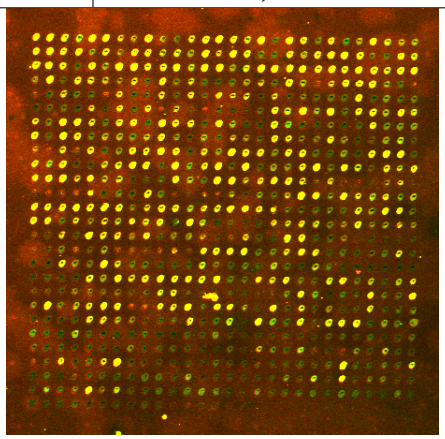
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| SMD | ? | M. Cherry/ T. Boussard, Stanford Microarray Database, Stanford University, USA | AM |
|  | | | |
| Intensitätsbereiche: 10.08%,18.83% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| TLG Human | Mensch | J. Landgrebe, Georg-August-Universität Göttingen | AM |
|  | | | |
| Intensitätsbereiche: 23.64%,15.28% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Apo A1 | Maus | M. Callow, Lawrence Livermore National Laboratory, Livermore, CA, USA | AM |
|  | | | |
| Intensitätsbereiche: 19.01%,11.36% | | | |

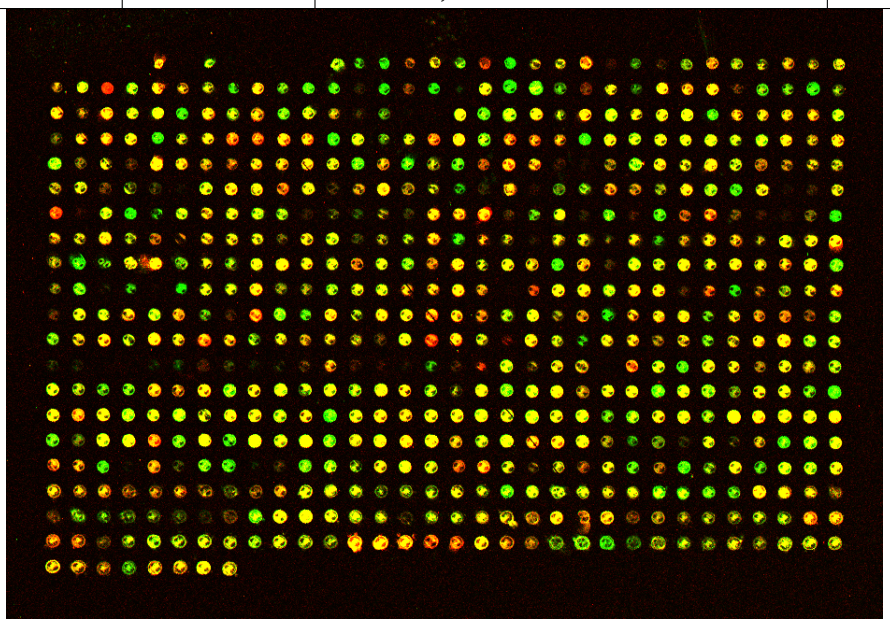
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Sporman | *Vibrio cholera* | A. Sporman, Stanford University, USA | AM |
|  | | | |
| Intensitätsbereiche: .58%,.61% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| WNT | Mensch | Shixia Huang, Sloan Kettering Memorial Cancer Center, New York | AM |

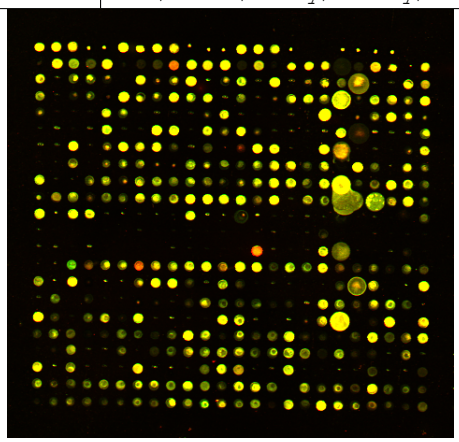


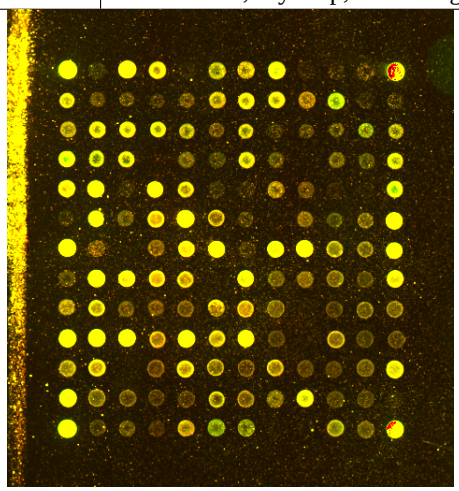
Intensitätsbereiche: 1.08%,1.40%

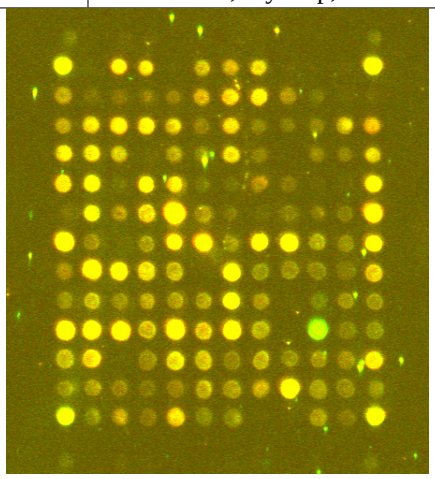
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| NMHy | Maus | Agnes Viale, Sloan Kettering Memorial Cancer Center, New York | AM |

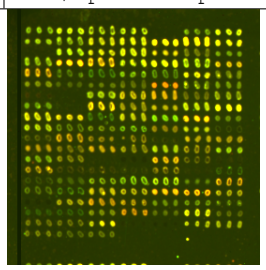


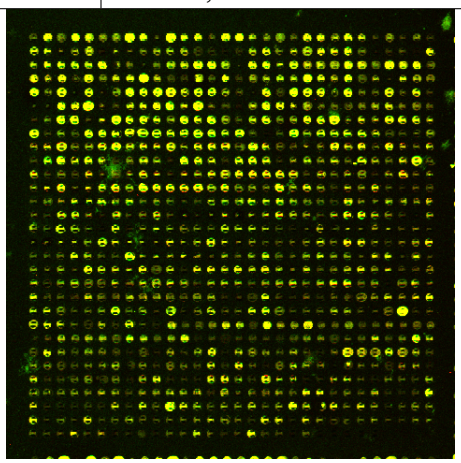
Intensitätsbereiche: 2.05%,4.92%

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Swirl | *Drosophila* | T. Speed,University of California at Berkeley, „Short Course in Microarray Data Analysis“http://www.stat.berkeley.edu/users/terry/zarray/Course/ | AM |
|  | | | |
| Intensitätsbereiche: 14.59%,21.44% | | | |

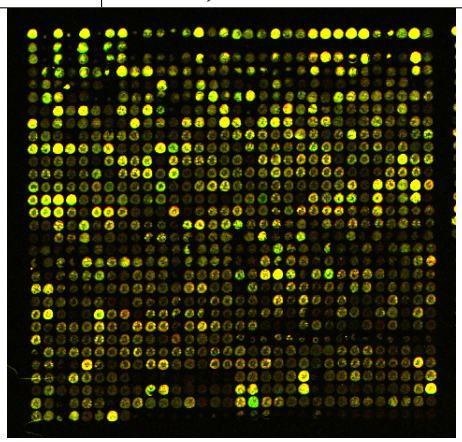
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| FC002 | *Drosophila* | David Kreil, FlyChip, Cambridge, UK | AM |
|  | | | |
| Intensitätsbereiche: 10.74%,12.90% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| FC002 G | *Drosophila* | David Kreil, FlyChip, Cambridge, UK | CCD |
|  | | | |
| Intensitätsbereiche: 6.41%,8.96% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Spot | Mensch | Jain et. Al [53] `http://jainlab.ucsf.edu/Spot-Examples.zip` | CCD |
|  | | | |
| Intensitätsbereiche: 6.24%,3.68% | | | |

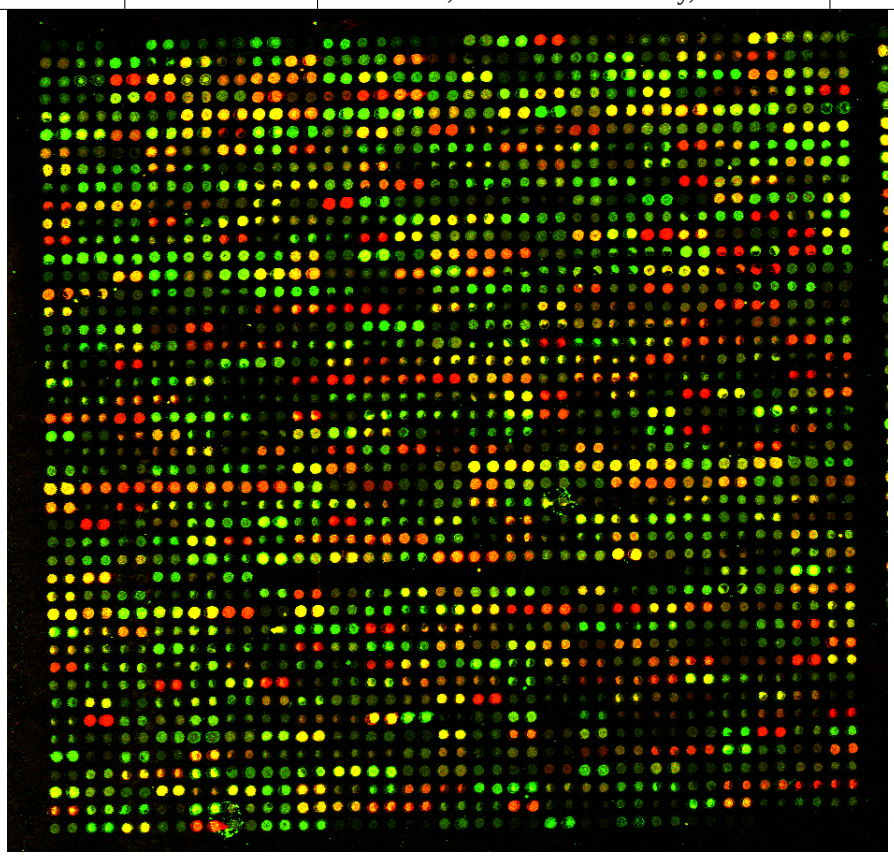
| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| marounC | Hefe | M. Beyrouthy, Florida State University, Tallahassee, USA | AM |
|  | | | |
| Intensitätsbereiche: 5.29%,4.26% | | | |

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| marounB | Hefe | M. Beyrouthy, Florida State University, Tallahassee, USA | AM |



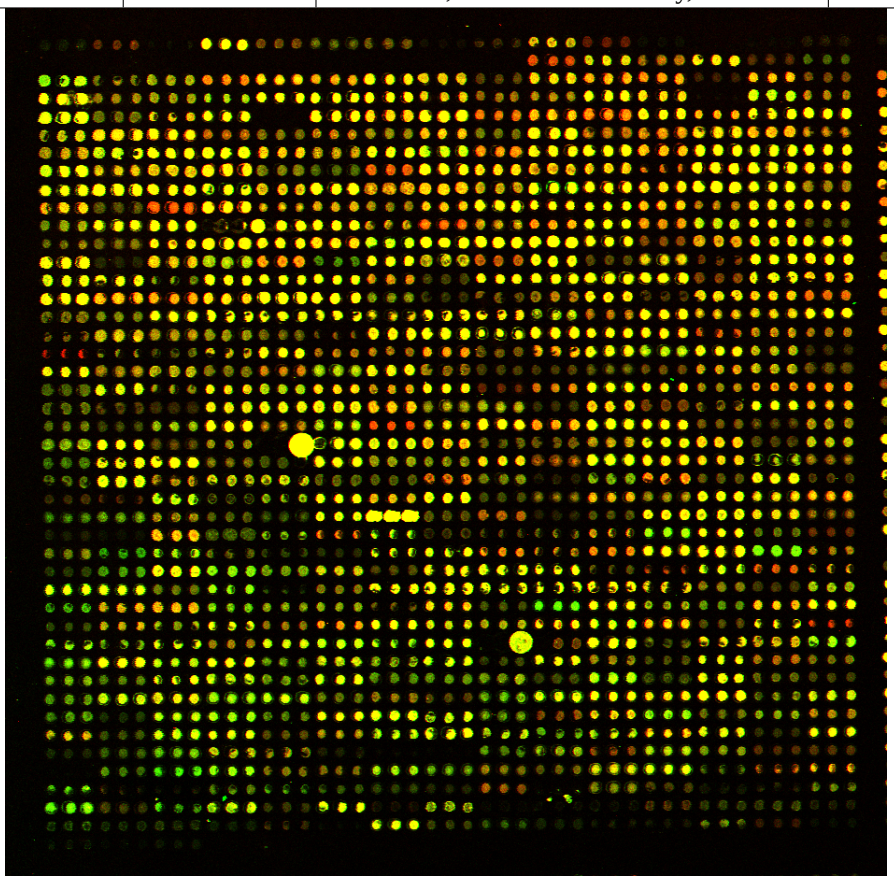
Intensitätsbereiche: 6.43%,6.94%

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Zmdb 605 | Mais | V. Brendel, Iowa State University, USA | AM |



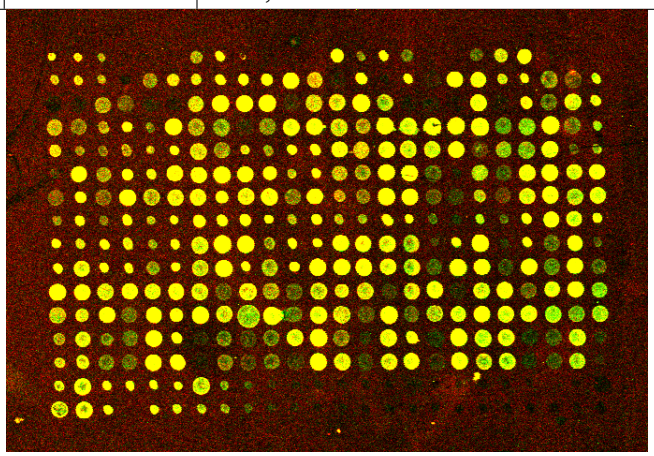
Intensitätsbereiche: 85.18%,20.26%

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Zmdb 606 | Mais | V. Brendel, Iowa State University, USA | AM |

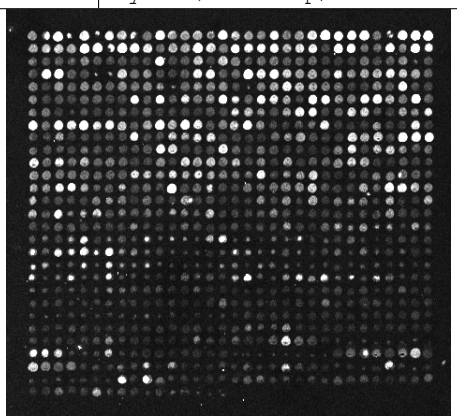


Intensitätsbereiche: 44.75%,33.18%

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Pine | Pinie | T. Gaasterland, Rockefeller University, New York, USA | AM |



Intensitätsbereiche: 8.21%,9.10%

| Name | Organismus | Quelle | Aufn. |
|---|---|---|---|
| Microzip2 | ? | Yu Luo, University of California, Riverside, USA, http://www.cs.ucr.edu/~yuluo/MicroZip/ | AM |
|  | | | |
| Intensitätsbereich: 3.41% | | | |

# Literaturverzeichnis


[1] ADAMS, R. und L. BISCHOF: *Seeded region growing*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6):641–647, 1994.

[2] ALIZADEH ET. AL. : *Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling*. Nature, 403, 2000.

[3] ALTMANN, C. R., E. BELL, A. SCZYRBA, J. PUN, S. BEKIRANOV, T. GAASTERLAND und A. H. BRIVANLOU: *Microarray-Based Analysis of Early Development in* Xenopus laevis. Developmental Biology, Vol. 236, No. 1, Aug 2001, pp. 64-75, 236(1):pp. 64–75, Aug 2001.

[4] AMINI, A. A., S. TEHRANI und T. E. WEYMOUTH: *Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints*. In: *Proceedings, Second International Conference on Computer Vision*, Seiten 95–99, 1988.

[5] ANGULO, J. und J. SERRA: *Automatic analysis of DNA microarray images using mathematical morphology*. Bioinformatics, 19(5):553–562, 2003.

[6] ARYA, S., D. M. MOUNT, N. S. NETANYAHU, R. SILVERMAN und A. Y. WU: *An optimal algorithm for approximate nearest neighbor searching fixed dimensions*. Journal of the ACM, 45(6):891–923, 1998.

[7] AVERY, O. T., C. M. MACLEOD und M. MCCARTY: *Studies on the chemical nature of the substance inducing transformation of pneumonococcal types. Induction of transformation by a deoxyribo-nucleic acid fraction isolated from pneumococcus type III.* J. Exp. Med., 79:137–158, 1944.

[8] BARTELS, D., L. KRAUSE, B. LINKE und O. RUPP: *EMMA - EST Meets MicroArray*. Technischer Bericht, Center for Genome Research, Bielefeld University, Germany, 2002.

[9] BERG, MARK DE: *Computational geometry : algorithms and applications*. Springer, 2. Auflage, 2000.

[10] BERMAN, H. M., J. WESTBROOK, Z. FENG, G. GILLILAND, T. N. BHAT, H. WEISSIG, I. N. SHINDYALOV und P. E. BOURNE: *The Protein Data Bank.* Nucleic Acids Research, 28:235–242, 2000.

[11] BILBAN, M.: *Normalizing DNA microarray data*. Current Issues in Molecular Biology, 34(1):48–57, 2002.

[12] BLAKE, A. und M. ISARD: *Active contours : the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion.* Springer, 2000.

[13] BOGGS, P. T. und J. W. TOLLE: *Sequential quadratic programming*. Acta numerica, Seiten 1–51, 1995.

[14] BOZINOV, D. und J. RAHNENFÜHRER: *Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering.* Bioinformatics, 18(5):747–756, 2002.

[15] BRAZMA, A., P. HINGAMP, J. QUACKENBUSH, G SHERLOCK, P SPELLMAN, C STOECKERT, J AACH, W ANSORGE, C A BALL, H C CAUSTON, T GAASTERLAND, P GLENISSON, F C P HOLSTEGE, I F KIM, V MARKOWITZ, J C MATESE, H PARKINSON, A ROBINSON, U SARKANS, S SCHULZE-KREMER, J STEWART, R TAYLOR, J VILO und M VINGRON: *Minimum information about a microarray experiment (MIAME)-toward standards for microarray data*. Nature Genetics, 29(4):365 – 371, 2001.

[16] BRAZMA, A., H. PARKINSON, U. SARKANS, SHOJATALAB M., VILO J., ABEYGUNAWARDENA N., HOLLOWAY E., KAPUSHESKY M., KEMMEREN P., LARA G. G., OEZCIMEN A., ROCCA-SERRA P. und SANSONE S-A.: *ArrayExpress-a public repository for microarray gene expression data at the EBI*. Nucleic Acids Research, 31(1):68–71, 2003.

[17] BRETZ, F., J. LANDGREBE und E. BRUNNER: *Efficient Design and Analysis of Two Colour Factorial Microarray Experiments*. Biostatistics, eingereichtes Manuskript, 2003.

[18] BRIGGER, P., J. HOEG und M. UNSER: *B-Spline Snakes: A Flexible Tool for Parametric Contour Detection*. IEEE Transactions on Image Processing, 9(9):1484–1496, September 2000.

[19] BRIGNAC, S. J., R. GANGADHARAN und M. MCMAHON: *A Proximal CCD Imaging System for High-Throughput Detection of Microarray-Based Assays*. IEEE Engineering in Medicine and Biology Magazine, 18:120–2, 1999.

[20] BRÄNDLE, N.: *Robust Analysis of Spot Array Images*. Technischer Bericht PRIP-TR-70, Technische Universität Wien, 2002.

[21] BROWN., C. S., P.C. GOODWIN und P.K. SORGER: *Image metrics in the statistical analysis of DNA microarray data*. PNAS, 98(16):8944–8949, 2001.

[22] BROWN, P. und M. EISEN: *The MGuide. Version 2.0 The Brown Lab's complete guide to microarraying for the molecular biologist.* `http://cmgm.stanford.edu/pbrown/mguide/index.html`, 2000.

[23] BUHLER, J., T. IDEKER und D. HAYNOR: *Dapple: Improved Techiques for Finding Spots on DNA Microarrays*. Technischer Bericht UWTR 2000-08-05, University of Washington, 2000.

[24] BURGES, C. J. C.: *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2):121–167, 1998.

[25] CARSTENSEN, J. M.: *An Active Lattice Model In A Bayesian Framework*. Computer Vision and Image Understanding, 63:380–387, 1996.

[26] CARSTENSEN, J. M. und K. HARTELIUS: *Bayesian Grid Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2):162–173, 2003.

[27] CHANG, C. und C. LIN: *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[28] CHEN, Y., E. R. DOUGHERTY und M. L. BITTNER: *Ratio-Based Decisions and the Quantitative Analysis of cDNA Microarray Images*. Journal of Biomedical Optics, 2(4):364–374, 1997.

[29] CHOU, P. B. und C. M. BROWN: *The theory and practice of Bayesian image labeling*. International Journal of Computer Vision, 4(3):185–210, 1990.

[30] CHOU, P. B., P. R. COOPER, M. J. SWAIN, C. M. BROWN und L. E. WIXSON: *Probabilistic Network Inference for Cooperative High and Low Level Vision*. In: CHELLAPPA, RAMA (Herausgeber): *Markov random fields*, Seiten 211–243. Academic Pr., 1993.

[31] CHO, Y., J. FERNANDES, S. KIM und V. WALBOT: *Gene-expression profile comparisons distinguish seven organs of maize*. Genome Biology, 3(9):0045.1–0045.16, 2002.

[32] CLARK, T., C. SUGNET und M. ARES: *Genomewide analysis of mRNA processing in yeast using splicing-specific microarrays*. Science, 296:907–910, 2002.

[33] DAI, H., M. MEYER, S. STEPANIANTS, M. ZIMAN und R. STOUGHTON: *Use of hybridization kinetics for differentiating specific from non-specific binding to oligonucleotide microarrays*. Nucleic Acids Research, 30(16):e86, 2002.

[34] DARNELL, JAMES, HARVEY LODISH und DAVID BALTIMORE: *Molecular cell biology*. Freeman, 4. Auflage, 1999.

[35] DERISI, J. L., V. R. IYER und P. O. BROWN: *Exploring the metabolic and genetic control of gene expression on a genomic scale*. Science, 278:680–686, 1997.

[36] DONDRUP, M.: *Dokumentation: Verfahren zur Datennormalisierung mit EMMA*. Technischer Bericht, Universität Bielefeld,Fakultät für Biologie, Zentrum für Genomforschung, 2002.

[37] EDGAR, R., M. DOMRACHEV und A. E. LASH: *Gene Expression Omnibus: NCBI gene expression and hybridization array data repository*. Nucleic Acids Research, 30(1):207–210, 2002.

[38] EISEN, M.: *Scanalyze*. `http://www.microarrays.org/software/ScanAlyzeDoc.pdf`, 1999.

[39] EISEN, M. und P. BROWN: *DNA arrays for analysis of gene expression*. Methods in Enzymology, 303:179–205, 1999.

[40] FINK, G. A.: *Developing HMM-based Recognizers with ESMERALDA*. In: MATOUŠEK, VÁCLAV, PAVEL MAUTNER, JANA OCELÍKOVÁ und PETR SOJKA (Herausgeber): *Lecture Notes in Artificial Intelligence*, Band 1692, Seiten 229–234, Berlin Heidelberg, 1999. Springer.

[41] FODOR, S. P., R. P. RAVA, X. C. HUANG, A. C. PEASE, C. P. HOLMES und C. L. ADAMS: *Multiplexed biochemical assays with biological chips*. Nature, 364(6437):555–556, 1993.

[42] GEMAN, S. und D. GEMAN: *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(6):721–742, 1984.

[43] GLASBEY, C. A. und P. GHAZAL: *Combinatorial image analysis of DNA microarray features*. Bioinformatics, 19(2):194–203, 2003.

[44] GOLLUB, J., C. A. BALL, G. BINKLEY, J. DEMETER, D. B. FINKELSTEIN, J. M. HEBERT, T. HERNANDEZ-BOUSSARD, H. JIN, M. KALOPER, J. C. MATESE, M. SCHROEDER, P. O. BROWN, D. BOTSTEIN und SHERLOCK G.: *The Stanford Microarray Database: data access and quality assessment tools*. Nucleic Acids Research, 31(1):94–96, Jan 2003.

[45] HÄRDLE, W. und M. STEIGER: *Optimal Median Smoothing*. Technischer Bericht RePEc:wop:humbsf:1994-15, Sonderforschungsbereich 373 / Humboldt Universität Berlin, 1994.

[46] HIRATA, R., J. BARRERA und R. F. HASHIMOTO: *Segmentation of Microarray Images by Mathematical Morphology*. Real-Time Imaging, 8(6):491–505, 2002.

[47] HOLLOWAY, A. J., R. K. VAN LAAR, R. W. TOTHILL und D. D. L BOWTELL: *Options available from start to finish for obtaining data from DNA microarrays II*. Supplement to Nature Genetics, 32:481–489, Dezember 2002.

[48] HUANG, T., G. YANG und G. TANG: *A fast two-dimensional median filtering algorithm*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 27(1):13–18, February 1979.

[49] HUGHES, T.R., M. J. MARTON, A. R. JONES, C. J. ROBERTS, R. STOUGHTON, C. D. ARMOUR, H. A. BENNETT, E. COFFEY, H. DAI, Y. D. HE, M. J. KIDD, A. M. KING, M. R. MEYER, D. SLADE, P. Y. LUM, S. B. STEPANIANTS, D. D. SHOEMAKER, D. GACHOTTE, K. CHAKRABURTTY, J. SIMON, M. BARD und S. H. FRIEND: *Functional discovery via a compendium of expression profiles*. Cell, 102(1):109–126, 2000.

[50] HU, G. K., S. J. MADORE, B. MOLDOVER, T. JATKOE, D. BALABAN, J. THOMAS und Y. WANG: *Predicting Splice Variant from DNA Chip Expression Data*. Genome Res., 11:1237–1245, 2001.

[51] JACOB, F. und J. MONOD: *Genetic regulatory mechanisms in the synthesis of proteins*. Journal of Molecular Biology, 3:318–356, 1961.

[52] JÄHNE, BERND: *Digitale Bildverarbeitung*. Springer Verlag Berlin, Heidelberg, New York, 4. Auflage, 1997.

[53] JAIN, A.N., T.A. TOKUYASU, A.M. SNIJDERS, R. SEGRAVES, D.G. ALBERTSON und D. PINKEL: *Fully Automatic Quantification of Microarray Image Data*. Genome Res., 12(2):325–332, 2002.

[54] JANG, EL-KWAE und CHOI: *Shaking snakes using color edge for contour extraction*. In: *International Conference on Image Processing*. IEEE, 2002.

[55] JONES, M. C., J. S. MARRON und S. J. SHEATHER: *Progress in Data-Based Bandwith Selection for Kernel Density Estimation*. Computational Statistics, 11:337–381, 1996.

[56] JUNG, HO-YOUL und HWAN-GUE CHO: *An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis*. Bioinformatics, 18(90002):141–151, 2002.

[57] KASS, M., A. WITKIN und D. TERZOPOULOS.: *Snakes: Active Contour Models*. International Journal of Computer Vision, Seiten 321–331, 1988.

[58] KATZER, M.: *Automatische Auswertung von Microarraybildern zur Expressionsanalyse*. Diplomarbeit, Universität Bielefeld, Technische Fakultät, Sept. 2000.

[59] KATZER, M., F. KUMMERT und G. SAGERER: *Robust Automatic Microarray Image Analysis*. In: *Proceedings of the International Conference on Bioinformatics: North-South Networking*, Bangkok, 2002.

[60] KATZER, M., F. KUMMERT und G. SAGERER: *A Markov Random Field Model of Microarray Gridding*. In: *Proc. 18th ACM Symposium on Applied Computing (SAC)*, Seiten 72–77. ACM, 2003.

[61] KATZER, M., F. KUMMERT und G. SAGERER: *Methods for Automatic Microarray Image Segmentation*. IEEE Transactions on Nano-Bioscience, 2(4):202–214, 2003.

[62] KELLAM, P.: *Microarray gene expression database: progress towards an international repository of gene expression data*. Genome Biology, 2(5):4011.1–4011.3, 2001.

[63] KERR, M. K. und G. A. CHURCHILL: *Experimental design for gene expression microarrays*. Biostatistics, 2(2):183–201, 2001.

[64] KINDERMANN, R. und J. L. SNELL: *Markov random fields and their applications*. Contemporary Mathematics 1. American Mathematical Society, Providence, RI, 1980.

[65] KINGMAN, J. F. C.: *Poisson processes*. Clarendon Press, 1993.

[66] KNIPPERS, R.: *Molekulare Genetik*. Georg Thieme Verlag, 6. Auflage, 1995.

[67] KOOPERBERG, C., T. G. FAZZIO, J. J. DELROW und T. TSUKIYAMA: *Improved Background Correction for Spotted DNA Microarrays*. Journal of Computational Biology, 2002.

[68] KRENGEL, U.: *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Vieweg/Bertelsmann, 1991.

[69] KÜSTER, H., N. HOHNJEC, F. KRAJINSKI, F. E. YAHYAOUI, K. MANTHEY, J. GOUZY, M. DONDRUP, F. MEYER, J. KALINOWSKI, L. BRECHENMACHER, D. VAN TUINEN, V. GIANINAZZI-PEARSON, A. PÜHLER, P. GAMAS und A. BECKER: *Construction and validation of comprehensive cDNA-based macro- and microarrays to explore root endosymbioses in the model legume Medicago truncatula*. Eingereichtes Manuskript, 2003.

[70] KUMMERT, F., G. A. FINK und G. SAGERER: *A hybrid speech recognizer combining HMMs and polynomial classification*. In: *International Conference on Spoken Language Processing*, Band 3, Seiten 814–817, Beijing, China, 2000.

[71] LAWRENCE, N. D., M. MILO, M. NIRANJAN, P. RASHBASS und S. SOULLIER: *Reducing the variability in cDNA microarray image processing by Bayesian inference*. Bioinformatics, 2003. In Press.

[72] LEVENBERG, K.: *A method for the solution of certain nonlinear problems in least squares*. Quarterly Journal of Applied Mathematics, (2):164–168, 1944.

[73] LEWIN, B.: *Genes VI*. Oxford Univ. Press, 1997.

[74] LI, S. Z.: *Markov random field modeling in image analysis*. Computer science workbench. Springer, Tokyo, 2. Auflage, 2001.

[75] MANGALAM, H., J. STEWART, J. ZHOU, K. SCHLAUCH, M. WAUGH, G. CHEN, A. D. FARMER, G. COLELLO und J. W. WELLER: *GeneX: An Open Source gene expression database and integrated tool set*. IBM Systems Journal, 40(2), 2001.

[76] MARQUARDT, D. W.: *An algorithm for least squares estimation of nonlinear parameters*. Journal of the Society for Industrial and Applied Mathematics, 11(2):431–441, 1963.

[77] MCHARDY, A. C., A. PÜHLER, J. KALINOWSKI und F. MEYER: *Comparing expression level-dependent features in codon usage with protein abundance: An analysis of 'predictive proteomics'*. Proteomics, 2003. im Druck.

[78] NATTKEMPER, T. W., H. RITTER und W. SCHUBERT: *A Neural Classificator Enabling High-Throughput Topological Analysis of Lymphocytes in Tissue Sections*. IEEE Transactions on Information Technology in Biomedicine, 5(2):138–149, 2001.

[79] NICK, T.: *Verfahren zur Konstruktion von Sonden für Oligonukleotid-Arrays, Eine Einführung in die DNA-Chiptechnologie*. Diplomarbeit, Universität Bielefeld, 1999.

[80] NOVAK, S. YU: *On the Mode of an Unknown Probability Distribution*. Theory of Probability and Its Applications, 44(1):109–113, 2000.

[81] OKAMOTO, T., T. SUZUKI und N. YAMAMOTO: *Microarray fabrication with covalent attachment of DNA using Bubble Jet technology*. Nature Biotechnology, 18:438–441, 2000.

[82] OTSU, N.: *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man, and Cybernetics, 9:62–66, 1979.

[83] PEVZNER, P. A.: *Computational Molecular Biology: an Algorithmic Approach*. MIT Press, 2000.

[84] PREPARATA, F. P. und M. I. SHAMOS: *Computational Geometry: An Introduction*. Springer Verlag, New York, Berlin, Heidelberg, Tokyo, 1995.

[85] RAUFER, D. und B. SOMMER: *Der Partition-Heap-Median*. Projektbericht aus dem Seminar „Anwendungsorientierte Bildverarbeitung“ (G. Fink, M. Hanheide), 2003.

[86] REXHEPI, A., A. ROSENFELD, F. MOKHTARIAN und Z. DURIC: *Adaptation in Active Contour Models*. In: *Proc. IASTED International Conference on Signal and Image Processing*, 2003.

[87] SAEED, A. I., V. SHAROV und J. WHITE: *TM4: A Free, Open-Source System for Microarray Data Management and Analysis*. Biotechniques, 34(2):374–379, 2003.

[88] SCHENA, M. (Herausgeber): *DNA Microarrays*. Oxford University Press, 1999.

[89] SCHENA, M., D. SHALON und P. O. BROWN: *Quantitative monitoring of gene expression patterns with a complementary DNA microarray*. Science, 270:467–470, 1995.

[90] SCHERF, U., D.T. ROSS, M. WALTHAM, L.H. SMITH, J. K. LEE, L. TANABE AN K. W. KOHN, W. C. REINHOLD, T. G. MYERS, D. T. ANDREWS, D. A. SCUDIERO, M. B. EISEN, E. A. SAUSVILLE, Y. POMMIER, D. BOTSTEIN, P. O. BROWN und J. N. WEINSTEIN: *A gene expression database for the molecular pharmacology of cancer*. Nature Genetics, 24(3):236–244, 2000.

[91] SCHÜRMANN, J.: *Pattern Classification*. Wiley, 1996.

[92] SCHWARZ, H. R.: *Numerische Mathematik*. B. G. Teubner, 3. Auflage, 1993.

[93] SIEGRIST, K.: *Virtual Laboratories in Probability and Statistics*. `http://www.math.uah.edu/stat/`, 2003.

[94] SINGH-GASSON, S.: *Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array*. Nature Biotechnology, 17:974–978, 1999.

[95] SMYTH, G. K., Y. H. YANG und T. P. SPEED: *Functional Genomics: Methods and Protocols*. In: BROWNSTEIN, M J. und A. B. KHODURSKY (Herausgeber): *Statistical Issues in cDNA Microarray Data Analysis*, Methods in Molecular Biology. Humana Press, Totowa, NJ, 2002.

[96] SOUTHERN, E. M.: *Detection of Specific Sequences Among DNA Fragments Separated by Gel Electrophoresis*. J. Mol. Biol, 98:503–517, 1975.

[97] SPELLMAN, P. T., M. MILLER, J. STEWART, C. TROUP, U. SARKANS, D. REK BERNHART, G. SHERLOCK, C. BALL, M. LEPAGE, M. SWIATEK, W. L. MARKS, J. GONCALVES, S. MARKEL, D. IORDAN, M. SHOJATALAB, A. PIZARRO, J. WHITE, R. HUBLEY, E. DEUTSCH, M. SENGER, B. J. ARONOW, A. ROBINSON, D. BASSETT, C. J. JR STOECKERT und A. BRAZMA: *Design and implementation of microarray gene expression markup language (MAGE-ML)*. Genome Biology, 3(9), 2002.

[98] SPELLMAN, P. T., G. SHERLOCK, M. Q. ZHANG, V. R. IYER, K. ANDERS, M. B. EISEN, P. O. BROWN, D. BOTSTEIN und B. FUTCHER: *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast* Saccharomyces cerevisiae *by Microarray Hybridization*. Molecular Biology of the Cell, 9:3273–3297, 1998.

[99] SPRENT, P. und N. C. SMEETON: *Applied nonparametric statistical methods*. Chapman & Hall / CRC, 3. Auflage, 2001.

[100] STEINFATH, M., W. WRUCK und H. SEIDEL: *Automated image analysis for array hybridization experiments*. Bioinformatics, 2001, Vol. 17, T. 7, S. 634-641, 2001.

[101] STOECKERT, CHRISTIAN J. JR., HELEN C. CAUSTON und CATHERINE A. BALL: *Microarray databases: standards and ontologies*. Supplement to Nature Genetics, 32(4):469–473, Dezember 2002.

[102] TURK, M. und A. PENTLAND: *Eigenfaces for Recognition*. Journal of Cognitive Neuro Science, 3(1):71–86, 1991.

[103] VESANEN, P.: *Calibration-free Methods in Segmentation of cDNA Microarray Images*. Proc. IS&T/SPIE 12th Symposium on Electronic Imaging Science and Technology, (No. 4667):291–302, 2002.

[104] WANG, X., S. GHOSH und S. GUO: *Quantitative quality control in microarray image processing and data acquisition*. Nucleic Acids Research, 29(15):e75, 2001.

[105] WILLIAMS, D. und M. SHAH: *A Fast Algorithm for Active Contours and Curvature Estimation*. Computer Vision, Graphics and Image Processing, 55(1):14–26, 1992.

[106] WILSON, M., J. DERISI, H. KRISTENSEN, P. IMBODEN, S. RANE, P. O. BROWN und G. K. SCHOOLNIK: *Exploring drug-induced alterations in gene expression in* Mycobacterium tuberculosis *by microarray hybridization*. Proceedings of the National Academy of Science USA, 96(22):12833–12838, October 1999.

[107] YANG, Y. H., M. J. BUCKLEY, S. DUDOIT und T. P. SPEED: *Comparison of Methods for Image Analysis on cDNA Microarray Data*. Journal of Computational and Graphical Statistics, 11:108–136, 2002.

[108] YANG, Y. H., S. DUDOIT, P. LUU, D. M. LIN, V. PENG, J. NGAI und T. P. SPEED: *Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation*. Nucleic Acids Research, 30:e15, 2002.

[109] YANG, Y. H. und T. SPEED: *Design issues for cDNA microarray experiments*. Nature Reviews Genetics, 3(8):579–588, 2002.

[110] YEAKLEY, J. M.: *Profiling alternative splicing on fiber-optic arrays*. Nature Biotechnology, 20:353–358, 2002.

# Index


A-Wert, 113
Adressierung, 34
AIM, 48
Aktives Konturmodell, 86

cDNA, 144
Codon, 13

Denaturierung, 7
Differentiation, 3, 13, 32
Dye-Swap, 31

Eigenspot, 84
Eukaryoten, 3
Exon, 10
Expressionsprofil, 32

Fluoreszenz, 26

Gen, 1, 9, 12
Genexpression, 9
Genom, 9
Gitterhypothese, 70
Gittersegmentierung, 34
    Korrektheit, 109
Grid, 24
Gridding, 34

Hybridisierung, 7, 28
    konkurrierende, 28, 101

Intron, 10

kd-Baum, 65
Kreuzhybridisierung, 7

M-Wert, 113
Mann-Whitney-Test
    Definition der Teststatistik, 103
    für Merkmalsberechnung, 95
Markov-Zufallsfeld
    Definition, 71
    Energieminimierung, 77
    zur Gittersegmentierung, 74
Messpunkt, 24
Methylierung, 8
Modus, 56
Mutation, 5

Nachbarschaft
    horizontal/vertikal, 59
    MZF-Konfiguration, 78
Normalisierung, 30

Objekt
    Definition, 63
    Erzeugung, 63
ORF, 13

PCR, 15, 143
Poisson-Prozess, 145
Poisson-Verteilung, 145
Polymerase-Kettenreaktion, *siehe* PCR
Projekt, 48
Prokaryoten, 3

Quantil, 56

Reassoziation, 7
Region, 51
Replikate
    biologische, 31
    technische, 31

Scanalyze, 44, 107
schwarzes Loch, 31, 102
Seeded region growing, 45
Signalsegmentierung, 34
Simulated Annealing, 77
Slide, 48
Southern-Blot, 18
Spot, *siehe* Messpunkt
SRG, *siehe* Seeded region growing
Stabilität
    HCF-Algorithmus, 79
    MZF-Parameter, 77

t-Test, 32, 113
Transkription, 10
    reverse, 15, 18, 144

Verschiebungskorrektur, 64