Präprozessierungs-Algorithmen für Affymetrix Microarrays

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Frau Dipl.-Biol. Claudia Döring
aus Frankfurt am Main

Frankfurt (2009)
(D30)

vom Fachbereich Informatik und Mathematik der
Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan: Prof. Dr. Detlef Krömker
Gutachter: Prof. Dr. Dirk Metzler
Prof. Dr. Lars Hedrich
Prof. Dr. Martin-Leo Hansmann

Datum der Disputation: 26.05.2010

Für meine Mutter

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Biologische Grundlagen 4
1.2 Lymphome 5
1.3 Die *Micorarray*-Technologie von Affymetrix 7
1.4 Das *Microarray-Chip*-Design von Affymetrix 9
1.5 Ziel dieser Arbeit 11

2 Material und Methoden **15**

2.1 Verwendete Genexpressionsdatensätze 15
2.1.1 Datensatz von Küppers et al. 15
2.1.2 Datensatz von Piccaluga et al. 16
2.1.3 Test-Datensatz 17
2.1.4 Datensatz von Brune et al. 18
2.2 Ablauf eines *Microarray*-Experiments 18
2.3 Technische Vorraussetzungen 19
2.4 Explorative Datenanalyse 20
2.4.1 Boxplot 20
2.4.2 Histogramm 20
2.4.3 Q-Q-Plot 21
2.5 Präprozessierungsalgorithmen für die Normalisierung von Affymetrix-*Microarray*-Rohdaten 21
2.5.1 Ziel der Präprozessierung 21
2.5.2 Das Modell von Affymetrix 23
2.5.3 Modell von Klein et al. 28
2.5.4 Das VSN-Modell von Huber et al. 29
2.5.5 *Medianpolish* 36

2.5.6 Das RMA-Modell von Irizarry et al. 37
2.5.7 Das sRMA-Modell von Cope et al. 38
2.5.8 Das sVSN-Modell 40
2.6 Simulation von Daten 42
2.6.1 Kontinuierliche Verteilungen 42
2.6.2 Simulieren der Daten 47
2.6.3 Lineare gemischte Modelle 48
2.7 *Cluster-Analysen* 51
2.8 Heatmap 55
2.9 Definition der differentiellen Genexpression 55
2.9.1 Definition des Fold-Changes 57
2.9.2 Das multiple Testen-Problem 58
2.10 Signalweg- und Genfunktionsgruppen-Analysen 59
2.10.1 Signalweg-Software und Datenbanken 59
2.10.2 Überrepräsentationsanalysen mit Gen-Ontologien 60
2.11 Qualitätskriterien zur Beurteilung der *Microarray*-Ergebnisse 63

3 Ergebnisse **65**
3.1 Auswertung des Datensatzes von Küppers et al. 65
3.1.1 Beschreibung der statistischen Analyse-Strategie 65
3.1.2 Betrachtung der differentiell-exprimierten Gene 66
3.1.3 Betrachtung der GO-Analyse 71
3.2 Auswertung des Datensatzes von Piccaluga et al. 73
3.2.1 Beschreibung der statistischen Analyse-Strategie 73
3.2.2 Clusteranalyse 74
3.2.3 Betrachtung der differentiell-exprimierten Gene 76
3.2.4 Betrachtung der GO-Analyse 77
3.3 Zusammenfassung 80
3.4 Vergleich der Normalisierungsmethoden mit simulierter Expressionsdaten 81
3.4.1 Wahl der kontinuierlichen Verteilungsfunktion 81
3.4.2 Simulation der Daten auf Basis einer angepassten Gammaverteilung 94
3.4.3 Gemischt-lineares Modell zum Simulieren von Expressionsdaten 102

3.5 Vergleich der Normalisierungsmethoden mit den Daten von Brune et al. . . . 111
3.5.1 Deskriptive Betrachtung der Expressionsdaten . . . 112
3.5.2 Betrachtung der differentiell-exprimierten Gene . . . 114
3.6 Zusammenfassung . . . 117

4 Diskussion **119**
4.1 Reanalyse von publizierten Datensätzen . . . 119
4.1.1 Reanalyse des Datensatzes von Küppers et al. . . . 119
4.1.2 Reanalyse des Datensatz von Piccaluga et al. . . . 120
4.2 Simulation von Expressionsdaten . . . 121
4.2.1 Simulation der Expressionsdaten — basierend auf einer kontinuierlichen Verteilung . . . 121
4.2.2 Simulation der Expressionsdaten — basierend auf einer angepassten Gammaverteilung . . . 122
4.2.3 Simulation von Expressionsdaten — basierend auf einem gemischten linearen Modell . . . 122
4.3 Vergleich der verschiedenen Präprozessierungsmethoden bei einem ausgewählten Vergleich aus dem Datensatz von Brune et al. . . . 123
4.3.1 Differentielle Expression von cHL im Vergleich zu normalen B-Zellen . . . 124
4.4 Ausblick . . . 124

5 Zusammenfassung **127**

6 Appendix **129**
6.1 R-Skript sVSN . . . 129
6.2 R-Skripte für die Simulation von Expressionsdaten . . . 133
6.3 Reanalyse der *Microarray*-Daten von Küppers et al. . . . 149
6.4 GO-Analyse der Gesamt-Genliste nach Reanalyse der Daten von Küppers et al. . . . 161
6.5 GO-Analyse der 299 *Probesets*, die nur nach Reanalyse der Daten von Küppers et al. gefunden wurden . . . 162
6.6 Der Zytokine-Zytokine-Interaktionssignalweg . . . 163
6.7 Der humane B-Zellen-Netzwerk-Signalweg . . . 164

6.8 Reanalyse der *Microarray*-Daten von Piccaluga et al. 165
6.9 Stabilitätsüberprüfung des Clusterverfahrens nach Suzuki et al.[Suzuki and Shimodaira, 2006] 167
6.10 GO-Analyse der gesamten Genliste (599 *Probesets*), die nach Reanalyse der Daten von Piccaluga et al. gefunden wurde 168
6.11 GO-Analyse der gesamten Genliste (363 *Probesets*), die nur nach Reanalyse der Daten von Piccaluga et al. gefunden wurden 169
6.12 Ausschließlich nach sVSN-Methode gefundene *Probesets* an einem Teildatensatz von Brune et al. 169
Literatur 173
Abkürzungsverzeichnis 184
Danksagung 185
Publikationen 188
Lebenslauf 191
Erklärungen 193

Abbildungsverzeichnis

1.1 Affymetrix Chip 2

1.2 Molekulare Struktur der DNA, inklusive Transkription und Bildung der mRNA (Quelle: National Human Genome Research Institute) . . 4

1.3 Eine B-Zelle wird durch eine T-Helferzelle aktiviert. (Quelle: Charles A. Janeway jr. u. a.: Immunologie. 5. Auflage. Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin 2002) 6

1.4 Eine B-Zelle wird nach Antigenkontakt zur Antikörper produzierenden Plasmazelle. (Quelle: Dr. med. Mario Schubert, Heidelberg, Deutschland) 6

1.5 Allgemeines *Probeset*-Design für alle 3'-Expressions-Microarrys von Affymetrix 8

2.1 Ablauf eines Micorarray-Experiments 19

2.2 Darstellung eines horizontalen Boxplots 20

2.3 Histogramm-Beispiel 21

2.4 Schematische Darstellung der Delta-Methode 31

2.5 Funktionsverlauf von f(x) und $h_s(x)$ [Huber et al., 2002] 33

2.6 Schematischer Aufbau einer mRNA 39

2.7 Dichtefunktion der Weibullverteilung bei verschiedenen Form- und Skalenparametern. 44

2.8 Dichtefunktion der Lognormalverteilung bei verschiedenen σ und $\mu = 0$. 45

2.9 Dichtefunktion der Gammaverteilung bei verschiedenen Form- und Skalenparametern. 46

3.1 Heatmap der 17 Gene, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probesets*/Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert. 69

3.2 Unsupervised Clustering der 291 *Probesets* mit $\sigma \geq 1,5$ nach Reanalyse der Daten von Piccaluga et al.. Spalten repräsentieren individuelle Patienten, Zeilen entsprechen den verschiedenen dargestellten Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert. Die Unterscheidung der Gruppen unter dem Dendrogramm der Spalten erfolgt durch zusätzlichen Farbcode: Rot für PTZLs, Grün für AILTs und Blau für ALCLs. 75

3.3 Heatmap der 30 *Probesets*, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probsets*/Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert. . . 78

3.4 Heatmap der 7 Gene, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probsets*/Genen. Die Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbcodiert. 80

3.5 Histogramm der logarithmierten realen Expressionsdaten 82

3.6 Histogramme zu den simulierten Expressionswerten und die dazugehörigen Q-Q-Plots 83

3.7 Histogramme zu den simulierten und gemessenen Mittelwerten und Varianzen und den dazugehörigen Q-Q-Plots 84

3.8 Die Q-Q-Plots der simulierten und gemessenen Expressionswerte, Mittelwerte und Varianzen nach der Anpassung der empirischen Verteilung 85

3.9 Vergleich der Boxplots von realen und simulierten Expressionsdaten . 86

3.10 Boxplot der Expressionsdaten mit einfacher IVT 86

3.11 Vergleich der RNA-Degradation bei einfacher und doppelter Invitrotranskription. Die einzelnen Linien repräsentieren jeweils einen *Microarray*-Chip. 87

3.12 Die RNA Degradation bei simulierten Daten 88

3.13 Grafische Darstellung der simulierten differentiell-exprimierten *Probesets*, die bei allen 10 Simulationen wiedergefunden wurden. Jedes Feld entspricht einem anderen Quantilbereich. 90
3.14 Boxplots zum Wiederfinden der simulierten differentiell-exprimierten *Probesets* für die 10 Simulationen, d. h. die Expressionswerte sind über alle Expressionsstärken für die einzelnen Simulationen dargestellt. 92
3.15 Sensitivität und Spezifität der einzelnen Methoden für die 10 Simulationen 93
3.16 Sensitivität und Spezifität, dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit einer empirischen Verteilung 94
3.17 Histogramm der simulierten PM-Expressionswerte resultierend aus einer angepassten Gammaverteilung und der QQ-Plot im Vergleich zu den realen PM-Expressionsdaten 96
3.18 Boxplot der Gamma-simulierten Expressionsdaten 97
3.19 RNA-Degradationsplot der Gamma simulierten Expressionsdaten . . 98
3.20 Prozentualer Anteil der richtig Positiven differentiell-exprimierten *Probesets* bei gamma-verteilten Daten. Jedes Feld entspricht einem anderen Quantilbereich. 99
3.21 Boxplot für die Präprozessierungsmethoden für alle 10 Simulationen . 100
3.22 Sensitivität und Spezifität der Präprozessierungsmethoden für alle 10 Simulationen 101
3.23 Sensitivität und Spezifität, dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit einer angepassten Gammaverteilung . 102
3.24 Histogramm der simulierten PM-Expressionswerte resultierend aus einem linearen gemischten Modell und der QQ-Plot im Vergleich zu den realen PM-Expressionsdaten 104
3.25 Boxplot der simulierten Expressionsdaten — basierend auf einem linearen gemischten Modell 105
3.26 RNA-Degradationsplot der simulierten Expressionsdaten resultierend aus einem linear gemischten Modell 106
3.27 Prozentualer Anteil der richtig positiven differentiell-exprimierten *Probesets*. Der Faktor gibt die Stärke der differentiellen Expression an. . 108
3.28 Boxplot der Präprozessierungsmethoden für alle 10 Simulationen . . . 109

3.29 Sensitivität und Spezifität der Präprozessierungsmethoden für alle 10 Simulationen . . . 110
3.30 Sensitivität und Spezifität dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit eines linear gemischten Modells . . . 111
3.31 Histogramm der logarithmierten Expressionsdaten . . . 112
3.32 Boxplot der logarithmierten Expressionsdaten, der 12 HLs und 10 Keimzentrum-B-Zellen . . . 113
3.33 Degradationsgrad der PM-Werte in Abhängigkeit von ihrer Position in den einzelnen *Microarrays* . . . 114

6.1 GO-Analyse der Gesamt-Genliste nach der Reanalyse der Daten von Küppers et al. . . . 161
6.2 GO-Analyse der 299 *Probesets* die nur nach der Reanalyse der Daten von Küppers et al. gefunden wurden. . . . 162
6.3 Signalweg Zytokine-Zytokine Interaktion . . . 163
6.4 Signalweg humane B-Zellen Netzwerk . . . 164
6.5 Stabilitätsprüfung des Clusterverfahrens nach Suzuki et al.[Suzuki and Shimodaira, 2006]. . . . 167
6.6 GO-Analyse über die gesamte Genliste (599 *Probesets*) die nach der Reanalyse der Daten von Piccaluga et al. gefunden wurde . . . 168
6.7 GO-Analyse über die Genliste (363 *Probesets*) die nur nach der Reanalyse der Daten von Piccaluga et al. gefunden wurden . . . 169

Tabellenverzeichnis

2.1 Schematische Darstellung der einzelnen Schritte der verschiedenen Präprozessierungsalgorithmen bei Affymetrix-*Microarrays* 41

3.1 Die 299 differentiell-exprimierten *Probesets*, die nur nach Reanalyse identifiziert wurden. Der Schwerpunkt der Regulationsstärke liegt bei einem FC zwischen 2 und 3 im Vergleich von HL-Linien und normalen B-Zellen. 67

3.2 Die 17 Gene, die nach Reanalyse identifiziert wurden (FDR ≤ 0.05), mit dem größten FC zwischen HL-Linien und normalen B-Zellen. Gene die mit „*“ markiert sind, wurden im Vergleich zwischen cHLs und Keimzentrums B-Zellen als signifikant-reguliert gefunden. 68

3.3 Die fünf Gene mit dem höchsten FC zwischen HL-Linien und normalen, reifen B-Zellen, die nur nach Reanalyse identifiziert werden konnten. Diese Gene haben eine FDR ≤ 0.05 und wurden als signifikant-differentiell exprimiert in primären HL-Patienten gefunden ([Brune et al., 2008]) 70

3.4 Anzahl der Gene, die differentiell sind, und nach Durchführung der verschiedenen Normalisierungsmethoden am Datensatz von Brune et al. gefunden wurden 115

3.5 Ein Teil der Gene, die ausschließlich mit der sVSN-Methode im Datensatz von Brune et al. als differentiell gefunden wurden 116

6.1 Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] 149

6.2 Genliste der zusätzlich gefundenen 30 Probesets mit $FC \geq 10$ oder ≤ -10 nach Reanalyse des Datensatzes von Piccaluga et al. [Piccaluga et al., 2007] 165

6.3 Genliste der zusätzlich gefundenen 66 Probesets nach Reanalyse des Datensatzes von Brune et al. [Brune et al., 2008] 170

Kapitel 1

Einleitung

Die *Microarray*-Technologie ermöglicht heute, simultan mehr als 20000 Gene zu messen. Seit 1994 produziert die Firma Affymetrix kommerzielle *Microarrays*. Zunächst arbeiteten nur wenige Institute und Firmen mit diesen *Microarrays*. Dies lag an den hohen Kosten und der anfangs schlechten Reproduzierbarkeit der Ergebnisse. Inzwischen ist Affymetrix Marktführer für die *Microarray* Technologie mit vielen verschiedenen *Microarrays*. Verschiedene Spezies werden abgedeckt. Gemessen werden kann sowohl auf mRNA- und DNA-Ebene. 3'Expressions-*Microarrays* messen auf der mRNA-Ebene. Außerdem gibt es sogenannte SNP-*Microarrays* (DNA-Ebene) und mikro RNA (regulatorisch wirkende RNAs) *Microarrays*.
Führt man das Experiment mit selbst hergestellten DNA-*Microarrays* durch, gibt es oft technische Fehler. Diese werden durch die standardisierte Produktion der *Microarrays* und das Vorhandensein standardisierter Protokolle zur experimentellen Aufbereitung der Gewebeproben verhindert.
Mit *Microarrays* können neue Gene und Signalwege gefunden werden, die für einen Krankheitssubtyp spezifisch sind. So führen *Microarrays* zu einem besseren Verständnis der Pathogenese und dadurch zu einem besseren Therapiekonzept. 2006 gelang es Hummel et al. [Hummel et al., 2006] zum ersten Mal mit Affymetrix *Microarrays* ein Genset zu identifizieren, das eindeutig zwischen dem diffus großzelligen B-Zell-Lymphom (DLBCL) und den Burkitt-Lymphom (BL) unterscheiden kann. Affymetrix-*Microarrays* sind ein fester Bestandteil der neuen biologischen und medizinischen Forschung geworden. Zukunftsversion ist, dass *Microarrays* als Standard in Kliniken eingesetzt werden können und so jeder Patient von dieser Technologie profitiert.

Dieser Schritt verlangt neben einer standardisierten Produktion und experimentellen Durchführung auch eine standardisierte Dokumentation der verwendeten statistischen Analysen. Hierzu gibt es viele Arbeiten, die zeigen, dass verschiedene statistische Methoden zu verschiedenen Ergebnissen führen können [Hoffmann et al., 2002; Parrish and Spencer, 2004].
Der *Microarray* — auch Chip genannt — ist dabei der Träger auf dem biologisches Material, in unserem Fall Nukleotide (Sonden) in hoher Dichte und Anzahl (10^6 - 10^7 Kopien) und in einer definierten Anordnung aufgetragen sind (Abb.1.1).

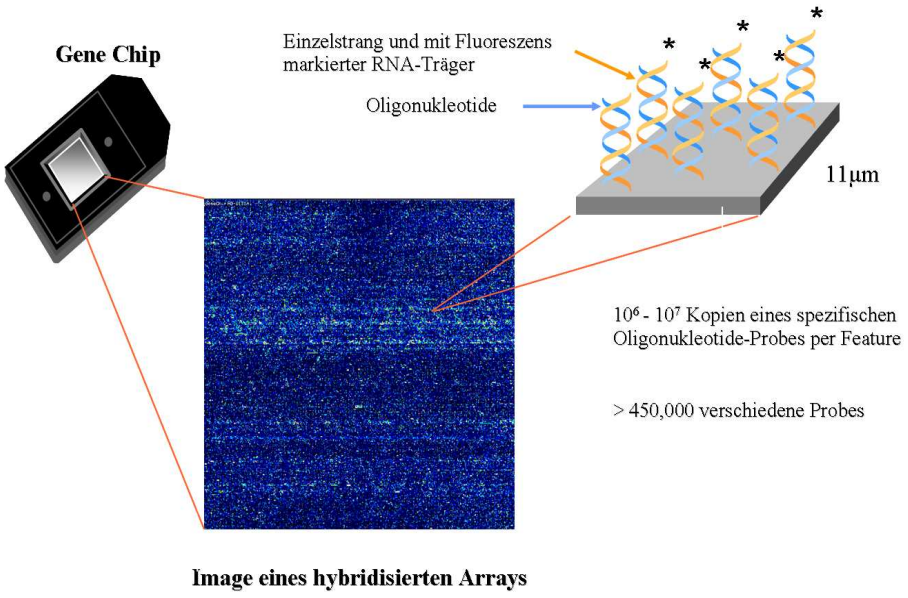


Abbildung 1.1: Affymetrix Chip

Microarrarys basieren auf dem Prinzip der Hybridisierung. Bei der Hybridisierung bildet unter bestimmten Bedingungen eine RNA-Kette mit dem komplementären DNA-Strang eine stabile Doppelhelix. Die Eigenschaften des so durch Basenpaarung entstandenen Hybridmoleküls führten zur Entwicklung der Methode der Nukleinsäurehybridisierung.
Um die Genexpression messen zu können, muss vor der Hybridisierung die mRNA mit fluoreszierenden Farbstoff markiert werden. Messbar wird die Fluoreszensintensität auf Grund des linearen Zusammenhangs, dass umso mehr mRNA an den immobilisierten Sonden bindet, desto mehr Fluoreszens wird gemessen. Die Fluoreszenz kann nur mit einem speziellen Scanner aus den *Microarrays* gemessen werden. Die beim Scannen erzeugten Bilddaten müssen in Signalintensitäten umgesetzt

werden. Die Signalintensitäten können dann je nach Fragestellung mit weiteren statistischen Methoden ausgewertet werden. Die Signalintensitäten sind Rohdaten, die normalisiert werden müssen, um systematische und technische Fehler (z.B. bei unterschiedlicher mRNA Qualität und Quantität) zu eliminieren.
Bei allen für diese Arbeit durchgeführten Analysen werden verschiedene Generationen humaner *Microarrays* der Firma Affymetrix verwendet. Die statistische Auswertung der numerischen Rohdaten wird mit der frei verfügbaren Statistiksoftware R [Ihaka and Gentleman, 1996] und den im Bioconductor existierenden Paketen *affy* [Gautier et al., 2004], *VSN* [Huber et al., 2002], *geneplotter*, *multest*, usw. durchgeführt. Ende 2001 wurde das Open-Source-and-Development-Softwareprojekt Biconductor am Institut des Dana Farber Cancer Institut der Harvard Univerität initiiert [Gentleman et al., 2004]. R und Bioconductor sollen das statistisch und grafisch adäquate Auswerten immer größerer Menge genomischer Daten ermöglichen.
Insbesondere Bioconductor ist fester Bestandteil für die statistische Auswertung von *Microarrays* geworden. Deshalb integriert auch kommerziell entwickelte *Micorarray*-Software (GeneSpring, Spotfire, Partek, Affymetrix, usw.) entweder Pakete aus Bioconductor oder stellt eine direkte Schnittstelle zu R und Bioconductor bereit.
In der Literatur gibt es bereits einige Evaluationen zu den Unterschieden zwischen den verschiedenen Auswertungsalgorithmen [Irizarry et al., 2006; Ploner et al., 2005; Choe et al., 2005; Hoffmann et al., 2002]. Allerdings wurden diese nur mit standardisierten Daten entwickelt, das heißt unter Berücksichtigung der Angaben des *Microarray*-Herstellers zur Ausgangsmenge an Gesamt-mRNA. Bei manchen Erkrankungen sind nur wenige Tumorzellen vorhanden. Deshalb ist hier das Ausgangsmaterial sehr begrenzt. Zum Zeitpunkt der Datenerhebung gab es noch kein kommerzielles Protokoll, das mit so einer geringen Ausgangsmenge zu Ergebnissen geführt hätte. Kommerzielle Protokolle benötigen immer noch die zehnfache Menge an Zellen ($\geq$10000), die für die Experimente bereitgestellt werden können. Da die verwendeten *Microarrays* von Affymetrix nicht für eine solche Vorbehandlung entwickelt wurden, sind spezielle Anpassungen an die weiterführenden statistischen Methoden nötig, vor allem bei der Normalisierung. Zu klären ist, ob und im welchem Umfang solche Anpassungen zu einem Vorteil gegenüber vorhandenen Algorithmen führten.

1.1 Biologische Grundlagen

Die DNA liegt als doppelsträngige Helix in jeder eukaryotischen Zelle vor. Das Grundgerüst besteht aus Zucker und Phosphatmolekülen, die um zwei Nukleotidketten angeordnet sind. An diesen Ketten sind die vier Basen Adenin, Guanin, Thymin und Cytosin angeordnet. Die beiden DNA-Stränge werden durch Wasserstoffbrücken zusammengehalten, die von den komplementären Basen Adenin und Thymin oder Cytosin und Guanin gebildet werden. Bei der Transkription müssen zuerst die Wasserstoffbrücken aufgebrochen werden, damit die DNA teilweise in zwei einzelnen Strängen vorliegt. Nur einer der beiden Stränge, der sogenannte codogene Strang, wird verwendet um mit Hilfe der RNA-Polymerase die mRNA zu bilden. Die Gensequenz auf der DNA wird während der Transkription in mRNA übersetzt und dann durch die Translation an den Ribosomen zu funktionell-biologisch wirkenden Proteinen synthetisiert. Die RNA-Polymerase verknüpft die angelagerten komplementären Basen. Dabei wird statt Thymin Uracil in die mRNA eingebaut. Bei der Basenfolge ist, abgesehen von Uracil, die mRNA eine komplementäre Kopie eines Teilabschnittes des codogenen Stranges der DNA [Wehner and Gehring, 1995] (Abb.1.2).

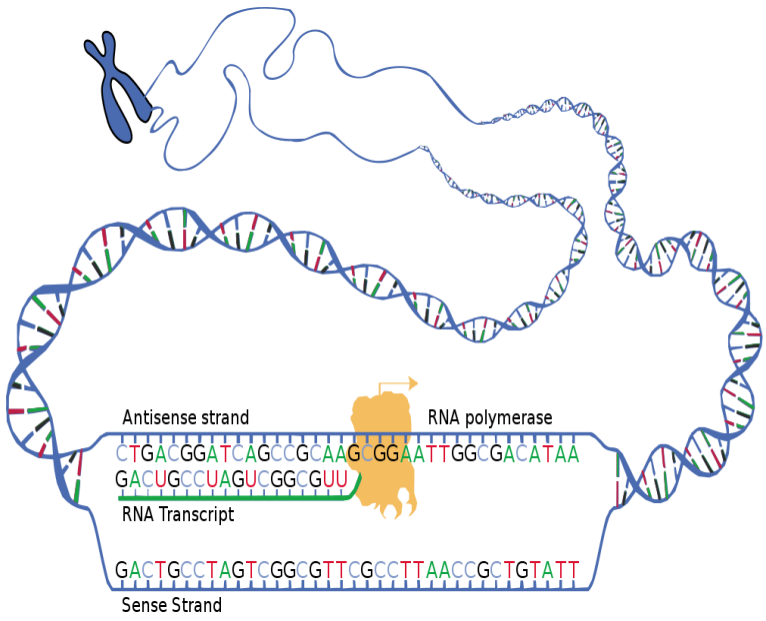


Abbildung 1.2: Molekulare Struktur der DNA, inklusive Transkription und Bildung der mRNA (Quelle: National Human Genome Research Institute)

Zwar besteht kein linearer Zusammenhang zwischen der Menge der gebildeten mR-

NA und der daraus resultierenden Proteinmenge, aber durch *Microarrays* kann man die mRNA-Menge im Vergleich zu einem anderem Experiment ermitteln. Eine Validierung der Ergebnisse eines *Microarray*-Experiments auf mRNA- und auf Proteinebene mit einer unabhängigen anderen Methode ist unerlässlich, um Artefakte auszuschließen. Die Bestätigung durch eine unabhängige andere Methode ist gängige Laborpraxis.

1.2 Lymphome

Allgemein sind Lymphome Schwellungen der Lymphknoten oder Tumore im Lymphsystem bei gutartigem oder bösartigem Gewebe. Die bösartigen (malignen) Lymphome werden nach der WHO-Klassifizierung in Hodgkin-Lymphome (HL) und Non-Hodgkin-Lymphome (NHL) unterteilt [Jaffe et al., 1998]. Alle malignen Lymhome stammen von malignen Lymphozyten ab. Etwa 90 % der NHLs stammen von B-Zellen ab [Fisher, 2003].
Zu den Lymphozyten gehören die B-Lymphozyten (B-Zellen) und T-Lymphozyten (T-Zellen). B-Zellen und T-Zellen werden beim Menschen im Knochenmark gebildet. B-Zellen sind als einzige Lymphozyten in der Lage, Antikörper herzustellen. Sie bilden zusammen mit den T-Zellen das adaptive Immunsystem. Die T-Zellen übernehmen die Rolle der zellvermittelten Immunantwort. Wenn naive B-Zellen, die in unserem Blutkreislauf und im Lymphsystem zirkulieren, auf ein fremdes Antigen stoßen, werden die B-Zellen durch ihren B-Zell Rezeptor gebunden, in die Zelle aufgenommen und dann mit einem MHC-Molekül wieder an der Membran-Oberfläche präsentiert (Abb.1.3).
Bei der T-Zellen abhängigen Aktivierung bedarf es nun einer geeigneten T-Helferzelle, die durch ihren Rezeptor die T-Zelle an den Antigen-MHC-Komplex binden kann. Erst jetzt wird die B-Zelle durch Ausschüttung von Zytokinen der gebundenen T-Zelle aktiviert. Aus der aktivierten naiven B-Zelle (Abb.1.4) vermehrt und differenziert sich eine antikörperproduzierende Plasmazelle (PC). Werden B-Zellen durch körperfremde Antigene aktiviert, können sich Plasmazellen oder Gedächtnis-Zellen entwickeln. Diesen Vorgang bezeichnet man als Klassenwechsel der B-Zelle. Die Vermehrung (klonale Expansion) findet im Keimzentrum (GC) des Lymphknotens oder in der Milz statt; die B-Zellen werden als Keimzentrums B-Zellen (GC B-Zellen)

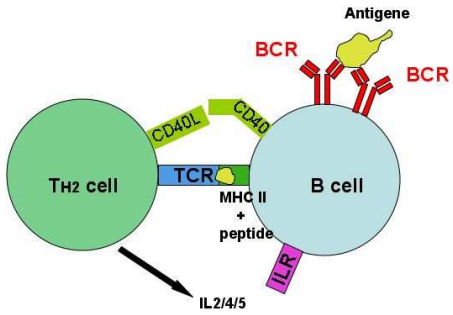


Abbildung 1.3: Eine B-Zelle wird durch eine T-Helferzelle aktiviert. (Quelle: Charles A. Janeway jr. u. a.: Immunologie. 5. Auflage. Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin 2002)

bezeichnet [Alberts et al., 1995].

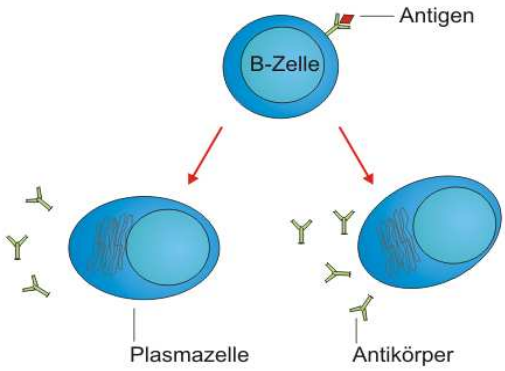


Abbildung 1.4: Eine B-Zelle wird nach Antigenkontakt zur Antikörper produzierenden Plasmazelle. (Quelle: Dr. med. Mario Schubert, Heidelberg, Deutschland)

Sind die Antigene oft wiederholte Polysaccharide (Unterklasse der Kohlenhydrate), wie sie z. B. bei Bakterien vorkommen, ist eine zusätzliche Aktivierung durch eine T-Helferzelle nicht nötig. Die B-Zelle ist aktiviert allein über ihren B-Zell-Rezeptor, die sogenannte T-Zell-unabhängige Aktivierung, ohne einen Klassenwechsel oder die Bildung von Gedächtnis-B-Zellen [Alberts et al., 1995].
Die starke Proliferation (Zellteilung) der GC-B-Zellen erhöhen das Risiko, chromosomale Translokationen zu generieren und spielen deshalb bei der Pathogenese

(Entstehung) von B-Zell-Lymphomen eine Rolle [Ralf Küppers, 2003]. Während der Keimzentrumsreaktion finden verschiedene Umbauprozesse statt, wie z. B. die Einführung von DNA-Strangbrüchen [Bross et al., 2000; Goossens et al., 1998; Papavasiliou and Schatz, 2000], die somatische Hypermutation (Einfügen von Mutationen in die Antikörpergene einer reifenden B-Zelle) und den Klassenwechsel. Genau in diesen Regionen findet man oft Translokationen von Onkogenen [Klein et al., 1998; Küppers and Dalla-Favera, 2001]. Auch Punktmutationen können eine Folge der somatischen Hypermutation sein [Küppers, 2005].
Keimzentrums-B-Zellen zeigen charakteristisch somatisch mutierte V-Gene (V = Variablitätselement der V-Gene). Dieses Merkmal ermöglicht Vergleiche zwischen B-Zell-Lymphomen und normalen B-Zellen und kann damit zur Identifizierung des Entwicklungsstadiums der Vorläuferzellen dienen.
T-Zellen werden im Knochenmark gebildet, aber im Thymus ausdifferenziert. Zellen, die fremde Antigene auf ihrer Membranoberfläche repräsentieren, werden von zytotoxischen T-Zellen ($CD4^{-}CD8^{+}$), die einen MHCI erkennenden T-Zell Rezeptor besitzen, erkannt und getötet. T-Zellen, die einen MHCII erkennenden T-Zell Rezeptor besitzen, werden allgemein als T-Helfer-Zellen bezeichnet. Es gibt des Weiteren noch regulatorische T-Zellen (Treg), T-Gedächtniszellen und NK (Natürliche Killer T-Zellen). Da bei der Entwicklung und Ausdifferenzierung der T-Zellen weder ein Klassenwechsel noch eine somatische Hypermutation stattfindet, ist es schwer, die Vorläuferzellen der T-Zell-Lymphome zu definieren [Alberts et al., 1995].

1.3 Die *Micorarray*-Technologie von Affymetrix

Die *Microarrays* von Affymetrix verwenden eine Kombination aus Photolithographie und einer kombinatorischen chemischen Synthese, um die einzelsträngigen DNA-Fragmente (sogenannte Oligonukleotide) auf eine Glasoberfläche zu synthetisieren [Lipshutz et al., 1999].
Diese Fragmente haben eine Länge von 25 Basen (auch *Probe* genannt). Jedes zu messende Gen wird durch eine bestimmte Anzahl von *Probes*, abhängig von *Microarray*-Typ und Gen, repräsentiert. Die Anzahl schwankt zwischen 11 und 20 *Probes* pro Gen. Diese sogenannten *Perfect Matches* (PM) repräsentieren die komplementäre Sequenz der mRNA und bilden zusammen mit dem *Mismatch* (MM) das *Probeset*

eines Gens. Das MM hat an der 13. Basenposition einen komplementären Austausch der reellen Base und stellt damit die nicht spezifische Bindung und das Hintergrundrauschen dar.
In Abbildung 1.5 ist schematisch der Aufbau eines *Probesets* dargestellt. Die PMs eines *Probesets* sind nicht an nur einer Stelle auf dem *Microarray*, sondern randomisiert auf dem Chip verteilt, um mögliche lokale räumliche Effekte zu minimieren. Auch wenn nur ein Teil der PMs eines *Probesets* detektiert werden konnte, ist das Signal aus den verbliebenen PMs noch zu verwenden. Jedes *Probe* ist mit 10^6 bis 10^7 Kopien auf dem Chip vertreten und bildet die sogenannte *Probe-Cell*. Das Sequenzdesign der einzelnen *Probes* auf dem Chip ist so gewählt, dass Sequenzen der mRNA nahe 3'-Ende abgebildet werden, um Probleme von teilweise degradierter mRNA oder durch die Invitrotranskription (Verfahren zur Vermehrung der mRNA) verkürzter RNAs zu verringern. Am 3'-Ende bindet die Polymerase, die für das Invitrotranskriptionsverfahren benötigt wird. Je weiter die Polymerase Richtung 5'-Ende wandert, umso höher ist die Wahrscheinlichkeit, dass diese ihre Funktion abbricht. Daher kommt es vor allem bei langen Transkripten oder bei schlechter RNA-Qualität zu einer schlechten Abdeckung der *Probes* Richtung 5'-Ende.

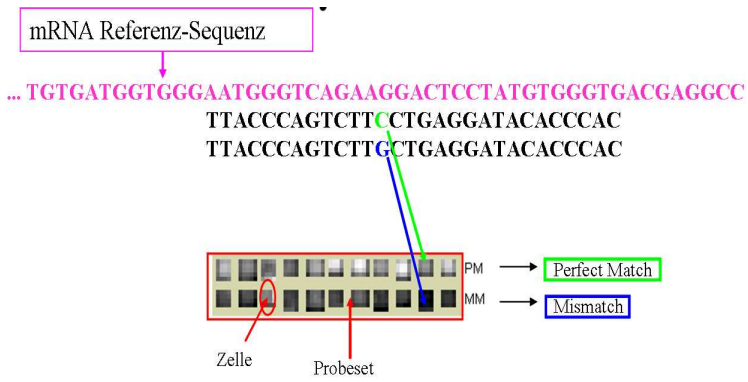


Abbildung 1.5: Allgemeines *Probeset*-Design für alle 3'-Expressions-Microarrys von Affymetrix

Für die vorliegende Arbeit wurden verschiedene Chip-Generationen von Affymetrix verwendet (HGU95, HGU133A, HGU133Plus2.0), die sich im *Probe*-Design und in der Anzahl der Gene unterscheiden. Beim Scannen wird die Fluoreszenz aller *Probe Cells* gemessen und im sogenannten DAT-File gespeichert. Diesen kann man mit der frei erhältlichen AGCC (*Affymetrix-Gene-Chip-Console*) Software von Affyme-

trix visualisieren. Aus den einzelnen *Probe-Cells* wird die Fluoreszenzstärke in einen numerischen Expressionswert umgewandelt und im sogenannten CEL-File gespeichert. Das CEL-File repräsentiert also die Rohdaten in Form einer numerischen Matrix und kann für weitere statistische Analysen verwendet werden.
Affymetrix selbst hat Algorithmen entwickelt, um aus den einzelnen Expressionswerten der *Probes* einen aggregierten Wert für das *Probeset* oder Gen zu berechnen. Allerdings wurde dieses Verfahren schon mehrfach in der Literatur kritisiert. Dort wurde belegt, dass diese Methode einige Schwächen aufweist [Irizarry et al., 2003b; Lazaridis et al., 2002]. Es gibt eine Vielzahl kommerzieller und frei erhältlicher Software zur statistischen Analyse von *Microarray*-Daten. R ist die freie Software mit einer Sammlung von Softwarepaketen mit neu implementierten Algorithmen. Alle statistischen Analysen dieser Arbeit wurden mit R und Bioconductor durchgeführt. Bioconductor ist ein frei erhältliches Softwareprojekt, aufbauend auf R zur Erstellung von Werkzeugen, zur Analyse und zur Interpretation genetischer Daten.
Das vierstufige Verfahren (Hintergrundkorrektur, Normalisierung, PM-Adjustierung, PM-Signal Zusammenfassung), gehört zum *affy*-Paket von Bioconductor. Es wird im Material- und Methodenteil detailliert beschrieben und aggregiert aus den *Probe*-Signalen des CEL-Files einen Expressionswert pro *Probeset* oder Gen.

1.4 Das *Microarray-Chip*-Design von Affymetrix

Gegeben ist ein Projekt mit n Chips, wobei jeweils ein Chip eine Gewebeprobe repräsentiert. Für jeden Chip $i = 1, \ldots, n$ sind m Gene vorhanden, die durch $l_j (j = 1, \ldots, m)$ *Probe-Pairs* vertreten sind. Jedes *Probe Pair* besteht aus einem PM und einem MM (Siehe Abb.1.5). Gegeben ist also PM_{ijk}, wobei $i = 1, \ldots, n$ der Chip, $j = 1, \ldots, m$ das Gen und $k = 1, \ldots, l_j$ das *Probe* ist. Abhängig vom Chiptyp und dem jeweiligen Gen kann die Anzahl von l_j differieren. Somit ergeben sich $n \cdot \sum_{j=1}^{m} l_j$ Werte für die PM-Werte, äquivalent erhält man die MM-Werte mit MM_{ijk}.

Übersicht der Analyse im Bioconductor

Zunächst wird eine Hintergrundkorrektur vorgenommen. Dies ist nötig, da auch ohne eine Hybridisierung eine Fluoreszenz-Intensität gemessen werden kann. Bio-

conductor stellt im *affy*-Paket drei verschiedene Möglichkeiten zur Auswahl. Da in der Normalisierungmethode VSN von Huber et al. dieser Schritt schon enthalten ist, wird auf diese Korrektur verzichtet. Aufgrund unterschiedlicher mRNA-Mengen kann es zu verschiedenen Fluoreszenzintensitäten kommen. Im zweiten Schritt wird deshalb eine Normalisierung durchgeführt. Bioconductor bietet dazu eine Vielzahl Möglichkeiten. Alle Reanalysen wurden mit der Normalisierungsmethode VSN von Huber et al. normalisiert. Im Material- und Methodenteil wird hierauf detaillierter eingegangen. Bei diesem Schritt erhält man die normalisierten Werte PM_{ijk}^{norm} und MM_{ijk}^{norm}. Für den nächsten Schritt, die PM-Adjustierung, wird auf die MM_{ijk}^{norm} verzichtet, das heißt hier findet keine Verrechnung der PM_{ijk}^{norm} mit den MM_{ijk}^{norm} statt. Dieses Vorgehen schlagen Bolstad et al. [Bolstad et al., 2003] vor. Berücksichtigt wird damit, dass die MM Signale oft — nicht wie von Affymetrix angenommen — unspezifische Bindungen darstellen [Naef et al., 2002]. Der Wert für das *Probe-Pair-k* wird durch PM_{ijk}^{norm} repräsentiert.
Die $n \cdot \sum_{j=1}^{m} l_j$ Werte haben sich in diesem Schritt halbiert. Wie wird nun aus l_j Werten des Gens j ein Wert für die Expression dieses Gens j berechnet? Bioconductor stellt einige Möglichkeiten zur Verfügung. Dieser abschließende Schritt bei den Reanalysen wird mit der Methode von Tukey et al. [Tukey, 1977] durchgeführt. Der Material- und Methodenteil enthält eine detaillierte Beschreibung dieser Methoden. Die verrechneten PM_{ijk}^{norm}, oder allgemeiner y_{ijk}, für jeden Chip i und jedes Gen j werden im Weiteren mit x_{ij} bezeichnet und bilden zusammen die Matrix X.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Diese Matrix ist die essentielle Grundlage für alle statistischen Analysen. Abhängig von der Fragestellung gibt es verschiedene weitere statistische Analysen-Methoden. Eine wichtige Fragestellung ist z. B., welche Gene differentiell zwischen zwei definierten Gruppen exprimiert sind und in welchem Ausmaß.
Das Ausmaß, also die Stärke der differentiellen Expression wird auch Fold Change (FC) genannt. Werden die Gruppen im Vorfeld definiert, spricht man auch von „überwachten (supervised) Analysen“. Sowohl die Signifikanz eines Gens als auch der FC sind sehr wichtig für das weitere Vorgehen. Nur Gene, die eine entsprechende FC ha-

ben, werden als biologisch relevant angesehen (z. B. FC mindestens 2- oder 3-fach). So können z. B. Subtypen einer Erkrankung, z. B. mit Hilfe von Clusteranalysen entdeckt werden. Bei diesen Clustermethoden wird im Allgemeinen „nicht überwacht (unsupervised)“ vorgegangen, das heißt die Gruppen werden nicht vorher definiert. Auch gibt es Methoden, die erlauben, einen Prädiktor zu erstellen, der zukünftige Patientengruppen eindeutig einer bestimmten Klasse zuordnet. Immer mehr an Bedeutung gewinnt die Fragestellung nach involvierten Signalwegen und Genfunktionsgruppen. Dabei wird nicht mehr auf der Einzel-Gen-Ebene analysiert, sondern in zu einem Signalweg gehörenden Gengruppen oder einer Funktionsgruppe. Von besonderem biologischen Interesse sind Gengruppen oder Signalwege, die eine Überrepräsentierung oder Unterrepräsentierung zeigen.

1.5 Ziel dieser Arbeit

Die vorliegende Arbeit ist in zwei Teile untergliedert:

1. Die Vorteile neuer statistischer Methoden bei der *Micorarray* Auswertung an schon veröffentlichten Datensätzen werden dargestellt:
   - Am Beispiel der Analyse von Klein et al. [Klein et al., 2001] werden Ansätze zur Verbesserung der Methode aufgezeigt.
   - Ergebnisse der Reanalyse sind zusätzliche Gene, die nur nach der Reanalyse gefunden werden.
   - die biologische Relevanz der zusätzlich gefundenen Gene.
2. Expressionsdaten, die aus geringen mRNA-Mengen stammen, werden statistisch analysiert:
   - Präprozessierungs-Algorithmen werden an simulierten Rohdaten geprüft und bewertet.
   - Welche Methoden zeigen welche Vorteile?
   - Zeigt die Gewichtung der *Probes* in Abhängigkeit ihrer Position im Transkript die Vorteile der Analyse?

- Reanalyse eines Vergleiches mit allen angewendeten Präprozessierungs-Algorithmen.

Für die Reanalyse wurden die Datensätze von Küppers et al. [Küppers et al., 2003] und Piccaluga et al. [Piccaluga et al., 2007] verwendet. Dr.Ulf Klein hat beide Datensätze mit derselben Methode statistisch analysiert.
Wie schon im Vorfeld erwähnt, gibt es bestimmte Erkrankungen, bei denen das Ausgangsmaterial nur begrenzt zur Verfügung steht. Mit dem in unserem Institut entwickeltem Protokoll ist es erstmals möglich, aus nur 1000 Zellen genügend mRNA zu amplifizieren, um ein *Micorarray*-Experiment durchführen zu können. Da der verwendete *Microarray* für diese Art amplifizierter mRNA nicht konzipiert ist, kommt es zu systematischen Fehlern. Der Anteil an falsch negativen Werten ist hoch. In der vorliegenden Arbeit wird nach der Normalisierung noch eine Gewichtung der *Probes* vorgenommen. Diese Methode berücksichtigt wesentliche Teile des systematischen Fehlers, der während der mRNA-Prozessierung entsteht.
Zwar haben Cope et al. [Cope et al., 2006] gewichtete *Probes* in ihren Algorithmus RMA (*Robust Multichip Average*) implementiert, konnten aber keine deutlichen Unterschiede und Verbesserungen zum Standard RMA-Algorithmus feststellen. Die verwendete Normalisierungsmethode VSN von Huber et al. zeigte eine höhere Spezifität und Sensitivität bei der differentiellen Expression. Damit war ein anderes Ergebnis als bei Cope et al. zu erwarten.
Zuerst wurden simulierte Daten verwendet, um die verschiedenen Präprozessierungsalgorithmen vergleichen zu können. An einem dafür konzipierten Testdatensatz wurden die verschiedenen Methoden verglichen. Dazu wurden zwei verschiedene Gruppen von Ausgangsgewebe gewählt, die ein hohes Maß an differentieller Expression erwarten ließen. Anhand von Qualitätskriterien sollte geklärt werden, ob die verschiedenen Methoden einen Effekt auf die weitere statistische Analyse haben. Qualitätskriterien sind dabei Sensitivität und Spezifität, Anzahl der differentiell-expremierten Gene und der Einfluss auf resultierende Gengruppen (z. B. Gen-Ontologien und Signalweg-Analysen). Letztendlich sollten die verschiedenen Methoden an einem realen Teildatensatz von Brune et al. [Brune et al., 2008] angewendet und miteinander verglichen werden.
Im zweiten Kapitel werden die verwendeten Datensätze bezüglich ihrer mRNA-Prozessierung, der verwendeten Materialien und Gewebe beschrieben. Im Anschluss werden die verwendeten Präprozessierungs-Algorithmen erörtert, die für alle Daten-

sätze angewendet wurden. Es folgt eine Beschreibung der gewichteten *Probes* durch ein angepasstes lineares Modell. Im Ergebnisteil wird deren potentieller Vorteil erörtert.

Im dritten Kapitel werden zunächst die Ergebnisse der Reanalyse der beiden Datensätze dargestellt. Anschließend geht es um die Simulation von Expressionsdaten mit verschiedenen Modellen und die Anwendung der verschiedenen Methoden, die zuletzt auf einen Teildatensatz angewendet und verglichen werden. Das Verhalten im biologischen Kontext wird besonders herausgearbeitet und — wenn möglich — eine Empfehlung zur weiteren statistischen Analyse gegeben.

In Kapitel 4 werden die Aspekte des Ergebnisteils ausführlich diskutiert.

Kapitel 2

Material und Methoden

2.1 Verwendete Genexpressionsdatensätze

2.1.1 Datensatz von Küppers et al.

Küppers et al. [Küppers et al., 2003] haben die erste genomweite Genexpressionsanlayse mit Affymetrix-*Microarrays* an Hodgkin-Lymphom-Zelllinien (HL-Zelllinien) im Vergleich zu normalen B-Zellen durchgeführt. Bei den HL-Zelllinien, handelt es sich um kontrollierte, etablierte und kultivierte Zelllinien die von Patienten mit HL stammen.
Thomas Hodgkin beschrieb 1832 zum ersten Mal das HL [Hodgkin, 1832]. Fast 70 Jahre später wurde es von Sternberg [Sternberg, 1898] und Reed [Reed, 1902] weiter charakterisiert. In der westlichen Welt ist das HL mit 2 - 4 Fällen pro 100.000 eine seltene Erkrankung, aber mit einem Anteil von 30 % eines der häufigsten malignen Erkrankungen des lymphatischen Systems. Das HL lässt sich in zwei Identitäten unterteilen: das klassische HL (cHL) und das noduläre lymphozyten-prädominante HL (NLPHL). 95 % aller HLs sind der Identität cHL zuzuordnen. Die Besonderheit, die beide Identitäten charakterisiert, sind die wenigen großen neoplastischen (neubildenden) Zellen, die sich in einem zellulären Infiltrat nicht neoplastischer Zellen befinden [Jaffe et al., 1998]. Bei cHL spricht man dann von HRS-Zellen und bei NLPHL von L&H-Zellen. Weniger als 1 % des infiltrierten Gewebes machen die HRS-Zellen oder L&H-Zellen aus [Hansmann et al., 1999; Weiss et al., 1999]. Allerdings unterscheiden sich beide Identitäten des cHLs u. a. im Immunphänotyp, in der Epidemiologie und in klinischen und genetischen Eigenschaften.

Das cHL wird vor allem in vier Subtypen unterteilt: nodulär sklerotisierend (NS; 60 - 80 % der cHL), mischzellig (MC; 15 - 30 % der cHL), lymphozytenarm (LD; 1 % der cHL) und lymphozytenreich (IrcHL; 6 % der cHL) [Harris, 1999]. Moderne Therapiekonzepte haben zu einem sehr guten Therapie- und Heilungserfolg beim HL geführt unabhängig vom Subtyp. Eine genetische Disposition liegt nahe, da ein gehäuftes Auftreten des HLs in einigen Familien und ein 99-fach höheres Risiko einer Erkrankung bei eineiigen Zwillingen vermutet wird [Grufferman and Delzell, 1984]. Epidemiologischen Studien zeigten eine Abhängigkeit des Erkrankungsrisikos von Geschlecht, sozialem Lebenssituation und ethnischen Hintergrund [Gutensohn and Cole, 1980].
Die HL-Zelllinien dienten bei Küppers et al. als repräsentatives Modell zum primären cHL des lymphatischen Gewebes. Die HL-Zelllinien L428 und HDLM2 stammen von Patienten mit HL mit dem Subtyp NS und die HL-Zelllinien KMH2 und L1236 vom Subtyp MC. Die HDLM2 stammen von T-Zellen, die anderen drei stammen von B-Zellen ab. Alle normalen B-Zellen der verschiedenen Subtypen (GC-B-Zellen, Gedächtnis B-Zellen, Naive B-Zellen und PC-Zellen) wurden aus menschlichen Tonsillen („Mandeln“) gewonnen.
Der Datensatz von Küppers et al. wurde mit dem Affymetrix *Microarray* Typ HGU95A erstellt. Auf diesem *Microarray* befinden sich 12626 *Probesets*. Verwendet wurde das Standard-Affymetrix Protokoll (Einsatzmenge ungefähr $1 \cdot 10^6$ Zellen pro *Microarray*).

2.1.2 Datensatz von Piccaluga et al.

Der zweite reanalysierte Datensatz ist von Piccaluga et al. [Piccaluga et al., 2007]. In dieser Arbeit wurde eine bestimmte Gruppe von T-Zell-Lymphomen im Vergleich zu normalen T-Zellen untersucht. Bei den T-Zell-Lymphomen handelt es sich um sogenannte periphere T-Zell-Lymphome Typ unspezifisch (PTZL/U). In diese Gruppe fallen alle T-Zell-Lymphome, die nicht weiter charakterisiert werden können. Die PTZL/U-Lymphome sind die am meisten verbreiteten T-Zell-Lymphome und zeigen morphologisch und phänotypisch eine sehr große Heterogenität. Außerdem sprechen diese Lymphome nur schwach auf die Therapie an und haben eine schlechte Prognose. 60 - 70 % der PTZLs fallen in die Gruppe der PTZL/U und können nicht mit der Morphologie, mit dem Phänotyp oder durch molekulare Untersuchungen klassifiziert werden [Harris et al., 1994]. Klinisch gesehen ist das PTZL/U das ag-

gressivste NHL. Insgesamt wurden 28 PTZL/U und 20 normle T-Zell-Populationen (bestehend aus CD4 und CD8) verwendet. Außerdem wurden zusätzlich zwei weitere Subtypen des T-Zell-Lymphoms mit jeweils 6 *Microarays* verwendet (AITL = angioimmunoblastisches T-Zell-Lymphom, ALCL = anaplastisches großzelliges Non-Hodgkin-Lymphom). Die normalen T-Zell-Populationen stammen entweder aus dem Blut oder aus Tonsillen gesunder Spender.
Für diese Studie wurde der HGU133Plus2.0 *Microarray*-Typ von Affymetrix verwendet. Dieser hat 54676 *Probesets* auf dem *Microarray*, was ungefähr 20 000 einzelnen Genen entspricht. Verwendet wurde das Standardprotokoll von Affymetrix.
Piccaluga et al. haben sogenannte Ganzschnitte für die einzelnen Tumorprofile verwendet. Hier wurde ein ganzer Querschnitt eines Tumorareals gewählt, der zum Teil große Mengen an Kontaminationen enthält. Mit Kontamination sind hier alle Zellen gemeint, die nicht Tumorzellen sind.

2.1.3 Test-Datensatz

Bevor der Datensatz von Brune et al. [Brune et al., 2008] generiert wurde, musste eine geeignete Methode etabliert werden, um aus dem zur Verfügung stehenden Material überhaupt Genexpressionsanalysen mit *Microarrays* von Affymetrix durchführen zu können.
Die Zellen des cHLs wurden einzeln per Lasermikrodissektion aus dem Gewebeschnitt isoliert. Deshalb konnte nur eine Menge von maximal 1 000 Zellen pro Patient isoliert werden. Die Laser-Mikrodissektion und die *Pressure-Catapulting*-Technik (LMPC) mit UV-Laser der Firma P.A.L.M. machten es erstmals möglich, einzelne Zellen so aus dem Gewebe zu isolieren, dass die DNA- und RNA-Qualität ausreichend gut war, um weitere Experimente durchzuführen. Zum Zeitpunkt der Datenerhebung gab es nur das Standardprotokoll von Affymetrix und anderen Firmen, deren Protokolle aber mindestens $1 \cdot 10^6$ Zellen benötigen.
Um trotzdem Genexpressionsanalysen an nur 1 000 Zellen durchführen zu können, musste eine neue mRNA-Isolierungs- und mRNA-Amplifikationsmethode gefunden werden. Nachdem eine geeignete Methode gefunden wurde [Brune et al., 2008] testete man zunächst an einem kleinen Datensatz die Qualität des neuen Protokolls. Hierfür wurden eine HL-Zelllinie und ein Burkitt-Lymphom-Probe verwendet. Da in Folge des neuen Protokolls zwei „Vermehrungsschritte“ (Invitrotranskription) der mRNA durchgeführt wurden, ist unklar, ob die Daten nicht zu stark durch Artefakte

maskiert sind. Das *Probe*-Design von Affymetrix ist auf dem verwendeten *Microarray* HGU133Plus2.0 nicht für solche vorbehandelten Zellproben vorgesehen. Zwar konnten die Daten erfolgreich mit der VSN-Methode von Huber et al. [Huber et al., 2002, 2003] analysiert und publiziert [Brune et al., 2008] werden, die Frage nach einer geeigneten Anpassung der Daten an das *Microarray*-Design blieb allerdings offen. Der Testdatensatz eignet sich, um eine Methode zu entwickeln, die das *Microarray*-Design mit einfließen lässt. Duplikate einer HL-Zelllinie und eines BL-Falles wurden mit dem neuen Protokoll und dem Standardprotokoll von Affymetrix durchgeführt, um beide zu vergleichen.

2.1.4 Datensatz von Brune et al.

Der Datensatz von Brune et al. [Brune et al., 2008] wurde mit etablierten statistischen Verfahren ausgewertet, allerdings ohne auf das *Microarray*-Design und die zusätzliche Problematik mit der geringen Ausgangsmenge an mRNA einzugehen.
Der Datensatz von Brune et al. enthält zwei Gruppen: Zum einen die mikrodissizierten Lymphomzellen, die entweder in Zellgruppen oder als Einzelzellen isoliert wurden, zum anderen durch Fluoreszenz-aktivierter Zellseperation (FACS) sortierte normale B-Zellen aus der Tonsille von gesunden Spendern. In der Arbeit von Brune et al. wurden Gene identifiziert, die bei der Pathogenese des NLPHL eine wichtige Rolle spielen. Zum Vergleich wurden die nicht-malignen Ursprungszellen, die GC-B-Zellen, verwendet. Der Vergleich von cHL und GC-B-Zellen wird in der vorliegenden Arbeit reanalysiert und mit den Standardverfahren verglichen.

2.2 Ablauf eines *Microarray*-Experiments

In Abbildung 2.1 ist schematisch dargestellt, wie ein Genexpressionsprofil aus Zellen erstellt wird. Im ersten Schritt wird aus einem Zellpool aus Tumorgewebe oder einer Zellkultur die mRNA extrahiert. Diese wird im zweiten Schritt vermehrt und in unserem Fall, mit Fluoreszenz markiert. Im dritten Schritt wird die markierte mRNA auf dem *Microarray*-Chip hybridisiert. Im letzten vierten Schritt wird die Fluoreszenz mit dem Affymetrix-Scanner gemessen. Genauere Informationen enthält die Affymetrix Homepage (`www.affymetrix.com`).

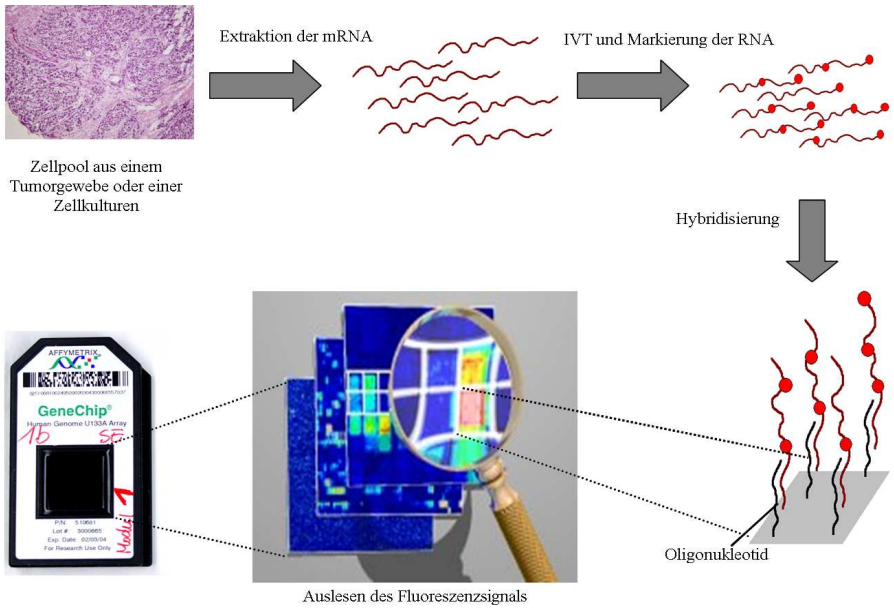


Abbildung 2.1: Ablauf eines Micorarray-Experiments

2.3 Technische Vorraussetzungen

Die statistische Auswertung wurde mit der frei verfügbaren Statistiksoftware R, Versionen 2.1 und 2.8, durchgeführt [Ihaka and Gentleman, 1996; Gentleman et al., 2004]. Außerdem wurden viele Pakete aus dem Bioconductor-Projekt benutzt (*affy*, *vsn*, *geneplotter*, *matchprobes* usw. [Gautier et al., 2004; Huber et al., 2002; Gentleman et al., 2004; Huber and Gentleman, 2004]). Bioconductor ist fester Bestandteil in der Analyse verschiedener, hochdimensionaler genomischer Daten geworden. Das liegt an der großen Anzahl implementierter Algorithmen und der Tatsache, dass die Software frei verfügbar ist und von herausragenden Wissenschaftlern gepflegt, und durch neue Pakete erweitert wird. Die Software ist nicht nur im akademischen Bereich verbreitet, sondern bietet auch für kommerzielle Software Schnittstellen zu Bioconductor. Für die Datenanalyse wurden ein Rechner mit Intel Architektur (3,2 GHz CPU) und 4 GB RAM mit Betriebssystem Windows XP und ein Linux-Server mit 16 GB RAM verwendet. Vor allem der VSN-Algorithmus von Huber et al. [Huber et al., 2002, 2003] benötigt bei entsprechender Chip-Anzahl viel Arbeitsspeicher.

2.4 Explorative Datenanalyse

2.4.1 Boxplot

Zur grafischen Visualisierung eines Sets von *n Microarrays* kann man Boxplots verwenden. Durch die grafische Dartsellung bekommt man eine gute Übersicht über die Verteilung der Signalintensitäten der einzelnen Chips. Wichtige Streuungs- und Lagemaße (Median, die zwei Quartile und Extremwerte) zeigten schnell, in welchem Bereich die Daten liegen und wie die Daten über diesen Bereich verteilt sind.
Die Box beinhaltet die mittleren 50 % der Daten. Sie wird durch das obere und untere Quartil begrenzt. Die Ausdehnung der Box repräsentiert die Interquartilrange (IQR). Sie ist ein Maß für die Streuuung der Daten und wird aus der Differenz des oberen und unteren Quartils berechnet. *Whiskers* sind die Datenpunkte außerhalb der Box. Die Länge der Whiskers sind nicht unbedingt festgelegt. John Tukey [Tukey, 1977] legte fest, dass die *Whiskers* maximal die 1,5 fache der IQR haben. Allerdings endet der *Whisker* nicht genau an diesem Punkt, sondern an dem jeweils äußersten Datenwert, der noch innerhalb dieser Grenze liegt. Die Daten außerhalb der *Whiskers* werden als Ausreißer definiert (Abb.2.2).

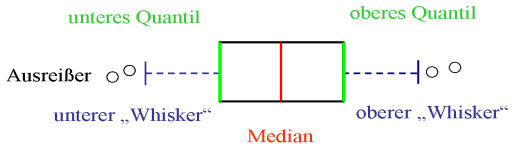


Abbildung 2.2: Darstellung eines horizontalen Boxplots

2.4.2 Histogramm

Ein Histogramm kann einen guten Überblick über die Verteilung der Daten geben (Abb.2.3). Zu bedenken ist allerdings, dass das Aussehen von Histogrammen oft stark von der Klassenanzahl und bei wenigen Klassen auch von der Platzierung der Klassengrenzen abhängt.

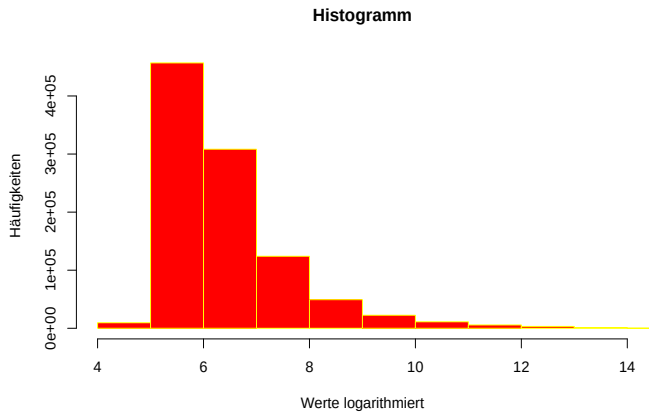


Abbildung 2.3: Histogramm-Beispiel

2.4.3 Q-Q-Plot

Der Quantil-Quantil-Plot (Q-Q-Plot) stellt die Quantile zweier statistischen Variablen grafisch gegenüber. Dabei werden die Beobachtungsmerkmale zweier Merkmale der Größe nach sortiert, als Wertepaare zusammengefasst und grafisch in einem Koordinatensystem dargestellt. Ergeben die Punkte annähernd eine Gerade $y = x$ kann man davon ausgehen, dass beide Merkmale gleich verteilt sind. Beim Vergleich mit den Quantilen der Standard-Normalverteilung gilt zusätzlich, dass die Gerade $y = a + bx$ der Normalverteilung mit (μ, σ^2) entspricht.

2.5 Präprozessierungsalgorithmen für die Normalisierung von Affymetrix-*Microarray*-Rohdaten

2.5.1 Ziel der Präprozessierung

Aufgabenstellung:
Aus den Bilddaten von Affymetrix sollen mit geeigneten Verfahren Expressionswerte für die *Probesets* ermittelt werden. Typischerweise lässt sich die Präprozessierung in vier Schritte unterteilen, die sich je nach angewandter Methode stark unterscheiden. Zunächst wird erläutert, aus welchem Grund die einzelnen Präprozessierungsschritte so wichtig sind.

1. **Die Hintergrundkorrektur:**
   Ein Hintergrundsignal kann durch Eigenfluoreszenz der Reagenzien, Streulicht oder sonstige technischer Probleme entstehen. Selbst nach dem Scannen eines Arrays ohne RNA (z. B. mit destilliertem Wasser) wird man ein Fluoreszenzsignal messen können. Ziel der Hintergrundkorrektur ist, die Stärke dieses artifiziellen Signals zu schätzen und von den gemessenen Werten zu eliminieren. Ist dieses Verfahren erfolgreich, sollte ohne RNA auf dem *Microarray* keine Fluoreszenzsignal mehr meßbar sein [Gautier et al., 2004].

2. **Normalisierung:**
   Unabhängig davon, welche der vielen verschiedenen Normalisierungsmethoden verwendet wird, alle haben das Ziel, die n-Arrays eines Experiments vergleichbar zu machen. Besteht ein Experiment aus mehreren Arrays, so ist die Variabilität der Messwerte sehr groß. Dabei muss man zwischen der biologischen (Unterschied zwischen Tumorzellen und normalen Zellen) und der technischen Variabilität (Benutzung von einem oder verschiedenen Protokollen) unterscheiden. Die biologische Variabilität ist die für den Wissenschaftler interessante. Die technische Variabilität ist hier eher störend und kann dazu führen die biologische Variabilität zu verändern. Ziel ist deshalb, durch die Normalisierung die technische Variabilität zu minimieren. Die Ursachen für die technische Variabilität können vielfältig sein. Einflussreichste Quelle ist wahrscheinlich die Durchführung des Experiments, aber auch die Herstellung der *Microarrays* und der Laborreagenzien sind eine Fehlerquelle. Schon geringfügige Änderungen des Protokolls (z. B. Ausgangsmenge mRNA) können dabei zu massiven Unterschieden des Fluoreszenzsignals führen.
   Mit Hilfe von Boxplots, QQ-Plots und Histogrammen kann man wesentliche Unterschiede der verschiedenen Experimente erkennen. Ohne Normalisierung käme es hier zu differentiell exprimierten Genen.
   Die Normalisierungsverfahren lassen sich in zwei Gruppen aufteilen. Es gibt Verfahren, die einen *Microarray* als Basis verwenden, mit dessen Werten die anderen n - 1 Arrays normalisiert werden. Andere Methoden benötigen für die Normalisierung alle Arrays. Es gibt bis heute keinen „Goldstandard“ für die Normalisierungmethode. Allerdings geben Boes und Neuhäuser [Boes and Neuhäuser, 2005] einen guten Überblick über verschiedene Normalisierungsmethoden, inklusive einiger Evaluationen.

3. ***Probe*-spezifische Hintergrundkorrektur:**

   Nach der Normalisierung ist zu entscheiden, ob und wie die PM-Werte und MM-Werte miteinander verrechnet werden. Ziel der PM-Adjustierung ist, einen Wert für das *Probe Pair* zu berrechnen. Bolstad et al. [Bolstad et al., 2003] haben als Erste vorgeschlagen, nur die PM-Werte für die PM-Adjustierung zu verwenden.

4. **Aggregationsmethoden:**

   Nach der PM-Adjustierung hat man einen Wert y_{ijk} für jedes *Probeset* berechnet, dabei ist i der *Microarray*, j das Gen und k das *Probe*. Ziel der Aggregationsmethode ist es, aus den Einzelwerten eines *Probesets* einen Wert x_{ij} zu berechnen.

Die detaillierte Beschreibung der in dieser Arbeit verwendeten Modelle folgt diesem Kapitel. Es gibt außer den genannten eine Vielzahl weiterer Modelle für die Präprozessierung.

2.5.2 Das Modell von Affymetrix

Affymetrix [Affymetrix, 2002] hat einen eigenen Algorithmus entwickelt und diesen in eine Software Namens *Microarray Suite* (MAS) implementiert. Mit diesem Algorithmus werden die vier Schritte der Präprozessierung durchgeführt. Die MAS gab es zunächst in den Versionen 4.0 und 5.0. Bei Version 4.0 waren negative Signalwerte möglich. Das ist aus biologischer Sicht nicht sinnvoll und für weitere Analysen ist es problematisch, mit negativen Werten umzugehen. Bei Version 5.0 wurde dies geändert. Heute ist die Software von Affymetrix (Expression Console 1.1 = EC1.1) frei verfügbar und neben dem MAS 5.0-Algorithmus sind weitere Algorithmen (unter anderem RMA) implementiert.

MAS-4.0-Algorithmus

Die Methode zur Berechnung der Expression bei MAS 4.0 wird auch *„Average Difference“* genannt. Diese mittlerweile veraltete Methode wurde bei dem Datensatz von Küppers et al. [Küppers et al., 2003] verwendet. Deshalb hier eine kurze Beschreibung:
Der MAS-4.0-Algorithmus basiert auf der einfachen Idee, dass die Genexpression der

Durchschnitt der Differenzen der PM-Werte und der dazugehörigen MM-Werte eines *Probeset* ist. Ziel von Affymetrix für die *Probe*-spezifische Hintergrundkorrektur war das Einstellen von PM-MM. Bis November 2002 war dieser Algorithmus Standard bei Affymetrix. Um einen robusten Durchschnittswert für die einzelnen *Probe Pairs* zu gewährleisten, werden zunächst mehrere Werte aus der Durchschnittsberechnung herrausgenommen.
Minimalen und Maximalen Wert der y_{ijk} über alle *Probe Pairs* k eines Gens j im i-ten *Microarray* werden im ersten Schritt zunächst ignoriert, da es sich um Ausreißer handeln könnte. Aus den verbleibenden Werten werden der Mittelwert $\overline{y}_{ij}$ und die Standardabweichung $s(y_{ij})$ berechnet. Im letzten Schritt werden alle Werte y_{ijk} gelöscht, die mehr als drei Standardabweichungen vom Mittelwert zeigen. Wenn die gelöschten Werte nicht mehr als drei Standardabweichungen vom Mittelwert entfernt abweichen, können sie dabei wieder berücksichtigt werden. Die Menge A_{ij} wird also aus den verbliebenden Werten der *Probe-Pairs* gebildet. Dieses wird als *„Average-Difference“*, oder kurz AvgDiff, bezeichnet und lässt sich für jedes Gen j eines Chips i folgendermaßen berechnen:

$$AvgDiff_{ij} = \frac{1}{N(A_{ij})} \sum_{k \in A_{ij}} y_{ijk} \tag{2.1}$$

Dabei ist $N(A_{ij})$ die Anzahl der Werte in der Menge A_{ij}.
Für die Analyse mit MAS 5.0 werden zunächst die Werte der *Probe-Cells* eines Chips benötigt. Als Nächstes wird eine Hintergrundkorrektur durchgeführt. Für die Hintergrundkorrektur wird der *Microarray* in K gleich große Segmente $Z_k (k = 1, \ldots, K$, Voreinstellung K=16) unterteilt. Kontroll-*Probe-Cells* und maskierte *Probe Cells* bleiben bei der Kalkulation unberücksichtigt. Die *Probe-Cell* Intensitäten werden nach Größe sortiert. Die kleinsten 2 % repräsentieren das Hintergrundsignal b für das Segment (bZ_k). Die Standardabweichung von den kleinsten 2 % *Probe Cell*-Intensitäten wird berechnet und als Schätzer für die Hintergrundvariabilität n für jedes Segment (nZ_k) verwendet.
Um einen nahtlosen Übergang zwischen den Segmenten zu gewährleisten, wird der räumliche Abstand von jedem *Probe Cell* auf dem *Microarray* zu den verschiedenen K-Segmentschwerpunkten mit $d_k(x, y)$ $(k = 1, \cdots, L)$ berechnet. Als Gewicht für die Berechnung des Hintergrunds jeder *Probe Cell* wird der reziproke Abstand der

jeweiligen *Probe Cell* zu den K-Segmentschwerpunkten berechnet. Deshalb hat ein Segment mit naheliegendem Schwerpunkt ein größeres Gewicht bei der Bestimmung des Hintergrundsignals als ein von der *Probe-Cell* weit entferntes Segment. Die ermittelten Gewichte werden mit $w_k(x, y)$ bezeichnet. Sie können berechnet werden:

$$w_k(x, y) = \frac{1}{d_k^2(x, y) + i} \tag{2.2}$$

i is dabei eine Konstante (Voreinstellung = 100), die verhindert, dass der Nenner Null wird. Mit dem berechneten Segmenthintergrundsignal b_Z und den Gewichten $w_k(x, y)$ kann man das Hintergrundsignal jeder einzelnen *Probe-Cell* berechnen:

$$b(x, y) = \frac{1}{\sum_{k=1}^{K} w_k(x, y)} \sum_{k=1}^{K} w_k(x, y) bZ_k \tag{2.3}$$

Um hintergrundbereinigte Werte für jedes *Probe-Cell* zu erhalten, subtrahiert man die in (2.3) geschätzten Hintergrundwerte von den jeweils gemessenen *Probe*-Intensitäten. Die Schätzung des Hintergrundsignals kann größer als die gemessene *Probe*-Intensität sein. Um keine negativen Werte zu erhalten, werden die hintergrundbereinigten Werte folgendermaßen berechnet:

$$A(x, y) = max(I'(x, y) - b(x, y), 0.5 \cdot (x, y)), \tag{2.4}$$

wobei $I'(x, y) = max(I\prime(x, y), 0.5)$ ist. I(x,y) repräsentiert die PM_{ijk} bzw. die MM_{ijk} mit den jeweiligen Koordinaten x und y.
Bei Affymetrix findet eine Normalisierung auf einer anderen Ebene statt als bei den anderen aufgeführten Verfahren. Die Normalisierung wird nicht auf *Probe*-Ebene durchgeführt, sondern mit den schon aggregierten Werten des *Probesets*.
Ziel ist es, jedem *Microarray* mit einem konstanten Wert zu multiplizieren, sodass alle Mittelwerte gleich dem Mittelwert des Basis-Chips sind. Der erste *Microarray* dient dabei als Basis-Array. Die übrigen n-1-Arrays werden dann normalisiert: Angenommen $z_{\text{base}_{jk}}$ ist der Vektor der Expressionswerte des Basis-Chips und $\overline{z}_{i}..$ ist

der Mittelwert der Werte von $z_i(\overline{z}_{i\cdot\cdot} = \frac{1}{\sum_{j=1}^{m} l_j} \sum_{j=1}^{m} \sum_{k=1}^{l_j} z_{ijk})$. Dann berechnet man β_i wie folgt:

$$\beta_i = \frac{z_{\text{base}_{jk}}}{\overline{z}_{i\cdot\cdot}} \tag{2.5}$$

Die Expressionswerte der normalisierten Chips erhält man folgendermaßen:

$$z_{ijk}^{norm} = \beta_i z_{ijk} \tag{2.6}$$

Das beschriebene Skalieren ist äquivalent zu einer linearen Anpassung des Basis-Arrays an jeden anderen *Microarray* mit einem Achsenabschnitt 0. Die MM-Signale repräsentieren das Ausmaß der unspezifischen Bindung und damit die Hybridisierung von Nicht-Zielsequenzen auf dem *Microarray*. Verwendet man diese Methode kommt es zu negativen Signalwerten. Biologisch ist dies Unsinn, weil ein Gen entweder ausgeprägt ist oder nicht, negativ ausgeprägt kann es nicht sein. Affymetrix änderte deshalb ihren Algorithmus [Affymetrix, 2001]. Dieser Algorithmus — auch *Microarray Suite* 5 (MAS5) genannt — verhindert negative Werte nach der Subtraktion der PM-Werte von den MM-Werten.
Hat ein MM einen größeren Signalwert als das dazugehörige PM und kommt es deshalb zum negativem Signalwert, ist ein idealer MM (IM) zu wählen, der kleiner als das PM ist. Deshalb berechnet man zuerst für jedes *Probeset* einen *Probe* spezifischen Hintergrund, der das durchschnittliche Verhältnis zwischen PM und MM in einem *Probeset* angibt:

$$\begin{aligned} SB_{ij} &= T_{bi}(\log_2(PM_{ijk}) - \log_2(MM_{ijk})), k = 1, \cdots, l_j \\ &= T_{bi}\left(\log_2 \left(\frac{PM_{ijk}}{MM_{ijk}} \right), k = 1; \cdots, l_j \right), \end{aligned} \tag{2.7}$$

dabei ist mit T_{bi} der Tukey's Biweight [Affymetrix, 2002] gemeint. Der Tukey's Biweight ist ein robuster Schätzer für den Lageparameter einer Funktion. Die Funktion *g(z)* und ihre Ableitung sind gegeben durch:

$$g(z) = \begin{cases} \frac{a^2}{6} \cdot [1 - (1 - \frac{z^2}{a^2})^3] & \text{für } |z| \leq a \\ \frac{a^2}{6} & \text{für } |z| > a \end{cases} \tag{2.8}$$

$$g'(z) = \begin{cases} z(1-\frac{z^2}{a^2})^2 & \text{für } |z| \leq a \\ 0 & \text{für } |z| > a \end{cases} \tag{2.9}$$

Der Schätzwert für den Tukey's Biweight minimiert

$$\sum_i g(z_i - m) \tag{2.10}$$

für eine gegebene Stichprobe $z_i, i = 1, \cdots, n$.
M ist der gesuchte Lageparameter. Im Paket *affy* wird m zunächst als Median geschätzt und dann für die $z_i - m$ wie folgt fortgefahren: Da h die Ableitung von g ist, gilt

$$\sum_i h(z_i) = 0 \tag{2.11}$$

und die Gewichtefunktion von $w(z) = h(z)/z$ für Tukey's Biweight ergibt:

$$w(z) = \begin{cases} (1-\frac{z^2}{a^2})^2, & \text{für } |z| \leq a \\ 0, & \text{für } |z| > a \end{cases} \tag{2.12}$$

Datenabhängig von der Anzahl der Ausreißer wird der Parameter a gewählt. Der IM berechnet sich deshalb:

$$IM_{ijk} \begin{cases} MM_{ijk} & \text{für } MM_{ijk} < PM_{ijk} \\ \frac{PM_{ijk}}{2^S B_{ij}} & \text{für } MM_{ijk} \geq PM_{ijk} \text{und} SB_{ij} > \tau \\ \frac{PM_{ijk}}{2^{\tau/(1+((\tau - SB_{ij})/\nu))}} & \text{für } MM_{ijk} \geq PM_{ijk} \text{und} SB_{ij} \leq \tau \end{cases} \tag{2.13}$$

Affymetrix schlägt Standardeinstellungen für τ und ν vor, die bei 0.03 bzw. 10 liegen. Nachdem man den IM berechnet hat, lassen sich PM und MM zusammenfassen:

$$y_{ijk} = \max(PM_{ijk} - IM_{ijk}, \kappa), \tag{2.14}$$

wobei Affymerix für κ eine Standardeinstellung vorgibt ($\kappa = 2^{-20}$). Nach der *Probe*-spezifischen Hintergrundkorrektur der *Probe-Pair*-Werte werden diese zunächst logarithmiert ($PV_{ijk} = log_2(V_{ijk})$), um die Varianz zu stabilisieren. Mit Tukey Biweight [Affymetrix, 2002] werden diese logarithmierten Werte zu einem Expressionswert pro *Probeset* berechnet:

$$\text{MAS5 Signal}_{ij} = T_{bi}(PV_{ij1}, \cdots, PV_{ijl_j}) \tag{2.15}$$

Das oben berechnete MAS5 Signal wird dann abschließend zur Basis zwei exponiert:

$$x_{ij} = 2^{\text{MAS5 Signal}_{ij}} \tag{2.16}$$

2.5.3 Modell von Klein et al.

Klein et al. [Klein et al., 2001] haben zur Präprozessierung des Datensatzes von Küppers et al. die Methode von Affymetrix (MAS 4.0) mit einer globalen Skalierung verwendet. Alle negativen Expressionswerte und sehr kleine positive Werte wurden durch den Wert 20 ersetzt. Eine Definition ab welcher Grenze ein Wert als „klein positiv“ klassifiziert wird wird, geben Klein et al. nicht an. Piccaluga et al. verwenden für die Analyse des Datensatzes die Software MAS 5.0.

2.5.4 Das VSN-Modell von Huber et al.

Erst dadurch, dass nicht alle Gene nach einem Experiment differentiell exprimiert sind, wird eine Normalisierung möglich. Denn nur *Probesets*, die keine differentielle Expression zeigen, lassen technische Variabilität erkennen. Bei den nachfolgenden Präprozessierungsmodellen werden ausschließlich die PM-Werte verwendet. Die MM-Werte werden — falls nötig — (so bei Irizarry et al.[Irizarry et al., 2003a]) nur für die Hintergrundkorrektur berücksichtigt. Vor der Normalisierung mit VSN erfolgt keine Hintergrundkorrektur. VSN selbst schätzt und subtrahiert einen Gesamthintergrundschätzer (pro *Microarray*), sodass eine zusätzliche Hintergrundkorrektur nur bei lokaler Variabilität über einen *Microarray*, z. B. räumlicher Gradient, sinnvoll wäre.
Der *VSN*-Algorithmus lässt sich in zwei Komponenten aufteilen: Bei der ersten handelt es sich um eine affine Transformation, um die systematischen experimentellen Faktoren, wie die Markierungseffzienz und die Detektionssensivität, zu kalibrieren. Bei der zweiten handelt es sich um eine $g\log_2$-Transformation, um die Varianz zu stabilisieren [Huber et al., 2002]. Beide Komponenten werden im Folgenden beschrieben.
Das *VSN*-Modell basiert auf dem Fehlermodell von Rocke und Durbin [Rocke and Durbin, 2001]. Bei diesem Fehlermodell wird angenommen, dass die gemessene Signalintensität y eine Realisierung der Zufallsvariablen Y ist, die wie folgt definiert ist:

$$Y = \alpha + \beta e^{\eta} + \nu, \tag{2.17}$$

α ist dabei der *Offset* und β die tatsächlich gemessene Expression. e^{η} und ν sind der multiplikative und der additive Fehlerterm, wobei η mit $N(0,1)$ und ν mit $N(0,s_{\nu})$ verteilt ist.
Als Erwartungswert von Y ergibt sich damit:

$$\begin{aligned} E(Y) &= E(\alpha + \beta e^{\eta} + \nu) \\ &= E(\alpha) + E(\beta e^{\eta}) + E(\nu) \\ &= \alpha + \beta m_{\eta} \end{aligned} \tag{2.18}$$

wobei m_{η} der Erwartungswert von e^{η} ist. Für die Varianz von Y folgt daraus:

$$\begin{aligned} Var(Y) &= E(Y - E(Y))^2 \\ &= E(\alpha + \beta e^\eta + \nu - (\alpha + \beta m_\eta))^2 \\ &= E(\nu + \beta(e^\eta - m_\eta))^2 \\ &= E(\nu^2 + 2\nu\beta(e^\eta - m_\eta) + \beta^2(e^\eta - m_\eta)^2) \\ &= E(\nu^2) + \beta^2 E(e^\eta - m_\eta)^2 \\ &= s_\nu^2 + \beta^2 s_\eta^2 \end{aligned} \tag{2.19}$$

Dabei sind s_ν^2 und s_η^2 die Varianzen von ν und e^η.
Wenn man die Gleichung 2.18 nach β umformt,

$$\beta = \frac{E(Y) - a}{m_\eta} \tag{2.20}$$

und β in die Gleichung 2.19 einsetzt, so erhält man die Varianz als Funktion und damit die quadratische Abhängigkeit der Varianz vom Mittelwert:

$$v(u) = (c_1 u + c_2)^2 + c_3, \quad \text{mit } c_3 > 0 \tag{2.21}$$

v(u) ist die Varianz als Funktion vom Erwartungswert $u(\rightarrow E(Y))$, wobei

$$c_1 = \frac{s_\eta}{m_\eta}, \quad c_2 = -\frac{\alpha s_\eta}{m_\eta}, \quad c_3 = s_\nu^2 \tag{2.22}$$

gegeben ist. Gesucht wird eine Transformation *h(y)*, für die *Var(h(y)) = konstant* gilt. Die Methode, die man in diesem Fall anwendet ist die Delta-Methode (2.4).
Bei dieser Methode wird *h(y)* um den Erwartungswert u mit den ersten beiden Gliedern der Taylor-Reihe approximiert:

$$\begin{aligned} h(y) &\approx h(u) + (y - u)h'(u) \\ &\approx h(u) - uh'(u) + yh'(u) \end{aligned} \tag{2.23}$$

Für den Erwartungswert folgt:

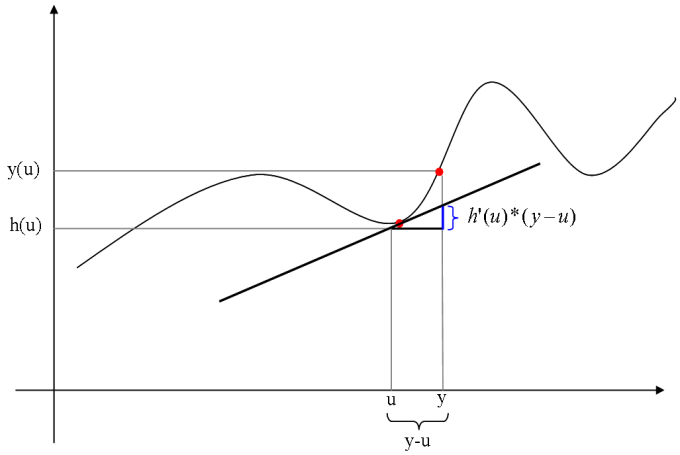


Abbildung 2.4: Schematische Darstellung der Delta-Methode

$$\begin{aligned} E(h(Y)) &\approx E(h(u) - uh'(u) + Yh'(u)) \\ &= h(u) - uh'(u) + h'(u)E(Y) \\ &= h(u) - uh'(u) + uh'(u) \\ &= h(u) \end{aligned} \tag{2.24}$$

Damit gilt für die Varianz:

$$\begin{aligned} Var(h(Y)) &= E(h(Y) - E(h(Y)))^2 \\ &\approx E(h(u) - uh'(u) + Yh'(u) - h(u))^2 \\ &= E(Yh'(u) - uh'(u))^2 \\ &= E((h'(u))(Y - u))^2 \\ &= h'(u)^2 \cdot E((Y - u))^2 \\ &= h'(u)^2 \cdot Var(Y) \\ &= h'(u)^2 \cdot v(u) \end{aligned} \tag{2.25}$$

Da die Varianz von *h(Y)* konstant bleiben soll, ergibt sich zusammen mit 2.21:

$$1 = h'(u)^2 \cdot v(u) \tag{2.26}$$

$$h'(u) = \frac{1}{\sqrt{v(u)}} \tag{2.27}$$

$$h(y) = \int^{y} \frac{1}{\sqrt{v(u)}} du = \int^{y} \frac{1}{\sqrt{(c_1 u + c_2)^2 + c_3}} du \tag{2.28}$$

$$h(y) = \gamma \cdot \text{arsinh}(\text{a} + \text{by}) \tag{2.29}$$

dabei entspricht $\gamma = c_1^{-1} = \frac{m_\eta}{s_\eta}$, $a = \frac{c_2}{\sqrt{c_3}} = -\frac{\alpha s_\eta}{s_\nu m_\eta}$ und $b = \frac{c_1}{\sqrt{c_3}} = \frac{s_\eta}{a_\nu m_\eta}$. Da γ nur ein Skalierungsfaktor ist, kann dieser unberücksichtigt bleiben [Tibshirani, 1988]. Die Transformation durch die Umkehrfunktion des Areasinus Hyperbolicus führt zur gleichmäßigen Verteilung der Varianz über alle Intensitätswerte. Dabei gilt folgender Zusammenhang: $\int \frac{1}{\sqrt{x^2+1}} = \text{arsinh(x)}$.
Wendet man diese Varianzstabilisierungstransformation auf reale Daten an, muss man bedenken, dass die Parameter a und b für jeden *Microarray* unterschiedlich sind. Für jeden *Microarray* i, für $i \in \{1, \cdots, d\}$ ergibt sich die folgende Varianzstabilisierungstransformation:

$$h_i(y_{ki}) = \text{arsinh}(\text{a}_\text{i} + \text{b}_\text{i}\text{y}_\text{ki}) \tag{2.30}$$

Die Parameter a_i und b_i werden mit einem Maximum-Liklihood-Schätzers angenähert. Ein Vorteil dieser Transformation gegenüber dem Logarithmus ist das Fehlen der Singularität bei 0. In Abbildung 2.5 findet sich der Vergleich von Logarithmus-Transformation und arsinh-Transformation. Das Histogramm zeigt die Expressionswerte eines *Microarry* Experiments. Negative Expressionswerte sind aufgrund von Bild-Hintergrundkorrekturen möglich. Die arsinh-Transformation kann auch mit negativen Werten umgehen, für große Werte sind log und arsinh asymptotisch äquivalent [Horst Stöcker, 2007].
Beim Vergleich eines Experiments i mit einem Experiment j der transformierten Expressionsraten bei Experimenten k, für $k \in \{1, \cdots, n\}$ verwendet man:

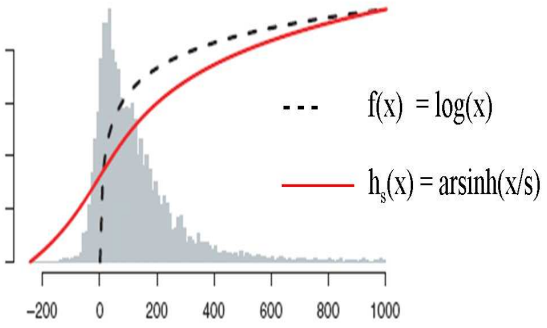


Abbildung 2.5: Funktionsverlauf von f(x) und $h_s(x)$ [Huber et al., 2002]

$$\triangle h_{k;ij} = \hat{h}_i(y_{ki}) - \hat{h}_j(y_{kj}), \quad \text{für } k = 1, \cdots, n \tag{2.31}$$

Große Signalwerte entsprechen dann ungefähr dem *Log-Ratio*, kleine Signalwerte nahe Null entsprechen eher der Differenz. Als nächstes ist eine Paramaterabschätzung der Parameter a_i und b_i der Transformation 2.30 aus den nicht differentiell exprimierten *Probesets* durchzuführen, die die geringsten Messschwankungen über alle *Microarrays* i zeigen. Wenn man davon ausgeht, das der Fehlerterm ε_{ki} normal verteilt ist, ergibt sich Folgendes:

$$h_i(Y_{ki}) = \mu_k + \varepsilon_{ki} \tag{2.32}$$

μ_k ist dabei der Erwartungswert des *Probesets* k über alle Samples. So werden die Parameter aus der Menge k der nicht differentiell exprimierten *Probesets* nach dem Maximum-Likelihood-Prinzip, geschätzt. K wird durch einen Algorithmus von Huber et al. gesucht, der auf der LTS (*least trimmed sum of squares*) Regression beruht. Die Parameter $\{a_i\}$, $\{b_i\}$, c, $\{\mu_k\}$ werden mit der Likelihood-Funktion maximiert:

$$\prod_{k=1}^{n} \prod_{i=1}^{d} p\left(\frac{h_i(y_{ki}) - \mu_k}{c}\right) h_i'(y_{ki}) \tag{2.33}$$

wobei $p(.)$ die Gauß'sche Dichte der Standardnormalverteilung darstellt. Diese Funktion ergibt sich aus dem Produkt der Wahrscheinlichkeiten der beobachteten Ergeb-

nisse unter der Annahme 2.32. Dabei nimmt man an, dass:

$$\begin{aligned}\hat{\mu}_k &= \frac{1}{d}\sum_{i=1}^{d} h_i(y_{ki}) \\ \hat{c}^2 &= \frac{1}{nd}\sum_{k=1}^{n}\sum_{i=1}^{d}(h_i(y_{ki}) - \hat{\mu}_k)^2 \\ \hat{c} &= \hat{c}(a_i, b_i)\end{aligned} \tag{2.34}$$

gegeben ist. Setzt man nun 2.34 in 2.33 ein, erhält man einen Profile Likelihood-Term. Die Maximierung der Profile Likelihood, d.h. einige Parameter werden als Funktionen anderer Parameter geschrieben, verkleinert die Anzahl der zu maximierenden Parameter:

$$\begin{aligned}pl(a_1, b_1, \cdots, a_d, b_d) = & \\ &= \prod_{k=1}^{n}\prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\hat{c}} exp\left(\frac{(h_i(y_{ki}) - \hat{\mu}_k)^2}{\frac{2}{nd}\sum_{k=1}^{n}\sum_{i=1}^{d}(h_i(y_{ki} - \hat{\mu}_k)^2}\right) h_i'(y_{ki}) \\ &= \frac{e^{nd/2}}{(2\pi)^{nd/2}\hat{c}^{nd}} \prod_{k=1}^{n}\prod_{i=1}^{d} h_i'(y_{ki})\end{aligned} \tag{2.35}$$

und wird als eine Funktion der Parameter a_i und b_i dargestellt. Allerdings sucht man das Maximum von dem „logarithmierten Profile Likelihood“ damit die Produkte in Summen umgewandelt werden können:

$$\begin{aligned}pl(a_1, b_1, \cdots, a_d, b_d) &= -nd\log\,\hat{c} + \sum_{k=1}^{n}\sum_{i=1}^{d}\log\, h_i'(y_{ki}) \\ &= -\frac{nd}{2}\log\left(\sum_{k=1}^{n}\sum_{i=1}^{d}(h_i(y_{ki}) - \hat{\mu}_k)^2\right) \\ &+ \sum_{k=1}^{n}\sum_{i=1}^{d}\log\, h_i'(y_{ki})\end{aligned} \tag{2.36}$$

Aufgrund der Monotonie von *log* ist dies zulässig. Durch eine numerische Maximierung von 2.35 können nun die Parameter a_i und b_i ermittelt werden. Allerdings reagiert der Maximum-Likelihood-Schätzer empfindlich auf differentiell exprimierte

Probesets. Um den Maximum-Likelihood-Schätzer robust gegen Ausreißer zu machen, wird eine Modifikation per LTS-Regression durchgeführt.
Die LTS-Regression ist eine lineare Regression, die die Summe der Quadrate der Abstände über alle nicht differentiell exprimierten *Probesets* minimiert. Gegenüber der einfachen linearen Regression (LS Regression) bietet LTS den Vorteil, dass die Fehlerrate nicht durch Ausreißer verzerrt wird. Dies gilt allerdings nur solange die Anzahl der Ausreißer auf maximal $1-q_{lts}$ beschränkt ist. Dabei repräsentiert $1-q_{lts}$ den erwarteten Anteil der nicht differentiell exprimierten *Probesets* mit einer relativen Häufigkeit von 0.5 bis 1.
Um eine Verbindung zwischen LTS und Maximum-Likelihood-Schätzer zu erreichen, werden die Summen über alle $k = 1, \cdots, n$ von Formel 2.36 durch die Summen von $k \in K$ ersetzt. Durch folgende iterative Prozedur erhält man die Menge der nicht differenten *Probesets* K.

1. Über alle *Probesets* werden die Parameter geschätzt und die Expression transformiert.

2. Die *Probesets* werden nach ihren Erwartungswerten sortiert und in 10 Quantile aufgeteilt.

3. Für alle *Probesets* eines Quantils wird der quadratische Fehler berechnet und nach Fehlergröße sortiert.

4. Für weitere Berechnungen nutzt man von jedem Quantil den ersten $qlts$-Anteil.

5. Die Parameter, die durch eine Schätzung über die zuletzt erzielten *Probesets* gewonnen wurden, werden für die nächste Iteration verwendet.

Die Prozedur wird solange wiederholt, bis die Parameter eine Stabilisierung erreichen. Die robuste Abschätzung ist gewährleistet, da aus jedem Quantil die gleiche Anzahl von *Probesets* verwendet wird. 10 Wiederholungen reichen in der Praxis aus, um die Parameter zu schätzen.
Nachdem die PM-Werte erfolgreich normalisiert wurden, muss im nächsten Schritt die Aggregation der einzelnen PM-Werte zu einem Expressionswert erfolgen. Huber et al. empfehlen die Methode Medianpolish, die Irizarry et al. in ihrem RMA-Algorithmus verwenden.

2.5.5 *Medianpolish*

Die Methode ist an das Verfahren Medianpolish von Tukey [Tukey, 1977] angelehnt. Ziel ist, für jedes *Probeset* ein additives Modell anzupassen. Dieses Modell besteht aus einem Gesamteffekt und einem Spalten-(*Microarray*-)Effekt. Betrachtet man nun ein *Probeset* mit dazugehörigen *Probepairs* über n Chips mit $(i, \cdots, n)$ zeigt sich folgende Matrix:

$$Y = \begin{pmatrix} y_{1j1} & y_{2j1} & \cdots & y_{nj1} \\ y_{1j2} & y_{2j2} & \cdots & y_{nj2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1jl_j} & y_{2jl_j} & \cdots & y_{njl_j} \end{pmatrix} \tag{2.37}$$

Im ersten Schritt wird für jedes *Probe-Pair* (Zeile) der Median über alle n Chips bestimmt ($med(y_{.j1})$. Dieser wird dann zeilenweise von den *Probe-Pairs* subtrahiert:

$$\tilde{Y} = \begin{pmatrix} y_{1j1} - med(y_{.j1}) & y_{2j1} - med(y_{.j1}) & \cdots & y_{nj1} - med(y_{.j1}) \\ y_{1j2} - med(y_{.j2}) & y_{2j2} - med(y_{.j2}) & \cdots & y_{nj2} - med(y_{.j2}) \\ \vdots & \vdots & \ddots & \vdots \\ y_{1jl_j} - med(y_{.jl_j}) & y_{2jl_j} - med(y_{.jl_j}) & \cdots & y_{njl_j} - med(y_{.jl_j}) \end{pmatrix} \tag{2.38}$$

Im zweiten Schritt wird der Median der einzelnen *Microarrays* (Spalten) über alle *Probe-Pairs* bestimmt ($med(\tilde{y}_{2j.}, i = 1, \cdots, n)$ und von allen Spalten subtrahiert:

$$\tilde{\tilde{Y}} = \begin{pmatrix} \tilde{y}_{1j1} - med(\tilde{y}_{1j.}) & \tilde{y}_{1j2} - med(\tilde{y}_{2j.}) & \cdots & \tilde{y}_{nj1} - med(\tilde{y}_{nj.}) \\ \tilde{y}_{1j2} - med(\tilde{y}_{1j.}) & \tilde{y}_{2j2} - med(\tilde{y}_{2j.}) & \cdots & \tilde{y}_{nj2} - med(\tilde{y}_{nj.}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}_{1jl_j} - med(\tilde{y}_{1j.}) & \tilde{y}_{2jl_j} - med(\tilde{y}_{2j.}) & \cdots & \tilde{y}_{njl_j} - med(\tilde{y}_{nj.}) \end{pmatrix} \tag{2.39}$$

Per Definition sind der Gesamteffekt des *Probesets* $j\mu_j$ der Median der Zeilenmediane der Matrix $\tilde{\tilde{Y}}$ und die Spalteneffekte $\lambda_i, i = 1, \cdots, n$ die Spaltenmediane der Matrix $\tilde{\tilde{Y}}$. Damit ist die Expression eines *Probesets* $j = 1, \cdots, n$ im i-ten *Microarray*, $i = 1, \cdots, n$ folgendermaßen definiert:

$$x_{ij} = \mu_j + \lambda_{ji} \tag{2.40}$$

2.5.6 Das RMA-Modell von Irizarry et al.

Irizarry et al. [Irizarry et al., 2003a] verwenden eine Hintergrundkorrektur, die von der Normalisierung getrennt ist. Das Hintergrundsignal eines Chips i setzt sich zusammen aus dem Hintergrundsignal bg_{ijk}, das unspezifische Bindungen und optische Schwankungen repräsentiert, und dem PM-Wert s_{ijk}. Die Formel hierzu lautet: $PM_{ijk} = bg_{ijk} + s_{ijk}$. Irizarry et al. setzen dabei voraus, das bg_{ijk} Werte normalverteilt sind und die s_{ijk} Werte einer Exponentialverteilung folgen, sodass die PM_{ijk} aus einer gemischten Verteilung kommen. Die adjustierten PM_{ijk}^{korr} können als Erwartungswert der s_{ijk} mit den gegebenen PM_{ijk} bestimmt werden.
Die Quantil-Normalisierung schließt sich der Hintergrundkorrektur an. Dabei nimmt man bei dieser Methode an, dass eine grundlegend bekannte Verteilung der Intensitäten über alle Chips vorliegt. Das Ziel der Quantil-Normalisierung ist, die Verteilungen der Intensitäten (PM und MM) für alle Chips identisch zu machen. Falls zwei Datensätze dieselbe Verteilung haben, befinden sich die Quantile auf einer Diagonalen. Durch die Transformierung der Quantile zweier ungleicher Datensätze mit der gleichen Verteilung erhält man den gleichen Wert. Dieses kann man durchführen, indem man auf die Einheitsdiagonale $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ projiziert. Auf n-Dimensionen ausgeweitet haben n Chips die gleiche Verteilung, sofern der n-dimensionale Q-Q-Plot die Einheitsdiagonale durch den Punkt $\left(\frac{1}{\sqrt{n}}, ..., \frac{1}{\sqrt{n}}\right)$ aufweist. Wenn sich die Punkte des n-dimensionalen Q-Q-Plots auf die Einheitsdiagonale projizieren lassen, hat man das Ziel, n-Datensätze auf dieselbe Verteilung zu bringen, erreicht.
Ist der Vektor der k-ten Quantile aller n Chips mit $q_k = (q_{k1}, \cdots, q_{kn})$ gegeben und man projeziert q_k auf die Einheitsdiagonale $d = \left(\frac{1}{\sqrt{n}}, \cdots, \frac{1}{\sqrt{n}}\right)$ erhält man:

$$\begin{aligned} proj_d q_k &= \frac{1}{\sqrt{n}} \sum_{j=1}^{n} q_{kj} d \\ &= \left(\frac{1}{n} \sum_{j=1}^{n} q_{kj}, \cdots, \frac{1}{n} \sum_{j=1}^{n} q_{kj} \right) \end{aligned} \tag{2.41}$$

Ersetzt man die wahren Werte durch die Mittelwerte der dazugehörigen Quantile, wird aus Formel 2.41 deutlich, dass die Verteilungen identisch sind. Mit diesem Ansatz werden die Verteilungen der PM-Werte und der MM-Werte identisch gemacht. Die Grundlage der Normalisierung bilden also die Matrizen der PM-Werte und der MM-Werte. Dabei wird die Datenmatrix der PM-Werte und der MM-Werte als N bezeichnet, wobei jede Spalte einen *Microarray* repräsentiert. Folgende Schritte werden dazu durchgeführt:

1. Jede Spalte von N ist nach der Größe der Genexpression zu sortieren und man erhält N_{sort}.

2. Der Mittelwert jeder Zeile wird berechnet und man ersetzt die Werte in den Zeilen durch die jeweiligen Mittelwerte. In jeder Zeile steht dann n mal derselbe Wert und man erhält N'_{sort}.

3. N'_{sort} wird in die ursprüngliche Reihenfolge von N zurücksortiert und man erhält so Z_{norm}.

Die nachfolgenden Schritte sind identisch mit zu der VSN-Methode (Kapitel 2.5.4). Für die *Probe*-spezifische Hintergrundkorrektur werden nur die PM-Werte weiter berücksichtigt [Bolstad et al., 2003], als Aggregationsmethode wird das Verfahren *Medianpolish* von Tukey [Tukey, 1977] angewendet.

2.5.7 Das sRMA-Modell von Cope et al.

Die bis heute zur Verfügung stehenden Protokolle für die Prozessierung der *Microarrays* im Labor benötigt viel Ausgangsmaterial und damit eine große Menge an mRNA. Auch wenn sich in den letzten Jahren die Grenze der Ausgangsmenge kontinuierlich reduziert hat, benötigte man bei unseren speziellen Fragestellungen immer noch das 10-fache der verfügbaren Ausgangsmenge.
Die Expressionsdaten, die durch spezielle Anpassungen, z. B. zusätzliche Vermehrungsschritte der mRNA (Amplifikation), erhoben werden, sind reproduzierbar. Vergleicht man diese aber mit Daten, die mit mehr Ausgangsmenge (Faktor 1000) gewonnen wurden, so korrelieren diese Daten nicht mehr. Cope et al. haben deshalb einen Algorithmus [Cope et al., 2006] entwickelt, der die besonderen Eigenschaften

der Daten aus geringer Ausgangsmenge berücksichtigt.
Eine Amplifikationsmethode ist z. B. eine Zwei-Runden-Invitroamplifikation, die auf T7 (RNA-Polymerase) basiert. Generell beginnt die Transkriptionsreaktion am 3'-Ende. Problematisch ist, dass für die zweite Runde die cDNA (komplementäre DNA) verwendet wird, die in der ersten Runde entstanden ist (RNA → cDNA). Die schon im ersten Schritt durch die Transkriptionsreaktion verkürzten Transkripte werden im zweiten Schritt nochmal verkürzt.
Das kann dazu zu führen, dass *Probes* auf dem Affymetrix *Microarray*, die weiter Richtung 5'-Ende liegen, nicht mehr abgedeckt werden. Das Design der *Probes* von Affymetrix ist so gewählt, das diese nahe dem 3'-Ende und damit nahe dem PolyA-Schwanz liegen (Abb.2.6). Allerdings ist die Grundlage für das *Probe*-Design das Standardprotokoll mit nur einem Invitroamplifikationsschritt (IVT). Hier sind die generierten Transkripte im Mittel etwa 600 Basen lang. Ein weiterer Amplifikationsschritt verkürzt die generierten Transkripte auf nur noch etwa 300 und 400 Basen und ist damit im Design der *Probes* nicht berücksichtigt.

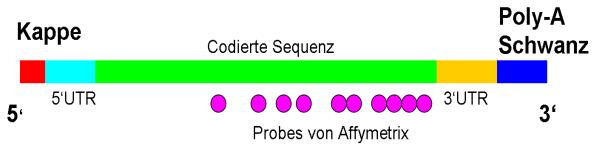


Abbildung 2.6: Schematischer Aufbau einer mRNA

Idee von Cope et al. ist, die Position der Probes im Transkript in ihrem Algorithmus einzubinden und zu vergleichen, ob es Verteilungsunterschiede gibt zwischen Daten, die nur mit einer Runde IVT oder mit doppelter IVT generiert wurden.
Die angepasste Methode RMA, die das doppelter IVT Expressionsdaten auswertet, wird nun sRMA (*small-sample-Version-RMA*) genannt. Hintergrundkorrektur und Normalisierung sind identisch mit dem Standard-RMA-Algorithmus. Die Hintergrundkorrigierten und normalisierten *Probe*-Intensitäten werden dann wie folgt modelliert:

$$\log_2(Y_{ijk}) = \theta_{ik} + \phi_{jk} + \epsilon_{ijk}, \tag{2.42}$$

für $i = 1, \cdots, I$; $j = 1, \cdots, J$; $k = 1, \cdots, K$, wobei, k einen *Microarray*, i ein *Probe-set*, j eine *Probe*, θ eine zur Menge an mRNA proportionale Größe (auf log-Skala),

ϕ einen Probe-spezifischen Effekt und ϵ den Meßfehler (mit einer *Probe*-spezifischen Varianz (σ_{ij}^2) repräsentiert.
Der Standard-RMA-Algorithmus benutzt *Medianpolish* als Aggregationsmethode, ein ad-hoc robustes Verfahren, um θ zu schätzen. Das obige Modell wird durch formale, robuste statistische Verfahren angepasst. Für jede *Probe* werden benutzerdefinierte Gewichte verwendet. Das Verfahren sRMA macht sich dies zunutze, indem der Anteil jeder *Probe* entsprechend seiner relativen Position im Transkript gewichtet wird. Verwendet wird das Inverse der Position spezifischen Variationskoeffizienten ($\frac{\sigma}{\mu}$). Definiert wird für jede *Probe* j und jedes *Probeset* i die relative Position der Probe innerhalb ihres Transkripts α_{ij} ($0 \leq \alpha_{ij} \leq 1$).
Der positionsabhängige Effekt wird durch Regression von Log-Skala Intensitäten $log_2(y_{ijk})$ auf α_{ij} berechnet. Um die positionsabhängigen Varianzen zu berechnen, werden die *Probe*-spezifischen Standardabweichungen (σ_{ih}) durch Verwendung von Replikaten geschätzt und mittels Regression auf die α_{ij} abgebildet.
Die α abhängigen Standardabweichungen und Effekte werden verwendet, um den spezifischen Variationskoeffizienten ($\frac{\sigma}{\mu}$) zu berechnen. Sie dienen damit zur Definition der Gewichte. Für die Evaluierung dieser Methode wurden keine simulierten Expressionsdaten verwendet, sondern nur reale Expressionsdaten aus einem Experiment.

2.5.8 Das sVSN-Modell

Der hier neu vorgestellte sVSN-Algorithmus (small-sample-VSN) ist eine Kombination aus dem sRMA-Algorithmus und den VSN-Algorithmus. Bei sVSN wird die Methode VSN für die Normalisierung anstatt der Quantilnormalisierung von RMA verwendet. Es gibt vorher keine Hintergrundkorrektur, da dieser Schritt in den VSN-Algorithmus integriert ist. Nachdem die PM-Werte mit VSN normalisiert wurden, werden die PM_{ijk}^{norm} wie in Formel 2.42 modelliert.
In Tabelle 2.1 sind alle Präprozessierungsalgorithmen und deren einzelne Schritte der Präprozessierung, die in dieser Arbeit verwendet wurden, in einer Übersicht dargestellt. Das kommentierte R-Script zu sVSN ist im Anhang (6.1) kommentiert.

	VSN	sVSN	RMA	sRMA	MAS4/MAS5
Hintergrundkorrektur	-	-	+	+	+
Normalisierung	**VSN**	**VSN**	**RMA**	**RMA**	**MAS5/MAS4**
Probe-spezifische Hintergrundkorrektur	**nur PM**	**nur PM**	**nur PM**	**nur PM**	**PM-MM (MAS4); PM-IMM (MAS5)**
Probe-spezifische Gewichtung in Abhängigkeit der Transkriptposition	-	+	-	+	-
Aggregationsmethode	**Medianpolish**	**Medianpolish**	**Medianpolish**	**Medianpolish**	**MAS5/MAS4**

Tabelle 2.1: Schematische Darstellung der einzelnen Schritte der verschiedenen Präprozessierungsalgorithmen bei Affymetrix-*Microarrays*

2.6 Simulation von Daten

Um eine neue Methode zu evaluieren und Genexpressionsdaten zu analysieren, muss man wissen, wie die einzelnen *Probesets* exprimiert sind. Ist die differentielle Expression von Genen bekannt, ist zu prüfen, ob das Finden dieser Gene durch die Analyseprozedur beeinflusst wird. Simulierte Daten haben im Gegensatz zu gemessenen Daten den Vorteil, dass alle Parameter zur Berechnung des Signals bekannt sind und nicht aus realen Daten geschätzt werden müssen.
Es werden Daten simuliert, die den Verteilungseigenschaften der realen Daten genügen. Darüber hinaus soll die Abhängigkeit der Position im Transkript der einzelnen *Probes* in die Simulation integriert und deren Einfluss auf die Normalisierung und die differentielle Expression der Gene untersucht werden.

Ziel dieser Simulation ist, Expressionsdaten zu simulieren, die die Verteilungseigenschaften der zwei Runden IVT-Daten repräsentiert. Die simulierten Daten werden mit den verschiedenen Präprozessierungsmethoden anlaysiert und verglichen. Die Ergebnisse werden auf ihre Sensitivität und Spezifität untersucht.

2.6.1 Kontinuierliche Verteilungen

Bevor Daten simuliert werden können, muss entschieden werden, welche Verteilungsfunktion dabei die Grundlage sein soll. Abhängig von der Verteilung der Daten kommen verschiedene kontinuierliche Verteilungen in Frage. Folgende kontinuierliche Verteilungen wurden betrachtet:

1. Weibull-Verteilung
2. Log-Normalverteilung
3. Gamma-Verteilung
4. Empirische Verteilung

Die Weibullverteilung

Die Weibullverteilung, benannt nach dem schwedischen Ingenieur und Mathematiker Waloddi Weibull, ist eine typische Verteilung bei der Untersuchung von Lebensdauer

und Zuverlässigkeiten. Die Weibullverteilung mit Formparameter a, Skalenparameter b und für $x \geq 0$ ist mit der Dichte gegeben:

$$f(x) = \frac{a}{b}\left(\frac{x}{b}\right)^{(a-1)} e^{-(\frac{x}{b})^a} \tag{2.43}$$

Die kumulative Verteilungsfunktion bei $x \geq 0$ ist:

$$F(x) = 1 - e^{-(\frac{x}{b})^a} \tag{2.44}$$

Der Mittelwert ist $E(X) = b\Gamma(1 + \frac{1}{a})$ und die Varianz durch $Var(X) = b^2 \cdot (\Gamma(1 + \frac{2}{a}) - (\Gamma(1 + \frac{1}{a}))^2)$ ist gegeben.
Das Erscheinungsbild der Weibullverteilung wird wesentlich durch den Formparameter a bestimmt. Für $0 < a \leq 1$ fällt die Dichtefunktion monoton, aus für $a > 1$ nimmt sie eine Glockenform an (siehe Abb.2.7).

Die Log-Normalverteilung

Die Log-Normalverteilung kommt in natürlichen Prozessen häufiger vor, wenn eine kontinuierliche Größe auf einer Seite begrenzt ist. Dabei ist eine Zufallsvariable X log-normalverteilt, wenn $\ln(X)$ normalverteilt ist. Die Log-Normalverteilung ist mit folgender Dichte gegeben:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{(\log x - \mu)^2}{2\sigma^2}}, \tag{2.45}$$

wobei μ den Mittelwert und σ die Standardabweichung darstellen. Der Mittelwert ist mit $E(X) = e^{\mu + \frac{\sigma^2}{2}}$, der Median durch $med(X) = e^{\mu}$ und die Varianz durch $Var(X) = e^{2 \cdot \mu + \sigma^2} \cdot e^{\sigma^2} - 1$ gegeben. In Abbildung 2.8 ist die Dichtefunktion mit unterschiedlichen σ und $\mu = 0$ dargestellt.

Die Gamma-Verteilung

Die Gamma-Verteilung beruht auf der Gamma-Funktion und ist eine direkte Verallgemeinerung der Exponentialverteilung. Sie wird unter anderem in der Warteschlan-

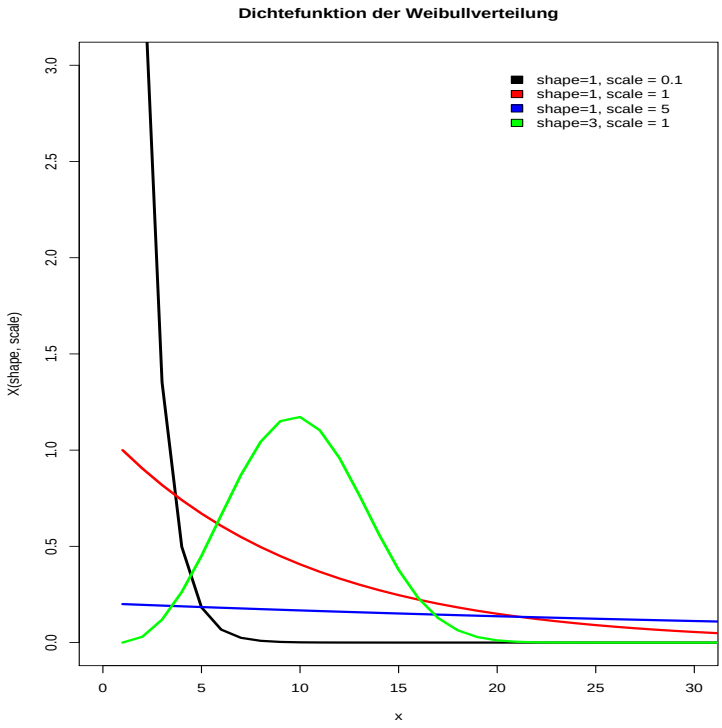


Abbildung 2.7: Dichtefunktion der Weibullverteilung bei verschiedenen Form- und Skalenparametern.

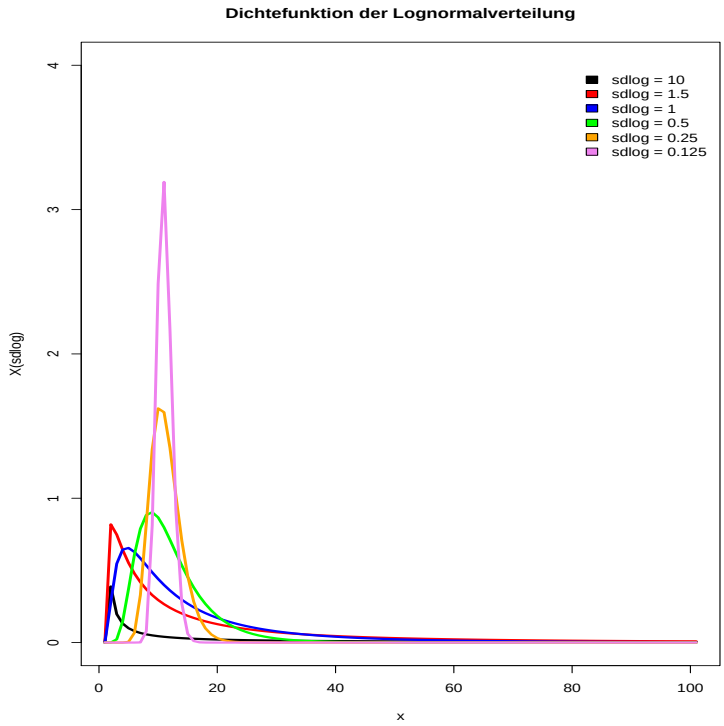


Abbildung 2.8: Dichtefunktion der Lognormalverteilung bei verschiedenen σ und $\mu = 0$.

gentheorie und in der Versicherungsmathematik verwendet. Die Gammaverteilung hat mit dem Formparameter a, dem Skalenparameter s und für $a > 0$ und $s > 0$ die Dichte:

$$f(x) = \begin{cases} \frac{s^a}{\Gamma(a)} x^{a-1} e^{-bx} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.46}$$

Der Mittelwert und die Varianz sind durch $E(X) = a \cdot s$ und $Var(X) = a \cdot s^2$ gegeben. In Abbildung 2.9 ist die Dichtefunktion der Gammaverteilung mit verschiedenen Form- und Skalenparametern dargestellt.

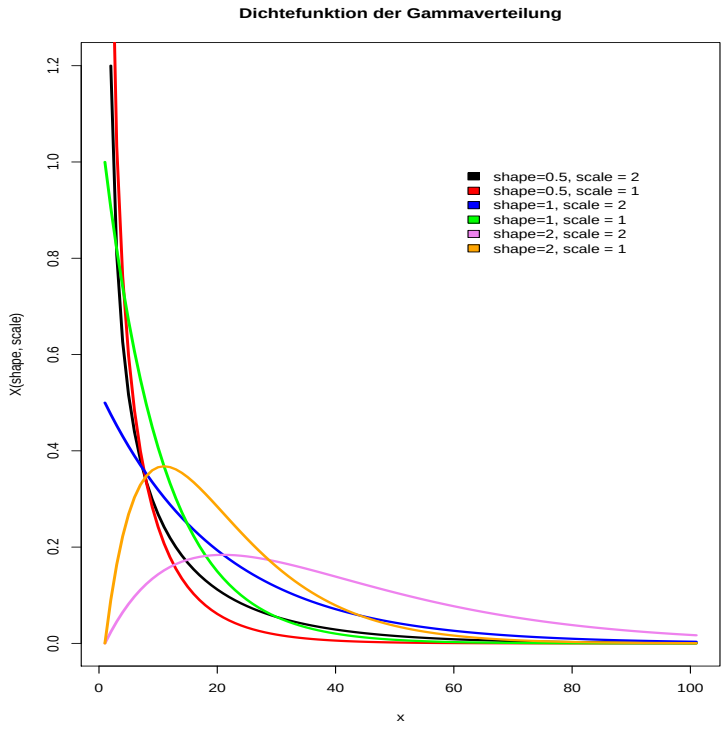


Abbildung 2.9: Dichtefunktion der Gammaverteilung bei verschiedenen Form- und Skalenparametern.

Die empirische Verteilung

Bei der empirischen Verteilung teilt man den Wertebereich der Expressionsdaten in viele gleichgroße Bereiche auf und betrachtetet die Wahrscheinlichkeiten für die

Expressionswerte des jeweiligen Bereiches. Innerhalb eines Bereichs wird die Verteilungsfunktion stückweise linear gehalten. Für die Berechnung der empirischen Verteilungsfunktion werden die gesamte Stichprobe, als verteilungsbeschreibender Parameter, die Anzahl zu generierender Zahlen und die Bereiche benötigt.
Die vier Verteilungsvorschlägen werden durch Q-Q-Plots und Histogramme mit der Verteilung der realen Expressionswerte verglichen, um zu einer Entscheidung bei der Wahl der Verteilung zu kommen.

2.6.2 Simulieren der Daten

Bei der Simulation werden zufällig numerische Zahlen aus der verwendeten empirischen Verteilungsfunktion generiert und den einzelnen *Probes* zugewiesen. Zusätzlich werden die Positionsabhängigkeit der *Probes* und die differentielle Expression in die Daten integriert.

Positionsabhängigkeit

Von der Lage im Transkript ist die differentielle Expression von bestimmten *Probesets* abhängig.
Die Lage im Transkript wird mit $1, ..., n$ definiert, wobei n die Anzahl der *Probes* in einem *Probeset* repräsentiert. Es werden dann zufällig n-*Probes* pro *Probeset* gezogen und der Größe nach sortiert. Der größte Wert bekommt die erste Position und repräsentiert die *Probe*, die am nächsten am 3'-Ende des Transkriptes liegt. So ist gewährleistet, dass eine Abhängigkeit von der Position im Transkript mit der Expression der *Probe* besteht.

Simulation der differentiellen Expression

Die differentielle Expression muss so definiert werden, dass die Verteilungseigenschaften der simulierten Daten nicht verändert werden. Im ersten Schritt teilt man die simulierten Expressionswerte in 10 Quantile auf. Die 10 Quantile repräsentieren die unterschiedliche Stärke der Expression. Im zweiten Schritt wird festgelegt, wie viele *Probesets* differentiell sein sollen. Die ausgewählten *Probesets* bekommen nach dem Zufallsprinzip einen Bereich aus den 10 Quantilen zugewiesen. Zufällig wird ein Expressionwert gezogen und überprüft, ob dieser in dem zuvor festgelegten Quantil

liegt. Ist dies der Fall, so wird dieser Expressionswert einem differentiellen *Probeset* zugewiesen. Ansonsten wird so lange weiter gezogen, bis die Quantil-Bedingung erfüllt ist. Die Anzahl der Spalten einer Gruppe wird dabei berücksichtigt. So erhält man *Probesets*, die die verschiedenen Expressionsgrößen abdecken. Die andere Gruppe bekommt die Expressionswerte zufällig zugewiesen. Auch hier wird die Position im Transkript berücksichtigt und in deren Abhängigkeit die Expressionswerte sortiert.
Da einige Präprozessierungsmethoden, die hier miteinander verglichen werden, auch die MM-Werte verwenden, werden diese wie die PM-Werte simuliert. Allerdings ist davon auszugehen, dass die MM-Werte unabhängig von den PM-Werten sind und diese zufällig auf die einzelnen Positionen eines *Probesets* verteilt.
Durch mehrmaliges Wiederholen der Simulation soll die Variabilität der Simulation geprüft werden. Alle dafür entwickelten R-Skripte sind im Anhang kommentiert (6.2).

2.6.3 Lineare gemischte Modelle

Bei der Simulation auf der Basis kontinuierlicher Verteilungen wurde die Sortierung der *Probe*-Expression in Abhängigkeit der Position im Transkript sehr extrem vorgenommen. Bei der durchgeführten Sortierung ist immer der größte Expressionswert an erster Position und der kleinste Wert an letzter Position im *Probeset*. Allerdings entspricht das nicht den realen Daten. Hier liegt keine Sortierung der einzelnen *Probes* innerhalb eines Probesets vor. Mit einem linearen Modell werden die Positionsabhängigkeiten der *Probe*-Expression in die Datensimulation mit eingebracht.
Das lineare Modell gehört zu den mathematischen Methoden, die versuchen, natürliche Beobachtungen in nachvollziehbaren und wiederkehrenden Parametern zusammenzufassen. Das lineare Modell hat als Grundvoraussetzung, dass ein linearer, geradliniger Zusammenhang zwischen mindestens einer unabhängigen und einer abhängigen Variablen vorliegt. Zusätzlich wird angenommen, dass die Daten nicht direkt beobachtet werden und fehlerhaft sind. Allgemeine lineare Modelle lassen sich durch folgende Matrixgleichung darstellen:

$$\overrightarrow{y} = X\overrightarrow{\beta} + \overrightarrow{\epsilon} \tag{2.47}$$

Dabei stellt $\overrightarrow{y}$ den Vektor der abhängigen Variablen und die Matrix X den Vektor

der unabhängigen Variablen dar.

$$\overrightarrow{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{2.48}$$

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nk} \end{pmatrix} \tag{2.49}$$

$\overrightarrow{\beta}$ ist der Vektor der Regressionskoeffizienten der mit X beschriebenen Variablen und $\overrightarrow{\epsilon}$ der Vektor der Fehlergröße.

$$\overrightarrow{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} \tag{2.50}$$

$$\overrightarrow{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{2.51}$$

Vorausgesetzt wird, dass der wirkliche Datensatz (abgesehen vom Fehlerterm) vom linearen Modell beschrieben wird. Die Art des Fehlers ist dabei nicht weiter spezifiziert. Das können Meßfehler oder andere zusätzliche Faktoren sein, jedoch ist der Erwartungswert von $\overrightarrow{\epsilon}$ in allen Komponenten 0. Das heißt man kann die Formel 2.47 wie folgt darstellen:

$$E\overrightarrow{y} = X\overrightarrow{\beta} \tag{2.52}$$

Dabei sind die beobachteten Abweichungen als zufällig und unabhängig anzusehen. Eine typische Annahme ist, dass die Komponenten des Vektors $\overrightarrow{\epsilon}$ unkorreliert sind und dieselbe Varianz σ^2 besitzen. Damit kann das Verfahren der kleinsten Quadrate als Schätzer für $\overrightarrow{\beta}$ und σ^2 verwendet werden. In der Realität kann es aber vorkommen, dass einige der unabhängigen Variablen und damit auch die Fehler teilweise korreliert sind. Das bringt erhebliche methodische Probleme beim üblichen Schätzverfahren mit sich, dieses ist daher nicht anwendbar. Genau das liegt bei den untersuchten Expressionsdaten vor. Einige Variablen (Position der *Probe* von der Expressionsstärke) sind nicht unabhängig und zum Teil stark korreliert. Daher ist

ein einfaches lineares Modells nicht einsetzbar.

Das einfache lineare Modell betrachtet nur sogenannte feste Effekte in Form der Variablen β, die für jede Beobachtung gleich sind. Lineare gemischte Modelle enthalten darüber hinaus noch zufällige Effekte, sogenannte Random-Effekte, γ_i, die Realisierungen von Zufallsvariablen sind und sich somit für jede Beobachtung unterscheiden können. Zufällige Effekte werden eingesetzt, sofern man davon ausgeht, dass nicht alle relevanten Kovariablen auf der Zielvariablen beobachtet werden können. Dabei kann es sich um viele nicht beobachtete Einflussgrößen mit jeweils nur geringem Einfluss handeln. Das lineare gemischte Modell wird häufig bei Longitudinal- (wiederholten Messungen am selben Versuchstier) oder Clusterdaten eingesetzt [Pinheiro and Bates, 2000]. Das lineare gemischte Modell (LMM) wird allgemein wie folgt dargestellt:

$$Y = X\beta + Z\gamma + \epsilon \tag{2.53}$$

$$\begin{pmatrix} \gamma \\ \epsilon \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} G & 0 \\ 0 & R \end{pmatrix} \right) \tag{2.54}$$

X und Z : feste bekannte Design-Matrix
β : Vektor der Regressionskoeffizienten
γ : Vektor der zufälligen Effekte
R : Kovarianzmatrix der Fehlerterme, also $Cov(\epsilon_i, \epsilon_j)$
G : Kovarianzmatrix der zufälligen Effekte, also $Cov(\gamma_i, \gamma_j)$

Die Designmatrix umfasst alle Werte der unabhängigen Variablen und wird als Basis für die Berechnung von Modellen benutzt, da in der Design-Matrix alle unabhängigen Variablen gespeichert sind. Zu beachten ist, dass es bei geringer Variabilität bei einer der Variabeln, die für die Random-Effekte angesetzt wurden, zu Konvergenzproblemen bei der numerischen Maximierung der Likelihood kommen kann. Dies kann häufig durch Reskalierung behoben werden.
Das R-Paket lme4 [Baayen et al., 2008] beinhaltet R-Funktionen für das Anpassen und Analysieren von linearen gemischten Modellen.

Auf Basis des angepassten Modells werden die PM-Expressionsdaten und MM-Expressionsdaten simuliert.
Die simulierten Expressionsdaten enthalten noch keine *Probesets*, die differentiell exprimiert sind. Dies wird dadurch erreicht, dass n *Probesets* in $1 < i <$ Anzahl Spalten mit einer zufälligen Zahl von 2 - 10 multipliziert wurden. Der Anteil der differentiellen *Probesets* sollte nicht groß sein (≤ 1000), da ansonsten die zuvor simulierte Verteilung verfälscht würde.
Alle dafür entwickelten R-Skripte sind im Anhang kommentiert (6.2).

2.7 *Cluster-Analysen*

Die Clusteranalyse ist ein Verfahren aus der multivariaten Statistik, um auf Basis von Ähnlichkeiten oder Distanzen Objekte zu Clustern mit gemeinsamen Eigenschaften zu gruppieren. Clustermethoden haben eine lange Tradition und werden schon seit den 60er Jahren zur Konstruktion von Phylogeniebäumen in der Biologie eingesetzt.
In zahlreichen wissenschaftlichen Disziplinen wird diese Analyse angewendet. Basis ist die sogenannte Distanzmatrix, die den Abstand zwischen den Elementen definiert. Der Cluster-Algorithmus übersetzt diese Distanzmatrix in einem Baum. Mit Hilfe von *Bootstrap*-Methoden [Efron and Tibshirani, 1993] kann die Robustheit des Baumes untersucht und damit seine Aussagefähigkeit überprüft werden.
Analysiert man *Mircoarray*-Daten taucht unweigerlich irgendwann die Frage auf, ob man bei den Daten Expressionsmuster findet. Cluster-Algorithmen können für diese Frage verwendet werden. Allerdings werden Cluster-Algorithmen nicht zum Hypothesentest verwendet und liefern daher keine Information über die Signifikanz der Ergebnisse. Vielmehr sollen durch das Clustern neue biologische Muster gefunden werden, die zu neuen Hypothesen führen.
Da bei einem *Mircoarray*-Experiment Tausende von Genen gleichzeitig gemessen werden, stellt sich die Frage, für welche Gene das Clusterverfahren angewendet werden soll. Dieser Schritt muss genau definiert werden, weil er einen massiven Einfluss auf das Ergebnis des Clusters hat. Welche Gen-Filter man letztendlich anwendet, hängt davon ab, welcher Cluster-Algorithmus angewendet wird. Deshalb ist es unverzichtbar, die Stabilität des generierten Clusters z. B. durch ein *Bootstrap*-Verfahren zu überprüfen. Dies wird nach der Beschreibung der verschiedenen Clusterverfahren

erläutert.
Vorteil der Cluster-Methoden ist, dass sie einen visuellen Überblick über die hochdimensionalen Daten ermöglichen und zur Klassenerkennung genutzt werden können. Für die Analyse von Genexpressionsdaten werden am häufigsten hierarchische Clustermethoden verwendet. Dieses Verfahren produziert einen Baum, das sogenannte Dendrogramm, das aus einer Anzahl ineinandergeschachtelter Klassen besteht. Bei Genexpressionsdaten wird dabei nach ähnlichen Expressionsmustern über die Spalten (Patienten) oder Zeilen (Gene) gesucht. Dabei berücksichtigen die zu bildenden Cluster nur die gemessene und damit die intrinsischen Probeneigenschaften. Daraus entwickelte sich der Name des *„unsupervised Clustering“* oder unüberwachtes Clustern.
Es gibt viele Clustermethoden, die man auf Genexpressionsdaten anwenden kann. Zu nennen sind hier das „Hierarchische Clustern“ [Eisen et al., 1998; Alon et al., 1999], „Self-organizing Maps“ (SOM) [Tamayo et al., 1999], „Relevance Networks“ [Butte et al., 2001] oder das „k-Means Clustering“ [Sherlock, 2000]. Alle erwähnten Methoden scheinen geeignet zu sein, in einem Datensatz Strukturen aufzuzeigen [Golub, 2001],
M. Eisen hat als erster das hierarchische Clustern auf Genexpressionsdaten angewendet, der Grundlage die Pearson-Korrelation ist. Diese Methode erkennt Muster in Gexpressionsdaten und ist im Programm „Cluster and TreeView“ implementiert. Diese Software ist frei erhältlich und wurde in einigen Veröffentlichungen verwendet [Iyer et al., 1999; Notterman et al., 2001; Sørlie et al., 2001].
Im ersten Schritt benötigt der Cluster-Algorhitmus ein Distanzmaß, um festzustellen, ob zwei Objekte zum selben Cluster gehören. Bei *Microarray*-Daten wird hierzu die Distanz aus den Expressionsvektoren berechnet. Je nachdem, ob *Probesets* oder Patienten geclustert werden sollen, werden Gen- oder Patientenvektoren zur Distanzbestimmung verwendet. Metrische paarweise Distanzen sind z. B. die euklidische Metrik:

$$d_{euc}(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{2.55}$$

oder die Manhattan-Metrik:

$$d_{man}(\vec{x},\vec{y}) = \sum_{i=1}^{m} |x_i - y_i| \tag{2.56}$$

Es gibt eine Reihe korrelationsbasierter Distanz-Maßen, die bei *Microarray*-Daten angewendet werden.
Pearson Korrelationsdistanz:

$$d_{pear}(\vec{x},\vec{y}) = 1 - r(\vec{x},\vec{y}) = 1 - \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x})^2 \sum_{i=1}^{m}(y_i - \overline{y})^2}} \tag{2.57}$$

Spearman Korrelationsdistanz:

$$d_{spear}(\vec{x},\vec{y}) = 1 - \frac{\sum_{i=1}^{m}(x'_i - \overline{x}')(y'_i - \overline{y}')}{\sqrt{\sum_{i=1}^{m}(x'_i - \overline{x}')^2 \sum_{i=1}^{m}(y'_i - \overline{y}')^2}} \tag{2.58}$$

Hierbei beschreibt $x'_i = rank(x_i)$ den *Spearman's rank* d. h. sortiert man $x_1, ..., x_n$ der Größe nach, so erhält man $x'_1, ..., x'_n$.
Das Hierarchische Clustern ist ein agglomerativer Ansatz, der mit einem Cluster für jeden Expressionsvektor beginnt. Zunächst werden paarweise die Distanzen für alle Cluster berechnet und die zwei ähnlichsten Cluster werden zu einem neuen Cluster vereinigt. Nach diesem ersten Schritt werden die Abstände in der Distanzmatrix neu kalkuliert. Das Verfahren wird solange fortgeführt bis, ein komplettes Dendrogramm entstanden ist. Unabhängig von der Distanzfunktion können die Distanzen zwischen den Clustern mit unterschiedliche Methoden berechnet werden.

1. *Single-linkage* (Minimum, nächster Nachbar): Die Distanz zwischen zwei Clustern i und j wird aus der minimalen Distanz zwischen je einem Mitglied aus Cluster i und j berechnet. Hier können Cluster zusammengefasst werden, bei denen nur zwei Mitglieder ähnlich sind.

2. *Complete-linkage* (Maximum, entferntester Nachbar): Hier wird die Distanz zwischen zwei Clustern aus der größten Differenz zwischen den Mitgliedern der Cluster berechnet.

3. *UPGMA*: Hier werden die Durchschnittswerte des Clusters zur Distanzberechnung verwendet. Dabei gibt es verschiedene Methoden zur Bestimmung des Durchschnitts. Die am häufigsten verwendete Methode ist die ungewichtete gepaarte Gruppen-Methode *Average* (UPGMA). Die Durchschnittsdistanz wird aus der Distanz zwischen jedem Punkt im Cluster und allen Punkten aus dem anderen Cluster berechnet. Zu einem neuen Cluster werden die Cluster zusammengefasst, die die geringste Durchschnittsdistanz haben.

4. *Ward's* Methode: Hier wird die Summe der quadrierten Abweichungen vom Mittelwert des Clusters berechnet und die Cluster so zusammengefasst, dass der kleinste mögliche Anstieg in der Summe der quadrierten Fehler entsteht.

In der vorliegenden Arbeit wurde für alle Cluster-Analysen die Distanzmatrix mit der *Manhattan*-Metrik berechnet und die UPGMA-Methode für das hierarchische Clustern der Zeilen (*Probesets*) und Spalten (Patienten) verwendet.

Bootstrap-Methoden zur Überprüfung der Stabilität der Cluster-Analyse

Die *Bootstrap*-Methode von Suzuki et al. [Suzuki and Shimodaira, 2006], die im *pvclust* Paket von *Bioconductors* implementiert ist, wird verwendet, um die Stabilität der Cluster-Analyse zu prüfen. In der Praxis stehen meistens keine weiteren Daten zur Validierung der Ergebnisse zur Verfügung, denoch möchte man eine Aussage über die Robustheit des Baums machen können. Kernidee aller *Bootstrap*-Verfahren ist, aus den Originaldaten eine Anzahl modifizierter Datensätzen zu erzeugen, sogenannte *Bootstrap-Samples*, und mit denselben Methoden auszuwerten. Die Modifikation der Daten können ein Resampling oder ein Weglassen bestimmter Segmente sein. Das Ziel jeden *Bootstrap*-Verfahrens ist es zu prüfen, ob die Ergebnisse von stochastischen Effekten beeinflusst werden. Eine ausführliche Darstellung der verschiedenen *Bootstrap*-Verfahren und der Erzeugungsvorschriften findet man im Buch von Efron und Tibshirani [Efron and Tibshirani, 1993]. Im Paket *pvclust* wurden zwei Methoden zur Bewertung des p-Wertes verwendet: AU (Approximately unbiased) p-Wert und BP (Bootstrap Probability) p-Wert. Der BP-Wert wird über eine Standard-Bootstrap-Wiederholungsprobennahme (standard bootstrap resampling) errechnet. Für das *Multiscale-Bootstrap-Resampling* werden verschiedene Datengrößen der *Bootstrap*-Stichproben verwendet. Dabei ist N die Originalgröße des Datensatzes und N' die Größe der *Bootstrap*-Stichprobe. Für jeden Cluster erhält man einen BP-Wert

für jeden Wert von N'. Außerdem untersucht man die Änderung der $z = -\Phi^{-1}$ (BP)-Werte, wobei $\Phi^{-1}(\cdot)$ die inverse Funktion von $\Phi(\cdot)$ darstellt, der Verteilungsfunktion der Standard-Normalverteilung. Im nächsten Schritt wird eine theoretische Kurve mit $z(N') = v\sqrt{N'/N} + c\sqrt{N/N'}$ an die beobachteten Werte angepasst. Da nur ein Anteil der Datenmatrix N in der Bootstrap-Stichprobe N' betrachtet wird, müssen die Koeffizienten v und c skaliert und für jeden Cluster geschätzt werden. N'/N entspricht dem Anteil der Dimensionen der Datenmatrizen N. Dabei ist v der mit Vorzeichen versehene Abstand mit $N(0,1)$ verteilt. c ist die Krümmung der Datenmatrix [Shimodaira, 2002].
Der AU p-Wert wird berechnet mit $AU = \Phi(-v + c)$. Suzuki et al. bewiesen, dass der AU-Wert ein besserer Annäherungswert an einen erwartungstreuen p-Wert ist als der BP-Wert. Bei der Bewertung der Clusterbäume wird untersucht, wie häufig ein bestimmter Cluster im Baum zufällig auftritt. *Bootstrap*-Werte werden für jede Verzweigung im Baum ermittelt und man zählt dabei, wie oft in den Bäumen eine Verzweigung auftritt. Je größer die *Bootstrap*-Werte sind, desto robuster ist dabei die jeweilige Verzweigung.

2.8 Heatmap

Eine Heatmap (Hitzekarte) ist eine Falschfarben-Darstellung einer Matrix numerischer Werte. Heatmaps wurden mit der Spotfire DecisionSite 9.1, 1996-2007 Software (Spotfire) und mit dem Bioconductor-Paket *geneplotter* erstellt.

2.9 Definition der differentiellen Genexpression

Eine der häufigsten Hypothesen, die durch ein *Mircoarray*-Experiment untersucht werden soll, ist die Frage nach Unterschieden in der Genexpression zwischen verschiedenen biologischen Zuständen (z. B. gesundes Gewebe und Tumorgewebe). Dabei wird jeder Zustand durch mehrere Experimente repräsentiert. Die Nullhypothese, dass keine Unterschiede in der Expression zwischen den Patienten besteht, kann getestet werden und dabei ein Signifikanztest durchgeführt werden. Innerhalb des Stichprobenraums kann man eine kritische Größe D_α so bestimmen, dass im Falle der Richtigkeit von H_0 das Ergebnis einer Zufallsstichprobe vom Umfang n höchstens mit der vorgegebenen kleinen Wahrscheinlichkeit α in diese Region fällt. Fällt

das Ergebnis in diese Region, kann die Nullhypothese mit der Irrtumswahrscheinlichkeit α abgelehnt werden. Der p-Wert beschreibt den kleinsten Wert α, für den H_0 abgelehnt wird.
Die einfachste Methode, um festzustellen, ob eine differentielle Expression zwischen zwei definierten Gruppen besteht, ist, sich das Expressionsverhältnis (Fold Change = FC), anzuschauen. Alle Gene, die sich um einen bestimmten Wert von FC unterscheiden, werden dann als differentiell exprimiert betrachtet. Bei diesem Test handelt es sich nicht um einen statistischen Test. Er gibt keine Fehlerwahrscheinlichkeit für die Zuweisung eines Gens an. Allerdings kann der FC als ergänzender Filter für einem statistischen Test herangezogen werden.

Zwei-Stichproben-t-Test

Ein statistisches Verfahren für die Ermittlung von differentieller Expression ist der Zwei-Stichproben-t-Test. Der Zwei-Stichproben-t-Test, der in dieser Arbeit verwendet wurde, hat folgende Modellannahme: Sind $x_1, \cdots, x_n$ bzw. $y_1, \cdots, y_n$ Realisierungen von Zufallsvariablen $X_1, \cdots, X_n$ bzw. $Y_1, \cdots, Y_n$, dann sind alle X_i, Y_i unabhängig mit der Verteilung:

$$\begin{aligned} X_i &\sim \mathcal{N}(\mu, \sigma^2) \quad (i = 1, \cdots, n), \\ Y_i &\sim \mathcal{N}(\nu, \sigma^2) \quad (j = 1, \cdots, n) \end{aligned} \tag{2.59}$$

wobei μ, ν, σ^2 unbekannt sind.
Unter Normalverteilungsannahme liegt also ein *Shift*-Modell vor. Beim zweiseitigen Zwei-Stichproben-*t*-Test mit $H_0 : \mu = \nu$ gegen $H_1 : \mu \neq \nu$ ist die Prüfgröße durch Folgendes gegeben:

$$T = \frac{\sqrt{\frac{m \cdot n}{m+n}} \cdot (\overline{x} - \overline{y})}{\sqrt{\frac{1}{m+n-2} \cdot ((n-1) \cdot s_x^2 + (m-1) \cdot s_y^2)}} \tag{2.60}$$

Der Testentscheid ist, falls $|T| \geq t_{m+n-2,1-\alpha/2}$ erfüllt ist, H_0 abzulehnen und, falls $|T| < t_{m+n-2,1-\alpha/2}$ erfüllt ist, keinen Widerspruch zu H_0 zu erheben. Das heißt, bei Ablehnung von H_0 sind $\overline{x}$ und $\overline{y}$ signifikant verschieden auf dem α-Niveau.

Viele der Gene, die auf dem *Microarray* repräsentiert sind, sind in den meisten Experimenten nicht ausgeprägt oder zeigen nur eine geringe Variabilität unter den Experimenten. Für alle paarweisen Vergleiche wird ein globaler Filter angewendet, um die Dimension des *Microarrays* zu reduzieren. Zusätzlich werden ein Intensitätsfilter (Signalintensität eines *Probesets* soll über 100 in mindestens 50 % der Stichprobe sein, wenn die zu vergleichenden Gruppen gleich groß sind) und ein Varianzfilter (Quartilsabstand der $\log_2$-Intensitäten soll mindestens 0,5 betragen, wenn die zu vergleichenden Gruppen gleich groß sind) angewendet. Wenn die zu vergleichenden Gruppen nicht gleich groß waren, sollten die Signalintensitäten eines *Probesets* über 100 in mindestens einer Fraktion α der Stichprobe liegen, wobei α die kleinere Gruppengröße minus 1, dividiert durch die Gesamtanzahl der Stichprobe der zwei Gruppen ist. Der Quartilsabstand der $\log_2$-Intensitäten soll mindestens 0,1 betragen, wenn die zu vergleichenden Gruppen nicht gleich groß sind. Unter der Annahme, dass die Varianz in beiden Gruppen gleich groß ist, wurden nach der globalen Vorfilterung p-Werte mit einem Zwei-Stichproben t-Test berechnet, um diejenigen Gene zu identifizieren, die zwischen den beiden Gruppen differentiell ausgeprägt sind.

2.9.1 Definition des Fold-Changes

In der vorliegenden Arbeit wurde eine einfache FC-Berechnung verwendet, die einen Eindruck über die Stärke der differentiellen Expression geben soll. Der FC ist wie folgt für zwei verschiedene Gruppen definiert:

$$FC_i := \begin{cases} 2^{\mu_{i1}-\mu_{i2}}, & \text{if } \mu_{i1} - \mu_{i2} \geq 1 \\ \frac{-1}{2^{\mu_{i1}-\mu_{i2}}}, & \text{if } \mu_{i1} - \mu_{i2} < 1 \end{cases} \tag{2.61}$$

Dabei ist μ der jeweilige Stichproben-Mittelwert der Gruppe 1 oder 2 des Gens i.
Klein et al. [Klein et al., 2001] haben den *z-score* für die Signifikanzberechnung der Gene verwendet. Die in dieser Arbeit reanalysierten Datensätze wurde mit dieser Methode verglichen. Hierbei wird der Standard-FC noch durch die Summe der Varianz der beiden Gruppen dividiert. Der *z-score* (z_g) von Klein et. al ist wie folgt definiert:

$$z_g = \frac{\mu_p - \mu_C}{\sigma_p + \sigma_C} \tag{2.62}$$

Dabei ist μ der Stichproben-Mittelwert und σ die Stichproben-Varianz der Gruppe p und C.

2.9.2 Das multiple Testen-Problem

Testet man Gene mit einem parametrischen Test — in unserem Falle dem t-Test — auf signifikante Genexpression, so liegt die Wahrscheinlichkeit, dass mindestens eine Nullhypothese fälschlicherweise abgelehnt wird und damit ein Gen als differenziell ermittelt wird, ohne es zu sein, bei $1-(1-\alpha)^m$. Voraussetzung für diese Annahme ist die Unabhängigkeit der Hypothesen. Die Anzahl der durchgeführten Tests ist durch m definiert. Wenn man $\alpha = 0.05$ wählt und eine große Anzahl von Tests durchführt, ist die Wahrscheinlichkeit groß, falsch positive Gene zu bekommen. Die Unabhängigkeit der Hypothesen ist bei einem *Microarray*-Experiment nicht gegeben, da es viele Gene gibt, die koreguliert sind. Trotzdem ist die Wahrscheinlichkeit hoch, ein Gen als fälschlicherweise differentiell exprimiert einzustufen.
Unter der Annahme, dass man unabhängige Hypothesen testet, liegt der Erwartungswert der Anzahl der falsch Positiven bei $m \cdot \alpha$. Bei z.B. 10000 Probesets die simultan getestet werden, entspricht bei $\alpha = 0.05$ der Anzahl der falsch positiven bei 500 *Probesets*. Obwohl die Unabhängigkeitsannahme nicht gegeben ist, kann man davon ausgehen, dass der Erwartungswert groß ist.
Bei der Durchführung eines Hypothesentests unterscheidet man zwei Arten von Fehlern:

1. Fehler 1.Art oder „falsch positiv“, die entstehen, wenn ein Gen als differentiell exprimiert deklariert wird, obwohl das nicht der Fall ist.

2. Fehler 2.Art oder „falsch negativ“, bei dem ein differentiell exprimiertes Gen nicht erkannt wird.

Ein statistischer Test ist so konstruiert, dass er die Fehlerwahrscheinlichkeit des Fehlers 1.Art kontrollieren kann. Aufgrund der hochdimensionalen Daten der *Micorarrays* werden simultan viele tausende Hypothesentests gleichzeitig durchgeführt. Dabei kann sich der Fehler der 1.Art akkumulieren. Dieses Problem wird als „multiples Testen Problem“ bezeichnet.

Die Kontrolle der *„family-wise-error-rate“* (FWER) ist ein Ansatz zur Lösung des multiplen Testens Problems. Sie gibt die Wahrscheinlichkeit an, einen oder mehrere Fehler der 1.Art über mehrere statistische Tests zu akkumulieren. Von jedem statistischen Test wird eine Erhöhung der Genauigkeit verlangt. Die einfachste und älteste FWER-Methode ist die Bonferroni-Korrektur. Dabei wird das Signifikanzniveau α durch die Anzahl der durchgeführten Tests geteilt. Sind die Nullhypothese mit $H_1, \cdots, H_m$ und die entsprechenden p-Werte mit $p_1, \cdots, p_m$ gegeben, wird H_z abgelehnt, wenn $p_z \leq \frac{\alpha}{m}$ ist. Dieses Verfahren ist allerdings nur eine grobe Näherung und sehr konservativ.
Die Bonferroni-Holm *Step-Down*-Methode [Holm, 1979] bietet einen besseren Ansatz. Dabei werden die p-Werte der einzelnen *Probesets* aufsteigend sortiert. Dabei bezeichnen $p_{1'}, \cdots, p_{m'}$ die sortierten p-Werte. H_z wird abgelehnt, wenn $p_{z'} \leq \frac{\alpha}{m-(z+1)}$ erfüllt ist.
Ein Verfahren, das sich bei *Microarray*-Analysen durchgesetzt hat, ist die *False-Discovery-Rate* (FDR) . Hier wird der Anteil von Genen, die ursprünglich als differentiell erkannt wurden, als falsch positiv definiert. Das bekannteste und in der vorliegenden Arbeit verwendete Verfahren ist von Bejamini und Hochberg [Benjamini et al., 2001]: Sind die Nullhypothesen mit $H_1, \cdots, H_m$, die p-Werte mit $p_1, \cdots, p_m$ und die sortierten p-Werte mit $p_{1'}, \cdots, p_{m'}$ gegeben, dann sucht man zu einem gegebenen α das größte z. Die Defintion lautet:

$$p_{k'} \leq \frac{k}{m}\alpha \tag{2.63}$$

Damit werden alle $H_{i'}$ mit $i = 1, \cdots, k$ abgelehnt.

2.10 Signalweg- und Genfunktionsgruppen-Analysen

2.10.1 Signalweg-Software und Datenbanken

Nachdem eine Liste von differentiell exprimierten Genen erstellt wurde, können Signalwege-Analysen dabei helfen zu verstehen, welche Konsequenzen ein dereguliertes Gen auf ein Netzwerk von Interaktionspartnern (anderen Genen) haben kann. Verschiedene Datenbanken und Softwarelösungen stellen die Informationen über die

Mitglieder eines Signalweges bereit. Softwarewerkzeuge bieten können eine Verbindung zwischen den eigenen Daten und den Signalwegen herstellen und visualisieren. Zum Visualisieren wichtiger Signalwege wurde die frei erhältliche Software *Advanced-Pathway-Painter* (V. 2.08, GSA-Bansemer & Scheel GbR) verwendet. Der *Pathway-Painter* beinhaltet alle Signalwege von Biocarta (`http://www.biocarta.com/`), GenMapp (`http://www.genmapp.org/`) und KEGG (`http://www.genome.jp/kegg/`). Im *Pathway-Painter* hat man die Möglichkeit, eine definierte Liste von Genen über alle bekannten Signalwege zu legen und damit grafisch sichtbar zu machen, welche Gene aus der Liste in welchen Signalwegen zu finden sind. Eine farbkodierte Darstellung der Art der Expression (z. B. Grün heißt herunterreguliert (↓), Rot heißt hochreguliert (↑)) kann vorher festgelegt werden. Im Gegensatz zur Software GenMapp können keine Signalwege verändert oder neue erstellt werden.

2.10.2 Überrepräsentationsanalysen mit Gen-Ontologien

Eine weitere Möglichkeit, die zum Teil langen Genlisten zu analysieren, ist die *Gen-Ontologien* (GO) zu diesen Genen zu bewerten. GO bieten ein strukturiertes, kontrolliertes Vokabular zur Klassifikation der Gene für diverse Ebenen der Molekular- und Zellbiologie [Ashburner et al., 2000]. GO sind in drei Kategorien unterteilt: 1. Molekulare Funktion, 2. Biologischer Prozess und 3. Zelluläres Kompartiment. Die Kategorien bilden einen gerichteten azyklischen Graphen (DAG), wobei die individuellen Kategorien als Knoten dargestellt sind, die wiederum eine direkte Verbindung zu mehreren spezifischeren Knoten haben können. Die GO Annotations-(GOA) Datenbank enthält die komplette Annotation für Gene und deren Genprodukte für über 50 verschiedenen Spezies [Camon et al., 2004].
Insbesondere die Überrepräsentatierung einer GO-Kategorie, verursacht durch eine erhöhte Anzahl von differentiellen-exprimierten Genen zu dieser Kategorie, ist vom biologischen Interesse. Dabei gibt es verschiedene statistische Verfahren für die GO-Analyse. Das am meisten verbreitete statistische Verfahren zur GO-Analyse verwendet die hypergeometrische Verteilung [Breitling et al., 2004]. Dabei wird einfach nach einer Überrepräsentierung von individuellen GO-Kategorien (*Term-for-Term*-Ansatz) gesucht, und das für jeden Knoten individuell berechnet und mit Multiplen-Testen-Verfahren korrigiert. Das Ergebnis der GO-Analyse ist eine Liste von GO-Kategorien, die in der zuvor definierten Genliste (Studienset) signifikant überrepräsentiert sind.

Ist die Anzahl aller Gene (z. B. alle Gene auf einem *Microarray*) mit N gegeben, die Anzahl aller Gene, die zur getesteten GO-Kategorie gehören, mit n gegeben, die Anzahl der Gene aus dem Studienset mit K gegeben und die Anzahl der Gene aus dem Studienset, die zur getesteten GO-Kategorie gehören, mit k bezeichnet, so ergibt sich:

$$P(\kappa = x) = \frac{\binom{n}{x}\binom{N-n}{K-x}}{\binom{N}{K}} \tag{2.64}$$

Die Zufallsvariable ist mit κ gegeben, deren Realisierung der beobachtete Wert K ist. Leider hat dieses Verfahren den Nachteil, dass sobald eine GO-Kategorie überrepräsentiert ist, eine zugehörige spezifischere GO-Kategorie nur aus diesem Grund auch als überrepäsentiert deklariert wird.

Verfahren von Grossman et al.

Die Abhängigkeit konnten Grossman et al. [Grossmann et al., 2007] zeigen und haben gleichzeitig ein Verfahren entwickelt, dass die GO-Kategorien nicht isoliert voneinander betrachtet (*Parent-Child*-Verfahren). Daraus ergibt sich

$$P(\kappa = x | \kappa_{pa(t)} = K_{pa(t)}) = \frac{\binom{n}{x}\binom{N_{pa(t)}-n}{K_{pa(t)}-x}}{\binom{N_{pa(t)}}{K_{pa(t)}}} \tag{2.65}$$

wobei *pa* den *Eltern-Knoten* definiert. Man geht davon aus, dass jede Go-Kategorie nur einen Eltern-Knoten hat. Wenn mehr als ein Eltern-Knoten pro GO-Kategorie gegeben ist, muss die Anzahl an Eltern-Knoten zu einer GO-Kategorie t untersucht werden. Grossmann et al. haben dazu zwei Ansätze definiert.

1. *parent-child-union*-Verfahren:
   Für den ersten Ansatz definiert man die Menge der Eltern für eine Kategorie t in der Population und die Studienset-Menge als die Vereinigung der Gene, die mit Eltern von t annotiert sind:

$$
\begin{aligned}
P^{\cup}_{pa(t)} &:= \bigcup_{u \in pa(t)} P_u \\
S^{\cup}_{pa(t)} &:= S \cap P^{\cup}_{pa(t)}
\end{aligned}
\tag{2.66}
$$

Dafür seien $n_{pa(t)}$ und $K_{pa(t)}$ die Anzahlen der Gene, die als Eltern von der jeweiligen Menge annotiert sind.

2. *parent-child-intersection*-Verfahren:
   Hier wird die Anzahl der Eltern zu einer Kategorie t als die Schnittmenge von Genen, die als Eltern von der jeweiligen Menge von t annotiert sind, angegeben:

$$
\begin{aligned}
P^{\cap}_{pa(t)} &:= \bigcap_{u \in pa(t)} P_u \\
S^{\cap}_{pa(t)} &:= S \cap P^{\cap}_{pa(t)}
\end{aligned}
\tag{2.67}
$$

   Berechnet wird die Anzahl der Gene, die zu allen Eltern annotiert werden können.

Grossman et al. konnten mit beiden Ansätzen zeigen, dass — abhängig von der Definition des Studiensets —, beide Ansätze im Vergleich zu Standardverfahren weit robustere Ergebnisse liefern. Sie begründen dies unter anderem damit, dass die Standardverfahren die Struktur des GO-DAG ignorieren und es so zu einer großen Anzahl von falschen Positiven kommt.
Bezüglich des multiplen Testen Problems empfehlen Grossmann et al., die Methode von Westfall und Young anzuwenden [Westfall and Young, 1993]. Dieses Verfahren kontrolliert die FWER. Alle Verfahren, die bei Grossman et al. [Grossmann et al., 2007] vorgestellt wurden, und einige Standardverfahren sind im *Ontologizer* (`http://compbio.charite.de/index.php/ontologizer2.html`) enthalten. In der vorliegenden Arbeit wurde für alle GO-Analysen der *Ontologizer* mit dem *parent-child-intersection*-Verfahren verwendet. Die Korrektur der p-Werte wurde mit der Methode von Westfall und Young [Westfall and Young, 1993] durchgeführt.

2.11 Qualitätskriterien zur Beurteilung der *Microarray*-Ergebnisse

Grafische Visualisierung

Die Qualität der verschiedenen Verfahren lässt sich durch unterschiedliche Parameter charakterisieren. Boxplots, Q-Q-Plots und Histogramme sind für die Visualisierung und Interpretation der Datenverteilung.

Sensitivität und Spezifität

Als weiteres wichtiges Qualitätswerkzeug werden Sensitivität und Spezifität zum Finden differentieller Gene verwendet. Ein Gen, das als differentiell-exprimiert simuliert wurde, aber durch einen statistischen Test nicht gefunden wurde, wird als falsch negativ (RN) definiert. Analog dazu ist ein nicht differentiell simuliertes Gen, das aber als differentiell durch einen statistischen Test gefunden wurde, als falsch positiv (RP) definiert. Die wahren Gene, die als differentiell simuliert wurden, und als solche erkannt wurden sind die richtig Positiven (RP). Die Gene, die als nicht differentiell simuliert und als solche erkannt wurden, werden als richtig Negativ (RN) festgelegt.
Die Sensitivität zeigt, wie wahrscheinlich ein als differentiell-exprimiert simuliertes Gen als solches gefunden wird. Sie wird wie folgt berrechnet:

$$Sens := \frac{RP}{FN + RP} \tag{2.68}$$

Die Spezifität schätzt wie wahrscheinlich ein als nicht differentiell exprimiert simuliertes Gen als solches gefunden wird. Sie wird wie folgt berechnet:

$$Spec := \frac{RN}{RN + FP} \tag{2.69}$$

Für das Berechnen von Sensitivität und Spezifität ist der Nenner bei allen Methoden gleich (Sensitivität: 500, Spezifität: 21783). Für die Beurteilung von Sensitivität und Spezifität ist es daher ausreichend, die Anzahl der richtig positiven und richtig negativen Ergebnisse zu betrachten. Zusätzlich kann der Unterschied zwischen den Normalisierungsmethoden unabhängig von einem *Cut-Off*-Wert (die

verwendeten FDR-Filterkriterien) quantifiziert werden. Dazu verwendet man die *Receiver-Operating-Characteristics* (ROC)-Kurven, die Sensitivität und Spezifität als Wertepaare für verschiedene FDR-*Cut-Offs* darstellt. Hierfür werden zu jedem FDR-*Cut-Off* Sensitivität und Spezifität nach den Formeln 2.68 und 2.69 berechnet. Jeder FDR-Wert erhält einen Punkt in der ROC-Kurve. Die Spezifität (1-Spezifität) wird auf die Abzisse und die Sensitivität auf der Ordinate aufgetragen. Je größer der Abstand der ROC-Kurve von der Diagonalen ist, desto besser ist die verwendete Normalisierungsmethode.
Mit der Sensitivität und der Spezifität kann der Einfluss verschiedener Normalisierungsmethoden auf die differentielle Expression von simulierten Daten gemessen werden.
Außerdem werden die FDR-Werte, die FC-Werte und die GO-Analyse für die verschiedenen Analysen dieser Arbeit als Qualitätskriterien herangezogen.

Kapitel 3

Ergebnisse

3.1 Auswertung des Datensatzes von Küppers et al.

3.1.1 Beschreibung der statistischen Analyse-Strategie

Für die Reanalyse der Genexpressionsdaten unseres Kooperationspartners Ralf Küppers wurden neuere statistische Verfahren angewendet, zum Beispiel *Microarrays* vom Typ HGU95A mit 12625 *Probesets*. Um die Qualität der *Microarrays* zu prüfen wurde die Software von Affymetrix GCOS1.4 (MAS5.0 Algorithmus) eingesetzt. Bevor jeder *Microarray* normalisiert wurde (siehe Kapitel 2.5.4), wurde ein Sollwert, den sogenannten Zielwert, für eine definierte Liste von Genen bestimmt. Affymetrix gibt eine Liste von 100 Kontrollgenen vor, die nicht differentiell exprimiert sein sollen und sich so für eine Normalisierung eignen. Dafür wurde ein Sollwert ausgewählt, in unserem Fall 2000, und ein Faktor mit, dem die Kontrollgene multipliziert werden, berechnet müssen, damit jedes der 100 *Probesets* mindestens einen Wert von 2000 erreicht. Die anderen *Probesets* wurden mit diesem Faktor skaliert. Damit die *Microarrays* untereinander vergleichbar sind, sollte der Skalierungsfaktor um nicht mehr als den Faktor 3 bei den *Microarrays* abweichen.
Die Software von Affymetrix wurde ausschließlich zur Qualitätskontrolle der *Microarrays* herangezogen.
Für alle weiteren statistischen Analysen wurden R [Ihaka and Gentleman, 1996] und Bioconductor [Gentleman et al., 2004] verwendet. Für die *Probe*-Normalisierung wurde das in Kapitel 2.5.4 beschriebene VSN von Huber et al. [Huber et al., 2002] eingesetzt.

Für die differentielle Genexpression wurde der Vergleich von HL-Linien und GC-B-Zellen aus der Veröffentlichung von Küppers et al. [Küppers et al., 2003] gewählt. Bevor ein statistischer Test — in unserem Falle der t-Test — angewendet wurde, musste zunächst die Datendimension minimiert werden. Mit einem Vorfilter wurde bei der FDR das multiple Testen-Problem gelöst. Beim Intensitätsfilter musste in mindesten 25 % der Fälle ein Expressionswert von $\geq$100 erfüllt sein. Ein Varianzfilter war aufgrund der sehr unterschiedlichen Gruppengrößen nicht sinnvoll. Nach dem Vorfiltern wurde ein 2-Stichproben-t-Test durchgeführt. Das multiple Testen-Problem wurde mit der FDR [Benjamini et al., 2001] durchgeführt (siehe Kapitel 2.9.2).
Um die Stärke der differentiellen Expression angeben zu können, wird zu allen *Probesets* der FC berechnet (siehe Kapitel 2.9.1). Zur Visualisierung bestimmter Genlisten wurden *Heatmaps* erstellt (Bioconductor Paket: geneplotter) und Clusteranalysen durchgeführt (R Paket: hclust). GO- und Signalwegeanalysen dienen zur Beurteilung der biologischen Relevanz.

3.1.2 Betrachtung der differentiell-exprimierten Gene

HL-Zelllinien (L1236, L428, KMH2, HDLM2) und insgesamt 20 der für diese vier korrespondierenden normalen reifen B-Zellen, und zwar fünf zu jeder Identität (Keimzentrum: Zentrocyten, Zentroblasten, naive B-zellen und Gedächtnis-B-Zellen), wurden verglichen. Küppers et al. [Küppers et al., 2003] identifizierten 243 *Probesets* mit einem z-Score ≥ 2 und 374 *Probesets* mit einem z-Score ≤ -2, also insgesamt 617 *Probesets*. Der z-Scores ist in Kapitel 2.9.1 definiert.
Durch die Reanalyse dieses Datensatzes konnten zusätzliche differentielle Gene gefunden werden. So konnten 356 *Probesets* mit einem FC ≥ 2 und 333 *Probesets* mit einem FC $\leq$-2 mit einer FDR ≤ 0.05 identifiziert werden (insgesamt 689 *Probesets*). Die Schnittmenge der gefundenen *Probesets* liegt bei den Analysen bei 390 *Probesets*. 227 *Probesets* waren zunächst in der Reanalyse nicht zu finden. Wurde allerdings nur der FC-Filter verändert, konnten bis auf sieben *Probesets* alle *Probesets* gefunden werden, die zwar einen kleineren FC zwischen beiden Gruppen zeigten, aber immer noch einen FDR ≤ 0.05 haben. Bei genauerem Betrachten der sieben Probesets, war eine große Varianz zwischen den beiden Gruppen bei diesen *Probesets* festzustellen, sodass fragwürdig ist, ob diese Gene wirklich robust differentiell-exprimiert waren. Zwei Gene (ID2, KCNC4) wurden außerdem durch andere *Probesets* in der

Reanalyse-Genliste gefunden. Also blieben letztendlich fünf *Probesets*, die bei der Reanalyse nicht gefunden wurden. Die nicht in der Reanalyse gefundenen fünf *Probesets* wurden allerdings in der Arbeit von Küppers et al. nicht als differentiell-exprimiert bestätigt. Bei den fünf nicht gefundenen *Probesets* handelt es sich um die Gene DEFA1, ESRRB, IGKV1-13, SNX13 und LOC94431.
299 *Probesets* sind nicht in der publizierten Genliste (Tabelle 6.1) enthalten. In Tabelle 3.1 ist in Abhängigkeit des FC die Zusammensetzung der Genliste aufgeführt:

FC	FDR	Anzahl der Gene
≥ 2 or $\leq$ -2	$\leq$ 0,05	299
≥ 3 or $\leq$ -3	$\leq$ 0,05	41
≥ 4 or $\leq$ -4	$\leq$ 0,05	17
≥ 5 or $\leq$ -5	$\leq$ 0,05	6

Tabelle 3.1: Die 299 differentiell-exprimierten *Probesets*, die nur nach Reanalyse identifiziert wurden. Der Schwerpunkt der Regulationsstärke liegt bei einem FC zwischen 2 und 3 im Vergleich von HL-Linien und normalen B-Zellen.

Als nächstes wurden die Gene betrachtet, die eine FDR$\leq$ 0.05 und einen FC von ≥ 4 oder ≤ -4 haben. Dabei handelt es sich um zehn herunterregulierte Gene, im Vergleich von HL-Linien zu normalen B-Zellen, und sieben hochregulierte Gene (Tabelle 3.2).

FC	Gensymbol	Affymetrix ID	Gen Funktion
5,8	CYP19A1	987_g_at	cytochrome P450
5,2	MLLT11	36941_at	mixed-lineage leukemia
4,6	CCL5*	1403_s_at	chemokine ligand 5
4,6	CCR7*	1097_s_at	chemokine receptor 7
4,4	IL6	38299_at	interleukin 6
4,2	NFIL3	37544_at	nuclear factor
4,2	CCND2	1983_at	cyclin D2
-4	IGHA1/IGHA2	33499_s_at	immunoglobulin
-4,1	TUBA1	36591_at	tubulin, alpha1
-4,3	CCDC69	34183_at	coiled-coil domain containing 69
-4,4	SELL	245_at	selectin L
-4,8	CD27 (TNFRSF7)	38578_at	tumor necrosis factor 7
-4,9	—	33613_at	fusion protein IRTA1/IGA1
-5,8	CR2*	36874_at	complement component receptor 2
-5,9	PTPRCAP	32070_at	protein tyrosine phosphatase
-7	RGS13*	35459_at	regulator of G-protein signalling 13
-9,4	TCL1A*	39318_at	T-cell leukemia/lymphoma 1A

Tabelle 3.2: Die 17 Gene, die nach Reanalyse identifiziert wurden (FDR ≤ 0.05), mit dem größten FC zwischen HL-Linien und normalen B-Zellen. Gene die mit „*“ markiert sind, wurden im Vergleich zwischen cHLs und Keimzentrums B-Zellen als signifikant-reguliert gefunden.

Die visuelle Darstellung dieser 17 Gene in einer Heatmap mit zusätzlicher Clusteranalyse über die Gene verdeutlicht den Expressionsunterschied zwischen cHL und normalen B-Zellen (Abb.3.1).

Fünf der 17 in Tabelle 3.2 beschriebenen Genen zeigten zusätzlich eine signifikante differentielle Expression in cHLs im Vergleich zu Keimzentrum-B-Zellen [Brune et al., 2008]. In Tabelle 3.3 sind diese Gene dargestellt und mit dem FC des Vergleiches von cHL und Keimzentrum-B-Zellen ergänzt. Da diese Gene nicht nur in dem als Modell dienenden HL-Linien gefunden wurden, sondern auch in dem primären Patientenmaterial von cHL-Tumoren, wurden diese Gene hinsichtlich ihrer biologischen Funktion und deren Signalwege genauer betrachtet.

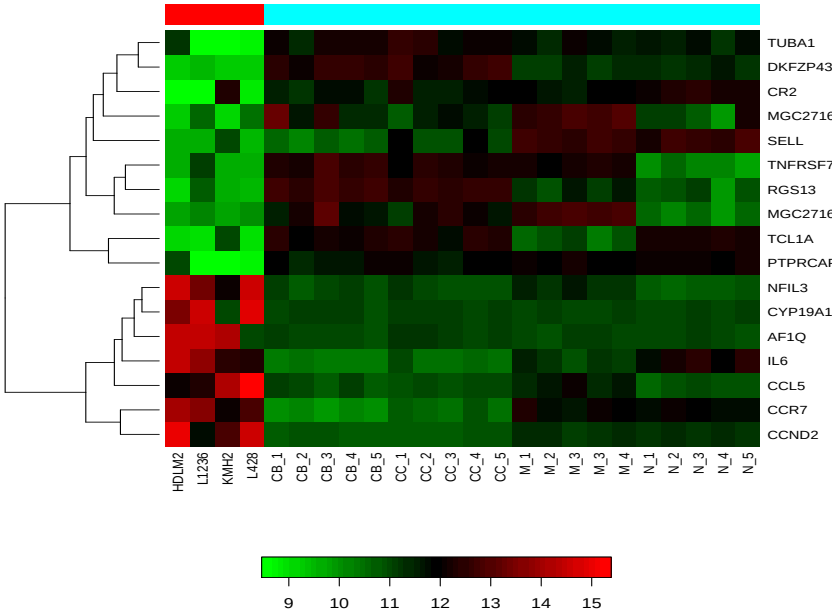


Abbildung 3.1: Heatmap der 17 Gene, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probesets*/Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert.

FC HL/cHL vs. B-Zellen	Gensymbol	Affymetrix Probeset ID HGU95av2	Gen Funktion
4,6 / 9,0	CCL5	1403_s_at	chemokine, ligand 5 (C-C motif)
4,6 /4,5	CCR7	1097_s_at	chemokine, receptor 7 (C-C motif)
-5,8 / -3	CR2	36874_at	complement component, receptor 2, (3d/EBV)
-7,0 / -4,8	RGS13	35459_at	signalling 13, regulator of G-protein
-9,4 / -9,3, -8,1	TCL1A	39318_at	T-cell leukemia/lymphoma 1A

Tabelle 3.3: Die fünf Gene mit dem höchsten FC zwischen HL-Linien und normalen, reifen B-Zellen, die nur nach Reanalyse identifiziert werden konnten. Diese Gene haben eine FDR ≤ 0.05 und wurden als signifikant-differentiell exprimiert in primären HL-Patienten gefunden ([Brune et al., 2008])

.

Mit Ausnahme eines Gens (RGS13) wurden die anderen durch andere wissenschaftliche Gruppen mittlerweile publiziert. CCL5 ist ein Chemokin, das in inflammatorischen und immunoregulatorischen Prozessen involviert ist. Es konnte gezeigt werden, dass CCL5 für die Rekrutierung der Mastzellen durch die HRS-Zellen eine wichtige Rolle spielt. Außerdem spielen Mastzellen eine aktive Rolle in der Tumorgenese des cHLs. [Fischer et al., 2003 2003].
CCR7 gehört zu der C-C-Chemokine-Rezeptor-Familie und ist wichtig für die Funktion und Struktur des Thymus, der Zielsuche von T-Zellen im Lymphknoten und in der Migration von dendritischen Zellen in entzündlich-induzierten Lymphknoten [Förster et al., 2008]. CCR7 ist in HRS-Zellen stark exprimiert, was an der konstitutiv-aktiven NFkappaB Aktivität liegen kann [Mathas et al., 2002; Höpken et al., 2002].
CR2 ist auch unter dem Namen CD21 bekannt und wird von reifen B-Zellen exprimiert. Der CR2-Rezeptor spielt bei der B-Zellen-Aktivierung und Reifung eine Rolle [Yefenof et al., 1976]. HRS Zellen, die EBV positiv sind, zeigen gleichzeitig eine CR2-Expression und sind damit häufig mit dem HL assoziiert [Jiwa et al., 1992].

TCL1A ist hauptsächlich in ausdifferenzierten B- und T-Zell-Lymphozyten exprimiert. Einige Leukämie- und Lymphomarten zeigen eine einmalige Translokation, bei der TCL1 involviert ist [Virgilio et al., 1994]. TCL1A ist in mehr ausdifferentierten B-Zellen exprimiert und in den letzten Stadien der B-Zell-Differentzierung [Narducci et al., 2000] herunterreguliert. Es hat sich außerdem gezeigt, dass TCL1A als diagnostischer Marker für den Differentzierungsgrad von B-Zell-Lymphomen eingesetzt werden kann [Herling et al., 2007].
RGS13 ist das einzige der fünf Gene, das noch nicht im Zusammenhang mit dem cHL-Lymphom publiziert wurde. RGS13 ist hauptsächlich in Tonsillen, Thymus, Lymphknoten und Milzgewebe exprimiert. Das Mitglied von Regulatoren der G-Protein-Signalgeber-Familie ist schon in MALT Lymphomen als stark exprimiert beschrieben [Chng et al., 2009]. RGS13 ist auch in Keimzentrum-B-Zellen und in einigen Brukitt-Lymphom-Zelllinien [Cahir-McFarland et al., 2004] exprimiert. Die Expression von RGS1 und RGS13 in humanen Keimzentrum-B-Zellen kann ein Grund sein, warum diese Zellen nur schwach auf Chemotherpeutika ansprechen [Bansal et al., 2008]. Die generelle biologische Funktion dieses Gens ist allerdings unbekannt.

Nach einem genaueren Betrachten einen Teils der differentiellen Gene, stellt sich die Frage, welche Funktionen und Prozesse zusätzlich gefunden werden können. Deshalb wurden die Gesamtliste (689 *Probesets*) und die Genliste, die nur nach Reanalyse gefunden wurde, mit der GO-Analysemethode von Grossman et al. [Grossmann et al., 2007] verglichen.

3.1.3 Betrachtung der GO-Analyse

Zunächst wurde die Gesamtliste der Gene betrachtet, die nach Reanalyse gefunden wurden. Von diesen 689 *Probesets* konnten 458 Gene einer GO-Kategorie zugeordnet werden. Die Kategorien, die als signifikant eingestuft wurden (Westfall-Young *Step Down* mit Grundeinstellung: 0,1) finden sich in allen drei Haupt-GO-Kategorien wieder:

1. Biologische Prozesse: *Death*, *Immune-System-Prozess*, *Response-to-stimulus* und *Respirators-Burst*

2. Zelluläres Kompartiment: *NADPH-Oxidase-Complex*

3. Molekulare Funktion: *Electron-Carrier-Activity*

Abbildung 6.1 stellt die grafische Darstellung dieser signifikanten GO-Kategorien dar. Die als signifikant klassifizierten GO-Kategorien sind farblich hervorgehoben. Betrachtet wurde das Verhältnis der Gene, die zu einer GO-Kategorie gesamt annotieren (Grundlage sind hierbei alle Gene des jeweils verwendeten *Microarray*-Typs), zu den Genen aus der zu untersuchenden Genliste.
Besonders die GO-Kategorie *Immune-System-Prozess* ist wegen der Hauptaussage von Küppers et al., dass die HL-Linien die komplette B-Zell Identität verlieren, von besonderem Interesse. In diesem Fall bedeutet der Verlust der B-Zell-Identität, dass alle typischen B-Zell Gene in den HL-Linien nicht exprimiert sind. Die GO-Katerogie kann hier durch die Reanalyse zusätzliche Gene zu diesem Bereich beitragen.
In einem zweiten Ansatz betrachtet man die GO-Analyse mit den zusätzlichen 299 *Probesets* (Abb.6.2), die nur nach Reanalyse gefunden wurden. Von diesen 299 *Probesets* konnten 213 Gene einer GO-Kategorie zugeordnet werden. Die Kategorien, die als signifikant eingestuft wurden (Westfall-Young-*Step-Down* mit Grundeinstellung: 0,1) sind folgende biologische Prozesse: *Respirators-Burst* und *Cytokine-mediated-Signaling-Pathway.*
Zwei GO-Kategorien sollten im Detail untersucht werden. Die biologischen Prozesse, die GO-Kategorien *Immune-System-Prozess* und *Cytokine-mediated-Signaling-Pathway.*
Der Zytokine/Zytokine-Rezeptor-Signalweg (KEGG ID:has04060) ist einer der wichtigsten Signalwege für die intrazelluläre Kommunikation und Chemotaxis. Insgesamt konnten 11 Gene aus den zusätzlichen 299 gefundenen *Probesets* diesem Signalweg zugeordnet werden (Abb.6.3). Die Gene CCL5 und CCR7 sind schon in Tabelle 3.3 enthalten. Die Expression der Gene FAS, IL6 und IL13RA sind in HLs beschrieben [Ghilardi et al., 2002; Nagel et al., 2005; Dutton et al., 2004].

Folgende Gene sind in HLs noch nicht beschrieben:

1. IFNGR1 (Interferon-Gamma-Rezeptor 1)und IL24 (Interleukin) sind herunterreguliert (FC= -2,8/-2,7) in HL-Linien im Vergleich zu normalen, reifen B-Zellen.

2. CX3CL1 und TNFSF9 (Chemokine), IL15 und IL1R1 (Interleukine) sind hoch-

reguliert (FC= 3/2/2,2/2) in HL-Linien im Vergleich zu normalen, reifen B-Zellen.

Allerdings werden diese Gene zum Teil mit anderen Tumorerkrankungen in Verbindung gebracht:

1. IFNGR1, CX3CL1, IL1R1 sind noch in keinem Zusammenhang mit Tumorerkrankungen publiziert worden.

2. IL24 spielt eine Rolle in Melanoma-Zellen. Die Wirkung als Antitumor Aktivität in Leukämien und Lymphomen ist beschrieben [Dong et al., 2008].

3. IL15 kann stimulierend auf die Expression des Zell-Tod-Überlebensgen BCL2 in Zellen der T-Zell-Lymphome der Haut wirken [Qin et al., 2001].

4. TNFSF9: Es besteht eine starke Expression in Mantel-Zell (MALT) Lymphom-Zelllinien.

Ein weiterer wichtiger Signalweg ist der des B-Zell-Rezeptors (Abb. 6.4). Damit konnten zusätzliche Gene gefunden werden, die aus diesem Signalweg herunterreguliert sind. Sie stärken die These von Küppers et al. [Küppers et al., 2003]. Für die extensive Herunterregulation der B-Linien-spezifischen-Gene im Genexpressionsprogramm der HRS-Zellen wurden folgendene herunterregulierte B-Zell-Marker zusätzlich in den HL-Linien gefunden: CR2, LYN, FCGR2B, PIK3CG und BCL6. Außerdem wurden B-Zell Marker gefunden, die in HL-Linien hochreguliert sind: STAT1, CCND2 und ATP2B4. Außer FCGR2B sind alle anderen zusätzlich gefundenen B-Zell Marker mittlerweile publiziert. Die Rolle von FCGR2B in HL ist noch unklar.

3.2 Auswertung des Datensatzes von Piccaluga et al.

3.2.1 Beschreibung der statistischen Analyse-Strategie

Bei der Auswertung des Datensatzes von Piccaluga et al. [Piccaluga et al., 2007] wurden ähnliche Analysestrategien angewendet wie bei dem Datensatz von Küppers et al. [Küppers et al., 2003]. Im Folgenden die Unterschiede zu den in Kapitel 3.1 vorgestellten Verfahren:

1. Bei dem Affymetrix-*Microarray*-Typ handelt es sich um den HGU133Plus2.0 mit 54708 *Probesets*.

2. Zwei Vergleiche wurden reanalysiert, zum einem der Vergleich zwischen T-Zell-Lymphomen Typ NOS (Not Other Spezified) und normalen T-Zellen vom Typ CD4 und CD8, zum anderen die T-Zell-Lymphome Typ NOS verglichen mit den T-Zell-Lymphomen vom Typ AITL (Angioimmunoblastic T-cell Lymphoma) und ALCL (Anaplastic Large Cell Lymphoma).

3. Bei dieser Analyse wendet man einen Intensitätsfilter und einen Varianzfilter an. Dabei müssen in mindesten 25 % der Fälle ein Expressionswert von ≥ 100 erfüllt sein und ein Quartilsabstand der log_2-Intensitätswerte muss mindestens 0,25 betragen. Das gilt für beide Vergleiche.

3.2.2 Clusteranalyse

Zuerst wird das *Unsupervised*-Clusterverfahren — so wie in Kapitel 2.7 beschrieben — auf alle T-Zell-Lymphome angewendet (Abb.3.2). Die normalen T-Zellen bleiben bei dieser Clusteranalyse unberücksichtigt. Da die Clusteranalyse mit der von Piccaluga et al. verglichen werden soll, wird dieselbe Stichprobe von Patientendaten für die Clusteranalyse verwendet.
Die Gene für die Clusteranalyse bei Piccaluga et al. wurden so gewählt, dass mindestens ein doppelter Unterschied über alle Expressionswerte besteht. Insgesamt 417 Gene haben diese Grenze erfüllt. Für das hierachische Clustern wurde die Methode *average-linkage* gewählt, für die Distanzberechnung die Pearson-Korrelation.
Bei den reanalysierten Daten wurden die Standardabweichung über alle T-Zell-Lymphom-Expressionswerte mit $\geq 1,5$ gewählt, um die Gene auszuwählen, über die das Clustern erfolgen sollte. Damit wurde über die 291 Gene mit $\sigma \geq 1,5$ ein *unsupervised Clustering* durchgeführt, mit der Methode *average-linkage*, und als Distanzberechnung die Manhattan-Distanz (3.2).
Dabei entsteht ein ähnlich herterogenes Clusterbild, wie es schon Piccaluga et al. [Piccaluga et al., 2007] festgestellt haben. Die einzelnen T-Zell-Lymphomgruppen sortieren sich nicht nach Gruppenzugehörigkeit. Das kann verschiedene Ursachen haben. Um auszuschließen, dass die Clusteranaylse und die Auswahl der Gene einen Einfluss auf die Robustheit der Cluseranalyse haben, wird über den hier in der Arbeit erzeugten hierachischen Cluster eine Stabilitätsprüfung — wie bei Suzuki et al.

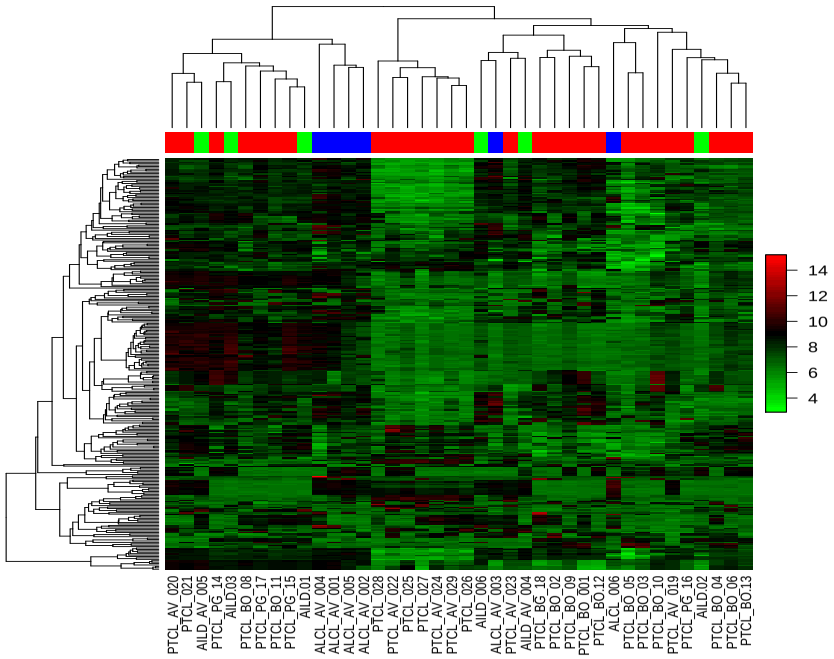


Abbildung 3.2: Unsupervised Clustering der 291 *Probesets* mit $\sigma \geq 1,5$ nach Reanalyse der Daten von Piccaluga et al.. Spalten repräsentieren individuelle Patienten, Zeilen entsprechen den verschiedenen dargestellten Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert. Die Unterscheidung der Gruppen unter dem Dendrogramm der Spalten erfolgt durch zusätzlichen Farbcode: Rot für PTZLs, Grün für AILTs und Blau für ALCLs.

[Suzuki and Shimodaira, 2006] beschrieben — durchgeführt. Die Stabilitätsüberprüfung zeigt (Abb. 6.5), dass die Ordnung der T-Zell-Lymphome aufgrund der hohen AU-Werte nicht zufällig sein kann und dass damit vermutlich ein anderer Mechanismus hinter diesem Muster steckt.

3.2.3 Betrachtung der differentiell-exprimierten Gene

Bei der Betrachtung der differentiellen Expression der T-Zell-Lymphome Typ NOS im Vergleich zu normalen T-Zellen sind aufgrund der sehr hohen biologischen Heterogenität der Tumore Gene auszuwählen, die homogen über eine Identität oder Erkrankung exprimiert sind und einen relativen großen Unterschied (FC) zwischen den Identitäten zeigen. Deswegen betrachtet man nicht die Genliste mit $FDR \leq 0,05$ und $FC \geq 2$ oder ≤ -2 (3575 *Probesets*), sondern mit $FC \geq 4 oder \leq -4$ (599 *Probesets*). Zur Genliste ist später auch die GO-Analyse zu betrachten. Von den 599 *Probesets* findet man 363 nicht in der Arbeit von Piccaluga et al. [Piccaluga et al., 2007]. Zu dieser Genliste wird eine gesonderte GO-Analyse durchgeführt. In Tabelle 6.2 sind die 30 *Probesets* zusammengefasst, die mindestens einen $FC \geq 10$ oder ≤ -10 zeigen und nur nach Reanalyse zu finden sind.
Diese Gene sind homogen in den jeweiligen Gruppen exprimiert und sind vermutlich als Markergene für die Diagnostik geeignet. Allerdings beruht das nur auf der mRNA-Expressionsebene. Ob diese Gene wirklich als diagnostische Marker fungieren können, müssen zusätzliche Validierungen und Proteinnachweise zeigen.
In Abbildung (3.3) sind diese 30 *Probesets* in einer Heatmap mit Clusteranalyse dargestellt. Man erkennt dort einen deutlichen Expressionsunterschied zwischen PTCL Typ NOS und normalen T-Zellen. Dabei sind vier Gene runtereguliert und 26 hochreguliert.
Außer einem Gen (CLU [Saffer et al., 2002]) sind alle 29 Gene noch nicht in T-Zell-Lymphomen beschrieben. Ein Teil dieser Gene wurde in anderen Tumorarten beschrieben. Die folgende Liste stellt diesen Zusammenhang dar:

Gene, die schon in anderen Tumoren beschrieben sind:

1. Im PTCL hochregulierte Gene:
   CCL19, CHI3L1, CXCL10, CXCL11, CXCL14, CYP1B1, GPNMB, MYLK,

RARRES1 und SEPP1

2. Im PTCL runterregulierte Gene:
   AREG/AREGB, FOSB und IL8

Gene, die noch nicht in anderen Tumoren beschrieben sind:

1. Im PTCL hochregulierte Gene:
   C1QB, C3, CFH/CFHR1, COL1A1, CTHRC1, CXCL9, FAM26F, FGL2, KCTD12, LUM, SLAMF8 und SLC40A1

2. Im PTCL runterregulierte Gene:
   ZBTB10

Piccaluga et al. konnten insgesamt 155 Gene identifizieren (91 runterregulierte, 64 hochregulierte), die zwischen PTCL Typ NOS und normalen T-Zellen dereguliert waren (mit einem $z - score \geq 2$).
Bei der Reanalyse wurden außer drei Gene alle Genen wiedergefunden. Bei zwei Genen hatte sich die Annotation geändert. Diese wurden deshalb zunächst nicht gefunden. Ein Gen hatte einen geringeren FC als 2 in der Reanalyse. Ein Gen, das nach der Reanalyse unter den ersten 30 gelandet ist, kommt zwar bei Piccaluga et al. (Supplemantary [Piccaluga et al., 2007]) auch vor, aber mit einer anderen Affymetrix ID, die nicht zum Gensymbol passt (Affymetrix ID: $228562\_at \neq$ ZBTB10). Nicht zu klären war, ob es sich um einen Fehler bei der Affymetrix-ID handelt oder ob ein Fehler bei der Annotation vorlag.
Auffällig in der 30er Genliste ist — wie schon bei Küppers et al. — das Vorhandensein von einigen Cytokinen und Chemokinen (CXCL9,10,11,14 und IL8). Eine GO-Analyse sollte klären, ob wirklich eine Überrepräsentierung dieser GO-Kategorie vorliegt.

3.2.4 Betrachtung der GO-Analyse

Die GO-Analyse wurde für zwei Genlisten durchgeführt. Einmal für die Gesamtliste der differentiell-exprimierten Gene, die nach Reanalyse einen $FC \geq 4$ oder ≤ -4 (599 *Probesets*) haben, und zweitens für die *Probesets*, die nur nach Reanalyse gefunden wurden ($FC \geq 4$ oder ≤ -4, 363 *Probesets*). Die FDR ist bei beiden Genlisten

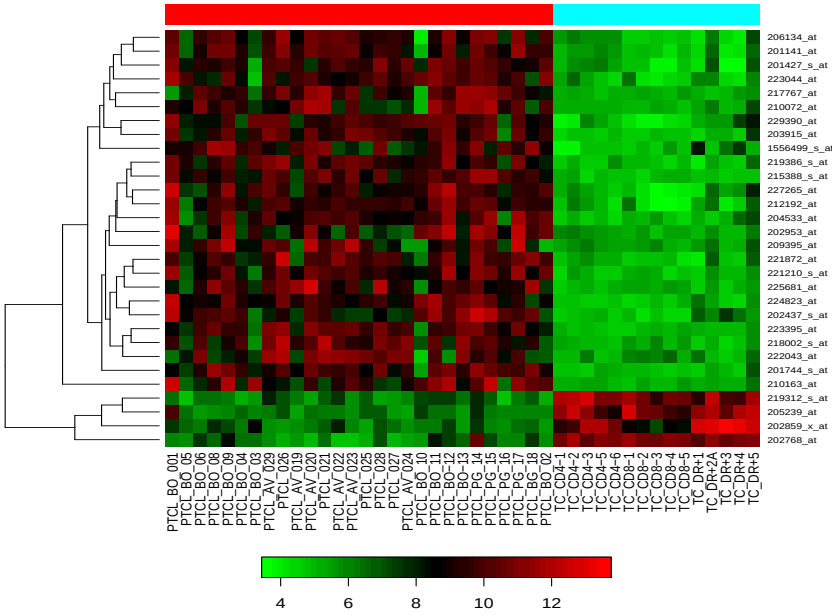


Abbildung 3.3: Heatmap der 30 *Probesets*, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probsets*/Genen. Die absoluten Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbkodiert.

gleich ($FDR \leq 0,05$). In den Abbildungen 6.6 und 6.7 sind die signifikanten GO-Kategorien (Westfall and Young *Step Down* $\leq 0,1$) farbig markiert.
In der Genliste mit 599 *Probesets* wurden folgende GO-Kategorien gefunden:

1. Biologischer Prozess:
   Locomotion, *Biological-Regulation*, *Localisation-of-Cell*, *Lipid-Transport*, *Skin-Development*, *Response-to-external-Stimulus*, *Biological-Adhesion*

2. Molekulare Funktion:
   Extracellular-Matrix-structural-constituent, *Cytokine-Aktivity*, *G-coupled-Receptor-binding*

3. Zelluläre Komponente:
   Extracellular-Region

In der Genliste mit 363 *Probesets* wurden folgende GO-Kategorien gefunden:

1. Biologischer Prozess:
   Locomotion, *Biological-Regulation*, *Gas-Transport*

2. Molekulare Funktion:
   Extracellular-Matrix-structural-constituent, *Cytokine-Aktivity*, *G-coupled-Receptor-binding*

3. Zelluläre Komponente:
   Extracellular-Region

In beiden GO-Analysen ist die Zytokin-Aktivität signifikant überrepräsentiert. Deswegen werden im Folgenden diese 12 Gene genauer betrachtet.
Ein Teil der Gene wurden schon bezogen auf die 30er Genliste beschrieben (CXCL9,10,11,14 und IL8). Bei den restlichen sieben Genen ist bisher nur eins in T-Zell-Lymphomen beschrieben (CCL2 [Mahadevan et al., 2005]). Die restlichen sechs Gene (CCL18, CCL19, CCL21, CCL8, CXCL12, IL18) sind zum Teil bei anderen Tumorenerkrankungen beschrieben. Dabei ist IL8 das einzige Gen, das runterreguliert ist.
Beim Vergleich von PTCL und den anderen T-Zell-Lymphomen AITL und ALCL wurden nur sieben zusätzliche Gene (S100A8, S100A9, LGALS1, IL1R2, TIMP1, PLTP, S100A11) gefundenen (FC ≤ -4, FDR $\leq 0,05$), die sich aber aufgrund ihrer

heterogenen Expression wahrscheinlich nicht als diagnostische Marker eignen (siehe Abbildung 3.4). Piccaluga et al. fanden 28 Gene, die differentiell zwischen PTCL und AILT/ALCL sind, allerdings mit einem zum Teil sehr niedrigen z-score (3,7-1,3, wobei nur acht Gene einen z-score ≥ 2 haben). Die Schnittmenge enthält nur 4 Gene (MT1E, MAP2K2, SOCS3, BCL3), die in beiden Analysen gefunden wurden.

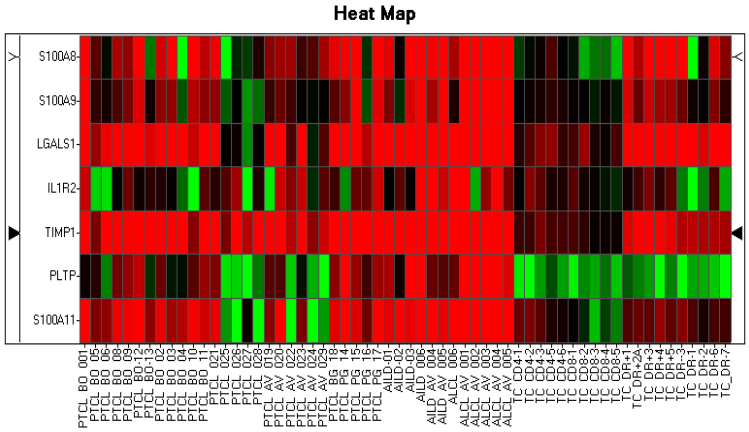


Abbildung 3.4: Heatmap der 7 Gene, die nur nach Reanalyse gefunden wurden und den größten FC haben. Spalten repräsentieren die Stichprobe, Zeilen entsprechen den *Probsets*/Genen. Die Expressionswerte sind in einer logarithmischen Skala zur Basis 2 farbcodiert.

Die GO-Analyse bei Piccaluga et al. wurde mit EASE (`http://bioinformatics.cau.edu.cn/easygo/`) erstellt und berechnet. Die Genlisten wurden getrennt nach Hoch- bzw. Runterregulation untersucht. Die für diese Arbeit erstellte GO-Analyse stimmte nicht mit denen von Piccaluga et al. überein. Auffällig bei Piccaluga et al. sind die mehrheitlich auftretenden großen Bonferoni-Werte (≥ 0.1) und das Beschränken auf die GO-Kategorie Biologische-Prozesse.

3.3 Zusammenfassung

Zusammengefasst zeigte sich, dass mit den aktuellen statistischen Verfahren die zuvor publizierten Gene reproduziert werden konnten und zusätzliche deregulierte Gene identifizierbar waren. Die inzwischen erfolgte Veröffentlichung einiger Gene in anderen Publikationen, zeigt die Wichtigkeit der Reanalyse und dient gleichzeitig

der Validierung der Ergebnisse. Küppers et al. haben 14 Gene gefunden, die der GO-Kategorie *Cytokine-mediated-Signaling-Pathway* zugeordnet wurden. Nach der Reanalyse kamen 11 weitere Gene hinzu, die die Wichtigkeit dieses stark deregulierten Signalweges betonen.
Die Reanalyse der Daten von Piccaluga et al. zeigten ähnliche Ergebnisse. Die Diskrepanz der GO-Analyse hat einen besonderen Stellenwert.
Weitere Experimente sind für die Validierung dieser deregulierten Gene nötig, um ihre biologische Funktion in den HL-Linien, den HRS-Zellen und in den PTCL-Lymphomen zu identifizieren und zu bestätigen.

3.4 Vergleich der Normalisierungsmethoden mit simulierter Expressionsdaten

Simulierte Expressionsdaten sind nötig, um Methoden miteinander vergleichen zu können. Differentielle *Probesets* können so definiert und der positionsabhängige Effekt der *Probes* kann so modelliert werden. Die Simulation lässt sich dabei in drei Schritte aufteilen:

1. Expression simulieren
2. Position der *Probe* im *Probeset* bestimmen
3. Differentielle Expression einbauen

3.4.1 Wahl der kontinuierlichen Verteilungsfunktion

Um zu entscheiden, welche Verteilungsfunktion für die Simulation der Expressionsdaten in Frage kommt, wird im ersten Schritt die Verteilung der gemessen rohen Expressionsdaten des Testdatensatzes betrachtet. Zunächst beschränkt auf die PM-Werte.
Abbildung 3.5 zeigt die Verteilung der Expressionsdaten des Testdatensatzes. Das Histogramm ist unsymmetrisch, es fällt auf der linken Seite steiler ab als auf der rechten. Aus den kontinuierlichen Verteilungen scheint die Normalverteilung ungeeignet, weil diese symmetrisch ist. Die Exponentialverteilung kommt auch nicht in Betracht. Daher werden vier Verteilungen (Weibull, Log-Normal, Gamma und

Empirische) untersucht, die ein ähnliches Histogramm erzeugen können wie die gemessenen Expressionsdaten.

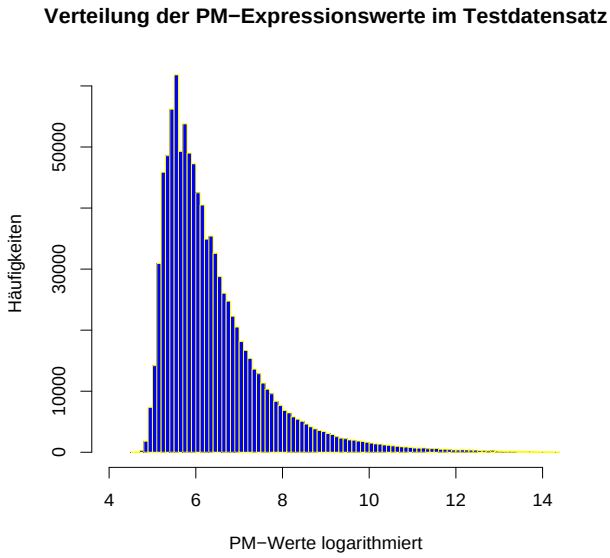


Abbildung 3.5: Histogramm der logarithmierten realen Expressionsdaten

Zunächst ist zu entscheiden, welche Verteilungsfunktion für die Simulation der Daten geeignet ist. Hierzu werden zunächst in Abbildung 3.6 zu allen Verteilungsfunktionen die Histogramme der simulierten Expressionswerte betrachtet und die simulierten Expressionswerte mit den gemessenen Espressionswerten mittels QQ-Plot verglichen. Die Verteilungsparameter der simulierten Expressionswerte werden mit der Funktion „fitdistr“ aus dem R-Paket „MASS“ mit Maximum-Likelihood an eine vorgegebene Verteilungfunktion angepasst. Hierzu kann die komplette simulierte Expressionsmatrix verwendet werden, oder nur eine Teilmatrix davon. Dieses hatte keinen wesentlichen Einfluss auf die resultierenden Verteilungsparameter. Hier die Verteilungsparameter der simulierten Expressionswerte:

1. Weibullverteilung: Shape = 4.59, Scale = 6.93

2. Log-Normalverteilung: Meanlog = 1.84, SDlog = 0.17

3. Gammaverteilung: Shape = 6.8

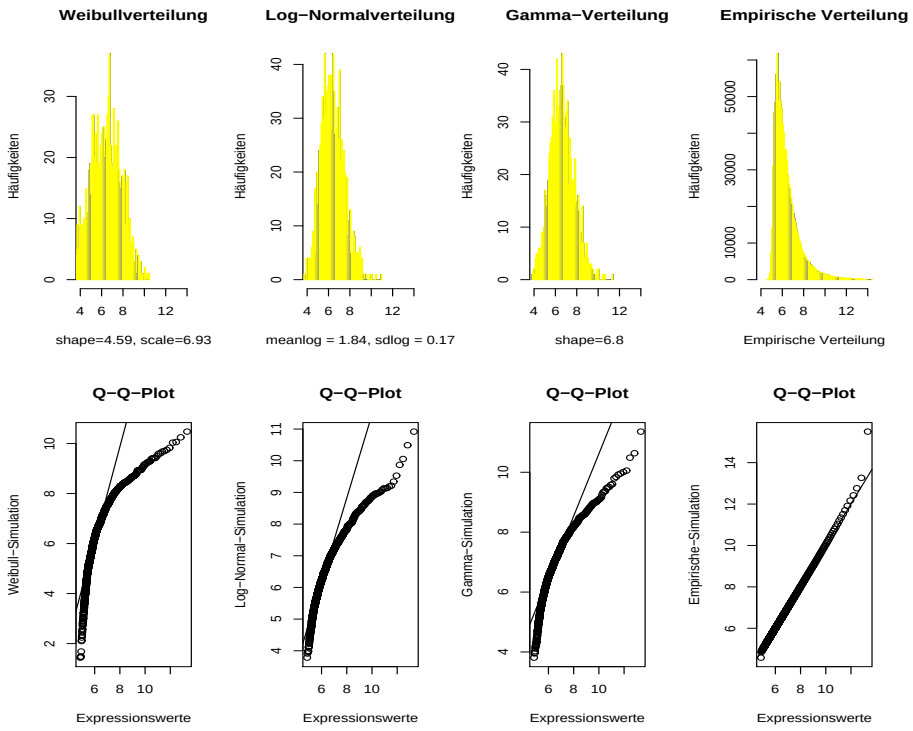


Abbildung 3.6: Histogramme zu den simulierten Expressionswerten und die dazugehörigen Q-Q-Plots

Abbildung 3.6 zeigt, dass die parametrischen Verteilungen nur zu einem kleinen Teil zu den realen Expressionswerten passen. Vor allem an den Enden der Verteilungen bilden Weibull-, Log-Normal- und Gammaverteilung die biologischen Expressionswerte nur sehr grob ab.
Die vielen Freiheitsgrade der empirischen Verteilung ermöglichen einen gradlinigen Q-Q-Plot über den gesamten Wertebereich. Deswegen wird zum Simulieren der Expressionsdaten die empirische Verteilung verwendet.
Zum Vergleich der Matrix von Zufallszahlen der simulierten Daten mit den gemessenen Daten werden probeweise Lage- und Streumaße betrachtet. Die Verteilung der probenweisen Mittelwerte und Varianzen werden dann in Histogrammen und Q-Q-Plots gegenübergestellt. Abbildung 3.7 zeigt, dass die erzeugte Zufallszahlenmatrix — zu probenweisen Mittelwerten und Varianzen — noch keine reale Struktur aufweist.

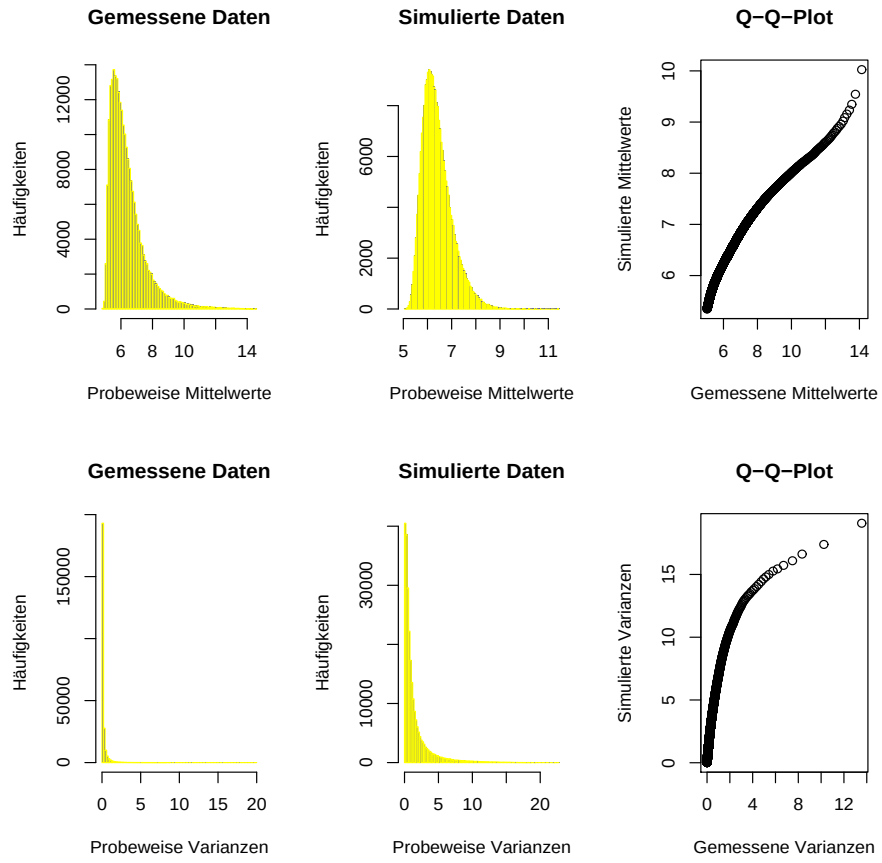


Abbildung 3.7: Histogramme zu den simulierten und gemessenen Mittelwerten und Varianzen und den dazugehörigen Q-Q-Plots

Diesen Effekt kann durch folgende Schritte korrigiert werden:

1. Jede Zeile wird standardisiert mit ($\mu = 0$, $s = 1$).
2. Aus den beobachteten Verteilungen von Varianzen wird eine zufällig gezogen und die Zeile mit deren Wurzel davon multipliziert.
3. Aus der beobachteten Verteilung der Mittelwerte ein neuer Mittelwert wird gezogen und zur Zeile hinzugefügt.

Abbildung 3.8 zeigt anhand der Q-Q-Plots der Expressionswerte, Mittelwerte und Varianzen der gemessenen und simulierten Daten, dass die Verteilung der zeilenweisen Mittelwerte und Varianzen der simulierten Matrix mit den Beobachteten übereinstimmt.

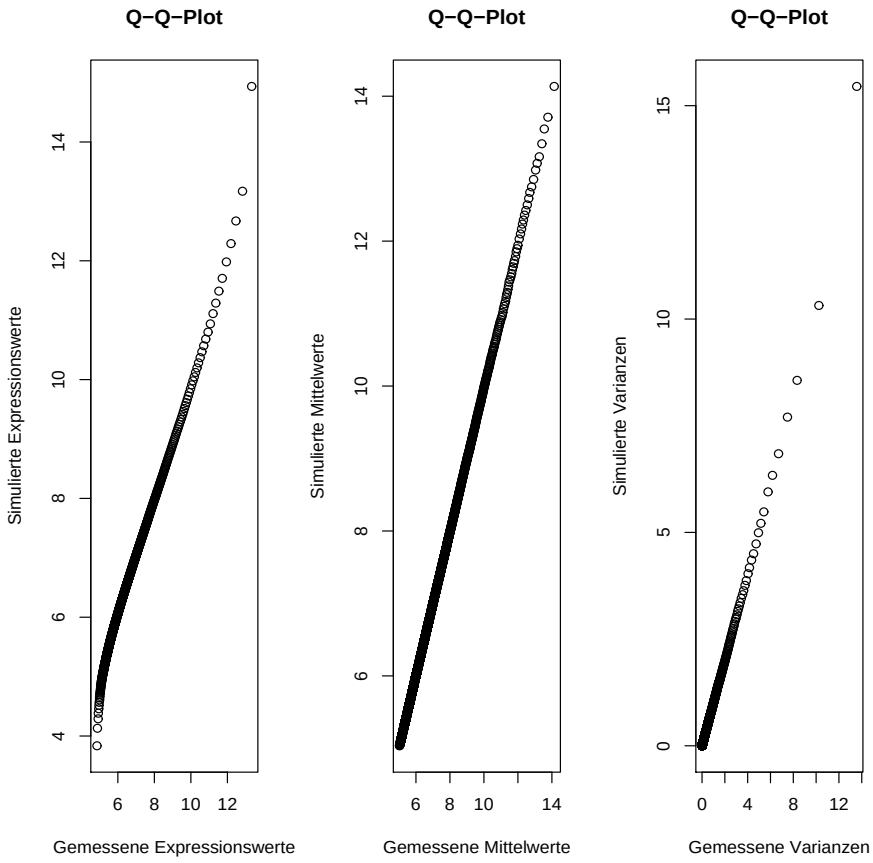


Abbildung 3.8: Die Q-Q-Plots der simulierten und gemessenen Expressionswerte, Mittelwerte und Varianzen nach der Anpassung der empirischen Verteilung

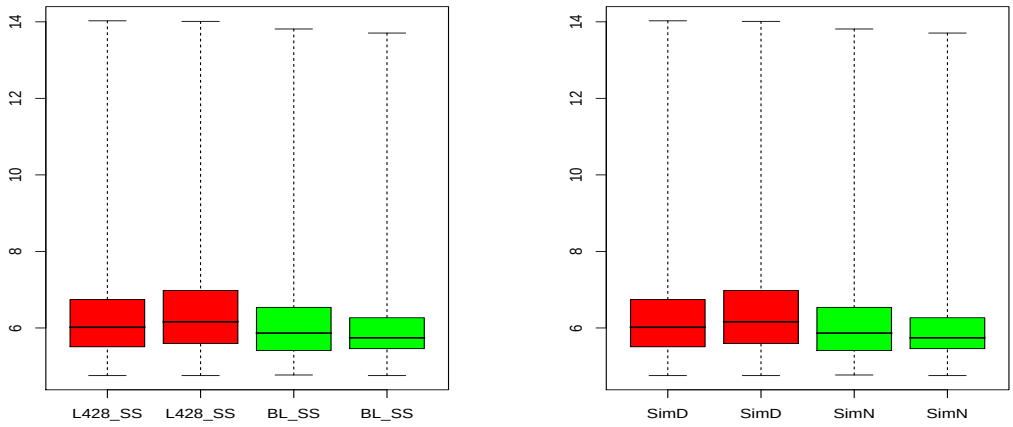
Simulation

Wie in Kapitel 3.4.1 beschrieben werden die Expressionsdaten mit der empirischen Verteilungsfunktion simuliert. Dabei können folgende Parameter übergeben werden: Definition und Anzahl der Spalten für die differentielle Expression und die Anzahl der differentiell exprimierten *Probesets*.

Deskriptive Betrachtung der simulierten Expressionsdaten

Die simulierten Daten erfüllen zwei zentrale Verteilungseigenschaften: Die Verteilungseigenschaften sollen den Expressionsdaten ähnlich sein, wie bei den realen Daten. Dieser Teil ist in Kapitel 3.4.1 dokumentiert. Bei den Boxplots, in denen die realen Daten und die simulierten Daten gegenübergestellt sind, erkennt man eine

ähnliche Verteilung der Expressionswerte. Allerdings zeigen die simulierten Expressionsdaten geringere Schwankungen als die realen Expressionsdaten (siehe Abb. 3.9)
.



(a) Reale Expressionsdaten (b) Simulierte Expressionsdaten

Abbildung 3.9: Vergleich der Boxplots von realen und simulierten Expressionsdaten

Beim Vergleich mit den Boxplots (siehe Abb. 3.10) der Expressionsdaten die mit einer Invitrotranskription erhoben wurden liegt der Median bei 7 und nicht bei 6, wie bei den Expressionsdaten mit doppelter Invitrotranskription.

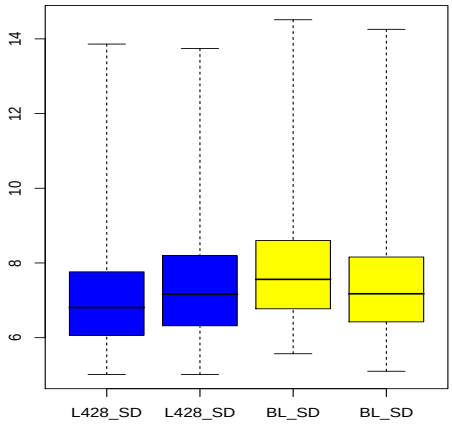
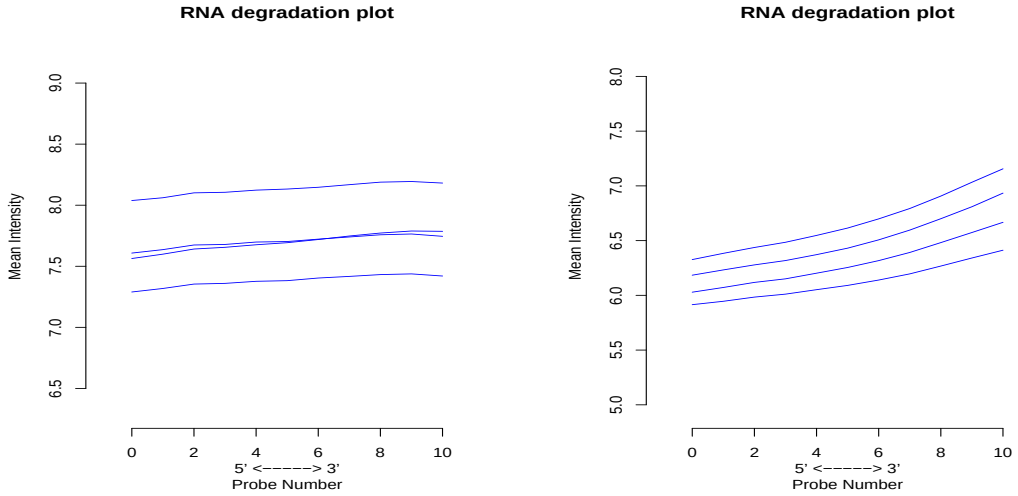


Abbildung 3.10: Boxplot der Expressionsdaten mit einfacher IVT

Bei der Prozessierung der mRNA mit doppelter Invitrotranskription entstehen kürzere Fragmente und damit eine schlechtere Abdeckung des *Probesets*-Designs von Affymetrix. Deutlich wird dies beim Vergleich von einfacher und doppelter Invitrotranskription — den RNA-Degradierungsplots —. Bei doppelter Invitrotranskription ist ein deutlicher Abfall der Expressionsstärke in Richtung 5'-Ende erkennbar und damit eine Expressionsabhängigkeit der Position im Transkript. Die Daten aus einfacher Invitrotranskription zeigen diese Abhängigkeit nicht. Abbildung 3.11 zeigt mit dem Bioconductor Paket „AffyRNAdeg“ von Leslie Cope die mittlere Expression über die Position im Transkript über alle *Probesets*.



(a) Einfache Invitrotranskription

(b) Doppelte Invitrotranskription

Abbildung 3.11: Vergleich der RNA-Degradation bei einfacher und doppelter Invitrotranskription. Die einzelnen Linien repräsentieren jeweils einen *Microarray*-Chip.

Im Vergleich hierzu zeigt die RNA-Degradationsgrafik der simulierten Daten (siehe Abb. 3.12), dass bei doppelter IVT die Expression nicht über alle Positionen gleich ist. Die Abhängigkeit von der Position im Transkript wird hier simuliert (siehe Kapitel 2.6.2), allerdings ausgeprägter als bei den realen Daten.

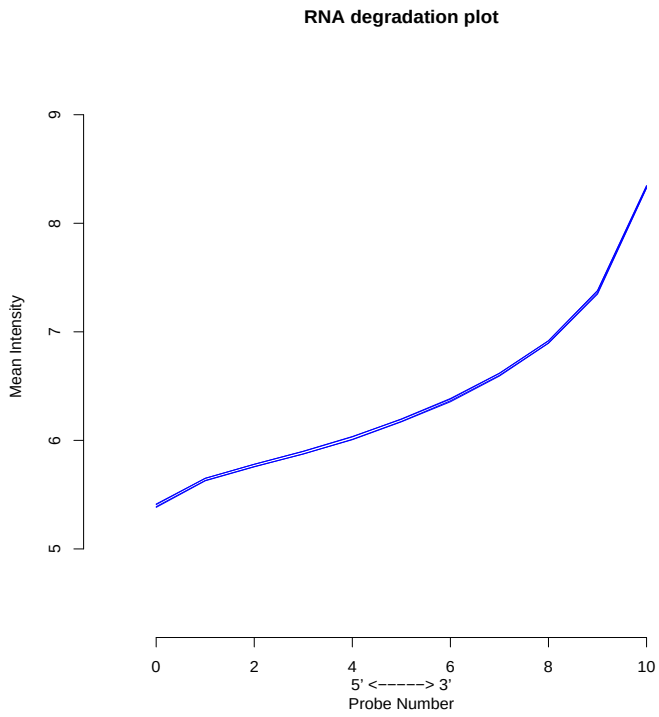


Abbildung 3.12: Die RNA Degradation bei simulierten Daten

Zusammenfassend ist zu erkennen, dass die simulierten Expressionsdaten die wesentlichen Verteilungseigenschaften der realen Expressionsdaten widerspiegeln.

Betrachtung der differentiell-exprimierten Gene

Eine Möglichkeit, die Sensitivität eines Verfahrens zu prüfen, ist das Simulieren der Gene, die eine differentielle Expression zeigen, und zu ermitteln, wie viele dieser Gene das Verfahren findet. Für unseren Versuch wurden 500 *Probesets* (wie in Kapitel 2.6 beschrieben) als differentiell exprimiert simuliert. Frage war, wie viele Gene man in Abhängigkeit von der jeweiligen Präprozessierungsmethode findet. Betrachtet werden hierfür zunächst die 500 *Probesets*, die als differentiell-exprimiert simuliert wurden. Unberücksichtigt blieben eventuelle Vorfilterungsmechanismen, die beim betrachten des gesamten Datensatz berücksichtigt würden und damit einen Einfluss auf das Wiederfinden der simulierten *Probesets* haben könnten. Für die Signifikanz eines *Probesets* wurde der p-Wert, nicht die FDR herangezogen.
Die Werte für die differentielle Expression stammten aus den Bereichen der 10 Quantile. Zu klären war, welchen Einfluss die Expressionsstärke auf das Wiederfinden der

differentiellen *Probesets* hat. Die Wahl des Quantil-Bereiches (Felder) eines *Probesets* war zufällig, der Anteil der Quantilsbereiche war in den verschiedenen Simulationen unterschiedlich. Um die Ergebnisse der 10 durchgeführten Simulationen vergleichen zu können, war jeweils der prozentuale Anteil der richtig Positiven aus dem jeweiligen Quantils-Bereich in Abhängigkeit der verwendeten Normalisierung zu berechnen. Abbildung 3.13 zeigt diesen Zusammenhang grafisch.

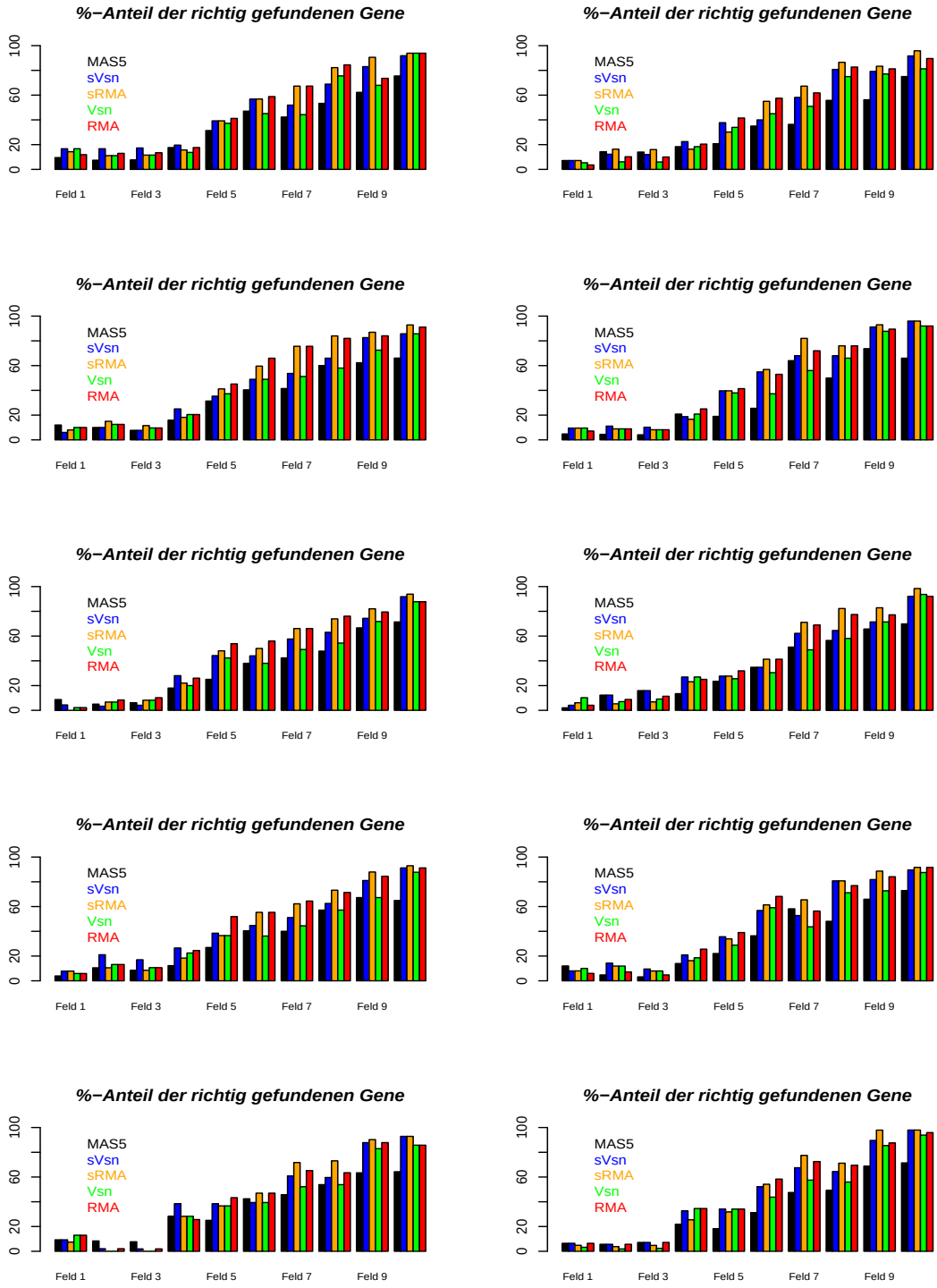


Abbildung 3.13: Grafische Darstellung der simulierten differentiell-exprimierten *Probesets*, die bei allen 10 Simulationen wiedergefunden wurden. Jedes Feld entspricht einem anderen Quantilbereich.

Unabhängig von der durchgeführten Methode in den Feldern im unterem Bereich (<4) wurden, auffällig wenig (<10 %) der simuliert-differentiell-exprimierten *Probesets* gefunden. Die Expressionswerte in diesem Bereich sind offenbar zu klein, um wirklich als differentiell detektiert zu werden.
Die Methode MAS5 findet — von einigen Ausnahmen abgesehen — die wenigsten differentiellen *Probesets*. sRMA, sVSN finden die meisten differentiellen *Probesets*, gefolgt von RMA und VSN. sRMA findet bei großen Expressionsunterschieden die meisten Gene wieder, sVSN bei kleinen Expressionsunterschieden. Die Expressiosstärke des *Probesets* korreliert mit dem prozentualen Anteil der wiedergefunden differentiellen *Probesets*, je größer die Expression ist, desto mehr differentielle *Probesets* werden gefunden.
Zu klären ist auch, wie die Varianz bei mehreren Simulationen varriert. Abbildung 3.14 stellt hierzu die Boxplots der Methoden zu den einzelnen Simulationen dar. Die zum Teil große Ausdehnung der Box resultiert daraus, dass alle Felder zu einer Simulation zusammengefasst wurden. Abbildung 3.13 zeigt die Schwankungen zwischen den Feldern. Die Median-Unterschiede zwischen den einzelnen Simulationen sind zum Teil sehr groß.
Insgesamt findet RMA die meisten simulierten differentiell-exprimierten *Probesets*. sRMA und sVSN folgen. Abhängig von der Simulation zeigt mal sRMA, mal sVSN die besseren Ergebnisse. MAS5 findet mit Abstand die wenigsten simuliert-differentiell-exprimierten *Probesets* wieder. Dies war unabhängig von der Simulation.

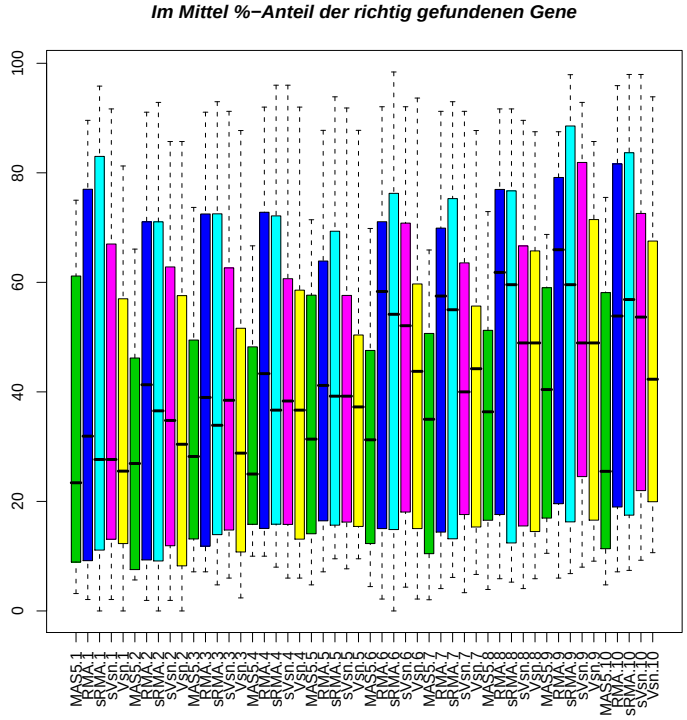


Abbildung 3.14: Boxplots zum Wiederfinden der simulierten differentiell-exprimierten *Probesets* für die 10 Simulationen, d. h. die Expressionswerte sind über alle Expressionsstärken für die einzelnen Simulationen dargestellt.

Sensitivität und Spezifität der simulierten Expressionsdaten

Wie in Kapitel 2.11 beschrieben gilt es, Sensitivität und Spezifität einer Methode zu bewerten. Nur die *Probesets* zu betrachten, die als differentiell simuliert wurden reicht nicht aus. Zu prüfen ist, wie viele *Probesets* fälschlicherweise als differentiell gefunden wurden (falsch Positive).
Dazu wurde jeder der 10 simulierten Expressionsdatensätze separat für jede Methode analysiert. Für alle Analysen wurden dieselben Filterkriterien angewendet. Eine Vorfiltererung wurde nicht durchgeführt, um in diesem Schritt nicht einige der *Probesets*, die einen kleinen Expressionswert haben, herauszufiltern. Zu berücksichtigen ist dies allerdings bei der Bewertung der FDR, da hier eine größere Menge an *Probesets* simultan getestet wurden (nämlich die Anzahl des gesamten Microarrays) als im normalen Experiment.
Wie in Kapitel 2.11 beschrieben, sind die richtig Positiven (RP) und die richtig

Negativen (RN) interessant. RP-*Probesets*, die ein $FDR \leq 0.8$ haben wurden als differentiell-simuliertes *Probeset* erkannt. RN-*Probesets*, die ein $FDR \geq 0.8$ haben wurden als nicht differentiell simuliertes *Probeset* erkannt. Eine Methode ist ideal, wenn sie eine gute Sensitivität und Spezifität, bei maximaler Anzahl RPs findet, aber gleichzeitig die wenigsten falsch Positiven und die meisten RN zeigt.
Abbildung 3.15 zeigt für alle Simulationen und Methoden die jeweilige Sensitivität gegen die Spezifität. MAS5 zeigte die schlechteste Sensitivität und Spezifität, es fand die wenigsten RPs und RNs. RMA und VSN fanden zwar die meisten RNs und sind damit spezifischer, sie sind aber nicht sensitiver als MAS5. Außerdem zeigten RMA und insbesondere VSN eine größere Streuung über die Simulationen als MAS5. sRMA und sVSN zeigten die beste Sensitivität allerdings auch eine schlechtere Spezifität als VSN und RMA. Die Streuung war über alle 10 Simulationen bei sVSN und sRMA am größten.

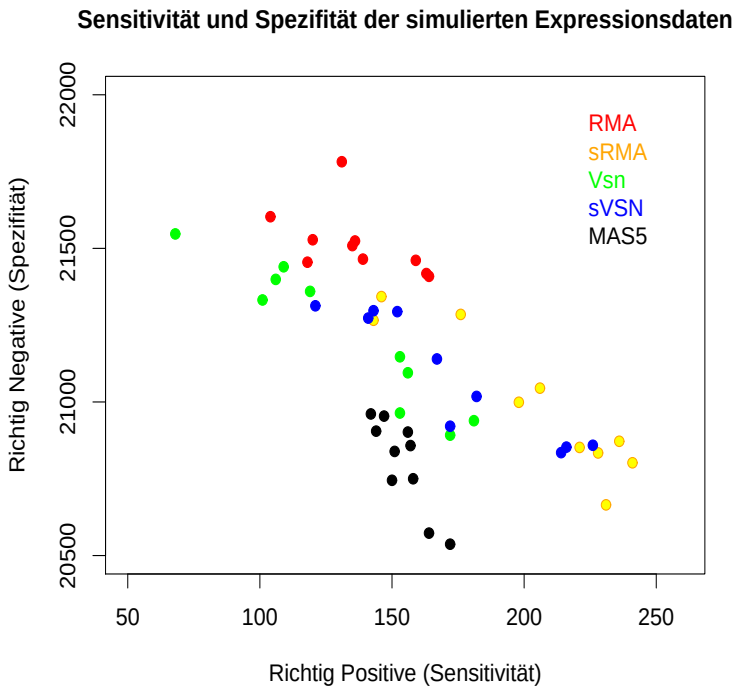


Abbildung 3.15: Sensitivität und Spezifität der einzelnen Methoden für die 10 Simulationen

Unabhängig vom gewählten FDR *Cut-Off* wurden ROC-Kurven für die verschiedenen Normalisierungsmethoden erstellt. Dabei wurde das FDR-Intervall $[0, ..., 1]$ in

1000 Bereiche unterteilt und dazu Sensitivität und Spezifität, wie in Kapitel 2.11 beschrieben, für alle 10 Simulationen berechnet. Abbildung 3.16 zeigt die über alle 10 Simulationen gemittelte ROC-Kurve.

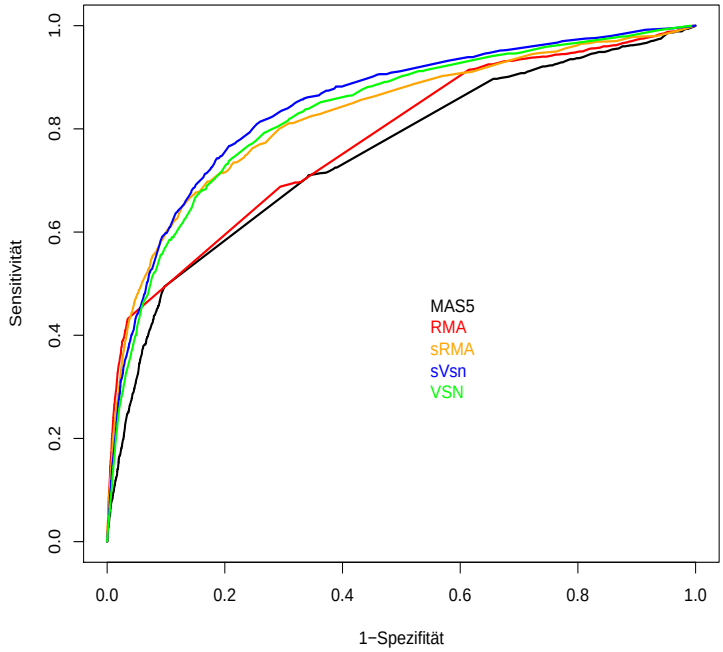


Abbildung 3.16: Sensitivität und Spezifität, dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit einer empirischen Verteilung

Betrachtet man die ROC-Kurve 3.16, zeigt auch hier die Methode MAS5 die schlechteste Sensitivität und Spezifität, gefolgt von der Methode RMA. VSN, sRMA und sVSN sind qualitativ nahe zusammen. sVSN zeigte eine minimal-bessere Sensitivität und Spezifität, als die Methoden VSN und sRMA.

3.4.2 Simulation der Daten auf Basis einer angepassten Gammaverteilung

Die realen Expressionsdaten folgten keiner Gammaverteilung, allerdings konnten die Expressionsdaten angepasst werden. Dafür wurde das Minimum von den realen Expressionswerten abgezogen (PM_{subset}) und mit diesen Expressionswerten eine Gammaverteilung angepasst. Dazu wurden zunächst zwei Parameter (Skalierungs- und

Formenparameter) benötigt, um Expressionswerte mit Hilfe einer Gammaverteilung zu erhalten. Diese Parameter wurden direkt aus den Expressionsdaten berechnet. Für den Skalierungsfaktor (= 0,8125) wird die Varianz der PM_{subset} durch den Mittelwert von PM_{subset} dividiert. Der Formenparameter (= 2.259) ist das Produkt aus dem Mittelwert von PM_{subset} geteilt durch den zuvor berechneten Skalierungsfaktor.
Mit der Funktion "*rgamma*" lassen sich mit dem zuvor bestimmten Skalierungs- und Formenparameter die Expressiondaten gammaverteilt simulieren, getrennt für die PMs und MMs.
Die simulierte Expressionsmatrix wurde äquivalent zu Kapitel 3.4.1 erstellt. Die differentiell-exprimierten Gene wurden definiert und die *Probes* eines *Probesets* sortiert.

Deskriptive Betrachtung der simulierten Expressionsdaten

Verglichen mit dem Histogramm der simulierten Expressionswerte (Abb. 3.17) zeigten die realen Expressionswerte (Abb. 3.5) eine ähnliche Form. Der anschließende QQ-Plot (Abb.3.17) zeigt allerdings eine schlechte Übereinstimmung im oberen Expressionsbereich (> 4).

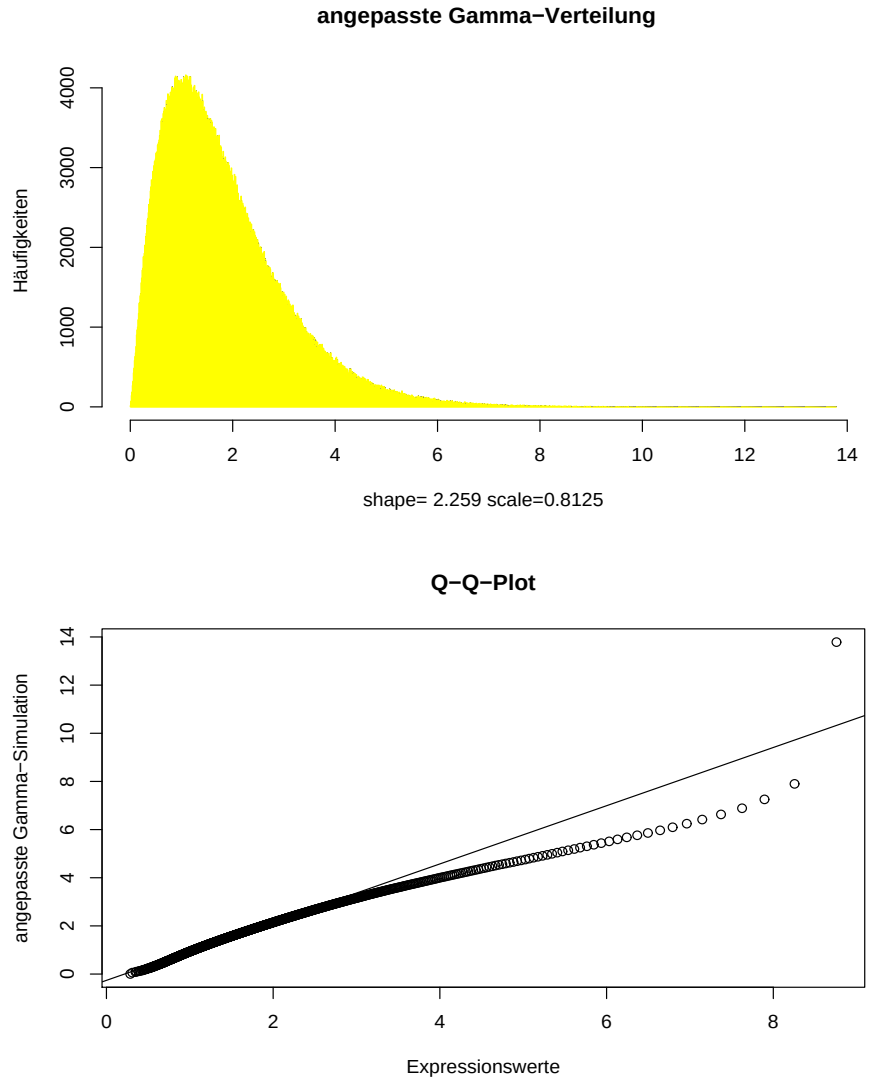


Abbildung 3.17: Histogramm der simulierten PM-Expressionswerte resultierend aus einer angepassten Gammaverteilung und der QQ-Plot im Vergleich zu den realen PM-Expressionsdaten

Die Boxplots in Abbildung 3.18 zeigen, dass die simulierten Expressionsdaten einen Median < 2 aufweisen. Der Median bei den realen Expressionsdaten ist bei 6.

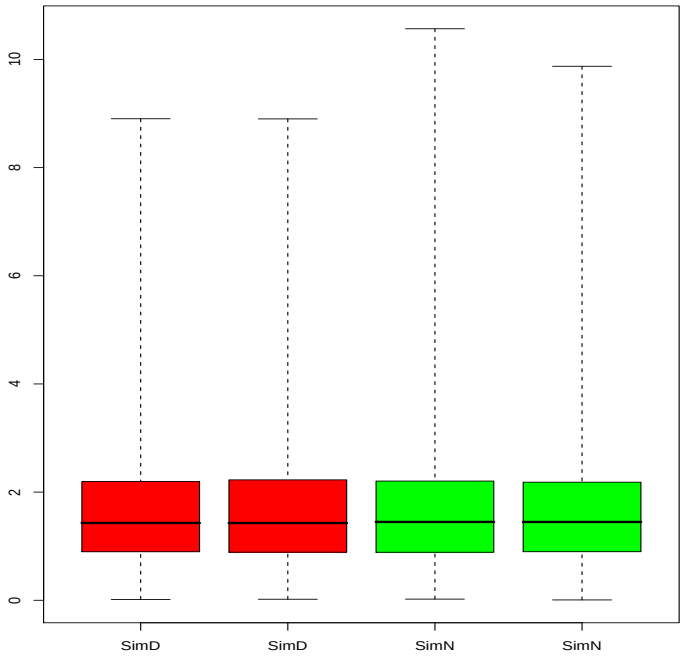


Abbildung 3.18: Boxplot der Gamma-simulierten Expressionsdaten

Die RNA-Degradationsgrafik (Abb. 3.19) zeigt, aufgrund derselben Sortierung der einzelnen *Probes* im *Probeset*, ein ähnliches Bild wie schon zuvor bei der Verwendung der empirischen Verteilung. Auch hier zeigt sich eine deutliche Abhängigkeit der *Probe*-Expression von der Position im Transkript.

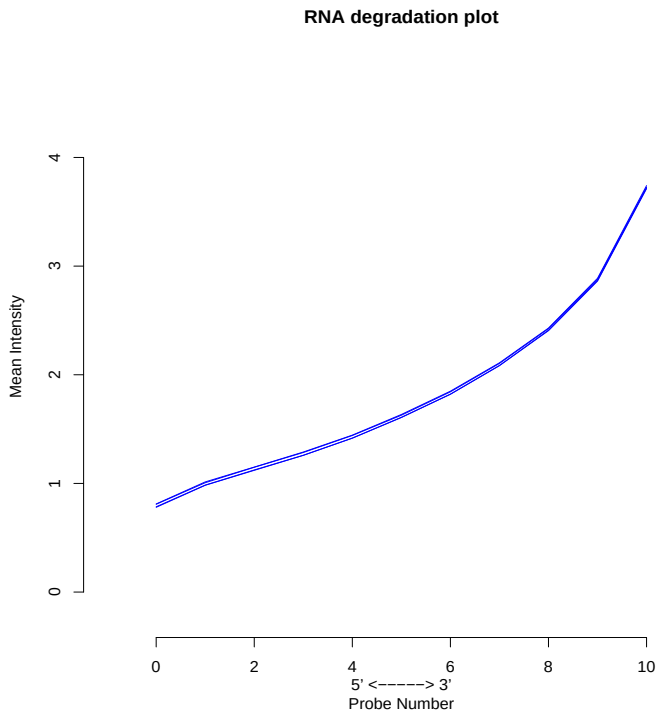


Abbildung 3.19: RNA-Degradationsplot der Gamma simulierten Expressionsdaten

Die angepasste Gamma-Verteilung zeigt im oberen Expressionsbereich keine gute Übereinstimmung mit den realen Daten und hat einen Median von < 2. Trotzdem wurde der Einfluß verschiedener Präprozessierungsmethoden auf die differentielle Expression untersucht, wenn derart verteilte Expressionsdaten vorlagen.

Betrachtung der differentiell-exprimierten Gene

Der prozentuale Anteil der richtig gefundenen Gene (Abb. 3.20) zeigt deutlich größere Schwankungen als bei empirisch simulierten Expressionsdaten (Abb. 3.13). Bei der 4. Simulation fielen aus nicht bekannten Gründen, mit Ausnahme von VSN, alle anderen Präprozessierungsmethoden aus. Bei der 5. Simulation fiel sRMA aus. Besonders MAS5 stach bei den Simulationen heraus. Die MAS5-Methode fand die meisten richtigen Gene. Insgesamt wurden allerdings weniger richtig Positive gefunden als bei empirisch simulierten Expressionsdaten. Sogar im Feld 10, mit den höchsten Expressionswerten, konnten mit keiner Methode mehr als 90 % richtig Positive gefunden werden.

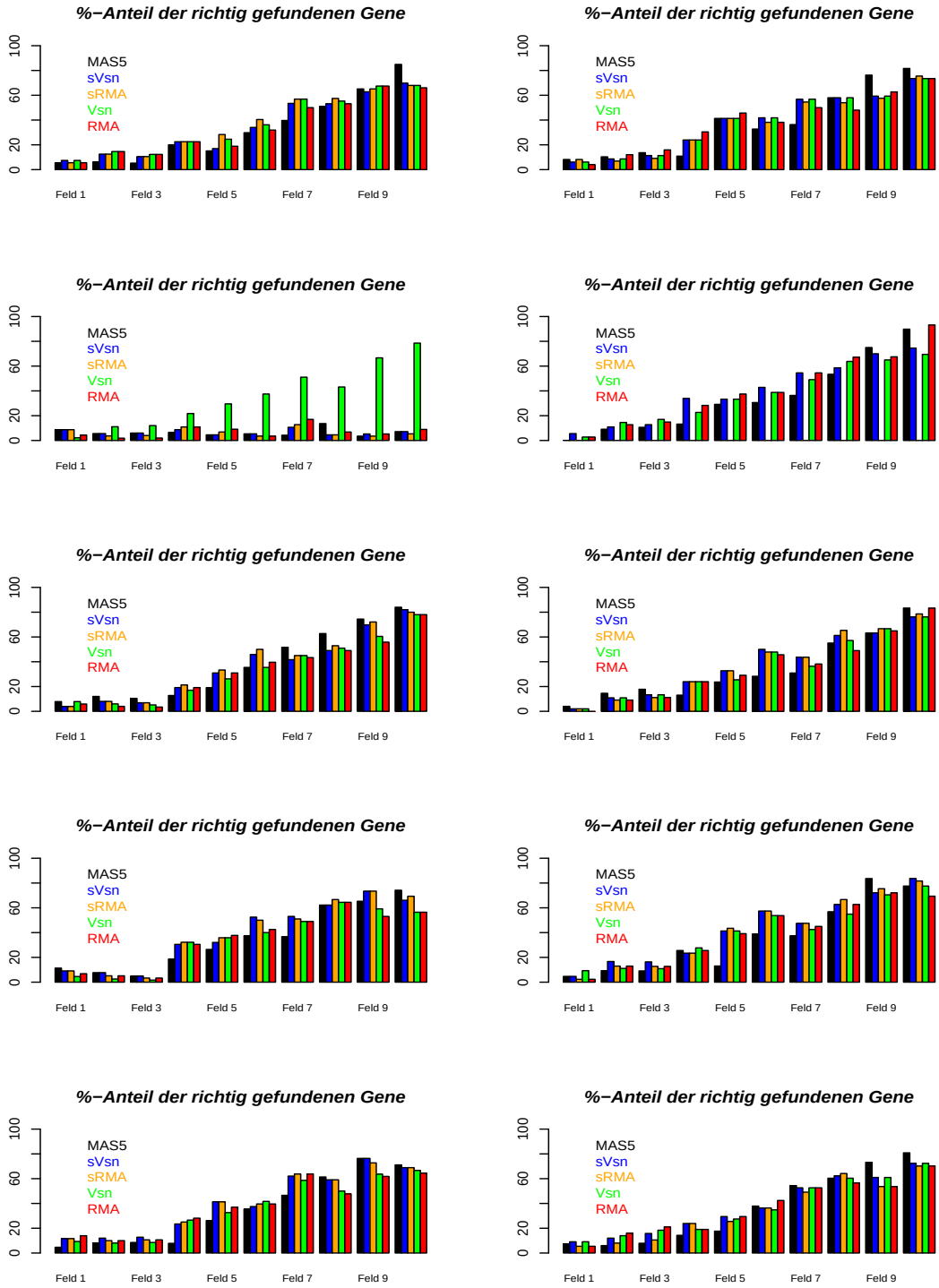


Abbildung 3.20: Prozentualer Anteil der richtig Positiven differentiell-exprimierten *Probesets* bei gamma-verteilten Daten. Jedes Feld entspricht einem anderen Quantilbereich.

In der Boxplot-Darstellung (Abb. 3.21) der Simulationen und Methoden erkennt man, dass die 80 % Grenze nur wenige Male überschritten wurde. Außerdem zeigen die Boxplots weitaus größere Schwankungen als bei den empirisch simulierten Expressionsdaten.

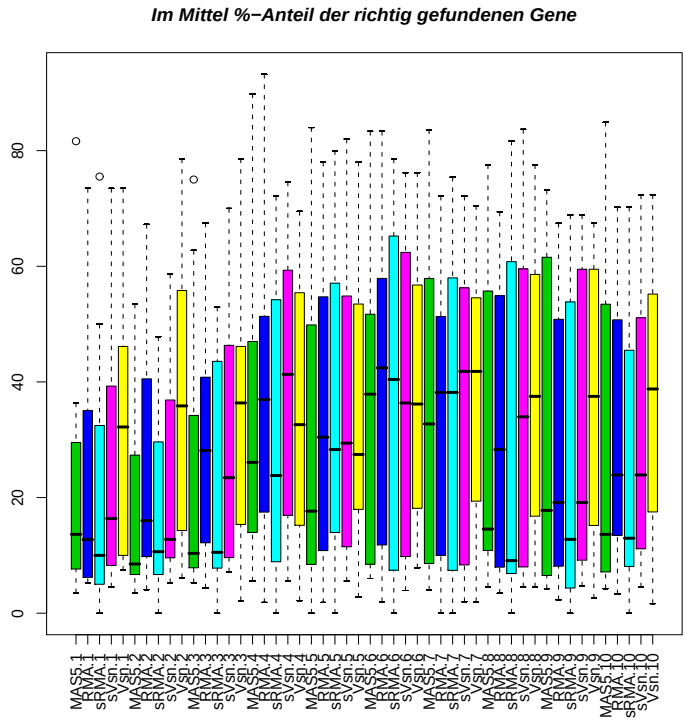


Abbildung 3.21: Boxplot für die Präprozessierungsmethoden für alle 10 Simulationen

Sentivitität und Spezifität der simulierten Expressionsdaten

Sensitivität und Spezifität der simulierten Expressionsdaten mit einer angepassten Gammaverteilung wichen von der Sensitivität und der Spezifität der empirisch simulierten Expressionsdaten ab. Die Methode MAS5 klassifizierte die meisten richtig Positiven. Allerdings fand MAS5 auch die geringste Anzahl an richtig Negativen. Die MAS5-Methode bildete bei der empirischen Simulation eine von den anderen Methoden isolierte Gruppe, analog zur Abbildung 3.15. Die anderen Methoden gruppierten sich, wobei die RMA-Methode die wenigsten richtig Positiven, aber dafür die meisten richtig Negativen fand. Der Ausfall von Simulation Nr.4 ist deutlich zu erkennen, wobei sich der Ausfall nur auf die richtig Positiven beschränkte.

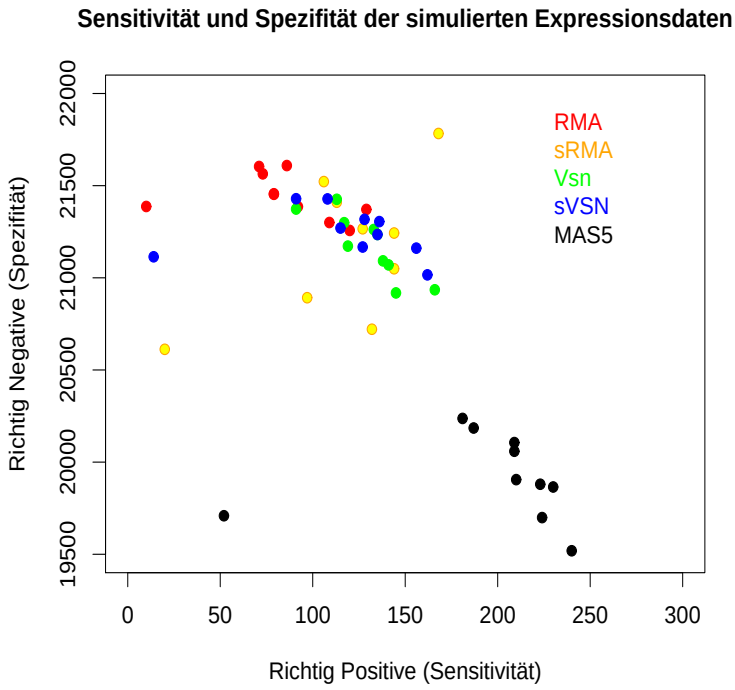


Abbildung 3.22: Sensitivität und Spezifität der Präprozessierungsmethoden für alle 10 Simulationen

Unabhängig vom gewählten FDR *Cut-Off* wurden ROC-Kurven für die verschiedenen Normalisierungsmethoden erstellt. Dabei wurde das FDR-Intervall $[0, ..., 1]$ in 1000 Bereiche unterteilt und dazu Sensitivität und Spezifität, wie in Kapitel 2.11 beschrieben, für alle 10 Simulationen berechnet. Abbildung 3.23 zeigt die für alle 10 Simulationen gemittelte ROC-Kurve.

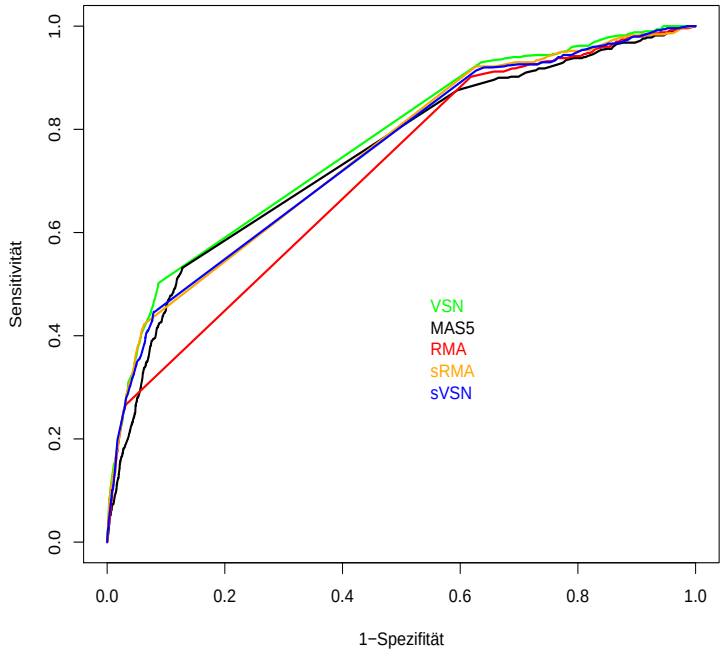


Abbildung 3.23: Sensitivität und Spezifität, dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit einer angepassten Gammaverteilung

Betrachtet man die ROC-Kurve 3.23, zeigt die Methode RMA die schlechteste Sensitivität und Spezifität, gefolgt von den Methoden sRMA, sVSN, Mas5 und VSN. Außer MAS5 lagen alle Methoden sehr dicht zusammen, wobei VSN eine minimal bessere Sensitivität und Spezifität zeigte, als die anderen Methoden.

3.4.3 Gemischt-lineares Modell zum Simulieren von Expressionsdaten

Bei der Simulation mit einer empirischen Verteilung oder mit der angepassten Gammaverteilung wurde eine strikte Sortierung der einzelnen *Probes* eines *Probesets* in Abhängigkeit der Position im Transkript vorgenommen.
Allerdings ist eine solche Sortierung nur ein Modell für die Annahme, dass mit einer größer werdenden Position (weiter weg vom 3'-Ende) die Expressionintensität abnimmt. Das trifft sicherlich nur für einen Teil der *Probesets* zu. Deswegen ist der Einsatz eines linearen gemischten Modells eine Möglichkeit diesen *Probe*-Effekt mit

einzubringen, da eine Korrelation zwischen *Probe* und Position vorlag.
Für die Simulation der vorliegenden Expressionsdaten wurde das *lme*4-Paket von Bates et al. [Baayen et al., 2008], das linear gemischte Modelle zur Verfügung stellt, verwendet. Mit der Funktion *lmer* können linear gemischte Modelle angepasst werden. Dabei wird der Funktion *lmer* Folgendes übergeben:

$$lmer(PMData \sim pos + chip + (pos|namen), data = daten)$$

Die Variable $PMData$ entspricht dabei allen PM-Expressionswerten. Hinter der $\sim$ ist zunächst der feste Effekt definiert. In unserem Fall setzte er sich aus *pos*, der Position im Transkript, und *chip*, den verschiedenen *Microarrayexperimenten*, zusammen. Es folgte der Random-Effekt, der aus einem Ausdruck eines linearen Modells und einem Gruppenfaktor besteht, der durch | getrennt wird. Dabei entspricht *pos* dem linearen Modell und *namen* (die einzelnen *Probeset-IDs*) dem Gruppenfaktor. Der Random-Effekt ist Ausdruck für ein lineares Modell bedingt auf den Gruppenfaktor. Für die MM-Expressionswerte wurde ein analoges Modell angepasst.
Der *Microarray-Chip* wurde im Gruppenfaktor nicht berücksichtigt, da die Variable *chip* eine zu geringe Variabilität zeigte und es zu Konvergenzproblemen kam.
Mit dem resultierendem angepassten Modell konnten Expressionsdaten simuliert werden. Das wurde — wie bei der empirischen und der angepassten Gammaverteilung — getrennt für die PM- und MM-Expressionswerte durchgeführt. Die Simulation wurde 10 Mal wiederholt und dann weiter mit den fünf verschiedenen Präprozessierungsmethoden ausgewertet.

Deskriptive Betrachtung der simulierten Expressionsdaten

Abbildung 3.24 zeigt das Histogramm für die simulierten Expressionsdaten — basierend auf einem linearem gemischten Modell. Die simulierten Expressionsdaten erscheinen normalverteilt im Gegensatz zu den realen Expressionsdaten (Abb.3.5), die nicht normalverteilt erscheinen. Das ist ein Grund, warum auch der QQ-Plot deutliche Unterschiede zwischen den simulierten und realen Expressionsdaten aufzeigte. Besonders an den Enden schienen die Expressionswerte bei den simulierten Daten schlechter abgedeckt zu sein.

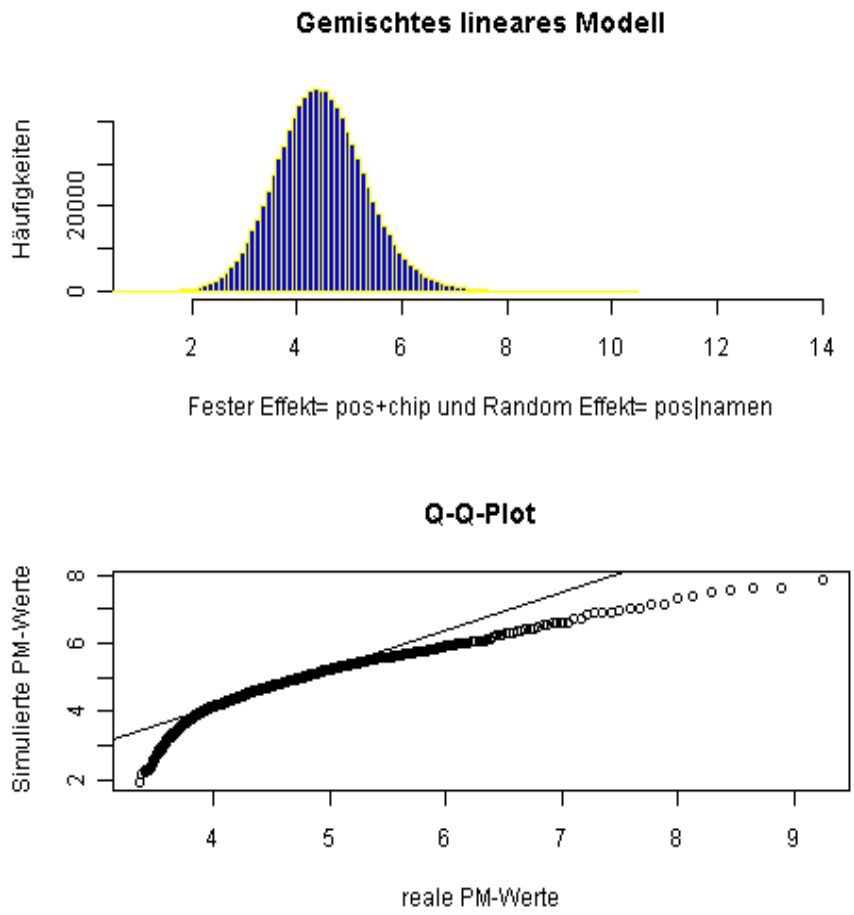


Abbildung 3.24: Histogramm der simulierten PM-Expressionswerte resultierend aus einem linearen gemischten Modell und der QQ-Plot im Vergleich zu den realen PM-Expressionsdaten

Die Verteilung der simulierten Expressionswerte scheint anders zu sein, als die der realen Expressionswerte. In Abbildung 3.25 liegt der Median der simulierten Expressionswerte bei etwas über 4. Bei den realen Expressionswerten liegt der Median bei 6 (Abb. 3.9(a)).

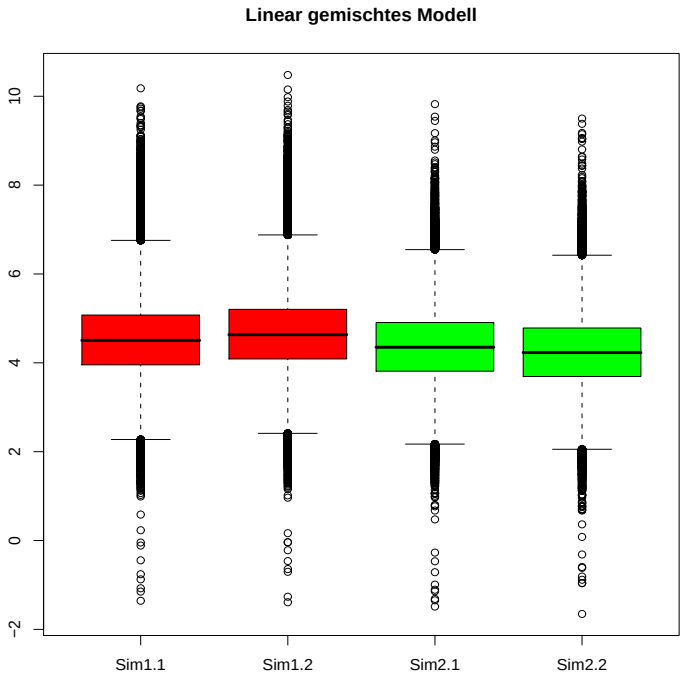


Abbildung 3.25: Boxplot der simulierten Expressionsdaten — basierend auf einem linearen gemischten Modell

Da bei der Simulation der Expressionsdaten — basierend auf einem linearen gemischten Modell — die *Probes* nicht nach ihrer Expressionstärke sortiert wurden, ähnelte die RNA-Degradation der simulierten Expressionsdaten in Abbildung 3.26 (Verlauf und Stärke der Degradierung) der der realen Expressionsdaten (Abb. 3.11(b)).

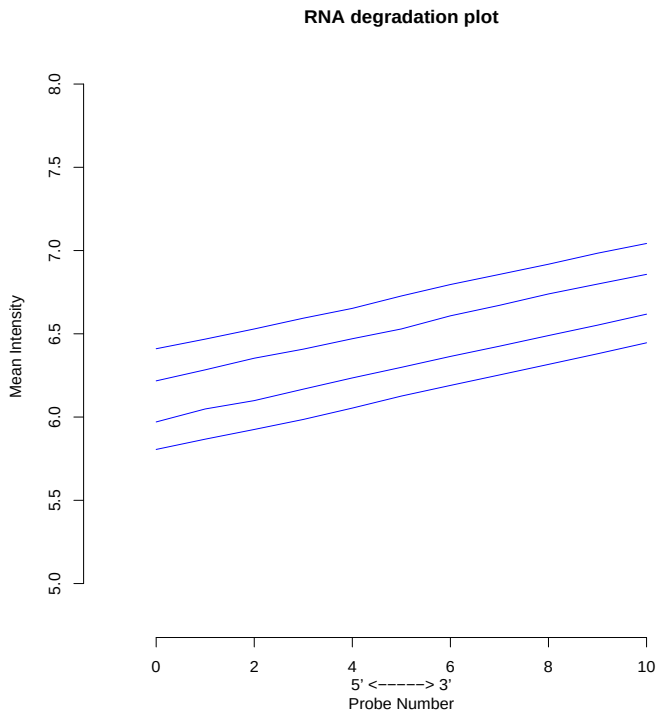


Abbildung 3.26: RNA-Degradationsplot der simulierten Expressionsdaten resultierend aus einem linear gemischten Modell

Betrachtung der differentiell exprimierten Gene

Bei der Simulation differentiell-exprimierter *Probesets* wurde ein anderes Verfahren als bei der empirischen und der angepassten Gamma-Verteilung gewählt. Nach der Simulation der Expressionsmatrix wurde für eine definierte Anzahl von *n-Probesets* ein zufälliger Faktor zwischen 2 und 10 gewählt, der auf die einzelnen *Probes* multipliziert wurde. Dabei war der Faktor für ein *Probeset* fest und konnte nicht innerhalb der *Probes* variieren.
Zunächst wurden nur die differentiell simulierten 500 *Probesets* betrachtet. Abbildung 3.27 zeigt wie viel Prozent der differentiellen *Probesets* wiedergefunden wurden. Als differentiell-exprimiert wurde ein *Probeset* klassifiziert, das einen p-Wert $\leq 0,05$ erfüllt. Ab einem Faktor von drei wurden mehr als 90 % der *Probesets* richtig klassifiziert, unabhängig davon, welche Methode man betrachtete. Es ist zu erkennen, dass die Methode MAS5 die wenigsten *Probesets* über alle Faktoren richtig klassifiziert. Bei den Faktoren 2 - 5 zeigten besonders die Methoden sVSN und sRMA eine

bessere Klassifizierung, gefolgt von VSN und RMA.

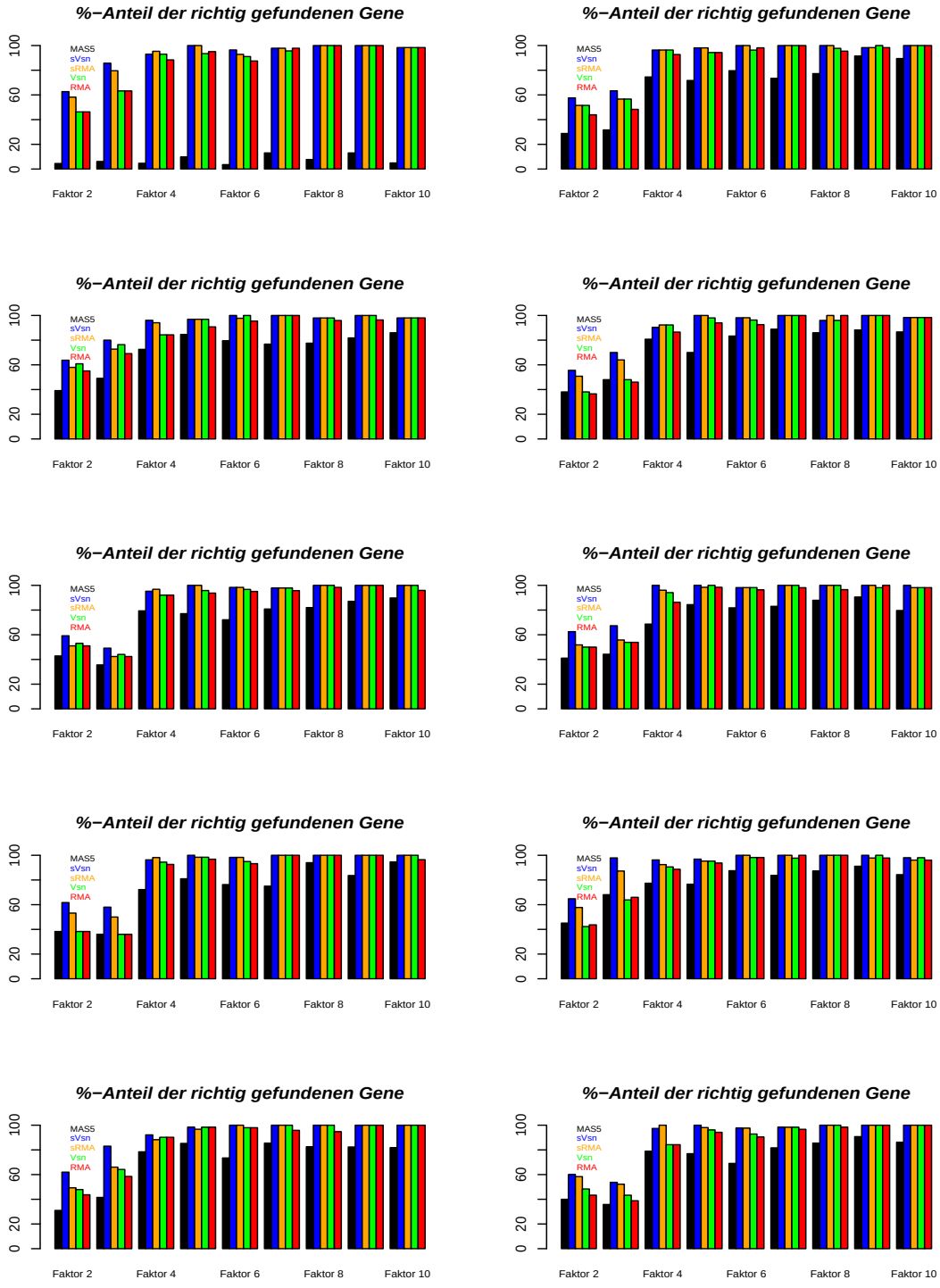


Abbildung 3.27: Prozentualer Anteil der richtig positiven differentiell-exprimierten *Probesets*. Der Faktor gibt die Stärke der differentiellen Expression an.

Dieses Ergebnis spiegelte sich auch in den Boxplots der einzelnen Simulationen und Methoden wieder (Abb. 3.28). Die größten Schwankungen und den geringsten Median wies dabei MAS5 auf. sVSN zeigte die geringste Variablität und den größten Median, gefolgt von sRMA, VSN und RMA.

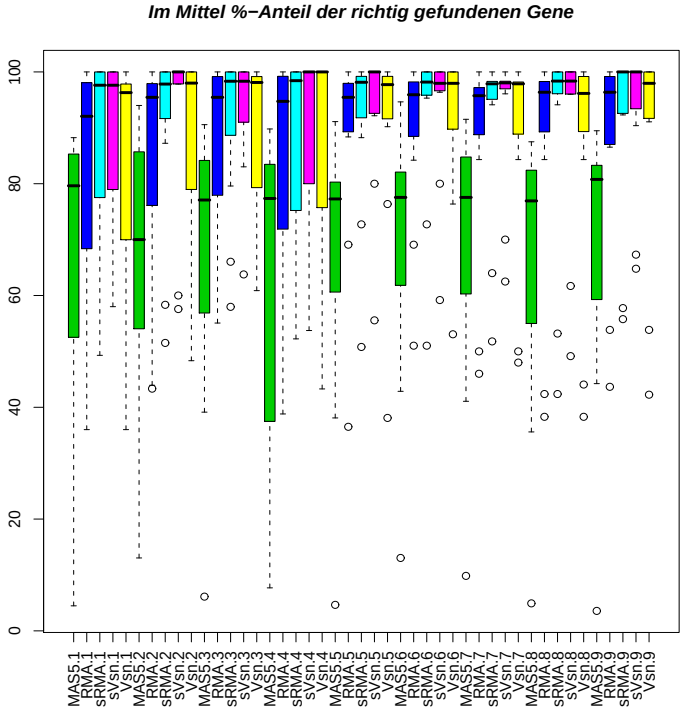


Abbildung 3.28: Boxplot der Präprozessierungsmethoden für alle 10 Simulationen

Sensitivität und Spezifität der simulierten Expressionsdaten

Im nächsten Schritt wurden nicht nur die richtig Positiven betrachtet, sondern auch die richtig Negativen, die für die Spezifität einer Methode stehen. Dabei wurden alle *Probesets*, die einen $FDR > 0,8$ haben und nicht als differentiell simuliert wurden, als richtig Negativ bewertet. Umgegekehrt wurden alle *Probesets*, die einen $FDR \leq 0,8$ haben und als differentiell simuliert wurden als richtig Positiv bewertet. Abbildung 3.29 zeigt dies für alle 10 Simulation bei allen fünf Methoden.
Die Abgrenzung der MAS5-Methode ist wieder deutlich zu erkennen. Zwar ist die MAS5 Methode ähnlich spezifisch wie die sVSN-Methode, allerdings zeigte sie die schlechteste Sensitivität. Die anderen Methoden, sRMA, VSN und RMA, zeigten eine ähnliche Sensitivität wie sVSN, aber eine schlechtere Spezifität. Zu favorisieren

ist daher die sVSN-Methode.

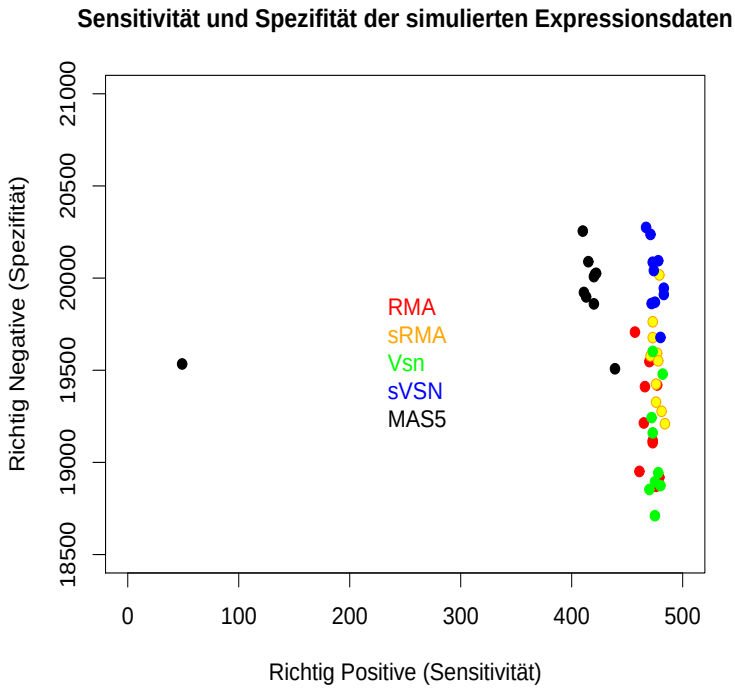


Abbildung 3.29: Sensitivität und Spezifität der Präprozessierungsmethoden für alle 10 Simulationen

Unabhängig vom gewählten FDR-*Cut-Off* wurden auch hier ROC-Kurven für die verschiedenen Normalisierungsmethoden erstellt. Dabei wurde das FDR-Intervall $[0, ..., 1]$ in 1000 Bereiche unterteilt und dazu Sensitivität und Spezifität, wie in Kapitel 2.11 beschrieben, für alle 10 Simulationen berechnet. Abbildung 3.30 zeigt die bei allen 10 Simulationen gemittelte ROC-Kurve.

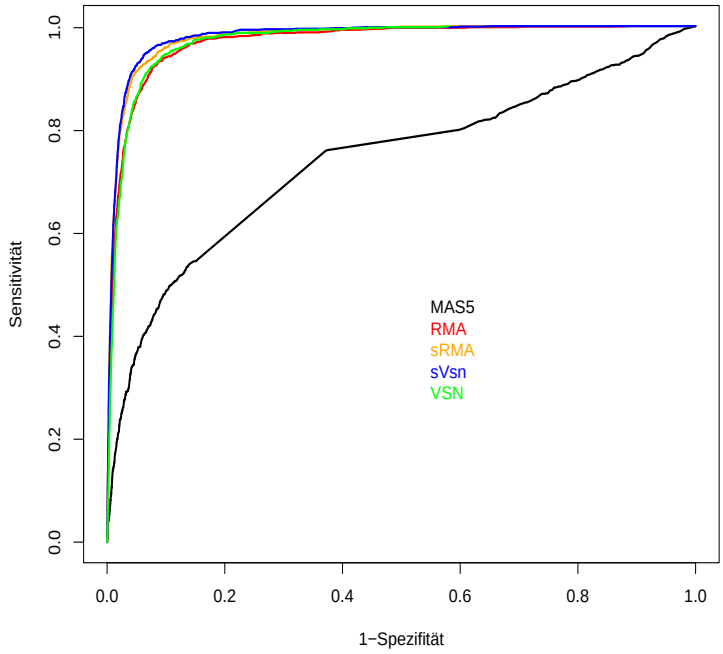


Abbildung 3.30: Sensitivität und Spezifität dargestellt in einer ROC-Kurve bei Simulation der Expressionsdaten mit eines linear gemischten Modells

Betrachtet man die ROC-Kurve 3.30, zeigte die Methode MAS5 die schlechteste Sensitivität und Spezifität. Alle anderen Methoden grenzen sich deutlich von der Methode MAS5 ab, wobei sVSN eine minimal bessere Sensitivität und Spezifität zeigte, als die anderen Methoden.

3.5 Vergleich der Normalisierungsmethoden mit den Daten von Brune et al.

Zusätzlich zu den theoretischen Simulationen wurde ein realer Datensatz mit den verschiedenen Präprozessierungsmethoden analysiert. Verwendet wurde ein Teildatensatz aus der Arbeit von Brune et al. [Brune et al., 2008]. Gearbeitet wurde mit zwei Gruppen, 12 HLs und 10 Keimzentrum-B-Zellen, deren differentielle Expression biologisch besonders interessant ist.

3.5.1 Deskriptive Betrachtung der Expressionsdaten

Betrachtet wurden zunächst die rohen Expressiondaten hinsichtlich ihrer Verteilung. Abbildung 3.31 zeigt das Histogramm der PM-Expressionswerte. Es zeigt ähnliche Verteilungseigenschaften wie Abb. 3.5. Die Expressionsdaten sind nicht normalverteilt zeigen auf der linken Seite einen steilen Anstieg und auf der rechten Seite einen lang-gezogenen Abfall der Expressionswerte.

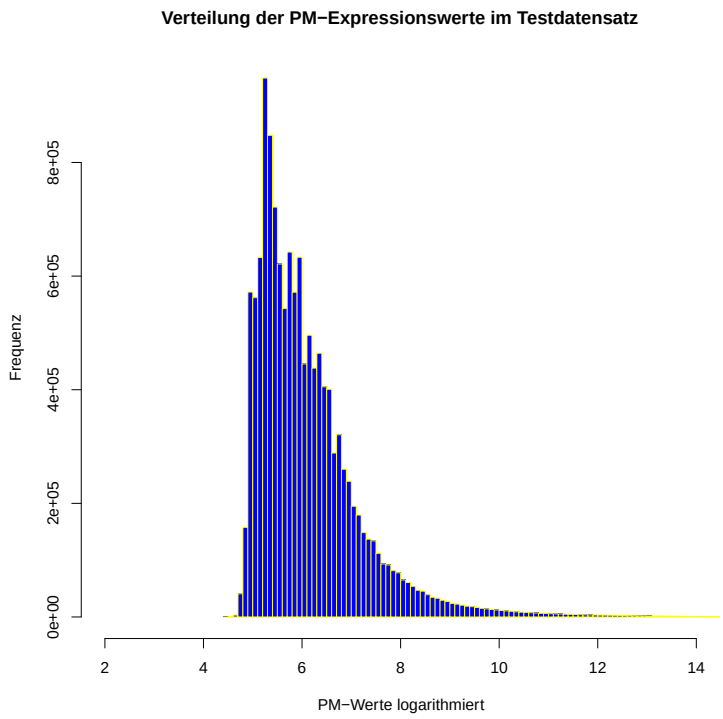


Abbildung 3.31: Histogramm der logarithmierten Expressionsdaten

Bei der Darstellung der einzelnen Boxplots der Expressionsdaten (Abb. 3.32) zeigen sich wesentlich größere Schwankungen als beim Testdatensatz. Grund hierfür können verschiedene Patienten und Spender sein und damit eine höhere biologische Variabilität als beispielsweise bei Zelllinien.

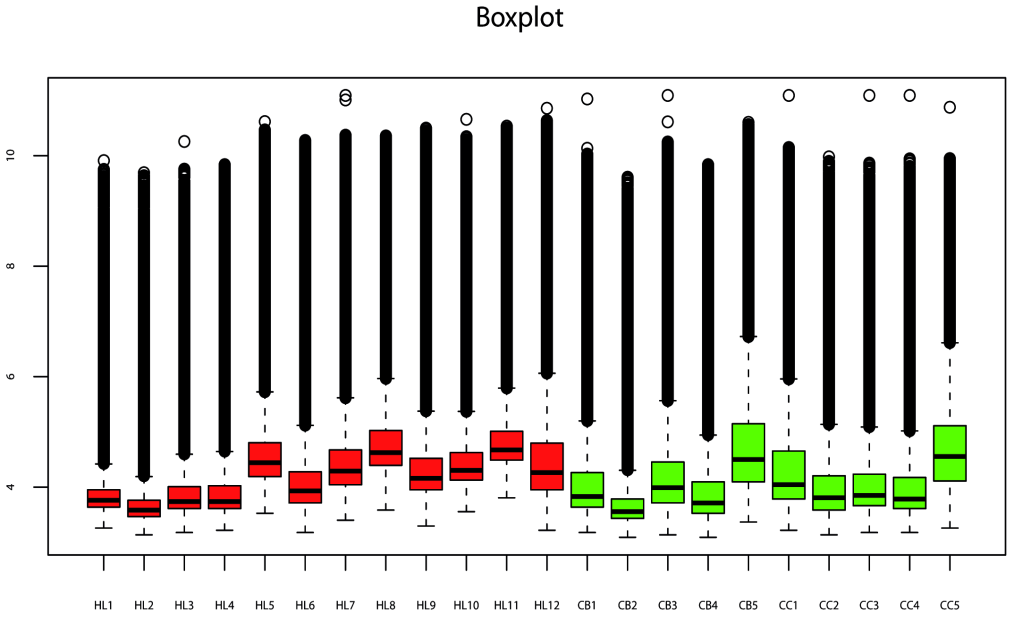


Abbildung 3.32: Boxplot der logarithmierten Expressionsdaten, der 12 HLs und 10 Keimzentrum-B-Zellen

Abbildung 3.33 zeigt den Zusammenhang von Expression und Position im *Probeset*. Zu erkennen ist ein schwacher Abfall der Expression mit abnehmender Position, d. h. zunehmender Entfernung zum 3'-Ende.

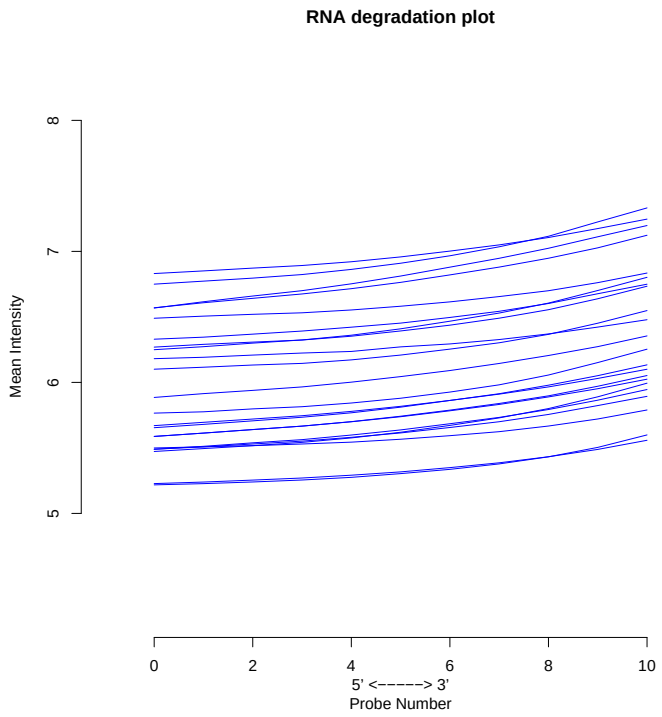


Abbildung 3.33: Degradationsgrad der PM-Werte in Abhängigkeit von ihrer Position in den einzelnen *Microarrays*

3.5.2 Betrachtung der differentiell-exprimierten Gene

Nachdem die Expressionsdaten mit allen fünf Präprozessierungsmethoden analysiert wurden, wurden im nächsten Schritt die differentielle Expression zwischen HL und Keimzentrum-B-Zellen betrachtet. Verwendet wurden zunächst Vorfilter, da ein Teil der Gene nicht ausgeprägt war oder eine geringe Variabilität zeigte. Intensitätsfilter und Interquartilsfilter liegen bei 0,4, d. h. bei dieser Intensität müssten 40 % der Expressionswerte einer Zeile über dem Hintergrundsignal liegen (< 6) und beim Varianzfilter sollte der Interquartilsabstand größer als 0,4 sein.
Unter der Annahme, dass die Varianz in beiden Gruppen gleich groß ist, wurden nach der globalen Filterung p-Werte mit einem Zwei-Stichproben-t-Test berechnet, um Gene zu identifizieren, die zwischen den beiden Gruppen differentiell ausgeprägt waren. Aufgrund des Multiplen-Testen-Problems wurden die FDR nach Benjamini und Hochberg berechnet [Benjamini et al., 2001].
In Tabelle 3.4 ist die Anzahl differentiell-exprimierter *Probesets* mit zwei unter-

schiedlichen FC-Grenzen für die verschiedenen Methoden zusammengefasst.

Genliste	FC≥ 2 / ≤ -2,FDR≤ 0.05	FC≥ 5 / ≤ -5,FDR≤ 0.05
cHL vs CC+CB sVSN	444	86
cHL vs CC+CB VSN	1119	259
cHL vs CC+CB RMA	1229	358
cHL vs CC+CB sRMA	599	179
cHL vs CC+CB MAS5	5463	1564

Tabelle 3.4: Anzahl der Gene, die differentiell sind, und nach Durchführung der verschiedenen Normalisierungsmethoden am Datensatz von Brune et al. gefunden wurden

Auffällig ist die große Anzahl differentiell exprimierter *Probesets* bei der Methode MAS5. Die Methoden sVSN und sRMA zeigen die geringste Anzahl differentiell-exprimierter *Probesets*. Die Methoden VSN und RMA haben manchmal sensitiver und manchmal spezifischer abgeschnitten als sRMA und sVSN, abhängig davon, welche simulierten Expressionsdaten betrachtet wurden. In der Simulation auf Basis eines gemischten linearen Modell zeigte das sVSN die größte Spezifität und Sensitivität. Da dies auch bei den Simulationen mit den anderen Methoden, außer MAS5, vergleichbar ist, wird im Folgenden im Detail nur auf die Genliste mit sVSN eingegangen.
Die Genliste umfasst nach der Durchführung mit Methode sVSN insgesamt 444 *Probesets*, die einen FC von ≥ 2 oder ≤ -2 und einen FDR von ≤ 0.05 haben. Zuerst wurden auf Einzel-Gen-Ebene die Gene betrachtet, die nur nach der Methode sVSN gefunden wurden. Das waren insgesamt 66 *Probesets* (siehe Tabelle 6.3). Nächste Frage war, ob diese Genliste Gene enthält, die entweder schon durch andere wissenschaftliche Arbeiten publiziert wurden oder zumindest in ihrer Funktion und mit ihrem Signalweg interessant für das HL sind.

Tabelle 3.5 enthält 9 Gene aus Tabelle 6.3 und zu ihrer Funktion im HL weiter untersucht wurden:
IGFBP7 — ein Insulin Wachstumsfaktor — war im Vergleich zu Keimzentrum-B-Zellen im HL hochreguliert. In der Literatur ist IGFBP7 im HL noch nicht beschrieben, aber in vielen anderen Tumoren, wie z. B. im Kolonadenomkarzinom [Ruan et al., 2006].

JUN ist ein Transkriptionsfaktor der AP-1 Familie, der schon länger in verschiedenen Tumoren bekannt ist, aber auch im HL vorkommt [Mathas et al., 2002]. JUN kontrolliert und beeinflusst Proliferation, Apoptose und maligne Transformation von Zellen. Die Überexpression von JUN wurde in einer Studie bei allen HL-Patienten nachgewiesen und spielt für die Pathogenese des HLs eine wichtige Rolle [Mathas et al., 2002].
CCND2 gehört zur Familie der Zykline und ist ein wichtiges Gen für den Zellzyklus. Es ist im HL mit erhöhter Expression beschrieben [Bai et al., 2004].
ID2 gehört zur Familie der Tyrosinkinasen und ist ebenfalls im HL beschrieben [Renné et al., 2006].
TIMP1 ist ein Kollagenase Inhibitor und scheint in der Pathogenese des HLs eine Rolle zu spielen [Thorns et al., 2002].
IL6 gehört zur Interleukin-Familie, wurde bei der Reanalyse der Expressionsdaten von Küppers et al. in den HL-Zellinien als differentiell exprimiert gefunden und ist im HL als hochreguliert beschrieben [Nagel et al., 2005].
LGALS2, G0S2 (Zellzyklus Gen) und TLR8 (Toll-ähnlicher-Rezeptor) sind nur in anderen Tumoren beschrieben.

Affymetrix ID	HL/B-Zellen FC	HL/B-Zellen FDR	Gensymbol
201162_at	2.9	0.031	IGFBP7
201464_x_at	2.1	0.040	JUN
200951_s_at	4.0	0.000	CCND2
201566_x_at	2.2	0.006	ID2
201666_at	3.3	0.004	TIMP1
205207_at	3.2	0.001	IL6
208450_at	3.0	0.002	LGALS2
213524_s_at	3.6	0.002	G0S2
229560_at	2.3	0.000	TLR8

Tabelle 3.5: Ein Teil der Gene, die ausschließlich mit der sVSN-Methode im Datensatz von Brune et al. als differentiell gefunden wurden

3.6 Zusammenfassung

Bei den Simulationsansätzen zeigten die fünf verschiedenen Präprozessierungsmethoden unterschiedliche Ergebnisse. Bei der Simulation — basierend auf einer empirischen Verteilung — zeigte sRMA die beste Spezifität, aber nicht die beste Sensitivität (zur Definition von Spezifität und Sensitivität siehe Kapitel 2.11). Die Methoden sVSN, sRMA und VSN streuten sehr stark über die wiederholten Simulationen, wobei sRMA und sVSN die beste Sensitivität zeigten. Die Methode MAS5 zeigte die schlechteste Sensitivität und die schlechteste Spezifität.
Bei der Simulation — basierend auf einer angepassten Gammaverteilung — zeigten die Methoden sRMA, sVSN, VSN und RMA die beste Spezifität, aber einer schlechtere Sensitivität als die Methode MAS5. Allerdings zeigte die Methode MAS5 die schlechteste Sensitivität.
Bei der Simulation — basierend auf einem linear gemischten Modell — zeigte sVSN die beste Sensitivität und die beste Spezifität. Die Methoden sRMA, RMA und VSN zeigten eine vergleichbare Sensitivität, aber eine schlechtere Spezifität. Die Methode MAS5 zeigte im Vergleich zur Methode sVSN eine ähnlich gute Spezifität, aber eine schlechtere Sensitivität.
Bei den ersten beiden Simulationsansätzen — basierend auf der empirischen und der Gamma-Verteilung — konnte keine Methode eindeutig favorisiert werden. Im letzten Simulationsansatz — mit einem linear gemischten Modell — hat eindeutig sVSN die beste Spezifität und die beste Sensitivität gezeigt.
Ein Teildatensatz von Brune et al. [Brune et al., 2008] wurde mit den fünf Präprozessierungsmethoden verglichen. Die Methode sVSN wurde genauer betrachtet und 66 zusätzliche *Probesets* ausschließlich mit dieser Methode gefunden. Ein Teil der Gene wurde mit der Literatur abgeglichen, fünf Gene (JUN, CCND2, ID2, TIMP1, IL6) sind in anderen Publikation im HL als differentiell-exprimiert beschrieben.

Kapitel 4

Diskussion

4.1 Reanalyse von publizierten Datensätzen

Wie schon in Abschnitt 3.3 zusammengefasst, bietet die Reanalyse von publizierten Datensätzen mit neuen statistischen Verfahren wichtige Vorteile. Zum einem konnten die schon gefunden Gene bestätigt werden, zum anderen konnten zusätzliche Gene ergänzt werden. Diese haben direkten Einfluss auf alle Analysen, die nicht nur genweise die Genlisten der differentiell exprimierten Gene untersuchen, sondern vor allem Gen-Ontologien und Signalwege auswerten.

4.1.1 Reanalyse des Datensatzes von Küppers et al.

Bei Reanaylse der Expressionsdaten von Küppers et al. [Küppers et al., 2003] konnte gezeigt werden, dass die in dieser Arbeit angewandte Analysestrategie zusätzliche wichtige Gene gefunden hat. Die Normalisierungsmethode von Huber et al. [Huber et al., 2003] hat entscheidenen Einfluss auf die Analyse. Hauptkritikpunkt an der Methode von Klein et al. ist, dass der Affymetrix-Algorithmus MAS4 oder MAS5 verwendet wurde. Vor allem das Einbeziehen der MM-Expressionswerte ist beim Berechnen der Expressionswerte ein Nachteil [Irizarry et al., 2003a; Lazaridis et al., 2002].

Dass die gesamte Genliste, die Küppers et al. als differentiell identifiziert haben, nach der Reanalyse wieder gefunden wurde zeigt, dass die Analyse von Klein et al. nicht grundsätzlich falsch ist. Die Methode ist bezüglich der Sensitivität allerdings schwächer als andere hier vorgestellte Methoden.

Die zusätzlich identifizierten differentiell-exprimierten Gene (299 *Probesets*) wurden

zum Teil durch andere Forschungsgruppen bestätigt. Außerdem konnte bei primären HL aus dem Datensatz von Brune et al. [Brune et al., 2008] die Genexpression von einem Teil der Gene auch im HL bestätigt werden. Das bestärkt noch einmal die Notwendigkeit neuer statistischer Methoden und bestätigt zugleich die Ergebnisse dieser Arbeit. Alle zusätzlichen Gene, die noch nicht im Zusammenhang mit HL-Linien oder HRS-Zellen veröffentlicht sind, müssen im nächsten Schritt validiert und funktional untersucht werden. Dies bietet die Chance, aufgrund der Ergebnisse dieser Arbeit, neue Signalwege und Gene für die Aufklärung der Pathogenese des HLs zu identifizieren.
Ein wichtiger Signalweg ist mit Sicherheit der Zytokine/Zytokine-Signalweg, der mit zusätzlichen 12 Genen aus der Reanaylse signifikant überrepräsentiert ist. Außerdem konnten zusätzliche B-Zellen-Marker identifiziert werden, die zum Teil noch nicht im Zusammenhang mit HLs beschrieben wurden.

4.1.2 Reanalyse des Datensatz von Piccaluga et al.

Die Reanalyse des Datensatz von Piccaluga et al. [Piccaluga et al., 2007] zeigte dieselbe Tendenz wie die Reanalyse der Expressionsdaten von Küppers et al. [Küppers et al., 2003]. Das liegt daran, dass die Analysestrategie von Ulf Klein [Klein et al., 2001] dieselbe ist, obwohl zwischen beiden Analysen einige Jahre liegen. Alle Gene konnten durch die Reanalyse wiedergefunden werden. Zusätzlich wurden 599 *Probesets* indentifiziert, die einen signifikanten Unterschied von mindestens $FC \geq 4$ bzw. ≤ -4 zwischen T-Zell-Lymphom vom Typ NOS im Vergleich zu normalen T-Zellen zeigen. Der Zytokine/Zytokine-Signalweg wurde in diesem Vergleich als überrepräsentiert klassifiziert. Von den 12 Genen, die hierzu gefunden wurden, war nur ein Gen in den T-Zell Lymphomen vom Typ NOS beschrieben. Ein Teil der anderen Gene wurde bei anderen Tumoren beschrieben. Dies lässt vermuten, dass diese eine wichtige Funktion in T-Zell-Lymphomen NOS spielen.
Piccaluga et al. führten auch eine GO-Analyse durch. Allerdings wurden bei der Reanalyse keine einzige Gen-Ontologie wiedergefunden. Dies kann verschiedene Ursachen haben. Maßgeblich ist natürlich die Genliste, die für die GO-Analyse verwendet wird. Nach der Reanalyse ist die Genliste mehr als doppelt so groß. Auffällig ist, dass die Bonferoni-Werte bei Piccaluga et al. zum Teil sehr groß und die Signifikanzen damit eher fragwürdig sind.

4.2 Simulation von Expressionsdaten

Bei der Simulation von Expressionsdaten auf *Probe*-Ebene konnte gezeigt werden, dass die Wahl des Modells, in der Praxis also die Verteilungsstruktur der Daten, entscheidenden Einfluss auf die Ergebnisse hat.

4.2.1 Simulation der Expressionsdaten — basierend auf einer kontinuierlichen Verteilung

Betrachtet man das Histogramm der PM-Expressionswerte, so sind die Expressionsdaten nicht normalverteilt. Es wurden einige Verteilungen (Weibull, Log-Normal, Gamma und Empirische Verteilung) mit Histogrammen und QQ-Plots zur Vergleichbarkeit der Verteilung der realen Expressionsdaten untersucht. Die Simulation von Expressionsdaten — basierend auf der empirischen Verteilung — zeigte die beste Übereinstimmung mit den realen Expressionsdaten.
Die anderen Verteilungen zeigten, vor allem an den Enden im QQ-Plot, eine schlechte Abdeckung der Expressionswerte. Die simulierten Expressionsdaten — basierend auf der empirischen Verteilung — wurden auf ihre Spezifität und Sensitivität untersucht. Für die fünf Präprozessierungsmethoden konnte keine zu favorisiertende Methode gefunden werden. Die Methoden sRMA und sVSN zeigten eine starke Streuung über alle durchgeführten Simulationen. Ursache könnte sein, dass immer andere *Probesets* für die Simulation der differentiellen Expression verwendet wurden. Allerdings zeigten sRMA und sVSN eine höhere Sensitivität. Die Methode MAS5 war auch in den folgenden Simulationen immer separiert von den restlichen Simulationen. Das zeigte sich vor allem in den ROC-Kurven. Ursache hiervon ist, dass die Methode MAS5 die MM-Expressionswerte in die PM-Adjustierung mit einbezieht. Für diese Simulation gibt es keine Methode, die sowohl die höchste Sensitivität als auch die höchste Spezifität aufweist. Wichtiger ist eine höhere Sensitivität, da eine niedrigere Spezifität und damit ein höherer Anteil von falsch positiven Genen einen höheren Validierungsaufwand bedeutet.
Die Simulation — basierend auf der empirischen Verteilung — hat den Nachteil, dass die simulierten Expressionsdaten extrem abhängig von den realen Expressiondaten sind. Außerdem wurde für die Abhängigkeit der Expression von der Position im Transkript eine extreme Sortierung vorgenommen, die so in den realen Expressionsdaten nur zum Teil vorkommt. Angenommen wird, dass die Expression mit

größer werdenden Abstand zum 3'-Ende die abnimmt.

4.2.2 Simulation der Expressionsdaten — basierend auf einer angepassten Gammaverteilung

Die Expressionsdaten folgen keiner Gammaverteilung (Abb. 3.6). Allerdings kann man versuchen, die geeignet transformierten realen Expressionsdaten hinsichtlich einer Gammaverteilung anzupassen. Hierzu wurde das Minimum von den Expressionsdaten abgezogen und dann Expressionsdaten auf Basis einer Gammaverteilung simuliert. Zwar zeigte der QQ-Plot eine schlechte Übereinstimmung im oberen Expressionsbereich im Vergleich zu den realen Expressionsdaten, dennoch sollten mit diesen simulierten Expressionsdaten die verschiedenen Präprozessierungen untersucht werden. Die Methoden sVSN, sRMA, VSN und RMA zeigten eine vergleichbare Sensitivität und Spezifität. Allerdings zeigte die Methode RMA in der ROC-Kurve die schlechteste Sensitivität und die schlechteste Spezifität.
Aufgrund der schlechten Abdeckung der Expressionswerte im oberen Expressionsbereich war die angepasste Gammaverteilung für die Beurteilung der verschiedenen Präprozessierungsmethoden nur bedingt verwendbar. Außerdem wurde die selbe totale Sortierung der *Probes* verwendet wie bei der Simulation, die auf einer empirischen Verteilung basiert.

4.2.3 Simulation von Expressionsdaten — basierend auf einem gemischten linearen Modell

Bei der Simulation auf Basis der empirischen oder der Gamma-Verteilung wurde der *Probe*-Effekt nicht mit einbezogen. Der Zusammenhang zwischen der *Probe*-Position und der Expression durch die vollständige Sortierung innerhalb des *Probesets* wurde erst im Nachhinein berücksichtigt.
Der Einsatz eines linearen gemischten Modells ermöglichte es, diesen Aspekt in den sogenannten Random-Effekt mit einzubeziehen. Ein einfaches lineares Modell kam hier nicht in Frage, da die Position im Transkript mit der Expression korreliert war. Bei der Durchführung mit verschiedenen Kombinationen im Random-Effekt war aufgefallen, dass die Variable *Chip* keine große Variablität zeigte und es zu Konvergenzproblemen bei der numerischen Maximierung der Likelihood kam. Deswegen war der Variable-*Chip* im Random-Effekt nicht enthalten. Außerdem setzten line-

ar gemischte Modelle eine Normalverteilung der Expressionsdaten voraus. Bei dem durch das linear gemischte Modell simulierten Expressionsdaten zeigten eindeutig die Methode sVSN die größte Sensitivität und die größte Spezifität. Die ROC-Kurve zeigte, außer bei MAS5, für alle anderen Methoden eine ähnliche Sensitivität und Spezifität. sVSN zeigte eine minimale bessere Sensitivität und Spezifität.
Keine Präprozessierungsmethode zeigte in den verschiedenen Simulationen die größte Sensitivität und die größte Spezifität. Allerdings zeigte sVSN sowohl bei der Simulation der Expressionsdaten mit einer empirischen Verteilung und bei Verwendung eines linear gemischten Modells die beste Sensitivität und die beste Spezifität. Bei Berücksichtigung des *Probe*-Effekts sollte man daher die Methode sVSN wählen.
Leider stand kein realer Datensatz mit bekannten differentiell-exprimierten Genen, die mit doppelter IVT erstellt wurden, zur Verfügung. Für den Datensatz von Cope et al. [Cope et al., 2006] wurde geschlechtspezifisches Gewebe (Brustgewebe, Hodengeweben) verwendet, um sicher zu sein, dass Gene die nur auf dem Y-Chromosom zu finden sind, dann differentiell exprimiert sein sollten. Allerdings geht das dort verwendete 2-Runden-Protokoll von der 100-fachen Ausgangs-RNA-Menge aus und ist daher mit in dieser Arbeit verwendeten Expressionsdaten nicht vergleichbar.

4.3 Vergleich der verschiedenen Präprozessierungsmethoden bei einem ausgewählten Vergleich aus dem Datensatz von Brune et al.

An einem Teildatensatz von Brune et al. [Brune et al., 2008] wurden die fünf Präprozessierungsmethoden durchgeführt und miteinander verglichen. Die große Anzahl differentiell exprimierter *Probesets* bei Verwendung der MAS5-Methode lässt vermuten, dass diese Methode fälschlicherweise viele differentiell-exprimierte Gene hervorbringt und entweder ein Problem mit der Spezifität oder mit der Sensitivität hat. Bei der Anwendung eines linear gemischten Modells, das den *Probe*-Effekt berücksichtigt, zeigte die Methode sVSN die größte Spezifität und die größte Sensitivität. Der VSN-Algorithmus zeigte, bei einfacher IVT prozessierten Expressionsdaten, eine größerer Spezifität und eine größere Sensitivität als die RMA-Algorithmus. Das zusätzliche Gewichten der einzelnen *Probes* nach der Normalisierung ergibt einen

zusätzlichen Vorteil für sVSN.

4.3.1 Differentielle Expression von cHL im Vergleich zu normalen B-Zellen

Einen Teil der 66 *Probesets*, die nur mit der Methode sVSN identifiziert wurden, wurde mit der Literatur abgeglichen. Dabei wurden fünf Gene (JUN, CCND2, ID2, TIMP1, IL6), im HL beschrieben. Alle anderen Methoden konnten diese Gene nicht identifizieren. Zusätzliche Gene, die noch nicht im Zusammenhang mit dem HL beschrieben sind, könnten daher eine Möglichkeit sein, zusätzliche Funktionen und Signalwege, die für die Pathogenese des HL wichtig sind, zu finden.

4.4 Ausblick

Die Möglichkeit, aus wenigen Zellen ein Genexpressionsprofil mit einem *Microarray* zu erstellen, war bis vor wenigen Jahren noch undenkbar. Vor allem mikrodissektierte Zellen — also einzelne Zellen, die aus einem Gewebeschnitt isoliert wurden — waren schwer zu handhaben. Für sie wurde ein Protokoll etabliert, das es zum ersten Mal ermöglichte aus, 1000 Zellen ein Genexpressionsprofil zu erstellen. Allerdings waren die resultierenden Expressionsdaten nicht unproblematisch. Dadurch, dass man die mRNA um ein Vielfaches vermehren musste, gelangten zusätzliche Fehler ins Expressionsprofil.
Diesen Fehler versucht man im Labor in den Griff zu bekommen. Neuere Protokolle versprechen an dieser Stelle eine wesentliche Verbesserung. Einige Protokolle gehen dazu über, sich nicht mehr am Poly-A-Schwanz der RNA zu orientieren, sondern durch randomisierte Makierungen davon unabhängig zu werden.
Bis jetzt gibt es aber keine systematische Untersuchung, inwieweit die Effekte der *Probe*-Position dadurch tatsächlich minimiert werden.
Die *Microarray*-Hersteller haben zum Teil auf diese Problematik reagiert. Neuere *Microarray*-Generationen gehen bei ihrem *Probedesign* nicht mehr vom 3'-Ende des mRNA-Transkriptes aus. Vielmehr werden jetzt alle Exone eines Transkripts mit speziell dafür entworfenen *Probes* abgedeckt. Dadurch hat sich das Protokoll für das Labor komplett geändert. Allerdings sind für diese Art von Arrays die Ausgangsmengen an mRNA noch zu groß und für eine doppelte IVT nicht geeignet. Deshalb

kommt diese Technologie für unserer Art von Zellproben noch nicht in Frage.

Kapitel 5

Zusammenfassung

Zur genomweiten Genexpressionsanalyse werden *Microarray*-Experimente verwendet. Ziel dieser Arbeit ist es, Methoden zur Präprozessierung von *Microarrays* der Firma Affymetrix zu evaluieren und die VSN-Methode für Experimente mit weniger als 1000 Zellen zu verbessern. Bei dieser Technologie wird die Expression jedes Gens durch mehrere *Probessets* gemessen. Jedes *Probeset* besteht aus einem *Perfect-Match* (PM) und einem dazugehörigen *Mismatch* (MM). Der Expressionswert pro Gen wird durch ein vierstufiges Verfahren aus den einzelnen *Probe*-Werten berechnet: Hintergrundkorrektur, Normalisierung, PM-Adjustierung und Aggregation. Für jeden dieser Schritte existieren mehrere Algorithmen. Dabei dienen die im affy-Paket des Bioconductor implementierten Methoden MAS5, RMA, VSN und die Methode sRMA von Cope et al. [Cope et al., 2006] in Kombination mit der Methode VSN-Methode von Huber et al. [Huber et al., 2002].
Den ersten Teil dieser Arbeit bildet die Reanalyse der Datensätze von Küppers et al. [Küppers et al., 2003] und Piccaluga et al. [Piccaluga et al., 2007] mit der VSN-Methode. Dabei konnte gezeigt werden, dass die VSN-Methode gegenüber Klein et al. [Klein et al., 2001] Vorteile zeigt. Bei beiden Datensätzen wurden zusätzliche Gene gefunden, die für die Pathogenese der jeweiligen Tumorarten wichtig sein können. Einige der zusätzlich gefunden Gene wurden durch andere wissenschaftliche Arbeiten bestätigt. Die Gene, die bisher in keinem Zusammenhang mit der untersuchten Tumorart stehen, sind eine Möglichkeit für die weitere Forschung. Vor allem der Zytokine/Zytokine Signalweg wurde bei beiden Reanalysen als überrepräsentiert erkannt.
Da für einige *Microarray*-Experimente die Anzahl der Zellen — und damit die Men-

ge an mRNA — nur begrenzt zur Verfügung stehen, müssen die Laborarbeit und die statistischen Analysen angepasst werden. Hierzu werden fünf Methoden für die Präprozessierung untersucht, um zu evaluieren, welche Methode geeignet ist, derartige Expressionsdaten zu verrechnen.
Auf Basis eines Testdatensatzes — der bereits zur Etablierung des Laborprozesses diente — werden Expressionswerte durch empirische Verteilung, Gammaverteilung und ein linear gemischtes Modell simuliert. Die Simulation lässt sich in vier Schritte einteilen: Wahl der Verteilung, Simulation der Expressionsmatrix, Simulation der differentiellen Expression, Sortierung der *Probes* innerhalb des *Probesets*. Anschließend werden die fünf Präprozessierungsmethoden mit diesen simulierten Expressionsdaten auf ihre Sensitivität und Spezifität untersucht.
Während sich bei den empirisch und gammaverteilt simulierten Expressionsdaten kein eindeutiges Ergebnis abzeichnet, hat sVSN bei den Daten aus dem linear gemischten Modell die größte Sensitivität und die größte Spezifität. Der in dieser Arbeit entwickelte sVSN-Algorithmus wurde zum ersten Mal angewendet und bewertet.
Abschließend wird ein Teildatensatz von Brune et al. verwendet und hinsichtlich der fünf Präprozessierungsmethoden untersucht. Die Ergebnisse der sVSN-Methode wird im Detail weiter verfolgt. Die zusätzlich gefunden Gene können durch bereits veröffentlichte Arbeiten bestätigt werden.
Letztendlich zeigt sich, dass neuere statistische Methoden (wie das im Rahmen dieser Arbeit entwickelte sVSN) bei der Analyse von Affymetrix *Microarrays* einen Vorteil bringen. Die sVSN und sRMA Methoden zeigen Vorteile, da die *Probes* nach der Normalisierung gewichtet werden, bevor diese aggregiert werden. Die MAS5-Methode schneidet am schlechtesten ab und sollte bei geringen Zellmengen nicht eingesetzt werden.
Für die Analyse mit geringer Menge an mRNA müssen weitere Untersuchungen vorgenommen werden, um eine geeignete statistische Methode für die Analyse der Expressionsdaten zu finden.

Kapitel 6

Appendix

6.1 R-Skript sVSN

```
##############################################################################
##
## Gewichtung der normalisierten Perfect Matches (PM) zur Anpassung der Daten,
## die mittels Small Sample Protokoll (zweifache Invitro-Transkriptionen [IVT])
## mit Affymetrix Microarray gewonnen wurden. In der Annahme dass die Expression
## der Probes abhängig ist von der Position im Transkript werden nach
## der Normalisierung diese gewichtet und dann mittels eines linearen Modells
## angepasst. So erhält man eine korrigierte Expressionsmatrix in der die Lage
## der Probes im Transkript berücksichtigt wurde.
## Das R-Skript mit RMA-Normalisierung ist aus der Publikation von
## Cope et al. [Cope et al., 2006].
##
## Die wichtigsten Variablen:
##
##   phenoData       -> Beschreibung der Datenmatrix
##   pl              -> nomierte Lokalisation der Probes
##   rowSD           -> Standardabweichung einer Stichprobe
##   NormPM          -> logarithmierte, normalisierte (mit VSN) PMs
##   Var             -> Varianz der jeweiligen Stichprobe
##   fit1            -> log. Varianz durch die Positionen modeliert
##   fit2            -> rowMean durch die Positionen modeliert
##   w1              -> nomierte Gewichte
##   eset            -> die gewichtete Expressionsmatrix
##   WVSN            -> die gewichtete Expressionsmatrix als ExpressionsSet
##                      inklusive der Pheno Daten
##
## ÜBERBLICK:
##
## 0. Init
## 1. Rohdaten mit VSN normalisieren
## 2. Gewichtung der Probes und Anpassung eines linearen Modells
## 3. Logarithmierung und Anwendung der Gewichte auf die Probes
## 4. ExpressionsSet erstellen und speichern
##
##############################################################################


##############################################################################
##
## 0. Init
##
```

```
##############################################################################

# Libraries laden
  library(affy)
  library(affyPLM)
  library(hgu133aprobe)
  library(hgu133acdf)
  library(hgu133a.db)
  library(vsn)
  library(MASS)

# In diesem Folder befinden sich die Cel-Files.
  setwd("I:/Cel")

# Einlesen der Rohdaten der Affymetrix Microarrays
#
# Testdatensatz mit 4 Microarrays mit 2 Gruppen mit jeweils 2 Replikaten. Ein
# Burkitt Lymphom und eine Hodgkin-Zelllinie (L428) wurden dazu als technische
# Duplikate mit dem Small Sample Protokoll (2-fache IVT) prozessiert.

# In den folgenden Zeilen werden Pheno Daten erstellt.
  df <- data.frame(Sample_ID=1:4,
  Chip_name=c("A_1", "A_2",
              "B_1","B_2"),
  Tissue=c("L428","L428","Burkitt","Burkitt"),
  RNA_synthesis=c("two-round","two-round","two-round","two-round),
  row.names=paste(1:4, sep="_"))

  metaData <- data.frame(labelDescription=c(
"Numbers",
"Factor levels",
"Characters",
"Tissue"))

  new("AnnotatedDataFrame")
  new("AnnotatedDataFrame", data=df)
  new("AnnotatedDataFrame",
  data=df, varMetadata=metaData)
  as(df, "AnnotatedDataFrame")

  a <- new("AnnotatedDataFrame")
  pData(a) <- df
  varMetadata(a) <- metaData
  validObject(a)

# Einlesen der Cel-Files in ein "AffyBatch"
  Data <- read.affybatch(filenames=c(
# Amplifikations Cel-Files
              "A_1.CEL","A_2.CEL",
              "B_1.CEL","B_2.CEL"))

  phenoData(Data) <- new("AnnotatedDataFrame", data=df, varMetadata=metaData)


# Ermöglicht den Aufruf einer Variablen nur mit dessen Namen.
  attach(hgu133aprobe)

# Vergibt eine fortlaufende Nummer der Interrogation Position.
  Index <- seq(along=Probe.Interrogation.Position)

# Vektor der Zuordnung von "Probe Interrogation Position" zu "Probe Set Id"
  vProbeInterPosToProbeSetId <- split(Probe.Interrogation.Position,Probe.Set.Name)

# Vektor der Zuordnung von "Probe Index" zu "Probe Set Id"
  vIndexToProbeSetId <- split(Index,Probe.Set.Name)


# Erstellen des Minimum Vektors (gleiche Länge wie vProbeInterPosToProbeSetId).
# Mit Hilfe von lapply kann man eine Funktion auf vProbeInterPosToProbeSetId
# ausführen. In dieser wird das Minimum der Position innerhalb eines Probeset
```

```
# mit der Anzahl der Probes wiederholt.
  Min <- lapply(vProbeInterPosToProbeSetId, function(x) rep(min(x),length(x)))

# Erstellen des Maximum Vektors (gleiche Länge wie vProbeInterPosToProbeSetId)
  Max <- lapply(vProbeInterPosToProbeSetId, function(x) rep(max(x),length(x)))

# In start befinden sich alle minimalen Positionen mit der Probe Set ID als
# fortlaufende Nummer, zur Identifizierung der einzelnen Probes.
  start <- unlist(Min)[unlist(Index)]

# siehe Start nur mit Max
  end <- unlist(Max)[unlist(Index)]

# In der Variablen pl wird der Anteil der Lokalisation normiert und ins
# Verhältnis gesetzt.
# Von der tatsächlichen Position wird das Minimum abgezogen und durch die
# Differenz von Maximum und Minimum geteilt.
  pl <- (Probe.Interrogation.Position-start)/(end-start)

# Folgende Variablen werden nicht weiter benötigt.
  rm(vProbeInterPosToProbeSetId,vIndexToProbeSetId,Min,Max)
  gc()

# get holt die Funktion xy2i aus dem package:hgu133acdf. xy2i convertiert die
# xy-Koordinate vom Cel-File in die fortlaufende Nummer aus dem CDF-File.
  tmp <- get("xy2i", "package:hgu133acdf")

# Lokalisation der PM-Probes von jedem Probe Set mit fortlaufender Nummer in
# der Affy-ID.
  pmIndex <- unlist(indexProbes(Data, "pm"))

# Gibt einen Vektor mit Übereinstimmungen zurück. In diesem Fall nur die PMs
# mit dazugehöriger Lokalisation.
  subIndex <- match(tmp(x, y), pmIndex)

# Berechnet die Standardabweichung einer Stichprobe.
  rowSD <- function(x)
    sqrt(ncol(x)/(ncol(x)-1)*(rowMeans(x^2)-rowMeans(x)^2))


##############################################################################
##
## 1. Rohdaten mittels VSN normalisieren
##
##############################################################################

  RawData<- Data[,which(pData(Data)[,4]=="two-round")]
# Normalisierung mit VSN
  NormData <- normalize.AffyBatch.vsn(nData2)

  NormPM <- log2(pm(NormData)[subIndex,])

  pd <- pData(NormData)

# Ermittlung der Anzahl der Replikate in der jeweiligen Gruppe.
  N11 <- sum(pd[,3]=="L428")
  N12 <- sum(pd[,3]=="Burkitt")

# Die Berechnung der Varianz in Abhängigkeit der Stichprobengröße.
  Var <- ((N11-1)*rowSD(NormPM[,pd[,3]=="L428"])^2+
          (N12-1)*rowSD(NormPM[,pd[,3]=="Burkitt"])^2)/(N11+N12-2)

##############################################################################
##
## 2. Gewichtung der Probes und Anpassung eines linearen Modells
##
##############################################################################


# Subset aus Var (hier die ersten 10000).
```

```
  Subset <- seq(1,length(Var),len=10000)

# logarithmierte Werte von Subset Var geordnet nach den Positionen aus pl.
  Y <- log((Var[order(pl)])[Subset])

# Affy-IDs zu Subset aus pl.
  X <- (sort(pl))[Subset]

# Anpassen eines linearen Modells  mit robuster Regression mittels eines M
# Schätzers. Hier wird die log. Varianz durch die Positionen modelliert.
  fit1 <- rlm(Y~X)

# Die Zeilenmittelwerte zu dem Subset mit den Positionen aus pl.
  Y <- rowMeans(NormPM)[order(pl)][Subset]

# Anpassen eines linearen Modells  mit robuster Regression mittels eines M
# Schätzers. Hier wird der rowMean durch die Positionen modelliert.
  fit2 <- rlm(Y~X)

# Produziert 10000 NA .
  w1 <- rep(NA,length=length(probeNames(Data)))

# Betrachtet das Verhältnis zw. echten Werten und Standardfehler (Standardabw.).
# Vor dem Modellfitting logaritmiert man die Varianzen, weil das Fitting
# dann robuster ist. Um dann Werte vorherzusagen (predict) also zu simulieren
# macht man es durch exp wieder umgekehrt.

  w1[subIndex] <-  predict(fit2,newdata=data.frame(X=pl))/sqrt(exp(predict
  (fit1,newdata=data.frame(X=pl))))

# Auf den restlichen Positionen nimmt man den Mittelwert (auf diesen Positionen
# will man keine Abweichungen durch Gewichte ausgleichen).
  w1[is.na(w1)] <- mean(w1,na.rm=TRUE)

# Auf 1 normieren, damit es ein multiplikativer Faktor (Gewicht) wird.
  w1 <- w1/mean(w1)
  range(w1)



##############################################################################
##
## 3. Logarithmierung und Anwendung der Gewichte auf die Probes
##
##############################################################################

  pns <- probeNames(NormData)

# Eindeutige probeNames finden.
  gns <- unique(pns)

# Ergebnis Matrix vorbereiten (alles NA).
  eset <- matrix(NA,length(gns),nrow(pData(NormData)))

# Die normalisierten PMs (Perfect Match) logarithmieren.
  pms <- log2(pm(NormData))

# Für jedes eindeutigen probeName.
  for(i in seq(along=gns))
  {
    cat(i," ")

# Index mit Positionen, wo pns dem i-ten probeName entspricht.
    Index <- which(pns==gns[i])

# Daten zu dem i-ten probeName.
    PMS <- pms[Index,]

# Die Probes entsprechen den Zeilen.
    probes <- as.factor(rep(1:nrow(PMS),ncol(PMS)))
```

```
# Die Chips ensprechen den Spalten.
    chips <- as.factor(rep(1:ncol(PMS),rep(nrow(PMS),ncol(PMS))))

# Gewichtete Matrix
    ws <- rep(w1[Index],ncol(PMS))

# Anpassung der Daten pro ProbeName i für alle Chips und alle ProbeNames unter
# Berücksichtigung der Gewichte (weighted least squares).
    fit1 <- rlm(as.vector(PMS)~-1+chips+probes,weight=ws)

# Unter der Annahme dass die einzelnen Probes zu ProbeName i (insbesondere
# deren Positionen) keinen Einfluss auf die Daten haben sollten, bleiben nur
# die Koeffizienten zu dem jeweiligen ProbeName i übrig.
    eset[i,] <- coef(fit1)[1:ncol(PMS)]
  }

##############################################################################
##
## 4. ExpressionsSet erstellen und speichern
##
## Erstellen des ExpressionsSets: Zuordnung der Zeilendefinintionen und
## Spaltendefinitionen.
## Hinzufügen der PhenoDaten und Ergebnis als .rda-Datei speichern.
##
##############################################################################

  rownames(eset) <- gns
  colnames(eset) <- rownames(pData(NormData))
  WVSN2 <- new("ExpressionSet",exprs=eset,phenoData=phenoData(NormData))
  save(WVSN2,file="newexprs.rda")
```

6.2 R-Skripte für die Simulation von Expressionsdaten

```
##############################################################################
##
## ZIEL
##
## In dem folgenden R-Skript sollen auf Basis von realen Daten
## (PM- und MM-Expressionsdaten) verschiedene Verteilungen angepasst
## werden (Weibull, LogNormal, Gamma und empirische Verteilung).
## R enthält im Paket MASS alle Funktionen für die genannten Verteilungen.
##
## ÜBERBLICK:
##
## 0. Init
## 1. Weibull-Verteilung
## 2. Log-Normalverteilung
## 3. Gamma Verteilung
## 4. Empirische Verteilung
## 5. Grafische Auswertung: der Daten aus 1. bis 4.
## 6. Betrachtung der probenweisen Mittelwerte und Varianzen für Daten aus 4.
## 7. Korrektur der Mittelwert- und Varianzstruktur für die empirische Simulation
##
##############################################################################



##############################################################################
##
## 0. Init
##
##############################################################################
```

```
# Libraries laden
  library(affy)
  library(MASS)
  library(vsn)

  setwd("I:/Cel")

# Funktionen laden
  source("I:/func_empVer.r")
  source("I:/func_fixMeansVars.r")

# Einlesen der Rohdaten der Affymetrix Microarrays
  dataRaw <- ReadAffy(filenames=c(
    "A-1","A_2.CEL",
    "B_1.CEL","B_2.CEL"))
  gc()

# Vektor mit logarithmierten PM-Daten erstellen
  PMs <- log2(pm(dataRaw))
  exprLevels <- as.vector(PMs)

# Histogramm der Rohdaten
# hist(exprLevels, breaks = 100,col="blue", border="yellow", xlim=c(4,14),
# main="Verteilung der PM-Expressionswerte im Testdatensatz",
#       xlab="PM-Werte logarithmiert", ylab="Häufigkeiten")

##############################################################################
##
## 1. Weibull-Verteilung: Maximum-Likelihood Anpassung an die Microarraydaten
##
##############################################################################

# Teildatensatz der Länge 1000
  x<-sort(exprLevels)[seq(1001, length(exprLevels), 1000)]

# Schätzen der Form- und Skalenparemeter mittels folgendem Aufruf möglich
# x_weibull <- fitdistr(exprLevels,densfun=dweibull, start=list(scale=1,shape=2))

# Simulation eines Datensatzes mit Weibull-Verteilung
  xWeibull <- rweibull(length(x), shape=4.59, scale = 6.93)
  hist(xWeibull)
  hist(exprLevels)

# Darstellung der simulierten Expressionsdaten im Vergleich zu den realen
# Expressionsdaten in einem QQ-Plot
  qqWeibull <- qqplot(x, xWeibull)
  abline(lmsreg(qqWeibull$x, qqWeibull$y))


##############################################################################
##
## 2. Log-Normalverteilung: : Maximum-Likelihood Anpassung an die Microarraydaten
##
##############################################################################

# Schätzen der Paremeter mittels folgendem Aufruf möglich
#x_lognorm <- fitdistr(exprLevels,densfun=dlnorm, start=list(meanlog = 0, sdlog = 1))

# Simulation eines Datensatzes mit Log Normalverteilung
  xLognorm <- rlnorm(length(x), meanlog = 1.84, sdlog = 0.17)
  hist(xLognorm)
  hist(exprLevels)

# Darstellung der simulierten Expressionsdaten im Vergleich zu den realen
# Expressionsdaten in einem QQ-Plot
  qqLognorm <- qqplot(x, xLognorm)
  abline(lmsreg(qqLognorm$x, qqLognorm$y))


##############################################################################
```

```
##
## 3. Gamma Verteilung: Maximum-Likelihood Anpassung an die Microarraydaten
##
################################################################################

# Schätzen der Form- und Skalenparemeter mittels folgendem Aufruf möglich
#x_gamma <- fitdistr(exprLevels,densfun=dgamma, start=list(shape=2, scale=2))

# Simulation eines Datensatzes mit Gamma Verteilung
  xGamma <- rgamma(length(x), shape=33, scale=0.2)
  hist(xGamma)
  hist(exprLevels)

# Darstellung der simulierten Expressionsdaten im Vergleich zu den realen
# Expressionsdaten in einem QQ-Plot
  qqGamma <- qqplot(x, xGamma)
  abline(lmsreg(qqGamma$x, qqGamma$y))


################################################################################
##
## 4. Empirische Verteilung
##
################################################################################

# Daten mit empirischer Verteilung simulieren
  mExprSim <- matrix(empVer(length(exprLevels), exprLevels, 1000), ncol=ncol(PMs))
  hist(mExprSim)
  qqExprSim <- qqplot(exprLevels, mExprSim)
  abline(lmsreg(qqExprSim$x, qqExprSim$y))

# Mittelwerte und Varianzen
  meanDataRaw <- apply(PMs, 1, mean)
  varDataRaw  <- apply(PMs, 1, var)

  meanSim <- apply(mExprSim, 1, mean)
  varSim  <- apply(mExprSim, 1, var)

# Histogramme
  hist(meanDataRaw)
  hist(varDataRaw)
  hist(meanSim)
  hist(varSim)

# Q-Q-Plots
  qqplot(meanDataRaw, meanSim)
  qqplot(varDataRaw, varSim)


################################################################################
##
## 5. Grafische Auswertung: Histogramme und Q-Q-Plots der simulierten Daten
##    aus den Verteilungen aus 1. bis 4.
##
################################################################################

  par(mfrow = c(2,4))

# Histogramme

  hist(xWeibull, breaks = 100,col="blue", border="yellow", xlim=c(4,14),
  main="Weibullverteilung",
  xlab="shape=4.59 und scale=6.93", ylab="Häufigkeiten")

  hist(xLognorm, breaks = 100,col="blue", border="yellow", xlim=c(4,14),
  main="Log-Normalverteilung",
  xlab="meanlog = 1.84 und sdlog = 0.17", ylab="Häufigkeiten")

  hist(xGamma, breaks = 100,col="blue", border="yellow", xlim=c(4,14),
  main="Gamma-Verteilung",
```

```
  xlab=" shape=6.8", ylab="Häufigkeiten")

  hist(mExprSim, breaks = 100,col="blue", border="yellow", xlim=c(4,14),
  main="Empirische Verteilung",
  xlab="Empirische Verteilung", ylab="Häufigkeiten")

# Q-Q-Plots

  qqplot(x, xWeibull, main="Q-Q-Plot",xlab="Expressionswerte", ylab="Weibull-Simulation")
  abline(lmsreg(qqWeibull$x, qqWeibull$y))

  qqplot(x, xLognorm, main="Q-Q-Plot",xlab="Expressionswerte", ylab="Log-Normal-Simulation")
  abline(lmsreg(qqLognorm$x, qqLognorm$y))

  qqplot(x, xGamma, main="Q-Q-Plot",xlab="Expressionswerte", ylab="Gamma-Simulation")
  abline(lmsreg(qqGamma$x, qqGamma$y))

  qqplot(x, mExprSim, main="Q-Q-Plot",xlab="Expressionswerte", ylab="Empirische-Simulation")
  abline(lmsreg(qqExprSim$x, qqExprSim$y))
  dev.off()


################################################################################
##
## 6. Empirische Simulation: Grafische Betrachtung der probenweisen Mittelwerte
##    und Varianzen
##
################################################################################

  par(mfrow = c(2,3))

# Mittelwerte: Histogramme und Q-Q-Plots
  hist(meanDataRaw, breaks = 100, col="blue", border="yellow", main="Gemessene Daten",
       xlab="Probeweise Mittelwerte", ylab="Häufigkeiten")

  hist(meanSim, breaks = 100, col="blue", border="yellow", main="Simulierte Daten",
       xlab="Probeweise Mittelwerte", ylab="Häufigkeiten")

  qqplot(meanDataRaw, meanSim, main="Q-Q-Plot", xlab="Gemessene Mittelwerte",
  ylab="Simulierte Mittelwerte")

# Varianzen: Histogramme und Q-Q-Plots
  hist(varDataRaw, breaks = 100, col="blue", border="yellow", main="Gemessene Daten",
       xlab="Probeweise Varianzen", ylab="Häufigkeiten")

  hist(varSim, breaks = 100, col="blue", border="yellow", main="Simulierte Daten",
       xlab="Probeweise Varianzen", ylab="Häufigkeiten")

  qqplot(varDataRaw, varSim, main="Q-Q-Plot", xlab="Gemessene Varianzen",
  ylab="Simulierte Varianzen")


################################################################################
##
## 7. Korrektur der Mittelwert- und Varianzstruktur
##
##    Die probenweisen Mittelwerte und Varianzen der simulierten Daten haben keine
##    realistische Struktur. Dies wird im Folgenden korrigiert.
##
################################################################################


# Korrektur der Daten
  mExprSimFixed <- fixMeansVars(mExprSim, meanDataRaw, varDataRaw)
  meanSimFixed <- apply (mExprSimFixed, 1, mean)
  varSimFixed <- apply(mExprSimFixed, 1, var)
  qqExprSimFixed <- qqplot(x, mExprSimFixed, main="Q-Q-Plot",
  xlab="Gemessene Expressionswerte", ylab="Simulierte Expressionswerte")

# Q-Q-Plots der korrigierten Daten
```

```
  par(mfrow = c(1,3))
  qqplot(x, mExprSimFixed, main="Q-Q-Plot",xlab="Gemessene Expressionswerte",
  ylab="Simulierte Expressionswerte")
  qqplot(meanDataRaw, meanSimFixed, main="Q-Q-Plot",xlab="Gemessene Mittelwerte",
  ylab="Simulierte Mittelwerte")
  qqplot(varDataRaw, varSimFixed, main="Q-Q-Plot",xlab="Gemessene Varianzen",
  ylab="Simulierte Varianzen")

################################################################################

################################################################################
##
## ZIEL
##
## Simulation von Expressionsdaten mittels der emprischen Verteilung
## unter Berücksichtigung der Probe-Positionen der PM-Werte
##
## ÜBERBLICK:
##
## 0. Init
## 1. PM-Werte simulieren
## 2. MM-Werte simulieren
## 3. Simulation der differentiellen Expression
##    (unter Berücksichtigung der "Probe Interrogation Position")
##
################################################################################


################################################################################
##
## 0. Init
##
################################################################################

# Libraries laden
  library(affy)
  library(MASS)
  library(vsn)

  library(affyPLM)
  library(hgu133aprobe)
  library(hgu133acdf)
  library(hgu133a.db)

  setwd("I:/Cel")

# Funktionen laden
  source("I:/func_empVer.r")
  source("I:/func_fixMeansVars.r")
  source("I:/func_replaceRealData.r")
  source("I:/func_simPMDataWithDiffExpr.r")


# Einlesen der Rohdaten der Affymetrix Microarrays
  dataRaw <- ReadAffy(filenames=c(
    "1_A.CEL","2_A.CEL",
    "1_B.CEL","2_B.CEL"))
  gc()

# Ermöglicht den Aufruf einer Variablen nur mit dessen Namen.
  attach(hgu133aprobe)

# Vergibt eine fortlaufende Nummer der Interrogation Position.
  Index <- seq(along=Probe.Interrogation.Position)

# Vektor der Zuordnung von "Probe Interrogation Position" zu "Probe Set Id"
  vProbeInterPosToProbeSetId <- split(Probe.Interrogation.Position,Probe.Set.Name)

# Vektor der Zuordnung von "Probe Index" zu "Probe Set Id"
  vIndexToProbeSetId <- split(Index,Probe.Set.Name)
```

```
# Lokalisation der PM-Probes von jedem Probe Set mit fortlaufender Nummer in der Affy-ID.
  pmIndex <- unlist(indexProbes(dataRaw, "pm"))


##############################################################################
##
## 1. PM: PM-Werte mittels empirischer Verteilung simulieren
##
##############################################################################

# Vektor mit logarithmierten PM-Daten erstellen
  PMs <- log2(pm(dataRaw))
  exprLevelsPM <- as.vector(PMs)

# Daten mit empirischer Verteilung simulieren
  mExprSimPM      <- matrix(empVer(length(exprLevelsPM), exprLevelsPM, 1000), ncol=ncol(PMs))
  #hist(mExprSimPM)
  #qqExprSim  <- qqplot(exprLevelsPM, mExprSimPM)
  #abline(lmsreg(qqExprSim$x, qqExprSim$y))

# Mittelwert und Varianzen der PM-Daten berechnen
  meanDataRawPM       <- apply(PMs, 1, mean)
  varDataRawPM        <- apply(PMs, 1, var)

# Korrektur der Mittelwert- und Varianzstruktur
  mExprSimPMFixed     <- fixMeansVars(mExprSimPM, meanDataRawPM, varDataRawPM)

##############################################################################
##
## 2. MM: MM-Werte mittels empirischer Verteilung simulieren
##
##############################################################################

# Vektor mit logarithmierten MM-Daten erstellen
  MMs <- log2(mm(dataRaw))
  exprLevelsMM <- as.vector(MMs)

# Daten mit empirischer Verteilung simulieren
  mExprSimMM    <- matrix(empVer(length(exprLevelsMM), exprLevelsMM, 1000), ncol=ncol(MMs))

# Mittelwert und Varianzen der MM-Daten berechnen
  meanDataRawMM       <- apply(MMs, 1, mean)
  varDataRawMM        <- apply(MMs, 1, var)

# Korrektur der Mittelwert- und Varianzstruktur
  mExprSimMMFixed     <- fixMeansVars(mExprSimMM, meanDataRawMM, varDataRawMM)

##############################################################################
##
## 3. Simulation der differentiellen Expression
##    (unter Berücksichtigung der "Probe Interrogation Position")
##
##############################################################################

  setwd("I:/output")

# Differentielle Expression für simulierte PM-Daten simulieren
  simMat <- simPMDataWithDiffExpr(mExprSimPMFixed, vIndexToProbeSetId,
  vProbeInterPosToProbeSetId, 500, 10, c(1,2), log=TRUE)

# Reale Daten mit simulierten Daten überschreiben (Erhaltung der Array-Struktur)
  dataSim <- Data
  pm(dataSim) <- replaceRealData(pm(dataSim), simMat, vIndexToProbeSetId, pmIndex)
  mm(dataSim) <- mExprSimMMFixed

# Ausgabe
  ### write.exprs(dataSim, paste("newSimTestData_",format(Sys.time(), "%Y%m%d%H%M%S"),".txt"))
  save.image(paste("newSimTestData_",format(Sys.time(), "%Y%m%d%H%M%S"),".RData"))
```

```
################################################################################


################################################################################
##
## ZIEL
##
## Simulation von Expressionsdaten mittels einer angepassten Gamma-Verteilung
## unter Berücksichtigung der Probe-Positionen der PM-Werte
##
## ÜBERBLICK:
##
## 0. Init
## 1. PM-Werte simulieren
## 2. MM-Werte simulieren
## 3. Simulation der differentiellen Expression
##    (unter Berücksichtigung der "Probe Interrogation Position")
##
################################################################################


################################################################################
##
## 0. Init
##
################################################################################

# Libraries laden
  library(affy)
  library(MASS)
  library(vsn)

  setwd("I:/Cel")

# Funktionen laden
  source("I:/func_replaceRealData.r")
  source("I:/func_simPMDataWithDiffExpr.r")

# Daten laden
  dataRaw <- ReadAffy(filenames=c(
    # Amplifikations Cel-Files
    "1_A.CEL","2_A.CEL",
    "1_B.CEL","2_B.CEL"))
  gc()

# Ermöglicht den Aufruf einer Variablen nur mit dessen Namen.
  attach(hgu133aprobe)

# Vergibt eine fortlaufende Nummer der Interrogation Position.
  Index <- seq(along=Probe.Interrogation.Position)

# Vektor der Zuordnung von "Probe Interrogation Position" zu "Probe Set Id"
  vProbeInterPosToProbeSetId <- split(Probe.Interrogation.Position,Probe.Set.Name)

# Vektor der Zuordnung von "Probe Index" zu "Probe Set Id"
  vIndexToProbeSetId <- split(Index,Probe.Set.Name)

# Lokalisation der PM-Probes von jedem Probe Set mit fortlaufender Nummer in der Affy-ID.
  pmIndex <- unlist(indexProbes(dataRaw, "pm"))



################################################################################
##
## 1.1 PM: PM-Werte mittels angepasster Gamma-Verteilung simulieren
##
################################################################################

# Vektor mit logarthmierten PM Daten erstellen
  PMs <- log2(pm(dataRaw))
```

```
  exprLevelsPM <- as.vector(PMs)

# Nur Abweichungen vom Minimum betrachten
  exprLevelsPMSubset <- exprLevelsPM - min(exprLevelsPM) + 0.001

# Parameterberechnung für Gamma-Verteilung
  scale <- var(exprLevelsPMSubset)/mean(exprLevelsPMSubset)
  shape <- mean(exprLevelsPMSubset)/scale

# Alternative 1: Simulation mit Gamma-Verteilung auf Basis von Subset
  x <- sort(exprLevelsPMSubset)[seq(1001, length(exprLevelsPMSubset), 1000)]
  exprLevelsPMSubsetGamma <- rgamma(exprLevelsPMSubset,shape=shape, scale=scale)
  exprLevelsPMSubsetGamma <- matrix(data=exprLevelsPMSubsetGamma, nrow=nrow(PMs),ncol=ncol(PMs))
  mExprSimGamma <- qqplot(x,exprLevelsPMSubsetGamma)
  abline(lmsreg(mExprSimGamma$x, mExprSimGamma$y))

# Alternative 2: Simulation mit Gamma-Verteilung auf Basis von allen Daten
  exprLevelsPMGamma <- rgamma(exprLevelsPMSubset,shape=shape, scale=scale)
  exprLevelsPMGamma <- matrix(data=exprLevelsPMGamma, nrow=nrow(PMs),ncol=ncol(PMs))
  mExprSimGamma <- qqplot(exprLevelsPMSubset,exprLevelsPMGamma)
  abline(lmsreg(mExprSimGamma$x, mExprSimGamma$y))

##############################################################################
##
## 1.2 PM: Histogramm und Q-Q-Plot der mittels Gamma-Verteilung simulierten Daten
##
##############################################################################

  postscript("I:/Gamma_angepasst_Testdata.eps",width = 10, height = 10,horizontal=FALSE)

  par(mfrow = c(2,1))
  hist(exprLevelsPMSubsetGamma , breaks = 1000,col="blue", border="yellow",
  main=" angepasste Gamma-Verteilung",
  xlab="shape= 2.259 scale=0.8125", ylab="Häufigkeiten")
  qqplot(x,exprLevelsPMSubsetGamma, main="Q-Q-Plot",xlab="Expressionswerte",
  ylab="angepasste Gamma-Simulation")
  abline(lmsreg(mExprSimGamma$x, mExprSimGamma$y))
  dev.off()

##############################################################################
##############################################################################


##############################################################################
##
## 2.1 MM: MM-Werte mittels angepasster Gamma-Verteilung simulieren
##
##############################################################################

# Vektor mit logarithmierten MM-Daten erstellen
  MMs <- log2(mm(dataRaw))
  exprLevelsMM <- as.vector(MMs)

# Nur Abweichungen vom Minimum betrachten
  exprLevelsMMSubset <- exprLevelsMM - min(exprLevelsMM) + 0.001

# Parameterberechnung für Gamma-Verteilung
  scale <- var(exprLevelsMMSubset)/mean(exprLevelsMMSubset)
  shape <- mean(exprLevelsMMSubset)/scale

# Alternative 1: Simulation mit Gamma-Verteilung auf Basis von Subset
  x<-sort(exprLevelsMMSubset)[seq(1001, length(exprLevelsMMSubset), 1000)]
  exprLevelsMMSubsetGamma <- rgamma(exprLevelsMMSubset,shape=shape, scale=scale)
  exprLevelsMMSubsetGamma <- matrix(data=exprLevelsMMSubsetGamma, nrow=nrow(PMs)
  ,ncol=ncol(PMs))
  #mExprSimGammaMM <- qqplot(x,exprLevelsMMSubsetGamma)
  #abline(lmsreg(mExprSimGammaMM$x, mExprSimGammaMM$y))
```

```
##############################################################################
##
## 2.2 MM: Histogramm und Q-Q-Plot der mittels Gamma-Verteilung simulierten Daten
##
##############################################################################

  #postscript("I:/Gamma_angepasst_Testdata_mm.eps",width = 10, height = 10,
  #horizontal=FALSE)

  #par(mfrow = c(2,1))
  #hist(exprLevelsMMSubsetGamma, breaks = 100,col="blue", border="yellow",
  #main=" angepasste Gamma-Verteilung",
  #xlab=" shape= 2.786 scale= 0.545", ylab="Häufigkeiten")
  #qqplot(exprLevelsMMSubset,exprLevelsMMSubsetGamma, main="Q-Q-Plot",
  #xlab="Expressionswerte", ylab="angepasste Gamma-Simulation")
  #abline(lmsreg(mExprSimGammaMM$x, mExprSimGammaMM$y))
  #dev.off()


##############################################################################
##
## 3. Simulation der differentiellen Expression
##    (unter Berücksichtigung der "Probe Interrogation Position")
##
##############################################################################

  setwd("I:/outputgamma")

# Simulation der differentiellen Expression unter Berücksichtigung der
# "Probe Interrogation Position"
  simMat <- simPMDataWithDiffExpr(exprLevelsPMSubsetGamma, vIndexToProbeSetId,
  vProbeInterPosToProbeSetId, 500, 10, c(1,2), log=TRUE)

# Reale Daten mit simulierten Daten überschreiben (Erhaltung der Array-Struktur)
  dataSim <- dataRaw
  pm(dataSim) <- replaceRealData(pm(dataSim), simMat, vIndexToProbeSetId, pmIndex)
  mm(dataSim) <- exprLevelsMMSubsetGamma

# Ausgabe
  #write.exprs(dataSim, paste("newSimTestData_",format(Sys.time(), "%Y%m%d%H%M%S"),".txt"))
  save.image(paste("newSimTestData_gamma_",format(Sys.time(), "%Y%m%d%H%M%S"),".RData"))


##############################################################################

##############################################################################
##
## ZIEL
##
## Simulation von Expressionsdaten mittels eines gemischten linearen Modells
## unter Berücksichtigung der Probe-Positionen der PM-Werte
##
## ÜBERBLICK:
##
## 0. Init
## 1. PM-Werte simulieren
## 2. MM-Werte simulieren
## 3. Simulation der differentiellen Expression
##    (unter Berücksichtigung der "Probe Interrogation Position")
##
##############################################################################


##############################################################################
##
## 0. Init
##
##############################################################################

# Libraries laden
```

```
  library(affy)
  library(affyPLM)
  library(hgu133aprobe)
  library(hgu133acdf)
  library(hgu133a.db)
  library(vsn)
  library(lme4)

  setwd("I:/Cel")

# Pfade definieren
  tPathOutput <- "/tmp/output/"

# Daten einlesen: Einlesen der Cel-Files in ein AffyBatch
  dataRaw <- read.affybatch(filenames=c(
    "1_A.CEL","2_A.CEL",
    "1_B.CEL","2_B.CEL"))

# Ermöglicht den Aufruf einer Variablen nur mit dessen Namen.
  attach(hgu133aprobe)

# Vergibt eine fortlaufende Nummer der Interrogation Position.
  Index <- seq(along=Probe.Interrogation.Position)

# Vektor der Zuordnung von "Probe Interrogation Position" zu "Probe Set Id"
  vProbeInterPosToProbeSetId <- split(Probe.Interrogation.Position,Probe.Set.Name)

# Vektor der Zuordnung von "Probe Index" zu "Probe Set Id"
  vIndexToProbeSetId <- split(Index,Probe.Set.Name)

# Lokalisation der PM-Probes von jedem Probe Set mit fortlaufender Nummer in der Affy-ID.
  pmIndex <- unlist(indexProbes(dataRaw, "pm"))

##############################################################################
##
## 1. PM-Werte mit linear gemischtem Modell simulieren
##
##############################################################################

# Dataframe mit Variablen erstellen, die in lineares Modell eingehen sollen

  ncols          <- length(pm(dataRaw)[1,])
  nrows          <- length(pm(dataRaw)[,1])
# Spaltenbeschriftung
  az <- c('a', 'b', 'c', 'd')

  PMData         <- as.vector(log(pm(dataRaw)))
  namen          <- rep(Probe.Set.Name, ncols)
  namenMitIndex  <- rep(names(pmIndex), ncols)
  interpos       <- as.vector(unlist(vProbeInterPosToProbeSetId))
  chip           <- rep(az[1:ncols], rep(nrows,ncols))

  dfPMData <- data.frame(PMData, namen, namenMitIndex, interpos, chip)


# Model fitting (Linear Mixed Model)

# Dabei entspricht die Variable PMData allen PM-Expressionswerten. Hinter der |
# ist zunächst der feste Effekt definiert. In diesem Fall setzt er sich aus pos,
# der Position im Transkript, und chip, den verschiedenen Microarrayexperimenten,
# zusammen. Es folgt der Random-Effekt, der aus einem Ausdruck eines linearen
# Modells und einem Gruppenfaktor besteht, der durch | getrennt ist. Dabei
# entspricht pos dem linearen Modell und namen (die einzelnen Probeset-IDs)
# dem Gruppenfaktor. Der Random-Effekt ist ein Ausdruck für ein lineares Modell
#  bedingt auf den Gruppenfaktor. Für die MM-Expressionswerte wird ein analoges
# Modell angepasst.

  modelPM <- lmer(PMData~interpos+chip+(interpos |chip/namen),data=dfPMData )
  #summary(modelPM)
  #plot(modelPM)
```

```
# Daten mit Modell simulieren

  dataSimPM <- simulate(modelPM)


###############################################################################
##
## 2. MM-Werte mit linear gemischtem Modell simulieren
##
###############################################################################

# Dataframe mit Variablen erstellen, die in lineares Modell eingehen sollen

  nColsMM          <- length(mm(dataRaw)[1,])
  nRowsMM          <- length(mm(dataRaw)[,1])
  # Spaltenbeschriftung
  az <- c('a', 'b', 'c', 'd')

  MMData        <- as.vector(log(mm(dataRaw)))
  namen          <- rep(Probe.Set.Name, nColsMM)
  namenMitIndex <- rep(names(pmIndex), nColsMM)
  interpos       <- as.vector(unlist(vProbeInterPosToProbeSetId))
  chip           <- rep(az[1:ncols], rep(nRowsMM,nColsMM))

  dfMMData <- data.frame(MMData, namen, namenMitIndex, interpos, chip)


# siehe modelPM

  modelMM <- lmer(MMData~interpos+chip+(interpos |chip/namen),data=dfMMData )
  summary(modelMM)
  #plot(modelMM)

# Daten mit Modell simulieren

  dataSimMM <- simulate(modelMM)


###############################################################################
##
## 3. Differentielle Expression für PM-Werte simulieren
##
##    Es werden zufällig ProbeSets ausgewählt, die differentiell sein sollen.
##    Diese werden mit einem zufälligen Faktor (hier zwischen 2 und 10)
##    multipliziert.
##
###############################################################################

# Anzahl der differentiellen ProbeSets festlegen
  nDiffExprGenes <- 500

  myrand <- sample(1000, 1)
  filenameout = paste(tPathOutput,"LME4_sampleDEGenes_",nDiffExprGenes,"_",
  format(Sys.time(), "%Y%m%d%H%M"),"_",myrand,".txt", sep="")

# Dataframes erstellen
  dfSimPM <- data.frame(exp(dataSimPM), namen, namenMitIndex, interpos, chip)
  names(dfSimPM) <- c("exprSim", "namen", "namenMitIndex", "interpos", "chip")

  dfSimMM <- data.frame(exp(dataSimMM), namen, namenMitIndex, interpos, chip)
  names(dfSimMM) <- c("exprSimMM", "namen", "namenMitIndex", "interpos", "chip")

# Zufällig differentielle ProbeSets auswählen
  namenRandom    <- sample(Probe.Set.Name, nDiffExprGenes)

# Vektor der "Multiplikatoren" mit 1 initialisieren
  vMultiplikatoren <- rep(1, nrow(dfSimPM))

# vOut merkt sich, welche ProbeSets differentiell sind und wie stark
```

```
  vOut <- c()

  for (myname in namenRandom)
  {
    # Zufällig Multiplikatoren festlegen (hier zwischen 2 und 10) und (für
    # die Spalten "a" und "b"
            vIndex <- which(dfSimPM$namen == myname & dfSimPM$chip %in% c("a", "b") )
            iFactor <- sample(c(2:10),1)
            vMultiplikatoren[vIndex] <- rep(iFactor, length(vIndex))
            vOut <- rbind(vOut, c(myname, iFactor))
  }

# Mittels Multiplikatoren "differentielle Expression" erzeugen
  dfSimPM$exprSim <- dfSimPM$exprSim * vMultiplikatoren

# Ausgabe von vOut
  colnames(vOut) <- c("probe_set_id", "Factor")
  write.table(vOut, filenameout, row.names = FALSE)

# Reale Daten mit simulierten Daten überschreiben (Erhaltung der Array-Struktur)
  dataSim <- dataRaw

        # PM-Werte mit simulierten Werten ersetzen
  pm(dataSim) <- dfSimPM$exprSim

        # MM-Werte mit simulierten Werten ersetzen
  mm(dataSim) <- dfSimMM$exprSimMM

# Ausgabe
  save.image(paste(tPathOutput,"LME4_newSimTestData_",format(Sys.time(),
  "%Y%m%d%H%M%S"),"_",myrand,".RData"))
###############################################################################

###############################################################################
##
## empVer
##
## Ziel: Zufallszahlen gemäß empirischer Verteilung erzeugen
##
## Binning-Ansatz mit gleich breiten Bins
##   - Wertebereich der Expressionswerte in n "Bins" unterteilen
##   - Gewichte der Bins ergeben Unterteilung des [0,1]-Intervalls
##   - Mittels uniform auf [0,1] verteilter Zufallszahlen und "Rückprojektion"
##     erhält man die empirisch verteilte Zufallszahl
##
## n:       Wie viele Zufallswerte sollen generiert werden?
## values:  Vektor der Daten gemäß dem die empirische Verteilung bestimmt wird
## nbins:   Anzahl der Bins
##
## res:     Rückgabevektor der generierten Zufallszahlen
##
###############################################################################

empVer <- function(n, values, nbins=1000)
{
  # Rückgabevektor res: wird die generierten Zufallszahlen enthalten (alles auf 0 setzen)
    res <- rep(0, n)

  # Generieren von Bins gleicher Ausdehnung
    h <- hist(values, nclass=nbins, plot=FALSE)

  # Anzahl der realen Daten
    nvalues <- length(values)

  # Anzahl der Bins
    nbins <- length(h$counts)

  # Für jedes Bin wird das "Gewicht" berechnet, indem der Anteil der enthaltenen
  # Datenpunkte bestimmt wird. ("Anzahl Datenpunkte in Bin b" / "Gesamtanzahl Datenpunkte")
```

```
    # breaks: Unterteilung des [0,1] Intervalls gemäß der Gewichte
    breaks <- rep(0, nbins+1)
    for (b in 1:nbins)
    {
      breaks[b+1] <- breaks[b] + h$counts[b]/nvalues
     }

  # Generieren von n Zufallszahlen aus dem [0,1] Intervall
    z <- runif(n)

  # Für jede dieser Zufallszahlen wird von dem [0,1] Wert auf das
  # Intervall der empirischen Daten zurückprojeziert
    for (i in 1:n)
    {
      # Entsprechendes Bin b ausfindig machen, dass mit Zufallszahl assoziiert wird
      for (b in 1:(nbins+1))
      {
        if (breaks[b] > z[i]) break
      }

      # Lage von z[i] innerhalb des Bins bestimmen
      # w liegt zwischen 0 und 1.
      w <- (z[i]-breaks[b-1])/(breaks[b]-breaks[b-1])

      # Mittels w wird zwischen den realen Werten an den Bin-Grenzen linear gewichtet
      res[i] <- (1-w)*h$breaks[b-1] + w*h$breaks[b]
    }

    # return
    return(res)
}

##############################################################################

##############################################################################
##
## fixMeansVars
##
## Funktion zum Korrigieren der Mittelwert- und Varianzstruktur
##
## Zeilenweise werden Mittelwerte und Varianzen der Datenmatrix an vorgegebene
## Mittelwerte und Varianzen angepasst.
##
##
## matrix: Datenmatrix (hier: simulierte Probe-Daten)
## means:  Vektor der Mittelwerte
## vars:   Vektor der Varianzen
##
##############################################################################

fixMeansVars <- function(matrix, means, vars)
{
  fixGene <- function(v, n)
  {
    return((v[1:n] - mean(v[1:n]))/sd(v[1:n])*sqrt(v[n+2]) + v[n+1])
  }

  return(t(apply(cbind(matrix,
                       empVer(nrow(matrix),means),
                       empVer(nrow(matrix),vars)
                       ),
                 1, fixGene, ncol(matrix))))
}

##############################################################################

##############################################################################
##
## Funktion replaceRealData
##
```

```
## Um die Datenstruktur des Microarrays zu erhalten, werden die Rohdaten durch die
## simulierten Werte unter Berücksichtigung der ProbeSetIds beziehungsweise der Probe
## Indizes überschrieben.
##
##   probeMat:            Matrix mit Probe-Werten (PM oder MM)
##   simMat:              Enthält simulierte Probe-Werte
##   vIndexToProbeSetId:  Vektor der Zuordnung von "Probe Index" zu "Probe Set Id"
##   probeIndex:          Lokalisation der Probes von jedem Probe Set mit
##                        fortlaufender Nummer in der Affy-ID.
##
##   Gibt Matrix mit den überschriebenen Probe-Werten zurück
##
#############################################################################

replaceRealData <- function(probeMat, simMat, vIndexToProbeSetId, probeIndex)
{
  # Zuordnung von ProbeSetId zu "realem" Index
    t_all <- cbind(as.vector(unlist(vIndexToProbeSetId)), as.vector(probeIndex))

  # Daten entsprechend der Zuordnung überschreiben
    probeMat[as.character(t_all[,2]), ] <-  simMat[t_all[,1],]

  # return
    return(probeMat)
}

#############################################################################

#############################################################################
##
## Funktion simPMDataWithDiffExpr
##
## vExpr ist eine Matrix mit simulierten Expressionswerten, die noch keine
## "differentielle Expression" enthält und keine Abhängigkeit von der
## "Probe Interrogation Position" besitzt.
##
## Bezüglich der Simulation der differentiellen Expression ist Grundidee,
## eine "Umsortierung" von vExpr vorzunehmen und dabei zu Beginn bei einigen
## ProbeSets für eine Auswahl von größeren Werten zu sorgen. Durch Betrachtung bzw.
## Unterteilung in Quantile werden "Gruppen" verschieden starker differentieller
## Expression definiert, mit deren Hilfe "differentielle Expression" simuliert
## werden soll.
## Dazu werden zufällig ProbeSets bestimmt, die "differentiell" sein sollen.
## Für diese wird zufällig eine "Gruppe" bestimmt und solange Werte aus vExpr
## (mit Zurücklegen) gezogen bis diese "stark" genug sind. Anschliessend
## werden die "nicht differentiellen" Werte (ohne Zurücklegen) simuliert.
##
## Die Abhängigkeit der Daten von der "Probe Interrogation Position" wird durch
## eine Sortierung der Werte pro ProbeSets simuliert.
##
##   vExpr                        (simulierte) Expressionsdaten
##   vIndexToProbeSetId           Vektor der Zuordnung von "Probe Index" zu
##                                "Probe Set Id"
##   vProbeInterPosToProbeSetId   Vektor der Zuordnung von "Probe Interrogation
##                                Position" zu "Probe Set Id"
##   nDiffExprGenes               Anzahl der als differenziell zu simulierenden Gene
##   nDiffExprGroups              Anzahl der Gruppen (verschieden starke differentielle Expression)
##   vColDiffExpr                 Für welche Spalten der Expressionsdatenmatrix
##                                soll differentielle Expression simuliert werden
##
## Rückgabe:
##   vReturn      Expressionsmatrix mit "simulierter" differentieller Expression
##
#############################################################################

simPMDataWithDiffExpr <- function(vExpr, vIndexToProbeSetId,
vProbeInterPosToProbeSetId, nDiffExprGenes = 100, nDiffExprGroups = 10, vColDiffExpr = c(1))
{
  # Dateiname für Ausgabe definieren
    myrand <- sample(1000, 1)
```

```
  filenameout = paste("sampleDEGenes_",nDiffExprGenes,"_",nDiffExprGroups,"_",
  format(Sys.time(), "%Y%m%d%H%M"),"_",myrand,".txt", sep="")

# Rückgabe Matrix initialisieren
  vReturn <- vExpr
  vReturn[,] = NA

  nCols <- length(vReturn[1,])

# ProbeSetIds
  vPSIds <- names(vProbeInterPosToProbeSetId)

# Quantile werden benutzt, um "differential Expression" zu simulieren.
# 0%-Quantilwert und 100%-Quantilwert machen keinen Sinn, daher + 1 + 1,
# um nDiffExprGroups brauchbare Werte zu erhalten.
  nDEQuantiles <- as.vector(quantile(vExpr, seq(0, 1, length.out=nDiffExprGroups + 1 + 1)))

# 0%-Quantilwert entfernen
  nDEQuantiles <- nDEQuantiles[-1]

# 100%-Quantilwert enfernen
  nDEQuantiles <- nDEQuantiles[-length(nDEQuantiles)]

# ProbeSetIds (zufällig), die "differentiell" sein werden
  vPSIdsDiffExpr <- sample(vPSIds, nDiffExprGenes)

#Output Variablen initialisieren
  listOut <- list(vPSIds     = c(),
                  vCol       = c(),
                  vGroupIdx  = c(),
                  vGroupQMin = c())

# Für jede der als differentiell zu simulierenden ProbeSetIds ...
  for (psid in vPSIdsDiffExpr)
  {
    # Quantil bzw. Stärke der differentiellen Expression zufällig aussuchen
      groupIdx <- sample(1:nDiffExprGroups,1)
      groupQMin <- nDEQuantiles[groupIdx]

    # Alle Spalten durchlaufen
      for (col in 1:nCols)
      {
        # Wenn es sich um eine Spalte handelt, für die differentielle
        # Expression simuliert werden soll ...
          if (col %in% vColDiffExpr)
          {
            # Vektor der Indexe zu dieser ProbeSetId
              probeIndex     <- vIndexToProbeSetId[[psid]]

            # Vektor der Interrogation Positionen zu dieser ProbeSetId
              probeInterPos  <- vProbeInterPosToProbeSetId[[psid]]

            # Anzahl
              nprobeInterPos <- length(probeInterPos)

            # Variable initiieren
              mySample = c()

            # Solange noch nicht genügend (valide) Werte gefunden wurden ...
              while (length(mySample) < nprobeInterPos)
              {
                # Variable initiieren
                  validSample <- c()

                # Solange kein gültiger Wert zufällig gefunden wurde ...
                  while (0 == length(validSample))
                  {
                    # Zufällig einen Index aus den Daten ziehen ...
                      idx <- sample(1:length(vExpr), 1, replace=TRUE)
```

```
                    # ... und prüfen, ob dazugehöriger Expressionswerte "stark"
                    # genug ist
                      if (vExpr[idx] > groupQMin)
                      {
                          validSample <- vExpr[idx]

                        # Wert aus vExpr entfernen ...
                          vExpr <- vExpr[-idx]

                        # ... und zu bisheriger Stichprobe hinzufügen.
                          mySample <- c(mySample, validSample)
                      }
                  }
              }

            # Merke groupIdx, groupQMin (nur eine Zeile erzeugen, daher die
            # if Abfrage auf [1])
              if (col == vColDiffExpr[1])
              {
                listOut$vPSIds     <- c(listOut$vPSIds, psid)
                listOut$vCol       <- c(listOut$vCol, col)
                listOut$vGroupIdx  <- c(listOut$vGroupIdx, groupIdx)
                listOut$vGroupQMin <- c(listOut$vGroupQMin, groupQMin)
              }

            # Stichprobe absteigend sortieren
              mySampleSortedDesc <- sort(unlist(mySample), decreasing=F)

            # Uns interessieren die Indizes der sortierten probeInterPositionen
            # (probeInterPosSorted$ix), ...
              probeInterPosSorted <- sort(probeInterPos, index.return=T)

            # ... um die gesampleten Werte sortiert an die Positionen (probeIndex)
            #  zu schreiben
              vReturn[probeIndex, col] <- mySampleSortedDesc[probeInterPosSorted$ix]
          }
      }
  }

# Ausgabe der "Differentiellen" in Datei
  vOut <- cbind(listOut$vPSIds, listOut$vCol, listOut$vGroupIdx, listOut$vGroupQMin)
  colnames(vOut) <- c("PSID", "Col", "groupIdx", "groupQMin")
  write.table(vOut, filenameout, row.names = FALSE)


############## SAMPLE NORMAL ################

# Für die verbliebenen (nicht differentiellen) ProbeSets ebenfalls sortierte
# Werte gemäß der Interrogation Positionen erzeugt

# Für alle ProbeSetIds ...
  for (psid in vPSIds)
  {
    # ... für alle Spalten ...
      for (col in 1:nCols)
      {
        # ... wobei es sich nicht um einen bereits als differentiell
        # simuliertes ProbeSet handeln darf ...
          if ( !(col %in% vColDiffExpr && psid %in% vPSIdsDiffExpr ) )
          {
            # Vektor der Indexe zu dieser ProbeSetId
              probeIndex     <- vIndexToProbeSetId[[psid]]

            # Vektor der Interrogation Positionen zu dieser ProbeSetId
              probeInterPos  <- vProbeInterPosToProbeSetId[[psid]]

            # Anzahl
              nprobeInterPos <- length(probeInterPos)

            # Sample nehmen ...
```

```
                mySample <- vExpr[1:nprobeInterPos]

              # ... und aus vExpr entfernen
                vExpr <- vExpr[-c(1:nprobeInterPos)]

              # Stichprobe absteigend sortieren
                mySampleSortedDesc <- sort(mySample, decreasing=F)

              # Uns interessieren die Indizes der sortierten probeInterPositionen
              # (probeInterPosSorted$ix), ...
                probeInterPosSorted <- sort(probeInterPos, index.return=T)

              # ... um die gesampleten Werte sortiert an die Positionen
              # (probeIndex) zu schreiben
                vReturn[probeIndex, col] <- mySampleSortedDesc[probeInterPosSorted$ix]
            }
        }
    }

  # Return
    vReturn
}

###############################################################################
```

6.3 Reanalyse der *Microarray*-Daten von Küppers et al.

Tabelle 6.1: Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003]

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
38826_at	-2,0	9,03E-05	Sep6
33613_at	-4,9	0,005704503	—
330_s_at	-3,3	1,45E-06	—
292_s_at	-2,4	2,65E-05	—
677_s_at	-2,3	0,010776494	ACP5
32755_at	2,2	0,006218446	ACTA2
39329_at	2,3	0,001287208	ACTN1
40585_at	-2,1	0,003035617	ADCY7
34792_at	2,9	3,16E-07	AHCYL1
41302_at	3,7	6,36E-08	AHCYL1
37027_at	3,2	0,007522676	AHNAK

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
34439_at	-2,5	0,029729785	AIM2
37099_at	-2,1	0,023168689	ALOX5AP
40083_at	-2,4	6,68E-05	ALS4
40347_at	2,2	0,001543853	ANP32E
40348_s_at	2,2	0,000216412	ANP32E
769_s_at	2,1	0,008118773	ANXA2
37670_at	2,1	0,000386912	ANXA7
31873_at	2,0	3,31E-07	ARD1A
36585_at	2,0	4,31E-06	ARF4
38278_at	2,0	0,004533232	ARID5A
36872_at	2,4	5,31E-06	ARPP-19
37114_at	2,3	2,65E-05	ATBF1
39158_at	3,3	0,000212624	ATF5
32563_at	2,1	4,63E-06	ATP1B3
39791_at	2,0	0,000943494	ATP2A2
40913_at	2,4	1,86E-05	ATP2B4
36107_at	2,2	0,00017199	ATP5J
37616_at	2,3	1,94E-06	AUH
37821_at	2,1	0,000139578	BCAS1
38201_at	2,1	0,000529496	BCAT1
41355_at	-2,1	1,90E-05	BCL11A
40091_at	-3,7	0,00613672	BCL6
40790_at	2,8	0,012034199	BHLHB2
32806_at	-2,5	9,78E-05	BZRP
38411_at	-2,1	0,044987278	C11orf32
40766_at	2,6	2,64E-06	C4A /// C4B
1421_at	2,0	3,60E-07	CAMK4

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
40125_at	2,4	6,66E-07	CANX
37001_at	3,2	0,000252454	CAPN2
40408_at	2,0	1,54E-09	CARS
1405_i_at	3,7	9,85E-05	CCL5
1403_s_at	4,6	1,93E-05	CCL5
36650_at	3,6	0,008098457	CCND2
1983_at	4,2	3,08E-06	CCND2
1913_at	-2,1	0,000196773	CCNG2
33553_r_at	-2,7	0,018803575	CCR6
1097_s_at	4,6	0,001823245	CCR7
38720_at	2,2	5,69E-08	CCT7
40323_at	-2,2	0,026683671	CD38
37177_at	2,1	0,000150917	CD58
37536_at	-2,5	0,001584485	CD83
41535_at	2,3	6,07E-05	CDK2AP1
39219_at	2,0	1,09E-05	CEBPG
1868_g_at	2,0	0,000105446	CFLAR
1867_at	2,2	1,64E-06	CFLAR
31891_at	-2,9	0,010693195	CHI3L2
37347_at	2,4	0,002639544	CKS1B
33891_at	-2,1	0,006934165	CLIC4
32833_at	-2,4	5,98E-06	CLK1
32523_at	2,3	3,10E-05	CLTB
36780_at	2,6	0,034149625	CLU
41755_at	-2,1	0,00052794	COBLL1
38976_at	-2,7	6,16E-05	CORO1A
34723_at	2,1	0,002411622	COX11

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
41223_at	2,3	0,000486686	COX5A
36687_at	2,1	0,000140908	COX7B
37999_at	2,2	2,54E-05	CPOX
36874_at	-5,8	0,000176845	CR2
41202_s_at	-2,4	0,000166195	CTDSP2
2085_s_at	2,2	5,62E-07	CTNNA1
133_at	3,6	4,69E-07	CTSC
37021_at	-2,1	0,012297323	CTSH
823_at	3,0	9,68E-06	CX3CL1
38459_g_at	2,0	1,43E-06	CYB5
35807_at	-2,3	1,51E-06	CYBA
986_at	3,0	1,99E-05	CYP19A1
987_g_at	5,8	1,08E-05	CYP19A1
37199_at	2,5	2,05E-05	D2LIC
38989_at	-2,0	0,037518089	DC12
630_at	2,0	0,007239166	DCTD
631_g_at	2,7	0,002564002	DCTD
39814_s_at	2,7	0,000150917	DHRS7
38764_at	-2,1	5,25E-05	DICER1
34183_at	-4,3	9,01E-07	DKFZP434C171
39118_at	2,1	0,000415951	DNAJA1
41233_at	2,1	0,003932422	DNAJB6
36135_at	2,1	1,54E-06	EBNA1BP2
35174_i_at	2,6	0,012631045	EEF1A2
34278_at	2,1	0,000386164	EIF1AX
39182_at	2,8	0,04037469	EMP3
32964_at	2,4	5,22E-09	EMR1

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
32585_at	3,2	1,28E-07	EPB41L2
34331_at	2,1	3,34E-05	EPHB1
37305_at	2,4	0,013616336	EZH2
37838_at	2,3	5,21E-06	F12
37907_at	2,2	1,39E-05	F8A1
37281_at	2,1	0,000309189	FAM38A
37643_at	2,3	6,74E-05	FAS
1440_s_at	3,0	2,14E-05	FAS
32535_at	2,2	0,000631135	FBN1
34663_at	-2,1	0,014895848	FCGR2B
37325_at	2,0	9,08E-05	FDPS
40336_at	2,2	7,65E-07	FDXR
41583_at	2,1	0,031542932	FEN1
32749_s_at	2,2	0,014867026	FLNA
37506_at	2,2	4,16E-06	FNBP3
1107_s_at	2,6	0,003470371	G1P2
41839_at	2,5	0,000165112	GAS1
36596_r_at	-2,3	3,08E-05	GATM
37357_at	2,0	0,000452534	GCSH
32626_at	2,4	4,88E-05	GFPT1
37263_at	2,2	1,11E-05	GGH
36201_at	2,4	6,56E-07	GLO1
40764_at	2,6	0,000290204	GOT2
39122_at	2,3	2,90E-07	GPI
33931_at	2,1	2,07E-06	GPX4
869_at	2,0	6,66E-05	GTF2A2
319_g_at	2,2	0,005542761	H1FX

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
39337_at	2,2	0,012378495	H2AFZ
37483_at	-2,1	3,59E-06	HDAC9
36446_s_at	2,3	0,000255633	HDGF
32773_at	-2,3	0,020179267	HLA-DQA1
36878_f_at	-2,3	0,004007902	HLA-DQB1
36773_f_at	-2,2	0,002823277	HLA-DQB1
33261_at	-2,2	0,023801313	HLA-DRB3
32321_at	-3,2	0,00380392	HLA-E
38094_at	2,1	0,000181026	HNRPAB
41259_at	2,1	1,64E-07	HSPC111
32316_s_at	2,0	5,79E-06	HSPCA
37720_at	2,3	2,85E-06	HSPD1
39353_at	2,6	2,19E-06	HSPE1
402_s_at	-3,0	0,001276797	ICAM3
1237_at	3,4	1,42E-05	IER3
1456_s_at	-2,2	0,000101712	IFI16
1038_s_at	-2,8	0,008121674	IFNGR1
33499_s_at	-4,0	0,003683264	IGHA1
33500_i_at	-3,9	0,008814301	IGHA1
33501_r_at	-3,0	0,003436516	IGHA1
359_at	2,5	3,45E-07	IL13RA1
38488_s_at	2,2	3,50E-05	IL15
1368_at	2,0	0,000107121	IL1R1
41847_at	-2,7	0,014709075	IL24
38299_at	4,4	0,000379965	IL6
35303_at	2,1	0,004861785	INSIG1
478_g_at	2,1	0,011639543	IRF5

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
33304_at	-3,1	0,000581833	ISG20
37272_at	-2,0	0,000171655	ITPKB
182_at	-2,2	0,000147327	ITPR3
37343_at	-2,0	0,000301615	ITPR3
34877_at	-2,4	1,55E-06	JAK1
41250_at	2,0	1,80E-05	JTV1
2049_s_at	2,5	4,09E-06	JUNB
38116_at	2,4	0,034934874	KIAA0101
41191_at	2,3	0,007088672	KIAA0992
34563_at	2,2	0,020769665	KIF14
40407_at	2,1	0,045598519	KPNA2
41485_at	2,1	0,000307557	LDHA
33819_at	2,8	3,21E-05	LDHB
36021_at	2,1	0,003051571	LEF1
35367_at	2,3	0,021532018	LGALS3
36023_at	2,1	5,22E-07	LOC440123
34800_at	-2,3	0,00019544	LRIG1
41754_at	2,2	2,06E-06	LRPPRC
33956_at	-2,1	0,000315297	LY96
1402_at	-2,6	2,31E-05	LYN
32616_at	-2,6	2,54E-05	LYN
2024_s_at	-2,4	2,42E-06	LYN
37282_at	2,4	0,000497701	MAD2L1
32426_f_at	2,1	0,046691065	MAGEA1
36302_f_at	2,4	0,035601397	MAGEA4
35097_at	3,7	7,32E-06	MAGEB2
41772_at	2,1	0,005293662	MAOA

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
36926_at	2,1	0,000141148	MAPK6
1637_at	2,4	5,78E-07	MAPKAPK3
36174_at	-3,4	0,001160225	MARCKSL1
39342_at	2,0	2,23E-05	MARS
34306_at	-2,3	0,000155015	MBNL1
409_at	2,7	3,67E-07	MIB1
38251_at	2,1	0,001276827	MLC1SA
36941_at	5,2	7,08E-06	MLLT11
40861_at	2,2	1,82E-05	MORF4L2
673_at	2,2	2,55E-07	MTHFD1
2042_s_at	2,2	0,000306531	MYB
37941_at	2,3	0,003181161	MYBPC2
38803_at	2,8	5,12E-06	NCALD
41038_at	2,1	0,002721635	NCF2
33381_at	-2,1	0,000246308	NCOA3
35297_at	2,2	6,75E-05	NDUFAB1
38060_at	2,2	6,08E-07	NDUFS5
37544_at	4,2	2,53E-07	NFIL3
39088_at	2,1	6,28E-06	NIFIE14
39073_at	2,4	5,12E-06	NME1
1985_s_at	2,4	1,99E-05	NME1
41322_s_at	2,3	2,36E-07	NOLA2
36880_at	2,1	1,63E-05	NQO2
41743_i_at	2,2	0,000160324	OPTN
41742_s_at	2,2	0,000334817	OPTN
358_at	-2,0	0,030631579	P2RY10
36413_at	-2,0	0,02881581	P2RY10

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
34390_at	2,8	0,001101601	P4HA2
39056_at	2,2	0,000207648	PAICS
41562_at	2,1	0,007305814	PCGF4
37576_at	3,0	0,003533582	PCP4
39696_at	2,3	0,005628088	PEG10
38839_at	2,2	0,000334591	PFN2
38840_s_at	2,5	0,000196735	PFN2
36502_at	-3,6	3,67E-06	PFTK1
41221_at	2,0	3,25E-05	PGAM1
41715_at	-2,6	0,000152025	PIK3C2B
36287_at	-2,1	0,001038173	PIK3CG
35938_at	2,1	0,001865206	PLA2G4A
33452_at	2,5	1,30E-05	PLAT
37326_at	2,3	0,005293662	PLP2
35631_at	2,3	1,22E-05	POLR2H
1248_at	2,3	8,76E-06	POLR2H
35841_at	2,1	5,52E-05	POLR2L
503_at	2,1	5,35E-05	POLR2L
39729_at	-3,0	0,020064152	PRDX2
36631_at	2,0	0,004934006	PRDX3
798_at	2,2	9,50E-05	PRIM1
38042_at	2,6	1,91E-07	PRKCA
41579_s_at	2,1	0,000142168	PRKCI
33247_at	2,1	0,000322764	PSMD14
40276_at	2,1	1,08E-05	PSMD7
36008_at	2,0	5,77E-05	PTP4A3
35342_at	-2,1	0,000293669	PTPLB

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
32932_at	2,3	0,000416168	PTPN20
36808_at	-2,3	5,20E-06	PTPN22
32070_at	-5,9	6,32E-08	PTPRCAP
36160_s_at	2,1	0,00067315	PTPRN2
34446_at	-2,2	0,002519709	RABGAP1L
2050_s_at	2,0	0,003916309	RAC1
32736_at	-2,5	1,74E-06	RAC2
32593_at	-2,4	0,002887772	RAFTLIN
35848_at	-2,5	0,000116586	RAI17
228_at	2,2	6,73E-05	RALB
32776_at	3,0	3,43E-05	RALB
40556_at	2,0	8,00E-08	RCN1
1055_g_at	2,0	0,001980006	RFC4
35459_at	-7,0	0,001707008	RGS13
36550_at	2,5	0,002485686	RIN2
34660_at	-3,3	0,028301096	RNASE6
39150_at	2,8	1,11E-05	RNF11
32963_s_at	2,0	0,003040823	RRAGD
40992_s_at	2,1	2,90E-07	SAP30
245_at	-4,4	0,006783909	SELL
34789_at	3,1	4,92E-06	SERPINB6
38147_at	2,8	3,16E-07	SH2D1A
37485_at	2,5	1,07E-06	SLC27A2
38797_at	3,1	1,68E-07	SLC39A14
40961_at	2,6	9,85E-06	SMARCA2
38679_g_at	2,2	5,66E-05	SNRPE
37337_at	2,5	4,92E-05	SNRPG

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
36620_at	2,4	9,87E-05	SOD1
32140_at	-2,4	0,02331518	SORL1
37353_g_at	-2,1	0,000850618	SP100
40039_g_at	2,0	2,00E-05	ST7
32860_g_at	2,1	0,03338465	STAT1
33339_g_at	2,2	0,017370486	STAT1
AFFX-HUMISGF3A_3_at	2,3	0,02696884	STAT1
33338_at	2,4	0,009449233	STAT1
38823_s_at	-2,2	0,000243039	STK17A
40419_at	2,0	0,000320749	STOM
34380_at	2,2	1,82E-05	STOML2
38473_at	2,4	5,04E-07	TARS
39318_at	-9,4	9,71E-05	TCL1A
37324_at	2,1	0,000444094	TFRC
41057_at	2,1	3,68E-06	THEM2
41058_g_at	2,7	3,32E-06	THEM2
32830_g_at	2,1	2,16E-06	TIMM17A
36655_at	2,5	6,42E-05	TJP2
40578_s_at	2,3	1,59E-05	TMOD1
38578_at	-4,8	0,010890314	TNFRSF7
34054_at	2,6	0,000405324	TNFSF7
35000_at	2,0	0,0013347	TNFSF9
34003_at	2,0	0,000147978	TPI1
32314_g_at	3,7	0,0031332	TPM2
717_at	-2,5	4,63E-05	TRIB2
40113_at	-2,4	2,65E-05	TRIB2

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 299 Probesets nach Reanalyse des Datensatzes von Küppers et al. [Küppers et al., 2003] – Fortsetzung

Affymetrix ID	cHL-Linien/B-Zellen FC	cHL-Linien/B-Zellen FDR	Gensymbol
36591_at	-4,1	1,49E-06	TUBA1
38350_f_at	2,4	0,002641117	TUBA2
37899_at	2,1	0,011801797	TYMS
1505_at	2,1	0,02658974	TYMS
33781_s_at	2,0	4,99E-06	UBE2M
40619_at	2,1	0,016391975	UBE2S
894_g_at	2,2	0,006708717	UBE2S
893_at	2,2	0,020936149	UBE2S
35336_at	-2,2	0,000136819	UNC84B
32715_at	-2,0	0,002682585	VAMP8
40414_at	2,1	9,76E-08	VARS
171_at	2,1	7,04E-07	VBP1
31608_g_at	2,1	9,52E-06	VDAC1
34091_s_at	3,6	0,000398159	VIM
40167_s_at	2,3	0,000162845	WSB2
38740_at	-2,4	5,55E-06	ZFP36L1
32588_s_at	-2,4	0,031457514	ZFP36L2
35824_at	-2,4	0,001007457	ZNF238
933_f_at	-2,0	4,68E-06	ZNF91

Ende der Tabelle

6.4 GO-Analyse der Gesamt-Genliste nach Reanalyse der Daten von Küppers et al.

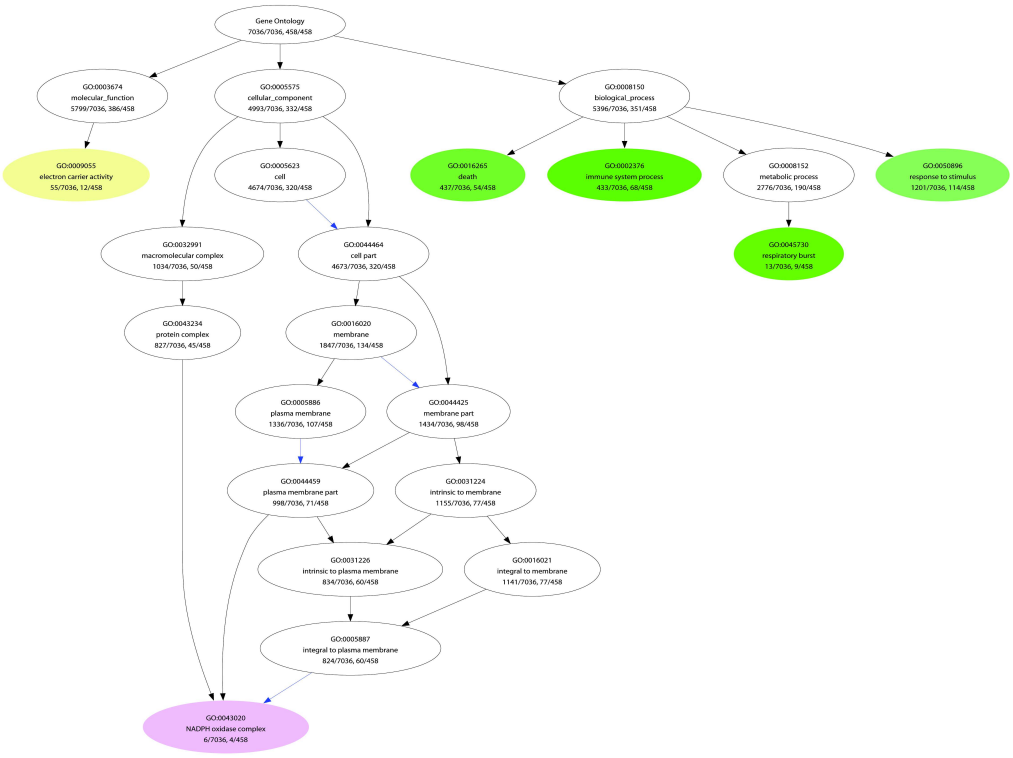


Abbildung 6.1: GO-Analyse der Gesamt-Genliste nach der Reanalyse der Daten von Küppers et al.

6.5 GO-Analyse der 299 *Probesets*, die nur nach Reanalyse der Daten von Küppers et al. gefunden wurden

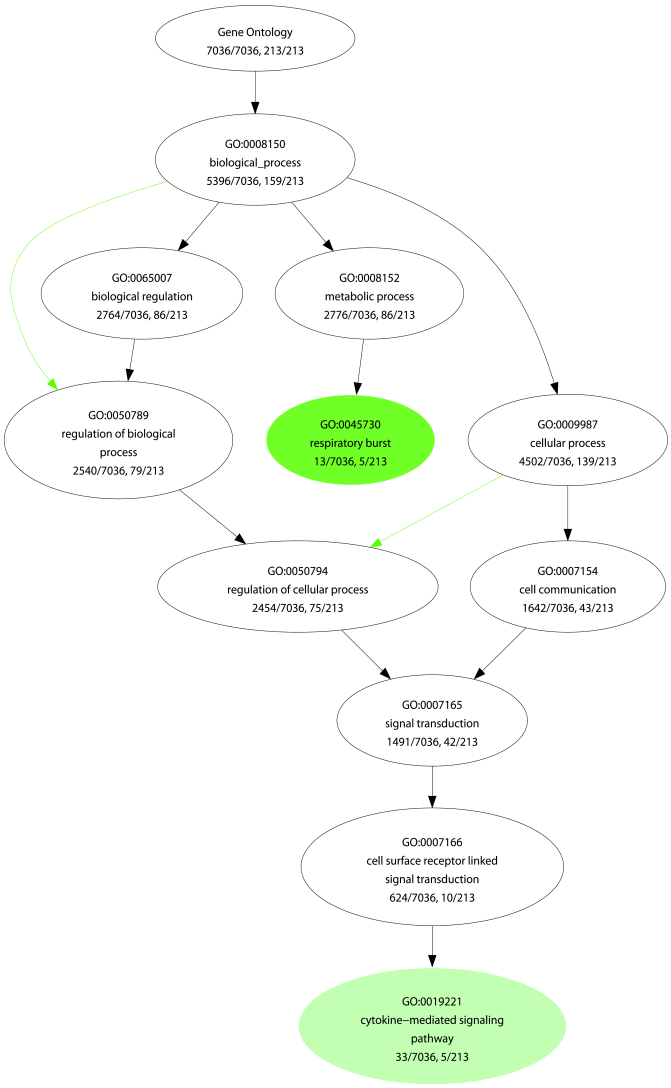


Abbildung 6.2: GO-Analyse der 299 *Probesets* die nur nach der Reanalyse der Daten von Küppers et al. gefunden wurden.

6.6 Der Zytokine-Zytokine-Interaktionssignalweg

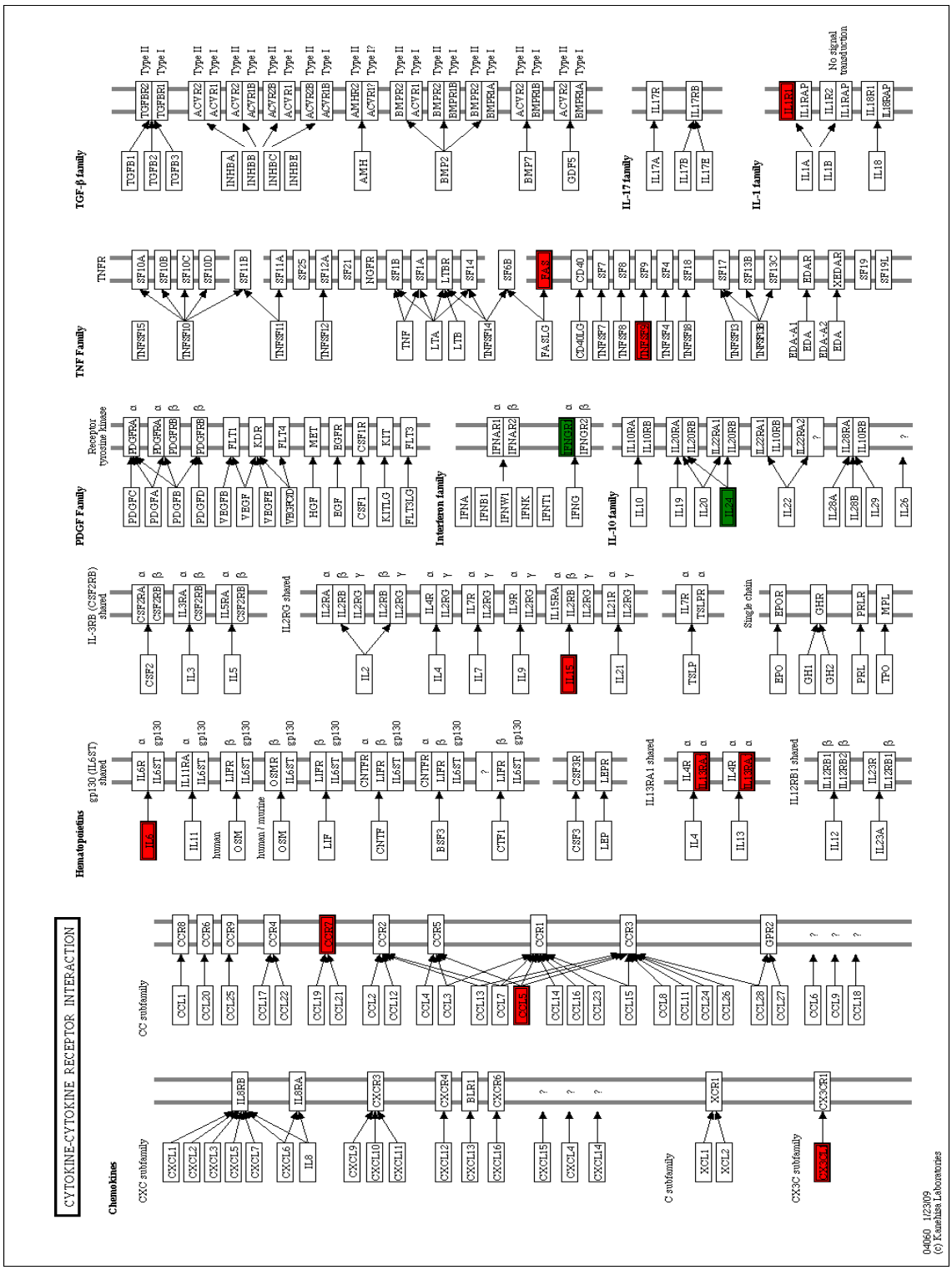


Abbildung 6.3: Signalweg Zytokine-Zytokine Interaktion

6.7 Der humane B-Zellen-Netzwerk-Signalweg

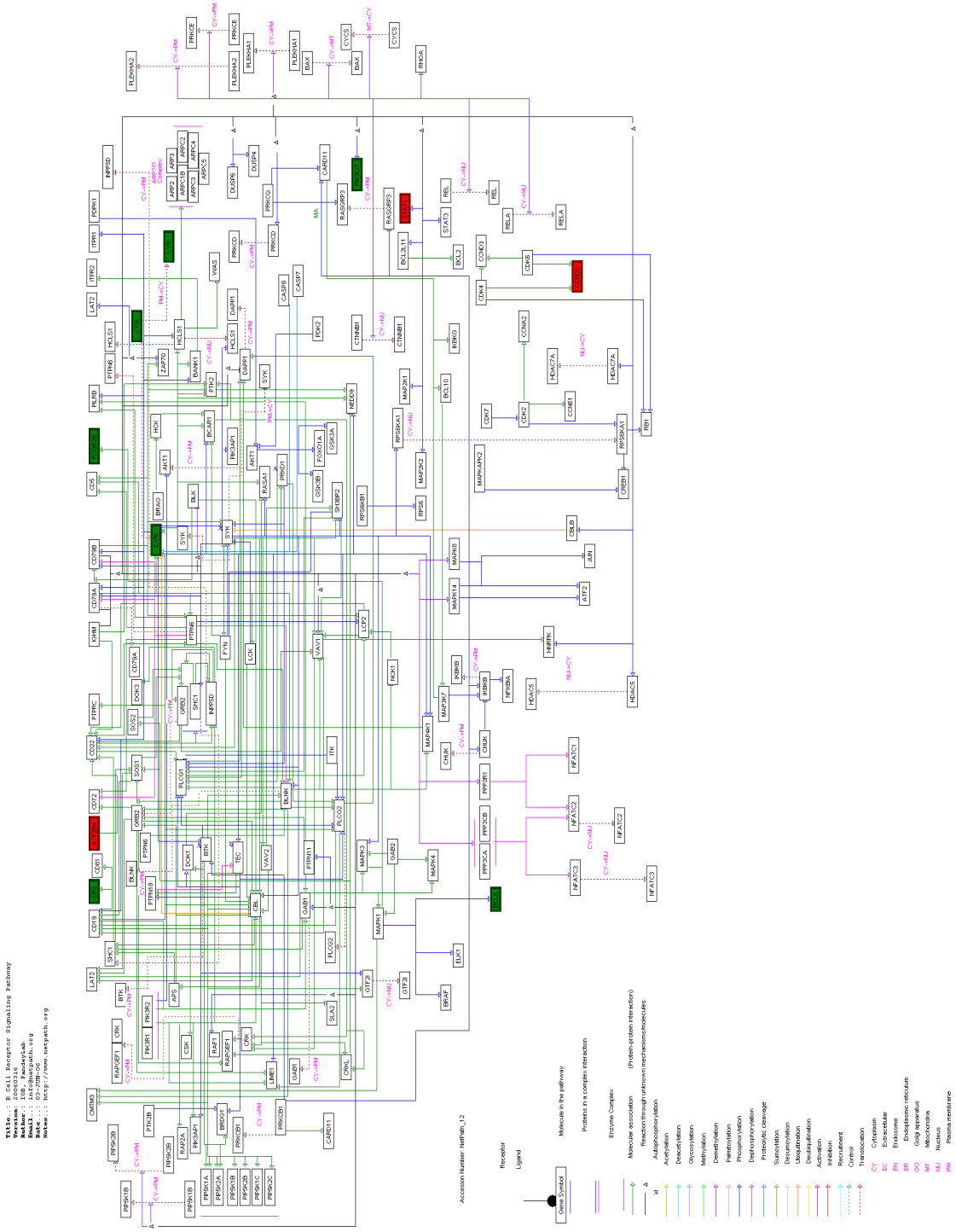


Abbildung 6.4: Signalweg humane B-Zellen Netzwerk

6.8 Reanalyse der *Microarray*-Daten von Piccaluga et al.

Tabelle 6.2: Genliste der zusätzlich gefundenen 30 Probesets mit $FC \geq 10$ oder ≤ -10 nach Reanalyse des Datensatzes von Piccaluga et al. [Piccaluga et al., 2007]

Affymetrix ID	PTCL NOS/T-Zellen FC	PTCL NOS/T-Zellen FDR	Gensymbol
223395_at	17,4	3,18E-13	ABI3BP
206134_at	16,8	7,77E-10	ADAMDEC1
205239_at	-12,9	2,58E-15	AREG /// AREGB
202953_at	12,2	8,51E-08	C1QB
217767_at	12,6	2,94E-10	C3
210072_at	21,5	6,36E-09	CCL19
215388_s_at	13,7	1,62E-16	CFH /// CFHR1
209395_at	11,7	1,33E-06	CHI3L1
222043_at	11,1	4,97E-07	CLU
1556499_s_at	16,7	3,10E-10	COL1A1
225681_at	14,3	5,56E-11	CTHRC1
204533_at	14,1	5,98E-11	CXCL10
210163_at	10,2	3,05E-09	CXCL11
218002_s_at	11,4	5,56E-11	CXCL14
203915_at	42,2	3,30E-14	CXCL9
202437_s_at	10,4	1,78E-09	CYP1B1
229390_at	13,3	3,47E-12	FAM26F
227265_at	11,6	1,97E-11	FGL2
202768_at	-20,6	2,08E-14	FOSB
201141_at	29,3	2,37E-11	GPNMB
202859_x_at	-16,9	4,95E-10	IL8
212192_at	15,8	1,65E-14	KCTD12

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 30 Probesets mit $FC \geq 10$ oder ≤ -10 nach Reanalyse des Datensatzes von Piccaluga et al. [Piccaluga et al., 2007] – Fortsetzung

Affymetrix ID	PTCL NOS/T-Zellen FC	PTCL NOS/T-Zellen FDR	Gensymbol
201744_s_at	48,3	2,54E-13	LUM
224823_at	13,1	1,54E-15	MYLK
221210_s_at	11,3	3,03E-13	NPL
221872_at	12,3	1,30E-11	RARRES1
201427_s_at	32,8	9,24E-13	SEPP1
219386_s_at	11,2	1,75E-14	SLAMF8
223044_at	23,6	6,68E-12	SLC40A1
219312_s_at	-12,7	5,85E-16	ZBTB10

Ende der Tabelle

6.9 Stabilitätsüberprüfung des Clusterverfahrens nach Suzuki et al.[Suzuki and Shimodaira, 2006]

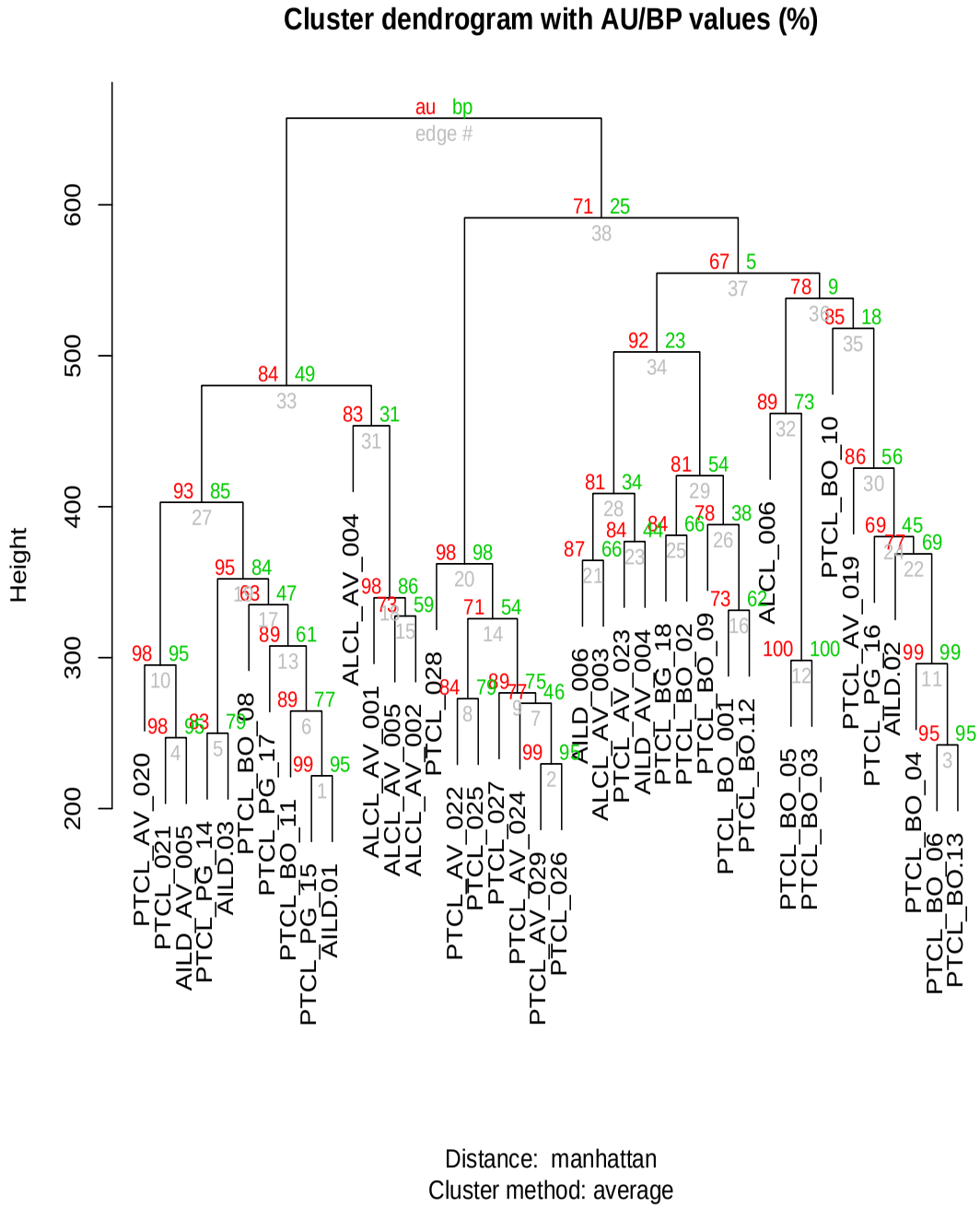


Abbildung 6.5: Stabilitätsprüfung des Clusterverfahrens nach Suzuki et al.[Suzuki and Shimodaira, 2006].

6.10 GO-Analyse der gesamten Genliste (599 *Probesets*), die nach Reanalyse der Daten von Piccaluga et al. gefunden wurde

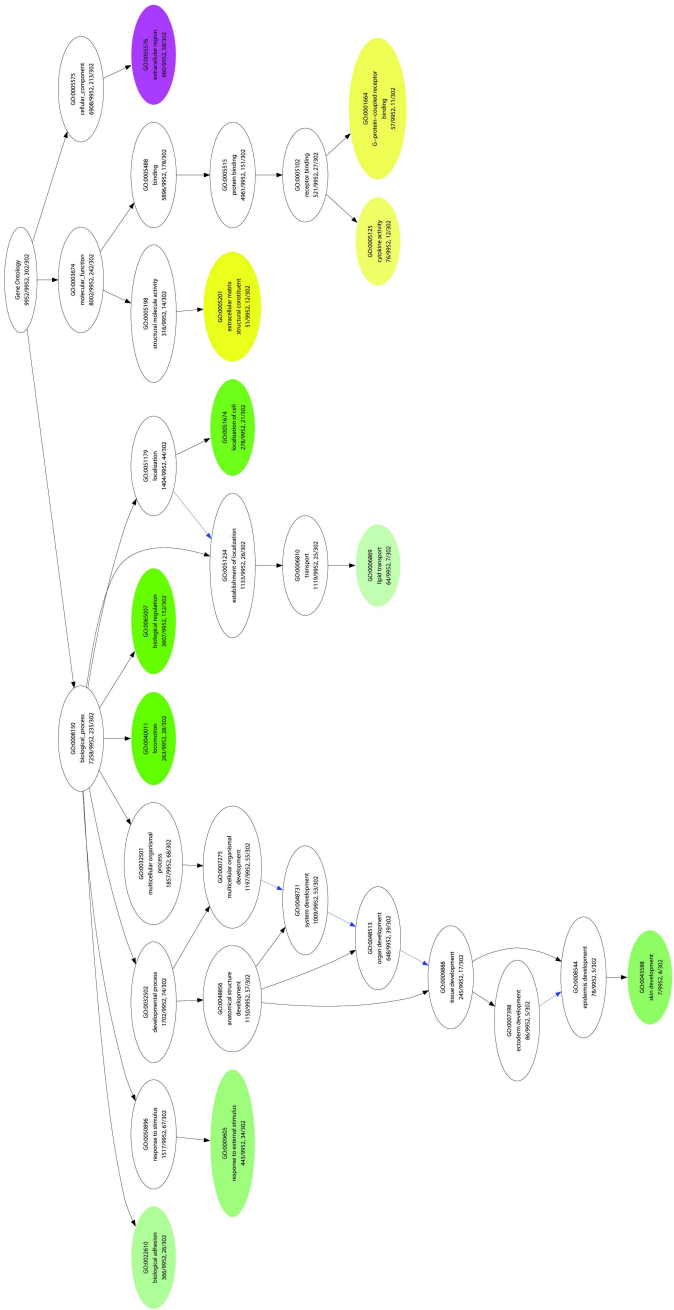


Abbildung 6.6: GO-Analyse über die gesamte Genliste (599 *Probesets*) die nach der Reanalyse der Daten von Piccaluga et al. gefunden wurde

6.11 GO-Analyse der gesamten Genliste (363 *Probesets*), die nur nach Reanalyse der Daten von Piccaluga et al. gefunden wurden

Abbildung 6.7: GO-Analyse über die Genliste (363 *Probesets*) die nur nach der Reanalyse der Daten von Piccaluga et al. gefunden wurden

6.12 Ausschließlich nach sVSN-Methode gefundene *Probesets* an einem Teildatensatz von Brune et al.

Tabelle 6.3: Genliste der zusätzlich gefundenen 66 Probesets nach Reanalyse des Datensatzes von Brune et al. [Brune et al., 2008]

Affymetrix ID	HL/B-Zellen FC	HL/B-Zellen FDR	Gensymbol
1558662_s_at	-2.1	0.031	BANK1
200838_at	3.3	0.001	CTSB
200951_s_at	4.0	0.000	CCND2
201153_s_at	-2.6	0.027	MBNL1
201162_at	2.9	0.031	IGFBP7
201458_s_at	-2.7	0.024	BUB3
201464_x_at	2.1	0.040	JUN
201566_x_at	2.2	0.006	ID2
201589_at	-2.7	0.028	SMC1A
201666_at	3.3	0.004	TIMP1
202633_at	-2.5	0.018	TOPBP1
202786_at	-2.2	0.047	STK39
203537_at	-3.1	0.010	PRPSAP2
204204_at	2.6	0.001	SLC31A2
204483_at	2.2	0.029	ENO3
204490_s_at	2.4	0.001	CD44
204803_s_at	2.2	0.010	RRAD
205207_at	3.2	0.001	IL6
205676_at	3.7	0.001	CYP27B1
205774_at	2.4	0.013	F12
205786_s_at	3.3	0.010	ITGAM
205801_s_at	-4.0	0.001	RASGRP3
205922_at	-3.2	0.004	VNN2
206958_s_at	-2.6	0.029	UPF3A
207843_x_at	2.2	0.040	CYB5A
207957_s_at	-2.2	0.016	PRKCB1
208168_s_at	2.5	0.001	CHIT1

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 66 Probesets nach Reanalyse des Datensatzes von Brune et al. [Brune et al., 2008] – Fortsetzung

Affymetrix ID	HL/B-Zellen FC	HL/B-Zellen FDR	Gensymbol
208450_at	3.0	0.002	LGALS2
209390_at	-2.4	0.012	TSC1
210858_x_at	-2.4	0.032	ATM
211373_s_at	2.4	0.006	PSEN2
212063_at	2.5	0.023	CD44
212205_at	-2.7	0.026	H2AFV
212388_at	-2.3	0.014	USP24
212399_s_at	-2.2	0.022	VGLL4
212590_at	-2.4	0.023	RRAS2
212632_at	-3.0	0.006	STX7
213044_at	-2.3	0.043	ROCK1
213415_at	2.1	0.002	CLIC2
213524_s_at	3.6	0.002	G0S2
213869_x_at	3.2	0.018	THY1
216241_s_at	-4.0	0.003	TCEA1
216609_at	3.3	0.000	TXN
216841_s_at	2.5	0.004	SOD2
217826_s_at	-2.2	0.036	UBE2J1
217941_s_at	-2.5	0.029	ERBB2IP
218761_at	-2.1	0.029	RNF111
221185_s_at	4.3	0.007	IQCG
221210_s_at	2.4	0.003	NPL
221234_s_at	-4.1	0.000	BACH2
222285_at	-3.1	0.047	IGHD
223382_s_at	3.1	0.010	ZNRF1
224641_at	-2.9	0.011	FYTTD1
224944_at	-2.5	0.034	TMPO

Fortsetzung auf der nächsten Seite

Genliste der zusätzlich gefundenen 66 Probesets nach Reanalyse des Datensatzes von Brune et al. [Brune et al., 2008] – Fortsetzung

Affymetrix ID	HL/B-Zellen FC	HL/B-Zellen FDR	Gensymbol
225174_at	-3.1	0.008	DNAJC10
225232_at	-2.2	0.019	MTMR12
227354_at	-2.5	0.009	PAG1
228376_at	2.0	0.039	GGTA1
229513_at	-2.5	0.013	STRBP
229560_at	2.3	0.000	TLR8
231418_at	-3.1	0.001	—
231577_s_at	2.4	0.016	GBP1
233261_at	-3.7	0.002	EBF1
235229_at	2.9	0.000	—
242334_at	-2.4	0.027	NLRP4
41047_at	2.4	0.023	C9orf16

Ende der Tabelle

Literaturverzeichnis

Affymetrix. *Affymetrix Microarray Suite User Guide*. Santa Clara, CA: Affymetrix, 5.Auflage, 2001.

Affymetrix(2002). *Affymetrix (2002): Statistical algorithms description document. Affymetrix Clara, CA*.

B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molekularbiologie der Zelle*. VCH, 1995.

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A*, 96(12):6745–6750, Jun 1999.

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29, May 2000. doi: 10.1038/75556. URL `http://dx.doi.org/10.1038/75556`.

R. Baayen, D. Davidson, and D. Bates. Mixed-effects modeling with crossed random effects for subject and items. *Journal of Memory and Language*, 59:390–412, 2008.

M. Bai, E. Tsanou, N. J. Agnantis, S. Kamina, C. Grepi, K. Stefanaki, D. Rontogianni, V. Galani, and P. Kanavaros. Proliferation profile of classical hodgkin's lymphomas. increased expression of the protein cyclin d2 in hodgkin's and reed-sternberg cells. *Mod Pathol*, 17(11):1338–1345, Nov 2004. doi: 10.1038/modpathol.3800183. URL `http://dx.doi.org/10.1038/modpathol.3800183`.

G. Bansal, J. A. DiVietro, H. S. Kuehn, S. Rao, K. H. Nocka, A. M. Gilfillan, and K. M. Druey. RGS13 controls g protein-coupled receptor-evoked responses of human mast cells. *J Immunol*, 181(11):7882–7890, Dec 2008.

Y. Benjamini, D. Drai, G. Elmer, N. Kafkafi, and I. Golani. Controlling the false discovery rate in behavior genetics research. *Behav Brain Res*, 125(1-2):279–284, Nov 2001.

T. Boes and M. Neuhäuser. Normalization for Affymetrix GeneChips. *Methods Inf Med*, 44(3):414–417, 2005. doi: 10.1267/METH05030414. URL `http://dx.doi.org/10.1267/METH05030414`.

B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, Jan 2003.

R. Breitling, A. Amtmann, and P. Herzyk. Iterative group analysis (iga): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, 5:34, Mar 2004. doi: 10.1186/1471-2105-5-34. URL `http://dx.doi.org/10.1186/1471-2105-5-34`.

L. Bross, Y. Fukita, F. McBlane, C. Démollière, K. Rajewsky, and H. Jacobs. DNA double-strand breaks in immunoglobulin genes undergoing somatic hypermutation. *Immunity*, 13(5):589–597, Nov 2000.

V. Brune, E. Tiacci, I. Pfeil, C. Döring, S. Eckerle, C. J. M. van Noesel, W. Klapper, B. Falini, A. von Heydebreck, D. Metzler, A. Bräuninger, M.-L. Hansmann, and R. Küppers. Origin and pathogenesis of nodular lymphocyte-predominant Hodgkin lymphoma as revealed by global gene expression analysis. *J Exp Med*, 205(10):2251–2268, Sep 2008. doi: 10.1084/jem.20080809. URL `http://dx.doi.org/10.1084/jem.20080809`.

A. J. Butte, L. Bao, B. Y. Reis, T. W. Watkins, and I. S. Kohane. Comparing the similarity of time-series gene expression using signal processing metrics. *J Biomed Inform*, 34(6):396–405, Dec 2001. doi: 10.1006/jbin.2002.1037. URL `http://dx.doi.org/10.1006/jbin.2002.1037`.

E. D. Cahir-McFarland, K. Carter, A. Rosenwald, J. M. Giltnane, S. E. Henrickson, L. M. Staudt, and E. Kieff. Role of NF-kappa B in cell survival and transcription

of latent membrane protein 1-expressing or Epstein-Barr virus latency III-infected cells. *J Virol*, 78(8):4108–4119, Apr 2004.

E. Camon, D. Barrell, V. Lee, E. Dimmer, and R. Apweiler. The gene ontology annotation (goa) database–an integrated resource of go annotations to the uniprot knowledgebase. *In Silico Biol*, 4(1):5–6, 2004.

W. J. Chng, E. D. Remstein, R. Fonseca, P. L. Bergsagel, J. A. Vrana, P. J. Kurtin, and A. Dogan. Gene expression profiling of pulmonary mucosa-associated lymphoid tissue lymphoma identifies new biologic insights with potential diagnostic and therapeutic applications. *Blood*, 113(3):635–645, Jan 2009. doi: 10.1182/blood-2008-02-140996. URL `http://dx.doi.org/10.1182/blood-2008-02-140996`.

S. E. Choe, M. Boutros, A. M. Michelson, G. M. Church, and M. S. Halfon. Preferred analysis methods for Affymetrix GeneChips revealed by a wholly defined control dataset. *Genome Biol*, 6(2):R16, 2005. doi: 10.1186/gb-2005-6-2-r16. URL `http://dx.doi.org/10.1186/gb-2005-6-2-r16`.

L. Cope, S. M. Hartman, H. W. H. Göhlmann, J. P. Tiesman, and R. A. Irizarry. Analysis of Affymetrix GeneChip data using amplified RNA. *Biotechniques*, 40 (2):165–6, 168, 170, Feb 2006.

C.-Y. Dong, F. Zhang, Y.-J. Duan, B.-X. Yang, Y.-M. Lin, and X.-T. Ma. mda-7/IL-24 inhibits the proliferation of hematopoietic malignancies in vitro and in vivo. *Exp Hematol*, 36(8):938–946, Aug 2008. doi: 10.1016/j.exphem.2008.03.009. URL `http://dx.doi.org/10.1016/j.exphem.2008.03.009`.

A. Dutton, J. D. O'Neil, A. E. Milner, G. M. Reynolds, J. Starczynski, J. Crocker, L. S. Young, and P. G. Murray. Expression of the cellular FLICE-inhibitory protein (c-FLIP) protects Hodgkin's lymphoma cells from autonomous Fas-mediated death. *Proc Natl Acad Sci U S A*, 101(17):6611–6616, Apr 2004. doi: 10.1073/pnas.0400765101. URL `http://dx.doi.org/10.1073/pnas.0400765101`.

B. Efron and R. Tibshirani. *An introduction to the bootstrap.* New York: Chapman and Hall., 1993.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25): 14863–14868, Dec 1998.

M. Fischer, M. Juremalm, N. Olsson, C. Backlin, C. Sundström, K. Nilsson, G. Enblad, and G. Nilsson. Expression of CCL5/RANTES by Hodgkin and Reed-Sternberg cells and its possible role in the recruitment of mast cells into lymphomatous tissue. *Int J Cancer*, 107(2):197–201, Nov 2003 2003. doi: 10.1002/ijc.11370. URL `http://dx.doi.org/10.1002/ijc.11370`.

R. I. Fisher. Overview of non-Hodgkin's lymphoma: biology, staging, and treatment. *Semin Oncol*, 30(2 Suppl 4):3–9, Apr 2003. doi: 10.1053/sonc.2003.23797. URL `http://dx.doi.org/10.1053/sonc.2003.23797`.

R. Förster, A. C. Davalos-Misslitz, and A. Rot. CCR7 and its ligands: balancing immunity and tolerance. *Nat Rev Immunol*, 8(5):362–371, May 2008. doi: 10.1038/nri2297. URL `http://dx.doi.org/10.1038/nri2297`.

L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry. affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315, Feb 2004. doi: 10.1093/bioinformatics/btg405. URL `http://dx.doi.org/10.1093/bioinformatics/btg405`.

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, 5(10): R80, 2004. doi: 10.1186/gb-2004-5-10-r80. URL `http://dx.doi.org/10.1186/gb-2004-5-10-r80`.

N. Ghilardi, J. Li, J.-A. Hongo, S. Yi, A. Gurney, and F. J. de Sauvage. A novel type I cytokine receptor is expressed on monocytes, signals proliferation, and activates STAT-3 and STAT-5. *J Biol Chem*, 277(19):16831–16836, May 2002. doi: 10.1074/jbc.M201140200. URL `http://dx.doi.org/10.1074/jbc.M201140200`.

T. R. Golub. Genome-wide views of cancer. *N Engl J Med*, 344(8):601–602, Feb 2001.

T. Goossens, U. Klein, and R. Küppers. Frequent occurrence of deletions and duplications during somatic hypermutation: implications for oncogene translocations and heavy chain disease. *Proc Natl Acad Sci U S A*, 95(5):2463–2468, Mar 1998.

S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron. Improved detection of overrepresentation of Gene-Ontology annotations with parent child analysis. *Bioinformatics*, 23(22):3024–3031, Nov 2007. doi: 10.1093/bioinformatics/btm440. URL `http://dx.doi.org/10.1093/bioinformatics/btm440`.

S. Grufferman and E. Delzell. Epidemiology of Hodgkin's disease. *Epidemiol Rev*, 6:76–106, 1984.

N. Gutensohn and P. Cole. Epidemiology of Hodgkin's disease. *Semin Oncol*, 7(2): 92–102, Jun 1980.

M. Hansmann, L. Weiss, H. Stein, N. Harris, and E. Jaffe. *Pathology of lymphocyte-predominance Hodgkin's disease.* Lippencott Williams &Wilkins: Philadelphia, 1999.

N. Harris. Hodgkin's lymphomas: classification, diagnosis, and grading. *Semin Hematol*, 36:220–232, 1999.

N. L. Harris, E. S. Jaffe, H. Stein, P. M. Banks, J. K. Chan, M. L. Cleary, G. Delsol, C. D. Wolf-Peeters, B. Falini, and K. C. Gatter. A revised European-American classification of lymphoid neoplasms: a proposal from the International Lymphoma Study Group. *Blood*, 84(5):1361–1392, Sep 1994.

M. Herling, K. A. Patel, E. D. Hsi, K.-C. Chang, G. Z. Rassidakis, R. Ford, and D. Jones. TCL1 in B-cell tumors retains its normal B-cell pattern of regulation and is a marker of differentiation stage. *Am J Surg Pathol*, 31(7):1123–1129, Jul 2007. doi: 10.1097/PAS.0b013e31802e2201. URL `http://dx.doi.org/10.1097/PAS.0b013e31802e2201`.

T. Hodgkin. On some morbid appearances of the absorbent glands and spleen. *Med Chir Trans*, 17:68–114, 1832.

R. Hoffmann, T. Seidl, and M. Dugas. Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biol*, 3(7):RESEARCH0033, Jun 2002.

S. Holm. A simple sequentially rejective multiple test procedure. *Scand J Stat*, 6: 65–70, 1979.

Horst Stöcker. *Taschenbuch mathematischer Formeln und moderner Verfahren*. Harri Deutsch Verlag, 2007.

U. E. Höpken, H.-D. Foss, D. Meyer, M. Hinz, K. Leder, H. Stein, and M. Lipp. Up-regulation of the chemokine receptor CCR7 in classical but not in lymphocyte-predominant Hodgkin disease correlates with distinct dissemination of neoplastic cells in lymphoid organs. *Blood*, 99(4):1109–1116, Feb 2002.

W. Huber and R. Gentleman. matchprobes: a bioconductor package for the sequence-matching of microarray probe elements. *Bioinformatics*, 20(10):1651–1652, Jul 2004. doi: 10.1093/bioinformatics/bth133. URL `http://dx.doi.org/10.1093/bioinformatics/bth133`.

W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:S96–104, 2002.

W. Huber, A. von Heydebreck, H. Sueltmann, A. Poustka, and M. Vingron. Parameter estimation for the calibration and variance stabilization of microarray data. *Stat Appl Genet Mol Biol*, 2:Article3, 2003. doi: 10.2202/1544-6115.1008. URL `http://dx.doi.org/10.2202/1544-6115.1008`.

M. Hummel, S. Bentink, H. Berger, W. Klapper, S. Wessendorf, T. F. E. Barth, H.-W. Bernd, S. B. Cogliatti, J. Dierlamm, A. C. Feller, M.-L. Hansmann, E. Haralambieva, L. Harder, D. Hasenclever, M. Kühn, D. Lenze, P. Lichter, J. I. Martin-Subero, P. Möller, H.-K. Müller-Hermelink, G. Ott, R. M. Parwaresch, C. Pott, A. Rosenwald, M. Rosolowski, C. Schwaenen, B. Stürzenhofecker, M. Szczepanowski, H. Trautmann, H.-H. Wacker, R. Spang, M. Loeffler, L. Trümper, H. Stein, R. Siebert, and M. M. in Malignant Lymphomas Network Project of the Deutsche Krebshilfe. A biologic definition of Burkitt's lymphoma from transcriptional and genomic profiling. *N Engl J Med*, 354(23):2419–2430, Jun 2006.

R. Ihaka and R. Gentleman. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996.

R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*, 31(4): e15, Feb 2003a.

R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, Apr 2003b. doi: 10.1093/biostatistics/4.2.249. URL `http://dx.doi.org/10.1093/biostatistics/4.2.249`.

R. A. Irizarry, Z. Wu, and H. A. Jaffee. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, 22(7):789–794, Apr 2006. doi: 10.1093/bioinformatics/btk046. URL `http://dx.doi.org/10.1093/bioinformatics/btk046`.

V. R. Iyer, M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. Lee, J. M. Trent, L. M. Staudt, J. Hudson, M. S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P. O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283(5398):83–87, Jan 1999.

E. S. Jaffe, N. L. Harris, J. Diebold, and H. K. Müller-Hermelink. World Health Organization Classification of lymphomas: a work in progress. *Ann Oncol*, 9 Suppl 5:S25–S30, 1998.

N. M. Jiwa, P. V. der Valk, H. Mullink, W. Vos, A. Horstman, M. M. Maurice, D. E. Olde-Weghuis, J. M. Walboomers, and C. J. Meijer. Epstein-Barr virus DNA in Reed-Sternberg cells of Hodgkin's disease is frequently associated with CR2 (EBV receptor) expression. *Histopathology*, 21(1):51–57, Jul 1992.

U. Klein, T. Goossens, M. Fischer, H. Kanzler, A. Braeuninger, K. Rajewsky, and R. Küppers. Somatic hypermutation in normal and transformed human B cells. *Immunol Rev*, 162:261–280, Apr 1998.

U. Klein, Y. Tu, G. A. Stolovitzky, M. Mattioli, G. Cattoretti, H. Husson, A. Freedman, G. Inghirami, L. Cro, L. Baldini, A. Neri, A. Califano, and R. Dalla-Favera. Gene expression profiling of B cell chronic lymphocytic leukemia reveals a homogeneous phenotype related to memory B cells. *J Exp Med.*, 194(11):1625–38, 2001.

R. Küppers. Mechanisms of B-cell lymphoma pathogenesis. *Nat Rev Cancer*, 5(4): 251–262, Apr 2005. doi: 10.1038/nrc1589. URL http://dx.doi.org/10.1038/nrc1589.

R. Küppers and R. Dalla-Favera. Mechanisms of chromosomal translocations in B cell lymphomas. *Oncogene*, 20(40):5580–5594, Sep 2001. doi: 10.1038/sj.onc.1204640. URL http://dx.doi.org/10.1038/sj.onc.1204640.

R. Küppers, U. Klein, I. Schwering, V. Distler, A. Bräuninger, G. Cattoretti, Y. Tu, G. A. Stolovitzky, A. Califano, M.-L. Hansmann, and R. Dalla-Favera. Identification of Hodgkin and Reed-Sternberg cell-specific genes by gene expression profiling. *J Clin Invest*, 111(4):529–537, Feb 2003. doi: 10.1172/JCI16624. URL http://dx.doi.org/10.1172/JCI16624.

E. N. Lazaridis, D. Sinibaldi, G. Bloom, S. Mane, and R. Jove. A simple method to improve probe set estimates from oligonucleotide arrays. *Math Biosci*, 176(1): 53–58, Mar 2002.

R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart. High density synthetic oligonucleotide arrays. *Nat Genet*, 21(1 Suppl):20–24, Jan 1999. doi: 10.1038/4447. URL http://dx.doi.org/10.1038/4447.

D. Mahadevan, C. Spier, K. D. Croce, S. Miller, B. George, C. Riley, S. Warner, T. M. Grogan, and T. P. Miller. Transcript profiling in peripheral t-cell lymphoma, not otherwise specified, and diffuse large b-cell lymphoma identifies distinct tumor profile signatures. *Mol Cancer Ther*, 4(12):1867–1879, Dec 2005. doi: 10.1158/1535-7163.MCT-05-0146. URL http://dx.doi.org/10.1158/1535-7163.MCT-05-0146.

S. Mathas, M. Hinz, I. Anagnostopoulos, D. Krappmann, A. Lietz, F. Jundt, K. Bommert, F. Mechta-Grigoriou, H. Stein, B. Dörken, and C. Scheidereit. Aberrantly expressed c-Jun and JunB are a hallmark of Hodgkin lymphoma cells, stimulate proliferation and synergize with NF-kappa B. *EMBO J*, 21(15):4104–4113, Aug 2002.

F. Naef, D. A. Lim, N. Patil, and M. Magnasco. Dna hybridization to mismatched templates: a chip study. *Phys Rev E Stat Nonlin Soft Matter Phys*, 65(4 Pt 1): 040902, Apr 2002.

S. Nagel, M. Scherr, H. Quentmeier, M. Kaufmann, M. Zaborski, H. G. Drexler, and R. A. F. MacLeod. HLXB9 activates IL6 in Hodgkin lymphoma cell lines and is regulated by PI3K signalling involving E2F3. *Leukemia*, 19(5):841–846, May 2005. doi: 10.1038/sj.leu.2403716. URL `http://dx.doi.org/10.1038/sj.leu.2403716`.

M. G. Narducci, E. Pescarmona, C. Lazzeri, S. Signoretti, A. M. Lavinia, D. Remotti, E. Scala, C. D. Baroni, A. Stoppacciaro, C. M. Croce, and G. Russo. Regulation of TCL1 expression in B- and T-cell lymphomas and reactive lymphoid tissues. *Cancer Res*, 60(8):2095–2100, Apr 2000.

D. A. Notterman, U. Alon, A. J. Sierk, and A. J. Levine. Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. *Cancer Res*, 61(7):3124–3130, Apr 2001.

F. N. Papavasiliou and D. G. Schatz. Cell-cycle-regulated DNA double-stranded breaks in somatic hypermutation of immunoglobulin genes. *Nature*, 408(6809): 216–221, Nov 2000. doi: 10.1038/35041599. URL `http://dx.doi.org/10.1038/35041599`.

R. S. Parrish and H. J. Spencer. Effect of normalization on significance testing for oligonucleotide microarrays. *J Biopharm Stat*, 14(3):575–589, Aug 2004.

P. P. Piccaluga, C. Agostinelli, A. Califano, M. Rossi, K. Basso, S. Zupo, P. Went, U. Klein, P. L. Zinzani, M. Baccarani, R. D. Favera, and S. A. Pileri. Gene expression analysis of peripheral T cell lymphoma, unspecified, reveals distinct profiles and new potential therapeutic targets. *J Clin Invest*, 117(3):823–834, Mar 2007. doi: 10.1172/JCI26833. URL `http://dx.doi.org/10.1172/JCI26833`.

J. C. Pinheiro and D. M. Bates. *Mixed-Effects Models in S and S-Plus*. Springer New York, 2000.

A. Ploner, L. D. Miller, P. Hall, J. Bergh, and Y. Pawitan. Correlation test to assess low-level processing of high-density oligonucleotide microarray data. *BMC Bioinformatics*, 6:80, 2005. doi: 10.1186/1471-2105-6-80. URL `http://dx.doi.org/10.1186/1471-2105-6-80`.

J. Z. Qin, C. L. Zhang, J. Kamarashev, R. Dummer, G. Burg, and U. Döbbeling. Interleukin-7 and interleukin-15 regulate the expression of the bcl-2 and c-myb genes in cutaneous T-cell lymphoma cells. *Blood*, 98(9):2778–2783, Nov 2001.

K. R. Ralf Küppers. *Developmental anf Functional Biology of B lymphocytes*. Baltimore: Lippincott Williams & Wilkins, 2003.

D. Reed. On the pathological changes in Hodgkin's disease with special reference to its relation to tuberculosis. *John Hopkins Hosp Rep*, 10:133–193, 1902.

C. Renné, J. I. Martin-Subero, M. Eickernjäger, M.-L. Hansmann, R. Küppers, R. Siebert, and A. Bräuninger. Aberrant expression of id2, a suppressor of b-cell-specific gene expression, in hodgkin's lymphoma. *Am J Pathol*, 169(2):655–664, Aug 2006.

D. M. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *J Comput Biol*, 8(6):557–569, 2001. doi: 10.1089/106652701753307485. URL `http://dx.doi.org/10.1089/106652701753307485`.

W. Ruan, J. Lin, E. ping Xu, F. ying Xu, Y. Ma, H. Deng, Q. Huang, B. jian Lv, H. Hu, J. Cui, M. juan Di, J. kang Dong, and M. de Lai. IGFBP7 plays a potential tumor suppressor role against colorectal carcinogenesis with its expression associated with DNA hypomethylation of exon 1. *J Zhejiang Univ Sci B*, 7(11):929–932, Nov 2006. doi: 10.1631/jzus.2006.B0929. URL `http://dx.doi.org/10.1631/jzus.2006.B0929`.

H. Saffer, A. Wahed, G. Z. Rassidakis, and L. J. Medeiros. Clusterin expression in malignant lymphomas: a survey of 266 cases. *Mod Pathol*, 15(11):1221–1226, Nov 2002. doi: 10.1097/01.MP.0000036386.87517.AA. URL `http://dx.doi.org/10.1097/01.MP.0000036386.87517.AA`.

G. Sherlock. Analysis of large-scale gene expression data. *Curr Opin Immunol*, 12 (2):201–205, Apr 2000.

H. Shimodaira. An approximately unbiased test of phylogenetic tree selection. *Syst Biol*, 51(3):492–508, Jun 2002. doi: 10.1080/10635150290069913. URL `http://dx.doi.org/10.1080/10635150290069913`.

T. Sørlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, P. E. Lønning, and A. L. Børresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc Natl Acad Sci U S A*, 98(19):10869–10874, Sep 2001. doi: 10.1073/pnas.191367098. URL `http://dx.doi.org/10.1073/pnas.191367098`.

C. Sternberg. Über eine eigenartige unter den Bilde der Pseudoleukämie verlaufende Tuberkulose des lymphatischen Apparates. *Z Heilkunde*, 19, 1898.

R. Suzuki and H. Shimodaira. Pvclust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542, Jun 2006. doi: 10.1093/bioinformatics/btl117. URL `http://dx.doi.org/10.1093/bioinformatics/btl117`.

P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A*, 96(6):2907–2912, Mar 1999.

C. Thorns, A. C. Feller, and H. Merz. Emmprin (cd 174) is expressed in hodgkin's lymphoma and anaplastic large cell lymphoma. an immunohistochemical study of 60 cases. *Anticancer Res*, 22(4):1983–1986, 2002.

R. Tibshirani. Estimating Transformations for Regression via Additivity and Variance Stabilisation. *J.Amer.Stat.Assoc.*, 83:394–405, 1988.

J. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

L. Virgilio, M. G. Narducci, M. Isobe, L. G. Billips, M. D. Cooper, C. M. Croce, and G. Russo. Identification of the TCL1 gene involved in T-cell malignancies. *Proc Natl Acad Sci U S A*, 91(26):12530–12534, Dec 1994.

R. Wehner and W. Gehring. *Zoologie*. Thieme, 1995.

L. Weiss, J. Chan, K. MacLennan, and R. Warnke. *Pathology of classical Hodgkin's Disease*. Lippencott Williams &Wilkins: Philadelphia, 1999.

P. Westfall and S. Young. *Resampling-Based Multiple Testing*. Wiley, New York, 1993.

E. Yefenof, G. Klein, M. Jondal, and M. B. Oldstone. Surface markers on human B and T-lymphocytes. IX. Two-color immunofluorescence studies on the association between ebv receptors and complement receptors on the surface of lymphoid cell lines. *Int J Cancer*, 17(6):693–700, Jun 1976.

Abkürzungsverzeichnis

$B-Zellen$	B-Lymphozyten
BL	Burkitt-Lymphom
$cDNA$	komplementäre DNA
$DLBCL$	diffus großzelliges B-Zell-Lymphom
FC	*Fold Change*
FC	Änderungsverhältnis (Fold Change)
FDR	*False Discovery Rate*
$FWER$	*family-wise Error rate*
GC	Keimzentrum
GO	*Gen-Ontologien*
HL	Hodgkin-Lymphom
IM	idealer *Mismatch*
MHC	Haupthistokompatibilitätskomplex
MM	Mismatch
$mRNA$	Boten-RNA (messenger RNA)
NHL	Non-Hodgkin-Lymphome
NK	natürliche Killer T-Zellen
PC	Plasmazellen
PM	Perfect Match
RMA	Robust Multichip Average
SNP	*Single Nucleotide Polymorphism*
$T-Zellen$	T-Lymphozyten
$Treg$	regulatorische T-Zellen
VSN	Variance Stabilisation Model
WHO	World Health Organization

Danksagung

Auf das Entstehen dieser Arbeit hatten viele Personen einen positiven Einfluss und Anteil.

Mein besonderer Dank gilt Herrn Prof. Dr. Dirk Metzler für die intensive und konstruktive Betreuung meiner Promotion.

Herrn Prof. Dr. Martin-Leo Hansmann danke ich für die Möglichkeit der Anfertigung dieser Arbeit am Senckenbergischen Institut für Pathologie des Universitätsklinikum Frankfurt am Main für die Bereitstellung der Computerhardware und Software und für die kontinuierliche anregende Diskussionsbereitschaft.

Besonders danken möchte ich Anja von Heydebreck für die Hilfe bei den ersten Schritten der statistischen Datenauswertung von *Microarrays*, Dr. Verena Brune für die Unterstützung zur Klärung von biologischen Fragestellungen und Bereitstellung ihrer Expressionsdatensätze.

Für das sehr gute Arbeitsklima, die tatkräftige Unterstützung, die erfolgreichen Kooperationen, die vielen hilfreichen Diskussionen und persönlichen Gesprächen danke ich den Mitarbeitern der Arbeitsgruppe von Herrn Prof. Ralf Küppers am Institut für Zellbiologie (Tumorforschung) der Universitätsklinik Essen und den Mitarbeitern des Senckenbergischen Instituts für Pathologie des Universitätsklinikums Frankfurt am Main.
Ganz besonders danke ich Verena Brune, Matthias Frank, Ralf Lieberz, Lin Himmelmann, Ronja Düffel, Nicole Diekert, Tina Feeser und Wie-

land Keilholz.

Meine besondere Dankbarkeit gilt meiner Mutter, Bernard, meinem Lebensgefährten Olaf, Evelyn und allen Freunden, die mich mit viel Liebe und Tatkraft unterstützt und begleitet haben.

Publikationen

Baer P.C., **Döring C.**, Hansmann M.L., Schubert R., Geiger H.: New Insights into Epithelial Differentiation of human Adipose-derived Stem Cells. In Bearbeitung

Döring C., Hansmann M.L., Küppers R.,Metzler: Reanalysis of microarray data: New insight to Hodgkin lymphoma. Eingereicht

Hartmann S., Gesk S., Scholtysik R., Kreuz M., Bug S., Vater I., **Döring C.**, Cogliatti S., Parrens M., Merlio J.P., Kwiecinska A., Porwit A., Piccaluga P.P., Pileri S., Hoefler G., Küppers R., Siebert R., Hansmann M.L.: High resolution SNP array genomic profiling of peripheral T cell lymphomas, not otherwise specified, identifies a subgroup with chromosomal aberrations affecting the REL locus, submitted to British Journal of Haematology. Br J Haematol. 2009 Oct 22, PMID: 19863542

Eckerle S, Brune V, **Döring C**, Tiacci E, Sundström C, Kodet R, Paulli M, Falini B, Klapper W, Baur Chaubert A, Willenbrock K, Metzler D, Bräuninger A, Küppers R, Hansmann ML.: Gene expression profiling of isolated tumor cells from anaplastic large cell lymphoma: insights into its cellular origin, pathogenesis, clinico-pathogenic subtypes and relation to Hodgkin lymphoma. Leukemia. 2009 Aug 6. PMID: 19657361

Schumacher M.A., Schmitz R., Brune V., Tiacci E., **Döring C.**, Hans-

mann M.L, Siebert R. and Küppers R.: Mutations in the genes coding for the NF-κB regulating factors IκBα and A20 are uncommon in nodular lymphocyte-predominant Hodgkin lymphoma. Haematologica. 2009 July 31. PMID: 19648161

Baus D, Nonnenmacher F, Jankowski S, **Döring C**, Bräutigam C, Frank M, Hansmann ML, Pfitzner E: STAT6 and STAT1 are essential antagonistic regulators of cell survival in Hodgkin lymphoma. Leukemia. 2009 May 14. PMID: 19440213

Hinsch N, Frank M, **Döring C**, Vorländer C, Hansmann ML: QPRT: a potential marker for follicular thyroid carcinoma including minimal invasive variant; a gene expression, RNA and immunohistochemical study. BMC Cancer. 2009 Mar 26;9:93. PMID: 19321014

Brune V, Tiacci E, Pfeil I, **Döring C**, Eckerle S, van Noesel CJ, Klapper W, Falini B, von Heydebreck A, Metzler D, Bräuninger A, Hansmann ML, Küppers R.: Origin and pathogenesis of nodular lymphocyte-predominant Hodgkin lymphoma as revealed by global gene expression analysis. J Exp Med. 2008 Sep 15. PMID: 18794340

Renné C, Willenbrock K, Martin-Subero JI, Hinsch N, **Döring C**, Tiacci E, Klapper W, Möller P, Küppers R, Hansmann ML, Siebert R, Bräunin-

ger A.: High expression of several tyrosine kinases and activation of the PI3K/AKT pathway in mediastinal large B cell lymphoma reveals further similarities to Hodgkin lymphoma. Leukemia. 2007 Apr;21(4):780-7. PMID: 17375124

Knauer SK, Bier C, Schlag P, Fritzmann J, Dietmaier W, Rödel F, Klein-Hitpass L, Kovács AF, **Döring C**, Hansmann ML, Hofmann WK, Kunkel M, Brochhausen C, Engels K, Lippert BM, Mann W, Stauber RH.: The survivin isoform survivin-3B is cytoprotective and can function as a chromosomal passenger complex protein. Cell Cycle. 2007 Jun 15;6(12):1502-9. PMID: 17582222

Frank M, **Döring C**, Metzler D, Eckerle S, Hansmann ML.: Global gene expression profiling of formalin-fixed paraffin-embedded tumor samples: a comparison to snap-frozen material using oligonucleotide microarrays. Virchows Arch. 2007 Jun;450(6):699-711. PMID: 17479285

Lebenslauf

Frankfurt am Main, im Dezember 2009

Persönliche Daten

Claudia Döring
geb. 16.08.1976 in Frankfurt am Main

Schulausbildung

1983-1987	Robinson Grundschule in Hattersheim am Main
1987-1996	Oberstufengymnasium der Heinrich-Böll Gesamtschule Hattersheim am Main

Akademische Ausbildung

1996-2001	Studium der Biologie an der Johann-Wolfgang Goethe-Universität zu Frankfurt am Main
10.2000-09.2001	Diplomarbeit an der Johann-Wolfgang Goethe

	Universität, Frankfurt am Main, Fachbereich Biologie, Arbeitskreis Neurobiologie circadianer Rhythmen, Betreuung durch Prof.Dr.G. Fleißner, Diplom mit Auszeichnung
10.2000-09.2003	Studium der Bioinformatik an der Johann-Wolfgang Goethe Universität, Frankfurt am Main
08.2000	Grundkenntnisse der Chronopharmakologie, Fakultät für Klinische Medizin Mannheim der Universität Heidelberg
02.2003-08.2003	Praktikum bei Sanofi Aventis GmbH, Frankfurt am Main, Abteilung Functional Genomics, Arbeitsgruppe Bioinformatik
09.2003	Beginn der Promotion an der Johann-Wolfgang Goethe Universität Institut für Informatik, Frankfurt am Main, Betreuung durch Prof.Dr.Dirk Metzler (seit Okt.2008: Ludwig-Maximilians Universität, München; zuvor: Institut für Informatik der Johann-Wolfgang Goethe Universität, Frankfurt am Main)

Eidesstattliche Erklärung

Die selbständige Anfertigung dieser Arbeit erkläre ich an Eides statt.

Frankfurt am Main, den 01.10.2009

(Claudia Döring)