Genome Expression Pathway Analysis Tool
Analyse und Visualisierung von Microarray Genexpressionsdaten unter genomischen, proteomischen und metabolischen Gesichtspunkten

Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Julius-Maximilians-Universität Würzburg

vorgelegt von
Markus Weniger
Würzburg

Würzburg 2007

Eingereicht am: .....

Mitglieder der Promotionskommission:

Vorsitzender: Prof. Dr. Martin J. Müller

Gutachter : Prof. Dr. Jörg Schultz

Gutachter: Prof. Dr. Rainer Spang

Tag des Promotionskolloquiums: .....

Doktorurkunde ausgehändigt am: .....

„Würdest du mir wohl sagen, wenn ich bitten darf, welchen Weg ich von hier nehmen muss?“
"Das hängt zum guten Teil davon ab, wohin du gehen willst," sagte die Katze.
"Es kommt mir nicht darauf an, wohin -; " sagte Alice.
"Dann kommt es auch nicht darauf an, welchen Weg du nimmst," sagte die Katze.
" -;wenn ich nur irgendwo hinkomme," fügte Alice als Erklärung hinzu.
"O, dass wirst du ganz gewiss," sagte die Katze, "wenn du nur lange genug gehst."

Lewis Carroll, Alice im Wunderland [1]

Inhaltsverzeichnis

1 Einleitung 4
1.1 Hintergrund 4
1.2 Ziele 6

2 Methoden 8
2.1 Genexpression 8
2.1.1 Zelle 8
2.1.2 Proteine 9
2.1.3 DNA 9
2.1.4 RNA 11
2.1.5 Transkription 11
2.1.6 Translation 12
2.1.7 Regulation der Genexpression 13
2.2 Microarray-Technik 15
2.2.1 Microarrays 15
2.2.2 Experimentdesign 21
2.2.3 Transkriptom und Proteom 23
2.2.4 Qualität der Messung 24
2.2.5 Andere Messmethoden 26
2.2.6 Vergleichbarkeit von Arrayplattformen 27
2.2.7 Comparative Genome Hybridization 28
2.3 Datenbanken 29
2.3.1 Ensembl 29
2.3.2 STRING 31
2.3.3 KEGG 33
2.3.4 Gene Ontology 34
2.4 Statistische Auswertung 36
2.4.1 Datenrepräsentation 36
2.4.2 Fehlende Werte 36
2.4.3 Normalisierung 37
2.4.4 Differentielle Expression 41
2.4.5 Clustering 44
2.4.6 GO Term Enrichment Analyse 48
2.5 Datenhaltung 49
2.5.1 MIAME-Standard 49
2.5.2 ArrayExpress 53
2.5.3 Gene Expression Omnibus 54
2.6 Java Platform, Enterprise Edition 55

2.6.1 Drei-Schicht-Architektur....56
2.6.2 JavaBeans.... 58
2.6.3 Enterprise JavaBeans.... 58
2.6.4 Java Servlets.... 59
2.6.5 Java ServerPages.... 61
2.6.6 Java ServerFaces.... 63
2.6.7 Web Services.... 66

3 Resultate.... 67
3.1 Server.... 67
3.2 Benutzeroberfläche.... 70
3.3 Datenhaltung.... 73
3.3.1 Expressionsdaten.... 73
3.3.2 Datenbanken.... 74
3.3.3 Benutzerverwaltung.... 74
3.4 Module.... 75
3.4.1 Teilmengenkonzept.... 78
3.4.2 Datenimport 81
3.4.3 Annotation.... 81
3.4.4 Analyse, Interpretation und Visualisierung.... 82
3.4.5 Geninformation.... 82
3.4.6 Help & Tutorial.... 83
3.5 Datenimport & Annotation.... 83
3.5.1 Dateneingabe.... 83
3.5.2 Fehlende Werte.... 86
3.5.3 Normalisierung.... 86
3.5.4 Annotation.... 90
3.6 Ensembl.... 91
3.6.1 Geninformation.... 92
3.6.2 Proteininformation.... 92
3.6.3 Literaturreferenzen.... 93
3.7 Datenanalyse.... 95
3.7.1 Differentielle Expression.... 95
3.7.2 Varianz & Median.... 96
3.7.3 Clustering.... 97
3.8 Chromosomale Daten.... 100
3.8.1 Chromosomposition.... 100
3.9 Gene Ontonolgy.... 102
3.9.1 GO Term Enrichment Analysis.... 102
3.9.2 OBO.... 105
3.10 KEGG Stoffwechselkarten.... 105
3.10.1 Datenbank.... 108
3.11 STRING.... 109

3.11.1 Assoziierte Gene 112
3.11.2 Tarbase & Drugbank 113

4 Fallstudie 114
4.1 Medizinische Grundlagen 114
4.1.1 B-Zell Lymphome 114
4.1.2 Non-Hodgkin-Lymphome 114
4.1.3 Diffuses großzelliges B-Zell-Lymphom 115
4.2 Analyse 117
4.2.1 IPI 117
4.2.2 Microarray-Untersuchungen 117
4.3 Analyse mit GEPAT 119
4.3.1 Differentielle Expression 120
4.3.2 GO Analyse 126
4.3.3 CGH Analyse 128

5 Diskussion 131

6 Zusammenfassung 137

7 Summary 138

Stichwortverzeichnis 139

Abbildungsverzeichnis 141

Tabellenverzeichnis 143

Literaturverzeichnis 144

Eigene Publikationen 157

Danksagung 158

Lebenslauf 159

Erklärung 160

1 Einleitung

1.1 Hintergrund

Die Genexpressionsanalyse mit Microarrays erlaubt neue Einblicke in die Funktionsweise lebender Zellen und hat die biologische Forschung in vielen Bereichen revolutioniert. Die Transkriptionsaktivität eines komplexen biologischen Systems, der Zelle, kann mit einem Mal ausgelesen werden und es wird möglich, die Quantität der mRNA-Expression für jedes einzelne Gen zu bestimmen. Durch diese Fähigkeiten sind Microarrays zum Standardwerkzeug in der Biologie und Medizin geworden. Ihre Anwendung reicht von der Erkennung von aktiven Genen während den Phasen des Zellzyklus über die Klassifizierung von Krankheitstypen bis hin zur Entwicklung neuer Medikamente. Eine große Herausforderung beim Einsatz von Microarrays stellt die Auswertung der enormen Datenmengen dar, die bei einem Experiment entstehen [2]. Spezielle Softwarepakete wurden zur Analyse dieser Daten entwickelt, jedoch besitzen viele dieser Programme Defizite im Bereich der Benutzeroberfläche und Integration. Der Großteil der derzeit verfügbaren Programme teilt die Auswertung der Messwerte in zwei Teile: Zum einen in die Datenanalyse, die verwendet wird, um eine Liste von statistisch interessanten Genen zu ermitteln, zum anderen in die Dateninterpretation, die diese Listen auf biologische Relevanz hin untersucht. Obwohl diese beiden Schritte eigentlich miteinander verwoben sein sollten, fehlt den meisten Programmen eine solche Integration.

Eine der am häufigsten verwendeten Softwarelösungen ist das Bioconductor-Toolkit [3], das auf dem Statistikpaket R [4] basiert. Nahezu jeder Algorithmus, der zur Microarray-Datenanalyse entwickelt wurde, ist durch Bioconductor verfügbar, meist kann auch der Quellcode eingesehen und verändert werden. Bioconductor muss jedoch durch Texteingabe auf der Kommandozeile bedient werden, eine einheitliche graphische Benutzeroberfläche existiert nicht. Deshalb stehen die verfügbaren Analysemethoden und die durch zahlreiche Bibliotheken gegebene Erweiterungsmöglichkeit nur erfahrenen Benutzern zur Verfügung. Benutzer, die keine Erfahrung in R und nur wenige Erfahrungen mit Programmiersprachen im Allgemeinen besitzen, werden unweigerlich auf Schwierigkeiten bei der Benutzung stoßen. Im schlechtesten Fall können diese Schwierigkeiten zur Missinterpretation von Resultaten führen, wenn unklar ist, mit welcher Art von Daten gearbeitet wird, wie Analysen ausgeführt werden und wie die

Ergebnisse zu deuten sind.

Um eine einfachere Benutzung des Bioconductor-Toolkits zu ermöglichen, wurden verschiedene Programme entwickelt, welche Module von Bioconductor in eine graphische Benutzeroberfläche verpacken. AMDA [5] ist ein R-Paket, das eine solche graphische Benutzeroberfläche zur Auswertung von Affymetrix-Microarraydaten bereitstellt. CARMAWeb [6] ist ein internetbasiertes Benutzerinterface, das Bioconductor-Module im Internet verfügbar macht. Beide Pakete machen den Zugang zu R einfacher, die Einstiegshürde für unerfahrene Benutzer ist aber auch hier noch hoch.

Neben diesen Ansätzen, die alle auf den Algorithmen von Bioconductor basieren, sind noch andere Analyse-Tools verfügbar, die eigene Algorithmen implementieren. Expression Profiler [7] ist ein integrierter, internetbasierter Ansatz zur Microarray-Datenanalyse. Auch die Microarray-Auswertung mit GEPAS [8] erfolgt im Internet. Es bietet eine größere Funktionalität als Expression Profiler, die Benutzerschnittstelle entspricht aber nicht mehr dem Stand der Technik und verlangt deshalb eine längere Einarbeitung. Andere Programme sind nicht internetbasiert, sondern werden auf dem lokalen Rechner installiert. EXPANDER [9] bietet einen leistungsfähigen Bicluster-Algorithmus, TM4 [10] ist eine Sammlung aus 4 Programmen, die alle Gebiete der Microarray-Analyse abdecken.

All diesen Programmen ist gemein, dass sie zwar die statistische Analyse der Microarray-Messwerte unterstützen, aber kaum Möglichkeiten zur biologischen Interpretation der Resultate bieten. Nur GEPAS unterstützt mit BABELOMICS [11] die biologische Interpretation, aber auch hier fehlt eine umfassende Integration mit dem Analyseteil. Allerdings gibt es auch eine ganze Reihe an Werkzeugen die ausschließlich die Interpretation der Daten unterstützen. Viele dieser Programme, z.B. WebGestalt [12], bieten einen automatischen, ontologiebasierten Ansatz, um die Häufung bestimmter biologischer Begriffe im Datensatz feststellen zu können. Andere Programme wie MAPPFinder [13], GFINDer [14], PathwayExplorer [15] und DAVID [16] untersuchen Genlisten auf weitergehende biologische Funktionen wie das Vorkommen in Stoffwechselwegen oder die Häufigkeit bestimmter Proteindomänen. Das Ensembl-Annotationssystem ENSMART [17][18] ermöglicht es dem Benutzer, Genom-Informationen für eine Liste von Genen zu finden, unterstützt aber nicht bei der Auswertung. Cytoscape [19] erlaubt es, die Genexpressionswerte von Microarrayexperimenten auf beliebige Graphen zu übertragen. All diesen Werkzeugen

fehlt die Möglichkeit einer integrierten Analyse. Sie benötigen vom Benutzer definierte Genlisten als Eingabe und machen daher andere Werkzeuge zur Normalisierung, zum Clustering und zur Bestimmung der Genlisten notwendig.

1.2 Ziele

Zur Interpretation der Microarray-Analyse-Resultate ist daher das übliche Vorgehen, zuerst eine Liste von Genen mit einem Analyse-Programm zu erstellen und diese Liste anschließend mit einem Interpretationsprogramm nach biologischer Information zu untersuchen. Die Ergebnisse dieser Untersuchung müssen dann wieder mit dem Analyse-Programm verifiziert werden. Dieses Vorgehen ist für eine kleine Anzahl an Experimenten praktikabel, aber langwierig und kompliziert, falls eine größere Zahl an Experimenten untersucht werden sollen.

Da für diese Trennung zwischen Interpretation und Analyse keine Notwendigkeit besteht, wurde ein Tool entwickelt, das beide Schritte beherrscht und miteinander verbinden kann. GEPAT bietet Methoden zur Datenanalyse, integriert mit Methoden zur Genom-, Expressions- und Stoffwechselwegsinterpretation. Ziel ist es, die Untersuchung von Genexpressionsdaten in das zelluläre Regulations- und Interaktionsnetzwerk zu integrieren. Dafür wurden diverse biologische Datenbanken integriert, um sie direkt zur Datenanalyse und -Interpretation heranziehen zu können.

GEPAT implementiert verschiedene Module, um diese Interpretation von Genen und Genmengen im zellulären Kontext zu unterstützen. Jedes dieser Module arbeitet mit allen anderen Analysemodulen und anderen Interpretationsmodulen zusammen, dadurch kann jede Art von Analyse und Interpretation auf einer beliebigen Teilmenge der Daten durchgeführt werden. Diese Integration ist einer der Hauptgesichtspunkte beim Entwurf von GEPAT gewesen und unterscheidet es von vielen anderen verfügbaren Programmen zur Analyse von Genexpressionsdaten. Da die Dateninterpretation in GEPAT vollkommen modular aufgebaut ist, ist auch eine Erweiterung mit neuen, bisher nicht implementierten Analyse- und Interpretationsmethoden einfach möglich. Von den Ergebnissichten aus ist es stets möglich, Informationen über die einzelnen Gene anzeigen zu lassen.

Da die meisten integrierten Datenbanken nicht öffentlich zugänglich sind und teilweise eine aufwändige Installation auf einem lokalen Rechner erfordern, wurde GEPAT als Internet-Applikation entwickelt. Dennoch wurde Wert auf eine für den Benutzer bekannte

Oberfläche gelegt, die an eine normale Desktop-Anwendung angelehnt ist. Das System wird als Webserver installiert und erlaubt so entweder die Benutzung auf einem einzelnen System (dem Server selbst), die Benutzung durch eine kleine Arbeitsgruppe im Intranet oder, in Verbindung mit einem Rechencluster, die Benutzung durch größere Arbeitsgruppen im Internet. GEPAT wird unter der LGPL [20] verbreitet und kann frei bezogen werden [21]. Für einen einfachen Einstieg, auch für unerfahrene Benutzer, stellt GEPAT für die ersten Schritte Videotutorials zur Verfügung und besitzt eine umfangreiche Online Hilfe. Ein Gast-Login ermöglicht einen Überblick und enthält vorbereitete Datensätze, die zur Klassifikation von B-Zell-Lymphomen [22] und zur Bestimmung von Genexpressions-Signaturen [23] verwendet wurden, kombiniert mit Daten über chromosomale Änderungen [24] in B-Zell Lymphomen.

2 Methoden

2.1 Genexpression

2.1.1 Zelle

Die Zelle ist die kleinste lebende Einheit. Trotz enormer Unterschiede zwischen den verschiedensten Arten von Zellen sind alle Zellen im Wesentlichen nach dem gleichen Prinzip aufgebaut. Zellen werden von einer Membran umgrenzt, die vom *Cytoplasma* ausgefüllt wird. Das Cytoplasma besteht aus dem *Cytoskelett* und den *Zellorganellen*, der Bereich dazwischen wird vom *Cytosol* ausgefüllt.

Die Unterschiede zwischen verschiedenen Zelltypen entstehen durch unterschiedliche Proteine, die durch die Zelle produziert werden, sowie durch deren Regulation. Bauplan aller Zellproteine ist die DNA, eine langkettige Nukleinsäure, die alle Funktionen einer Zelle kodiert. Diese Informationen werden in einem Prozess, der *Transkription* genannt wird, in mRNA umgeschrieben, die mRNA wird in der *Translation* in Proteine umgewandelt. Transkription und Translation sind ein wesentlicher Teil der Genexpression, der Ausprägung vom Genotyp zum Phänotyp.

Zellen lassen sich in zwei verschiedene Kategorien einteilen:

- *Prokaryoten* besitzen keinen echten Zellkern, die genetische Information wird in einem DNA-Molekül gespeichert, das sich frei im Zellplasma befindet. Es ist hier meist ringförmig und existiert nur in haploider Form. Transkription und Translation finden im Cytosol statt.
- Als *Eukaryoten* werden alle Lebewesen mit Zellkern zusammengefasst. Der Zellraum der Eukaryotenzelle ist in verschiedene Kompartimente eingeteilt. Der Großteil der genetischen Information ist hier diploid in Form von *Chromosomen*, einer Mischung aus DNA und Proteinen, im Zellkern (*Nukleus*) enthalten, zusätzliche DNA findet sich in den Mitochondrien und, bei Pflanzen, in den Chloroplasten. Die Transkription der genetischen Informationen findet im Zellkern statt, die Translation an den Ribosomen im Cytosol.

2.1.2 Proteine

Proteine sind langkettige Aminosäuremoleküle, die durch Peptidbindungen verbunden sind. Die Wirkungsweise eines Proteins wird durch seine dreidimensionale Struktur bestimmt. Oft lagern sich mehrere Proteine zu einem Proteinkomplex zusammen, um bestimmte Aufgaben zu erfüllen. Die Aktivität von Proteinen, sowie ihre Bindung an andere Moleküle, kann durch verschiedene Regulationsmechanismen eingeschränkt werden, z.B. durch Phosphorylierung durch eine Proteinkinase oder über Blockierung und Regulation durch kleine Moleküle.

2.1.3 DNA

Nukleinsäuren wie die DNA und RNA sind sehr lange Polymerketten, Grundelemente dieser Ketten sind die so genannten *Nukleotide*. Ein Zuckermolekül, die Desoxyribose bei der DNA, die Ribose bei der RNA, ist an einen Phosphatrest angelagert. Das Zuckermolekül besteht aus fünf Kohlenstoff-Atomen, die von 1' bis 5' benannt sind. An das 1'-Kohlenstoffatom ist ein weiteres organisches Molekül gebunden, die so genannte *Base*. Einzelne Nukleotide unterscheiden sich nur in der Art der gebundenen Base. Nukleotide sind in Nukleinsäuren durch eine Phosphatesterbrücke zwischen dem 3'-Kohlenstoff einer Einheit und dem 5'-Kohlenstoff einer anderen Einheit verbunden, diese Verbindung bildet das Gerüst für die Kette. Aufgrund dieser Verbindung besitzen DNA-Moleküle eine eindeutige Orientierung mit jeweils einem freien 3'- und 5'-Ende. Die Abfolge der Basen eines DNA-Strangs wird die Sequenz des Strangs genannt.

James Watson und Francis Crick entdeckten 1953, dass DNA-Moleküle aus Doppelsträngen bestehen, die eine Helix-Struktur ausbilden [25]. Diese Helix-Struktur besteht immer aus zwei Nukleotidsträngen, deren Basen über Wasserstoffbrücken miteinander verknüpft sind. Adenin steht in dieser Struktur immer Thymin gegenüber und Cytosin ist immer mit Guanin gepaart. Diese Paare sind als Watson-Crick-Basenpaare bekannt. Basenpaare, abgekürzt bp, liefern die Längeneinheit, die für DNA-Moleküle verwendet wird. Die beiden miteinander verbundenen Stränge besitzen eine gegenläufige Orientierung. Das 3'-Ende des eines Strangs steht dem 5'-Ende des anderen Strangs gegenüber.

Bei Prokaryoten liegt die DNA direkt im Zellplasma vor. Sie ist hier meist ringförmig und existiert nur in haploider Form. Bei Eukaryoten liegt die DNA in Form von diploiden

Chromosomen vor. Die Chromosomen sind fadenförmige Strukturen, die aus einem Komplex von Nukleinsäuren und Proteinen, der *Chromatin* genannt wird, bestehen. Im Chromatin ist die DNA um Proteine, die *Histone* gewickelt. Histone sind kleine, basische Proteine, die aus einem globulären Zentrum und flexiblen endständigen Schwänzen bestehen. DNA und Histone bilden die *Nukleosomen*, die aus einem Histon-Kern, einem Linker-Histon und 160-200 bp DNA bestehen. Die Histone sind wahrscheinlich für die superspiralförmige Anordung der DNA in der sogenannten 30-nm-Faser verantwortlich. Abbildung 1 illustriert die Struktur der Chromosomen.

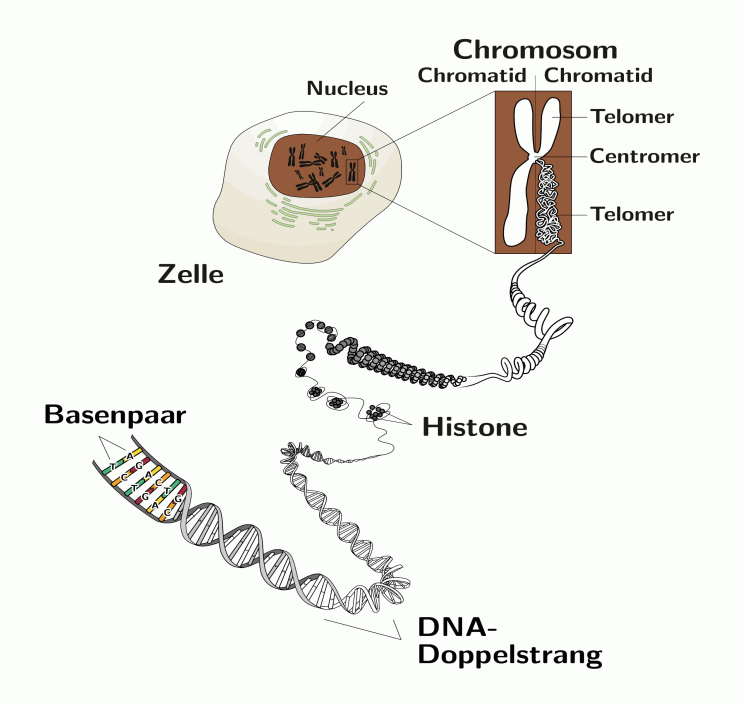


Abbildung 1: Von der DNA zum Chromosom.
Die Primärstruktur der DNA ist die Abfolge der Basenpaare, die Sekundärstruktur die Anordnung zur DNA Doppelhelix. Die Doppelhelix ist um die Histone gewickelt, die zu einer superspiralförmigen Anordnung der DNA führen. Histone und weitere Proteine bilden das Chromatin, aus dem die Chromosomen bestehen. Nach [26].

Die für die Erzeugung eines Proteins verantwortliche Basensequenz der DNA wird *Gen* genannt. Die gesamte Erbinformation eines Lebewesens wird *Genom* genannt. Ein kurzes Genom ist ungefähr eine Million Basen lang, die Sequenz eines typischen Bakteriengenoms ist mehrere Millionen Basen lang. Das menschliche Genom besteht aus ungefähr 3,5 Milliarden Basen, für die 21724 Gene bekannt sind, die 44567 Proteine kodieren (Ensembl 44.36f).

2.1.4 RNA

Die RNA ähnelt im Aufbau der DNA, ist aber im Regelfall einsträngig. Im Gegensatz zur DNA enthält sie eine zusätzliche Hydroxyl-Gruppe an der 2'-Position und es wird die Base Uracil anstelle von Thymin verwendet. Man unterscheidet verschiedene RNA-Moleküle, die unterschiedliche Funktionen ausüben, darunter sind:

- *mRNA* (messenger RNA) wird aus dem zu einem Gen gehörenden Abschnitts der DNA produziert. Bei Prokaryoten wird dieser Abschnitt vom Enzym RNA-Polymerase während der Transkription direkt abgelesen und in mRNA umgewandelt, bei Eukaryoten entsteht die mRNA durch Spleißen der hnRNA.
- *hnRNA* (heterogeneous nuclear RNA) ist in Eukaryoten der Vorläufer der mRNA. Sie entsteht bei der Transkription und wird durch Spleißen in die mRNA umgewandelt.
- *tRNA* (transfer RNA) dient als Hilfsmolekül bei der Proteinsynthese, indem sie die für die Translation benötigten Aminosäuren an den Ribosomen bereitstellt
- *ncRNA* (non-conding RNA) sind RNA Moleküle, die vom Genom gelesen, aber nicht in Proteine umgeschrieben werden. Sie umfassen die siRNAs und microRNAs mit einer Länge von 18-25 bp, die small RNAs mit einer Länge von bis etwa 300 bp und medium und large RNAs bis mehrere 10000bp Länge. Die Bedeutung der ncRNA ist noch nicht vollständig bekannt, wahrscheinlich ist sie an der Regulation von zellulären Vorgängen beteiligt [27].
- *siRNA* sind etwa 21-28 bp lange einsträngige RNAs, die aus langen doppelsträngigen RNAs durch ein Enzym herausgeschnitten werden. Die siRNA werden in einen Proteinkomplex eingebaut, führen mit diesem zum Abbau von bestimmten mRNAs und verhindern so die Translation des entsprechenden mRNA-Strangs [28].

2.1.5 Transkription

Unter Transkription versteht man die Synthese von RNA-Molekülen aus der DNA. Die Sequenzinformation der DNA wird dabei in RNA umgeschrieben. Dazu wird die DNA-Doppelhelix aufgetrennt und durch Anlagerung komplementärer Ribonukleotide an den codogenen Strang in 3' nach 5' Richtung wird die RNA in 5' nach 3' Richtung synthetisiert.

Bei Prokaryoten findet die Transkription im Cytosol statt und noch während der mRNA-Strang synthetisiert wird, kommt es zur Translation am Ribosom. Bei Eukaryoten findet die Transkription im Zellkern statt. Hier entsteht zuerst die hnRNA, die neben proteinkodierenden Regionen, den *Exons*, auch nichtproteinkodierende Regionen, die *Introns*, enthält. Die Introns werden in einem *Spleißen* genannten Vorgang vom Spleißosom, einem Komplex aus RNAs und Proteinen, aus der hnRNA ausgeschnitten, das 5' Ende wird gekappt, die mRNA entsteht. An das 3' der mRNA wird ein bis zu mehrere hundert bp langer Schwanz aus Adenin-Nukleotiden angehängt. Die nun polyadenylierte mRNA wird aus dem Zellkern zu den Ribosomen im Cytosol transportiert, wo die Translation stattfindet. Überwachungsmechanismen sorgen dafür, dass mRNA nach einer gewissen Zeit oder bei Mutationen abgebaut wird.

Das Zentrale Dogma der Molekularbiologie [29] legt die DNA als alleinigen Träger der genetischen Informationen fest, eine Umwandlung von RNA in DNA erfolgt nicht. Im Gegensatz zum Genom ist das Transkriptom stark dynamisch und verändert sich rapide in Antwort auf Signale von außen oder während normaler zellulärer Ereignisse, wie DNA-Replikation oder Zellteilung [30].

2.1.6 Translation

Die mRNA enthält alle Informationen, die zum Aufbau eines Proteins notwendig sind. Diese Information wird in der Translation zur Proteinsynthese genutzt. Die Proteinsynthese findet am Ribosom statt und benötigt neben der mRNA als Vorlage auch noch tRNAs, die als Transporter für die Aminosäurebausteine der Proteine dienen. Für je 3 aufeinander folgende Nukleotide der mRNA steht je eine bestimmte Aminosäure. Dieses Basentriplett wird *Codon* genannt.

Beginnend bei einem Startcodon benutzt das Ribosom das entsprechende tRNA-Molekül für das nächste Codon der mRNA, um die bereits bestehende Polypeptidkette mit der Aminosäure der tRNA zu verlängern. Nach der Verlängerung wird die mRNA um 3 Basen verschoben und die Synthese fährt mit dem nächsten Codon fort.

Die Proteinfaltung erfolgt entweder spontan bereits bei der Entstehung am Ribosom, oder spezielle Proteine, die *Chaperone*, unterstützen das Protein bei der Faltung.

2.1.7 Regulation der Genexpression

Genexpression, auch Expression oder Exprimierung genannt, bezeichnet die Transkription der DNA in die RNA, also die Ausprägung des Genotyps zum Phänotyp einer Zelle. Jede eukaryotische Zelle enthält genetische Information für zwischen 5 000 – 60 000 proteinkodierende Gene, die allerdings nicht alle gleichzeitig aktiv sind. Vielmehr werden nur Teile dieser Gene in RNA transkripiert. Die Menge der exprimierten Gene eines Genoms wird auch als Transkriptom bezeichnet. Die Gründe für die Expression bestimmter Gene sind unter anderem:

- Gewebeart
- Entwicklungsstand des Organismus
- Antwort auf Umgebungsbedingungen
- Krankheiten
- Genveränderungen durch Mutation

Der Genexpression liegt eine feine Steuerung zugrunde, die im Folgenden beschrieben wird. Grundlage ist die Initialisierung der Transkription durch das Enzym RNA-Polymerase an der DNA.

Als *Promotor* bezeichnet man den Abschnitt der DNA, der vor der proteinkodierenden Stelle liegt und der eine Sequenz enthält, welche die RNA-Polymerase zum Startpunkt eines Gens leitet. Er enthält die sogenannten *cis-Elemente*, die zusammen mit Transkriptionsfaktoren, den *trans-Elementen*, die Transkription regulieren. Die Transkriptionsfaktoren sind hierbei keine harten Ein- oder Ausschalter, sondern können fein gesteuert werden [31]. Die Gesamtheit der cis-/trans-Aktivitäten im Promoter bestimmt schließlich die Häufigkeit, mit der die RNA-Polymerase die Transkription durchführt.

In Prokaryoten besteht der Promotor mindestens aus zwei kurzen Sequenzen 10 bp und 35 bp vor Beginn des Gens. Die RNA-Polymerase und ein assoziierter σ-Faktor binden direkt an diese Sequenzen. Neben den cis-Elementen bestimmt auch die Sequenz dieser Region die Stärke des Promotors und damit die Häufigkeit, mit der das Gen transkripiert wird.

Auch Eukaryoten benötigen einen Promotor zum Start der Transkription. Hier bindet die

RNA-Polymerase jedoch nicht direkt an die DNA, sondern wird durch eine Gruppe von Transkriptionsfaktoren an die Startstelle dirigiert. Oft enthält der Promotor eine sogenannte *TATA Box* mit der Sequenz TATAA, an die sich ein TATA-Bindeprotein anlagert, das die Bildung des RNA-Polymerase-Komplexes unterstützt. Die TATA Box liegt normalerweise sehr nahe an der Transkriptionsstartstelle, in der Regel zwischen den Positionen -30 und -100. Zusätzliche Elemente liegen zwischen -40 und -150. Neben den Promotoren enthält das Genom von Eukaryoten noch zusätzlich *Enhancer*, welche die Anlagerung des Transkriptionskomplexes am Promotor beeinflussen und die Transkriptionsaktivität am Gen verstärken. Eine genaue Positionierung der Enhancer relativ zum Promotor ist nicht notwendig, sie kann erheblich variieren und kann viele tausend bp up- oder downstream des Promotors liegen. Durch die superspiralförmige Anordnung der DNA im Chromosom gelangt der Enahncer wieder in die räumliche Nähe des Promotors. Durch Anlagerung von Regulatorgenen an den Enhancer kann dessen Funktion auch eingeschränkt werden.

Nach der Transkription werden sowohl das 5' als auch das 3' Ende der hnRNA modifiziert. Aus der beim Ablesen entstehenden hnRNA werden die Introns herausgeschnitten. Die Spleißstellen sind durch Sequenzen an den Enden der Introns gekennzeichnet. Oft gibt es mehrere Möglichkeiten, aus der hnRNA eine proteinkodierende mRNA zu spleißen, man spricht hier von *alternativen Spleißen*. Dabei werden unterschiedliche Exons in die fertige mRNA übernommen. Das alternative Spleißen ist ein sehr wirksamer Weg, um die Sequenzen des Genoms vielfältiger zu nutzen.

Bei Eukaryoten findet die Transkription im Nukleus statt, die Translation im Cytosol. In manchen Fällen verbleibt die mRNA jedoch im Nukleus, um bei Bedarf entlassen zu werden, und der Zelle somit eine schnelle Reaktion auf geänderte Umweltbedingungen zu ermöglichen [32].

Wird auch der komplementäre Strang eines Gens abgelesen, entsteht die Antisense-RNA, die sich aufgrund ihres komplementären Aufbaus mit der entsprechende mRNA zu einem Doppelstrang verbindet. Dadurch sind die Bindungsstellen an der mRNA für die Ribosomen blockiert, die Translation kann nicht stattfinden. Die Regulation der Transkription von ncRNA funktioniert ähnlich der Regulation der Genexpression. Auch hier finden sich Transkriptionsfaktorbindestellen vor dem Abschnitt, der transkripiert wird [33].

2.2 Microarray-Technik

Durch die wachsende Anzahl der sequenzierten Genome und deren Sequenzinformationen stehen der Biologie neue und bisher unbekannte Möglichkeiten zur Verfügung, Informationen aus diesen Milliarden Basen zu gewinnen. Obwohl es möglich ist, Gene in den Sequenzen weitgehend automatisch zu finden, ist es leider nicht ohne weiteres möglich, die Funktion der Zelle, deren Entwicklung, das Zusammenspiel im Organismus oder die Reaktion auf Medikamente nur durch Betrachtung der Genomsequenz zu entschlüsseln. Ziel ist es deshalb, die Sequenzinformationen als Grundlage für weitere Techniken zu verwenden, die es erlauben, die Art der Zusammenarbeit der biologischen Komponenten zu bestimmen.

2.2.1 Microarrays

Neue Technologien helfen, aus der wachsenden Anzahl der Sequenzinformationen den größtmöglichen Nutzen zu schlagen. Eine der im Moment am häufigsten eingesetzten Hilfsmittel sind dichtgepackte RNA-Messpunkte aus Oligonukleotiden oder komplementärer DNA, die als DNA-Microarrays bezeichnet werden. Sie können das Expressionsniveau von vielen tausend mRNAs auf einmal erfassen und haben so eine Vielzahl von Untersuchungen rasant beschleunigt. Die Messung der Genexpression mit Microarrays wird in vielen Gebieten der Biologie und Medizin verwendet und dient zur Untersuchung von Krankheiten, Behandlungen und Entwicklungsstadien. Microarrays kommen seit Mitte der 1990er Jahre zum Einsatz [34][35] und wurden von der Forschungsgemeinschaft seitdem im stark wachsenden Umfang zur Untersuchung verschiedenster biologischer Prozesse eingesetzt. Einen Überblick über den Anstieg der Veröffentlichungen, die Microarray-Technik verwenden, zeigt Abbildung 2.

Microarrays funktionieren durch Hybridisierung einer Lösung aus fluoreszent markierter RNA oder DNA an DNA-Moleküle, die an einer definierten Stelle auf der Chipoberfläche aufgebracht wurden. Dies entspricht einer parallelen Suche, bei der jedes Molekül der Lösung einen passenden Partner auf dem Array sucht, der durch die Regeln der molekularen Bindung bestimmt wird. DNA Microarrays messen so die Menge an RNA, die in der Zelle vorhanden ist. Man nennt diese Art von Untersuchung Expressionsanalyse oder Expression Profiling.

Da Microarrays eine Vielzahl an Informationen gleichzeitig ermitteln, eigenen sie sich

nicht nur, um bestehende Hypothesen zu überprüfen, sondern können vielmehr auch verwendet werden, um Antworten auf biologische Fragen zu finden, z.B. um die Behandlung von Krankheiten besser zu verstehen. Eine frühe Anwendung von Microarrays war die Untersuchung von verschiedenen Arten von Tumoren. Proben von Patienten mit oder ohne akute Leukämie [36] und Patienten mit diffusen großzelligen B-Zell Lymphomen [22] wurden untersucht und mRNA-Marker wurden gefunden, die zur Klassifikation der Tumorarten geeignet sind. Die Resultate können verwendet werden, um die Therapien speziell auf den spezifischen Tumortyp des Patienten abzustimmen. Die Ergebnisse können auch dazu verwendet werden, um zu verstehen, wieso sich Körperzellen in Tumorzellen transformieren, und um die hierfür verantwortlichen Gene zu identifizieren. Gene, die im Tumor stärker exprimiert sind als im normalen Gewebe, sind ein potentionelles therapeutisches Ziel und nehmen häufig Einfluss auf andere Gene, die am selben Stoffwechsel- oder Signalweg beteiligt sind. Tumorbildung wird sehr häufig von Veränderungen der Chromosomen begleitet, wie Translokation, Amplifikation oder Verlust bestimmter Chromosomabschnitte. Auch die Änderung der Chromosomenanzahl spielt bei vielen Tumorarten eine Rolle. Die Untersuchung auf chromosomale Veränderungen kann verwendet werden, um Tumore zu klassifizieren und um Regionen zu finden, die tumorunterdrückende Gene enthalten können [24].

Die ersten Microarrays bestanden aus DNA-Fragmente, manchmal mit unbekannter Sequenz aus genomischer DNA oder Plasmid-Bibliotheken, die auf einer porösen Membran, meist Nylon, aufgetragen wurden. Mittlerweile besteht das Trägermedium der Microarrays meist aus Glas, auf dem einzelsträngige DNA-Fragmente mit unterschiedlicher Sequenz in Form eines geordneten Gitters fixiert werden. Neue Technologien zur Synthetisierung oder Auftragung von Nukleotiden auf die Glasscheiben ermöglichen sehr hohe Dichten, die zu einer Miniaturisierung der Arrays geführt haben und die damit die Experiment-Effizienz und den Informationsgehalt eines Experiments drastisch erhöhen. Damit ist es möglich, das gesamte Genom eines Organismus auf den Microarray aufzubringen, und so den aktuellen Zustand der Genexpression aller Gene in der Zelle zu bestimmen.

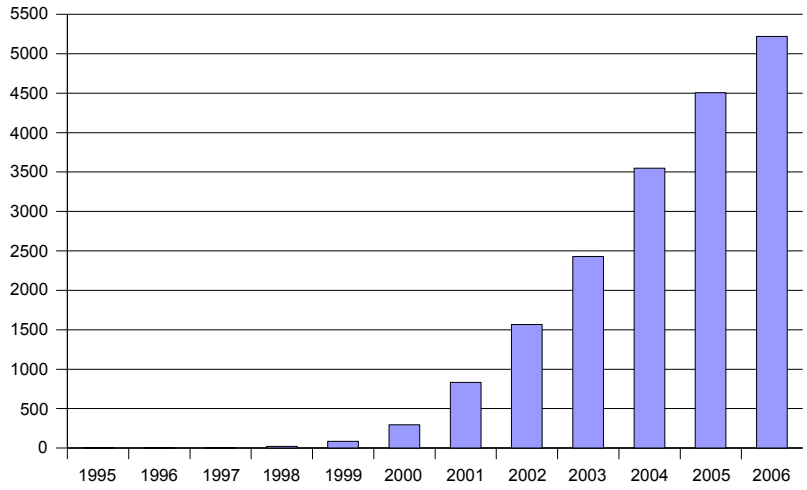


Abbildung 2: Veröffentlichungen mit Microarray-Technik
Das Diagramm zeigt die Anzahl der Publikation mit Microarray-Technik in den Jahren von 1995 bis 2006. Für jedes Jahr wurden die Veröffentlichungen, deren Zusammenfassungen in PubMed das Wort „Microarray" enthielten, gezählt.

Typischerweise befinden sich mehrere Nanogramm DNA als Gensonden auf dem Microarray. Gensonden können auf unterschiedliche Art und Weise erzeugt werden, z.B. durch cDNA-Klonierung von Genen oder ESTs (cDNA-Microarrays), durch Synthetisierung von DNA-Oligomeren (Oligomer-Arrays), oder durch Klonieren genomischer DNA Sequenzen (CGH-Arrays). Durch Hybridisierung, der Kombination von zwei komplementären, einsträngigen Nukleinsäuren in ein Molekül werden Gensonden und die in cDNA umgeschriebenen mRNAs der Proben miteinander verbunden. Da Nukleotide unter normalen Bedingungen ihr Komplement binden, binden sich hierbei je zwei komplementäre Stränge in einem *Annealing* genannten Prozess. Nach der Hybridisierung wird die restliche Lösung abgewaschen. Der Verlauf der Hybridisierung wird in Abbildung 3 illustriert.

Die mRNA-Fragmente müssen mit einer Markierung versehen werden, um ihre Intensität auslesen zu können. Zuerst wurden hierfür radioaktive Gruppen verwendet, mittlerweile hat sich die Verwendung von fluoreszierenden Farbstoffen durchgesetzt. Die Anzahl an Ziel-mRNA, die an jede Sonde gebunden ist, wird durch Scannen des Trägermediums ermittelt. Für fluoreszierende Markierungen wird ein Laser-Scanner verwendet, der Licht in einer Wellenlänge ausstrahlt, die von den fluoreszierenden Molekülen absorbiert wird. Durch die Absorption werden die Moleküle angeregt, Licht in anderen Wellenlängen auszustrahlen, das vom Scanner aufgezeichnet wird. Durch Bewegung des Lasers über

das Trägermedium kann anhand der Intensität des abgestrahlten Lichts die Menge der markierten Ziel-mRNA pro Position ermittelt werden. Die Intensität jedes Punktes auf dem Microarray steht für die Expression des entsprechenden Gens für die jeweilige Gensonde.

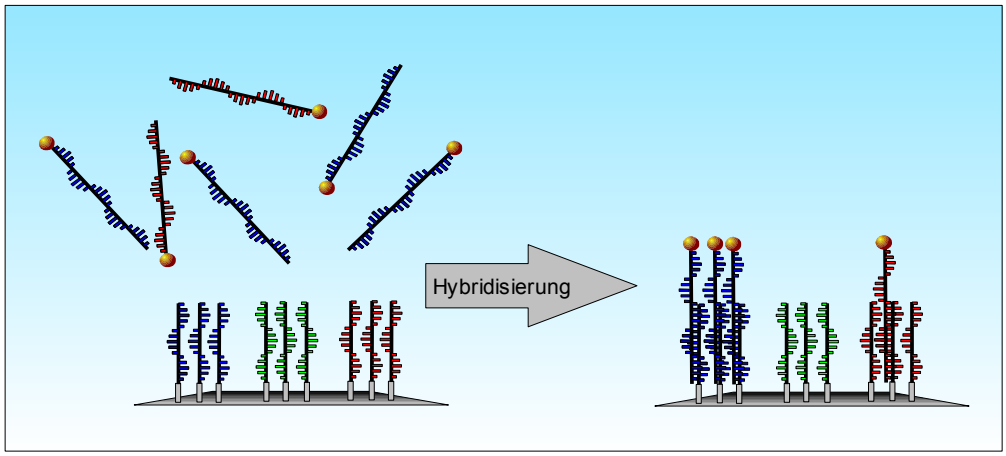


Abbildung 3: Hybridisierung
Bei der Hybridisierung binden die in cDNA umgeschriebenen mRNA-Transkripte an die entsprechenden Gensonden auf dem Microarray. Um das Messen der Intensität zu ermöglichen sind die cDNA-Moleküle mit einem Farbstoff markiert.

Man unterscheidet im Wesentlichen zwei verschiedene Typen von Microarrays:

- Bei cDNA Microarrays (Zweikanal-Microarrays) werden Gensonden aus komplementärer DNA (cDNA) mit speziellen Druckspitzen auf den Untergrund aufgebracht. Sie können automatisiert mit Array-Robotern im Labor hergestellt werden. Da die aufgebrachte Menge aufgrund technischer Einschränkungen nicht genau dosiert werden kann, wird hier meist das Expressionsverhältniss zweier Proben auf einem Array gemessen.
- Bei Oligonukleotid-Microarrays werden die Gensonden in situ auf den Untergrund synthetisiert. Oligomer-Microarrays werden meist kommerziell bezogen. Die Oligonukleotid-Microarrays unterscheiden sich durch die Länge der DNA-Sequenzen, man unterscheidet hier Typen mit kurzen Oligonukleotidsonden (20-30 bp) und lange Oligonukleotidsonden(50-80 bp).

Abbildung 4 illustriert die Unterschiede in der Verwendung der beiden Verfahren. Legt man als Kriterium den Wechsel in der Genexpression zugrunde und vernachlässigt die Stärke des Wechsels, so kommen die unterschiedlichen Plattformen zu vergleichbaren und reproduzierbaren Ergebnissen [37].

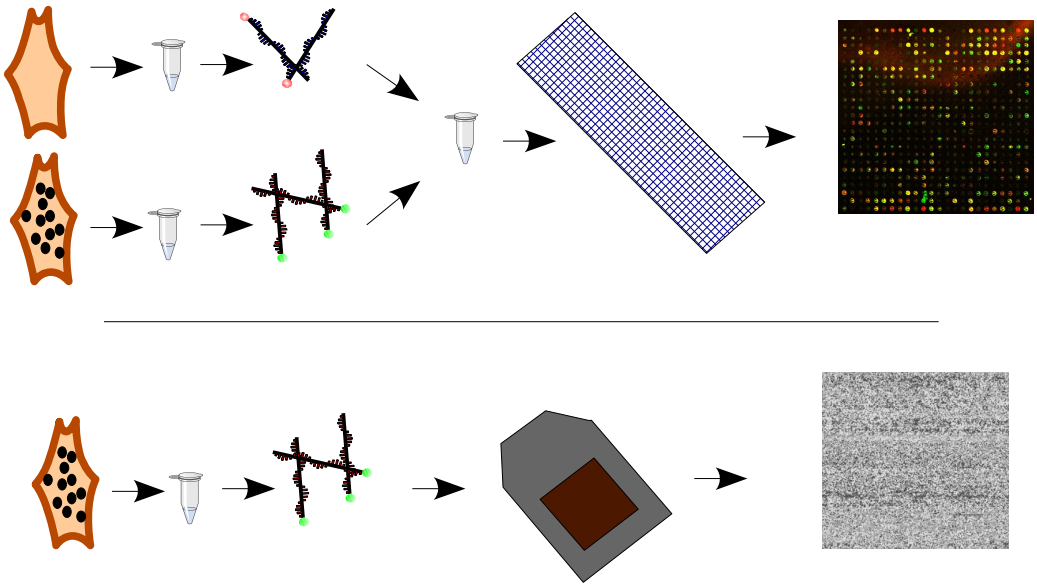


Abbildung 4: cDNA- und Oligonukleotid-Microarrays
Auf cDNA-Microarrays werden normalerweise die Expressionswerte zweier unterschiedlich markierter Proben vergleichen, Oligonukleotid-Microarrays ermitteln die Expressionswerte für eine Probe.

cDNA-Microarrays

Bei Zweikanal-Microarrays wird als Material für Gensonden meist cDNA verwendet, deren Sequenz Teilen der Sequenz der zu messenden mRNA entspricht. Die cDNA wird aus einer mRNA-Vorlage mithilfe des Enzyms *Reverse Transkriptase* synthetisiert. Die Microarray-Gensonden werden dabei nicht gleichzeitig auf das Array gebracht, sondern mit einer Druck-Spitze wird Schritt für Schritt eine Anzahl an Gensonden auf das Array aufgetragen. Bei diesem Verfahren kann die genaue Menge der aufgebrachten cDNA für eine Sonde nicht ohne weiteres festgestellt werden, daher ist es nicht möglich, die Genexpression absolut zu messen. Allerdings kann die Genexpression relativ zu einer anderen Probe verglichen werden. Deshalb wird diese Art von Microarrays normalerweise mit mRNAs von zwei verschiedenen Proben gleichzeitig hybridisiert. Die mRNA der Proben ist mit zwei unterschiedlichen Farbstoffen markiert, die Proben können so gemischt werden, und diese Mischung wird auf das Microarray hybridisiert. Diese Experimente werden oft auch als Zweikanal-Experimente bezeichnet, da die beiden Floureszenzstoffe als roter und grüner Farbkanal mit einem Laser ausgelesen werden. Hierbei wird oft der Wert als Logarithmus des Verhältnisses des roten Kanals zum grünen Kanal (log(R/G)) angegeben. Der Vorteil in der Verwendung des Logarithmus liegt darin, dass erhöhte und

verringerte Verhältnisse symmetrisch behandelt werden. Üblicherweise wird für den zweiten Kanal eine gemeinsame Referenzprobe verwendet, so dass bei Experimenten mit mehreren Microarrays die relativen Expressionswerte auf dem Array zu einer gemeinsamen Basis im Bezug stehen und damit direkt vergleichbar sind.

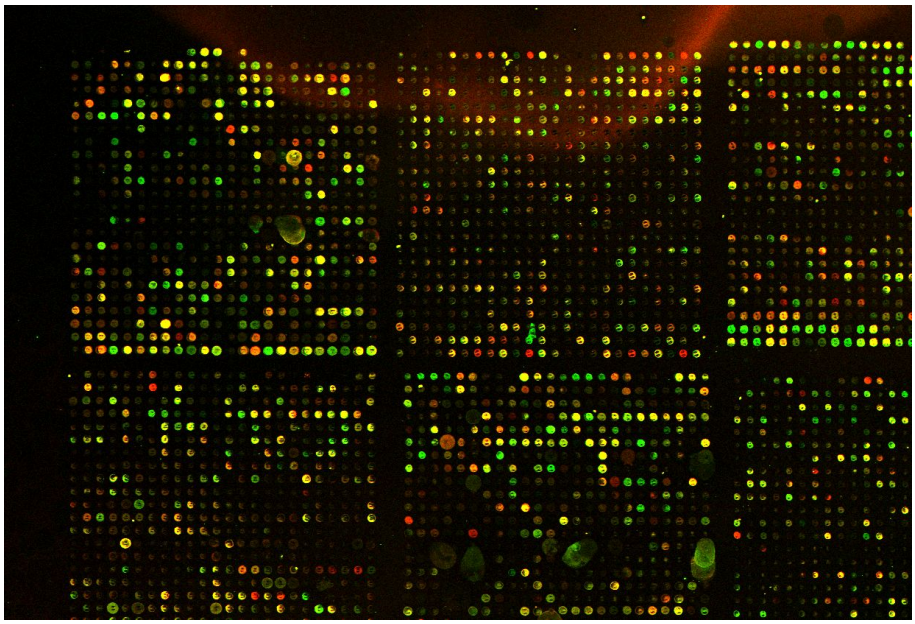


Abbildung 5: Scanergebnis eines cDNA Microarrays
Beide Kanäle, die aus Graustufenbildern für jeden Kanal erzeugt sind, wurden übereinandergelegt. Deutlich erkennbar sind Verschmutzungen auf dem Array

Die Rohdaten, die von Microarray-Experimenten erzeugt werden, sind Scanbilder der hybridisierten Microarrays, ein Beispiel zeigt Abbildung 5. Um Informationen über die Genexpression zu erhalten, werden die Bilder analysiert, jeder Punkt auf dem Array markiert, seine Intensität gemessen, mit der Hintergrundintensität verglichen und die Sequenz für die aufgedruckte Sonde annotiert. Dieser Vorgang wird als Image Quantation oder Image-Analyse bezeichnet.

Oligonukleotid-Microarrays

Bei Affymetrix-Oligonukleotid-Arrays [38], der derzeit am weitesten verbreiteten Microarray-Plattform, bestehen die Sonden aus etwa 25 bis 30 bp langen Nukleotidketten. Sie sind komplementär zu der Sequenz, die sie ermitteln sollen. Aufgrund der Kürze der Oligonukleotid-Sequenz werden mehrere Gensonden für jedes zu

messende Transkript verwendet, um eine erhöhte Spezifität zu erreichen. Affymetrix-Arrays verwenden typischerweise zwischen 11 und 20 Sondenpaare für jedes Gen. Eine Komponente dieser Paare ist die sogenannte Perfect Match Sonde (PM), die nur spezifisch mit den Transkripten des gewünschten Gens hybridisieren soll. Jedoch ist eine Kreuzhybridisierung mit anderen mRNA-Sequenzen nicht auszuschließen, man spricht dabei von nichtspezifischer Hybridisierung. Um die Intensitäten entsprechend korrigieren zu können wurde eine weitere Komponente, die sogenannte Mismatch Sonde (MM) eingeführt, die nur die nichtspezifische Hybridisierung zur entsprechenden PM Sonde messen soll. Affymetrix verwendet hierzu für MM die gleiche Sequenz wie bei PM, jedoch wird die 13te Base durch ihr Komplement ersetzt. Die Messwerte der zu einem Transkript gehörenden Gensonden werden zusammengefasst und auf diese Weise die Expression des entsprechenden Transkripts bestimmt, die in der weiteren Analyse verwendet wird.

Aktuelle Systeme besitzen genug Gensonden, um das mRNA-Level aller Gene eines Genoms zu erfassen. Im Gegensatz zu Zweikanal-Microarrays wird bei der Verwendung von Oligonukleotid-Microarrays jeweils nur die Expression einer Probe pro Array gemessen, die Messwerte sind dementsprechend Absolutwerte. Oligonukleotid-Arrays enthalten oft sogenannte RNA Spike-In Gensonden, die mit mengenmäßig bekannter Kontroll-RNA hybridisieren und anschließend zur Normalisierung verwendet werden können.

2.2.2 Experimentdesign

Zur Durchführung von Microarray-Untersuchungen gibt es verschiedene Möglichkeiten des Experimentdesigns. Bei Microarray-Systemen, die die Expression von zwei Proben vergleichen, werden die Transkripte mit Cy5 (einem roten Farbstoff) oder Cy3 (einem grünen Farbstoff) markiert. Die Farbstoffe werden durch einen Laser angeregt und die Intensität kann gemessen werden. Dieser Ansatz misst das Verhältnis der Genexpressions-Messwerten, nicht die absoluten Messwerte. Ein mögliches Experiment-Design ist der Vergleich von jeder Probe mit jeder anderen, um einen globalen Vergleich zu ermöglichen. Ein anderer Ansatz ist die Hybridisierung von je einer Probe und einer gemeinsamen Referenz auf je einem Microarray. Die Referenz kann hierbei aus einem Gemisch der verschiedenen Proben bestehen. Abbildung 6 zeigt mögliche Experimentdesigns. Ein *Dye-Swap* genanntes Verfahren kann zur Kontrolle eingesetzt

werden. Die Proben werden hierbei je zweimal markiert, einmal mit Cy5, einmal mit Cy3, und in unabhängigen Hybridisierungen verwendet. Dye-Swap hilft, Variationen zu kompensieren die durch unterschiedliche Effizienz der Farbstoffe entstehen.

Man unterscheidet zwei verschiedene Arten von Studien, für die Microarray-Analysen durchgeführt werden: Bei statischen Datensätzen entspricht jedes Array einer anderen biologischen Probe, z.B. Gewebe von unterschiedliche Patienten. Clustering und andere Analysemethoden werden verwendet, um die Daten zu klassifizieren, und um Beziehungen zwischen der Genexpression und verschiedenen Proben zu erkennen. Bei dynamischen Datensätzen entspricht jedes Microarray einem Zeitpunkt, zu dem eine biologische Probe genommen wurde, z.B. eine Zellkultur zu unterschiedlichen Stadien. Hier werden Analysemethoden verwendet, um ein gemeinsames Verhalten der Gene über der Zeit zu erkennen.

Die Daten, die durch die Microarray Datenanalyse aus den verschiedenen Arrays entstehen, werden in einem *Normalisierung* genannten Prozess auf eine gemeinsame Skalierung gebracht. Die Mittelwerte und Varianzen werden korrigiert, um Effekte, die aus systembedingten, nicht-biologischen Unterschieden zwischen den Arrays, Subarrays und Farbkanälen entstehen, zu korrigieren.

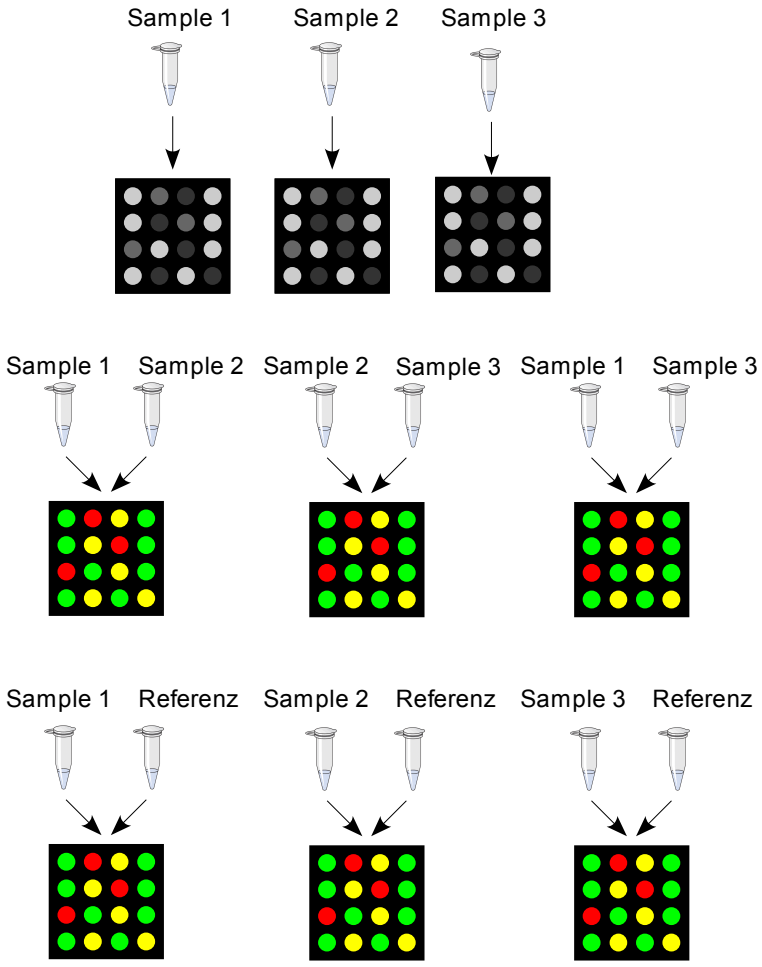


Abbildung 6: Möglich Experimentdesigns für Microarray-Studien
Bei Oligonukleotid-Microarrays (oben) wird eine Probe pro Array vergleichen. Bei cDNA-Microarrays können alle Proben miteinander verglichen werden (Mitte). Bei Studien mit vielen Proben wird zum Vergleich meist eine gemeinsame Referenzprobe auf allen Microarrays verwendet (unten)

2.2.3 Transkriptom und Proteom

Microarray-Experimente messen die Anzahl von mRNA-Transkripten, die bereits durch Mechanismen wie Transkriptionsfaktoren, Beeinflussung durch ncRNA, oder DNA-Methylierung reguliert worden ist. Die expressionsbasierte Technologie ist nur in der Lage, zelluläre Antworten auf der RNA-Ebene zu messen. Frühe Studien fanden nur geringe Korrelation zwischen mRNA-Menge und Proteinmenge, neueren Studien, möglicherweise bedingt durch bessere Methoden zur Proteomanalyse, konnten eine bedingte Korrelation der Mengen [39][40] bestätigen. Die Transkription ist jedoch nicht der einzige Vorgang in der Zelle, der reguliert wird, so dass von der mRNA-Menge nicht zwingend auf die Proteinmenge und Proteinaktivität geschlossen werden kann. Einige

regulatorische Mechanismen finden auch posttranslational auf Proteinebene statt [41][42] und können so nicht von Microarrays detektiert werden. Auch andere zelluläre Aspekte können bei der Expressionsanalyse mit Microarrays nicht erkannt werden, z.B. Deaktivierung von Proteinfunktionen durch Mutationen oder Konformationsänderungen durch Phosphorylierung oder die Lokalisation des fertigen Proteins in der Zelle. Ein Überblick über das Verhältnis zwischen Transkriptom und Proteom liefert [43]. Für die Mehrzahl der mRNA Transkripte geht man jedoch davon aus, dass eine Änderung in der Expression eine Änderung der Proteinmenge hervorruft, die zu einem geänderten Phänotyp führt. Abbildung 7 zeigt ein Schema dieser Annahme.

Bei der Auswertung der Transkriptionsprofile ist zwischen primären und sekundären Effekten zu unterscheiden. Eine anfängliche Beeinflussung des biologischen Systems führt zu Änderungen der Genexpression, die von weiteren, sekundären Änderungen gefolgt wird. Dabei ist es nur durch geschicktes Experiment-Design und der Verwendung mehrerer Zeitpunkte möglich, die Beziehung zwischen Ursache und Wirkung zu ergründen. Bei der Expression von Transkriptionsfaktoren ist in Erwägung zu ziehen, dass sich diese nach ihrer Erzeugung im Cytosol befinden und erst noch in den Zellkern transportiert werden müssen. Auch bei diesem Vorgang finden sich weitere Möglichkeiten zur Regulation.

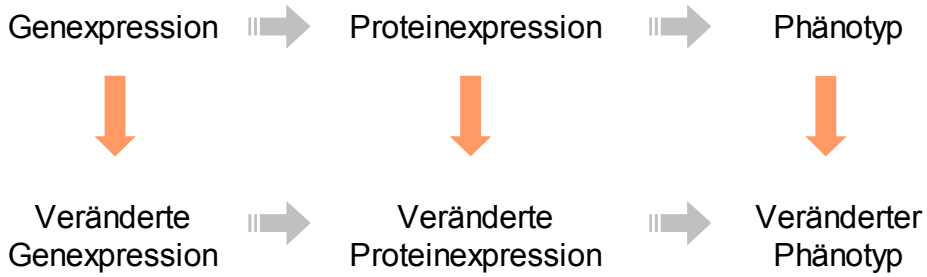


Abbildung 7: Grundannahme des experimentellen Ansatzes
Man geht davon aus, dass ein veränderter Phänotyp durch veränderte Proteinexpression hervorgerufen wird, die durch veränderte Genexpression entsteht.

2.2.4 Qualität der Messung

Wünschenswert wäre es natürlich, die Genexpression in einer natürlichen Einheit, wie mRNA-Kopien pro Zelle, zu messen, und eine Fehlerwahrscheinlichkeit oder ein Qualitätsmaß, wie z.B. die Standardabweichung, für jeden Messwert zu besitzen. Dies ist jedoch aufgrund einiger experimenteller Einschränkungen nicht möglich. Die Rohdaten

von Microarray-Experimenten sind die Bilder aus Scans der hybridisierten Arrays, aus denen der Messwert für jede Sonde bestimmt wird. Gensonden mit derselben Sequenz können jedoch öfters auf dem Microarray vorkommen und mehrere unterschiedliche Sequenzen können zu demselben Gen gehören. Um einen eindeutigen Messwert für ein Gen zu erhalten, müssen die entsprechenden Messungen kombiniert werden. Zusätzlich ist zum Vergleich verschiedener Microarrays eine Normalisierung notwendig, um systembedingte Einflüsse zu verringern. Weitere Faktoren beeinflussen die Qualität der Microarrayexperimente:

- Normalerweise wird davon ausgegangen, dass eine Sonde die Expression eines bestimmten Gens misst. Sind auf einem Chip jedoch verschiedene Gensonden für verschiedene Abschnitte desselben Gens vorhanden, so können diese Gensonden aufgrund von alternativen Spleißen, alternativer poly(A)-Stellen oder durch Fehler auf dem Array unterschiedliche Expressionswerte messen [44].
- Fehlerhafte Annotation der Daten kann durch Fehler in Sequenzdatenbanken entstehen, die eine falsche Zuordnung der Sondensequenz zum entsprechenden Gen enthalten. Insbesondere ältere Microarray-Experimente, deren Annotation noch mit Entwurfsversionen der Genome der entsprechenden Organismen arbeitet, sind von diesem Problem betroffen.
- In der Anfangszeit der cDNA-Microarray-Experimente wurden die Gensonden auf den Arrays nicht auf ihre Sequenz überprüft, was Fehlinterpretationen ermöglichte [45]. Mittlerweile werden jedoch immer häufiger resequenzierte cDNA-Gensonden verwendet, um die Spezifität für bestimmte mRNA sicherzustellen. Bei Oligomer-Microarrays ist die Sequenz der Gensonden durch die Synthetisierung eindeutig festgelegt.
- Qualitätsprobleme können durch Kreuzhybridisierung mehrerer mRNA-Moleküle mit einer Sonde entstehen. Untersuchungen zeigen, dass es bei bis zu 10% der Gensonden auf einem Chip zu Kreuzhybridisierung durch verschiedene Spleißvarianten kommen kann [46].
- Genexpressionsarrays können Unterschiede in der Expression einer Subpopulation von Zellen mit einer hohen Sensitivität feststellen, eine Änderung der Genexpression kann in 5% der Totalpopulation entdeckt werden [47]. Dies

bedeutet aber auch, dass in einer heterogenen Zellkultur oder Gewebe signifikante Änderungen der Genexpression nur von einem kleinen Teil der Zellen ausgehen können.

- Die Kosten für die Microarray-Experimente veranlassen vielen Forscher, nur wenige oder keine Kontrollexperimente durchzuführen, um die Verlässlichkeit und Richtigkeit ihrer Ergebnisse zu validieren.

Aufgrund dieser qualitativen Einschränkungen der Microarray-Methode werden die Ergebnisse der Microarray-Experimente meist mit anderen Messmethoden überprüft.

2.2.5 Andere Messmethoden

Microarrays sind nicht die einzige Möglichkeit, um die Menge an mRNA und Änderungen in der Genexpression zu messen. Zur Messung der Genexpression stehen für die extrahierte RNA der Probe weitere Techniken zur Verfügung:

- Northern Blots: Übertragung der durch Gelelektrophorese aufgetrennten RNA auf eine Membran und anschließender Markierung von RNA durch Hybridisierung mit komplementären Gensonden.
- Polymerase-Kettenreaktion nach reverser Transkription der RNA (RT-PCR): die RNA wird mit dem Enzym Reverse Transkriptase in cDNA umgeschrieben, die cDNA wird als Ausgangsprodukt in einer Polymerase-Kettenreaktion (PCR) verwendet, die Produkte lassen sich anschließend klonieren, oder elektrophoretisch auftrennen. Durch Verwendung spezifischer Primer kann nur nach bestimmten Sequenzen gesucht werden.
- Nuclease Protection: Die RNA wird mit Gegenstrang-RNA- oder DNA-Sonden gemischt, die eine komplementäre Sequenz besitzen, und die komplementären Stränge werden zu doppelsträngigen Molekülen hybridisiert. Anschließend wird die Lösung Ribonukleasen ausgesetzt, die alle einsträngigen Moleküle zersetzt, so dass nur die markierte RNA in der Lösung übrig bleibt. Deren Sequenz ist dabei jedoch auf die Länge der Sonde beschränkt.
- ESTs *Expressed Sequence Tag* (EST) sind kurze Teilsequenzen von mRNAs, die durch Sequenzierung an beiden Enden erzeugt werden. Dabei ist unwichtig, ob die RNA proteinkodierend ist oder nicht. ESTs werden verwendet, um Bereiche auf

dem Genom zu bestimmen, die von der RNA-Polymerase abgelesen werden. Die erzeugte Sequenz ist von geringer Qualität und heutige Sequenziertechniken erlauben nur Sequenzlängen zwischen 500 und 800 bp.

- Differential Display vergleicht zwei Proben. Die RNA wird jeweils in cDNA umgewandelt, mittels PCR amplifiziert und auf zwei Spuren mittels Gel-Elektrophorese aufgetrennt. Bei unterschiedlich starken Banden kann die entsprechende cDNA isoliert, kloniert und weiter analysiert werden.
- Serial Analysis of Gene Expression (SAGE) [48] ist eine Methode, den Zustand der mRNA eines Organismus im Ganzen zu erfassen. SAGE ist ein offenes System, und kann auch verwendet werden, um bisher unbekannte RNA-Sequenzen zu ermitteln. Zur Durchführung wird die Gesamt-mRNA einer Probe extrahiert, in DNA umgeschrieben und in kurze Abschnitte geschnitten. Diese Abschnitte werden miteinander verbunden und bilden lange Moleküle, die anschließend sequenziert werden.

Die mRNA ist nur ein Zwischenprodukt auf dem Weg zum Protein. Der direkte Ansatz, das Endprodukt, die Proteine, zu messen, ist leider erheblich komplizierter als die Messung der mRNA. Eine der Hauptgründe liegt darin, dass proteinbasierende Ansätze wesentlich schwieriger durchzuführen sind, eine geringere Empfindlichkeit und einen geringeren Durchsatz als die RNA-basierten Verfahren besitzen. Zur Messung der Proteinmenge stehen Verfahren wie Western blotting, zweidimensionale Gele, oder Massenspektrometrie zur Verfügung. Antikörper-Microarrays können verwendet werden, um die Expression von Proteinen in bestimmten Zelltypen zu bestimmen. Proteinbasierte Messmethoden erlauben es, postranslationale Proteinmodifikationen und Proteinkomplexe zu erkennen und sie können in manchen Fällen auch Informationen zur Lokalisation der Proteine liefern. Keine dieser Informationen kann durch die Messung der mRNA gewonnen werden.

2.2.6 Vergleichbarkeit von Arrayplattformen

Bei der Untersuchung der Messabweichungen von Microarray-Experimenten wurden in einer Studie 10 verschiedene Arrays und zwei QRT-PCR Methoden verglichen [49]. Dabei zeigt sich, dass jede der Plattformen für sich betrachtet konsistente und reproduzierbare Ergebnisse liefert. Es zeigt sich auch, dass eine Korrelation zwischen den Plattformen

vorhanden ist, die sich aber signifikant verbessert, wenn der Vergleich auf die Gensonden beschränkt wird, die sich überlappende Sequenzen besitzen. Dabei haben die Messwerte für Sequenzen aus dem gleichen Exon eine bessere Korrelation als für Sequenzen aus demselben Gen, deren Messwerte wiederum eine bessere Korrelation als Sequenzen aus dem gleichen UniGene-Cluster besitzen. Die Gründe hierfür reichen vom alternativen Spleißen bis zu unterschiedlichen Sequenzannotationen für dasselbe Gen.

Die Korrelation zwischen den Plattformen ist am höchsten bei Gensonden, die auf mittleren bis hohem Niveau exprimiert werden. Dies ist wichtig, da viele interessante Gene, inklusive vieler Transkriptionsfaktoren, auf relativ niedrigem Niveau exprimiert werden. Für alle Plattformen zeigt sich jedoch, dass bei Signalen, die nahe am Hintergrundrauschen liegen, Messungen nur schwer zu vergleichen sind.

Auch andere Studien kommen zu dem Ergebnis, dass eine Kombination von verschiedenen Array-Plattformen nicht sinnvoll ist, da technische Variationen die Veränderung in den biologischen Daten überdecken [50].

2.2.7 Comparative Genome Hybridization

In Tumorgewebe weisen die Zellen meist nicht nur eine unterschiedliche Genexpression im Vergleich zu normalen Gewebe auf, sondern haben meist auch ein spezielles Profil an genetischen Veränderungen. Comparative Genome Hybridization (CGH) [51] hat sich als Methode etabliert, um Chromosomabberationen im gesamten Genom erkennen zu können. Die Technik wird häufig in der molekularen Diagnostik [52] eingesetzt und viele Untersuchungen kombinieren CGH-Informationen und Microarray-Daten.

CGH erlaubt es, Änderungen in der Anzahl der Chromosom-Kopien festzustellen und kann so eine Übersicht über Zunahme oder Verlust von Chromosomteilen im gesamten Genom eines Tumors liefern. Hierzu wird DNA eines Tumors mit einem grünen Fluoreszenzfarbstoff markiert und mit rot markierter normaler DNA gemischt. Normale Metaphasechromosomen werden denaturiert, dehydratisiert und mit der zuvor hergestellten DNA-Mischung hybridisiert. Während der Hybridisierung konkurrieren die markierten einzelsträngigen DNA-Fragmente des Tumors und der Normalprobe um freie Bindungsstellen auf den denaturierten Chromosomen und binden in einer Verdrängungsreaktion entsprechend der Häufigkeit in den Ursprungsproben an den Chromosomen. Die jetzt gebundenen DNA-Fragmente können mit einem

Fluoreszenzmikroskop angezeigt werden und aufgrund des Verhältnisses des roten zum grünen Farbstoff können chromosomale Änderungen im Tumor computergestützt erkannt werden [53].

Einschränkungen der CGH-Technik liegen darin, dass keine chromosomale Änderungen außer Änderungen in der Chromosomenzahl erfasst werden, wie z.B. Translokationen oder Inversionen. Bei der Untersuchung von Tumormaterial kann es zur Kontaminierung durch normales Zellmaterial kommen, die Änderungen im Tumor überdecken kann. Außerdem kann die CGH-Technik nur Änderungen in einer Größenordnung von mindestens 10-20 Mbp erkennen.

Eine Erweiterung der CGH-Technik durch die Verwendung von Microarrays ist die ArrayCGH -Technik, die z.B. in [54] beschrieben wird, und die eine höhere Auflösung als die gewöhnliche CGH-Technik besitzt.

2.3 Datenbanken

2.3.1 Ensembl

Ensembl ist ein Bioinformatik-Projekt, das biologische Informationen zu Genomsequenzen erstellen, verwalten und anzeigen kann [55]. Es beherrscht die automatische Annotation [56] von Genomen, zusätzlich wurde ein Framework geschaffen, um jede Art biologischer Daten, die sich auf das Genom beziehen, integrieren zu können. Ensembl ist als interaktive Website, als Menge von Datenbank-Flatfiles, oder als komplettes OpenSource-System zur Genomannotation verfügbar. Die Ensembl-Internetseite ist neben dem USCS Genombrowser [57] und den NCBI Genome Resources [58] eines der drei Hauptsysteme zur Annotation und Anzeige von Genominformation. Durch Zusammenarbeit wurde erreicht, dass alle drei Systeme die selben Sequenzdaten zur Annotation verwenden und dass die Datenbanken untereinander verknüpft sind. Um Forschern die Möglichkeit zu geben, Teile des Genoms nach bestimmten Kriterien zu untersuchen, wurde mit EnsMart [18] ein internetbasiertes Data Mining System entwickelt.

Die Ensembl Internetseite ist auf dem Apache Webserver [59] mit mod_perl und der MySQL-Datenbank aufgebaut [60]. Systeme zur Verwaltung von Genomdaten müssen eine große Menge an Informationen speichern. Ensembl verwendet zur Speicherung

mehrere MySQL-Datenbanken. Die Kernbibliotheken der Software [61] erlauben einen einfachen und effektiven Zugriff auf diese Daten. Sie kapseln die unterliegende Datenbankstruktur und bieten ein abstraktes Interface zum komplexen Datenmodell. Dadurch sind Programme, die diese Core-Bibliotheken verwenden, unabhängig von Änderungen in der Datenbankstruktur. GEPAT verwendet das ensj-Modul, das die Abfragen an die Ensembl-Datenbank in eine Java-API kapselt.

Die Ensembl Analyse-Pipeline [62] wurde geschaffen, um im großen Maßstab automatisch Genomsequenzen annotieren zu können. Hauptaugenmerk lag hierbei am Anfang auf den Genomdaten von Maus und Mensch, aktuell kann mit Ensembl auf die Genominformation von 28 Chordata-Spezies zugegriffen werden, von Mensch bis zu einfachen Lebewesen wie *Ciona intestinalis.* Daneben enthält Ensembl die Genome der drei Modellspezies Hefe (*Saccharomyces cerevisiae*), Fruchtfliege (*Drosophila melanogaster*) und Fadenwurm (*Caenorhabditis elegans*).

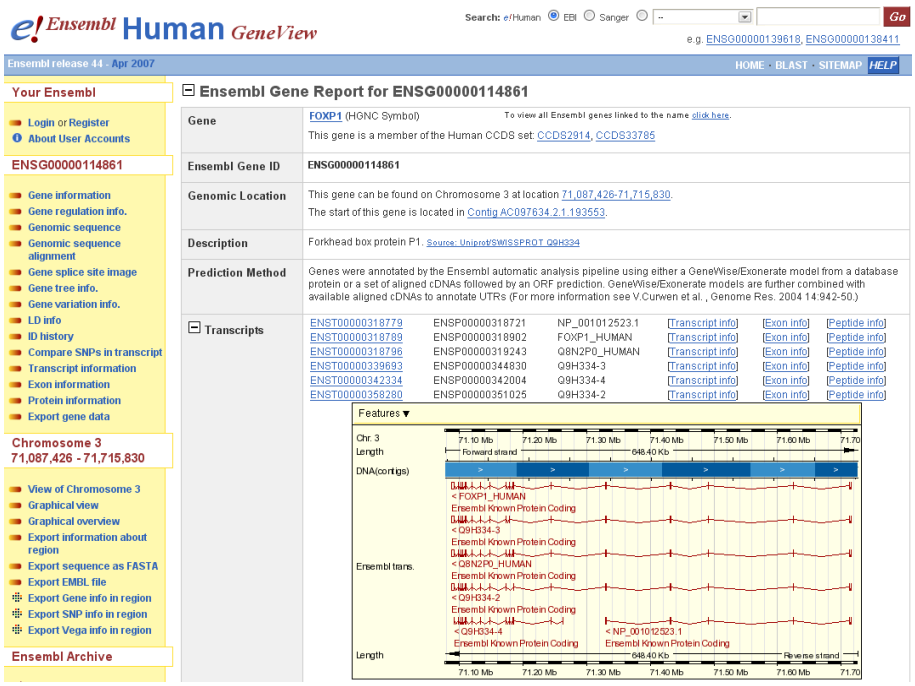


Abbildung 8: Ensembl
Die Ensembl Geninformationsseite für das menschliche FOXP1-Gen. Gezeigt wird nur ein Ausschnitt der verfügbaren Informationen.

Für jeden unterstützten Organismus enthält die Kerndatenbank eine Menge Annotationsinformationen, u. a.

- die Genomsequenz auf Sequenzlevel, entstanden aus BAC-Clonen oder Whole-Genome-Shotgun-Bruchstücken
- Informationen, wie diese Sequenzen zu übergeordneten Einheiten, wie den Chromosomen, zusammengesetzt werden
- Die nichtredundante Menge an Ensembl Gen-, Transkript- und Proteinmodellen aus der automatischen Annotationspipeline
- Alignments von cDNA und Proteinen zur Genomsequenz
- Sequence tagged sited (STS) Markerinformationen auf der Sequenz und auf Genkartenpositionen
- Karyotypinformationen wie z.B. Chromosombanden
- Annotation von Microarray-Sonden
- Externe Referenzen zu anderen Datenbanken mit biologischen Informationen

Abbildung 8 zeigt ein Beispiel für einen Ensembl-Geneintrag.

2.3.2 STRING

Beziehungen zwischen Proteinen können in Form direkter Protein-Protein Interaktionen, aber auch durch Vorkommen im gleichen Stoffwechselweg, ähnlicher Genexpression oder chromosomaler Nachbarschaft bestehen. Proteinassoziationen werden in der STRING-Datenbank gesammelt. Assoziationen werden aus verschiedenen Quellen gezogen:

- Genominformationen. Diese beinhalten Gene, die sich in direkter Nachbarschaft im Genom befinden. Diese Nachbarschaft ist besonders bei der Prokaryoten interessant, bei denen die Gene in Opterons abgelesen werden. Daneben tragen Genfusionen und Gene mit demselben phylogenetischen Profil, d.h. dem gemeinsamen Vorhandensein in Spezies, zu dieser Art von Assoziationen bei. Die Genominformationen basieren auf der Annahme, dass diese Gene unter gemeinsamen Selektionsdruck während der Evolution standen und somit funktionell assoziiert sind.
- Coexpression – Proteine, die ein ähnliches Genexpressionsmuster besitzen
- Experimente – Experimentell nachgewiesene Proteininteraktionen aus anderen

Datenbanken. Fehlinformationen, die durch den hohen Durchsatz von Proteininteraktionsexperimente entstehen, wurde durch Gewichtung unterschiedlicher Techniken Rechnung [63] getragen. Auch bestehende Datenbanken wie KEGG oder HPRD werden verwendet, um Assoziationen abzuleiten.

- Vorhandenes Wissen - Textmining wird verwendet, um Assoziationen aus PubMed-Zusammenfassungen zu generieren. Die hierbei entstehenden Assoziationen bedeuten jedoch nicht unbedingt, dass Proteine, die im selben Artikel vorkommen, in funktionaler Beziehung stehen müssen. Entscheidend ist nur die gemeinsame Erwähnung in der Zusammenfassung, nicht der Zusammenhang.

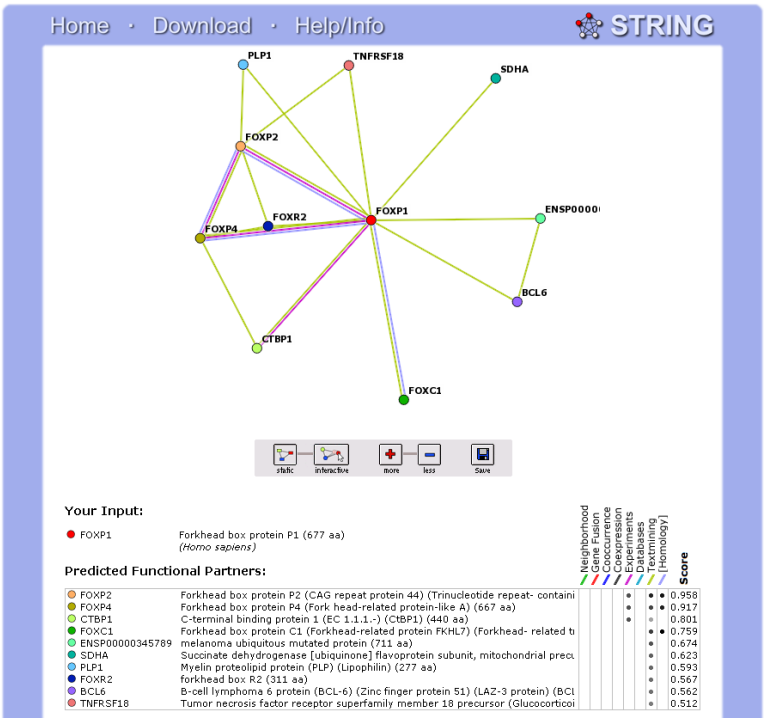


Abbildung 9: STRING
Der Assoziationsgraph für das menschliche FOXP1-Gen. Dargestellt werden die am besten bewerteten 10 Assoziationen mit einem Konfidenzwert > 0.5. Die unterschiedlichen Farben der Kanten stehen für unterschiedliche Assoziationstypen.

Die Assoziationen werden über orthologe Proteinpaare auf andere Organismen übertragen. Die Qualität der Assoziationen wird über einen Konfidenzwert ausgedrückt. Die Konfidenzwerte der verschiedenen Assoziationen werden in einem kombinierten Wert miteinander vereint. Die Anzeige kann nach Art der Assoziationen, Anzahl der

Assoziationen und Konfidenzwert eingeschränkt werden. Assoziationen können für ein Protein oder für eine Menge von Proteinen angezeigt werden. Neben einzelnen Informationen für jeden Assoziationstyp, wie in Abbildung 9 gezeigt, kann ein Übersichtsgraph angezeigt werden, in dem Assoziationen durch Kanten zwischen den Proteinnamen dargestellt werden. Ein weiterer Modus von STRING bietet die Möglichkeit, Assoziationen zwischen Cluster orthologer Gruppen (COG) anzeigen zu können.

2.3.3 KEGG

Die Kyoto Encyclopedia of Genes and Genomes (KEGG) Datenbank [64] dient als Ressource zum Verständnis der höheren Funktionen von biologischen Systemen und kann als eine künstliche Repräsentation dieser Systeme betrachtet werden. KEGG besteht aus vier Datenbanken:

- KEGG PATHWAY – Abbildung biologischer Stoffwechselwege (Network Knowledge)
- KEGG GENES – Genkatalog und orthologe Beziehungen zwischen Genomen (Genomic Knowledge)
- KEGG LIGAND – chemische Verbindungen und Reaktionen (Chemical Knowledge)
- KEGG BRITE – Funktionshierarchien biologischer Systeme (Ontologien)

Die KEGG PATHWAY Datenbank beschreibt molekulare Interaktionsnetzwerke in der Zelle und spezifische Varianten dieser Netzwerke für bestimmte Organismen. Die Datenbank besteht aus einer Sammlung manuell gezeichneter Karten aus den Bereichen Metabolismus, genetische Informationsverarbeitung, Signalweiterleitung und verschiedenen zellulären Prozessen und Krankheiten. Abbildung 10 zeigt ein Beispiel einer solchen Karte.

Sowohl zu metabolischen als auch zu den regulatorischen Wegen stehen XML-Informationen in der KEGG Markup Language (KGML) zur Verfügung, welche die Informationen der manuell generierten Karten enthalten und so eine automatische (Weiter-)Verarbeitung ermöglichen. KGML enthält die Positionen der Knoten in den KEGG PATHWAY Karten, die Informationen zur grafischen Darstellung der Kanten fehlen jedoch. In GEPAT werden deshalb die manuell erstellten Karten als Grundlage verwendet, auf die anhand der in KGML enthaltenen Daten und Positionen weitere Informationen eingezeichnet werden. Die KGML-Dateien spezifizieren Beziehungen zwischen je zwei

Proteinen oder orthologen Gruppen von Genen oder einem Protein und einem Molekül, die durch einen Pfeil oder eine Linie auf der Karte gekennzeichnet werden. Eine Beziehung besteht aus

- Enzym-Enzym Reaktionen, d.h. aufeinander folgende Reaktionsschritte
- Protein-Protein Reaktionen, wie Bindung oder Modifikation
- Genexpressions-Interaktion, eine Beziehung zwischen Transkriptionsfaktor und Zielgen
- Protein-Molekül-Interaktion
- Verweis auf eine andere Stoffwechselkarte

Eine genaue Beschreibung der Semantik von KGML findet sich in [65].

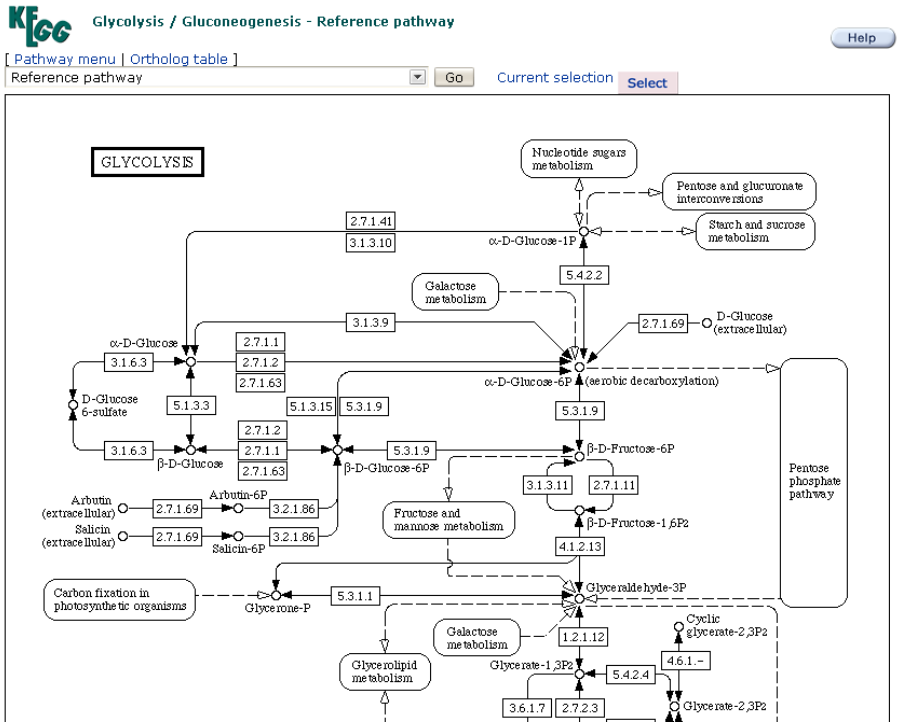


Abbildung 10: KEGG PATHWAY
Gezeigt wird die KEGG Stoffwechselkarte des Glykolyse und Glukoneogenese-Stoffwechselpfads. Dargestellt wird nur die obere Hälfte der Karte.

2.3.4 Gene Ontology

Der Begriff Ontologie entstammt der Philosophie und klärt Grundstrukturen des Vorhandenen. Sie beschreibt Kategorien und Beziehungen zwischen Dingen, um

Gegenstände, Eigenschaften und Prozesse innerhalb ihres Rahmens zu definieren. Die Informatik versteht unter einer Ontologie ein formal definiertes System von Begriffen und Relationen, das zusätzlich Inferenz- und Integritätsregeln enthalten kann.

Mit der Sequenzierung von immer mehr Genomen und der Analyse der hieraus erhaltenen Informationen entstand auch in der Biologie Bedarf, Funktionen über verschiedene Organismen hinweg mit einheitlichen Begriffen und mit definierten Abhängigkeiten zu beschreiben. Daraus entstand das Gene Ontology (GO) Projekt [66], das ein kontrolliertes Vokabular zur Beschreibung von Genen und Genprodukten liefert. Die Ontologie selbst ist in drei verschiedene Bereiche der Molekularbiologie eingeteilt:

- *Molekulare Funktion* beschreibt Aktivitäten, die auf molekularer Ebene stattfinden. Die GO-Kategorien stehen hier für Aktivitäten, nicht für Moleküle oder Komplexe, die diese Aktivitäten ausführen und geben keinen Kontext an, in dem diese Aktivitäten stattfinden.
- *Biologischer Prozess* ist eine Serie von Ereignissen, die durch eine oder mehrere molekulare Funktionen durchgeführt werden. Ein biologischer Prozess ist dabei kein Äquivalent zu einem Stoffwechselweg, da Dynamik und Abhängigkeiten nicht berücksichtigt werden.
- *Zelluläre Komponenten* sind alle Komponenten der Zelle, inklusive der Membran und externen, gekapselten Strukturen.

Auch die Annotation von Genen zu Begriffen aus der Ontologie wird vom GO-Projekt verwaltet und über die Internetseite öffentlich verfügbar gemacht [67].

Jeder GO-Eintrag besteht aus einer eindeutigen Kennung, einem Namen, Synonymen, falls diese vorhanden sind, und einer Definition. Die Terme gehören zu einer der drei Ontologien. Die Ontologien sind als gerichtete, azyklische Graphen (directed acyclic graph, DAG) aufgebaut, d.h. ein Kindknoten im Graph kann hier mehrere Eltern besitzen. Ein Kindknoten kann in einer von zwei Beziehungen zu seinen Eltern stehen. Ist ein GO-Eintrag eine Subklasse seiner Eltern, wird die Beziehung *is_a* verwendet, z.B. nuclear chromosome is_a chromosome. Die zweite Beziehung, *part_of*, ist gibt an, dass ein Kind Teil seiner Eltern ist. Dabei ist ein Kind immer Teil seiner Eltern, wenn es vorhanden ist, muss aber nicht immer vorhanden sein. Ein Beispiel ist nucleus part_of cell. Ein Zellkern ist immer Teil der Zelle, aber nicht alle Zellen besitzen einen Zellkern.

2.4 Statistische Auswertung

2.4.1 Datenrepräsentation

Werden die Messwerte für jedes Microarray aus einem Experiment mit n Microarrays durch den hochdimensionalen Spaltenvektor $\vec{x}_{,k}$ repräsentiert, so können die Messwerte aller Arrays durch folgende Matrix dargestellt werden:

$$\begin{array}{cccccccc} & & \vec{x}_{,1} & \vec{x}_{,2} & \cdots & \vec{x}_{,k} & \cdots & \vec{x}_{,n} \\ & & = & = & & = & & = \\ \vec{x}_{1,} & = & x_{1,1} & x_{1,2} & \cdots & x_{1,k} & \cdots & x_{1,n} \\ \vec{x}_{2,} & = & x_{2,1} & x_{2,2} & \cdots & x_{2,k} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vec{x}_{i,} & = & x_{i,1} & x_{i,2} & \cdots & x_{i,k} & \cdots & x_{i,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vec{x}_{m,} & = & x_{m,1} & x_{m,2} & \cdots & x_{m,k} & \cdots & x_{m,n} \end{array}$$

Die Expression der Sonde i über die Arrays wird durch den hochdimensionalen Zeilenvektor $\vec{x}_{i,}$ dargestellt [68]. Jedes der n Arrays kann so als ein Punkt im m-dimensionalen Sondenraum betrachtet werden, umgekehrt ist auch jedes Gen ein Punkt im n-dimensionalen Probenraum.

2.4.2 Fehlende Werte

Häufig kommt es zu fehlenden Messwerten in der Microarray-Datenmatrix, die unterschiedliche Ursachen besitzen können. Einige Analysemethoden sind nicht in der Lage, mit diesen fehlenden Informationen umzugehen. Deshalb ist es für diese Methoden notwendig, entweder die Zeilen und Spalten mit fehlenden Daten komplett zu entfernen, oder eine Abschätzung für die fehlenden Werte zu berechnen. Meist wird hier ein k nearest neighbor Algorithmus verwendet. Hierbei wird jeder Genexpressions-Vektor, dem Daten fehlen, mit anderen, kompletten Genexpressionsvektoren verglichen. Die fehlenden Daten werden dann geschätzt, indem ein gewichteter Durchschnitt aus den Datenpunkten der k ähnlichsten Sonden gebildet wird. Es zeigte sich, dass der Algorithmus robust ist gegenüber Veränderungen des Parameter k und dass die Technik akzeptabel ist, um bis zu 15% der Daten eines Genexpressionsvektors abzuschätzen [69].

2.4.3 Normalisierung

Microarray-Daten müssen normalisiert werden, um die tatsächlichen Variationen in der Genexpression von den Variationen, die durch den Messprozess entstehen, unterscheiden zu können, und um die Daten von verschiedenen Microarrays auf einen gleichen Maßstab zu bringen. Die systembedingten Variationen können z.B. durch unterschiedliche Mengen an Ausgangsprodukten und Unterschiede bei der Aufreinigung der mRNA entstehen.

Bei der Verwendung von Zweikanal-Microarrays gibt es weitere Faktoren, die Variation entstehen lassen. Systembedingte Variationen innerhalb eines Zweikanal-Microarrays sind z.B. Unterschiede in der Effizienz der beiden Farben, der Menge von markierten RNA-Molekülen für jeden Kanal und regionale Unterschiede auf der Oberfläche des Microarrays, die durch das Auftragverfahren der Gensonden mit der Druckspritze entstehen können.

Normalisierung versucht, die Unterschiede in der Intensität, die durch biologische Faktoren entstehen, zu erhalten, und Unterschiede, die durch technische Faktoren entstehen, zu entfernen. Werden die technischen Unterschiede zu groß, ist eine Normalisierung nicht mehr möglich.

Zu Beginn der Expressionsuntersuchungen wurde zur Normalisierung die Verwendung sogenannter Housekeeping-Gene diskutiert. Dazu werden einige Gene ausgewählt, von denen angenommen wird, dass sich ihre Expression unter den getesteten Bedingungen nicht ändert. Anhand der Expression der Housekeeping-Gene wird für jedes Array ein Normalisierungsfaktor bestimmt, der anschließend auf die Daten angewendet wird. Die Methode ist jedoch unzuverlässig, da die Voraussetzung, dass die Housekeeping-Gene ihre Expression nicht ändern, meist nicht erfüllt wird. Fast alle Gene zeigen Expressionsunterschiede unter gewissen Bedingungen.

Mittlerweile werden zur Normalisierung von Microarray-Techniken fortgeschrittene statistische Verfahren verwendet. Im Folgenden werden verschiedene Verfahren für Oligonukleotid-Microarrays und cDNA-Microarrays betrachtet:

Oligonukleotid-Arrays

Die Normalisierung der Daten wird für Oligonukleotid-Arrays in 4 unabhängige Schritte

aufgeteilt: Hintergrundkorrektur, Normalisierung, PM Korrektur und Zusammenfassung. Im Folgenden folgt eine Übersicht über die für jeden Schritt möglichen Algorithmen. Ein Vergleich verschiedener Normalisierungsmethoden findet sich in [70].

Die Hintergrundkorrektur wird verwendet, um eine Reihe von systembedingten Faktoren aus den Messwerten eines Arrays zu entfernen. Insbesondere soll für jedes Array die Skala der Sondenintensitäten so kalibriert werden, dass sie alle dasselbe Grundniveau besitzen. Verschiedene Verfahren stehen zur Hintergrundkorrektur zur Verfügung:

- RMA: Implementierung der robust multi-array average (RMA) Methode aus [71]. Es zeigte sich, dass einige MM-Gensonden eine höhere Intensität als entsprechenden PM-Gensonden besaßen, was aber aufgrund ihres Designs nicht auftreten sollte. Deshalb werden bei dieser Methode nur die PM-Werte korrigiert, die MM-Werte werden ignoriert. Die Intensitäten werden durch Verwendung eines globalen Modells über die Verteilung der Sondenintensitäten korrigiert.
- MAS: Entspricht dem Algorithmus, der in der Affymetrix Software Microarray Suite (MAS) 5.0 verwendet wird [72]. Das Array wird in 16 rechteckige Regionen eingeteilt, zu jeder Region werden die geringsten 2% an Sondenintensitäten bestimmt, um einen Hintergrundwert für die Region zu berechnen. Jede Sonde wird dann anhand eines gewichteten Durchschnitts der Hintergründe jeder Region geändert. Die Gewichte basieren dabei auf dem räumlichen Abstand zwischen der Sonde und dem Zentren der 16 verschiedenen Regionen. Diese Methode korrigiert sowohl PM als auch MM Gensonden.

Ziel der Normalisierung ist es, die nichtbiologische Variation aus den Arrays zu entfernen, und die Messwerte verschiedener Microarrays auf dieselbe Skala zu bringen. Da dieser Schritt alle Arrays miteinbezieht, benötigt er viel Rechenzeit und Hauptspeicher. Verschiedene Verfahren, die in GEPAT zur Normalisierung zur Verfügung stehen sind:

- Constant: Dieses Verfahren wird von Affymetrix in Version 4 und 5 der MAS verwendet. Ein Array wird als Grundlage ausgewählt und alle anderen Arrays werden skaliert, so dass sie dieselbe mittlere Intensität wie dieses Array besitzen. Vor der Berechnung des Mittelwerts werden die höchsten und niedrigsten 2% der Daten entfernt.
- Quantiles: Diese Methode wurde in [70] vorgestellt. Sie ist eine Transformation

$x'_i = F^{-1}[G(x_i)]$, wobei G durch die empirische Verteilung von jedem Array geschätzt wird und F die empirische Verteilung der gemittelten Proben-Quantile ist.

- Loess: Die für Oligonukleotid-Arrays verwendete cyclic-loess Methode ist eine Verallgemeinerung der loess – Methode aus [73] und [74], die zur Normalisierung von Zweikanal-Arrays verwendet wird. Für Oligonukleotid-Arrays werden die Arrays paarweise miteinander verglichen. Das Verfahren verwendet alle möglichen Paarkombinationen der Arrays und wiederholt den Prozess bis zur Konvergenz. Ein Nachteil des Verfahrens ist die Anzahl von $O(n^2)$ durchzuführenden loess-Transformationen.

- VSN: Diese Funktion kombiniert Hintergrundkorrektur und Normalisierung in einem Schritt [75][76]. Die Funktion kalibriert Variationen zwischen den Proben durch Verschiebung und Skalierung und transformiert die Intensitäten auf einen Maßstab, bei dem die Varianz annähernd unabhängig von der mittleren Intensität ist. Die varianzstabilisierende Transformation ist äquivalent zum natürlichen Logarithmus im Bereich der hohen Intensitäten und zu einer linearen Transformation im Bereich der niedrigen Intensitäten. Im dazwischenliegenden Bereich interpoliert die *arsinh*-Funktion zwischen den beiden Möglichkeiten.

PM Korrektur dient dazu, Kreuzhybridisierung und nichtspezifische Bindungen zu korrigieren. Ursprünglich waren die MM-Gensonden vorgesehen, um nichtspezifische Bindung der PM-Gensonden korrigieren zu können, indem die Intensität der MM Gensonden von der Intensität der PM Gensonden abgezogen werden sollte. Jedoch zeigte sich, dass bei den Daten eines typischen Arrays bis zu 30% der MM-Gensonden eine höhere Intensität als die entsprechende PM Sonde besitzen [77]. Affymetrix schlägt vor, mit dem so genannten Ideal Mismatch zu arbeiten, der immer kleiner als die PM-Sonde ist. Viele Forscher jedoch ignorieren die MM-Gensonden komplett,und benutzen nur die PM-Sondenintensitäten zur weiteren Analyse.

- MAS: Da das Abziehen der MM Intensität von der PM Intensität zu negativen Werten führen kann, Expressionswerte aber nicht geringer als Null sein können, wurde von Affymetrix das Konzept des Ideal Mismatch (IM) eingeführt [72], der stets geringer ist als der entsprechende PM-Wert. Die korrigierten PM Intensitäten

werden durch abziehen der IM-Intensitäten von den originalen PM Intensitäten gebildet.

- Pmonly: verwendet nur die Intensitäten der PM zur weiteren Berechnung

Die Zusammenfassung der Intensitäten der Gensondenmenge für ein Transkript ist der letzte Schritt in der Bearbeitung der Affymetrix-Files. Hier werden die Intensitäten der korrigierten PM-Gensonden in einen Wert vereint, der für die Expression des entsprechenden Transkripts steht. Mögliche Verfahren sind:

- MAS: Das von Affymetrix verwendete Verfahren, das den (robusten) Durchschnitt durch 1-Schritt Tukey Biweight Verfahren[78] auf $\log_2$ -Maßstab erstellt
- medianpolish: Diese Art der Zusammenfassung wurde in [71] vorgeschlagen. Ein lineares Modell wird mit dem median-polish Verfahren von Tukey auf den Datensatz angepasst. Die von diesem Verfahren berechneten Intensitäten sind im $\log_2$-Maßstab.

Eine Vielzahl weiterer Algorithmen existiert zur Normalisierung von Affymetrix-Arrays. Bekannt sind vor allem RMA [79], GCRMA [80][81] und Affymetrix PLIER [82]. Auch Algorithmen, die sich zur Normalisierung nicht nur auf den Datensatz, sondern auf zusätzliche Informationen aus Array-Datenbanken beziehen [83], sind verfügbar. Einen Vergleich der unterschiedlichen Verfahren liefert [84].

cDNA-Arrays

Die meisten Normalisierungsmethoden für cDNA-Arrays unterscheiden zwischen within-Array (local) und between-Array Normalisierung (scale). Within-Array Normalisierung passt die Quotienten der beiden Kanäle so an, dass diese für jedes Array und Teilarray im Mittel 0 betragen. Between-Array Normalisierung passt die Expressionswerte so an, dass sie eine ähnliche Verteilung über einer Reihe von Arrays besitzen.

Die local-Normalisierung wird durch die limma-Funktion **`normalizeWithinArrays`** durchgeführt. Die log-Ratios werden um Null zentriert, wobei die Intensität und die ortsabhängige Tendenzen in die Berechnung miteinbezogen werden. Eine mögliche Normalisierungsfunktionen für die local Normalisierung ist:

- Loess: Eine Implementierung der Loess-Normalisierungsmethode [73][74]. Da sich die Intensität der Sonde aufgrund der räumlichen Position durch die

Aufbringung über die Druckspitze unterscheiden kann, kann dies bei der Normalisierung zusätzlich berücksichtigt werden [85].

Skalierungsunterschiede können zu Arrays führen, die ein übermäßiges Gewicht in den logarithmierten Quotienten, verglichen mit den anderen Arrays, besitzen. Dies wird durch die scale-Normalisierung ausgeglichen, die durch die limma-Funktion **`normalizeBetweenArrays`** durchgeführt wird. Mögliche Normalisierungsarten sind:

- Scale: Skaliert die logarithmierten Quotienten, so dass diese dieselbe mittlere absolute Abweichung über alle Arrays besitzen.
- Quantile: Eine Adaption [86] der Quantile Normalisierungsmethode [70] für Zweikanal-Array. Quantile-Normalisierung stellt sicher, dass die Quotienten die gleiche empirische Verteilung über alle Arrays und Kanäle besitzen.
- VSN - Diese Funktion kombiniert local-Normalisierung und scale-Normalisierung, analog zu dem Verfahren für Affymetrix-Normalisierung. Die Rohdaten müssen hierfür ohne Hintergrundkorrektur als Eingabe verwendet werden.

2.4.4 Differentielle Expression

Meist sind Microarray-Experimente darauf ausgelegt, die Unterschiede in der Genexpression zwischen biologischen Zuständen zu ermitteln, wobei jeder Zustand dabei von einem oder mehreren Proben repräsentiert wird. Dabei kann auf die Nullhypothese, dass keine Unterschiede in der Expression zwischen den Proben existieren, getestet werden. Führt man einen Signifikanztest durch, so bestimmt man im Stichprobenraum eine sogenannte kritische Region K_α derart, dass im Falle der Richtigkeit von H_0 das Ergebnis einer Zufallsstichprobe vom Umfang n höchstens mit der vorgebbar kleinen Wahrscheinlichkeit α in diese kritische Region fällt. Fällt das Ergebnis der Zufallsstichprobe in diese kritische Region, wird die Nullhypothese abgelehnt bei Zugrundelegung der Irrtumswahrscheinlichkeit α. Der sogenannte p-Wert beschreibt dabei den kleinsten Wert α, für den H_0 abgelehnt würde. Einen Vergleich zwischen verschiedenen Verfahren zur Ermittlung differentiell exprimierter Gene findet sich in [87].

Differentiell exprimierte Gene

Die einfachste Methode, differentiell exprimierte Gene festzustellen, beruht darin, für jedes Gen das Expressionsverhältnis zwischen zwei verschiedenen Bedingungen zu

untersuchen, und alle Gene, die sich um mehr als einen festgelegten Wert unterscheiden, als differentiell exprimiert zu betrachten. Dieser Test, meist als Fold-Change bezeichnet, ist kein statistischer Test und es gibt keine Fehlerwahrscheinlichkeit für die Zuweisung eines Gens. Die Ergebnisse des Tests werden aus Symmetriegründen meist logarithmisch transformiert angegeben.

Der t-Test ist eine einfache statistische Methode, um differentiell exprimierte Gene zu finden. Nimmt man an, dass ein Experiment mit *k* Arrays zwei Bedingungen k_1 und k_2 mit $k=k_1+k_2$ besitzt, und sind Mittelwert und Varianz für ein Gen g mit $\bar{x}_{g1}, s^2_{g1}$ und $\bar{x}_{g2}, s^2_{g2}$ bezeichnet, so kann die Welch-T-Statistik für zwei unabhängige, annähernd normalverteilte Messreihen ohne Annahme von gleicher Varianz wie folgt geschrieben werden [88]:

$$t_g = \frac{\bar{x}_{g1} - \bar{x}_{g2}}{Se_g}, Se_g = \sqrt{\frac{s^2_{g1}}{k_1} + \frac{s^2_{g2}}{k_2}}$$

Ein Gen, das Aufgrund geringer Expressionswerte nur eine geringe Varianz besitzt, führt zu hohen absoluten t-Werten, unabhängig von der realen Differenz zwischen den beiden Bedingungen, und kann deshalb als differentiell exprimiertes Gen erkannt werden ohne eine Änderung zu besitzen.

Um dieses Problem zu umgehen, wurden verschiedene Verfahren entwickelt. Die B-Statistik [89] erlaubt genspezifische Varianz, vereinigt aber Informationen über viele Gene und ist deshalb stabiler als die t-Statistik. Das verwendete hierarchische Modell wurde mit einem linearen Modell erweitert und in der Software limma implementiert [90]. Zur statistischen Analyse und zur Zuweisung der differentiellen Expression verwendet limma eine empirische Bayes-Methode (moderated t-Test), um den Standardfehler zu verringern. Dies erlaubt einen stabileren Rückschluss, insbesondere für Experimente mit einer kleinen Anzahl an Arrays.

Nachdem eine Teststatistik berechnet wurde, wird diese meist in einen p-Wert umgewandelt. Gene mit einem p-Wert unter einem vorher festgelegten Wert werden als signifikant betrachtet, der p-Wert wird meist ausgegeben, um einen Messwert für die Richtigkeit der Ergebnisse zu besitzen und um weitere Informationen zur Signifikanz zu erhalten.

Sollen mehr als zwei Bedingungen miteinander verglichen werden, können erweiterte Verfahren, wie z.B. die Varianzanalyse verwendet werden [91].

Multiple Testing Problem

Bei der Durchführung eines Hypothesentests kann es zu zwei Arten von Fehlern kommen. Ein Fehler vom Typ I, oder false-positive Fehler, entsteht, wenn ein Gen als differentiell exprimiert deklariert wird, obwohl es dies nicht ist. Ein Fehler vom Typ II entsteht, wenn ein differentiell exprimiertes Gen nicht entdeckt wird. Ein statistischer Test ist gewöhnlich konstruiert, um die Fehlerwahrscheinlichkeit vom Typ I zu kontrollieren. In einem Microarray-Experiment werden bei der Untersuchung auf differentiell exprimierte Gene tausende Test durchgeführt und Fehler vom Typ I können sich so akkumulieren. Das Problem wird als multiple Testing Problem bezeichnet.

Ein Ansatz zur Lösung des multiple Testing Problems ist die Kontrolle der family-wise Error Rate (FWER), die Wahrscheinlichkeit, einen oder mehrere Fehler vom Typ I über mehrere statistische Test zu akkumulieren. Dies wird durch eine Erhöhung der Genauigkeit, die von jedem individuellen Test verlangt wird, erreicht. Die einfachste FWER-Prozedur ist die Bonferroni-Korrektur: das Signifikanzniveau α wird durch die Anzahl der durchgeführten Tests geteilt. Sind $H_1,\ldots, H_m$ die Nullhypothesen und $p_1,\ldots, p_m$ die entsprechenden p-Werte, so wird H_k zurückgewiesen, falls $p_k \leq \frac{\alpha}{m}$ Das Verfahren ist nur eine grobe Näherung und konservativ, ein besserer Ansatz bietet die Bonferroni-Holm Step-Down-Methode. Hierzu werden die Gene nach dem p-Werte aufsteigend sortiert, $p_{(1)},\ldots,p_{(m)}$ bezeichne die sortierten Werte. H_k wird zurückgewiesen, falls $p_{(k)} \leq \frac{\alpha}{m-(k+1)}$.

Ein alternativer Ansatz zum Lösen des Multiple Testing Problems behandelt die false-discovery Rate (FDR), der Anteil der false positives unter allen Genen, die ursprünglich als differentiell exprimiert erkannt wurden. Die bekanntesten Verfahren sind hier die Verfahren von Benjamini und Hochberg [92] und Benjamini und Yekutieli [93]. Sind $H_1,\ldots, H_m$ die Nullhypothesen, $p_1,\ldots, p_m$ die entsprechenden p-Werte und $p_{(1)},\ldots, p_{(m)}$ die aufsteigend sortierten p-Werte, so sucht die Benjamini-Hochberg-Methode zu einem gegebenen α das größte k, so dass $p_{(k)} \leq \frac{k}{m}\alpha$. Alle $H_{(i)}$ mit $i=1,\ldots,k$ werden

zurückgewiesen. Dieser Ansatz ist nur gültig, wenn die Test voneinander unabhängig sind, für abhängige Test wurde die Benjamini-Yekutieli-Methode entwickelt. Eine Vielzahl weiterer Ansätze zur multiplen Testkorrektur wurden vorgeschlagen, die auf einer Schätzung der wahren Hypothesen beruhen. Ein weiterer bekannter Ansatz ist der von Storey [94], der den Begriff q-Wert vorschlägt, der als eine Anpassung des p-Wertes an die Problematik des multiplen Testens gesehen werden kann.

2.4.5 Clustering

Bei der Analyse von Microarray-Daten kommt unweigerlich die Frage auf, ob sich innerhalb des Experiments bestimmte Signaturen oder Muster in der Genexpression finden lassen. Zum Aufspüren dieser Muster können Clustering-Algorithmen verwendet werden. Der Begriff Clustering steht für das Partitionieren der Daten in Teilmengen, den sogenannten Clustern, in denen die Elemente gemeinsame Eigenschaften besitzen. Clustering-Algorithmen werden nicht zum Hypothesentest verwendet und liefern von daher keine Informationen über die Signifikanz der Ergebnisse, sondern werden vielmehr dazu verwendet, um die Daten zu untersuchen, biologisch sinnvolle Muster zu entdecken und daraus neue Hypothesen zu entwickeln.

Eine der Haupteigenschaften der Microarray-Experimente liegt darin, dass sie tausende von Genexpressionswerten gleichzeitig messen und deshalb eine hohe Dimensionalität besitzen. Bei 10000 Messwerten für eine Probe kann jeder Genexpressionsmesswert als Punkt im 10000-dimensionalen Raum aufgefasst werden. Clustering-Methoden erlauben einen visuellen Überblick über die Daten und können zur Klassenerkennung genutzt werden, z.B. um verschiedene Krankheitstypen in den Proben zu entdecken und wurden hierfür schon früh in der Microarray-Datenanalyse verwendet [95].

Meist werden zur Analyse von Genexpressionswerten *hierarchische* Clustermethoden verwendet, die eine wachsende Anzahl ineinander geschachtelter Klassen zurück liefern, deren Zusammengehörigkeit in einem Baum, dem sogenannten Dendrogramm dargestellt wird. Es existieren auch nichthierarchische Clustermethoden, wie das *k-means* Clustering, das die Objekte in verschiedene Klassen einteilt, ohne Beziehungen zwischen den einzelnen Objekten einer Klasse anzunehmen. Einen Überblick über die Funktionen einiger verfügbarer Clusteralgorithmen gibt [68]. GEPAT unterstützt hierarchisches Clustern [95], k-Means-Clustern und die Hauptkomponentenanalyse

(PCA). Weitere Clusterverfahren wie z.B. self-organizing Maps [96] werden derzeit nicht unterstützt.

Clustering-Methoden können in zwei Klassen eingeteilt werden: Unsupervised clustering Methoden ermöglichen die Identifikation und Visualisierung von Gruppenstrukturen im Datensatz ohne vorher verfügbares Wissen von existierenden Gruppen. Supervised clustering Methoden beziehen vorhandenes Wissen, das vom Benutzer angegeben wird, beim Clustering mit ein. Das von GEPAT unterstützte hierarchische Clustern, k-means-Clustern und die Hauptkomponentenanalyse sind unsupervised Clustering Methoden, supervised Clustering Methoden wie z.B. Support Vector Machines [97] werden derzeit nicht unterstützt.

Distanz

Clustering-Algorithmen benötigen eine Möglichkeit, die Distanz zwischen zwei Objekten zu bestimmen, um festzustellen, ob diese zum selben Cluster gehören. Für Microarray-Daten wird hierzu die Distanz zwischen den Expressionsvektoren berechnet. Je nachdem, ob Transkripte oder Proben geclustert werden sollen, werden Transkript- oder Probenvektoren zur Distanzbestimmung herangezogen. Metrische paarweise Distanzen sind z.B.

- die euklidsche Metrik $d_{euc}(\vec{x},\vec{y})=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2}$
- die Manhattan-Metrik $d_{man}(\vec{x},\vec{y})=\sum_{i=1}^{m}|x_i-y_1|$

Daneben gibt es noch eine Reihe korrelationsbasierter Distanz-Maße, die für Microarray-Daten verwendet werden können, u.a.:

- Pearson Korrelationsdistanz

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

- Spearman Korrelationsdistanz

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

wobei $x'_i = rank(x_i)$ den Spearman's rank Korrelationskoeffizient beschreibt.

Standardisierung

Die Distanz zwischen zwei Punkten ist eng verwandt mit der Skalierung der Daten. Standardisierung ist deshalb wichtig, um die Daten vergleichbar zu machen, kann jedoch auch dazu führen, dass Eigenschaften der Daten verloren gehen. Wenn man mit Microarray-Daten arbeitet, kann man Gene und/oder Proben standardisieren. Werden Gene standardisiert, werden die Expressionvektoren wie folgt transformiert:

$$x_i = \frac{x_i - center(x)}{scale(x)}$$

wobei $center(x)$ ein Maß für den Mittelpunkt der Verteilung von x ist, wie der Mittelwert oder Median, und $scale(x)$ ein Maß für die Skalierung, wie die Standardabweichung.

Hierarchisches Clustern

Hierarchisches Clustern ist ein agglomerativer Ansatz, der mit einem Cluster für jeden Expressionsvektor beginnt. Eine paarweise Distanzmatrix wird für alle Cluster berechnet, die zwei ähnlichsten Cluster werden zu einem neuen Cluster vereinigt, der die Objekte des alten Clusters enthält. Anschließend werden die Abstände in der Distanzmatrix neu berechnet und das Verfahren weitergeführt. Der dabei entstehende Baum, das *Dendrogramm*, wird zurückgegeben. Unabhängig von der gewählten Distanzfunktion kann die Berechnung der Distanzen zwischen den Clustern nach verschiedenen Methoden erfolgen:

- Single linkage (Minimum, nächster Nachbar): Die Distanz zwischen zwei Clustern i und j berechnet sich aus der minimalen Distanz zwischen einem Mitglied aus Cluster i und einem Mitglied aus Cluster j. Dies führt dazu, dass auch Cluster vereinigt werden, bei denen nur zwei Mitglieder ähnlich sind.
- Complete linkage (maximum, entferntester Nachbar): Die Distanz zwischen zwei Clustern berechnet sich aus der größten Distanz zwischen Mitgliedern der Cluster.

- UPGMA: Verwendet die Durchschnittswerte des Clusters zur Distanzbestimmung. Verschiedene Methoden existieren zur Bestimmung des Durchschnitts, die am häufigsten verwendete Methode ist die unweighted Pair-Group Method Average (UPGMA). Hierbei wird die Durchschnittsdistanz aus der Distanz zwischen jedem Punkt im Cluster und allen Punkten aus dem anderen Cluster berechnet. Die zwei Cluster mit der geringsten Durchschnittsdistanz werden zu einem neuen Cluster vereinigt.
- Ward´s Methode: Berechnet die Summer der quadrierten Abweichungen vom Mittelwert des Clusters und vereinigt die Cluster derart, dass der kleinste mögliche Anstieg in der Summe der quadrierten Fehler entsteht.

PCA

PCA (Hauptkomponentenanalyse) ist eine explorative Technik, um Muster in Datensätzen durch Dimensionsreduktion zu finden, wobei diejenigen Charakteristiken des Datensatzes erhalten bleiben, die am meisten zu dessen Varianz beitragen. In der Genexpressionsanalyse wird die PCA verwendet, um Redundanz in den Daten zu entfernen und um so das Rauschen im Datensatz zu vermindern. Damit lassen sich Transkripte oder Proben finden, die ein ähnliches Muster besitzen [98]. PCA ermöglicht die Projektion hochdimensionaler Datensätze auf einen reduzierten, einfach zu visualisierenden Raum. Mathematisch wird eine Hauptachsentransformation durchgeführt: Man minimiert die Korrelation mehrdimensionaler Merkmale durch Überführung in einen Vektorraum mit neuer Basis. Dabei tragen die Principal Components in absteigender Reihenfolge zur Variabilität der Daten bei. Typischerweise wird das Ergebnis der PCA als zweidimensionaler Plot angezeigt, bei dem die erste Hauptachse auf der x-Achse und die zweite Hauptachse auf der y-Achse aufgetragen wird. Die Berechnung wird durch eine Eigenwert-Dekomposition der Datenmatrix durchgeführt. Da die Hauptkomponenten normalerweise nicht inhaltlich interpretiert werden können, spricht man davon, dass ihnen keine verständliche Hypothese zugeschrieben werden kann.

K-means Clustering

Oftmals ist bereits vor der Clusteranalyse bekannt, aus wie vielen Cluster die Daten bestehen sollen, z.B. wenn verschiedene biologische Zustände in einem Experiment

betrachtet werden. Hierfür kann das k-means Clustering Verfahren verwendet werden. Die Daten werden in eine vom Benutzer vorgegebene Anzahl k Gruppen eingeteilt. Die Cluster enthalten keine gemeinsamen Elemente und die k Gruppen zusammen enthalten den gesamten Datensatz. Das k-means Clustering ist ein iteratives Verfahren. Zuerst werden alle Objekte zufällig einem der k Cluster zugewiesen. Anschließend wird ein Durchschnittswert für jeden Cluster gebildet, um die Distanz zwischen Clustern bestimmen zu können. Dann werden in einer iterativen Methode Objekte zwischen den Clustern bewegt und die inter- und intra-Clusterdistanzen für jeden Schritt werden berechnet. Objekte dürfen nur in einem neuen Cluster verbleiben, wenn sie diesem ähnlicher als dem sind, aus dem sie stammen. Nach jedem Schritt werden die Durchschnittswerte der Cluster neu berechnet. Dieses Verfahren wird fortgesetzt, bis die Bedingungen ein weiteres Austauschen nicht mehr zulassen.

2.4.6 GO Term Enrichment Analyse

Ein üblicher Ansatz zur Analyse von Gen-Teilmengen in Microarraydaten ist der Vergleich der relativen Häufigkeit von GO-Kategorien (vgl. Kapitel 2.3.4) einer Teilmenge von Genen mit den GO-Kategorien einer Referenzmenge, die meist aus dem gesamten Datensatz besteht. Es gibt eine Reihe von Tools, die diese Art von Analyse für eine gegebene Genliste vornehmen können [99], jedoch bisher kaum integrierte Ansätze.

Ziel ist es, eine signifikante Erhöhung oder Minderung von GO- Kategorien in einer Menge von Genen, meist werden die signifikant differentiell exprimierter Gene verwendet, festzustellen. Die verfügbaren Tools verwenden hierzu verschiedene statistische Tests, wie den Binomialtest, den χ^2-Test, Fisher's exact test und den hypergeometrischen Test [100].

Aus den verschiedenen möglichen statistischen Tests, die von diesen Tools durchgeführt werden, wird in GEPAT ein Signifikanztest basierend auf einer hypergeometrischen Verteilung durchgeführt, da dieser ein adäquates Modell für die Wahrscheinlichkeit ist, dass eine Kategorie k-mal nur durch Zufall in der Liste der interessanten Gene auftaucht. Die Nullhypothese H_0 ist es, dass die Zugehörigkeit eines Gene zu einer bestimmten GO-Kategorie und zur Menge der ausgezeichneten Gene unabhängig sind, d.h. dass die ausgezeichneten Gene zufällig aus der Genpopulation ausgewählt wurden. Bezeichnet man die Anzahl aller Gene mit N, die Anzahl aller Gene, die zur getesteten GO-Kategorie

gehören, mit n, die Anzahl der ausgezeichneten Gene mit K und die Anzahl der ausgezeichneten Gene, die zur getesteten GO-Kategorie gehören, mit k, so ergibt sich

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

wobei κ die Zufallsvariable ist, deren Realisierung der beobachtete Wert k ist. Bei der Wahl der kritischen Region unterscheidet man einseitige Test, die verwendet werden können, um entweder Erhöhung oder Minderung von GO-Kategorien zu bestimmen, oder zweiseitigen Tests, die sowohl Erhöhung als auch Minderung bestimmen.

2.5 Datenhaltung

2.5.1 MIAME-Standard

Die Beschreibung von Genexpressionsdaten ist wesentlich komplexer als die von Sequenzdaten, da sie ohne detaillierte Beschreibung der Bedingungen, unter denen sie gewonnen wurden, wertlos sind. Anders als das Genom eines Organismus unterliegt das Transkriptom ständigen Änderungen, die sich u.a. durch den Zustand des Organismus und dessen Behandlung ergeben. Microarray-Experimente können deshalb nur ausgewertet werden, wenn der Zustand der Proben, deren Expression gemessen wird und die Messbedingungen bekannt sind. Deshalb ist es notwendig, nicht nur die Genexpressionsmatrix, sondern auch eine detaillierte Beschreibung, wie die Messwerte ermittelt wurden, aufzuzeichnen, um eine spätere Verifikation und Weiterverarbeitung der Daten zu ermöglichen. Um hier einen einheitliches Verfahren mit einer einheitlichen Notation zu ermöglichen, wurde der MIAME Standard entwickelt.

MIAME (Minimal Information about a Microarray Experiment) [101] ist ein Datenstandard, der von der MGED (Microarray gene expression data) Society [102] entwickelt wurde, um eine Richtlinie zur Beschreibung von Microarray-Experimenten zu schaffen. Bis zur Definition von MIAME gab es keinen einheitlichen Standard zur Repräsentation und zum Austausch von Microarraydaten.

MIAME beschreibt die Minimalinformation, die für eine Interpretation und unabhängige Überprüfung von Microarrayexperimenten notwendig ist. Die erste Version wurde 2001 verabschiedet, aktuelle ist Version 1.1 (Draft 6, 1.4.2002). Erweiterungen des Standards

erlauben es, zusätzlich noch CGH Microarrays und ChIP-Chip Experimente beschreiben zu können. Mit der wachsenden Verfügbarkeit anderer Hochdurchsatz-Technologien wurde offensichtlich, dass ein gemeinsamer Standard benötigt wird, um Teile von Experimenten, die verschiedenen Technologien gemeinsam sind, zu beschreiben. Dazu wurden für verschiedene zur biologischen Forschung verwendete Technologien die Reporting Structure for Biological Investigation (RSBI) [103] geschaffen, deren Aufgabe darin besteht, eine gemeinsame Terminologie zwischen den verschiedenen biologischen Standardisierungsverfahren zu finden. Eine Version 2.0 von MIAME, die auf dem Functional Genomics Experiment Object Model (FuGE) [104] und der Functional Genomics Ontology (FuGO) [105] basiert, befindet sich derzeit im Entwurf. Einen Überblick über die Entwicklung des Standards gibt [106].

MIAME spezifiziert nicht das Format, in dem die Informationen gespeichert werden, sondern nur den Inhalt, der gespeichert wird. Die Forderungen von MIAME sind

1. zu jedem Experiment Informationen zu speichern, die zur Interpretation ausreichend sind, und die detailliert genug sind, um Vergleiche zu ähnlichen Experimenten und eine Wiederholung des Experiments zu ermöglichen.
2. die Informationen in einer Art und Weise zu strukturieren, so dass eine einfache Abfrage so wie auch automatische Datenanalyse und Datamining möglich sind.

Eine Sammlung von Genexpressionsdaten kann abstrakt als Expressionsmatrix betrachtet werden, zusätzlich zu dieser Matrix sind noch Informationen zu den Genen nötig, deren Expression gemessen wurde, sowie die experimentellen Bedingungen, unter denen die Proben genommen wurden. Damit können die Informationen, die für die Beschreibung eines Microarray-Experiments benötigt werden, in drei Bereiche eingeteilt werden: Genannotation, Probenannotation und Genexpressionswerte.

Zur Beschreibung der Genexpression gibt es mindestens drei Ebenen, die relevant zur Experimentbeschreibung sind: Die Images aus dem Microarray-Scanner, die Ergebnisse der Image-Analyse und die daraus abgeleiteten Werte, welche die Genexpressionsmatrix bilden. Eine Reihe nichtstandardisierter Schritte sind notwendig, um die Ebenen ineinander zu überführen.

Ein großer Teil des MIAME-Standards behandelt die Beschreibung der Proben und der Bedingungen, unter denen diese gewonnen wurden. MIAME definiert ein Microarray-

Experiment als Menge von einer oder mehrerer Hybridisierungen, von denen jede eine oder mehr Proben mit einem oder mehreren Arrays in Beziehung setzt. Auch Genannotation ist ein Teil des Standards. Obwohl Datenbanken hierfür existieren, können die Verbindungen zwischen Gensonden auf dem Array und Verbindungen zu den entsprechenden Genen komplex sein, so dass hier Informationen zu den Genen im Standard gespeichert werden.

Die Minimalinformation für ein microarray-basiertes Experiment enthält eine Beschreibung der folgenden sechs Bereiche, die in Abbildung 11 schematisch dargestellt werden:

1. Experimentdesign: Beschreibt das Experiment als Ganzes. Normalerweise besteht ein Experiment aus verschiedenen Hybridisierungen, die eine gemeinsame biologische Fragestellung als Grundlage besitzen. Das Experimentdesign beschreibt, welche Proben mit welchen Arrays hybridisiert wurden.
2. Arraydesign: Beschreibt die Arrays, die im Experiment verwendet wurden, inklusive der Gensonden und dem Layout der Arrays. Zu jedem verwendeten Array wird ein Verweis auf ein Referenzdesign gespeichert. Das Referenzdesign enthält Informationen zu den Elementen auf dem Array, wie die DNA-Sequenz, oder Qualitätsindikatoren.
3. Proben: Beschreibt das biologische Material, aus dem das Genexpressionprofil extrahiert wird. Umfasst die Quelle der Originalprobe, wie Organismus und Zelltyp, sowie alle biologischen in vivo und in vitro Behandlungen die durchgeführt wurden und die technische Extraktion und Markierung der Nukleinsäuren.
4. Hybridisierung: Beschreibt die Laborbedingungen, unter denen die Hybridisierungen ausgeführt wurden. Kritische Hybridisierungsparamter müssen explizit angegeben werden.
5. Daten: Beschreibt die Experimentergebnisse. Enthält die Originalscans der Arrays (Images), die Microarray-Quantifizierungsmatrizen aus der Image-Analyse und die fertige Genexpressionsmatrix nach Normalisierung. Mögliche Rechenschritte, die zur Bearbeitung der Messwerte durchgeführt werden, sind in diesem Abschnitt festgehalten.
6. Normalisierung: Beschreibt Parameter, die relevant für die Normalisierung sind,

wie die Normalisierungsstrategie, die verwendeten Normalisierungs- und Qualitätssicherungsalgorithmen, die Positionen von Kontrollelementen und Informationen, wie Kontrollinformationen vor der Hybridisierung in die Proben eingefügt werden.

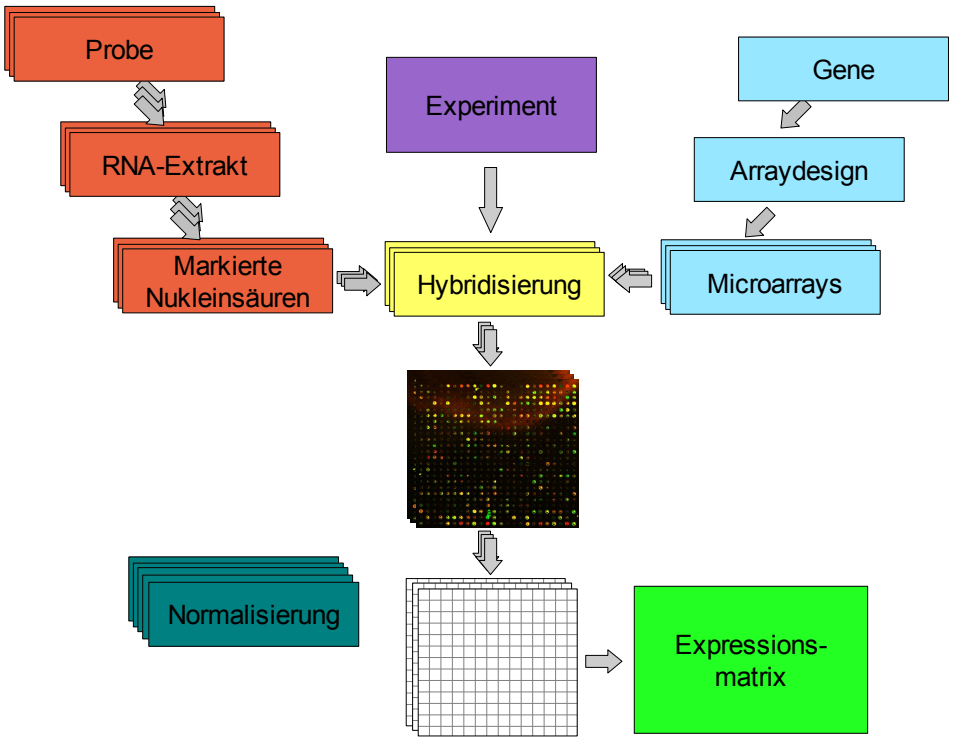


Abbildung 11: MIAME-Standard im Überblick
Jedes Microarray-Experiment kann als eine Anzahl von Hybridisierungen beschrieben werden, die ein oder mehrere markierte mRNA-Extrakte mit Microarrays verbindet. Die mRNA-Extrakte werden aus biologischen Proben gewonnen. Die hybridisierten Microarrays werden gescannt, die Rohdaten werden normalisiert und zur Expressionsmatrix vereinigt.

Da die Microarray-Technologie einer schnellen Entwicklung unterworfen ist, wurde keine bestimmte Microarray-Plattform, Software oder Datenanalysemethoden explizit in der MIAME-Spezifikation festgelegt. Stattdessen sollen die Daten detailliert genug beschrieben werden, um Interessenten alle benötigten Informationen zur Nachahmung eines Experiments zur Verfügung zu stellen.

MAGE-OM [107] ist ein datenzentriertes, objektorientiertes Modell für den durch MIAME definierten Standard. Dieses Datenmodell kann direkt auf XML-Dateien im MAGE-ML-Format abgebildet werden, die Microarray-Experimente und Daten zu beschreiben. Die MGED Ontologie [108] (MO) beschreibt einheitliche Begriffe und Annotationsregeln für Microarray-Experimente und ermöglicht so einen eindeutigen Datenaustausch.

Zum Zugriff auf MAGE-ML steht eine Reihe an Softwarepaketen, MAGE-STK genannt, zur Verfügung, die das MAGE-OM implementieren und ein API hierzu bereitstellen [109]. Unterstützt werden derzeit die Programmiersprachen Java, Perl und C++. Das API stellt Lese- und Schreibfunktionen für MAGE-ML Dateien bereit und bildet so eine Abstraktionsschicht für diese XML Dateien. Neben den MAGE-STK existieren verschiedene andere Tools, um MAGE-ML Dateien in bestehende Softwarepakete zu integrieren. So existieren auch Softwarebibliotheken, die MAGE-ML Dateien in Bioconductor einlesen können [110].

Probleme bereiten kann jedoch die Komplexität von MAGE-ML, die es Forschungsgruppen nicht ohne weiteres möglich macht, ihre Ergebnisse MIAME-konform zu veröffentlichen. Um eine einfachere MIAME-Notation der Ergebnisse zu ermöglichen, wurde das MAGE-TAB (MicroArray Gene Expression Tabular) Format geschaffen [111]. Es basiert auf der Verwendung von Tabulator-getrennten Tabellen-Dateien zur Erfassung von Experimentdesign, Arraybeschreibung und Daten. Die Rohdaten können in ihren ursprünglichen Formaten beigefügt werden. Protokolle werden durch nichtstrukturierten Text beschrieben. Probleme besonderer Art können hierbei entstehen, wenn die automatische Formatierung der Tabellenkalkulation mit den Gennamen in Konflikt gerät [112].

Zur Veröffentlichung der Daten im MIAME-Format stehen unterschiedliche Datenbanken zur Verfügung, die wichtigsten, öffentlich zugänglichen sind ArrayExpress und GEO.

2.5.2 ArrayExpress

Die Genexpressionsdatenbank ArrayExpress [113] besteht aus zwei Teilen, dem ArrayExpress Repository, einem MIAME-kompatiblen öffentlichem Archiv für Microarray-Daten und dem ArrayExpress Data Warehouse, einer Datenbank von Genexpressionsprofilen, die aus dem Repository ausgewählt und reannotiert wurden.

ArrayExpress ist mit dem Ziel entstanden, der wissenschaftlichen Gemeinschaft als Repositorium für publizierte Daten zu dienen, um einfachen Zugriff auf Genexpressionsdaten zu bieten, und um eine Verbreitung von Microarray-Designs und Experimentprotokollen zu fördern. ArrayExpress bietet die Möglichkeit, Daten entweder im MAGE-ML Format oder über ein internetbasiertes Tool, MIAMExpress, aufzunehmen.

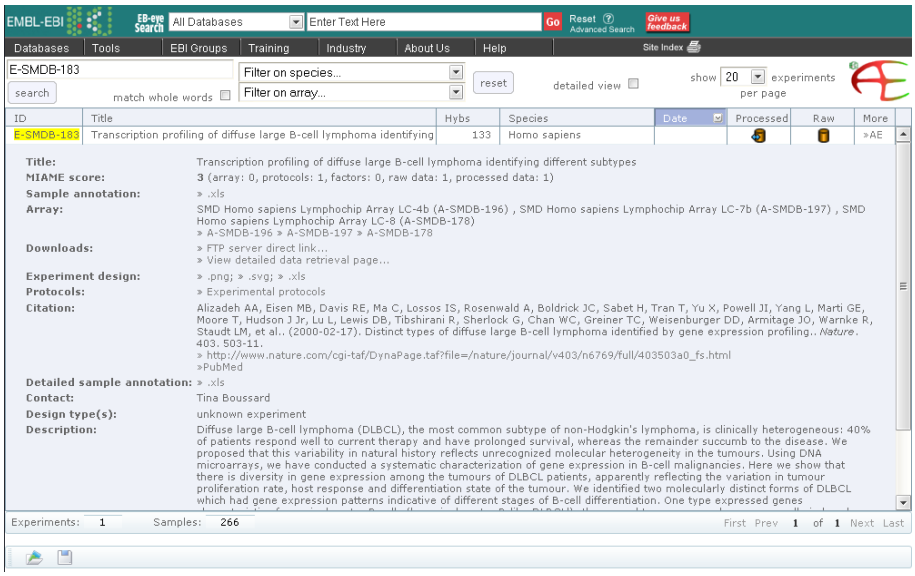


Abbildung 12: ArrayExpress
Der ArrayExpress-Datenbankeintrag zu den Genexpressionsdaten aus [22]

Das ArrayExpress Repository erlaubt es, den gesamten Inhalt der Datenbank durchzublättern oder die Datenbank über eine Texteingabe abzufragen. Die Ergebnisse enthalten Experimentbeschreibung und Publikationsreferenzen, falls vorhanden können die Originaldatenfiles heruntergeladen werden. Zu jedem Experiment stehen eine Tabelle und Graphen zur Verfügung, die Probeeigenschaften und das Experimentdesign beschreiben. Expressionsdaten zu verschiedenen Arrays können in eine Datei exportiert werden, oder direkt in das Programm ExpressionProfiler [7] übernommen werden. Abbildung 12 zeigt einen Datenbankeintrag.

Das MAGE-TAB Format wird voraussichtlich im Jahr 2007 Einzug in die ArrayExpress-Datenbank finden [114].

2.5.3 Gene Expression Omnibus

Auch die Datenbank Gene Expression Omnibus (GEO) des NCBI [115] dient hauptsächlich als öffentliches Archiv und als Verteilungsinstanz, wie ArrayExpress enthält auch sie Data-Mining-Tools, die ein einfaches Abfragen, Filtern und Untersuchen der Daten nach speziellen Fragestellungen ermöglichen.

GEO ist aktuell die größte eigenständige Datenbank für öffentliche Genexpressionsdaten. Sie unterstützt die Dateneingabe nach MIAME-Spezifikation. GEO wurde mit einer

flexiblen Struktur entwickelt, um auf die Komplexität der Eingabedaten Rücksicht zu nehmen, verschiedene Arten von Daten werden unterstützt.

Abbildung 13: Gene Expression Omnibus
Der GEO Datenbankeintrag zu den Expressionsdaten aus [22]. Gezeigt werden die Informationen zu einer der drei in diesem Experiment verwendeten Microarraytypen.

GEO ermöglicht zwei verschiedene Arten der Datendarstellung:

- Eine experimentzentrierte Darstellung, welche eine gesamte Studie umfasst. Diese Information enthält eine Experiment-Übersicht, eine Liste der Variablen, Zugriffsmöglichkeit zu zusätzlichen Informationen, Downloadmöglichkeiten und verschiedene Anzeige- und Analysemöglichkeiten für die Expressionsdaten. Eine solche Darstellung zeigt Abbildung 13.
- Eine genzentrierte Darstellung enthält eine Anzeige der quantitativen Genexpressions-Messwerte für ein Gen über einen Datensatz, zusätzliche Informationen sowie eine Grafik, die das Expressionslevel und den Rang des Gens in jeder Probe des Datensatzes anzeigt. Geneannotation erfolgt über die Sequenzidentifier der Gensonden durch Abgleich mit den Entrez Gene [116] und UniGene Datenbanken.

2.6 Java Platform, Enterprise Edition

Die Java Platform, Enterprise Edition, oder auch Java EE ist eine Softwarespezifikation,

die einen Rahmen zur Programmierung und Ausführung verteilter, mehrschichtiger Java-Anwendungen bietet [117]. Die Spezifikation liefert Schnittstellen für die hauptsächlich auf modularen Softwarekomponenten basierenden, auf einem Applikation Server ausgeführten Anwendungen, und sorgt so für eine Interoperabilität von Software unterschiedlicher Hersteller. Der Java EE Applikation Server stellt eine Reihe von Funktionalitäten zur Verfügung:

- Absicherung der Benutzer untereinander und der Geschäftslogik.
- eine hohe Verfügbarkeit und gute Skalierbarkeit
- Management der Softwarekomponenten über den gesamten Lebenszyklus
- Kapselung des Zugriffs auf die Ressourcen

Es sind zahlreiche Implementierungen für Java-EE-Server verfügbar, teils proprietär, teils in Form frei verfügbarer Open-Source-Lösungen [118]. Eine Referenzimplementierung wird von Sun Microsystems zur Verfügung gestellt [119]. Nicht alle verfügbaren Server decken die Spezifikation von Java EE vollständig ab.

2.6.1 Drei-Schicht-Architektur

In den meisten Fällen werden Anwendungen auf einem Applikation-Server als Mehrschicht-Anwendungen entwickelt, eingeteilt in drei Schichten, der Präsentationsschicht, Logikschicht und Datenschicht:

- Die Präsentationsschicht ist für die Repräsentation der Daten, Benutzereingaben und die Darstellung der Benutzerschnittstelle verantwortlich. Sie wickelt die Interaktion zwischen dem Benutzer und der Software ab und besteht meist aus einer graphischen Benutzeroberfläche oder einem Webbrowser in Verbindung mit entsprechender Software auf dem Server.
- Die Logikschicht beinhaltet alle Verarbeitungsmechanismen der Software. Hier ist die Geschäftslogik des Programms vereint, sie führt Berechnungen anhand der Benutzereingaben und der gespeicherten Daten durch.
- Die Datenschicht hält die Daten des Programms und ist für die Persistenz verantwortlich. Sie kann mit anderen Teilen des Systems kommunizieren, welche Transaktionsverwaltung und Nachrichtenübermittlung durchführen. Meist kommt

zur persistenten Speicherung der Daten ein Datenbankmanagementsystem zum Einsatz.

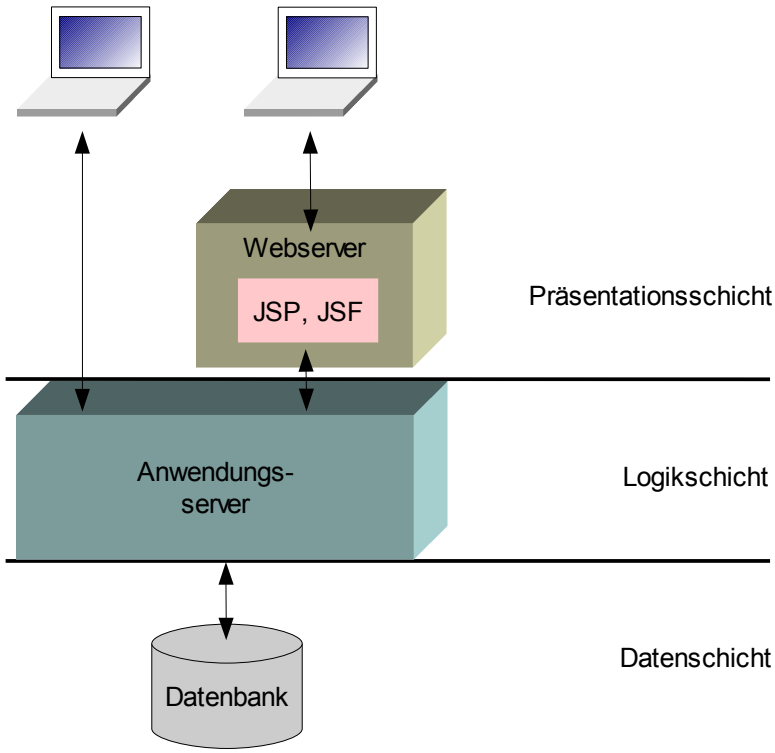


Abbildung 14: Einteilung in eine 3-Schichten-Architektur
Die drei Schichten können sich wie dargestellte auf verschiedenen Rechnern, oder auch auf dem gleichen Rechner befinden. Die Präsentationsschicht von Webanwendungen teilt sich meist in einen Teil auf dem Webserver und einem Teil im Webbrowser auf. Bei anderen Anwendungen kann die Präsentationsschicht auch komplett auf dem Client ausgeführt werden.

Ein Schema der Drei-Schicht-Architektur zeigt Abbildung 14. Der Vorteil der Aufteilung des Programms in drei Schichten liegt zum einen in der Möglichkeit, Präsentationsschicht und Datenschicht austauschen zu können. Dies erlaubt eine Änderung der Programmoberfläche, ohne Änderungen an der Logik durchführen zu müssen. Ein weiterer Vorteil liegt darin, dass sich die drei Schichten nicht unbedingt auf derselben Hardware befinden müssen. So können sich die Präsentationsschicht auf dem Client-Rechner, die Logikschicht auf dem Applikations-Server und die Datenschicht auf einem speziellen Datenbankserver befinden.

Die Java EE kapselt die verschiedenen Schichten in wiederverwendbare Komponenten. Die Interaktion mit dem Benutzer kann durch Webseiten über Java Applets, Java Servlets,

JSP und JSF (s.u.) oder durch eine eigenständige Java-Anwendung erfolgen. Die Logik des Programms kann in Enterprise JavaBeans (EJB) gekapselt werden. Beide Arten von Komponenten werden durch offene Standards miteinander verknüpft. Der Web-Zugriff auf Java EE-Komponenten erfolgt über einen Web-Container. Neben der Funktion als Servlet bzw. JSP-Container erlaubt dieser noch Zugriff auf die erweiterte Java EE Funktionalität.

2.6.2 JavaBeans

JavaBeans sind Klassen der Programmiersprache Java, die bestimmte Konventionen erfüllen müssen, um automatisierten Zugriff auf ihre Eigenschaften und Methoden zu ermöglichen. Sie zeichnen sich durch folgende Eigenschaften aus:

- Öffentlicher Standardkonstruktor
- Serialisierbarkeit
- Öffentliche Zugriffsmethoden

Mithilfe des Reflexion-API von Java können Informationen über ein Bean, seine Eigenschaften und Operationen ermittelt werden. Da die Bedingungen für ein Bean nur durch Konventionen,und nicht durch Interfaces bestimmt sind, werden Beans auch als „Plain Old Java Objects“ (POJO), also gewöhnliche Java-Objekte bezeichnet, die speziellen Namenskonventionen entsprechen.

2.6.3 Enterprise JavaBeans

Enterprise JavaBeans (EJB) sind in Java geschriebene, standardisierte Komponenten innerhalb eines Java EE Servers, die innerhalb eines EJB-Container ablaufen. Sie kapseln die Geschäftslogik der Anwendung und können entweder vom lokalen Rechner oder über Prozess- und Rechnergrenzen hinweg angesprochen werden. Sie vereinfachen die Entwicklung großer, verteilter Anwendungen, da auf diese Weise die Geschäftslogik mit beliebigen Client-Systemen kombiniert werden kann und durch die Wiederverwendbarkeit der EJB neue Anwendungen aus bestehenden Beans zusammengesetzt werden können.

Die EJB Spezifikation [120] beschreibt drei verschiedene Typen Enterprise JavaBeans:

Entity Beans modellieren die persistenten Daten des Systems. Meist bilden sie physikalisch vorhandene Objekte ab und repräsentieren Datensätze aus einer

Datenbank. Die Persistenz wird dabei vom Entwickler selbst implementiert, oder wird durch den EJB-Container gestellt.

Session Beans beschreiben den Arbeitsfluss, der durchgeführt werden muss, um eine bestimmte Aufgabe zu erfüllen, und interagieren mit Entity Beans und anderen Session Beans. Man unterscheidet zwei verschiedene Arten von Session Beans:

- Zustandsbehaftete Session Beans können Informationen aus einem Methodenaufruf speichern, so dass diese bei einem späteren Aufruf von der gleichen oder anderen Methoden wieder zur Verfügung stehen. Der Zustand wird durch die Vergabe einer eindeutigen ID umgesetzt, die für die Dauer einer Session gültig ist. Im Vergleich zu Entity Beans, welche Informationen persistent speichern, gehen die Informationen zustandsbehafteter Session Beans verloren, sobald die Session geschlossen wird.
- Zustandslose Session Beans speichern hingegen keine Informationen, so dass jedem Aufruf alle Informationen als Parameter übergeben werden, die für die Abarbeitung dieses Aufrufs benötigt werden. Da so alle Instanzen eines zustandslosen Session Beans gleich sind, kann jede Instanz einem beliebigen Client zugewiesen werden. Sie ermöglichen deshalb eine gute Skalierbarkeit für Anwendungen, die eine große Anzahl an Clients besitzen. Zustandslose SessionBeans können über einen Web Service aufgerufen werden, wobei die Schnittstelle des Web Service auf die Schnittstellen des EJB abgebildet wird.
- *Message Driven Beans* sind Komponenten, die in EJB-Systeme über asynchrone Kommunikation mit dem Java Messaging Service (JMS) aufgerufen werden.

2.6.4 Java Servlets

Die Java Servlet API wurde geschaffen, um dynamische Webseiten mit der Programmiersprache Java erzeugen zu können [121]. Der generierte Inhalt ist meist HTML, kann aber anderer Art sein, und z.B. Bilddateien beschreiben. Servlets sind das Java-Äquivalent zu dynamischen Webtechnologien wie CGI. Sie bieten die Möglichkeit, Informationen über mehrere Aufrufe hinweg in einem Session-Objekt zu speichern. Die Zuordnung zwischen Client und Session-Information erfolgt über eine Benutzerkennung in der URL oder in einem HTTP Cookie. Zur Verwendung der JavaServlet-Technologie wird ein Servlet Container benötigt, ein Teil des Webservers, der mit den Servlets interagiert.

Den Ablauf eines Java Servlet-Aufrufs zeigt Abbildung 15.

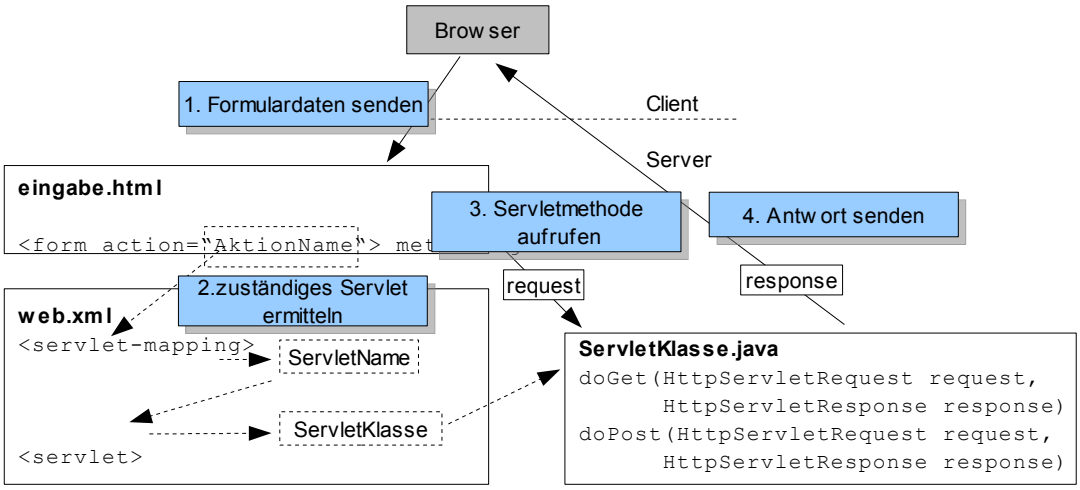


Abbildung 15: Ablauf eines Servlet-Aufrufs
Anhand der angeforderten HTML-Seite wird nach einem Servlet gesucht, das sich für diesen Namen registriert hat. Die entsprechende Servletklasse wird aufgerufen, die Antwort wird an den Client gesendet.

Verschiedene Teile einer Webapplikation benötigen Zugriff zu denselben Datenobjekten. Die Servlet API-ermöglicht dies durch die Verwendung von Daten im applikationsweiten Kontext, im Kontext der Benutzersession oder nur im Kontext der aktuellen Anfrage. Diese Kontexte werden als *application scope*, *session scope* und *request scope* bezeichnet. Abbildung 16 illustriert die Geltungsbereiche der Kontexte.

Neben den Servlets definiert die Spezifikation noch zwei weitere Typen von Komponenten: *Filter* und *Listener*. Filter ermöglichen es, die Anfrage für eine Seite zuerst zu filtern und bietet so einfache Möglichkeiten für Zugriffskontrolle oder Logging. Listeners erlauben es der Applikation, auf spezielle Ereignisse, wie z.B. die Erstellung einer Session, zu reagieren.

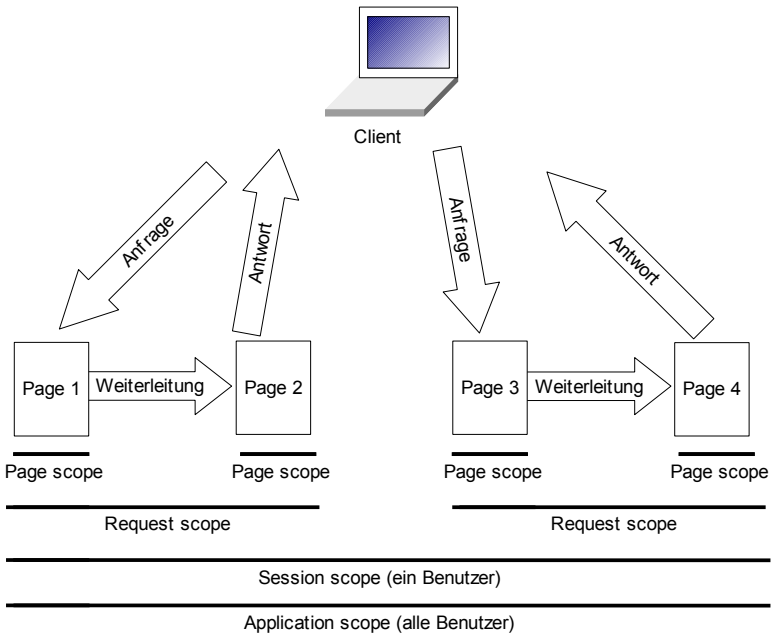


Abbildung 16: Gültigkeit der Kontexte in einer Webapplikation
Daten können entweder nur für eine Seite gültig sein (Page), während einer Anfrage (Request), die auch in interner Weiterleitung bearbeitet werden kann, für alle Anfragen eines Benutzers (Session) oder für alle Anfragen an die Webanwendung (Applikation)

2.6.5 Java ServerPages

Da die Erstellung dynamischer Webseiten mit Java Servlets sehr unübersichtlich werden kann und keine Trennung zwischen dem Design der Webseite und dem Code des Servlets möglich ist, wurden die Java ServerPages (JSP) entwickelt. Sie erlaubt es, Java-Code und spezielle JSP-Aktionen in statischen Inhalt der Webseite (Template text) einzubetten. Dies hat den Vorteil, dass die Logik unabhängig vom Design implementiert werden kann.

Ein Webcontainer mit JSP-Support fängt alle Zugriffe auf JSP Seiten ab, wandelt diese Seiten mit einem JSP Compiler in Java Servlet Code um und compiliert das Servlet (translation phase). Die Seite wird anschließend über das Servlet ausgegeben (request processing phase). Die Umwandlung erfolgt beim ersten Aufruf einer JSP Seite, was zu leichten Verzögerungen führen kann. Der Ablauf eines JSP-Seitenaufrufs wird in Abbildung 17 gezeigt.

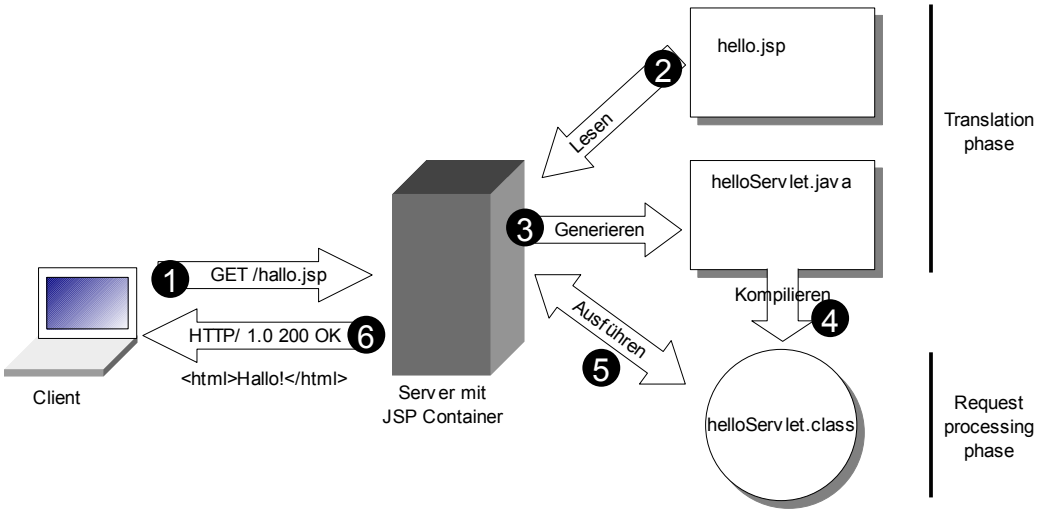


Abbildung 17: Ablauf eines JSP Seitenaufrufs
Wird eine Seite beim Server angefordert (1), so prüft dieser, ob die Seite schon erstellt wurde. Ist sie noch nicht vorhanden, wird in der Translation-Phase die JSP-Datei gelesen (2) und ein Servlet erstellt (3 und 4), dessen Ausgabe an den Benutzer zurückgegeben wird (5 und 6)

Der Syntax von JSP besteht aus drei verschiedenen Arten von Elementen: *directive*, *action* und *scripting*-Elementen, zusätzlich sind noch Ausdrücke in der *Expression Language* (EL) möglich.

- Directive Elements spezifizieren Informationen über eine Webseite, die sich über verschiedene Anfragen hinweg nicht ändern. Sie können z.B. verwendet werden, um eine Seite aus verschiedenen Segmenten aufzubauen. Sie werden nur in der translation phase ausgeführt.
- Action Elements werden in der Request Processing Phase ausgeführt, wenn eine JSP Seite angefordert wird. Eine Aktion kann beispielsweise Text in die Webseite schreiben, Parameter in JavaBeans setzen oder Methoden von JavaBeans aufrufen. Zusätzlich zu den vordefinierten JSP-Aktionen gibt es die Möglichkeit, benutzerdefinierte JSP-Aktionen zu verwenden. Dazu können spezielle JSP-Tag-Bibliotheken zur Verfügung gestellt werden, die den JSP-Syntax erweitern. Häufig findet die JavaServer Pages Standard Tag Library (JSTL) Verwendung.
- Scripting Elements erlauben es, kurze Stücke Java-Code in eine JSP-Seite einzubetten. Wie Action Elements werden auch sie in der Request Processing Phase ausgeführt. Scripting Elements sollten sparsam eingesetzt werden, da sie

die Seite unübersichtlich machen und bei exzessiver Verwendung kein Vorteil mehr gegenüber der Verwendung normaler Servlets besteht.

- Die Expression Language ist eine einfache Sprache, die Zugriffe auf die Daten der HTML-Anfrage und Applikationsobjekte erlaubt.

2.6.6 Java ServerFaces

Java ServerFaces [122] (JSF) ist ein Java-basiertes Web Application Framework, das die Entwicklung von Benutzerschnittstellen für Java EE Anwendungen vereinfacht. Es beinhaltet die Darstellung von Komponenten der Benutzerschnittstelle, die Verwaltung von deren Status, das Event-Handling, die Validierung der Eingabe und die Navigation. Eine Reihe von Standardkomponenten steht zur Verfügung, welche die Benutzerschnittstelle durch bestehende HTML-Elemente abbilden. JSF unterstützt *Managed Beans*, global verwaltete JavaBeans. Über ValueBinding können diese JavaBeans mit Komponenten verknüpft werden, so dass eine Kopplung zwischen dem Modell und der Benutzerschnittstelle entsteht.

JSF unterstützt JSP als Präsentationsschicht, wobei die JSF-Komponenten durch JSP action-Komponenten abgebildet werden. JSF implementiert das Model-View-Controller Design Pattern. Design Patterns beschreiben bewährte Lösungen für wiederkehrende Probleme und benennen, abstrahieren und identifizieren die relevanten Aspekte einer allgemeinen Entwurfsstruktur [123]. Das Model-View-Controller Pattern [124] wurde für graphische Benutzeroberflächen entwickelt und kommt in nahezu allen modernen Programmiersprachen zum Einsatz. Die Grundidee besteht darin, Daten und Logik, die Präsentation der Daten und die Interaktion mit den Daten in verschiedene Bereiche einzuteilen, die *Model*, *View* und *Controller* genannt werden.

- Das Modell repräsentiert die Geschäftsdaten und besitzt Regeln, wie diese Daten verwendet werden können. Es hat keine Informationen über View und Controller, weiß also nicht, wie die Daten angezeigt werden oder wie die Daten über das Benutzerinterface geändert werden können
- Die View ist für die Darstellung der Daten des Modells verantwortlich. Ändert sich das Modell, wird auch die View angepasst. Sie ist jedoch nicht für die Steuerung verantwortlich.

- Der Controller kennt das Modell und über seine Schnittstelle werden in der View gemachte Änderungen auf das Modell übertragen. Er nimmt die Benutzerinteraktionen entgegen und reagiert auf diese.

Die Verwendung des MVC-Design-Patterns erlaubt es, das Datenmodell zu ändern, ohne Änderungen am Benutzerinterface durchführen zu müssen. Ebenso kann die Ansicht ausgetauscht werden, ohne dass Modell ändern zu müssen. Auch können mehreren View-Control-Paaren dasselbe Modell als Grundlage verwenden..

In JSF wird das Modell durch Eigenschaften von JavaBeans definiert. Die Komponenten der Benutzerschnittstelle geben an, welche Ereignisse auftreten und welche Event-Listener diesen Ereignissen zugeordnet werden können. Mit den Event-Listenern können Eigenschaften der Applikations-Objekte geändert oder Methoden, welche die Daten bearbeiten, ausgeführt werden. Eine eigenständige Renderer-Klasse ist für die Erstellung der View verantwortlich, so dass Komponenten auf verschiedene Art und Weise dargestellt werden können.

Neben der Erzeugung der graphischen Benutzerschnittstelle bietet JSF noch die Möglichkeit, Validatoren und Konverter für die vom Benutzer eingegebenen Daten zu verwenden.

Wird eine JSF-Seite das erste Mal angefordert, wird sie als reguläre JSP-Seite behandelt. Tag-Handler für die JSF-Actions erstellen ihre entsprechenden JSF-Komponenten, konfigurieren sie entsprechend der Action-Tag-Attribute und fordern sie auf, sich selbst darzustellen. Komponenten können dabei ineinander verschachtelt sein und bilden einen Komponentenbaum. Template Text und Inhalt aus nicht JSF-Tags werden mit dem Inhalt gemischt, der durch die JSF Tags generiert wird.

Führt der Benutzer Eingaben auf einer Seite durch und überträgt diese auf den Server, so wird diese Anfrage in verschiedenen Phasen bearbeitet, wie in Abbildung 18 illustriert:

- In der Restore View Phase werden die Komponenten, welche zur Erstellung der Seite verwendet wurden, aus den auf dem Server gespeicherten Seiten wiederhergestellt
- In Apply Request Values Phase wird der Komponentenbaum durchlaufen, jede Komponente prüft die Anfrage nach eigenen Parametern und liest diese aus.

- In der Process Validation Phase werden die neuen Werte der Komponenten mit Validatoren überprüft.
- In der Update Model Values Phase werden die Werte der Komponenten an das angebundene Modell weitergereicht.
- Methoden der angebundenen Applikations-Objekte werden in der Invoke Application Phase aufgerufen, falls Aktionen auf den angebundenen Komponenten ausgeführt wurden.
- In der Render Response Phase wird schließlich die Antwort an den Client gesendet, wobei entweder dieselbe Ansicht, oder eine neue verwendet wird.

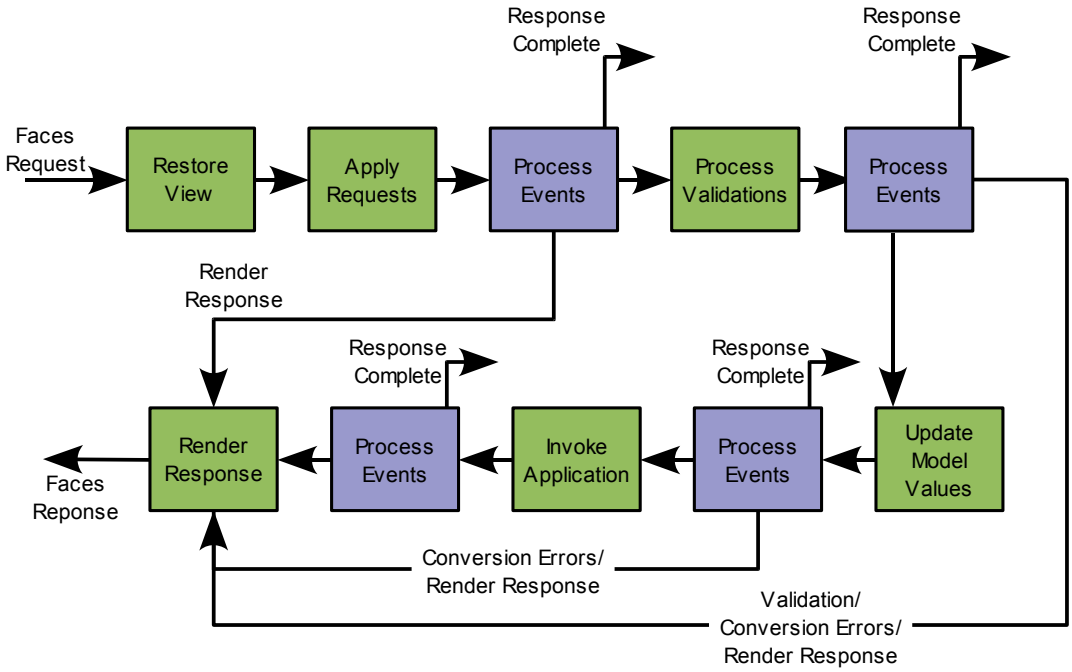


Abbildung 18: Ablauf einer JSF-Anfrage
Die Anfrage durchläuft verschiedene Phasen. Der Ablauf kann in der Ereignisbehandlung abgebrochen werden, falls in den vorherigen Phasen Fehler aufgetreten sind.

JSF steuert auch die Navigation zwischen verschiedenen Webseiten. Diese wird nicht statisch auf der jeweiligen Seite bestimmt, sondern durch eine Konfigurationsdatei festgelegt. Für verschiedene Ergebnisse von Aktionen kann auf verschiedene Seiten weiter verwiesen werden.

2.6.7 Web Services

Anwendungen in der Bioinformatik sind oft in unterschiedlichen Programmiersprachen geschrieben und laufen auf unterschiedlichen, oft verteilten Rechnersystemen. Eine lokale Installation ist oft nicht möglich oder wünschenswert. Deshalb wird ein einheitliches Verfahren benötigt, um Informationen zwischen Programmen austauschen zu können. Hier haben sich Web Services etabliert. Web Services sind eine Sammlung von Protokollen und Standards, die Interaktionen zwischen verschiedenen Maschinen über ein Netzwerk ermöglichen. Web-Services in der Bioinformatik ermöglichen meist automatische Datenabfragen in Datenbanken oder die Durchführung von Berechnungen auf entfernten Rechnern. Bei Web Services handelt es sich im Wesentlichen um ein Client-Server-System, das zur Kommunikation XML-Nachrichten im SOAP-Format verwendet. Die Operationen, die vom Server durchgeführt werden können, werden in der XML-basierten Web Service Description Language (WSDL) beschreiben. Aus der WSDL-Beschreibung können mit Hilfsprogrammen für die meisten Programmiersprachen automatische Abfragemodule für Web Services generiert werden.

3 Resultate

GEPAT wurde entwickelt, um eine Analyse von Genexpressionsdaten zu ermöglichen, bei der Informationen über das biologische System der Zelle integriert zur Analyse und Interpretation verwendet werden sollen. Dafür enthält das System eine Reihe von Möglichkeiten:

- Import von Microarray Genexpressionsdaten
- Verschiedene Analysemethoden für Microarray-Daten
- Interpretationsmethoden für Datenteilmengen
- Informationen über spezifische Gene im Datensatz

Die Analyse- und Interpretationsschritte sind integriert und können beliebig miteinander verknüpft werden, wobei die Ergebnisse der Dateninterpretation als Eingabe für die Datenanalyse verwendet werden können und umgekehrt. GEPAT ist ein frei verfügbares [21] Open Source-Projekt unter der LGPL [20]. Eine Installation findet sich am Biozentrum der Universität Würzburg [125].

3.1 Server

Zur Implementierung einer integrierten, einfach zu bedienenden Microarray-Applikation wurden fortgeschrittene Internettechnologien verwendet. GEPAT wurde als Internet-anwendung entwickelt, die auf der Programmiersprache Java [126] und dem Framework der Java EE beruht. Nicht immer ist der Einsatz des kompletten Java EE-Frameworks nötig und sinnvoll [127]. Eine der wichtigsten Entscheidungen bei der Auswahl des Serversystems liegt darin, ob Enterprise Java Beans (siehe Kapitel 2.6.3) verwendet werden sollen. EJB werden zwar häufig als Kern der Java EE bezeichnet, stellen jedoch nur ein der Möglichkeiten dar, die Java EE bietet. Gründe für den Einsatz von EJB sind:

- Entfernter Zugriff auf Applikationskomponenten. Sollen Daten mit anderen Komponenten über entfernten Prozeduraufruf (Remote Method Invocation, RMI) ausgetauscht werden, ist die Verwendung von EJB sinnvoll. Erfolgt kein entfernter Zugriff, oder nur im Rahmen von Web Services, kann auf EJB verzichtet werden.
- Verteilung von Applikationskomponenten auf mehrere Server. EJB bieten Unterstützung für verteilte Komponenten und sollte hier der Verwendung von

Webservices zur internen Verteilung vorgezogen werden.

- Unterstützung verschiedener Clienttypen. Sollen verschiedene Arten von Java-Clients mit graphischer Benutzeroberfläche für den Server verwendet werden, stellen EJB eine gute Lösung dar, da sie den Zugriff auf externe Applikationskomponenten über RMI unterstützen. In diesen Applikationen findet sich meist keine Web-Schicht, so dass EJB die notwendige Mittelschicht liefern können.
- Empfang von asynchronen Nachrichten. Message-driven Beans, eine Ausprägung von EJB, eignen sich gut, um einfach auf Nachrichten des Java Messaging Service (JMS) reagieren zu können.

Stellen sich diese Anforderungen nicht, ist auch die Verwendung von EJB nicht notwendig, da sie zu einer unnötigen Erhöhung der Komplexität führen. Für einfachere Projekte ist es meist ausreichend, nur einen einfachen Webcontainer einzusetzen und auf einen vollständigen Java EE Applikationsserver zu verzichten. Der verwendete Webcontainer kann evtl. durch zusätzliche Frameworks um Funktionalität erweitert werden. GEPAT wurde ohne die Verwendung von EJB implementiert, da keine Methodenaufrufe auf entfernten Rechnern durchgeführt werden. Als Webcontainer wird der Apache Tomcat [128] Server in der Version 5.5 verwendet. Da dieser nicht den vollen Umfang der Java EE zur Verfügung stellt, werden die benötigten Module als Bibliotheken hinzugefügt. JavaServer Faces Technologie wird für die Erzeugung von Webseiten verwendet. Diese bietet einen Model-View-Controller Ansatz für Internet-Applikationen und erlaubt so eine Applikationsentwicklung ähnlich der von Desktop-Applikationen. Die Erzeugung von Graphiken erfolgt über Java Servlets, Zugriffskontrolle für die HTML-Seiten erfolgt über Java Servlet Filter.

Alle Datenbanken und Analysemethoden, die von GEPAT verwendet werden, sind als unabhängige Module implementiert. Das Programm selbst bietet nur ein Grundgerüst mit Funktionen zum User-Management und zur Datenhaltung, jegliche weitere Funktionalität wird durch die Module ergänzt. Dadurch ist eine einfache Erweiterung zur Integration weiterer Datenbanken oder Analysemethoden möglich. Module werden für den Import von Genexpressionsdaten, der Selektion von Teilmengen der Daten, für Analyse und Interpretation und zur Geninformation verwendet. Die derzeit implementierten Module

zur Datenanalyse erlauben es, die Berechnungen entweder auf dem Server oder auf einem Computer-Grid-System mit einer DRMAA [129] kompatiblen Schnittstelle auszuführen. An der Universität Würzburg wird eine 10-Knoten Rechencluster verwendet, der auf der Sun Grid Engine [130] basiert. Zur Datenanalyse werden die Funktionen des Bioconductor-Toolkits mit einer einfach zu bedienenden Benutzerschnittstelle kombiniert. Für die Layout- und Visualisierungsfunktionen von Graphen wird die JUNG-Bibliothek [131] verwendet. Tabelle 1 zeigt eine Übersicht über die verwendeten Bibliotheken.

Tabelle 1: In GEPAT verwendete Bibliotheken

Art	Name	Version	Beschreibung
Java EE [117]	**jsf-api** **jsf-impl** **jstl** **standard** **mail** **activation**	1.1.1 1.1.01 1.1.0 1.1.0 1.3.1 1.0.2	Java ServerFaces [122] API JSF Implementierung JavaServer Pages Standard Tag Library [132] - API JSTL Implementierung [133] JavaMail [134] JavaBeans Activation Framework [135]
Jakarta Commons [136]	**commons-beanutils** **commons-collections** **commons-dbcp** **commons-digester** **commons-fileupload** **commons-io** **commons-logging** **commons-pool**	1.1 3.1 1.2.1 1.7 1.1.1 1.1 1.0.4 1.2	Wrappers um Reflection und Introspection APIs Erweiterung des Java Collections Framework Database connection pooling Service XML-nach-Java-Objektmapping Hilfsmittel File Upload Hilfsmittel für Servlets Sammlung von I/O Hilfsmitteln Wrapper um Logging API Implementierung Generische Object Pooling Komponente
JDBC [137]	**mysql-connector**	3.1.8	MySQL Connector/J [138]
	Log4j **junit**	1.2.8 3.8.1	Logging Framework [139] JRegression testing framework [140]
	Jung **colt**	1.7.2 1.2.0	Java Universal Network/Graph Framework [131] Cern Colt Scientific Library [141]
DRMAA [129]	**drmaa.jar**		Sun Grid Engine [130] DRMAA Java API
Ensembl	**ensj**	39.2	JAVA Api für Ensembl [142]

Zur statistischen Auswertung der Microarraydaten wird in GEPAT auf bestehende Systeme zurückgegriffen. GEPAT verwendet hierzu R [4], eine Programmiersprache und Softwareumgebung für statistische Berechnung und Grafiken, die eine Implementierung der Programmiersprache S darstellt. R wird häufig zur Entwicklung statistischer Software und zur Datenanalyse verwendet und ist die de-facto Standardsprache im Bereich der Statistik. R bietet eine Kommandozeile als Interface, graphische Bentuzeroberflächen sind für einige Funktionen verfügbar. R kann einfach durch Bibliotheken, Packages genannt, erweitert werden. Eine Stärke von R liegt in seinen Fähigkeiten, hochqualitative Graphen erzeugen zu können. Für die Microarray-Datenanalyse wurde das Paket

Bioconductor [3] geschaffen, das aus verschiedenen Erweiterungsbibliotheken zur Sprache R besteht. GEPAT verwendet Programmroutinen aus dem Bioconductor-Paket zur Datenanalyse.

Eine direkte Verbindung zwischen R und Java ist durch das rJava-Projekt möglich [143]. Es bildet über das Java Native Interface (JNI) [144] eine Brücke zwischen R und Java und erlaubt den Aufruf von Java aus R und umgekehrt den Aufruf von R aus Java heraus. Obwohl es eine einfache Methode zum Zusammenspiel der beiden Programmiersprachen darstellt, wurde in GEPAT ein anderer Ansatz gewählt, da mit dem JNI eine verteilte Ausführung der R-Programme auf verschiedenen Rechnern nicht möglich ist. Deshalb ruft GEPAT R über die Kommandozeile auf und übergibt die Programme und deren Parameter durch Eingabe auf der Konsole. Dieser Aufruf lässt sich auch über eine DRMAA-kompatible Gridengine durchführen.

Der Aufruf über die Kommandozeile, sowie Eingabeparameter und Rückgabewerte des Programms werden in GEPAT in Objekte gekapselt. Zwei verschiedene Ausführungsklassen ermöglichen es, die R Programme sowohl direkt auf dem Webserver, als auch auf dem Cluster auszuführen.

3.2 Benutzeroberfläche

Die Oberfläche ist der Teil des Programms, mit dem sich der Benutzer auseinandersetzen muss. Deshalb ist ein benutzerfreundliches Design dieser Schnittstelle eine absolute Notwendigkeit. Aus den Erfahrungen im Design von Benutzerschnittstellen wurden verschiedene Regeln abgeleitet, am bekanntesten sind die acht Regeln zum Schnittstellendesign von Shneiderman [145] :

1. Konsistenz – In ähnlichen Situationen sollen konsistente Folgen von Aktionen verwendet werden. Identische Begriffe sollen in Eingabeaufforderungen, Menüs und Hilfe verwendet werden. Die Verwendung von Farben, Layout und Schriftarten soll gleich eingesetzt werden, Ausnahmen sollen nur im geringen Umfang vorhanden sein.
2. Shortcuts – Je häufiger Benutzer eine Applikation verwenden, desto weniger Aktionen möchten sie zur Erreichung eines Ziels durchführen, um die Arbeitsgeschwindigkeit erhöhen zu können. Abkürzungen, Spezialtasten, Makros und Menu-Shortcuts sind Möglichkeiten, dies zu erreichen

3. Rückmeldungen – Für jede Benutzeraktion soll das System ein Feedback liefern. Für häufige Aktionen kann dieser Rückmeldung geringer ausfallen als für seltene, aufwendige Aktionen.
4. Abgeschlossenheit – Sequenzen von Aktionen sollen als Gruppe organisiert sein, die in Anfang, Verlauf und Ende eingeteilt sind. Informationen über den Abschluss einer Gruppe von Aktionen vermitteln dem Benutzer das Gefühl, dass eine Aufgabe erfüllt ist und er sich anderen Dingen widmen kann.
5. Fehlervermeidung – Das Design eines Systems soll soweit wie möglich so ausgelegt sein, dass der Benutzer keine Fehler machen kann. So ist zum Beispiel in einem Formular die Auswahl aus einem Menü dem Eintrag von Text vorzuziehen. Wenn es doch zu einem Fehler kommt, muss dieser vom System erkannt werden und konstruktive Lösungsvorschläge sollen angeboten werden. Bei Fehleingaben in einem Formular soll es nicht notwendig sein, die richtigen Formulareingaben auch nochmals angeben zu müssen. Fehlerhafte Aktionen sollen den Zustand eines Systems unverändert lassen, oder eine Rückkehr in den alten Zustand ermöglichen.
6. Zurücknahme – Aktionen sollen so häufig wie möglich zurücknehmbar sein. Dies ermutigt den Benutzer zum explorativen Durchführen von bisher unbekannten Aktionen, da er weiß, dass er auftretende Fehler zurücknehmen kann.
7. Kontrollübergabe – Erfahrene Benutzer benötigen das Gefühl, die Kontrolle über das System zu besitzen. Überraschende Systemreaktionen, langwierige Sequenzen von Dateneingabe und Schwierigkeiten, bestimmte Informationen abzurufen, führen hier zu Unzufriedenheit.
8. Kurzzeitgedächtnis – Die menschliche Informationsbearbeitung im Kurzzeitgedächtnis ist auf sieben +/- zwei Informationen begrenzt. Dies fordert einfache Anzeigen, den Verzicht auf mehrseitige Eingaben und eine geringe Anzahl neuer Fenster. Falls möglich, sollten Aktionen und Informationen im Programm erklärt werden.

GEPAT wurde als Web-Applikation entwickelt. Diese Applikation läuft auf einem Webserver, der Benutzer greift mit seinem Internet-Browser auf die Applikation zu (siehe Kapitel 2.6.4ff). Die Vorteile liegen darin, dass

- eine Installation auf dem lokalen Rechner nicht notwendig ist. Weder Datenbanksysteme noch eine Client-Software müssen installiert werden
- ein Großteil der Programme in der Bioinformatik nur als Internetanwendungen zur Verfügung steht. Damit ist es der Benutzer bereits gewohnt, mit diesen Anwendungen umzugehen.
- die Programmversion immer auf dem aktuellen Stand ist. Updates auf neue Versionen müssen nicht vom Benutzer eingespielt werden. Vielmehr genügt es, die Version auf dem Server zu aktualisieren
- ein Zugriff auf Programm und Daten nicht an einen Rechner gebunden ist. Benutzer können verschiedene Rechner verwenden, um mit den Daten zu arbeiten.
- Der Client-Rechner muss nicht auf die von GEPAT verwendeten Datenbanken zugreifen können, da die Datenbankabfrage über den Webserver erfolgt.

Nachteile der Internetapplikation liegen

- im eingeschränkten Benutzerinterface. Im Vergleich zu herkömmlichen graphischen Benutzeroberflächen erlaubt die HTML-Darstellung nur eine eingeschränkte Interaktivität. Durch Verwendung von CSS Elementen, kombiniert mit JavaScript, ist jedoch eine Erweiterung des Interfaces um interaktiven Elemente, z.B. mit einer Menüleiste oder Tooltips, möglich
- in eingeschränkter Interaktivität. HTML ist seitenorientiert und sieht nur eine Reaktion auf Aktionen des Benutzers vor. Bei Klick auf einen Link oder Formular-Knopf sendet der Internetbrowser eine Nachricht an den Server, der von diesem mit einer neuen HTML-Seite beantwortet wird. Durch die Verwendung von AJAX in JavaScript ist es jedoch möglich, asynchron XML-Dateien vom Server abzurufen und diese in das Objektmodell der Webseite (DOM) zu integrieren. Damit sind prinzipiell interaktive Webseiten möglich, jedoch auf Kosten einer gesteigerten Komplexität der Webseite und Serveranwendung.
- In der Akzeptanz des Benutzers. Viele Anwender sind unzufrieden mit der Tatsache, dass ihre Daten physisch außerhalb ihres Zugriffsbereichs liegen und Übertragungen über das Internet unsicher sein können. Abhilfe bietet hier eine

lokale Installation des GEPAT-Servers innerhalb eines beschränkten Netzwerks.

3.3 Datenhaltung

3.3.1 Expressionsdaten

Alle Benutzerdaten werden auf dem Server im passwortgeschützten Bereich gespeichert. Damit wird verhindert, dass der Benutzer für jede Berechnung erneut Daten auf den Webserver laden muss, und der Zugriff auf die Daten von jedem beliebigen Rechner aus wird möglich.

Zur Verteilung von Rechenjobs auf den Rechencluster ist es notwendig, dass sowohl auf dem Webserver als auch auf dem Cluster Zugriff auf die Dateien möglich ist. Weiterhin soll der Zugriff möglichst schnell erfolgen können. Deshalb wurde auf dem Webserver ein NFS-Server installiert, über den die Clusternodes auf die Benutzerdaten zugreifen können. Damit ist ein schneller Zugriff auf die Daten des Servers gewährleistet. Hierbei muss beachtet werden, dass die Clusternodes die Ergebnisse der Berechnung auf den NFS-Share schreiben und deshalb dort die entsprechenden Schreibrechte besitzen müssen.

Durch die Freigabe wurde die Möglichkeit geschaffen, sowohl vom WebServer als auch von den Clusternodes auf die Daten des Benutzers zugreifen zu können. Hierfür werden jedoch die Daten in einem Format benötigt, das sowohl von GEPAT als auch von R gelesen werden kann. Als einfachstes Format bietet sich hier die Speicherung der Expressionsmatrixwerte als Tabulator-getrennte Tabelle von Fließkommazahlen an. Allerdings zeigt sich rasch, dass diese Art der Speicherung sowohl in Java als auch unter R zu einem langwierigen Umwandeln der Daten in das jeweils intern verwendete Format führt. Deshalb wurde ein anderer Weg gewählt. Nach Import der Datenmatrix wird diese als Tabulator-getrennte Tabelle abgespeichert, anschließend wird diese Tabelle sowohl von R als auch von Java eingelesen und jeweils in ein internes Format gewandelt, das einen bedeutend schnelleren Zugriff ermöglicht.

Alle Benutzerdaten im Hauptspeicher zu halten gewährt schnellen Zugriff, ist jedoch aufgrund der Größe der Microarray-Daten nicht möglich, da dann auf dem Server nur Hauptspeicher für wenige Benutzer gleichzeitig auf zur Verfügung steht. Das Abspeichern der Daten auf der Festplatte benötigt wenig Hauptspeicher und erlaubt viele Benutzer

gleichzeitig auf dem Server. Wenn jedoch ein bestimmter Expressionswert benötigt wird, müssen die Daten von Festplatte abgerufen werden, was Zeit benötigt. Dies kann sich bei einer Vielzahl von Messwerten zu einer längeren Wartezeit für den Benutzer addieren. Um schnellen Zugriff für eine große Anzahl an Benutzern zu ermöglichen, wurde ein Zwischenweg zwischen diesen beiden Techniken gewählt. Nur ein Teil der Benutzerdaten verbleibt gleichzeitig im Hauptspeicher. Dazu werden die Expressionsmatrizen in Teilmatrizen zu je 1000 Zeilen eingeteilt und die Teilmatrizen werden bei Bedarf blockweise in den Hauptspeicher geladen. Ist kein weiterer Platz im Hauptspeicher verfügbar, werden die ältesten Blöcke aus dem Hauptspeicher entfernt und durch den neuesten zu ladenden Block ersetzt. Wird der ersetzte Block nochmals benötigt, wird er wieder von Platte geladen. Wenn nur einzelne Datenpunkte aus einer Teilmatrix benötigt werden, kann so ein schneller Zugriff erfolgen, ohne den ganzen Datensatz im Speicher zu behalten. Ist genügend Speicher vorhanden, kann direkt auf die Daten zugegriffen werden, ansonsten werden diese nachgeladen.

3.3.2 Datenbanken

Der modulare Ansatz in GEPAT erlaubt die Verwendung beliebiger Datenbanken in neuen Modulen, die bereits existierenden Module unterstützen eine Reihe biologischer Datenbanken. Die Ensembl-Datenbank wurde direkt übernommen, das Format der meisten anderen integrierten Datenbanken war jedoch nicht zur direkten Verwendung geeignet, so dass diese Datenbanken in eine neue Struktur gebracht wurden. Zum Speichern der Datenbanken wird ein mySQL-Datenbanksystem [146] verwendet. Skripte zum Erstellen der Datenbank-Tabellen und zum Umwandeln bestehender Datenbanken werden von GEPAT bereitgestellt.

3.3.3 Benutzerverwaltung

Da bei einer Webapplikation die Benutzerdaten auf einem Server gelagert werden, auf dem mehrere Benutzer gleichzeitig arbeiten, ist es wichtig, Benutzer eindeutig identifizieren und die Daten den entsprechenden Besitzern zuordnen zu können. Dazu wurde eine Datenbank angelegt, welche die Informationen zu jedem Benutzer speichert.

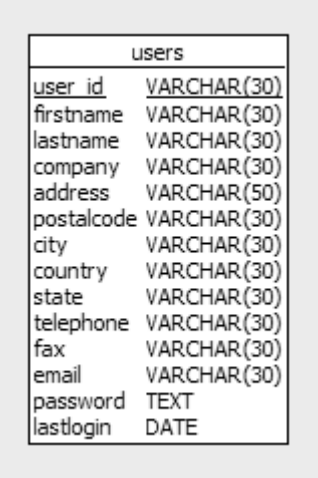


Abbildung 19: Tabelle zur Benutzerdatenspeicherung
Jedem Benutzer wird eine eindeutige ID zugewiesen, die als Primärschlüssel dient. Vorname, Nachname und Emailadresse müssen angegeben werden. Neben dem Passwort als MD5-Hashwert wird auch der Zeitpunkt des letzten Logins gespeichert.

Zur Verwendung von GEPAT ist eine Registrierung notwendig, um einen eindeutigen Zugang mit einer Benutzername-Passwort-Kombination zu erhalten. Bei der Erstellung des Benutzerkontos wird automatisch ein zufällig generiertes Passwort vergeben, das dem Benutzer über Email mitgeteilt wird. So ist sicher gestellt, dass nur Benutzer mit gültiger Email-Adresse GEPAT benutzen können. Bei der Erstellung eines Benutzeraccounts sind nur die Eingabe von Benutzer-ID, Vorname, Nachname und Email-Adresse verpflichtend. Alle weiteren Angaben sind freiwillig. Das Passwort wird aus Sicherheitsgründen nicht direkt in der Datenbank gespeichert. Vielmehr wird der MD5-Hashwert des Passworts berechnet und als Hexadezimalwert in der Datenbank gespeichert. Damit ist ein Auslesen des Passworts aus der Datenbank nicht mehr möglich. Als zusätzliche Information wird der Zeitpunkt des letzten Logins eines Benutzers gespeichert, um Informationen über die Aktivitäten der Benutzer zu erhalten. Zusätzlich werden diese Daten verwendet, um eine unerlaubte Anmeldung durch Benutzung des Zurück-Knopfes des Webbrowsers zu verhindern. Abbildung 19 zeigt die Struktur der Tabelle zur Benutzerverwaltung.

3.4 Module

Der modulare Aufbau von GEPAT erlaubt die beliebige Erweiterung mit neuen Modulen. Ohne modulare Erweiterung ist das Programm nur in der Lage, eine Tabelle mit

Expressiondaten einzuladen und die Genexpressionsmatrix anzuzeigen. Abbildung 20 zeigt diese Darstellung, Abbildung 21 zeigt die Möglichkeiten zur Selektion von Teilmengen.

Abbildung 20: GEPAT ohne Module
Zu sehen ist die Übersichtsseite, die nur die Namen der Gensonden sowie die Genexpressionsmatrix farblich kodiert anzeigt.

Abbildung 21: Modulunabhängige Teilmengenauswahlmöglichkeiten in GEPAT
a) erlaubt es, alle, keine, oder nur die als Arbeitsmenge verwendeten Transkripte zu selektieren, b) erlaubt es, vordefinierte Gruppen auszuwählen, c) ermöglicht die Suche nach Transkripten. Die Möglichkeiten a) und b) sind analog auch für Proben verfügbar, d) erlaubt die Suche nach Proben. Die zusätzlichen Reiter des Auswahldialogs werden durch die entsprechenden Module bereitgestellt.

Hierbei bilden die Zeilenvektoren die Expression für ein Transkript, die Spaltenvektoren die Expression für eine Probe ab. Der Wert der Expression wird durch eine Farbe kodiert, jede Farbschattierung entspricht dabei einem Expressionswert. Die Expressionswerte werden beim Datenimport berechnet, bei Zweikanal-Microarraydaten entsprechen sie normalerweise dem Logarithmus des Verhältnisses des roten zum grünen Kanals, bei Einkanal-Microarraydaten normalerweise den absoluten Messwerten, die bei der Normalisierung logarithmisch transformiert werden. Die Werte der Genexpressionsmatrix können zusätzlich noch zentriert und skaliert werden. Dies kann über Einstellungen der Dateneigenschaften im Daten-Menü erfolgen.

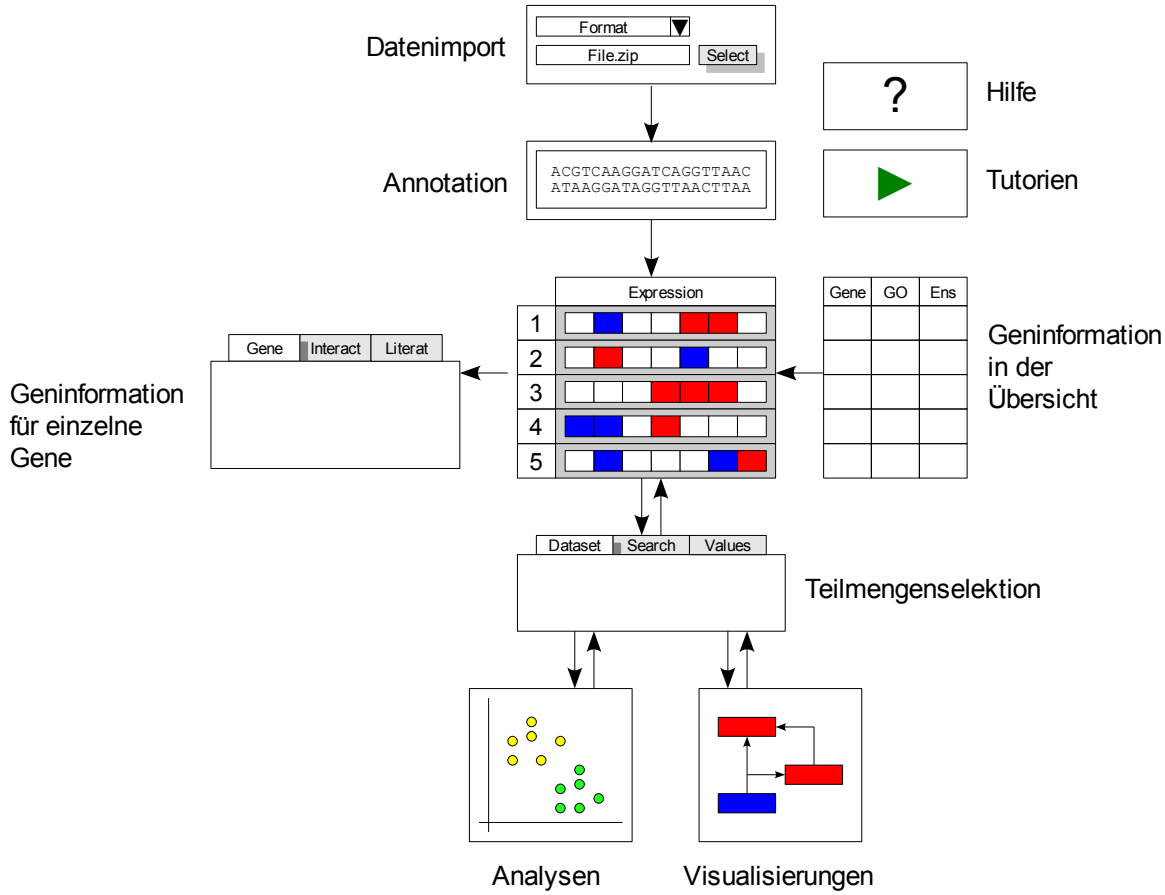


Abbildung 22: Übersicht über die verschiedenen Module
Neben Datenimport und Analyse können Geninformationen sowohl auf der Übersichtsseite als auch auf der Geninformationsseite angezeigt werden. Auch Teilmengenselektion, Analysen und Visualisierungen können modular erweitert werden. Jedes Modul kann auch Hilfeseiten und Tutorien für den Benutzer zur Verfügung stellen.

Eine Reihe von Modulen wurde implementiert, um den Funktionsumfang von GEPAT zu vergrößern. Welche Module verwendet werden, und welche Funktionalität ein Modul bereitstellt, wird in der Datei **`modules.xml`** definiert. Zu einem Modul gehört immer auch

ein Java-Bean, das die Modulfunktionalität bereitstellt. Der modulare Aufbau wird in Abbildung 22 skizziert. Module existieren für

- Datenimport. Diese Module behandeln den Import von Microarraydaten vom Upload bis zur fertigen Genexpressionsmatrix
- Annotation. Die Annotationsmodule sind für die Zuordnung von Informationen zu den Transkripten auf dem Microarray zuständig.
- Geninformation in der Übersicht: Diese Module stellen Geninformationen in der Übersichtstabelle bereit.
- Geninformation für einzelne Gene: Zu jedem einzelnen Transkript des Microarrays sind können weitere Informationen über diese Module angezeigt werden.
- Teilmengenselektion: Diese Module erlauben es, aus der Menge der Transkripte und Proben Teilmengen auszuwählen, auf die sich weitere Analysen, Interpretationen und Visualisierungen beziehen.
- Analysen: Mögliche Datenanalysen- und Interpretationsmethoden werden durch diese Module implementiert.
- Visualisierungen: Die Ergebnisse der Datenauswertung können mithilfe dieser Module visualisiert werden
- Hilfe: Diese Module ermöglichen Online-Hilfefunktionen
- Tutorien: Einführungen für neue Benutzer werden durch diese Module bereitgestellt.

3.4.1 Teilmengenkonzept

Microarray-Experimente erzeugen eine große Anzahl an Messwerten auf einmal. Auf jedem Array befindet sich viele zehntausende Messpunkte, meist werden über 100 Arrays miteinander verglichen. In den meisten Fällen will man nicht mit allen Messwerten gleichzeitig, sondern nur mit einer Teilmenge davon, arbeiten. Differentielle Genexpression besteht z.B. nur in einem Teil der Transkripte, und zur weiteren Analyse kann es sinnvoll sein, nur mit den differentiell exprimierten Transkripten weiterzuarbeiten. Auch die Proben können sich in mehreren Gesichtspunkten unterscheiden, z.B. durch unterschiedliche Gewebe oder Krankheitstypen. Eine wichtige

Funktionalität von GEPAT besteht deshalb darin, die Expressionsmatrix auf eine Teilmatrix zu beschränken und Analysen nur noch auf diese Teilmatrix zu beziehen. Damit ist es möglich, Multiple-Testing-Probleme zu reduzieren, fehlerhafte Messwerte nicht zu berücksichtigen, oder nur mit bestimmten Probenarten zu arbeiten. Die Expressionsmatrix kann sowohl in den Zeilen, die der Expression für ein Transkript über die Proben hinweg entsprechen, als auch in den Spalten, die der Expression einer Probe über verschiedene Transkripte entsprechen, eingeschränkt werden. Analysen können dann auf diese Teilmenge eingeschränkt werden, indem eine oder mehrere dieser Teilmengen als Eingabe verwendet werden. Zur Visualisierung kann eine Arbeitsmenge definiert werden und alle Ausgaben werden dann für diese Arbeitsmenge berechnet.

Verschiedene Charakteristiken stehen zur Einschränkung bereit, die von einfachen Bedingungen wie Transkript- oder Probenamen bis hin zu komplexen Analyseergebnissen reichen. Zum schnellen Zugriff können Teilmengen von Transkripten oder Proben zu Gruppen zusammengefasst werden, die dann mit einem Namen versehen werden können. So können bei klinischen Analysen alle Proben, die zu einem bestimmten Krankheitsbild gehören, in einer Gruppe mit dem entsprechenden Namen zusammengefasst werden. Als Quelle für eine Teilmenge können entweder der gesamte Datensatz oder vorher definierte Gruppen verwendet werden. Auch die Arbeitsmenge kann als Ausgangspunkt für ein neues Subset verwendet werden. Die Selektion von Teilmengen für Transkripte und Proben ist modular, neue Kriterien können einfach hinzugefügt werden. Eine Übersicht über die aktuell möglichen Kriterien gibt Tabelle 2.

An der Auswahl einer Teilmenge sind zwei verschiedene Komponenten beteiligt, eine Komponente zur Auswahl von Teilmengen und eine Komponenten, die Teilmengen als Eingabeparameter besitzt. Die Komponenten zum Setzen von Teilmengen besitzen eine eindeutige ID, die zur Teilmengenselektion verwendet wird. Soll in der Eingabemaske eine Teilmenge als Eingabeparameter verwendet werden, so wird eine JavaScript-Funktion mit der ID des Ziels aufgerufen. Die JavaScript-Funktion öffnet ein weiteres Fenster, das die Komponenten zur Auswahl von Teilmengen anzeigt. Wird eine Teilmenge ausgewählt, so wird diese in der durch die ID spezifizierte Empfangskomponente über die angegebene JavaBean-Methode gesetzt. Zusätzlich wird noch eine für den Benutzer verständliche Nachricht gespeichert, welche die Herkunft der Teilmenge beschreibt. Das Verfahren wird in Abbildung 23 illustriert.

Tabelle 2: Mögliche Kriterien zur Teilmengenauswahl

Transkripte	Proben
Suche – Erlaubt die Suche nach Sondennamen, Gennamen oder Ensembl-Kennung **Gruppe** – Zugehörigkeit zu einer vordefinierten Subgruppe der Daten **GO Kategorie** – Transkripte, die zu einer bestimmten GO-Kategorie gehören. **k-means-Cluster Analyse**- – Erlaubt Teilmengenauswahl auf den Ergebnissen einer k-Means Clusteranalyse **t-Test Analyse** – Erlaubt Teilmengenauswahl auf den Ergebnissen eines t-Tests **KEGG Karten** – Transkripte, die auf einer bestimmten KEGG Stoffwechselkarte vorkommen **Chromosomposition** – Transkripte, die in einem bestimmten chromosomalen Bereich vorkommen.	**Suche** – Suche nach Probenname **Gruppe** – Zugehörigkeit zu einer vordefinierten Subgruppe der Daten **k-means-Cluster Analyse**- – Erlaubt Teilmengenauswahl auf den Ergebnissen einer k-Means Clusteranalyse **Hauptkomponentenanalyse** – Erlaubt Teilmengenauswahl auf den Ergebnissen einer PCA.

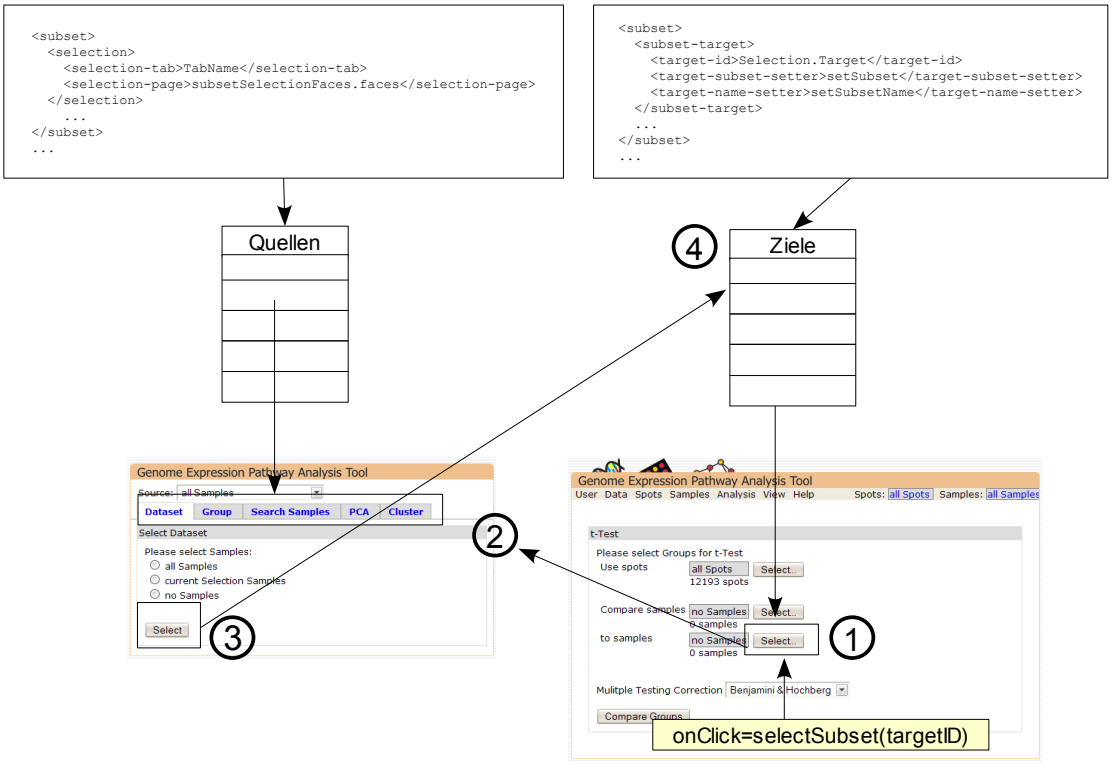


Abbildung 23: Verfahren zur Teilmengenselektion
Module können sich als Quelle oder Ziel von Teilmengen registrieren. Auf einer Webseite eines Zielmoduls (1) kann dann die Dialogbox zur Teilmengenauswahl aufgerufen werden (2). Die Quellen zur Teilmengenselektion werden angezeigt. Nach der Auswahl (3) wird die gewählte Teilmenge dem Zielmodul übergeben (4)

3.4.2 Datenimport

Ein funktionierender Datenimport, der häufig benötigte Dateiformate unterstützt, ist die Voraussetzung für jegliche Art von Analyse. Die Modularisierung des Datenimports ermöglicht es, weitere Formate zu ergänzen. Importer für Microarrays werden in der Datei **`modules.xml`** spezifiziert. Microarray-Importer müssen ein Interface implementieren, das Methoden bereitstellt, die für den Standardimportprozess benötigt werden. Das Interface bestimmt auch Methoden, über die mögliche Dateiformate, die der jeweiligen Importer unterstützt, angegeben werden können.

Zusätzlich zum Import von einzelnen Microarray-Dateien bietet GEPAT noch die Möglichkeit, Tabulator-getrennte Tabellen einzuladen, welche die Genexpressionsmatrix direkt beschreiben. Hier können entweder Absolutwerte, logarithmisch skalierte Verhältniswerte, oder die Werte beider Kanäle angegeben werden. Da bei dieser Art des Datenimports wesentliche Informationen zur Normalisierung, wie Hintergrundinformation oder Arraylayout, fehlen, sollte der Import, falls möglich, über die Microarray-Dateien erfolgen

3.4.3 Annotation

Mit der Durchführung des Import-Vorgangs wird die Genexpressionsmatrix erstellt. Für die weitere Analyse und zur Interpretation der Daten ist es jedoch notwendig, die Bedeutung der Transkripte, der Zeilenvektoren der Genexpressionsmatrix, zu kennen. Deshalb ist eine Annotation der Daten notwendig.

Die Information über die cDNA-Sequenzen, die sich auf dem Microarray befindet, lassen sich meist aus den Array-Files ermitteln, bei einigen Datenformaten ist eine weitere GAL-Datei notwendig, die das Array-Layout und die verwendeten Gensonden beschreibt. Aus den Informationen über diese cDNA-Sequenzen wird die zugehörige Ensembl-Genkennung ermittelt. Diese Ensembl- Genkennung kann von den Modulen von GEPAT verwendet werden, um weitere Informationen zu den Daten zur Verfügung zu stellen. Um Informationen nicht ständig aufs Neue abrufen zu müssen, kann jedes Modul im Annotationsprozess diese Informationen ermitteln und leicht zugreifbar abspeichern. Dazu kann von jedem Modul eine Klasse angegeben werden, welche die Annotation durchführt. Durch Implementierung einer Schnittstelle und einen Eintrag in die Datei **`modules.xml`** wird diese Klasse dann nach dem Importprozess zur Annotation

aufgerufen.

3.4.4 Analyse, Interpretation und Visualisierung.

Während andere Programme Analyse und Interpretation als zwei unterschiedliche Verfahren behandeln und meist nur einen Teil davon unterstützen, nimmt GEPAT keine solche Einteilung vor. Vielmehr werden alle Arten von Berechnungen, die ein Benutzer durchführen kann, unter dem Menüpunkt Analyse vereint. Die Einträge, die ein Modul in diesem Menü besitzt, werden über die Datei **`modules.xml`** bestimmt, ein Modul kann dabei mehrere Einträge besitzen. Diese Einträge bestimmen den Namen im Menü und die Internetseiten, auf die der Menüeintrag verweist.

Um Ergebnisse unabhängig von der durchgeführten Analyse darstellen zu können, wurde das Ansicht-Menu geschaffen. Hier können Verweise zu allgemeinen Datenansichten aufgenommen werden, die unabhängig von der verwendeten Analyse sind.

3.4.5 Geninformation

Da die Annotation bereits beim Import erfolgt, können für jedes genkodierende Transkript eine Reihe weiterer Informationen angezeigt werden. Zwei verschiedene Mechanismen wurden zur Bereitstellung von Geninformation implementiert:

- In der GEPAT-Übersichtsseite, welche die Expressionsmatrix, bzw. eine Teilmatrix davon, anzeigt, können zu jedem Transkript zusätzlich Spalten mit Information angezeigt werden.
- Zusätzlich kann zu jedem Transkript ein Fenster aufgerufen werden, das weitere Informationen zum entsprechenden Gen enthält.

Die Informationen auf der Übersichtsseite können verwendet werden, um einen schnellen Überblick über die Transkripte auf dem Microarray zu erhalten. Die erweiterten Transkript-Informationsseiten können detaillierte Informationen enthalten, deren Zusammenstellung unter Umständen auch mehr Zeit in Anspruch nehmen kann. Die Transkriptinformationen werden in einem neuen Fenster angezeigt, eine Leiste am oberen Rand des Fensters kann verwendet werden, um zwischen verschiedene Geninformationsansichten umzuschalten. Eine Übersichtstabelle für jedes Gen stellt die Informationen aus der Übersichtsseite auch für die einzelnen Informationsseiten

zusammen. Beide Mechanismen werden durch einen Eintrag in der Datei **`modules.xml`** für das entsprechende Modul registriert.

3.4.6 Help & Tutorial

Viele Aufgaben beinhalten die Eingabe von Parametern durch den Benutzer. Um die Bedeutung dieser Parameter schnell überblicken zu können, kann jedes Modul mehrere Hilfeseiten in der GEPAT-Onlinehilfe registrieren. Dabei kann eine frei wählbare Kategorie angegeben werden, unter der die Hilfeseite im Inhaltsverzeichnis der Onlinehilfe eingetragen werden soll. Von einer Internetseite in GEPAT aus kann die Hilfe mit dem als Parameter angegebenen Namen in einem extra Hilfefenster aufgerufen werden.

Um unerfahrenen Benutzer einen leichten Einstieg zu ermöglichen, ist sinnvoll, die Arbeitsweise von Programmen anhand von Beispielen zu verdeutlichen. Dafür wurde, analog zur Online-Hilfe, ein Tutorial geschaffen. Auch hier können GEPAT-Module einen Eintrag in einer frei wählbaren Kategorie anzeigen und eine Einführung anbieten.

3.5 Datenimport & Annotation

3.5.1 Dateneingabe

Der Datenimport ist ein wichtiger Schritt vor der Datenanalyse. Die meisten bestehenden Programme erwarten die Daten in einem bestimmten Format, meist als Tabellen, deren Einträge durch das Tabulator-Zeichen getrennt sind, oder unterstützen nur eine geringe Anzahl an anderen Eingabeformaten. Um eine möglichst große Anzahl an unterschiedlichen Formaten unterstützen zu können, wurde ein modulares System implementiert, das eine Erweiterung auf beliebige Arten von Dateiformaten ermöglicht. Alle Eingabedateien werden von einem entsprechenden Modul bearbeitet und werden nach dem Import in einem internen, eingabeunabhängigen und schnell zugreifbaren Format auf dem Server gespeichert.

Zurzeit sind drei Module für den Datenimport implementiert. Das erste Modul ermöglicht den Datenimport für Tabulator-getrennte Tabellendateien, die entweder normalisierte oder unnormalisierte Einzel- oder Doppelkanaldaten enthalten können. Die anderen zwei Module erlauben den Import von Affymetrix- und cDNA-Microarraydaten. Eine Übersicht über die unterstützten Formate gibt Tabelle 3. Die Affymetrix Arrays werden von der

Bioconductor-Funktion **`read.affybatch`** behandelt, für den Import von cDNA-Arrays wird die Funktion **`read.maimages`** verwendet.

Tabelle 3: Unterstützte Eingabeformate in GEPAT

Oligonukleotid-Microarrays Affymetrix CEL Files (Human)
cDNA-Microarrays Agilent Feature Extraction ArrayVision BlueFuse GenePix ImaGene QuantArray SPOT Stanford Microarray Database
Tabellen Unnormalisierte Zweikanal-Data Unnormalisierte Einkanal-Data Normalisierte Data

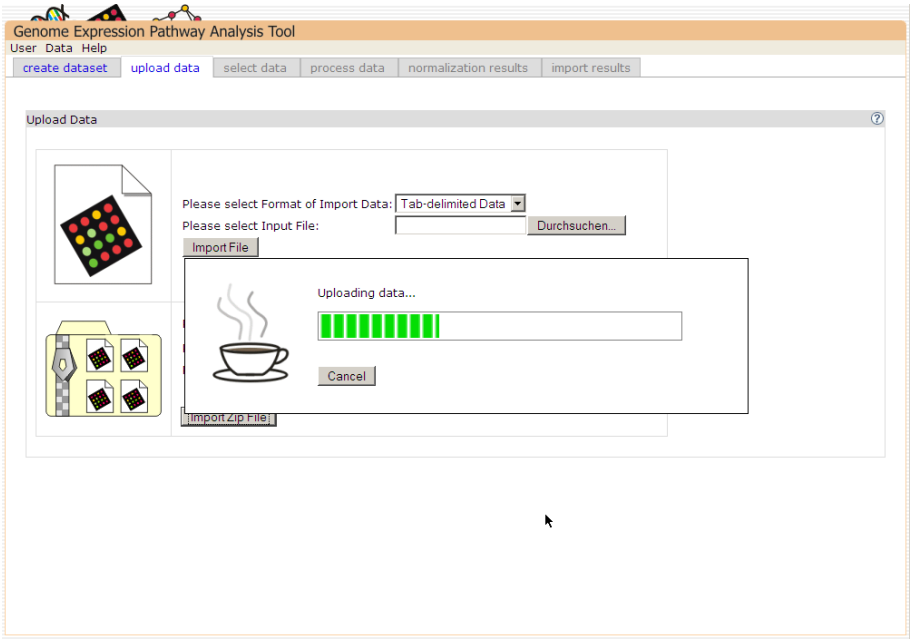


Abbildung 24: Datei-Upload
Da die übertragenen Datenmengen mehrere 100 MB betragen können, wird eine Fortschrittsanzeige eingeblendet.

Da sich die Dateien auf dem Rechner des Benutzers befinden, müssen diese vor der Verarbeitung auf den Server geladen werden. Um das Übertragungsvolumen gering zu halten und um mehrere Dateien auf einmal importieren zu können, akzeptiert GEPAT nur den Upload von Zip-gepackten Sammlungen von Microarray-Daten. Im Zip-File dürfen sich

nur Microarray-Datendateien befinden. Zusätzlich zu den Microarray-Files werden von einigen Import-Modulen noch GAL-Dateien [147] zur Array-Beschreibung benötigt. Diese Dateien können vom Benutzer zusätzlich ausgewählt werden. Vor dem Upload der Daten auf den Server muss vom Benutzer noch das Dateiformat der im Zip-File enthaltenen Dateien angegeben werden.

Die einzige Möglichkeit, ohne den Einsatz zusätzlicher Software beim Benutzer Dateien an den Server zu senden, liegt in der Verwendung des Datei-Upload Formulars von HTML. Da die in GEPAT verwendete Version 1.1 von JavaServer Faces jedoch keine Upload-Komponente unterstützt, wurde die notwendigen serverseitigen Komponenten ergänzt. Ein Problem in der Verwendung des Datei-Upload Formulars liegt darin, dass der Benutzer keine Rückmeldung über den Fortschritt des Vorgangs erhält. Besonders beim Upload von großen Dateien, die auch beim Import von Microarray-Experimenten entstehen, ist dies aufgrund der langen Upload-Zeit problematisch, da die verbleibende Zeit so nicht abschätzbar ist. Deshalb wurde Asynchronous HTML and HTTP [148], kurz AHAH verwendet, um ein dynamisches Update der Website mit dem Prozessfortschritt zu ermöglichen. AHAH verwendet, wie auch AJAX, die XMLHTTP-Funktionalität von JavaScript zum Ändern der Internetseiten, verwendet aber im Gegensatz zu AJAX die Antwort einer Anfrage direkt, ohne diese erst bearbeiten zu müssen. Die Funktionalität auf der Server-Seite wird durch eine erweiterte Upload-Komponente, die den Fortschritt überwacht und durch einer JSP-Seite bereitgestellt. Abbildung 24 zeigt den Datei-Upload.

Nach dem Upload werden Dichteplot der Arrays, Datencharakteristiken wie Mittelwert und Standardabweichung, und, falls in den Eingabedaten das Layout der Arrays enthalten ist, auch Bilder von den Arrays generiert, die wie in Abbildung 25 angezeigt werden. Dies ermöglicht es, die einzelnen Arrays nach Fehler zu untersuchen, die beim Hybridisierungsprozesses oder der anschließenden Behandlung entstanden sind und erlaubt es, verschwommene oder fehlerhafte Arrays von der weiteren Behandlung auszuschließen. Nach der Selektion der Microarrays müssen die Daten normalisiert werden.

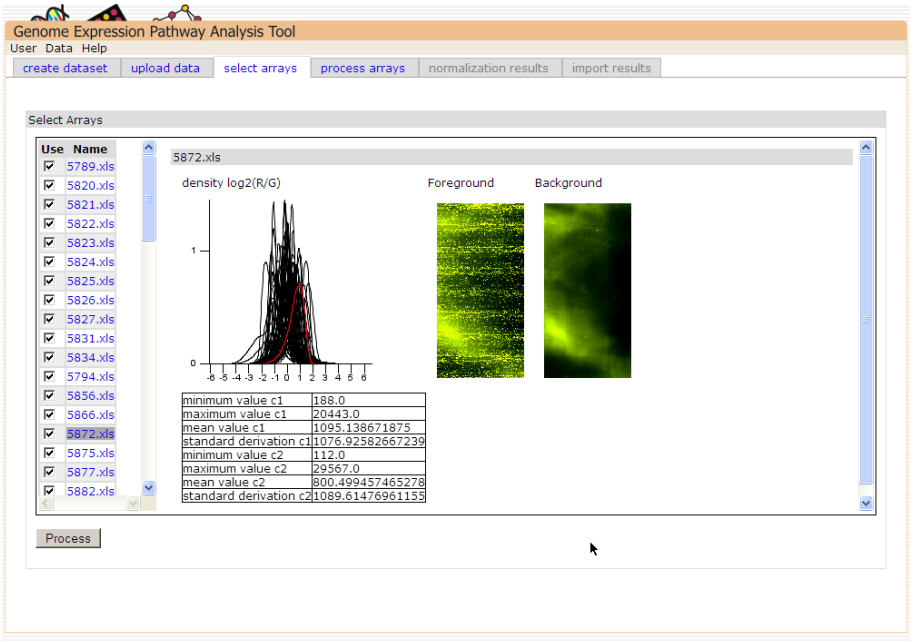


Abbildung 25: Übersicht über die Microarray-Daten
Neben Datencharakteristiken werden auch die Bilder von Vordergrund und Hintergrund des Arrays angezeigt, sofern das Arraylayout bekannt ist.

3.5.2 Fehlende Werte

Durch Fehler bei der Hybridisierung und während des Scanvorgangs kann es vorkommen, dass die Messwerte einzelner Gensonden nicht verfügbar sind. Da nur wenige Algorithmen existieren, die mit unbekannten Werten umgehen können, stellen diese fehlenden Werte ein Problem für die weitere Berechnung dar. Das Problem kann gelöst werden, indem Gensonden mit fehlenden Werten komplett aus der Genexpressionsmatrix entfernt werden, nur Algorithmen verwendet werden, die mit fehlenden Werten umgehen können, oder die fehlenden Werte aufgefüllt werden. GEPAT kann die Werte interpolieren und verwendet hierzu die Implementierung des knn-Imputationsverfahren durch das **`impute`** – Paket von R. Dabei werden Expressionswerte aus den k ähnlichsten Gensonden berechnet, wenn die Anzahl der fehlenden Werte einen Schwellwert von 50% nicht überschreitet. Fehlen mehr als 50% der Werte, wird der Mittelwert der vorhandenen Messwerte für die fehlenden Messwerte eingesetzt.

3.5.3 Normalisierung

Vor der Erstellung der Genexpressionsmatrix ist eine Normalisierung der Daten notwendig, um Störungen durch systembedingte Einflüsse auf die Daten zu entfernen

(siehe Kapitel 2.4.3). Da cDNA-Arrays die Genexpression zweier Proben vergleichen, Oligonukleotid-Arrays die Genexpression als absolute Werte messen, unterscheiden sich die möglichen Normalisierungsverfahren.

Bei den Normalisierungsverfahren ist die Kombination verschiedener Parameter nicht immer sinnvoll. So kann beim Import von Oligonukleotid-Arrays die RMA-Hintergrundkorrektur nicht mit der MAS-PM-Korrektur verwendet werden, da RMA die MM-Gensonden nicht korrigiert. Diese Korrektur ist aber für die Durchführung des MAS-Verfahrens notwendig. Ebenso sollte bei der Verwendung von VSN keine andere Behandlung der Daten erfolgen. Um hier die Fehlerquellen für unerfahrene Anwender gering zu halten, schränkt GEPAT die möglichen Auswahlmöglichkeiten entsprechend ein. Abbildung 26 zeigt die Eingabemaske, auf der die Parameter ausgewählt werden können. Nach erfolgter Normalisierung werden die Dichtekurven der Arrays vor und nach der Normalisierung gezeigt, um den Normalisierungseffekt bewerten zu können. Abbildung 27 zeigt das Ergebnis einer Normalisierung.

Affymetrix-Array

Im Affymetrix-System werden die Bild-Rohdaten in so genannten DAT-Files gespeichert. Die meisten Analyseprogramme benutzen jedoch nicht diese Rohdaten, sondern verwenden die so genannten CEL-Files, die durch Bearbeitung der DAT-Files entstehen, wobei die Sondenintensitäten aus den Rohdaten berechnet werden. Die Sonden-Information zu den CEL-Files wird durch so genannten CDF Files zur Verfügung gestellt. Diese geben an, welche Sonde zu welcher Gensondengruppe gehört, und ob es sich um eine PM oder MM Sonde handelt.

GEPAT verwendet zum Import von Affymetrix-Dateien die Bioconductor-Funktion **`expresso`**. Der Benutzer hat die Wahl zwischen 4 verschiedenen Normalisierungsverfahren, die in Tabelle 4 zusammengefasst werden.

Tabelle 4: Normalisierungsverfahren für Affymetrix-Microarraydaten

Name	*Hintergrund-korrektur*	*Normalisierung*	*PM-Korrektur*	*Zusammenfassung*
Quantile	RMA	Quantile	Pmonly	Medianpolish
Loess	RMA	Loess	Pmonly	Medianpolish
VSN	None	VSN	Pmonly	Medianpolish
MAS	MAS	Constant	MAS	MAS

Bei der quantile- und loess-Normalisierung wird auf die Verwendung der MM-Sonden verzichtet, weshalb die RMA-Hintergrundkorrektur zum Einsatz kommt. Dementsprechend wird auch keine PM-Korrektur vorgenommen. Bei der VSN-Normalisierung wird keine Hintergrundkorrektur vorgenommen, um unbehandelte Daten für die Normalisierung verwenden zu können. Das MAS-Verfahren dient zur Kompatibilität mit der Affymetrix Microarray Suite Software. Hier werden dieselben algorithmischen Verfahren angewandt, allerdings werden durch die unterschiedlichen Implementierungen nicht die gleichen Werte in beiden Anwendungen berechnet.

CDNA-Arrays

GEPAT verwendet das Bioconductor-Paket **`limma`** [90] zur Normalisierung von Zweikanal-Microarrays.

Je nach geladenen Array-Typen stellt GEPAT verschiedene Möglichkeiten für die Normalisierung zur Auswahl, die in Tabelle 5 zusammengefasst werden.

Tabelle 5: Normaliserungsverfahren für Zweikanal-Microarraydaten

Name	*Local Normalisierung*	*Scale Normalisierung*
Loess	Loess	Scale
Quantile	Loess	Quantile
VSN	None	VSN

Das loess-Normalisierungsverfahren verwendet innerhalb eines Arrays die loess-Normalisierung und normalisiert die Arrays mithilfe des scale-Verfahrens. Die Quantile-Normalisierungsmethode verwendet die quantile-Normalisierung für die Normalisierung der verschiedenen Arrays. Die VSN-Methode führt keine explizite lokale Normalisierung durch, da hier die Normalisierungen in den Arrays und zwischen den Arrays in einem Schritt behandelt werden.

Vor der Normalisierung kann entweder keine Hintergrundkorrektur vorgenommen werden, oder die **`normexp`**-Methode zur Hintergrundkorrektur verwendet werden. Diese Methode stellt sicher, dass keine negativen Werte in den korrigierten Daten enthalten sind.

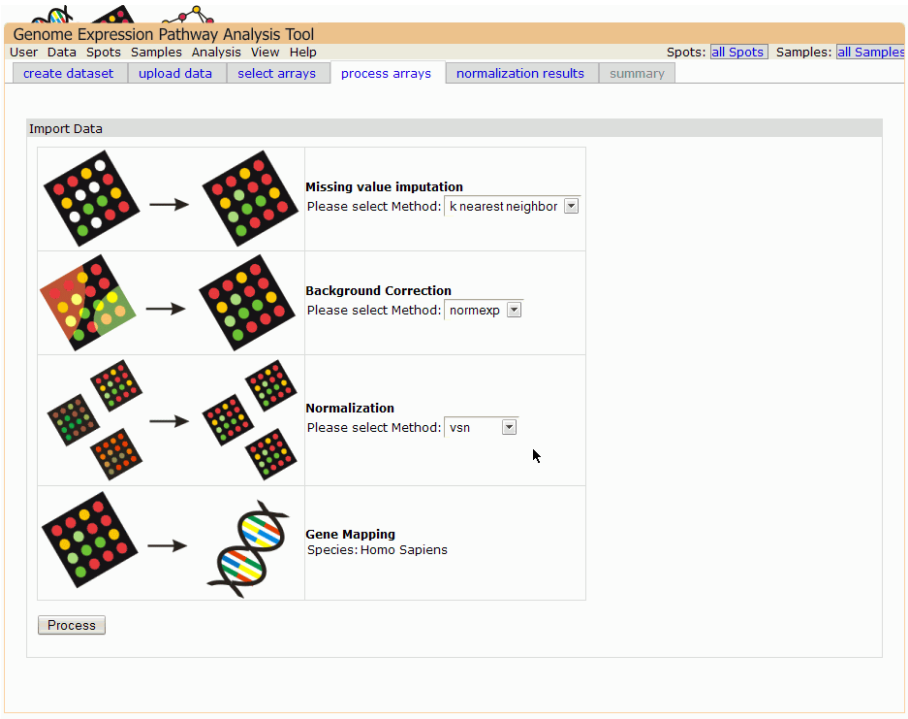


Abbildung 26: Einstellung der Normalisierungsparameter
Die Verfahren zur Berechnung fehlender Werte, zur Hintergrundkorrektur und zur Normalisierung können hier eingestellt werden.

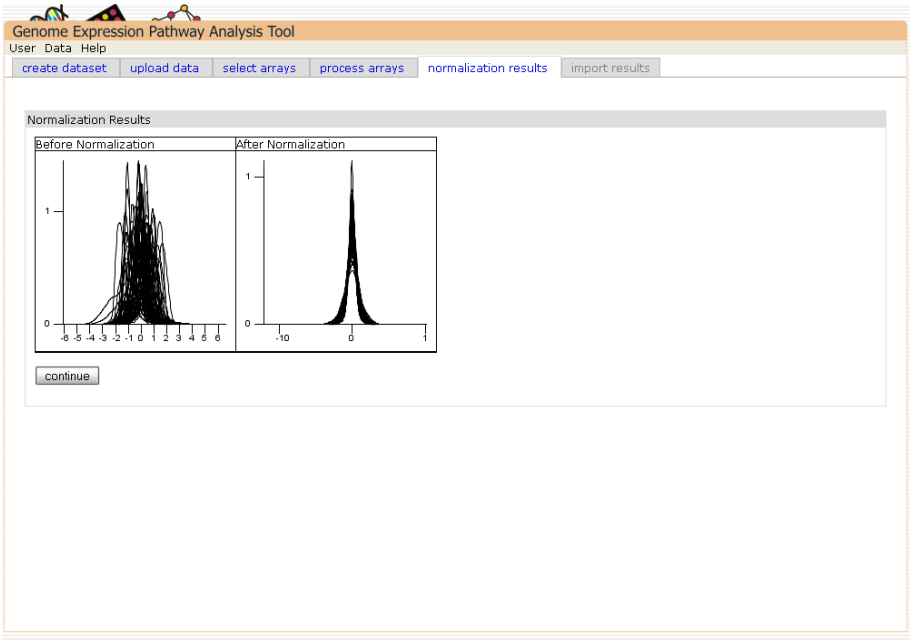


Abbildung 27: Ergebnisse der Normalisierung
Die Grafiken zeigen die Dichteverteilung der Microarrays vor und nach der Normalisierung.

3.5.4 Annotation

Mit dem Import der Microarray-Daten wird die Genexpressionsmatrix erstellt, und Analysen können auf den Daten vorgenommen werden. Allerdings sind noch keine Informationen zu den Transkripten bekannt, deren Expression auf den Microarrays gemessen wurde. Hierfür müssen die Daten annotiert werden. Geninformationen können aus einer Reihe von Datenbanken bezogen werden, GEPAT verwendet die Ensembl-Datenbank (siehe Kapitel 2.3.1) als Grundlage für alle weiteren Informationen. Bei Zweikanal-Microarrays ist hier der Name der Gensonden der Ausgangspunkt, bei Oligonukleotid-Microarrays der Name des Transkripts, der im Zusammenfassungsschritt ermittelt wird. Informationen zu den Transkripten der Oligonukleotid-Microarrays sind bereits in der Ensembl-Datenbank enthalten.

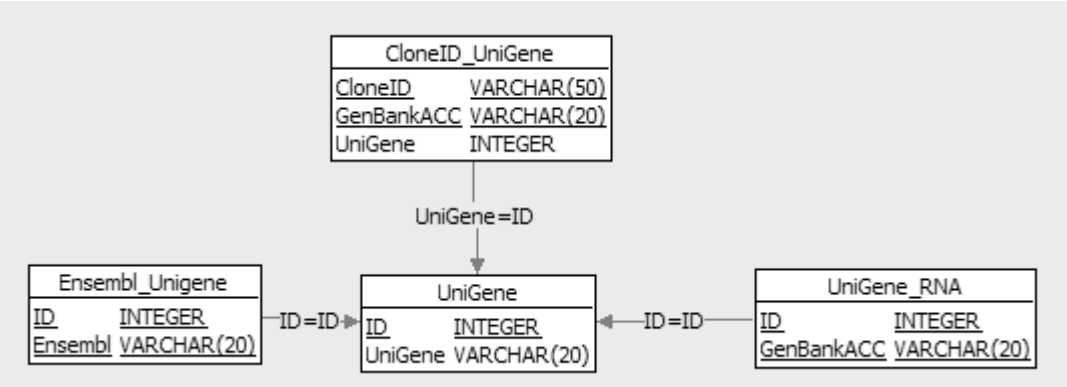


Abbildung 28: Struktur der zur Annotation verwendeten Datenbank
Zu CloneID und Genbank-Kennungen kann die entsprechende UniGene-Kennungen ermittelt werden. Zu dieser gehört eine Ensembl Genkennung.

Zur Genannotation gab es jedoch keine Datenbank, die in der Lage war, alle Kennungen für cDNA-Gensonden zu identifizieren. Deshalb wurde eine eigene Datenbank geschaffen. Die UniGene-Datenbank [149] wird verwendet, um eine Zuordnung von cDNA-Clone-Namen und GenBank-Einträgen zu UniGene-Clustern zu ermöglichen. Der UniGene-Cluster-Name wird anschließend verwendet, um das entsprechende Gen in der Ensembl-Datenbank zu finden. Damit ist eine Zuordnung von cDNA-Sonden auf einen Ensembl-Gennamen möglich. Affymetrix-Gensondennamen werden direkt über die Informationen aus der Ensembl-Datenbank identifiziert. Im Moment ist die Genannotation noch auf das menschliche Genom beschränkt. Abbildung 28 zeigt die Struktur der Annotationsdatenbank.

Nicht alle Gensonden auf dem Array können einer Ensembl-Kennung zugewiesen werden. Einige Sonden sind aus EST-Bibliotheken entwickelt worden, für die kein Gen annotiert werden kann, entweder weil es sich um nichtkodierende mRNA handelt, die Sequenz nicht eindeutig einem Gen zuordenbar ist, oder die Bibliothek fehlerhaft ist. Falls eine Ensembl-Kennung zugeordnet werden kann, werden die Informationen aus der Ensembl-Datenbank genutzt, um den Gennamen, die Position auf dem Chromosom, die GO-Kennung und die enzymatische Aktivität zu bestimmen. Jegliche Datenannotation in GEPAT findet über die Ensembl-Genkennung statt. Die Art der Bezeichnung der Gensonden, die für die Annotation verwendet wird, wird automatisch aus den Array-Files bestimmt, oder muss vom Benutzer angegeben werden, falls Tabellen zur Dateneingabe verwendet werden.

Die Verbindung einer Sonde zu einem Gen ist notwendig, um Ergebnisse interpretieren zu können. Nicht immer entspricht der gemessene Expressionswert jedoch dem Expressionswert des Gens. In manchen Fällen kann es vorkommen, dass nicht nur eine mRNA mit der Sonde hybridisiert, sondern es zur Kreuzhybridisierung mit der mRNA von verschiedenen Genen kommt. Es kann auch vorkommen, dass eine Sonde nur eine bestimmte Spleißvariante eines Gens erkennt, während andere Sonden alle Spleißvarianten erkennen. Verschiedene Sonden können unterschiedlich gut mit der mRNA binden, für die sie entwickelt wurden. Und schließlich ist nicht immer klar, ob die Gensonden auch das cDNA-Material enthalten, das als Kennung angegeben wurde. Deshalb ist es wichtig, die Ergebnisse von Microarray-Experimenten nochmals zu überprüfen. Ein Vergleich mit Sequenzdatenbanken zeigt, mit welcher mRNA die Gensonden binden können, anderer experimentelle Methoden sollten verwendet werden, um fehlerhafte Gensonden ausschließen zu können.

3.6 Ensembl

Natürlich ist es wichtig, den gemessenen Werten auf dem Microarray eine Bedeutung zuordnen zu können. Mit der Annotation ist der erste und wichtigste Schritt gemacht, da es jetzt möglich ist, die gemessenen Transkripte auf dem Microarray direkt den Genen zuzuordnen, aus denen sie entstanden sind. Ein Teil der Informationen, die in der Ensembl-Datenbank enthalten sind, können direkt in GEPAT angezeigt werden. Die Struktur der Ensembl-Datenbank ist umfangreich. Deshalb greift GEPAT nicht direkt auf

die Datenbank zu, sondern verwendet die ensj-Java API [142] für die Zugriffe.

3.6.1 Geninformation

Für jedes Gen in GEPAT steht eine Teilmenge der in Ensembl enthaltenen Informationen zur Verfügung, die direkt in die Ensembl-Datenbank verlinkt sind. Als Informationen stehen der Genname, eine Kurzbeschreibung, die Chromosomposition, GO Kennung, enzymatische Aktivitäten und die Expressionswerte über den Proben zur Verfügung und meist kann direkt zu den entsprechenden Informationsseiten in GEPAT weitergeleitet werden. Diese Informationen werden in der Datensatzübersichtsseite angezeigt, und sie werden auch bei den Geninformationen für ein einzelnes Gen angezeigt. Abbildung 29 zeigt die Geninformationen.

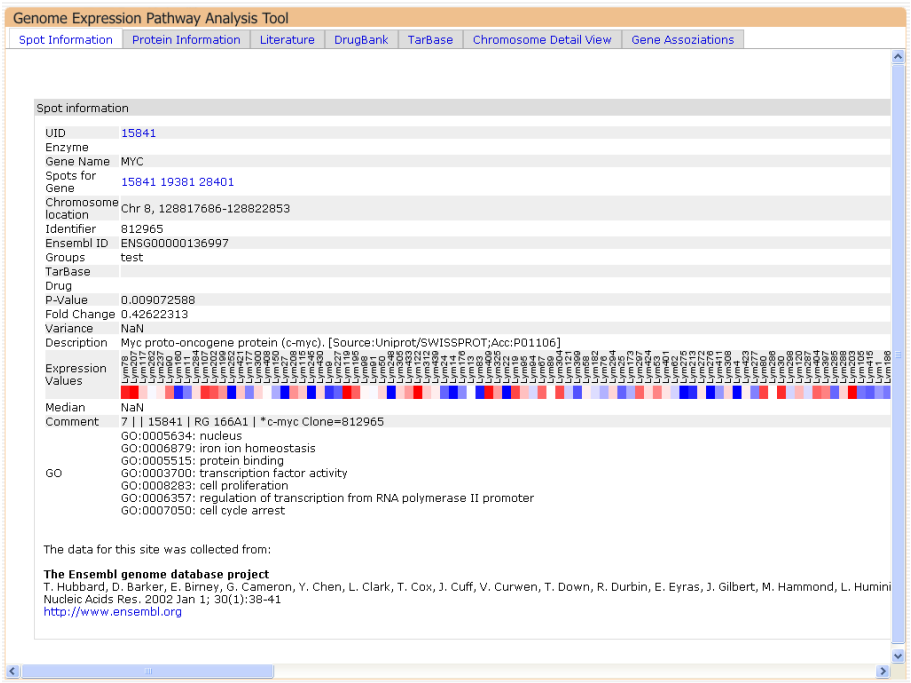


Abbildung 29: Geninformationsseite für ein Gen
Sie zeigt Auszüge aus der Ensembl-Datenbank. Über die Reiter am oberen Fensterrand können die Geninformationsseiten weiterer Module aufgerufen werden.

3.6.2 Proteininformation

Obwohl der aktuelle Stand der Microarraytechnik für jedes Gen die Expressionswerte der einzelnen Spleißvarianten erkennen kann [150], sind die meisten derzeit verwendeten Microarrays nur in der Lage die Expression auf Genebene zu bestimmen. Ob dabei die Expressionswerte für alle Spleißvarianten oder nur ausschließlich einer bestimmte

Spleißvariante gemessen werden, hängt vom Design der Gensonden ab. Manchmal kann es auch notwendig sein, weitere Informationen über die Proteine zu erhalten, die aus dem Gen entstehen können. Diese Information wird in der Proteininformationssicht gegeben, gezeigt in Abbildung 30. Die Proteininformationen werden aus der Ensembl-Datenbank erzeugt, ein direkter Link zum entsprechenden Eintrag in Ensembl ist für jedes Protein vorhanden. Neben den verschiedenen möglichen Transkripten für ein Gen werden Eigenschaften der Proteinsequenz angezeigt, z.B. Domänen aus SMART [151] und Pfam [152] oder Sequenzen mit geringer Komplexität.

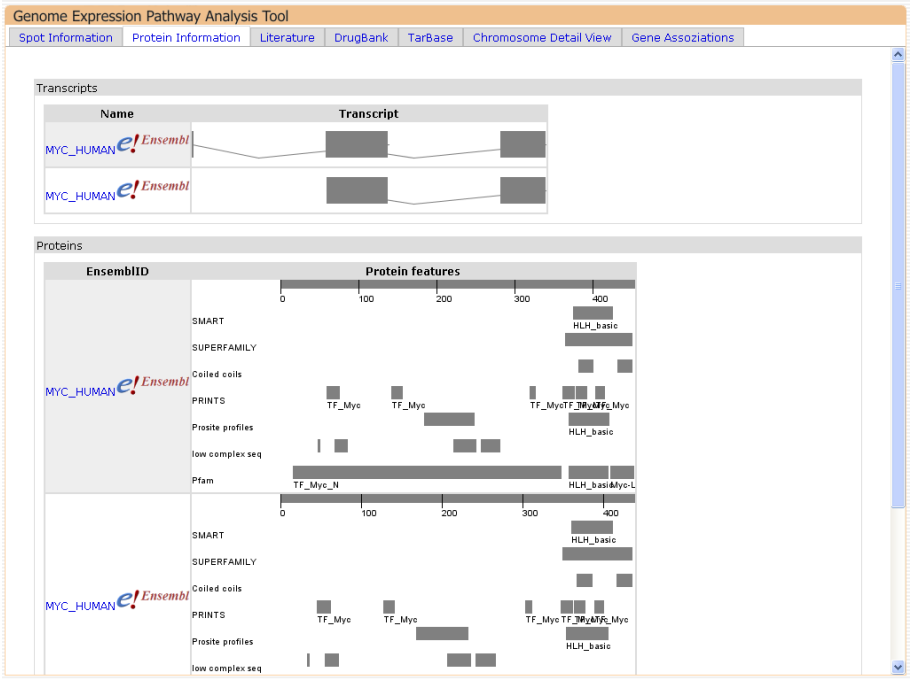


Abbildung 30: Proteininformationsseite
Sie fasst die Ensembl-Informationen der zu einem Gen gehörenden Proteine zusammen. Die grafische Übersicht zeigt die Intron- und Exonstruktur sowie die Proteineigenschaften in grafischer Zusammenfassung.

3.6.3 Literaturreferenzen

Artikel zu bestimmten Genen können in einer Vielzahl wissenschaftlicher Fachzeitschriften gefunden werden. Um eine schnelle Übersicht über wissenschaftliche Artikel zu einem Gen zu erhalten, ist in GEPAT eine Referenz-Übersicht implementiert.

Abbildung 31: Literaturübersicht
Die Kurzzusammenfassung der Genfunktion sowie die entsprechenden Artikel stammen aus der RefSeq-Datenbank

Das Reference Sequence (RefSeq) Datenbank [153] des National Center for Biotechnology Information (NCBI) enthält eine Sammlung von Sequenzen, die Genomdaten, Transkripte und Proteine umfassen, die im Gegensatz zur Genbank [154] nichtredundant sind. Ziel der Datenbank ist es, die kompletten Sequenzinformationen für alle Spezies zu erfassen. Derzeit enthält die Datenbank Sequenzinformationen zu 2400 Organismen und speichert über eine Million Proteinsequenzen, wobei eine signifikante Taxonomie-Diversität über Prokaryoten, Eukaryoten und Viren repräsentiert wird.

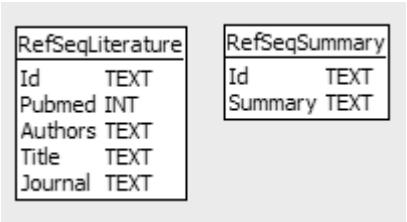


Abbildung 32: Datenbankstruktur zur Speicherung der RefSeq-Einträge
Zu jedem RefSeq-Eintrag wird die Ensembl-Genkennung ermittelt, die als Schlüssel dient.

GEPAT verwendet die RefSeq-Datenbank, um mithilfe der in Ensembl vorhandenen

RefSeq-Kennung Literaturinformationen für ein Gen zu erhalten. Neben Verweisen zu PubMed-Einträgen liefert RefSeq zu vielen Proteinen auch eine kurze Beschreibung der Funktion. Abbildung 31 zeigt die Literatursicht. Die Informationen werden aus dem RefSeq-Flatfile ausgelesen und in einer Datenbank gespeichert. Abbildung 32 zeigt die Struktur dieser Datenbank.

3.7 Datenanalyse

Mit dem Import der Microarraydaten wurde die Genexpressionsmatrix erstellt, durch die Annotation sind Informationen für die Einträge der Matrix verfügbar gemacht worden. Damit ist eine Analyse der Microarraydaten möglich. GEPAT stellt hierzu eine Vielzahl von Methoden zur Verfügung, die auf den in Bioconductor verfügbaren Algorithmen basieren. Weitere Methoden können durch die modulare Unterstützung hinzugefügt werden. Mit der ebenfalls modularen Teilmengenauswahl ist es möglich, jegliche Art von Teilmengen aus den Gensonden oder Proben als Grundlage der Analyse zu verwenden. Die Ergebnisse der Analyse können wiederum als Grundlage von weiterer Analyse oder Interpretation dienen, wenn ein entsprechendes Modul zur Teilmengenselektion zur Verfügung steht.

3.7.1 Differentielle Expression

Um Unterschiede in der Genexpression zwischen zwei Gruppen biologischer Proben feststellen zu können, kann ein Test auf differentielle Expression durchgeführt werden (siehe Kapitel 2.4.4).

GEPAT verwendet das limma-Paket von Bioconductor für diese Art der Analyse. Zwei Probenteilmengen können ausgewählt und verglichen werden. Bonferoni-Step-Down, Benjamini-Hochberg und Benjamini-Yekutieli Multiple-Testing Korrekturmethoden können angewandt werden. Für jedes Transkript wird als Testergebnis der $\log_2$-Foldchange zwischen den beiden Teilmengen und ein p-Wert berechnet.

Das Resultat des t-Tests kann in einem M/A – Plot dargestellt werden, der einen Überblick über die Datenverteilung gibt. Jeder Punkt steht dabei für einen Messwert des Datensatzes. Die Y-Achse zeigt den $\log_2$-Foldchange von Genexpressionswerten in den verglichenen Gruppen. Die X-Achse zeigt den A-Wert, den durchschnittlichen Expressionswert für das Transkript über alle Arrays und Kanäle. Zusätzliche Information

zum M/A Plot sind durch den Mauscursor über Tooltips erreichbar. Ein Klick auf einen Punkt öffnet ein neues Fenster mit Geninformationen. Abbildung 33 zeigt die Ergebnisse des Tests auf differentielle Expression. Das Fold-Change Ergebnis der differentielle Expressionsanalyse kann direkt auf die Visualisierungskomponenten übertragen werden, um so die differentielle Expression auf metabolischen Netzwerken oder Geninteraktionskarten anzeigen zu können.

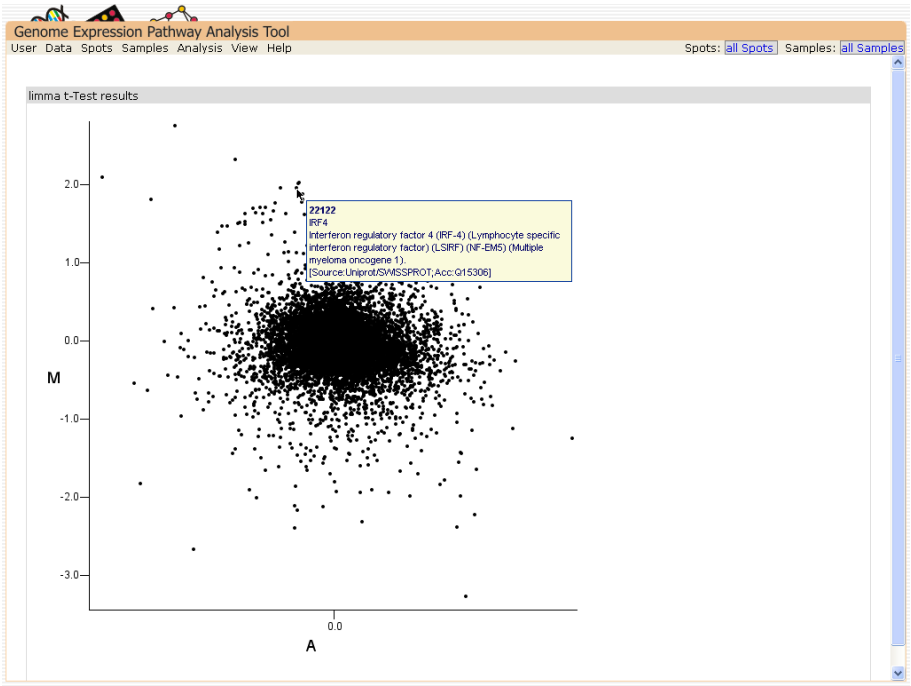


Abbildung 33: Ergebnisse des limma t-Test
Die Ergebnisse werden als M/A Plot dargestellt, in dem die X-Achse den durchschnittlichen Expressionswert einer Sonde und die Y-Achse den Unterschied zwischen den beiden Gruppen anzeigt. In diesem Beispiel wurde die Genexpression zweier Patientengruppen mit diffusen großzelligen B-Zell-Lymphomen verglichen.

Die Ergebnisse der differentiellen Expression können verwendet werden, um Teilmengen aus der Menge der Gensonden basierend auf dem p-Wert und/oder dem Fold-Change auszuwählen.

3.7.2 Varianz & Median

Andere Charakteristiken der Microarray-Daten können direkt aus den Expressionwerten berechnet werden. Medianwerte oder die Varianz für ein Transkript können für beliebige Teilmengen der Transkripte und Proben berechnet werden. Dies ermöglicht es beispielsweise, nur Transkripte mit hoher Varianz für eine Analyse auszuwählen, wie in

Abbildung 34 gezeigt.

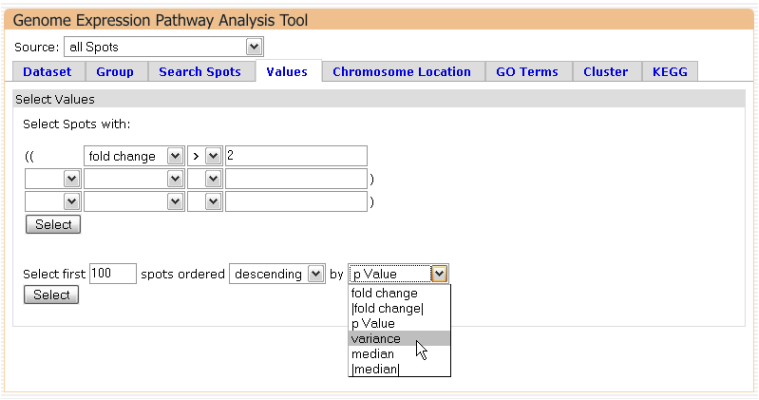


Abbildung 34: Teilmengenauswahl basierend auf Datencharakteristiken
Es ist möglich, nur Transkripte mit Werten innerhalb eines bestimmten Intervalls, oder eine gewisse Anzahl an Transkripten mit den höchsten oder geringsten Werten auszuwählen.

3.7.3 Clustering

Der Test auf differentielle Expression erlaubt es, Expressionsunterschiede zwischen verschiedenen Gruppen biologischer Proben zu finden. Sind die Eigenschaften der biologischen Proben jedoch unbekannt, ist das Verfahren nicht ohne weiteres möglich. Hier ist es sinnvoll, zuerst nach Muster in den Expressionsdaten zu suchen, die evtl. eine Einteilung der Proben in verschiedene Gruppen erlauben. Hierzu kann Clustering eingesetzt werden, das es ermöglicht, bestimmte Signaturen oder Muster in der Genexpression automatisch zu finden (siehe Kapitel 2.4.5). GEPAT unterstützt verschiedene Methoden zum Clustern von Daten:

- hierarchisches Clustern
- k-Means Clustern
- Hauptkomponentenanalyse

Als Datengrundlage, auf der das Clustering erfolgt, können beliebige Teilmengen an Transkripten oder Proben ausgewählt werden.

Hierarchisches Clustern

Hierarchisches Clustern wird über die R-Funktionen **`dist`** und **`hclust`** realisiert. Als Distanzfunktion können die euklidische Metrik, die Manhattan-Metrik und die Pearson- und Spearman-Korrelationsdistanzen verwendet werden. Als Clustering-Methoden

können UPGMA, Single linkage, Complete linkage und Ward´s Algorithmus verwendet werden. Beim Clustern werden die Transkripte und Samples neu geordnet und der durch das Clustering erzeugte Baum, das Dendrogramm, wird angezeigt. Der Export des Dendrogramms aus R erfolgt über die Funktion **`hc2Newick`** des Packets **`ctc`** im Newick-Format [155]. Die entsprechende R-Funktion ist fehlerhaft und wurde entsprechend korrigiert. Zum Einlesen der exportierten Datei wurde ein Newick-Parser in der Programmiersprache Java implementiert. Die Ergebnisse des hierarchischen Clustern werden als neu sortierte Genexpressionsmatrix zusammen mit dem durch das Clustern entstandene Dendrogramm angezeigt. Abbildung 35 zeigt diese Ergebnisse.

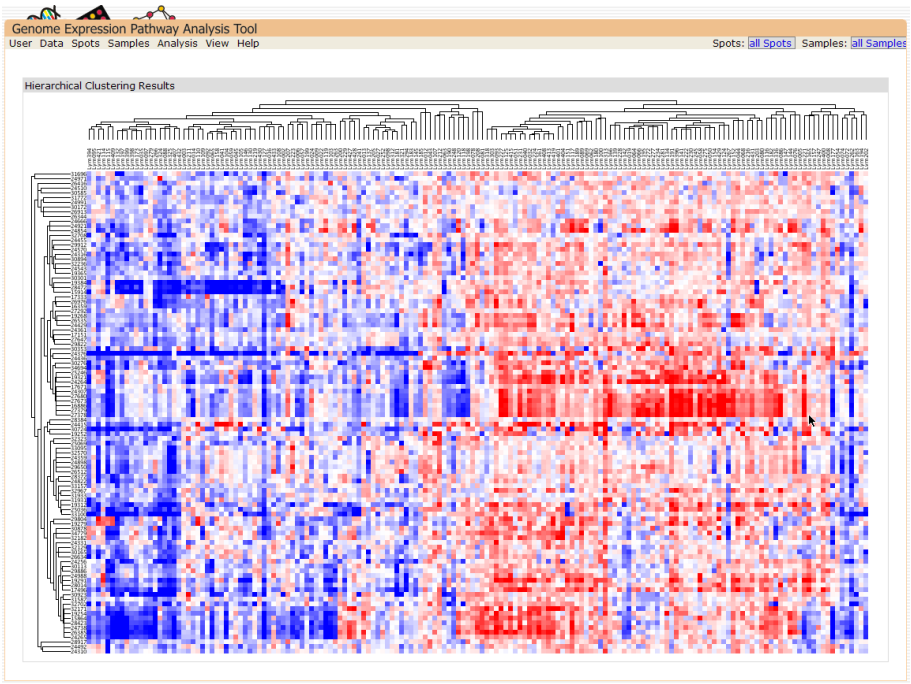


Abbildung 35: Ergebnisse des hierarchischen Clusterns
Als Eingabedaten wurden Microarraydaten aus diffusen großzelligen B-Zell-Lymphomen verwendet. Die Aufteilung der Proben in zwei Gruppen ist deutlich zu erkennen.

PCA

Die Hauptkomponentenanalyse kann in GEPAT für die Proben des Datensatzes, basierend auf den Messwerten der Transkripte, durchführen. Dabei wird die R-Funktion **`prcomp`** zur Durchführung verwendet. Als Ergebnisse der Hauptkomponentenanalyse werden die Proben in einem zweidimensionalen Diagramm angezeigt, wobei die Hauptachsen für jede Dimension frei gewählt werden können. Eine Auswahlfunktion zur Proben-Selektion ermöglicht eine einfache Teilmengenselektion basierend auf den

Resultaten der PCA. Abbildung 36 zeigt die Ergebnisse einer Hauptkomponentenanalyse. PCA erlaubt es unter Umständen, die Anzahl der Cluster visuell abzuschätzen. Dies kann verwendet werden, um die Anzahl der Cluster beim k-Means Clustering anzunehmen.

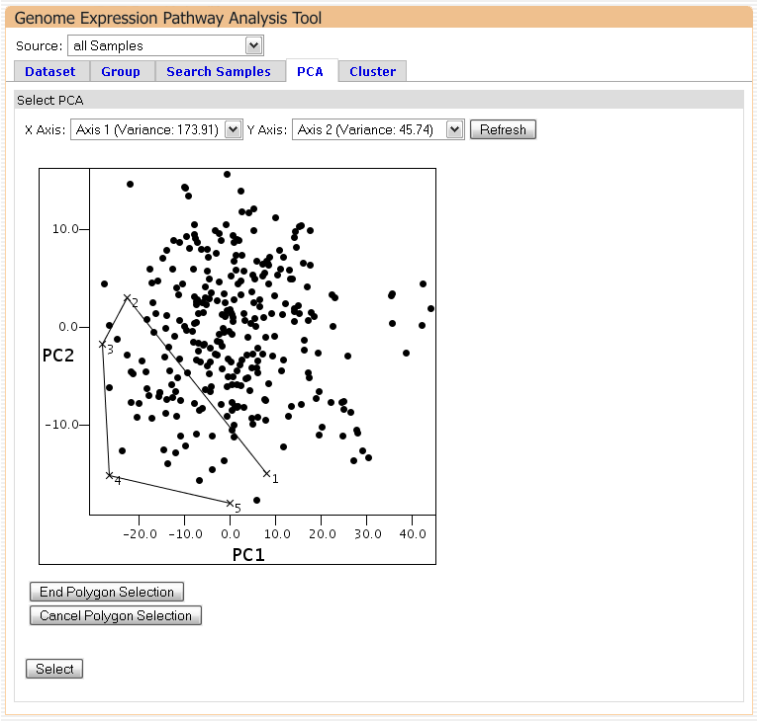


Abbildung 36: Die Ergebnisse der Hauptkompontenanalyse
Die beiden Hauptkomponenten mit der höchsten Varianz bilden die X und Y Achse, die Punkte markieren biologische Proben. GEPAT erlaubt die Verwendung der Ergebnisse zur Teilmengenauswahl. Mit der Maus können die Proben aus dem Ergebnis ausgewählt werden.

k-means Clustering

Das k-means Clustering benötigt eine Benutzereingabe, die erwartete Anzahl k an Clustern. GEPAT verwendet den **`kmeans`** Befehl von R um Clustering basierend auf dem Hartigan-Wong Algorithmus [156] durchzuführen. Als Resultat werden k Cluster zurückgeliefert, die mit der Teilmengenselektion als Grundlage für weitere Berechnungsschritte verwendet werden können. Die Ergebnisse können auch als Grundlage für weiteres Clustern dienen und sie erlauben so eine Schritt-für-Schritt Analyse komplexer Datensätze. Abbildung 37 zeigt Ergebnisse des kMeans-Algorithmus.

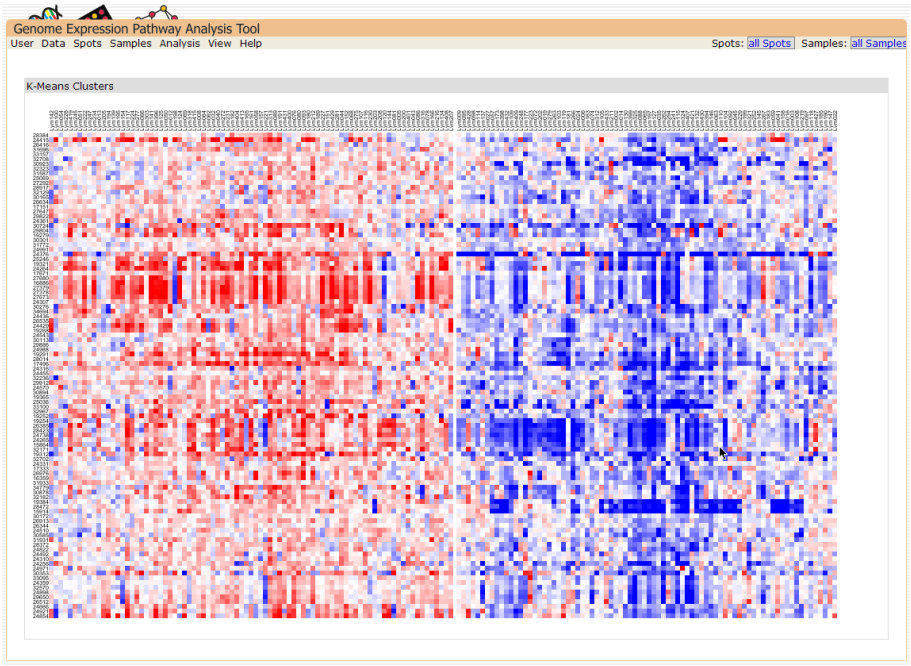


Abbildung 37: Ergebnisse des k-Means Clusterin
Das k-Means Clustern wurde auf dieselben Daten wie das hierarchische Clustern angewendet. Durch das k-Means Clustern werden die Daten in zwei Gruppen eingeteilt.

3.8 Chromosomale Daten

3.8.1 Chromosomposition

Um die Beziehung zwischen der Genexpression und der physischen Position des Gens auf dem Chromosom zu untersuchen ist eine kombinierte Ansicht von Expression und Chromosomposition der Transkripte nötig. Die Informationen aus der Ensembl-Datenbank beinhalten auch die chromosomale Lokalisation der zu den Transkripten gehörenden Gene. Mithilfe der ebenfalls in Ensembl enthaltenen Karyogramm-Informationen lässt sich so eine graphische Abbildung der Genexpression über die Chromosomen hinweg darstellen. Dabei werden auch die typischen Bandenfärbungen sowie die Bereiche des Zentromers oder ribosomkodierende Bereiche angezeigt. GEPAT erzeugt eine Übersicht über alle Chromosomen des Genoms, mit der Maus kann in interessante Chromosombereiche eingezoomt werden. In der Zoomansicht stehen dann Tooltips für jedes Gen zur Verfügung, über die sich weitere Daten abrufen lassen. So lassen sich schnell interessante Teile des Genoms finden und untersuchen. Abbildung 38 zeigt die Zoomansicht. Die Chromosomansicht kann auch verwendet werden, um Teilmengen aus der Menge der Transkripte auszuwählen, wie in Abbildung 39 gezeigt.

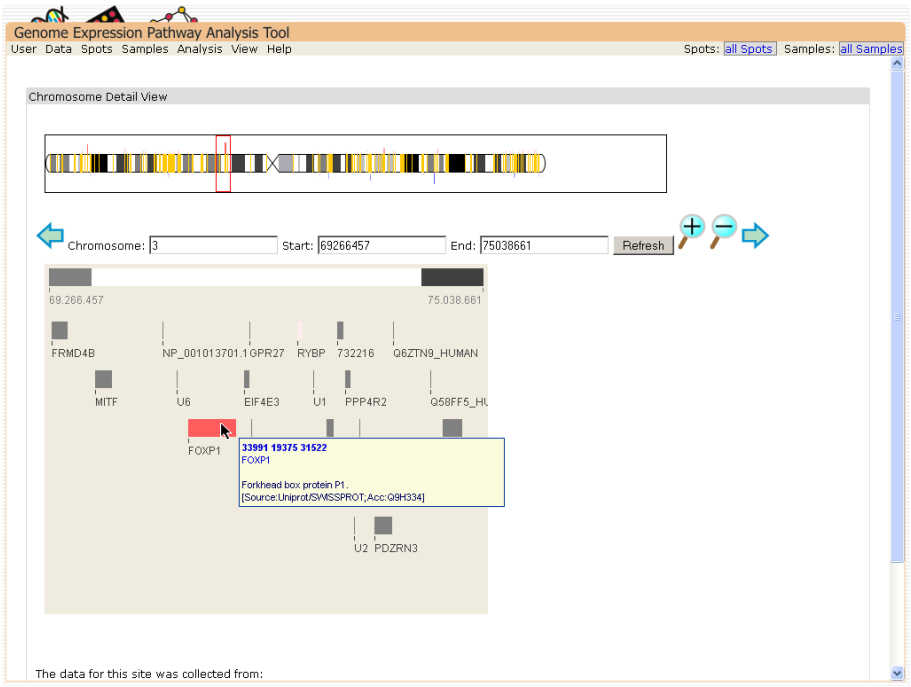


Abbildung 38: Genansicht
Auf der oberen Chromosomübersicht kann durch Aufziehen eines Bereiches mit der Maus eine Detailansicht der ausgewählten Region dargestellt werden. Die Expressionswerte einer Teilmenge der Gene können auf die Ansicht übertragen werden.

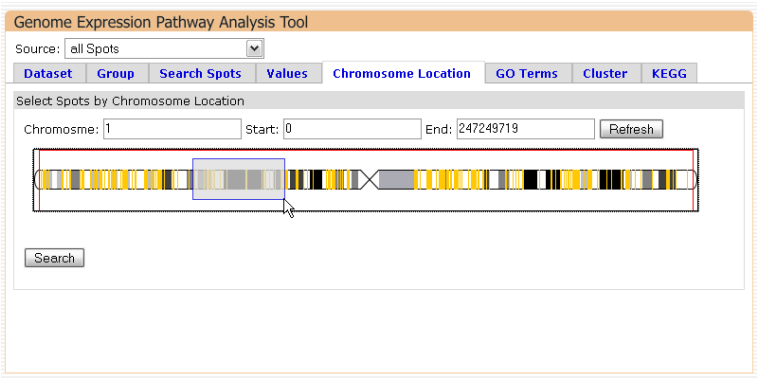
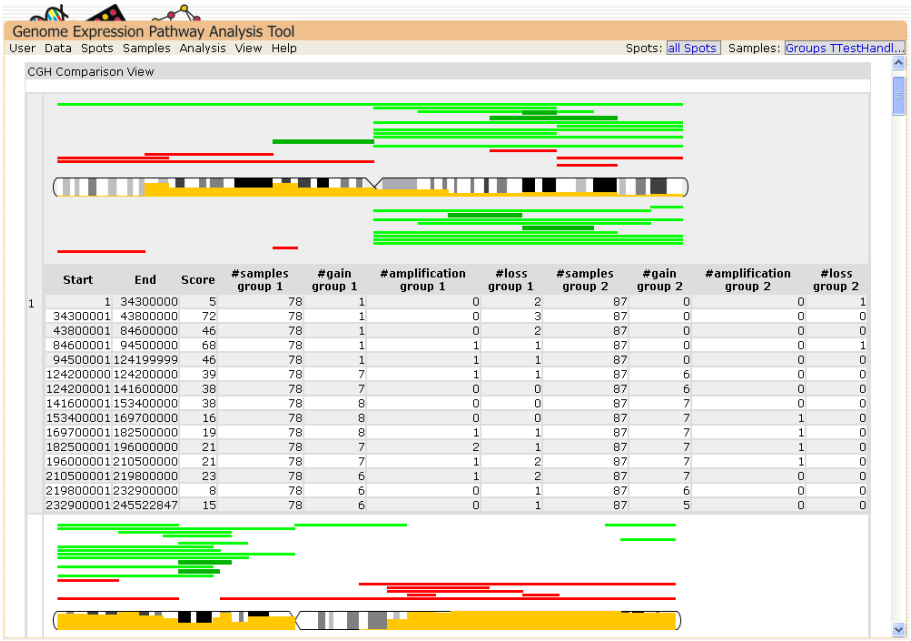


Abbildung 39: Teilmengenauswahl nach chromosomaler Lokalisation
Mit der Maus oder durch Positionsangabe können Transkripte basierend auf ihrer chromosomalen Position auswählt werden.

GEPAT verwaltet nicht nur Microarray-Daten, sondern kann auch zusätzliche Informationen über chromosomale Änderungen der Proben speichern (siehe Kapitel 2.2.7). Zu einem beliebigen Chromsomabschnitt kann festgehalten werden, ob dieser verdoppelt oder vervielfacht wurde, oder verloren ging. Durch diese Abschnitte kann ein CGH-Profil für jede biologische Probe gespeichert werden. Der Import erfolgt über eine Tabelle, in der eine Spalte jeweils die Probennamen enthält, und andere Spalten die

Chromosomenbereiche, die verdoppelt, vervielfacht oder verloren wurden, beschreiben. Damit sind die Eingabedateien wesentliche einfacher gehalten, als die Dateien, die von anderen Datenbanken, wie z.B. der NCBI SKY/M-FISH & CGH Datenbank [157], verwaltet werden. Array-CGH Dateien können derzeit noch nicht in GEPAT importiert werden.



	Start	End	Score	#samples group 1	#gain group 1	#amplification group 1	#loss group 1	#samples group 2	#gain group 2	#amplification group 2	#loss group 2
1	1	34300000	5	78	1	0	2	87	0	0	1
	34300001	43800000	72	78	1	0	3	87	0	0	0
	43800001	84600000	46	78	1	0	2	87	0	0	0
	84600001	94500000	68	78	1	1	1	87	0	0	1
	94500001	124199999	46	78	1	1	1	87	0	0	0
	124200000	124200000	39	78	7	1	1	87	6	0	0
	124200001	141600000	38	78	7	0	0	87	6	0	0
	141600001	153400000	38	78	8	0	0	87	7	0	0
	153400001	169700000	16	78	8	0	0	87	7	1	0
	169700001	182500000	19	78	8	1	1	87	7	1	0
	182500001	196000000	21	78	7	2	1	87	7	1	0
	196000001	210500000	21	78	7	1	2	87	7	1	0
	210500001	219800000	23	78	6	1	2	87	7	0	0
	219800001	232900000	8	78	6	0	1	87	6	0	0
	232900001	245522847	15	78	6	0	1	87	5	0	0

Abbildung 40: Vergleich von CGH-Daten
Es wurden zwei Gruppen biologischer Proben verglichen. Über dem entsprechenden Chromosom werden die chromosomalen Abberationen der ersten Gruppe dargestellt, unter dem Chromosom die der zweiten. Grüne Balken stehen für vervielfachte, rote Balken für verlorene Chromosombereiche.

GEPAT enthält ein Datenanalyse-Modul, das in der Lage ist, die CGH-Profile von zwei Proben-Teilmengen zu vergleichen. Dazu wird ein ungepaarter Wilcoxon-Rang Test für jeden chromosomalen Abschnitt ausgeführt, der aus diesem Test resultierende p-Wert wird direkt in eine Chromosomansicht eingezeichnet, zusammen mit einer Ansicht des CGH-Profils für die beiden Teilmengen, und erlaubt so eine schnelle Identifizierung von Abschnitten mit unterschiedlichem Profil. Abbildung 40 zeigt die Ergebnisse einer Profilanalyse.

3.9 Gene Ontonolgy

3.9.1 GO Term Enrichment Analysis

Wurde beim Test auf differentiell exprimierte Gene eine Anzahl von Genen gefunden,

oder wurde auf andere Art und Weise, z.B. bei der CGH-Analyse, eine Liste von interessanten Genen entdeckt, so ist die biologische Bedeutung dieser Gene interessant. Eine häufig benutzte Möglichkeit, um Informationen über die Gene zu erhalten, ist ein automatischer, ontologiebasierter Ansatz (siehe Kapitel 2.4.6). Hierfür wird die GO-Ontologie des Gene Ontologie Project [66] verwendet. Es wird überprüft, welche GO-Kategorien in einer Menge interessanter Gene signifikant über- oder unterrepräsentiert sind. Anhand dieser GO-Kategorien lassen sich Rückschlüsse auf die Funktion der Gene in der Genmenge ziehen.

GEPAT verwendet einen zweiseitigen Test und berechnet nach dem in [100] beschriebenen Verfahren einen mid-p-Wert nach der Formel

$$p(k)=2\times min(P(\kappa>k)+\frac{1}{2}P(\kappa=k), P(\kappa<k)+\frac{1}{2}P(\kappa=k))$$

Der mid-p-Wert wird verwendet, um den Einfluss der diskreten Verteilung auf den Signifikanztest zu minimieren. Zur Berechnung der Werte werden R-Funktion **`phyper`** und **`dhyper`** verwendet. Aufgrund der Graph-Struktur von GO ist eine Mutliple-Testing-Korrekter in der GO Analyse nicht trivial und wird noch immer diskutiert [158]. Deshalb steht sie im Moment noch nicht zur Verfügung.

Die Resultate der GO Term Enrichment Analyse werden in einem Baum gezeigt, der die gerichtete azyklische Graph-Struktur von GO abbildet. Da die Darstellung des Graphen relativ unübersichtlich werden kann, stellen viele Programme, auch GEPAT, den Graphen als Baum dar. Die Baumansicht des Graphen ist übersichtlicher und ermöglicht eine einfache Navigation. Hierbei entspricht jeder Pfad im Graphen einem Pfad im Baum, Knoten im Graph mit mehreren Eltern werden durch mehrere Knoten gleicher Art im Baum abgebildet. Damit kann ein GO-Eintrag öfters im Baum vorkommen.

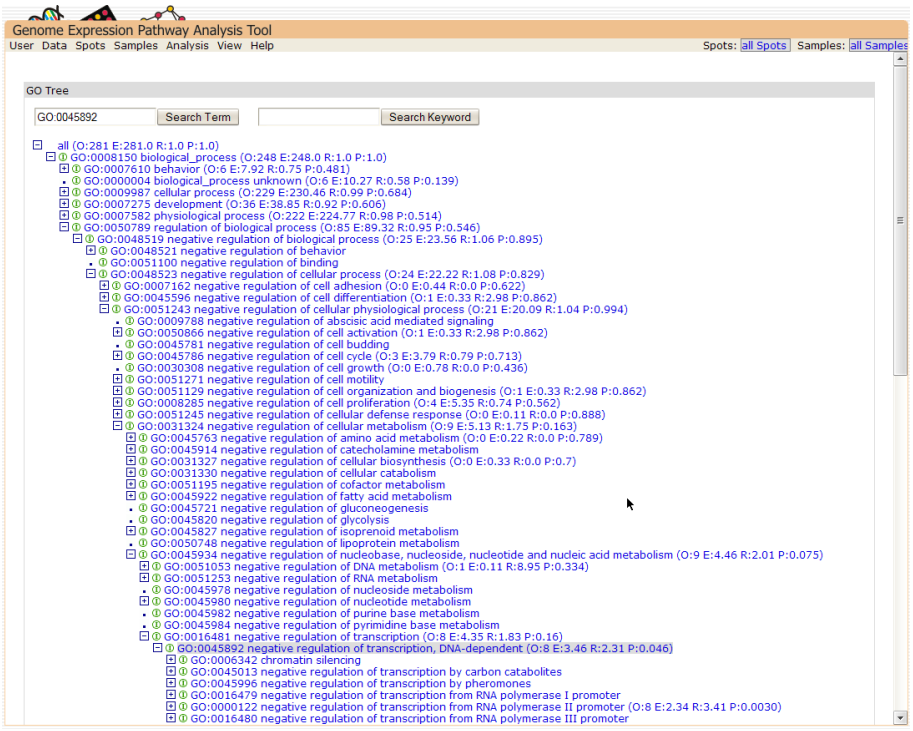


Abbildung 41: Darstellung des direkten azyklischen GO-Graphen als Baum
Ein Analyse auf über/unterrepräsentierte Terme wurde durchgeführt, die Ergebnisse werden hinter dem Kategorienamen angezeigt. O bezeichnet die beobachtete Anzahl an Genen, E die erwartete Anzahl, R das Verhältnis zwischen den beiden Werten und P den p-Wert.

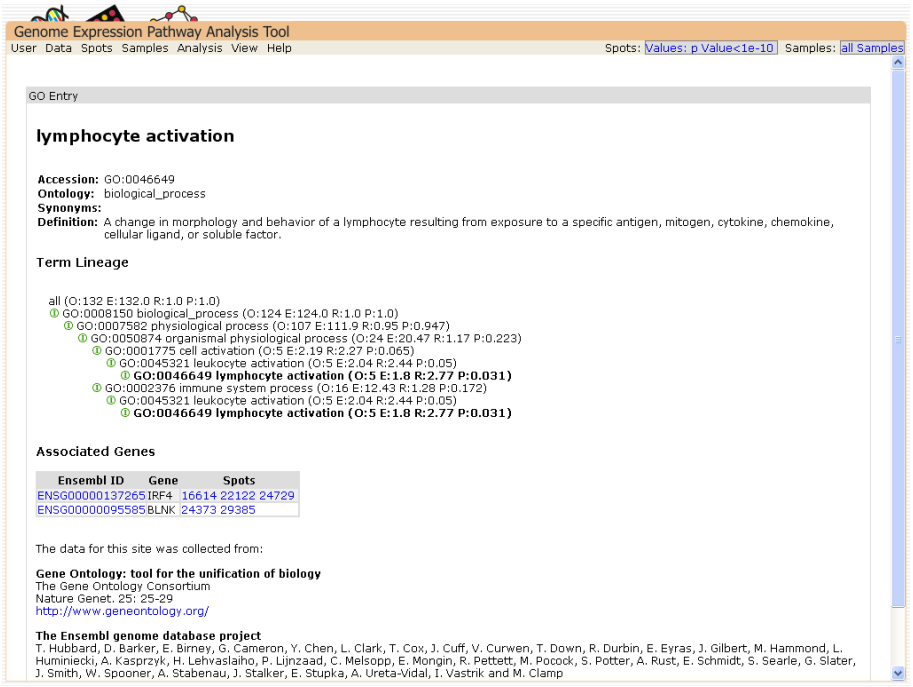


Abbildung 42: Detailansicht für eine GO-Kategorie
Neben Informationen zum Eintrag und den Vorkommen im GO-Baum werden auch die Gene angezeigt, die zu dieser Kategorie gehören.

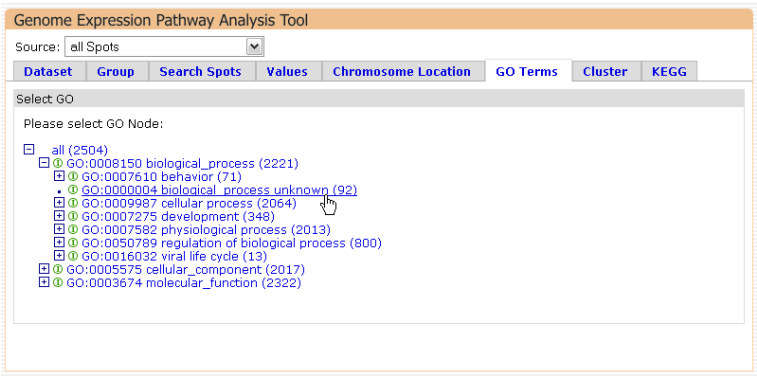


Abbildung 43: Teilmengenauswahl nach GO-Kategorien
Ein Eintrag kann aus dem GO-Baum ausgewählt werden, die Transkripte der entsprechenden Gene werden als Teilmenge übernommen. Die Zahl hinter der Kategorie gibt die Anzahl der Gene an, die zu dieser Kategorie gehören.

3.9.2 OBO

Die Gene Ontologie wird in Form einer Datei im OBO-Format [159] gespeichert. Diese Datei besteht aus einem Header und Absätzen für die verschiedenen GO-Kategorien, getrennt durch Leerzeilen. Jeder Absatz besteht aus einer Reihe Name/Wert-Paaren, die u.a. die Kennung, Namen, Definition und die Elternelemente in der Beziehung is_a und part_of enthalten. Weitere Informationen finden sich auf der Interseite des GO Konsortiums [160]. Die Ontologie steht auch als Datenbank für mySQL zur Verfügung. Allerdings eignet sich die Struktur des direkten azyklischen Graphen nur mit Einschränkungen zur Abbildung in einer relationale Datenbank.

Das Einlesen des GO-Graphen wird in GEPAT von einem OBO-Parser übernommen, der das OBO File einliest, die Struktur in einen Baum umwandelt, und ein Singleton-Objekt dieses Baums verwaltet, das die Beziehungen zwischen den GO-Einträgen enthält. Zusätzlich besitzt er noch die Informationen zu jedem GO-Eintrag aus der OBO-Datei. Dadurch, dass die GO-Informationen nur einmal angelegt werden, und danach im Speicher verbleiben, ist eine schnelle Durchsuchung und Analyse des GO-Graphen möglich.

3.10 KEGG Stoffwechselkarten

Die Gene Ontology liefert Informationen über den biologischen Hintergrund in den Gene involviert sind, verrät aber nichts über das Zusammenspiel dieser Gene. Ein wichtiger

Punkt in der Microarraydaten-Analyse ist jedoch die Identifizierung von Stoffwechselwege, an denen ausgezeichnete, meist differentiell exprimierte, Gene beteiligt sind. Die KEGG PATHWAY Datenbank [64] repräsentiert Netzwerke molekularer Interaktionen und Reaktionen in der Zelle auf graphische Art und Weise. Die verfügbaren Stoffwechselwege liefern Schlüsselinformationen über die funktionellen und metabolischen Abläufe in einer lebenden Zelle (siehe auch Kapitel 2.3.3). GEPAT verwendet diese Datenbanken, um Gene in den Stoffwechselwegen zu finden und um differentielle Genexpression farblich in den Stoffwechselwegen zu markieren, um so funktionelle Zusammenhänge zwischen ähnlich exprimierten Genen leichter erkennen zu können. Abbildung 44 zeigt diese Ansicht.

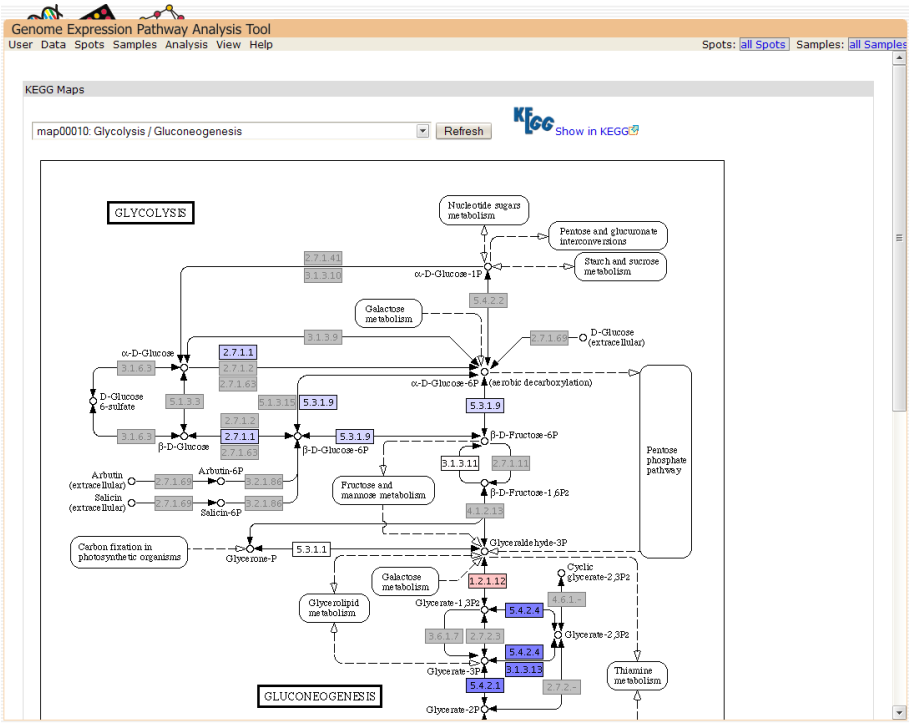


Abbildung 44: Ansicht einer KEGG-Stoffwechselkarte
Die Ergebnisse der Analyse auf differentielle Expression können auf die Enzyme und Gene der Karte überlagert werden.

Die enzymatische Aktivität, die durch EC-Nummern [161] in der Ensembl-Datenbank beschrieben wird, wird verwendet, um die Stoffwechselwege mit den Transkripten auf dem Microarray zu verknüpfen. Da eine enzymatische Aktivität durch mehr als ein Gen katalysiert werden kann, werden die Expressionswerte aller dieser Gene im Stoffwechselweg angezeigt. Falls mehrere Transkripte für ein Gen existieren, wird der

Median der Werte berechnet und zur Färbung verwendet. Um einen schnellen Überblick zu erhalten, welcher Stoffwechselweg welche Anzahl an Genen der aktuellen Arbeitsmenge enthält, kann eine sortierbare Übersichtstabelle angezeigt werden.

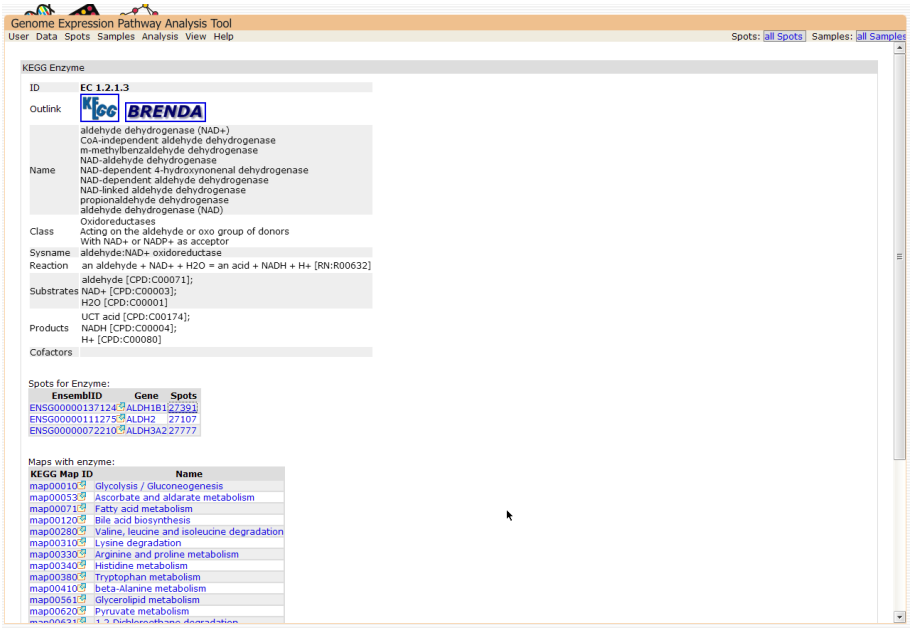


Abbildung 45: Detailinformationen zu einer enzymatischen Aktivität
Neben Verweisen zu weiteren Datenbanken und Informationen zum Enzym werden die Gene auf dem Microarray, die diese Aktivität katalysieren, angezeigt, sowie weitere Karten, auf denen dieses Enzym vorkommt.

Für jedes Gen auf dem Chip können alle KEGG Maps angezeigt werden, auf denen diese Gen vorkommt, sofern das Gen eine enzymatische Aktivität besitzt. Zu einer gegebenen Karte können auch alle zugehörigen Gene ermittelt werden, und zur weiteren Verarbeitung ausgewählt werden. Zu allen Enzymen und Genen, die in der Stoffwechselansicht angezeigt werden, können über einen Mausklick weitere Informationen eingeholt werden. Abbildung 45 zeigt die zu einem Enzym verfügbaren Informationen.

Die Informationen aus der KEGG-Datenbank können auch zur Teilmengenselektion verwendet werden. So ist es möglich, alle Transkripte, die in einer bestimmten Stoffwechselwegskarte vorkommen, als Teilmenge zu selektieren.

3.10.1 Datenbank

In den KEGG Karten können verschiedene Komponenten vorkommen, neben Enzymen auch einzelne Gene oder Moleküle. Ein Teil der Komponenten wurde zum schnelleren Zugriff in eine Datenbank gespeichert. Es wird unterschieden zwischen:

- Enzymen (Enzyme)
- Molekülen (Compound)

Zu den Enzymen und Produkten werden zusätzlich die KEGG-Stoffwechselkarten gespeichert, in denen diese vorkommen. Damit ist eine Suche nach zugehörigen Stoffwechselwegen für Enzym- oder Molekülnamen möglich. Zu einer enzymatisch katalysierten Reaktion werden die Ausgangsstoffe (Substrate), Ergebnisse (Products) und benötigte Zusatzstoffe (Kofaktoren) gespeichert.

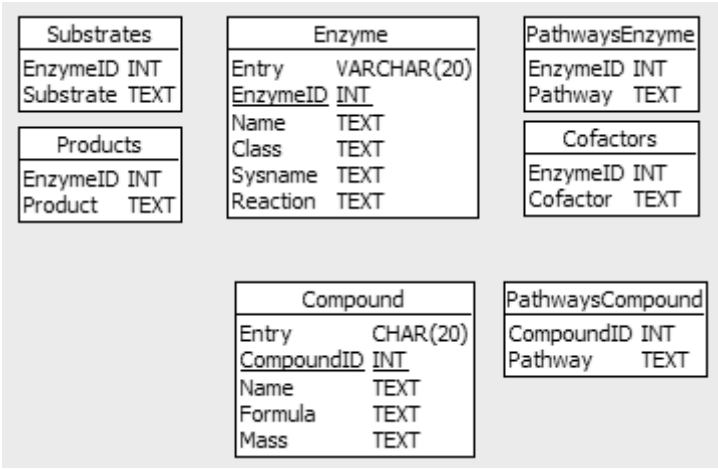


Abbildung 46: Die Struktur der Datenbank zur Speicherung der KEGG-Informationen
Aus der Vielzahl der in KEGG gespeicherten Informationen werden Informationen zu Enzymen und Molekülen verwendet, sowie die entsprechenden Substrate, Produkte, Cofaktoren und die KEGG-Karten, auf denen die Elemente vorkommen.

GEPAT verwendet zur Identifizierung von Genen zu den Enzymen auf den KEGG Maps die Ensembl-Datenbank, die Informationen zur enzymatischen Aktivität der Gene enthält. Gene, die direkt in der KEGG Map angegeben sind werden entweder organismusspezifisch gespeichert, oder es existiert ein KEGG Orthologieeintrag für verschiedene Organismen. Diese Orthologieeinträge können über UniGene-Kennungen, die in KEGG-Datenbank enthalten sind, in eine Ensembl-Genkennung umgewandelt werden. Abbildung 46 zeigt das Schema der Stoffwechselkarten-Datenbank von GEPAT, Abbildung 47 zeigt das Schema der Datenbank, die zur Identifikation von Genen

verwendet wird.

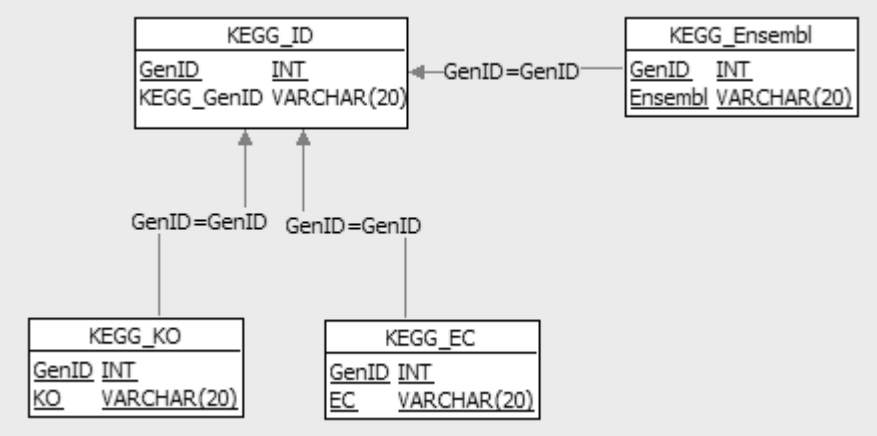


Abbildung 47: Die Struktur der zur Identifizierung der KEGG Gennamen verwendeten Datenbank
Zu einer KEGG-Genkennung wird eine Ensembl-Genkennung zugeordnet. KEGG-Orthologieeinträge und KEGG-Enzymeinträge verweisen auf KEGG Genkennungen.

Um die KEGG-Karten mit Genexpressionswerten oder den Werten der differentiellen Genexpression zu überlagern sind diejenigen Elemente interessant, die für ein Enzym oder eine orthologe Gruppe stehen. Mit der Möglichkeit, die Ensembl-Kennungen für Elemente aus KEGG-Karten zu ermittelt können mit den Informationen aus den KEGG-ML Dateien die Ergebnisse der Analyse auf differentielle Expression in das Bild von KEGG-Karten gezeichnet werden.

Um schnell ermitteln zu können, welche Ensembl-Einträge zu welcher EC-Nummer gehören, und welche EC-Nummern auf welchen KEGG-Stoffwechselkarten vorkommen, werden diese Daten beim Start des Servers ermittelt und anschließend als Singleton im Hauptspeicher gehalten.

3.11 STRING

KEGG Informationen sind nicht für alle Gene verfügbar, nicht jedes Gen ist Teil eines Stoffwechselwegs und nicht alle Stoffwechsel- und Signalwege sind in KEGG verfügbar. Informationen über funktionell relevante Proteininteraktionen sind jedoch essentiell, um das Verhalten der Zelle zu verstehen. Die STRING-Datenbank [162] liefert eine Übersicht über physische und funktionelle Assoziationen und Interaktionen zwischen Proteinen. Diese Assoziationen können zur Zusammenfassung in einem Netzwerk angezeigt werden,

in dem Gene als Knoten und verschiedene Arten von Assoziationen als Kanten dargestellt werden (siehe auch Kapitel 2.3.2).

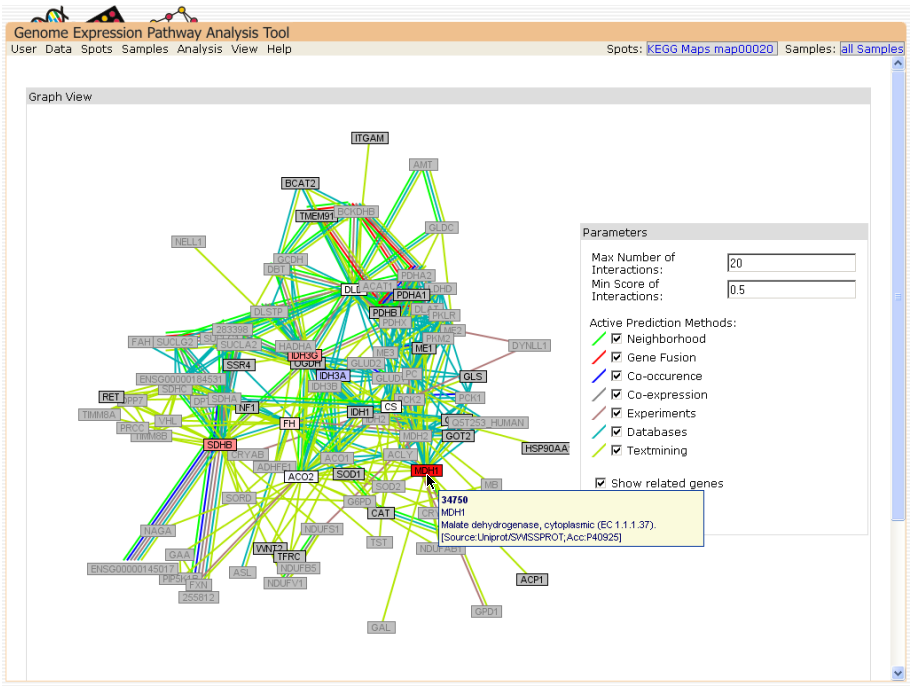


Abbildung 48: Die Assoziationsansicht
Gezeigt wird das Assoziationsnetzwerk für die auf einer KEGG-Karte vorkommenden Gene. Die unterschiedliche Farbe der Kanten steht für unterschiedliche Arten von Assoziationen zwischen den Genen. Durch die Auswahlmöglichkeiten auf der rechten Seite können die Art der Assoziationen, ihre Anzahl und Qualität eingeschränkt werden.

In GEPAT wurde diese Art der Ansicht übernommen. Eine lokale Version der STRING Datenbank kann auf dem Server installiert werden und eine Zuordnung von STRING Proteinkennungen zu den in GEPAT verwendeten Ensembl-Genkennungen wird vorgenommen. Mit diesen Informationen ist eine automatische Anzeige des STRING-Zusammenfassungsgraphen für eine Genmenge möglich. Die Ergebnisse der differentiellen Expressionsanalyse für Gene können auf die Knoten des Graphen übertragen werden, um so eine schnelle Übersicht über die Expression assoziierter Proteine zu erhalten. Falls mehr als ein Transkript für ein Gen existiert, wird hier der Medianwert zur Anzeige verwendet. Die Anzeige des Graphen kann durch eine Begrenzung auf bestimmte Assoziationen oder durch die Filterung der Assoziationen nach dem STRING-Assoziationsscore eingeschränkt werden. Für jeden Knoten des Graphen sind Tooltip-Informationen möglich, ein Mausklick auf den Knoten öffnet das Fenster der Genübersicht mit weiteren Informationen zum ausgewählten Gen. Die Interaktionssicht ist nicht für eine größere Anzahl an Genen geeignet, da in diesem Fall

kein vernünftiges Graphlayout erzeugt werden kann. Abbildung 48 zeigt ein STRING Assoziationsnetzwerk. Zur Erstellung des Interaktionsgraphen wird das JUNG-Framework [131] verwendet, das Layout des Graphen wird durch den Fruchterman-Reingold-Algorithmus [163] bestimmt.

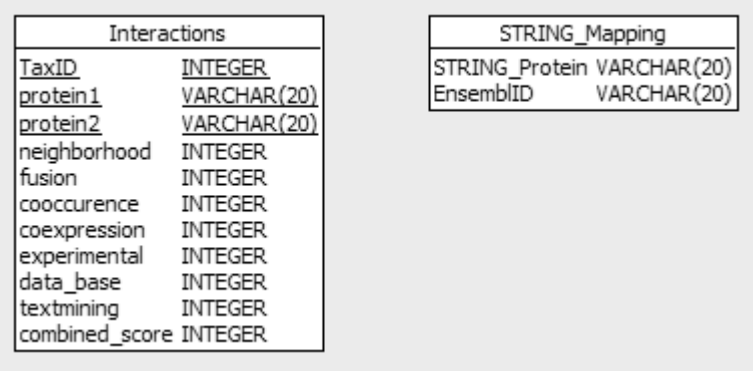


Abbildung 49: Struktur der Tabelle zur Speicherung der STRING Informationen
Die Tabelle Interactions ist im wesentlichen eine Abbildung der STRING Flatfile Datei, jedoch wurden hier die Interaktionen auf menschliche Gene begrenzt.

Die STRING-Daten können von akademischen Benutzern nach Registrierung bezogen werden. GEPAT benötigt zur Erstellung der Genassoziationsgraphen die Proteinnetzwerkdaten mit detaillierter Bewertung und die Aliase für STRING-Proteine, um die assoziierten Gene identifizieren zu können. STRING arbeitet auf Genebene und behandelt die Assoziationen aller Proteine eines Gens gleich, als Bezeichnung wird der Name des Proteins mit der längsten Aminosäurekette verwendet. Die Alias-Datei von STRING enthält für jeden Eintrag die NCBI Taxon-Kennung, die in der Datenbank verwendeten Proteinkennung, mögliche Aliase für diesen Eintrag und die Herkunft dieser Aliase. Zurzeit verwendet STRING Ensembl-Proteinnamen als Kennung in der Datenbank. Da sich diese Kennungen aber mit jedem Release der Ensembl-Datenbank ändern können, verwendet GEPAT die Alias-Datei zur Identifikation. GEPAT sucht zuerst nach Ensembl-Genkennung, die sich weniger schnell ändern als die Proteinkennung. Ist diese nicht Auffindbar, oder in der Ensembl-Datenbank nicht mehr vorhanden, versucht GEPAT der Reihe nach Uniprot, RefSeq, EMBL und LocusLink-Proteinkennungen zur Genidentifikation zu verwenden. Kann ein Gen zur STRING-Proteinkennung gefunden werden, so wird dieses Paar in der Mapping-Tabelle eingetragen.

Neben den Genassoziationen kann STRING auch Interaktionen für orthologe Gencluster (COG) anzeigen. Dieser Modus wird in GEPAT nicht verwendet. Abbildung 49 zeigt die Struktur der verwendeten Datenbank.

3.11.1 Assoziierte Gene

Die Assoziationsansicht ist nicht nur für eine Gruppe von Genen möglich. Auch die Genansicht für einzelne Gene ermöglicht es, die assoziierten Gene in einem Graphen anzuzeigen, der aus der STRING-Datenbank erzeugt wird. Um den Graph übersichtlich zu halten, kann die maximale Anzahl an Knoten eingeschränkt werden, indem die Art der Assoziationen, die Anzahl der Knoten und die Bewertung der Assoziationen beschränkt werden kann. Durch einen Mausklick auf einen der Knoten kann dieser als Mittelpunkt für einen neuen Graph ausgewählt werden. Abbildung 50 zeigt die Assoziationsansicht für ein Gen.

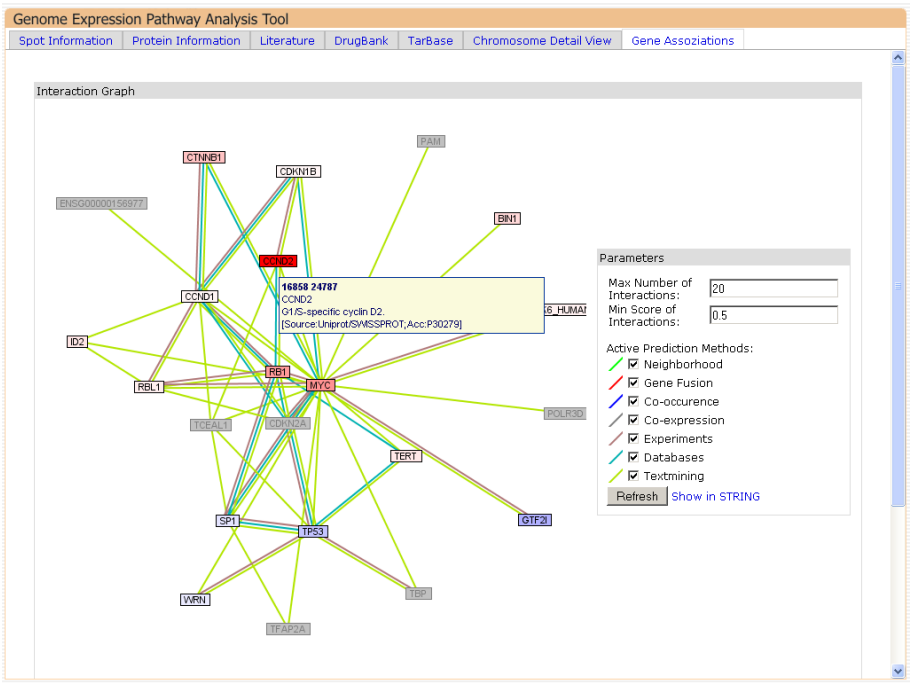


Abbildung 50: Assoziationsansicht für ein Gen
Im Gegensatz zur Ansicht für mehrere Gene wird diese Grafik als Modul der Geninformation erstellt und ist daher aus anderen Ansichten direkt aufrufbar.

3.11.2 Tarbase & Drugbank

In der Medizin sind vor allem diejenigen Gene in einer Menge ausgezeichneter Gene interessant, die zur Behandlung von Krankheiten verwendet werden können. Meist hofft man, die Proteinfunktion differentiell exprimierter Gene einschränken oder aktivieren zu können, um so bestimmte Stoffwechselwege zu beeinflussen. MicroRNAs können verwendet werden, um mRNA abzubauen und können so Überexpression von Genen entgegenwirken. Um hier mögliche Ansätze für eventuelle Behandlungsmethoden aufzuzeigen wurden in GEPAT Datenbanken integriert, die Informationen zu bestimmten Medikamenten und mRNAs und deren Zielgenen enthalten. Drugbank [164] ist eine Datenbank, die detaillierte Medikamentinformationen mit Informationen über die Ziele dieser Medikamente vereint. TarBase [165] enthält experimentell bestimmte microRNA-Ziele. Beschrieben werden die microRNA, die an das Ziel bindet, die Art von Hemmungen und Referenzen zu Experimenten, mit denen die microRNAs bestimmt wurden. GEPAT integriert diese Datenbanken und stellt einen Teil dieser Informationen in der Genübersicht zur Verfügung. So kann man schnell erkennen, welche Transkripte Ziele von Medikamenten oder microRNAs sind, und kann weitere Informationen über diese Gene abrufen. Zu jedem Medikament oder microRNA werden auch weitere Zielgene im Datensatz angezeigt.

4 Fallstudie

GEPAT wurde mit dem Ziel entwickelt, ein System zur Analyse der B-Zell Lymphomdaten von Rosenwald et al. [23] kombiniert mit Informationen über chromosomale Änderungen zu schaffen.

Die Analyse und Auswertung dieser Daten mit einer frühen Version von GEPAT wurde im Rahmen einer Bachelor-Abschlussarbeit durchgeführt [166]. Die nachfolgenden Grundlageninformationen sowie die Auswertung der differentiell exprimierten Gene sind dieser Arbeit entnommen.

4.1 Medizinische Grundlagen

4.1.1 B-Zell Lymphome

Tumore des lymphatischen Systems werden als Lymphome bezeichnet, da sie von den lymphatischen Organen wie Lymphknoten, Milz oder den lymphatischen T- und B-Zellen ausgehen. Maligne Lymphome sind Neoplasien des lymphatischen Gewebes, die von verschiedenen Entwicklungsstadien der Lymphozyten ausgehen und zytologisch den unterschiedlichen Differenzierungsformen reifer B- oder T-Lymphozyten ähneln.

Die malignen Lymphome werden aus historischen Gründen in die Gruppen der Hodgkin-Lymphome oder der Non-Hodgkin Lymphome eingeteilt, die Einteilung erfolgt durch histologische Untersuchung von Lymphknoten und Lymphgewebe. Der biologische Hintergrund dieser Unterteilung ist fragwürdig, aber wegen der unterschiedlichen Klinik und Therapie noch gerechtfertigt. Maligne Lymphome sind eher selten und machen zusammen etwa 3% aller Krebsfälle aus, wobei 40% Hodgkin-Lymphome und 60% Non-Hodgkin-Lymphome sind [167].

Da der Krankheitsverlauf auch bei histologisch ähnlichen Lymphomen individuell stark divergiert, wird eine Einteilung der verschiedenen Lymphomtypen unter anderem durch Untersuchung des Genexpressionsprofils durchgeführt [22].

4.1.2 Non-Hodgkin-Lymphome

- Die chronisch lymphatische Leukämie vom B-Typ ist eine Neoplasie nichtaktivierter, reif aussehender kleiner Lymphozyten, die meist leukämisch

ausgeschwemmt werden (B-CLL) und nur selten als lokalisierte Tumorerkrankung auffallen. Etwa 6% aller Non-Hodgkin-Lymphome gehören zu dieser Gruppe.

- Follikuläre Lymphome sind Neoplasien aus B-Zellen der Keimzentren mit follikulärem Wachstumsmuster. Sie machen etwa 25% aller Non-Hodgkin-Lymphome aus. Die kaum proliferativ aktiven Zellen sind apoptoseresistent und bilden die Struktur der Keimzentren (Follikel) nach. Bei ungefähr 40% der Patienten transformiert das follikuläre Lymphom zu einem diffusen großzelligen B-Zell Lymphom.
- Das Mantelzelllymphom ist eine Neoplasie naiver B-Zellen der Mantelzone des follikulären Keimzentrums. Mantelzelllymphome machen ungefähr 4% aller Non-Hodgkin-Lymphome aus.
- Extranodales Marginalzonen-B-Zell-Lymphom vom MALT-Typ entstehen durch Infektion mit Helicobacter pylori im Magen, die zu einer chronischen Gastritis führen. Im entzündlichen Infiltrat können auch Lymphfollikel gebildet werden, aus denen klonale Wucherungen hervorgehen können, deren Zellen einem postfollikulären Differenzierungsstadium von B-Lymphozyten entsprechen.
- Das Burkitt-Lymphom ist ein in westlichen Industrieländern sehr seltenes, in Äquatorialafrika sehr häufiges, hoch aggressives, sehr schnell wachsendes Lymphom.
- Das diffuse großzellige B-Zell-Lymphom ist die häufigste, heterogene Gruppe maligner Non-Hodgkin-Lymphome mit histologischem Aufbau aus großzelligen Tumorelementen und raschem, aggressivem Wachstum. Die Art von Lymphomen machen ungefähr 30% aller Non-Hodgkin-Lymphome aus.

4.1.3 Diffuses großzelliges B-Zell-Lymphom

Diese Lympome können de novo oder aufgrund sekundärer Chromosomabberationen aus vorbestehenden, primär niedrig malignen Lymphomen, wie follikulären Lymphom, CLL und Marginalzonen-B-Zell-Lymphom hervorgehen. Ungefähr 30% der Patienten weisen eine Translokation t(14,18) mit einer Umlagerung des BCL2 Gens auf. Diese Tumore entwickeln sich vermutlich aus follikulären Lymphomen [23].

Etwa ein Drittel der diffusen großzelligen B-Zell-Lymphome zeigen eine Umlagerung des

BCL6 Gens, das auf 3q27 lokalisiert ist und in vielen weiteren findet man eine Mutation in diesem Gen [168]. Regulatorische Elemente des BCL6 Gens sind beim diffusen großzelligen B-Zell-Lymphom häufig mutiert, was zu einer anhaltenden Expression von BCL6 führt. Ein Inhibitor-Peptid, das speziell an BCL6 bindet, verursacht in BCL6 positiven Lymphomzellen den Eintritt in die Apoptose und einen Stopp des Zellzyklus [169]. Translokationen der Region 15q11-13, die das BCL8 Gen enthält, werden mit vielen Partnern (14q32, 22q11, 9p13, 1p32, 7p13, 12q24, 15q22) beobachtet. BCL8 Expression wird in allen Patienten mit einer 15q11-13 Anomalie festgestellt. Im normalen Lymphgewebe wird BCL8 hingegen nicht exprimiert [170]. Des Weiteren werden gelegentlich aktivierende Mutationen im BRAF-Gen (7q34), einem Mitglieder der RAF-Gen-Familie, entdeckt. Der RAS-RAF-MEK-ERK-MAP Kinase Signalweg spielt eine entscheidende Rolle in der Zellproliferation und ist in Krebszellen häufig aktiviert [171].

Die Expression von Genen der Keimzentrums-Signatur (BCL6), der MHC-II-Signatur (HLA-DPα, HLA-DQα, HLA-DRα, HLA-DRβ) und der Lymphknoten-Signatur (α-Actinin, Collagen type III α 1, Connective-tissue growth factor, Fibronection, KIAA0233, Urokinase Plasminogen-Aktivator) lässt bei Patienten mit diffusem großzelligen B-Zell-Lymphom eine längere Überlebenszeit nach einer Chemotherapie erwarten. Dagegen ist die Expression von Genen der Proliferations-Signatur (c-myc, E2IG3, NPM3) und von BMP6 ein Hinweis auf eine kürzere Überlebenszeit [23].

Mutationen im Caspase-1ß-Gen, das auf 2q33 lokalisiert ist, wurden beim diffusen großzelligen B-Zell Lymphom, MALT-Lymphom, kleinzelligen lymphozytischen B-Zell-Lymphom (SLL) und follikulärem Lymphom gefunden. Nach Transfektion dieser Caspase-10-Mutanten in andere Zellen zeigten diese einen signifikanten Defekt in der Apoptose-Funktion [172].

CDKN2A (p16) und CDKN2B (p15) auf Chromosom 9p21 sind Inhibitoren von cyclin-abhänigigen Kinasen (CDK). Sie fungieren als Tumorsurpressor und sind bei Non-Hodgkin-Lymphomen auf aufgrund von Promotor-Hypermethylierungen, Deletionen und Mutationen stillgelegt. Sie inhibitieren die Phosphorylierung von Retinoblastom /pRb), das für die Freisetzung der E2F-Transkriptionsfaktoren notwendig ist. Die E2F-Transkriptionsfaktoren sind für das Fortschreiten des Zellzyklus in und durch die S-Phase erforderlich. [173]

Das BOB.1-Protein (POU2AF1) ist ein lymphozytenspezifischer transkriptioneller

Koaktivator. Es interagiert mit den Transkriptionsfaktoren OCT1 und OCT2 und trägt mit diesen zur transkriptionellen Aktivität von Oktamier-Motiven bei. Für Keimzentrumslymphome ist eine Überexpression von BOB.1 ein charakteristisches Kennzeichen, wobei bei diesen Tumoren auch eine hohe BCL6 Expression auftritt [174].

4.2 Analyse

B-Zell Lymphome stellen ein sehr heterogenes Krankheitsbild dar, bei dem verschiedene chromosomale Veränderungen und unterschiedliche Überexpressionen von Genen beobachtet werden.

4.2.1 IPI

Klinische Parameter, welche den Krankheitsgrad und Patientencharakteristiken beschreiben, können verwendet werden, um die Verlauf der Krankheit abzuschätzen. Ein prognostisches Modell wurde auf der Basis von 5 klinischen Parametern, die unabhängige Prädiktoren für den Krankheitsausgang sind, entwickelt [175]. Als prognostische Marker werden hier das Alter, das Stadium der Krankheit, extranodaler Befall, der Allgemeinzustand und die Höhe des Lactat-Dehydrogenasespiegels im Serum verwendet. Dieser als internationaler prognostischer Index (IPI) bekannter Wert bleibt trotz Fortschritten im Verständnis der biologischen Unterschiede der diffusen großzelligen B-Zell Lymphome das bestuntersuchte und erfolgreichste Werkzeug zur Vorhersage der Überlebenszeit von Patienten. Patienten, bei denen ein niedrige IPI-Wert festgestellt wird, haben eine 5-Jahres Überlebenswahrscheinlichkeit von 73%, vergleichen mit einer 26%igen Wahrscheinlichkeit der Gruppe mit hohem IPI-Wert.

4.2.2 Microarray-Untersuchungen

Eines der ersten Microarray-Experimente zur Untersuchung der Genexpression von diffusen großzelligen B-Zell Lymphomen wurden von Alizadeh et al. durchgeführt [22]. Hierfür wurde ein spezielles Microarray, der Lymphochip [176] entwickelt, der Sonden für Gene enthält, die in den Zellen des Lymphsystems exprimiert werden, und solche, die bekannte oder vermutete Rollen in der Immunologie und in Tumoren besitzen. Diese Microarrays wurden mit Proben von diffusen großzelligen B-Zell Lymphomen, follikulären Lymphomen und chronisch lymphatischer Leukämie hybridisiert. Für alle Arrays wurde für den zweiten Kanal eine gemeinsame Referenz aus einem Gemisch der mRNA von 9

verschiedenen Lymphom-Zelllinien verwendet. Durch hierarchisches Clustern wurden in der Gruppe der 42 Patienten mit diffusen großflächigem B-Zell Lymphom zwei Teilgruppen gefunden, von denen eine dem Expressionsmustern von Keimzentrums-B-Zellen (GCB) und die andere dem Expressionsmustern von aktivierten B-Zellen (ABC) ähnelt. Es zeigte sich, dass die Patientengruppen, die nach Einteilung in diese Teilmengen entstanden, einen signifikant geänderten Krankheitsverlauf besaßen, mit einer 76% 5-Jahres-Überlebenswahrscheinlichkeit der GCB-Gruppe, verglichen mit einer 16% 5-Jahres-Überlebenswahrscheinlichkeit der ABC-Gruppe, jeweils nach einer Anthracyclin-basierten Chemotherapie.

Diese Beobachtungen wurden in einer größeren Studie des Lymphom- und Leukämie Molekular-Profiling Projekts weiter bestätigt [23]. Hierfür wurden Proben von 240 Patienten mit DLBCL auf Lymphochip-Arrays hybridisiert. Durch hierarchisches Clustern konnte gezeigt werden, dass die t(14,18) Translokationen und die Amplifikation des cRel-locus nur in der Gruppe der Keimzentrums-B-Zellen ähnlichen Lymphome auftraten, die auch eine signifikant erhöhte Überlebenswahrscheinlichkeit nach der Chemotherapie besaß. Die 5-Jahres Überlebenswahrscheinlichkeit für Patienten der GCB-Gruppe betrug in dieser Studie 60%, die der Patienten aus der ABC-Gruppe 35%.

Ein Cox-Regressionsmodell wurde verwendet, um einzelne Gene zu finden, deren Expression mit der Überlebenswahrscheinlichkeit der Patienten korreliert. Zur Klassifizierung von Genen, die mit der Überlebenszeit korrelierten, wurde diese in Signaturen eingeteilt. Dabei ist eine Signatur eine Gruppe von Genen, die in speziellen Zelllinien, in verschiedenen Stadien der Differenzierung oder während einer bestimmten Art von biologischer Antwort exprimiert werden. Es zeigte sich, dass die Proliferationssignatur als bester Prädiktor für eine geringe Überlebenszeit verwendet werden konnte, während die Signaturen, die mit der Immunantwort im Zusammenhang standen als Prädiktor für eine erhöhte Überlebenszeit verwendet werden konnten. Unterschiede in der Keimzentrums-B-Zellen Signatur zeigten eine erhöhte Aktivität des NF-κB Signalwegs in der den aktivierten B-Zellen ähnlichen Teilgruppe. Aus den verschiedenen Signaturen wurden 17 Gene ausgewählt, welche die entsprechenden Signaturen repräsentieren und die zur Vorhersage der Gruppen verwendet werden können. Das zur Klassifizierung verwendete Verfahren ist jedoch nicht unumstritten, da das Modell aus einer großen Gruppe von 160 Patienten ausgewählt wurde und nur auf

einer kleine Gruppe von 80 Patienten validiert wurde [177][178].

Die Unterschiede zwischen den beiden Subgruppen werden in [179] genauer betrachtet, eine Übersicht über die vorhandenen Unterschiede liefert Tabelle 6. Verschiedene Arbeitsgruppen bestimmten weitere Biomarker, die zur Prognose des Krankheitsverlaufs verwendet werden können, aktuelle Übersichten enthalten [180] und [181].

Tabelle 6: Unterschiede der Gruppe GCB im Vergleich zu ABC
Molekulare, pathogenetische und klinische Besonderheiten, die Keimzentrums-B-Zellen ähnliche (GCB) und aktivierten B-Zellen ähnliche (ABC) diffuse großzellige B-Zell Lymphome unterscheiden.
Aus [179]

	Keimzentrums-B-Zellen ähnlich	*Aktivierten B-Zellen ähnlich*
Herkunftszelle	Keimzentrums B-Zelle	Möglicherweise reife B-Zelle
Fortschreitende Ig-Mutation	Ja	Nein
Onkogenetische Mechanismen	• BCL2 Translokation • Chr 2p Amplifikation des REL locus	Grundlegende Aktivierung von NF-κB
Intrazelluläre Signale	• cAMP moduliert AKT und pBAD • IL-4 induziert: -- pSTAT6 Akkumulation - Genexpression - Proliferation	• PDE4B inaktiviert cAMP • IL4 induziert: - keine Erhöhung von pSTAT6 - AKT Aktivierung - Arretierung des G0/G1 Zellzyklus
Klinische Ergebnisse	Günstige 60% 5-Jahres-Überlebenswahrscheinlichkeit	Schlechte 35% 5-Jahres Überlebenswahrscheinlichkeit

4.3 Analyse mit GEPAT

Im Rahmen des IZKF-Projekts B-36 der Universität Würzburg wurde der Datensatz von Rosenwald et al. erneut unter dem Gesichtspunkt einer integrierten Analyse von Genom-, Expressions- und Stoffwechselwegsdaten untersucht. Die Auswertung ist Gegenstand weiterer Arbeiten, im Folgenden soll gezeigt werden, wie die zur Analyse entwickelte Datenbank GEPAT eingesetzt werden kann.

Die Daten von Rosenwald et al. liegen als Zweikanal-Daten in einer durch Tabulatoren getrennten Tabelle vor. Zu jeder Gensonde auf dem Array befindet sich in dieser Tabelle eine Zeile, die einen eindeutigen Namen, einen Genbank-Kennung und die Werte des ersten und zweiten Kanals für jede biologische Probe enthält. Diese Daten wurden in GEPAT importiert, eine k Nearest Neighbor Missing Value Imputation wurde durchgeführt, anschließend wurden die Daten mit der vsn-Methode normalisiert und zentriert. Die Annotation erfolgte durch die in GEPAT integrierte Datenbank.

Von den 6515 Gensonden auf dem Chip konnte für 5541 ein Ensembl-Geneintrag identifiziert werden, da für einige Gene mehrere Sonden auf dem Chip vorhanden waren, sind die Messwerte für 2753 verschiedene Gene erfasst worden.

Aus diesen Genen wurde vor der Analyse entsprechend der Studie von Rosenwald et al. 4 Teilmengen ausgesucht, die in Tabelle 7 beschrieben werden. Die relativ große Proliferationssignatur besteht aus Genen, die direkt oder indirekt Einfluss auf die Proliferation der Zelle nehmen.

Tabelle 7: Gensonden-Teilmengen der Studie von Rosenwald et. al

Teilmenge	*Anzahl Sonden*	*Repräsentative Gene nach [23]*
Germinal Center Signature	138	BCL6, SERPINA9, GCET2
Lymph Node Signature	332	HLA-DPA1,HLA-DQA1, HLA-DRA, HLA-DRB1
MHC Class 2 Signature	36	ACTN1, COL3A1, PPBP, FN1, FAM38A, PLAU
Proliferation Signature	1181	MYC, GNL3, NPM3

4.3.1 Differentielle Expression

Die Gruppen ABC und GCB wurden zur biologischen Auswertung auf differentiell exprimierte Gene untersucht. Tabelle 8 und Tabelle 9 zeigen die Ergebnisse der Analyse auf differentielle Expression.

Tabelle 8: Ergebnisse der differentiellen Expression
Die Tabelle zeigt die Gene, die in der Gruppe ABC im Vergleich zur Gruppe GCB positiv exprimiert sind. Es werden nur Gensonden mit einem p-Wert < 10^{-7} angegeben, zu denen Genbeschreibungen verfügbar sind. Sind mehrere Sonden für ein Gen auf dem Array, wurde die Sonde mit dem niedrigsten p-Wert gewählt. Angegeben sind die Genbank-Kennung, der Genname, die Kurzbeschreibung aus Ensembl, sowie der Fold-Change Wert und der p-Wert des t-Tests.

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
N35315	ACY1	Aminoacylase-1 (EC 3.5.1.14) (N-acyl-L-amino-acid amidohydrolase) (ACY-1).	0.36	2.88E-14
AA761044	AFF3	AF4/FMR2 family member 3 (Protein LAF-4) (Lymphoid nuclear protein related to AF4).	1.16	3.09E-13
N63774	APEX1	DNA-(apurinic or apyrimidinic site) lyase (EC 4.2.99.18) (AP endonuclease 1) (APEX nuclease) (APEN) (REF-1 protein).	0.40	3.12E-10
W63749	BCL2	Apoptosis regulator Bcl-2.	1.25	2.23E-11
AA831869	BLNK	B-cell linker protein (Cytoplasmic adapter protein) (B-cell adapter containing SH2 domain protein) (B-cell adapter containing Src homology 2 domain protein) (Src homology 2 domain-containing leukocyte protein of 65 kDa) (SLP-65).	0.89	8.85E-20
AI027841	BMF	Bcl-2-modifying factor.	1.20	1.16E-19
AA828553	C13orf18	Isoform 3 of Q9H714	0.90	8.56E-14
N95059	CCDC50	Coiled-coil domain-containing protein 50 (Protein Ymer).	0.55	2.66E-11
AA831970	CCND2	G1/S-specific cyclin-D2.	1.30	2.13E-13
AA279047	CD44	CD44 antigen precursor (Phagocytic glycoprotein I) (PGP-1) (HUTCH-I) (Extracellular matrix receptor-III) (ECMR-III) (GP90 lymphocyte homing/adhesion receptor) (Hermes antigen) (Hyaluronate receptor) (Heparan sulfate proteoglycan) (Epican) (CDw44).	0.98	4.21E-13

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
AA455448	CD47	Leukocyte surface antigen CD47 precursor (Integrin-associated protein) (IAP) (Antigenic surface determinant protein OA3) (Protein MER6).	0.37	3.77E-9
AA284072	CDKN3	Cyclin-dependent kinase inhibitor 3 (EC 3.1.3.48) (EC 3.1.3.16) (CDK2- associated dual-specificity phosphatase) (Kinase-associated phosphatase) (Cyclin-dependent kinase-interacting protein 2) (Cyclin- dependent kinase interactor 1).	0.47	8.74E-9
AA002262	CFLAR	CASP8 and FADD-like apoptosis regulator precursor (Cellular FLICE-like inhibitory protein) (c-FLIP) (Caspase-eight-related protein) (Casper) (Caspase-like apoptosis regulatory protein) (CLARP) (MACH-related inducer of toxicity) (MRIT) (Caspase homolog)	0.56	1.27E-8
AK022569	EEF2K	Elongation factor 2 kinase (EC 2.7.11.20) (eEF-2 kinase) (eEF-2K) (Calcium/calmodulin-dependent eukaryotic elongation factor 2 kinase).	0.39	3.18E-10
AI382636	EML4	Echinoderm microtubule-associated protein-like 4 (EMAP-4) (Restrictedly overexpressed proliferation-associated protein) (Ropp 120).	0.40	7.51E-9
AA148098	ENTPD1	Ectonucleoside triphosphate diphosphohydrolase 1 (EC 3.6.1.5) (NTPDase1) (Ecto-ATP diphosphohydrolase) (ATPDase) (Lymphoid cell activation antigen) (Ecto-apyrase) (CD39 antigen).	1.07	2.43E-20
D79993	EPN4	Clathrin interactor 1 (Epsin-4) (Epsin-related protein) (EpsinR) (Enthoprotin) (Clathrin-interacting protein localized in the trans- Golgi region) (Clint).	0.51	7.99E-11
AA831368	ETV6	Transcription factor ETV6 (ETS-related protein Tel1) (Tel) (ETS translocation variant 6).	0.76	3.24E-16
AA744586	FAIM3	Fas apoptotic inhibitory molecule 3	0.98	7.22E-12
AA099570	FAM92A3	FAM92A3 protein.	0.59	1.24E-18
AI144309	FOXP1	Forkhead box protein P1.	1.28	1.52E-22
T77280	FUT8	Alpha-(1,6)-fucosyltransferase (EC 2.4.1.68) (Glycoprotein 6-alpha-L- fucosyltransferase) (GDP-fucose--glycoprotein fucosyltransferase) (GDP-L-Fuc:N-acetyl-beta-D-glucosaminide alpha1,6-fucosyltransferase) (alpha1-6FucT) (Fucosyltransferase 8).	1.15	3.79E-20
NM_014366	GNL3	Guanine nucleotide-binding protein-like 3 (Nucleolar GTP-binding protein 3) (Nucleostemin) (E2-induced gene 3-protein) (Novel nucleolar protein 47) (NNP47).	0.52	2.44E-10
H22856	GOT1	Aspartate aminotransferase, cytoplasmic (EC 2.6.1.1) (Transaminase A) (Glutamate oxaloacetate transaminase 1).	0.56	3.94E-8
AA487521	GOT2	Aspartate aminotransferase, mitochondrial precursor (EC 2.6.1.1) (Transaminase A) (Glutamate oxaloacetate transaminase 2).	0.64	6.95E-8
AA149097	HCK	Tyrosine-protein kinase HCK (EC 2.7.10.2) (p59-HCK/p60-HCK) (Hemopoietic cell kinase).	1.25	1.24E-15
AA702254	HLA-DOA	HLA class II histocompatibility antigen, DO alpha chain precursor (MHC class II antigen DOA) (MHC DZ alpha) (MHC DN-alpha).	0.56	8.08E-8
AA479188	ICAM3	Intercellular adhesion molecule 3 precursor (ICAM-3) (ICAM-R) (CDw50) (CD50 antigen).	0.57	3.38E-9
AA464139	IDH3A	Isocitrate dehydrogenase [NAD] subunit alpha, mitochondrial precursor (EC 1.1.1.41) (Isocitric dehydrogenase) (NAD(+)-specific ICDH).	0.39	7.09E-9
AA482203	IER5	Immediate early response gene 5 protein.	0.66	1.03E-9
AA729025	IKZF1	DNA-binding protein Ikaros (Lymphoid transcription factor LyF-1).	0.46	1.02E-10
R07094	IL10RA	Interleukin-10 receptor alpha chain precursor (IL-10R-A) (IL-10R1) (CDw210a antigen).	0.59	5.41E-11
AA293249	IL16	Interleukin-16 precursor (IL-16) (Lymphocyte chemoattractant factor) (LCF).	1.18	2.38E-21
AA416883	IRF2	Interferon regulatory factor 2 (IRF-2).	0.24	8.51E-8
AA743459	IRF4	Interferon regulatory factor 4 (IRF-4) (Lymphocyte-specific interferon regulatory factor) (LSIRF) (NF-EM5) (Multiple myeloma oncogene 1).	1.58	5.84E-27
AA766347	ITGAE	Integrin alpha-E precursor (Mucosal lymphocyte 1 antigen) (HML-1 antigen) (Integrin alpha-IEL) (CD103 antigen) [Contains: Integrin alpha-E light chain; Integrin alpha-E heavy chain].	0.48	6.43E-8
AA811374	KCNA3	Potassium voltage-gated channel subfamily A member 3 (Voltage-gated potassium channel subunit Kv1.3) (HPCN3) (HGK5) (HuKIII) (HLK3).	0.75	2.38E-11
N63398	LILRB4	Leukocyte immunoglobulin-like receptor subfamily B member 4 precursor (Leukocyte immunoglobulin-like receptor 5) (LIR-5) (Immunoglobulin- like transcript 3) (ILT-3) (Monocyte inhibitory receptor HM18) (CD85k antigen).	0.70	2.15E-8
AI356412	LYN	Tyrosine-protein kinase Lyn (EC 2.7.10.2).	0.56	3.81E-12

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
N25456	MCC	Colorectal mutant cancer protein (Protein MCC).	0.75	1.61E-14
H80623	MCM2	DNA replication licensing factor MCM2 (Minichromosome maintenance protein 2 homolog) (Nuclear protein BM28).	0.50	1.44E-8
AA054540	MYD88	Myeloid differentiation primary response protein MyD88.	0.41	8.70E-8
AA490263	NEK3	Serine/threonine-protein kinase Nek3 (EC 2.7.11.1) (NimA-related protein kinase 3) (HSPK 36).	0.51	2.44E-8
U90904	NIPA2	Non-imprinted in Prader-Willi/Angelman syndrome region protein 2.	0.55	1.11E-15
AA045090	NP	Purine nucleoside phosphorylase (EC 2.4.2.1) (Inosine phosphorylase) (PNP).	0.48	5.075E-10
AA789116	NP_066012.1	CDNA FLJ30993 fis, clone HLUNG1000064, weakly similar to KARYOGAMY PROTEIN KAR4 (KIAA1627 protein).	0.52	6.46E-10
AI475793	NP_115582.2	nucleotide-binding oligomerization domains 27	0.69	2.81E-8
AA764820	NP_689991.1	CDNA FLJ90083 fis, clone HEMBA1004982, weakly similar to TETRACYCLINE RESISTANCE PROTEIN, CLASS E.	0.44	6.71E-10
AA827665	NP_775815.2	B-cell novel protein 1	1.56	1.92E-15
AA854034	NR_002168.1	protein phosphatase 1, regulatory (inhibitor) subunit 2 pseudogene 3 (PPP1R2P3) on chromosome 5	0.45	5.99E-8
AB011139	OPA1	Dynamin-like 120 kDa protein, mitochondrial precursor (Optic atrophy 1 gene protein).	0.35	1.52E-8
N39081	P00973-2	2',5'-oligoadenylate synthetase 1 isoform 3	0.47	5.99E-8
AA490712	PAK2	Serine/threonine-protein kinase PAK 2 (EC 2.7.11.1) (p21-activated kinase 2) (PAK-2) (PAK65) (Gamma-PAK) (S6/H4 kinase).	0.36	4.40E-8
NM_014456	PDCD4	programmed cell death 4 isoform 1	0.73	4.66E-9
AA056219	PDE4B	cAMP-specific 3',5'-cyclic phosphodiesterase 4B (EC 3.1.4.17) (DPDE4) (PDE32).	0.66	2.91E-11
AA812195	PIM1	Proto-oncogene serine/threonine-protein kinase Pim-1 (EC 2.7.11.1).	0.99	2.91E-14
AA421212	PLK3	Serine/threonine-protein kinase PLK3 (EC 2.7.11.21) (Polo-like kinase 3) (PLK-3) (Cytokine-inducible serine/threonine-protein kinase) (FGF- inducible kinase) (Proliferation-related kinase).	0.46	5.78E-10
AA481464	PPIB	Peptidyl-prolyl cis-trans isomerase B precursor (EC 5.2.1.8) (PPIase) (Rotamase) (Cyclophilin B) (S-cyclophilin) (SCYLP) (CYP-S1).	0.38	6.44E-8
AA243358	PRKCB1	Protein kinase C beta type (EC 2.7.11.13) (PKC-beta) (PKC-B).	0.71	6.25-9
N55480	PRMT1	Protein arginine N-methyltransferase 1 (EC 2.1.1.-) (Interferon receptor 1-bound protein 4).	0.43	6.19-8
T54166	PSMB10	Proteasome subunit beta type 10 precursor (EC 3.4.25.1) (Proteasome MECl-1) (Macropain subunit MECl-1) (Multicatalytic endopeptidase complex subunit MECl-1).	0.45	1.11-8
AA252649	PTPN1	Tyrosine-protein phosphatase non-receptor type 1 (EC 3.1.3.48) (Protein-tyrosine phosphatase 1B) (PTP-1B).	1.19	1.65-10
AA477822	PTPN2	Tyrosine-protein phosphatase non-receptor type 2 (EC 3.1.3.48) (T-cell protein-tyrosine phosphatase) (TCPTP).	0.47	1.08E-11
AI358801	RAC2	Ras-related C3 botulinum toxin substrate 2 precursor (p21-Rac2) (Small G protein) (GX).	0.66	1.17E-12
N95176	RAP1A	Ras-related protein Rap-1A precursor (GTP-binding protein smg-p21A) (Ras-related protein Krev-1) (C21KG) (G-22K).	0.46	8.12E-8
AA057375	S100A4	Protein S100-A4 (S100 calcium-binding protein A4) (Metastasin) (Protein Mts1) (Placental calcium-binding protein) (Calvasculin).	0.81	3.85E-9
AA824616	SERTAD2	SERTA domain-containing protein 2 (Transcriptional regulator interacting with the PHD-bromodomain 2) (TRIP-Br2).	0.32	2.34E-9
AI538469	SH2D3C	SH2 domain-containing protein 3C (Novel SH2-containing protein 3).	0.53	3.75E-11
W37818	SH3BP5	SH3 domain-binding protein 5 (SH3 domain-binding protein that preferentially associates with BTK).	1.37	8.87E-24
AA284417	SLA	SRC-like-adapter (Src-like-adapter protein 1) (hSLAP).	1.25	4.54E-16

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
R26749	SP100	Nuclear autoantigen Sp-100 (Speckled 100 kDa) (Nuclear dot-associated Sp100 protein) (Lysp100b).	0.46	3.92E-9
H66484	SP140	Nuclear body protein SP140 (Nuclear autoantigen Sp-140) (Speckled 140 kDa) (LYSp100 protein) (Lymphoid-restricted homolog of Sp100).	0.78	1.34E-18
AA465158	SPIB	Transcription factor Spi-B.	0.64	2.27E-8
AF110647	SSR3	Translocon-associated protein subunit gamma (TRAP-gamma) (Signal sequence receptor subunit gamma) (SSR-gamma).	0.55	1.37E-12
AA765036	ST6GALNAC4	Alpha-N-acetyl-neuraminyl-2,3-beta-galactosyl-1,3-N-acetyl- galactosaminide alpha-2,6-sialyltransferase (EC 2.4.99.7) (NeuAc- alpha-2,3-Gal-beta-1,3-GalNAc-alpha-2,6-sialyltransferase) (ST6GalNAc IV) (Sialyltransferase 7D) (Sialyltransferase 3C).	0.52	1.34E-8
AA714029	TCEB3	Transcription elongation factor B polypeptide 3 (RNA polymerase II transcription factor SIII subunit A1) (SIII p110) (Elongin-A) (EloA) (Elongin 110 kDa subunit).	0.35	3.89E-8
AA807175	TGFBR2	TGF-beta receptor type-2 precursor (EC 2.7.11.30) (TGF-beta receptor type II) (TGFR-2) (TGF-beta type II receptor).	0.54	3.41E-8
AA504521	TLK1	Serine/threonine-protein kinase tousled-like 1 (EC 2.7.11.1) (Tousled- like kinase 1) (PKU-beta).	0.50	1.77E-13
AI380251	TRAM2	Translocation-associated membrane protein 2.	0.79	3.723E-13
AA649328	AA649328	Wiskott-Aldrich syndrome protein-interacting protein (WASP-interacting protein) (PRPL-2 protein).	0.61	1.51E-8
AB040918	ZNF406	Zinc finger protein 406 (Protein ZFAT).	0.75	6.39E-12

Tabelle 9: Ergebnisse der differentiellen Expression
Die Tabelle zeigt die Gene, die in der Gruppe ABC im Vergleich zur Gruppe GCB negativ exprimiert sind. Es werden nur Gensonden mit einem p-Wert < 10^{-7} angegeben, zu denen Genbeschreibungen verfügbar sind. Sind mehrere Sonden für ein Gen auf dem Array, wurde die Sonde mit dem niedrigsten p-Wert gewählt. Angegeben sind die Genbank-Kennung, der Genname, die Kurzbeschreibung aus Ensembl, sowie der Fold-Change Wert und der p-Wert des t-Tests.

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
AA769110	A4GALT	Lactosylceramide 4-alpha-galactosyltransferase (EC 2.4.1.228) (Alpha- 1,4-galactosyltransferase) (UDP-galactose:beta-D-galactosyl-beta1-R 4- alpha-D-galactosyltransferase) (Alpha-1,4-N- acetylglucosaminyltransferase) (Alpha4Gal-T1)	-0.83	8.67E-15
AA281781	BCL6	B-cell lymphoma 6 protein (BCL-6) (Zinc finger protein 51) (LAZ-3 protein) (BCL-5) (Zinc finger and BTB domain-containing protein 27).	-1.03	6.49E-18
W73473	BMP7	Bone morphogenetic protein 7 precursor (BMP-7) (Osteogenic protein 1) (OP-1) (Eptotermin alfa).	-0.66	2.834E-8
AA766198	BTNL9	Butyrophilin-like protein 9 precursor.	-1.09	3.31E-9
AF035296	C3orf37	UPF0361 protein DC12.	-0.89	1.44E-11
W88799	CASC1	cancer susceptibility candidate 1.	-1.37	3.58E-18
T65616	CCNG2	Cyclin-G2.	-0.62	1.56E-14
H69729	CD81	CD81 antigen (26 kDa cell surface protein TAPA-1) (Target of the antiproliferative antibody 1) (Tetraspanin-28) (Tspan-28).	-0.41	3.59E-8
AA482292	CR2	Complement receptor type 2 precursor (Cr2) (Complement C3d receptor) (Epstein-Barr virus receptor) (EBV receptor) (CD21 antigen).	-1.44	1.17E-10
AA287793	DAAM1	Disheveled-associated activator of morphogenesis 1.	-0.33	5.35E-8
AA741064	DENND3	DENN/MADD domain containing 3 (DENND3), mRNA	-0.71	2.34E-12
AA258552	DNAJC10	DnaJ (Hsp40) homolog, subfamily C, member 10	-0.59	3.42E-10
W33161	DNAJC12	DnaJ homolog subfamily C member 12 (J domain-containing protein 1).	-0.59	2.15E-8
H09055	DNMT1	DNA (cytosine-5)-methyltransferase 1 (EC 2.1.1.37) (Dnmt1) (DNA methyltransferase HsaI) (DNA MTase HsaI) (MCMT) (M.HsaI).	-0.42	1.01E-11

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
AA287913	DTX1	Protein deltex-1 (Deltex-1) (Deltex1) (hDTX1).	-0.77	9.80E-8
R99515	ELF1	ETS-related transcription factor Elf-1 (E74-like factor 1).	-0.30	4.71E-11
AA761617	ELOVL5	homolog of yeast long chain polyunsaturated fatty acid elongatio	-0.64	9.78-8
D87120	FAM3C	Protein FAM3C precursor (Protein GS3786).	-0.45	5.49E-9
AA827145	FEM1B	fem-1 homolog b.	-0.56	3.16E-10
AA825655	FGFR1OP2	FGFR1 oncogene partner 2.	-0.37	2.71E-10
AA480985	GCET2	germinal center expressed transcript 2 isoform 1.	-0.77	1.13E-8
AK025695	GSTZ1	Maleylacetoacetate isomerase (EC 5.2.1.2) (MAAI) (Glutathione S- transferase zeta 1) (EC 2.5.1.18) (GSTZ1-1).	-0.49	5.70E-9
AA122161	HDAC1	Histone deacetylase 1 (HD1).	-0.63	2.39E-16
AA687143	HMGN1	Nonhistone chromosomal protein HMG-14 (High-mobility group nucleosome- binding domain-containing protein 1).	-0.56	1.02E-9
N95053	HSD17B12	Estradiol 17-beta-dehydrogenase 12 (EC 1.1.1.62) (17-beta-HSD 12) (17- beta-hydroxysteroid dehydrogenase 12) (3-ketoacyl-CoA reductase) (EC 1.3.1.-) (KAR).	-0.46	1.79E-10
AA489064	ICOSLG	ICOS ligand precursor (B7 homolog 2) (B7-H2) (B7-like protein GI50) (B7-related protein 1) (B7RP-1) (CD275 antigen).	-0.75	9.46E-13
R68760	ITGA6	Integrin alpha-6 precursor (VLA-6) (CD49f antigen) [Contains: Integrin alpha-6 heavy chain; Integrin alpha-6 light chain].	-0.40	7.25E-8
AA832479	ITGB2	Integrin beta-2 precursor (Cell surface adhesion glycoproteins LFA- 1/CR3/p150,95 subunit beta) (Complement receptor C3 subunit beta) (CD18 antigen).	-0.92	2.18E-8
T84107	ITPKB	Inositol-trisphosphate 3-kinase B (EC 2.7.1.127) (Inositol 1,4,5- trisphosphate 3-kinase B) (IP3K B) (IP3 3-kinase B) (IP3K-B).	-0.93	1.26E-19
AA279650	KATNAL1	Katanin p60 ATPase-containing subunit A-like 1 (EC 3.6.4.3) (Katanin p60 subunit A-like 1) (p60 katanin-like 1).	-0.57	3.87E-8
AA482594	KLHL5	Kelch-like protein 5.	-0.60	5.43E-15
AA033713	KRR1	HIV-1 Rev-binding protein 2 (Rev-interacting protein 1) (Rip-1).	-0.45	8.56E-9
AA282059	LCK	Proto-oncogene tyrosine-protein kinase LCK (EC 2.7.10.2) (p56-LCK) (Lymphocyte cell-specific protein-tyrosine kinase) (LSK) (T cell- specific protein-tyrosine kinase).	-1.17	1.81E-13
AA280651	LMO2	Rhombotin-2 (Cysteine-rich protein TTG-2) (T-cell translocation protein 2) (LIM-only protein 2).	-1.76	3.83E-15
AA262888	LRMP	Lymphoid-restricted membrane protein (Protein Jaw1).	-1.38	1.65E-17
AA278881	MAN2C1	Alpha-mannosidase 2C1 (EC 3.2.1.24) (Alpha-D-mannoside mannohydrolase) (Mannosidase alpha class 2C member 1) (Alpha mannosidase 6A8B).	-1.52	5.41E-20
AA262140	MAP2K1	Dual specificity mitogen-activated protein kinase kinase 1 (EC 2.7.12.2) (MAP kinase kinase 1) (MAPKK 1) (ERK activator kinase 1) (MAPK/ERK kinase 1) (MEK1).	-0.42	5.07E-9
R39221	MAPK10	Mitogen-activated protein kinase 10 (EC 2.7.11.24) (Stress-activated protein kinase JNK3) (c-Jun N-terminal kinase 3) (MAP kinase p49 3F12).	-1.35	1.31E-15
AA805279	MAST2	Microtubule-associated serine/threonine-protein kinase 2 (EC 2.7.11.1).	-0.58	1.23E-9
AA099155	MEF2C	Myocyte-specific enhancer factor 2C.	-0.73	4.56E-8
AA744607	MFHAS1	malignant fibrous histiocytoma amplified sequence 1.	-0.65	1.99E-9
AA287043	MME	Neprilysin (EC 3.4.24.11) (Neutral endopeptidase) (NEP) (Enkephalinase) (Neutral endopeptidase 24.11) (Atriopeptidase) (Common acute lymphocytic leukemia antigen) (CALLA) (CD10 antigen).	-1.43	4.65E-23
AA279337	MOBKL2A	Mps one binder kinase activator-like 2A (Mob1 homolog 2A) (MOB-LAK) (Protein Mob3A).	-0.43	1.39E-8
AA768961	MYBL1	Myb-related protein A (A-Myb).	-1.76	6.25E-28
AA648536	MYO1E	Myosin Ie (Myosin Ic).	-0.67	1.27E-10
N47107	NEK6	Serine/threonine-protein kinase Nek6 (EC 2.7.11.1) (NimA-related protein kinase 6) (Protein kinase SID6-1512).	-0.32	2.52E-8
AA768888	NP_00100 1695	CDNA FLJ42418 fis, clone BLADE2001987.	-1.12	3.46E-12

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
AA279019	NP_861450.1	LOC283537 protein (OTTHUMP00000018184).	-0.29	9.82E-9
M14963	OAT	Ornithine aminotransferase, mitochondrial precursor (EC 2.6.1.13) (Ornithine--oxo-acid aminotransferase) [Contains: Ornithine aminotransferase, hepatic form; Ornithine aminotransferase, renal form].	-0.66	9.85E-12
AB014604	OSBPL3	Oxysterol-binding protein-related protein 3 (OSBP-related protein 3) (ORP-3).	-0.70	3.55E-11
AA831773	PAG1	Phosphoprotein associated with glycosphingolipid-enriched microdomains 1 (Transmembrane adapter protein PAG) (Csk-binding protein) (Transmembrane phosphoprotein Cbp).	-0.73	4.65E-17
AA281973	PARP1	Poly [ADP-ribose] polymerase 1 (EC 2.4.2.30) (PARP-1) (ADPRT) (NAD(+) ADP-ribosyltransferase 1) (Poly[ADP-ribose] synthetase 1).	-0.40	2.88E-8
AA279342	PCDHGC5	Protocadherin gamma A12 precursor (PCDH-gamma-A12) (Cadherin-21) (Fibroblast cadherin 3).	-0.61	3.73E-11
AI143449	PDK3	[Pyruvate dehydrogenase [lipoamide]] kinase isozyme 3, mitochondrial precursor (EC 2.7.11.2) (Pyruvate dehydrogenase kinase isoform 3).	-0.54	3.41E-9
N69643	PIK3R1	Phosphatidylinositol 3-kinase regulatory subunit alpha (PI3-kinase p85-subunit alpha) (PtdIns-3-kinase p85-alpha) (PI3K).	-0.29	3.13E-8
W89193	PRKAB1	5'-AMP-activated protein kinase subunit beta-1 (AMPK beta-1 chain) (AMPKb).	-0.55	1.66E-16
R55802	PTK2	Focal adhesion kinase 1 (EC 2.7.10.2) (FADK 1) (pp125FAK) (Protein- tyrosine kinase 2).	-0.91	2.24E-14
AA687472	RAPGEF5	Rap guanine nucleotide exchange factor 5 (Guanine nucleotide exchange factor for Rap1) (Related to Epac) (Repac) (M-Ras-regulated Rap GEF) (MR-GEF).	-0.82	9.72E-11
AA279919	REL	C-Rel proto-oncogene protein (C-Rel protein).	-0.59	2.28E-9
AA505184	RGS13	Regulator of G-protein signaling 13 (RGS13).	-0.45	3.11E-10
AA029960	RGS16	Regulator of G-protein signaling 16 (RGS16) (Retinally abundant regulator of G-protein signaling) (RGS-R) (A28-RGS14P).	-0.76	2.37E-9
AI522007	RRM2B	Ribonucleoside-diphosphate reductase M2 subunit B (EC 1.17.4.1) (TP53- inducible ribonucleotide reductase M2 B) (p53-inducible ribonucleotide reductase small subunit 2-like protein) (p53R2).	-0.44	2.01E-9
AA211820	RUNX2	Runt-related transcription factor 2 (Core-binding factor, alpha 1 subunit) (CBF-alpha 1) (Acute myeloid leukemia 3 protein) (Oncogene AML-3) (Polyomavirus enhancer-binding protein 2 alpha A subunit) (PEBP2-alpha A) (PEA2-alpha A) (SL3-3 enhancer factor 1	-0.54	1.45E-11
AA805575	SERPINA9	Serpin A9 precursor (Germinal center B-cell expressed transcript 1 protein).	-2.61	1.00E-24
AA283000	SFRS15	Splicing factor, arginine/serine-rich 15 (CTD-binding SR-like protein RA4).	-0.48	5.9E-8
AA040856	SH3KBP1	SH3-domain kinase-binding protein 1 (Cbl-interacting protein of 85 kDa) (Human Src-family kinase-binding protein 1) (HSB-1) (CD2-binding protein 3) (CD2BP3).	-0.57	1.0564392E-9
AA458996	SLAMF1	Signaling lymphocytic activation molecule precursor (IPO-3) (CD150 antigen) (CDw150).	-0.88	6.33E-10
AA768213	SLC1A1	Excitatory amino acid transporter 3 (Sodium-dependent glutamate/aspartate transporter 3) (Excitatory amino-acid carrier 1) (Neuronal and epithelial glutamate transporter).	-0.57	3.64E-8
NM_001152	SLC25A5	ADP/ATP translocase 2 (Adenine nucleotide translocator 2) (ANT 2) (ADP,ATP carrier protein 2) (Solute carrier family 25 member 5) (ADP,ATP carrier protein, fibroblast isoform).	-0.37	5.15E-9
AA278443	SPI1	Transcription factor PU.1 (31 kDa transforming protein).	-0.32	4.66E-8
AA215573	STAG3	Cohesin subunit SA-3 (Stromal antigen 3) (Stromalin 3) (SCC3 homolog 3).	-0.93	9.59E-11
AA760861	STAP1_HUMAN	Signal-transducing adaptor protein 1 (STAP-1) (Stem cell adaptor protein 1) (BCR downstream signaling protein 1) (Docking protein BRDG1).	-0.87	1.73E-12
AA250954	STK17A	Serine/threonine-protein kinase 17A (EC 2.7.11.1) (DAP kinase-related apoptosis-inducing protein kinase 1).	-0.62	1.65E-10
AA828335	SULT1A4	Monoamine-sulfating phenol sulfotransferase (EC 2.8.2.1) (Aryl sulfotransferase 1A3) (Sulfotransferase, monoamine-preferring) (M-PST) (Thermolabile phenol sulfotransferase) (TL-PST) (Placental estrogen sulfotransferase) (Catecholamine-sulfating phenol sul	-1.16	1.85E-15

Genbank	*Genname*	*Beschreibung*	*FC*	*p-Wert*
W31857	SWP70_HUMAN	Switch-associated protein 70 (SWAP-70).	-0.61	2.28E-8
AF082557	TNKS	Tankyrase-1 (EC 2.4.2.30) (TANK1) (Tankyrase I) (TNKS-1) (TRF1- interacting ankyrin-related ADP-ribose polymerase).	-0.56	4.93E-14
AA278411	TOX_HUMAN	Thymus high mobility group box protein TOX.	-0.98	1.05E-11
Z29328	UBE2H	Ubiquitin-conjugating enzyme E2 H (EC 6.3.2.19) (Ubiquitin-protein ligase H) (Ubiquitin carrier protein H) (UbcH2) (E2-20K).	-0.33	2.623E-8
R33016	UPP1	Uridine phosphorylase 1 (EC 2.4.2.3) (UrdPase 1) (UPase 1).	-0.66	1.68E-9
AA045285	VCL	Vinculin (Metavinculin). [Source:Uniprot/SWISSPROT;Acc:P18206]	-0.68	2.37E-11
AA769424	VNN2	Vascular non-inflammatory molecule 2 precursor (Vanin-2) (Glycosylphosphatidyl inositol-anchored protein GPI-80) (Protein FOAP- 4).	-0.81	4.59E-11
AA743090	VPREB3	Pre-B lymphocyte protein 3 precursor (VpreB3 protein) (N27C7-2).	-1.15	3.16E-10
R80974	WEE1	Wee1-like protein kinase (EC 2.7.10.2) (Wee1A kinase) (WEE1hu).	-0.69	9.23E-9
AA649066	ZCCHC10	Zinc finger CCHC domain-containing protein 10.	-0.36	9.12E-10
AL117587	ZNF608	zinc finger protein 608	-0.91	5.08E-11

4.3.2 GO Analyse

Um die Funktion dieser differentiell exprimierten Gene zu deuten, kann deren GO-Annotation verwendet werden. Die relative Anzahl der GO-Kategorien in der Kategorie der biologischen Prozesse verteilt sich wie in Abbildung 51 gezeigt. Abbildung 52 zeigt die relative Anzahl in der Kategorie der zellulären Komponenten, während Abbildung 53 die relative Anzahl in der Gruppe der molekularen Funktionen zeigt. Diese Zahlen müssen jedoch unter dem Gesichtspunkt betrachtet werden, dass bei der Auswahl der Gensonden des verwendeten Lymphochips ein Schwerpunkt auf lymphomtypische Gene gelegt wurde. Andere Gene kommen nur unterrepräsentiert auf dem Microarray vor.

Um die GO-Kategorien sinnvoll interpretieren zu können, ist es allerdings notwendig, die Anzahl der Gene in einer Kategorie mit einer Anzahl zufällig ausgewählter Gene zu vergleichen. Kommen mehr Gene vor, als eine zufällige Auswahl erwarten lässt, so kann eine biologischen Bedeutung dieser Gene vermutet werden Diese GO-Term Enrichment Analysis wurde auf den Gene, deren Expression sich in den Gruppen ABC und GBC unterscheidet, durchgeführt. Abbildung 54 zeigt die Ergebnisse dieser Analyse. Hier zeigt sich, dass Gene, die eine Transferase-Aktivität besitzen, häufiger in den differentiell exprimierten Genen vorkommen. Weitere Kategorien mit einer signifikant erhöhten Anzahl an Genen sind die Regulation der Signalweiterleitung, die Regulation der Immunantwort und die Aktivierung der Lymphozyten.

GO:0008150 biological_process (147)
- GO:0007610 behavior (5%)
- GO:0009987 cellular process (95%)
  - GO:0007154 cell communication (39%)
  - GO:0030154 cell differentiation (12%)
  - GO:0050875 cellular physiological process (84%)
  - GO:0050794 regulation of cellular process (44%)
- GO:0007275 development (25%)
- GO:0007582 physiological process (90%)
  - GO:0050875 cellular physiological process (84%)
  - GO:0016265 death (11%)
  - GO:0051179 localization (12%)
  - GO:0008152 metabolism (64%)
  - GO:0050874 organismal physiological process (26%)
  - GO:0050791 regulation of physiological process (39%)
  - GO:0050896 response to stimulus (31%)
- GO:0050789 regulation of biological process (45%)

Abbildung 51: Differentiell exprimierte Gene in biologischen Prozessen
Relative Anzahl der differentiell exprimierten Gene (p-Wert < 10^{-7}) in der Kategorie der biologischen Prozesse

GO:0005575 cellular_component (133)
- GO:0005623 cell (97%)
  - GO:0005622 intracellular (72%)
    - GO:0005737 cytoplasm (35%)
    - GO:0043229 intracellular organelle (62%)
    - GO:0005634 nucleus (40%)
  - GO:0016020 membrane (43%)
    - GO:0031224 intrinsic to membrane (30%)
    - GO:0005886 plasma membrane (18%)
- GO:0043226 organelle (62%)
- GO:0043234 protein complex (12%)

Abbildung 52: Differentiell exprimierte Gene in zellulären Komponenten
Relative Anzahl der differentiell exprimierten Gene (p-Wert < 10^{-7}) in der Kategorie der zellulären Komponenten

GO:0003674 molecular_function (152)
- GO:0005488 binding (80%)
  - GO:0043167 ion binding (22%)
  - GO:0003676 nucleic acid binding (22%)
  - GO:0005515 protein binding (53%)
- GO:0005102 receptor binding (7%)
- GO:0003824 catalytic activity (45%)
  - GO:0016787 hydrolase activity (14%)
  - GO:0016740 transferase activity (29%)
- GO:0004871 signal transducer activity (24%)
- GO:0030528 transcription regulator activity (24%)

Abbildung 53: Differentiell exprimierte Gene in molekularen Funktionen
Relative Anzahl der differentiell exprimierten Gene (p-Wert < 10^{-7}) in der Kategorie der molekularen Funktionen

GO:0003674 molecular_function (O:152 E:152.0 R:1.0 P:1.0)
- GO:0003824 catalytic activity (O:69 E:63.57 R:1.08 P:0.446)
  - GO:0016740 transferase activity (O:44 E:26.69 R:1.64 P:0.0)

GO:0050789 regulation of biological process (O:66 E:62.11 R:1.06 P:0.612)
- GO:0050794 regulation of cellular process (O:64 E:58.6 R:1.09 P:0.437)
  - GO:0009966 regulation of signal transduction (O:15 E:7.93 R:1.89 P:0.028)

GO:0050789 regulation of biological process (O:66 E:62.11 R:1.06 P:0.612)
- GO:0050791 regulation of physiological process (O:58 E:55.74 R:1.04 P:0.817)
  - GO:0051239 regulation of organismal physiological process (O:10 E:5.85 R:1.7 P:0.161)
    - GO:0050776 regulation of immune response (O:10 E:4.68 R:2.13 P:0.042)

GO:0008150 biological_process (O:147 E:147.0 R:1.0 P:1.0)
- GO:0007582 physiological process (O:133 E:136.2 R:0.97 P:0.301)
  - GO:0050874 organismal physiological process (O:38 E:31.09 R:1.22 P:0.213)
    - GO:0001775 cell activation (O:12 E:6.76 R:1.77 P:0.087)
      - GO:0045321 immune cell activation (O:12 E:6.76 R:1.77 P:0.087)
        - GO:0046649 lymphocyte activation (O:12 E:6.11 R:1.96 P:0.043)

Abbildung 54: Über/Unterrepräsentierte Gene in der Gruppe der differentiell exprimierten Gene verglichen mit allen Genen auf dem Microarray
Dargestellt werden die Zahl der beobachteten Gene (O), die Zahl der erwarteten Gene (E), das Verhältnis (R) und der P-Wert. Da der komplette Baum zu umfangreich ist, wird nur eine Auswahl angezeigt. Da die Kategorien mehrfach im Baum vorkommen, wurde nur ein Pfad angegeben. Es zeigt sich, dass Gene aus Kategorien, die Steuerungsfunktionen beschreiben, gehäuft differentiell exprimiert sind.

4.3.3 CGH Analyse

Zusätzlich zu den Genexpressionsdaten stehen zu den Patientendaten noch Informationen über chromosomale Änderungen zur Verfügung. In einer Studie wurde das chromosomale Profil von 224 Patienten des Datensatzes von Rosenwald et al. untersucht [24]: 164 der 224 Patienten zeigten Änderungen im chromosomalen Profil, wobei die Anzahl keine statistisch signifikanten Unterschiede zwischen der ABC und GCB Subgruppe zeigte. Waren chromosomale Änderungen vorhanden, so trat in 87% der Fälle mehr als eine Änderung auf.

Bemerkenswert ist jedoch, dass die Verteilung der chromosomalen Änderungen Unterschiede über die unterschiedlichen Subgruppen aufweist. Patienten aus der ABC-Gruppe zeigten häufig einen Gewinn des kompletten 3q-Chromosomarms (26%) oder eine Trisomie 3 (15%). Diese Ereignisse wurden jedoch nicht in der GCB-Subgruppe oder in den unklassifizierten Patienten gefunden. 42% der ABC-Patienten zeigten einen

Gewinn von 18q21-22, verglichen mit 10% aus der GCB-Gruppe. Auch eine Vervielfachung der Chromosomregion 18q21, die das BCL2-Gen enthält, wurde in 18% der ABC-Patienten, aber nur in 5% der GCB Patienten festgestellt. Bereits in anderen Studien [182][183] wurde gezeigt, dass die t(14,18) Translokation, die das BCL2-Gen umfasst, nur in der GCB-Subgruppe stattfindet (46% und 53% der Fälle), jedoch nie in der ABC-Gruppe. Eine hohe Expression von BCL2 ist eine charakteristische Eigenschaft der ABC-Subgruppe, kommt jedoch nur in GCB-Patienten vor, welche die t(14,18) Translokation besitzen. Dies lässt vermuten, dass eine Vervielfachung der 18q21-Region hauptsächlich in den Lymphomen vorkommt, die das BCL2-Gen ablesen können. GCB-Patienten zeigen in 21% der Fälle Zugewinne von 12q12, verglichen mit 5% der ABC-Patienten. Unklar ist jedoch, ob die chromosomale Änderungen für die Entstehung bestimmter Tumortypen verantwortlich sind, oder erst in diesen Tumortypen entstehen. Auch GEPAT bietet die Möglichkeit, chromosomale Änderungen von zwei Probengruppen miteinander zu vergleichen. Abbildung 55 zeigt eine Übersicht über diese Analyseergebnisse.

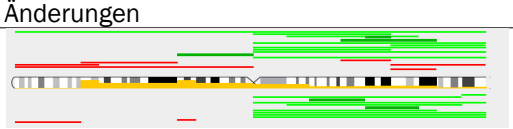
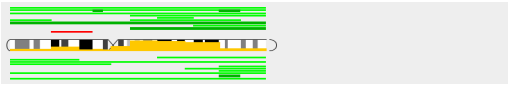
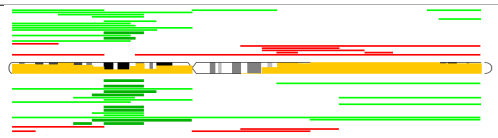
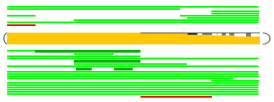
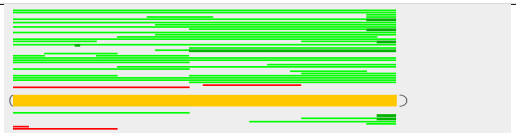
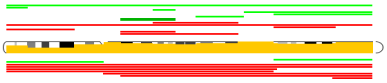
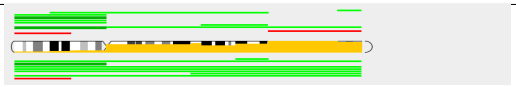
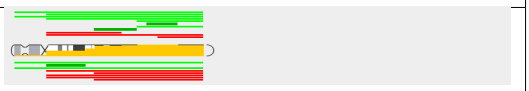
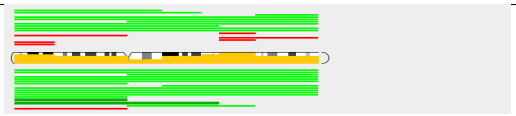
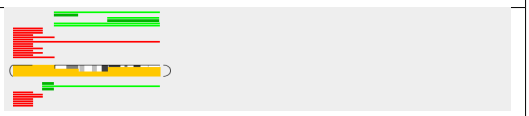
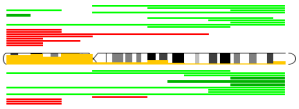
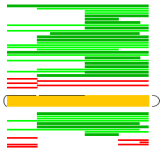
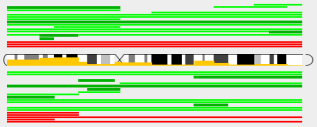
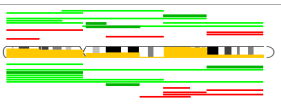
Chr	Änderungen	Chr	Änderungen
1		11	
2		12	
3		13	
4		14	
5		15	
6		16	
7		17	
8		18	
9		X	
10			

Abbildung 55: Unterschiede in den CGH-Daten
Die Gruppe ABC wird über den Chromosomen dargestellt, die Gruppe GCB darunter. Auffällige Unterschiede finden sich auf Chromosom 3 sowie auf Teilen von Chromosom 12 und 18. Der gelbe Balken auf dem Chromosom zeigt den p-Wert des Wilcoxon-Rang Tests. Grüne Balken stehen für vervielfachte, rote Balken für verlorene Chromosombereiche.

5 Diskussion

Trotz der Verfügbarkeit vieler Programme zur Microarray-Datenanalyse gibt es fast keine Programme, die eine integrierte Analyse und Interpretation ermöglichen. Um die Effekte der differentiellen Expression zu verstehen ist es nicht ausreichend, einzelne Gene isoliert zu betrachten. Vielmehr ist es notwendig, die Resultate im Kontext des zellulären Netzwerks zu betrachten. Mit der Einbeziehung von Stoffwechsel- oder Signalwegen in die Analyse differentiell exprimierter Gene können die Auswirkung der Genexpression auf die Zellbedingungen besser verstanden werden.

GEPAT ermöglicht die Durchführung des Microarray-Datenanalyse und -Interpretationsprozesses in einem Programm. Es integriert die Analyse- und Interpretationsmöglichkeiten der Daten in einem durchgängigen Konzept. Ein modulares System zum Auswählen von Transkript- und Probenteilmengen erlaubt es, Resultate von Analyse und Interpretation als Start für neue Analysen und Interpretationen zu verwenden. Dies erlaubt das einfache Überprüfen von bestehenden Hypothesen und unterstützt die Bildung neuer Hypothesen aus den Resultaten. Die integrierten Fähigkeiten und die Annotationsmöglichkeiten für menschliche Microarray-Daten machen GEPAT zu einem mächtigen Werkzeug zur Microarray-Datenanalyse.

Der schnelle Fortschritt in der biologischen Forschung macht es notwendig, für neue Techniken offen zu sein. Große Teile von GEPAT sind daher modular implementiert, um bestehende Techniken zu erweitern oder neue Techniken hinzufügen zu können. Datenverwaltungsfunktionen dienen als Gerüst, das mit verschiedensten Modulen zum Datenimport, zur Datenanalyse, Dateninterpretation, Teilmengenselektion und zur Geninformation erweitert werden kann. Da damit nahezu jede Analysemethode in das Framework implementiert werden kann, ist ein weiteres Wachstum des Open-Source-Systems in Zukunft möglich. Module, die sich Ziele von microRNAs und die Medikamententwicklung fokussieren, sind derzeit in Entwicklung. Neue Module profitieren von der Funktionalität, die von anderen Modulen bereitgestellt wird. So kann jedes Modul auf die Analyseergebnisse zugreifen, oder sich auf die während des Imports erfolgte Annotation der Gensonden beziehen.

Zur Analyse von Genexpressionsdaten existieren auch eine Reihe kommerzieller Pakete, wie z.B. J-Express [184], GeneSifter [185] oder GeneSpring [186]. Obwohl alle diese

Produkte Teile der Analysemethoden beinhalten, die auch in GEPAT verwendet werden und noch darüber hinausgehende Möglichkeiten bieten, wurden während der Evaluierung einige Nachteile sichtbar. Die meisten kommerziellen Pakete ermöglichen zwar eine Programmerweiterung über festdefinierte Schnittstellen, jedoch sind diese Möglichkeiten begrenzt, sobald komplexere Funktionalität und zusätzliche Daten integriert werden sollen. Auch ist es nicht möglich, die Erweiterungen anschließend ohne das kommerzielle Programmpaket benutzen zu können.

Die akademisch frei verfügbaren Softwarepakete bieten zwar herausragende Analysemethoden, sind aber weit von dem entfernt, was als intuitiv bedienbar bezeichnet werden kann, oder sind in Programmiersprachen wie Perl oder PHP implementiert, die eine Erweiterung deutlich erschweren. Deshalb wurde die Benutzeroberfläche und Datenhaltung von GEPAT unabhängig von bestehenden Systemen entwickelt, zur Datenanalyse wurden die fortgeschrittenen Analysemodule von Bioconductor verwendet. Zur Implementierung von GEPAT wurde die Programmiersprache Java mit der Erweiterung Java Enterprise Edition gewählt, die neben .NET von Microsoft die führende Plattform für professionelle Internetapplikationen darstellt und breite Verwendung findet.

Für die Implementierung von GEPAT zur Verwaltung und Analyse von Daten haben sich zwei mögliche Arten von Programmentwürfen angeboten: Eine Möglichkeit ist eine eigenständige Desktopanwendung, bei der Benutzerschnittstelle, Programmlogik und Datenhaltung auf dem Rechner des Benutzers ablaufen. Die Daten werden hierbei auf dem Benutzersystem gespeichert, auch die Berechnungen erfolgen hier. Der Vorteil für den Benutzer liegt hierbei darin, dass die Daten direkt auf seinem Rechner verwaltet werden. Die Nachteile liegen darin, dass bei schwacher Rechenleistung nicht alle Arten von Analysen sinnvoll durchführbar sind. Datenbankzugriffe erfolgen vom Benutzersystem zur jeweiligen Datenbank über das Internet. Die andere Möglichkeit besteht darin, das Programm als Internet-Applikation zu entwickeln. Hierbei wird die Benutzerschnittstelle des Programms von einem Internet-Browser dargestellt, die Programmlogik und Datenhaltung erfolgt auf einem Webserver. Der Server kann spezielle Systeme ansprechen, die auch die Analyse großer Datenmengen ermöglichen. Datenbanken können zentral auf dem Server gespeichert werden. Die Nachteile dieser Lösung liegen darin, dass die Benutzerdaten auf den Server geladen werden müssen und dort verbleiben. Ein weiterer Nachteil liegt darin, dass bei hoher Auslastung des Servers

die Antwortzeiten des Programms stark ansteigen.

GEPAT wurde als Internetapplikation entworfen, die in Hinblick auf eine einfache Benutzung entwickelt wurde, mit einem Design ähnlich einer Desktop-Applikation. Dies erlaubt eine plattformunabhängige Benutzung über das Internet, ohne das System auf dem Benutzerrechner installieren zu müssen. Durch die freie Verfügbarkeit des Web Servers sind auch separate Installationen für einzelne Arbeitsgruppen möglich. Die optionale Verwendung eines Rechenclusters zur Analyse erlaubt es, auch große Datensätze für viele Benutzer performant zu verarbeiten. Um Benutzer zu unterstützen, die noch unerfahren mit GEPAT sind, stehen animierte Anleitungen, eine Onlinehilfe und Testdatensätze zur Verfügung.

Wichtig für die Akzeptanz eines neuen Programms durch die Benutzer sind vor allem zwei Dinge: Eine einfach zu bedienende Benutzerschnittstelle und die Möglichkeit, Daten einfach Importieren und Exportieren zu können. Durch die Verknüpfung von CSS-Techniken mit JavaScript wurde die Verwendung einer Menüleiste sowie Dialogfenster möglich. Asynchrone Seitenaktualisierung erlaubt eine Fortschrittsanzeige beim Dateiupload. Zum einfachen Datenaustausch wurde Wert darauf gelegt, eine Vielzahl an Microarray-Dateiformaten zu unterstützen. Die Ergebnisse können vom Benutzer als Tabellen vom Server exportiert werden. Die Benutzerschnittstelle wurde absichtlich einfach gehalten, um unerfahrene Benutzer nicht zu überfordern. Auf erweiterte Optionen für Experten wurde verzichtet, da diese sich häufig bereits in das Bioconductor-Paket eingearbeitet haben und mit diesem Ihrer Analysen durchführen. Bei dieser Benutzergruppe ist eine Vereinfachung der Benutzerschnittstelle nicht notwendig und meist auch nicht erwünscht.

Bei der Entwicklung wurde Wert darauf gelegt, eine Drei-Schicht-Architektur anzuwenden. Diese bietet den Vorteil, einzelnen Schichten austauschen zu können, ohne Änderungen an den anderen Schichten vornehmen zu müssen. Die Präsentationsschicht wurde mit JavaServer Faces implementiert. Diese Technik war zu Beginn des Projektes relativ neu, als Alternative hätte sich das bewährte Jakarta Struts [187] Framework angeboten. Da sich der immer größer werdende Einfluss von JavaServer Faces jedoch abzeichnete, wurde diese Technologie zur Implementierung gewählt. Um die gewünschten Funktionen in die Benutzeroberfläche integrieren zu können, mussten einige verwendete Komponenten selbst implementiert werden. Mittlerweile wird ein Teil dieser Funktionen

auch über das Apache MyFaces Framework [188] zur Verfügung gestellt. Da keine entfernten Methodenaufrufe auf verteilten Systemen nötig waren, wurde beim Entwurf der Mittelschicht auf die Verwendung von EJB verzichtet. Deren Verwendung hätte nur zur Erhöhung der Komplexität geführt, aber keinen praktischen Nutzen besessen. In der Datenschicht wurde zuerst mit der Datenhaltung in einer MySQL-Datenbank experimentiert. Da es den Benutzern jedoch möglich sein sollte, eigene Daten importieren zu können, wurde diese Art der Datenhaltung verworfen, da sie zu einem starken Anstieg der Tabellengröße geführt hätte. Vielmehr wurde eine eigene Technik entwickelt, die nur die benötigte Datenpakete eines Benutzers im Speicher hält und nicht mehr benötigten Speicher automatisch für andere Benutzer freigibt. Zusammen mit der Verwendung Java-interner Dateiformate ist so ein schneller Zugriff auf alle Daten möglich. Zur Integration externer Datenbanken wurde die Abfrage über Web-Services geprüft. Jedoch sind nicht alle Datenbanken auf diese Weise erreichbar, bei manchen ist die Unterstützung derzeit allenfalls als experimentell zu bezeichnen. Deshalb wurde von dieser Idee Abstand genommen und alle Datenbanken wurden in eine eigene MySQL-Datenbank integriert. Diese speichert nur noch die zum Betrieb von GEPAT benötigten Informationen. Der Nachteil bei diesem Vorgehen besteht darin, dass Aktualisierungen in den Quelldatenbanken stets auch in die GEPAT Datenbanken eingespielt werden müssen. Hier steht noch kein automatisches Werkzeug zur Verfügung. Um den Aktualisierungsaufwand im Rahmen zu halten, wird derzeit nur der Organismus Mensch in den Datenbanken unterstützt. Eine Erweiterung auf andere Organismen ist jedoch problemlos möglich.

GEPAT bietet Unterstützung zum Import einer Reihe von Datenformaten und für Datentabellen. Daten können direkt von der Bildanalyse importiert werden, so dass ein Großteil der Informationen zur weiteren Verarbeitung genutzt werden kann. Im Bereich der Microarray-Forschung gewinnt das MAGE-ML Format immer mehr an Bedeutung. Obwohl es relativ komplex ist, stellt es bis jetzt die einzige übergreifende Möglichkeit dar, alle Aspekte einer Microarray-Studie austauschen zu können. Da allerdings jede beliebige Art von Dateien zur konkreten Speicherung der Microarray-Informationen verwendet werden kann, und damit eine automatische Verarbeitung erschwert wird, wird MAGE-ML in GEPAT derzeit noch nicht zum Import oder Export unterstützt.

Mit der Verwendung von Ensembl als Grundlage der Annotation steht den Modulen von

GEPAT eine breite Datengrundlage zur Interpretation und Analyse zur Verfügung. Neben den Geninformationen enthält Ensembl noch eine große Anzahl an Verweisen zu anderen Datenbanken. So stellt z.B. die Gene Ontology eine Reihe definierte Begriffe und Abhängigkeiten für biologische Prozesse, zelluläre Komponenten und molekularen Funktionen zur Verfügung. GEPAT ermöglicht neben der Anzeige dieser Funktionen für jedes Gen auch gezielte Analysen, die über- oder unterrepräsentierte Kategorien in einer Gruppe von Genen finden können. Damit lässt sich ein erster Überblick über die Funktionalität dieser Gene gewinnen. Mit der Integration der KEGG-Datenbank wurde eine Möglichkeit zur Analyse von grundlegenden Stoffwechselwegen geschaffen. Differentiell exprimierte Gene lassen sich direkt auf den KEGG-Karten anzeigen, so kann ein einfacher Überblick über mögliche Unterschiede in der Regulation zwischen zwei verschiedenen Zuständen gewonnen werden. Ein Nachteil liegt jedoch darin, dass die von KEGG verwendeten Stoffwechselwege allgemein gehalten sind und für mehrere Spezies angepasst werden können. Es existieren jedoch auch eine Reihe weiterer Datenbanken, wie z.B. Biocarta [189] oder GenMapp [190], die auch speziellere, teils benutzerdefinierte Stoffwechselwege unterstützen. Die Analysemöglichkeiten auf Stoffwechselkarten sind in GEPAT derzeit rudimentär, so dass hier Erweiterungen mit zusätzlichen Modulen wünschenswert sind. Um zusätzliche Informationen zu Assoziationen zwischen Proteinen zu erhalten, die nicht in Stoffwechselkarten enthalten sind, wird die Datenbank STRING verwendet. Diese bietet neben Informationen zum Vorkommen im gleichen Stoffwechselweg noch Informationen über experimentell bestätigte Proteininteraktionen, ähnliche Expressionsmuster und vorhandenes Literaturwissen. Die Assoziationen können für jedes Gen einzeln angesehen werden oder als Übersichtsgraph für eine Menge von Genen angezeigt werden. Auch hier erlaubt es die Anzeige differentiell exprimierter Gene ähnlich regulierte Genmengen identifizieren zu können. Zusätzlich zu diesen Datenbanken stehen noch chromosomale Informationen zu den Genen zur Verfügung, die um Information zu Chromosomabberationen erweitert werden können. Die Untersuchung der diffusen großzelligen B-Zell Lymphome zeigt hier Unterschiede im chromosomalen Profil der unterschiedlichen Subgruppen.

Um Informationen aus den Daten zu gewinnen, stehen in GEPAT eine Reihe von Analysemodulen zur Verfügung. Verschiedene Clusteringalgorithmen und Datencharakteristiken können verwendet werden, um Muster in den Daten zu finden.

Basierend auf diesen Mustern können weitere Analysen oder Interpretationen durchgeführt werden. Vergleicht man zwei unterschiedliche Mengen biologischer Proben, so ist der Grad der Unterschiede in der Genexpression von besonderem Interesse. Teilmengen der Daten können mit einem gemäßigten t-Test auf differentielle Expression analysiert werden, die Ergebnisse der Analyse können in vielen Ansichten angezeigt werden. Sicherlich sind die in GEPAT implementierten Algorithmen noch nicht ausreichend, um eine Untersuchung von Microarray-Daten vollständig durchführen zu können. Insbesondere zur Mustererkennung in den Daten wurden weitere Algorithmen entwickelt, die in GEPAT nicht vorhanden sind. Wünschenswert wäre die Integration von ANOVA, Support Vector Maschines und Korrespondenzanalyse. Insbesondere die Verknüpfung der Korrespondenzanalyse mit Geninformationen würde neue, interessante Analysemöglichkeiten eröffnen.

Ziel der Microarray-Untersuchung in der Medizin ist auch immer die Suche nach Ansatzpunkten für eine Therapie. So hofft man, stark differentiell exprimierte Gene als Ziel für Medikamente verwenden zu können. Insbesondere durch microRNA können sich weitere therapeutische Möglichkeiten ergeben. GEPAT unterstützt dieses Vorgehen, indem Proteininteraktionen und Stoffwechselwege direkt angezeigt werden können. Die Ziele von Medikamenten und microRNAs können durch Integration der Drugbank- und TarBase-Datenbanken direkt in der Genübersicht angezeigt werden. Wünschenswert wäre sicherlich noch die Integration weiterer, auch vom Benutzer definierbarer, Stoffwechselwege, sowie erweiterte Analysemethoden für diese. Da die Informationen, die Microarrays liefern können, alleine nicht ausreichend sind, um biologische Systeme verstehen zu können, ist es das Ziel, die Ergebnisse der verschiedenen Omics-Kategorien der Bioinformatik gemeinsam in einem Analysesystem verwenden zu können. GEPAT versucht, einen ersten Schritt in diese Richtung zu setzen.

6 Zusammenfassung

Die Messung der Genexpression ist für viele Bereiche der Biologie und Medizin wichtig geworden und unterstützt Studien über Behandlung, Krankheiten und Entwicklungsstadien. Microarrays können verwendet werden, um die Expression von tausenden mRNA-Molekülen gleichzeitig zu messen und ermöglichen so einen Einblick und einen Vergleich der verschiedenen zellulären Bedingungen. Die Daten, die durch Microarray-Experimente gewonnen werden, sind hochdimensional und verrauscht, eine Interpretation der Daten ist deswegen nicht einfach. Obwohl Programme für die statistische Auswertung von Microarraydaten existieren, fehlt vielen eine Integration der Analyseergebnisse mit einer automatischen Interpretationsmöglichkeit.

In dieser Arbeit wurde GEPAT, Genome Expression Pathway Analysis Tool, entwickelt, das eine Analyse der Genexpression unter dem Gesichtspunkten der Genomik, Proteomik und Metabolik ermöglicht. GEPAT integriert statistische Methoden zum Datenimport und -analyse mit biologischer Interpretation für Genmengen oder einzelne Gene, die auf dem Microarray gemessen werden. Verschiedene Typen von Oligonukleotid- und cDNA-Microarrays können importiert werden, unterschiedliche Normalisierungsmethoden können auf diese Daten angewandt werden, anschließend wird eine Datenannotation durchgeführt. Nach dem Import können mit GEPAT verschiedene statische Datenanalysemethoden wie hierarchisches, k-means und PCA-Clustern, ein auf einem linearen Modell basierender t-Test, oder ein Vergleich chromosomaler Profile durchgeführt werden. Die Ergebnisse der Analysen können auf Häufungen biologischer Begriffe und Vorkommen in Stoffwechselwegen oder Interaktionsnetzwerken untersucht werden. Verschiedene biologische Datenbanken wurden integriert, um zu jeder Gensonde auf dem Array Informationen zur Verfügung stellen zu können. GEPAT bietet keinen linearen Arbeitsablauf, sondern erlaubt die Benutzung von beliebigen Teilmengen von Genen oder biologischen Proben als Startpunkt einer neuen Analyse oder Interpretation. Dabei verlässt es sich auf bewährte Datenanalyse-Pakete, bietet einen modularen Ansatz zur einfachen Erweiterung und kann auf einem verteilten Computernetzwerk installiert werden, um eine große Zahl an Benutzern zu unterstützen. Es ist unter der LGPL [20] Open-Source Lizenz frei verfügbar und kann unter http://gepat.sourceforge.net heruntergeladen werden.

7 Summary

The measurement of gene expression data is relevant to many areas of biology and medicine, in the study of treatments, diseases, and developmental stages. Microarrays can be used to measure the expression level of thousands of mRNAs at the same time, allowing insight into or comparison of different cellular conditions. The data derived out of microarray experiments is highly dimensional and noisy, and interpretation of the results can get tricky. Although programs for the statistical analysis of microarray data exist, most of them lack an integration of analysis results and biological interpretation.

In this work GEPAT, Genome Expression Pathway Analysis Tool, was developed, offering an analysis of gene expression data under genomic, proteomic and metabolic context. GEPAT integrates statistical methods for data import and data analysis together with an biological interpretation for subset of genes or single genes measured on the chip. GEPAT imports various types of oligonucleotide and cDNA array data formats. Different normalization methods can be applied to the data, afterwards data annotation is performed. After import, GEPAT offers various statistical data analysis methods, as hierarchical, k-means and PCA clustering, a linear model based t-Test or chromosomal profile comparison. The results of the analysis can be interpreted by enrichment of biological terms, pathway analysis or interaction networks. Different biological databases are included, to give various informations for each probe on the chip. GEPAT offers no linear work flow, but allows the usage of any subset of probes and samples as start for a new data analysis or interpretation. GEPAT relies on established data analysis packages, offers a modular approach for an easy extension, and can be run on a computer grid to allow a large number of users. It is freely available under the LGPL open source license for academic and commercial users at http://gepat.sourceforge.net.

Stichwortverzeichnis

ABC..... 118
AHAH..... 85
alternatives Spleißen..... 14
Arbeitsmenge..... 79
ArrayCGH..... 29
ArrayExpress..... 53
B-Statistik..... 42
Benjamini und Hochberg Korrektur..... 43
Benjamini und Yekutieli Korrektur..... 43
Bioconductor..... 4
Biologischer Prozess..... 35
Bonferroni-Holm Step-Down-Methode..... 43
Bonferroni-Korrektur..... 43
cDNA..... 18
cDNA-Microarray..... 19
CGH..... 28
Chaperon..... 12
Chromatin..... 10
Chromosom..... 10
cis-Element..... 13
Clustering..... 44
Codon..... 12
COG..... 33
Comparative Genome Hybridization..... 28
Cytoplasma..... 8
Cytoskelett..... 8
Cytosol..... 8
Datenschicht..... 56
Dendrogramm..... 46
Differentiell exprimierte Gene..... 41
Differentielle Expression..... 41
Distanz..... 45
  euklidsche Metrik..... 45
  Manhattan-Metrik..... 45
  Pearson Korrelationsdistanz..... 45
  Spearman Korrelationsdistanz..... 45
  Standardisierung..... 46
DNA..... 9
DOM..... 72
Drei-Schicht-Architektur..... 57
Dye-Swap..... 21
EC-Nummer..... 106
EJB..... 58
EL..... 62
Enhancer..... 14
Ensembl..... 29
ensj..... 92
EnsMart..... 29
Enterprise JavaBeans..... 58
EST..... 26
Eukaryoten..... 8
Exon..... 12
Expression Language..... 62
Expressionsmatrix..... 36
false-discovery Rate..... 43
family-wise Error Rate..... 43
FDR..... 43
FuGE..... 50
FuGO..... 50
FWER..... 43
GAL-Datei..... 85
GCB..... 118
Gen..... 10
Gene Ontology..... 35
Geninformation..... 82
Genom..... 10
GEO..... 54
GO..... 35
GO Term Enrichment Analyse..... 48
Gruppe..... 79
Hauptkomponentenanalyse..... 47
Hierarchisches Clustern..... 46
  Complete linkage..... 46
  Single linkage..... 46
  UPGMA:..... 47
  Ward´s Methode..... 47
Hintergrundkorrektur.....
  MAS..... 38
  RMA..... 38
Histone..... 10
hnRNA..... 11
Housekeeping-Gen..... 37
hypergeometrische Verteilung..... 48
i..... 54
Intron..... 12
IPI..... 117
Java EE..... 55
Java ServerFaces..... 63
Java ServerPages..... 61
Java Servlets..... 59
JavaBeans..... 58
JNI..... 70

JSF....63
JSP....61
k nearest neighbor....36
k-Means Clustern....47
KEGG)....33
KGML....33
limma....42
Logikschicht....56
M/A–Plot....95
MAGE-ML....52
MAGE-OM....52
MAGE-STK....53
MAGE-TAB....53
MAS....87
Message Driven Beans....59
MGED....49
MGED Ontologie....52
MIAME....49
Microarray....15
MO....52
Model-View-Controller....63
moderated t-Test....42
Molekulare Funktion....35
mRNA....11
Multiple Testing Problem....43
NCBI....94
ncRNA....11
Normalisierung....37
Constant....38
Loess....39f., 88
Quantile....38, 41, 88
Scale....41, 88
VSN....39, 41, 88
Northern Blots....26
Nukleosom....10
Nukleotide....9
Nukleus....8
Oligonukleotid-Microarray....20
p-Wert....42
PCA....47
PCR....26
PM Korrektur....39
MAS....39
PM only....40
Präsentationsschicht....56
Prokaryoten....8
Promotor....13
Protein....9
Proteinsynthese....12
q-Wert....44
R....4
RefSeq....94
Remote Method Invocation....67
RMA....87
RMI....67
RNA....11
RNA-Polymerase....13
RSBI....50
RT-PCR....26
SAGE....27
Sequenz....9
Session Beans....59
siRNA....11
Spleißen....12
STRING....31
Supervised Clustering....45
t-Test....42
TATA Box....14
Teilmenge....78
trans-Element....13
Transkriptinformation....82
Transkription....11
Translation....12
tRNA....11
Unsupervised Clustering....45
Varianzanalyse....43
Web Services....66
Welch-T-Statistik....42
WSDL....66
Zelluläre Komponente....35
Zusammenfassung....
MAS....40
medianpolish....40
Zweikanal-Microarray....19

Abbildungsverzeichnis

Abbildung 1: Von der DNA zum Chromosom..... 10
Abbildung 2: Veröffentlichungen mit Microarray-Technik..... 17
Abbildung 3: Hybridisierung..... 18
Abbildung 4: cDNA- und Oligonukleotid-Microarrays..... 19
Abbildung 5: Scanergebnis eines cDNA Microarrays..... 20
Abbildung 6: Möglich Experimentdesigns für Microarray-Studien..... 23
Abbildung 7: Grundannahme des experimentellen Ansatzes..... 24
Abbildung 8: Ensembl..... 30
Abbildung 9: STRING..... 32
Abbildung 10: KEGG PATHWAY..... 34
Abbildung 11: MIAME-Standard im Überblick..... 52
Abbildung 12: ArrayExpress..... 54
Abbildung 13: Gene Expression Omnibus..... 55
Abbildung 14: Einteilung in eine 3-Schichten-Architektur..... 57
Abbildung 15: Ablauf eines Servlet-Aufrufs..... 60
Abbildung 16: Gültigkeit der Kontexte in einer Webapplikation..... 61
Abbildung 17: Ablauf eines JSP Seitenaufrufs..... 62
Abbildung 18: Ablauf einer JSF-Anfrage..... 65
Abbildung 19: Tabelle zur Benutzerdatenspeicherung..... 75
Abbildung 20: GEPAT ohne Module..... 76
Abbildung 21: Modulunabhängige Teilmengenauswahlmöglichkeiten in GEPAT..... 76
Abbildung 22: Übersicht über die verschiedenen Module..... 77
Abbildung 23: Verfahren zur Teilmengenselektion..... 80
Abbildung 24: Datei-Upload..... 84
Abbildung 25: Übersicht über die Microarray-Daten..... 86
Abbildung 26: Einstellung der Normalisierungsparameter..... 89
Abbildung 27: Ergebnisse der Normalisierung..... 89
Abbildung 28: Struktur der zur Annotation verwendeten Datenbank..... 90
Abbildung 29: Geninformationsseite für ein Gen..... 92
Abbildung 30: Proteininformationsseite..... 93
Abbildung 31: Literaturübersicht..... 94
Abbildung 32: Datenbankstruktur zur Speicherung der RefSeq-Einträge..... 94
Abbildung 33: Ergebnisse des limma t-Test..... 96
Abbildung 34: Teilmengenauswahl basierend auf Datencharakteristiken..... 97
Abbildung 35: Ergebnisse des hierarchischen Clusterns..... 98
Abbildung 36: Die Ergebnisse der Hauptkompontenanalyse..... 99
Abbildung 37: Ergebnisse des k-Means Clusterin..... 100
Abbildung 38: Genansicht..... 101

Abbildung 39: Teilmengenauswahl nach chromosomaler Lokalisation....101
Abbildung 40: Vergleich von CGH-Daten....102
Abbildung 41: Darstellung des direkten azyklischen GO-Graphen als Baum....104
Abbildung 42: Detailansicht für eine GO-Kategorie....104
Abbildung 43: Teilmengenauswahl nach GO-Kategorien....105
Abbildung 44: Ansicht einer KEGG-Stoffwechselkarte....106
Abbildung 45: Detailinformationen zu einer enzymatischen Aktivität....107
Abbildung 46: Die Struktur der Datenbank zur Speicherung der KEGG-Informationen..108
Abbildung 47: Die Struktur der zur Identifizierung der KEGG Gennamen verwendeten Datenbank....109
Abbildung 48: Die Assoziationsansicht....110
Abbildung 49: Struktur der Tabelle zur Speicherung der STRING Informationen....111
Abbildung 50: Assoziationsansicht für ein Gen....112
Abbildung 51: Differentiell exprimierte Gene in biologischen Prozessen....127
Abbildung 52: Differentiell exprimierte Gene in zellulären Komponenten....127
Abbildung 53: Differentiell exprimierte Gene in molekularen Funktionen....127
Abbildung 54: Über/Unterrepräsentierte Gene in der Gruppe der differentiell exprimierten Gene verglichen mit allen Genen auf dem Microarray....128
Abbildung 55: Unterschiede in den CGH-Daten....130

Tabellenverzeichnis

Tabelle 1: In GEPAT verwendete Bibliotheken..... 69
Tabelle 2: Mögliche Kriterien zur Teilmengenauswahl..... 80
Tabelle 3: Unterstützte Eingabeformate in GEPAT..... 84
Tabelle 4: Normalisierungsverfahren für Affymetrix-Microarraydaten..... 87
Tabelle 5: Normaliserungsverfahren für Zweikanal-Microarraydaten..... 88
Tabelle 6: Unterschiede der Gruppe GCB im Vergleich zu ABC..... 119
Tabelle 7: Gensonden-Teilmengen der Studie von Rosenwald et. al..... 120
Tabelle 8: Ergebnisse der differentiellen Expression..... 120
Tabelle 9: Ergebnisse der differentiellen Expression..... 123

Literaturverzeichnis

[1] Carroll, Lewis: *Alice's Adventures in Wonderland*; Penguin Books 1994.

[2] Smyth GK, Yang YH, Speed T: **Statistical issues in cDNA microarray data analysis.** *Methods Mol Biol* 2003, **224**:111-136.

[3] **Bioconductor** [http://www.bioconductor.org/]

[4] **The R Project For Statistical Computing** [http://www.r-project.org/]

[5] Pelizzola M, Pavelka N, Foti M, Ricciardi-Castagnoli P: **AMDA: an R package for the automated microarray data analysis.** *BMC Bioinformatics* 2006, **7**:335.

[6] Rainer J, Sanchez-Cabo F, Stocker G, Sturn A, Trajanoski Z: **CARMAweb: comprehensive R- and bioconductor-based web service for microarray data analysis.** *Nucleic Acids Res* 2006, **34**:W498-503.

[7] Kapushesky M, Kemmeren P, Culhane AC, Durinck S, Ihmels J, Körner C, Kull M, Torrente A, Sarkans U, Vilo J, Brazma A: **Expression Profiler: next generation–an online platform for analysis of microarray data.** *Nucleic Acids Res* 2004, **32**:W465-70.

[8] Vaquerizas JM, Conde L, Yankilevich P, Cabezón A, Minguez P, Díaz-Uriarte R, Al-Shahrour F, Herrero J, Dopazo J: **GEPAS, an experiment-oriented pipeline for the analysis of microarray gene expression data.** *Nucleic Acids Res* 2005, **33**:W616-20.

[9] Shamir R, Maron-Katz A, Tanay A, Linhart C, Steinfeld I, Sharan R, Shiloh Y, Elkon R: **EXPANDER–an integrative program suite for microarray data analysis.** *BMC Bioinformatics* 2005, **6**:232.

[10] Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J: **TM4: a free, open-source system for microarray data management and analysis.** *Biotechniques* 2003, **34**:374-378.

[11] Al-Shahrour F, Minguez P, Vaquerizas JM, Conde L, Dopazo J: **BABELOMICS: a suite of web tools for functional annotation and analysis of groups of genes in high-throughput experiments.** *Nucleic Acids Res* 2005, **33**:W460-4.

[12] Zhang B, Kirov S, Snoddy J: **WebGestalt: an integrated system for exploring gene sets in various biological contexts.** *Nucleic Acids Res* 2005, **33**:W741-8.

[13] Doniger SW, Salomonis N, Dahlquist KD, Vranizan K, Lawlor SC, Conklin BR: **MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data.** *Genome Biol* 2003, **4**:R7.

[14] Masseroli M, Galati O, Pinciroli F: **GFINDer: genetic disease and phenotype location statistical analysis and mining of dynamically annotated gene lists.** *Nucleic Acids Res* 2005, **33**:W717-23.

[15] Mlecnik B, Scheideler M, Hackl H, Hartler J, Sanchez-Cabo F, Trajanoski Z:

PathwayExplorer: web service for visualizing high-throughput expression data on biological pathways. *Nucleic Acids Res* 2005, **33**:W633-7.

[16] Dennis GJ, Sherman BT, Hosack DA, Yang J, Gao W, Lane HC, Lempicki RA: **DAVID: Database for Annotation, Visualization, and Integrated Discovery.** *Genome Biol* 2003, **4**:P3.

[17] Hubbard TJP, Aken BL, Beal K, Ballester B, Caccamo M, Chen Y, Clarke L, Coates G, Cunningham F, Cutts T, Down T, Dyer SC, Fitzgerald S, Fernandez-Banet J, Graf S, Haider S, Hammond M, Herrero J, Holland R, Howe K, Howe K, Johnson N, Kahari A, Keefe D, Kokocinski F, Kulesha E, Lawson D, Longden I, Melsopp C, Megy K, Meidl P, Ouverdin B, Parker A, Prlic A, Rice S, Rios D, Schuster M, Sealy I, Severin J, Slater G, Smedley D, Spudich G, Trevanion S, Vilella A, Vogel J, White S, Wood M, Cox T, Curwen V, Durbin R, Fernandez-Suarez XM, Flicek P, Kasprzyk A, Proctor G, Searle S, Smith J, Ureta-Vidal A, Birney E: **Ensembl 2007.** *Nucleic Acids Res* 2007, **35**:D610-7.

[18] Kasprzyk A, Keefe D, Smedley D, London D, Spooner W, Melsopp C, Hammond M, Rocca-Serra P, Cox T, Birney E: **EnsMart: a generic system for fast and flexible access to biological data.** *Genome Res* 2004, **14**:160-169.

[19] Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13**:2498-2504.

[20] **GNU Lesser General Public License** [http://www.gnu.org/licenses/lgpl.html]

[21] **GEPAT** [gepat.sourceforge.net]

[22] Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, Powell JI, Yang L, Marti GE, Moore T, Hudson JJ, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Warnke R, Levy R, Wilson W, Grever MR, Byrd JC, Botstein D, Brown PO, Staudt LM: **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling.** *Nature* 2000, **403**:503-511.

[23] Rosenwald A, Wright G, Chan WC, Connors JM, Campo E, Fisher RI, Gascoyne RD, Muller-Hermelink HK, Smeland EB, Giltnane JM, Hurt EM, Zhao H, Averett L, Yang L, Wilson WH, Jaffe ES, Simon R, Klausner RD, Powell J, Duffey PL, Longo DL, Greiner TC, Weisenburger DD, Sanger WG, Dave BJ, Lynch JC, Vose J, Armitage JO, Montserrat E, López-Guillermo A, Grogan TM, Miller TP, LeBlanc M, Ott G, Kvaloy S, Delabie J, Holte H, Krajci P, Stokke T, Staudt LM: **The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma.** *N Engl J Med* 2002, **346**:1937-1947.

[24] Bea S, Zettl A, Wright G, Salaverria I, Jehn P, Moreno V, Burek C, Ott G, Puig X, Yang L, Lopez-Guillermo A, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Gascoyne RD, Connors JM, Grogan TM, Braziel R, Fisher RI, Smeland EB, Kvaloy S, Holte H, Delabie J, Simon R, Powell J, Wilson WH, Jaffe ES, Montserrat E, Muller-Hermelink H, Staudt LM, Campo E, Rosenwald A: **Diffuse large B-cell lymphoma subgroups have distinct genetic profiles that influence tumor biology and improve gene-expression-based survival prediction.** *Blood* 2005, **106**:3183-3190.

[25] Watson JD, Crick FH: **The structure of DNA.** *Cold Spring Harb Symp Quant Biol* 1953, **18**:123-131.

[26] **National Human Genome Research Institute Glossary** [http://www.genome.gov/glossary.cfm]

[27] Costa FF: **Non-coding RNAs: Lost in translation?** *Gene* 2007, **386**:1-10.

[28] Hamilton AJ, Baulcombe DC: **A species of small antisense RNA in posttranscriptional gene silencing in plants.** *Science* 1999, **286**:950-952.

[29] Crick F: **Central dogma of molecular biology.** *Nature* 1970, **227**:561-563.

[30] Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.** *Mol Biol Cell* 1998, **9**:3273-3297.

[31] Pufall MA, Lee GM, Nelson ML, Kang H, Velyvis A, Kay LE, McIntosh LP, Graves BJ: **Variable control of Ets-1 DNA binding by multiple phosphates in an unstructured region.** *Science* 2005, **309**:142-145.

[32] Prasanth KV, Prasanth SG, Xuan Z, Hearn S, Freier SM, Bennett CF, Zhang MQ, Spector DL: **Regulating gene expression through RNA nuclear retention.** *Cell* 2005, **123**:249-263.

[33] Cawley S, Bekiranov S, Ng HH, Kapranov P, Sekinger EA, Kampa D, Piccolboni A, Sementchenko V, Cheng J, Williams AJ, Wheeler R, Wong B, Drenkow J, Yamanaka M, Patel S, Brubaker S, Tammana H, Helt G, Struhl K, Gingeras TR: **Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs.** *Cell* 2004, **116**:499-509.

[34] Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray.** *Science* 1995, **270**:467-470.

[35] Lipshutz RJ, Morris D, Chee M, Hubbell E, Kozal MJ, Shah N, Shen N, Yang R, Fodor SP: **Using oligonucleotide probe arrays to access genetic diversity.** *Biotechniques* 1995, **19**:442-447.

[36] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES: **Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.** *Science* 1999, **286**:531-537.

[37] Petersen D, Chandramouli GVR, Geoghegan J, Hilburn J, Paarlberg J, Kim CH, Munroe D, Gangi L, Han J, Puri R, Staudt L, Weinstein J, Barrett JC, Green J, Kawasaki ES: **Three microarray platforms: an analysis of their concordance in profiling gene expression.** *BMC Genomics* 2005, **6**:63.

[38] Lipshutz RJ, Fodor SP, Gingeras TR, Lockhart DJ: **High density synthetic oligonucleotide arrays.** *Nat Genet* 1999, **21**:20-24.

[39] Greenbaum D, Jansen R, Gerstein M: **Analysis of mRNA expression and protein**

abundance data: an approach for the comparison of the enrichment of features in the cellular population of proteins and transcripts. *Bioinformatics* 2002, **18**:585-596.

[40] Sharabiani MTA, Siermala M, Lehtinen TO, Vihinen M: **Dynamic covariation between gene expression and proteome characteristics.** *BMC Bioinformatics* 2005, **6**:215.

[41] Beyer A, Hollunder J, Nasheuer H, Wilhelm T: **Post-transcriptional expression regulation in the yeast Saccharomyces cerevisiae on a genomic scale.** *Mol Cell Proteomics* 2004, **3**:1083-1092.

[42] Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network.** *Science* 2001, **292**:929-934.

[43] Hegde PS, White IR, Debouck C: **Interplay of transcriptomics and proteomics.** *Curr Opin Biotechnol* 2003, **14**:647-651.

[44] Stalteri MA, Harrison AP: **Interpretation of multiple probe sets mapping to the same gene in Affymetrix GeneChips.** *BMC Bioinformatics* 2007, **8**:13.

[45] Kothapalli R, Yoder SJ, Mane S, Loughran TPJ: **Microarray results: how accurate are they?** *BMC Bioinformatics* 2002, **3**:22.

[46] Harbig J, Sprinkle R, Enkemann SA: **A sequence-based identification of the genes detected by probesets on the Affymetrix U133 plus 2.0 array.** *Nucleic Acids Res* 2005, **33**:e31.

[47] Hamadeh HK, Bushel P, Tucker CJ, Martin K, Paules R, Afshari CA: **Detection of diluted gene expression alterations using cDNA microarrays.** *Biotechniques* 2002, **32**:322, 324, 326-9.

[48] Velculescu VE, Zhang L, Vogelstein B, Kinzler KW: **Serial analysis of gene expression.** *Science* 1995, **270**:484-487.

[49] Kuo WP, Liu F, Trimarchi J, Punzo C, Lombardi M, Sarang J, Whipple ME, Maysuria M, Serikawa K, Lee SY, McCrann D, Kang J, Shearstone JR, Burke J, Park DJ, Wang X, Rector TL, Ricciardi-Castagnoli P, Perrin S, Choi S, Bumgarner R, Kim JH, Short GF3, Freeman MW, Seed B, Jensen R, Church GM, Hovig E, Cepko CL, Park P, Ohno-Machado L, Jenssen T: **A sequence-oriented comparison of gene expression measurements across different hybridization-based technologies.** *Nat Biotechnol* 2006, **24**:832-840.

[50] Järvinen A, Hautaniemi S, Edgren H, Auvinen P, Saarela J, Kallioniemi O, Monni O: **Are data from different gene expression microarray platforms comparable?** *Genomics* 2004, **83**:1164-1168.

[51] Kallioniemi A, Kallioniemi OP, Sudar D, Rutovitz D, Gray JW, Waldman F, Pinkel D: **Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors.** *Science* 1992, **258**:818-821.

[52] Lichter P, Joos S, Bentz M, Lampel S: **Comparative genomic hybridization: uses and limitations.** *Semin Hematol* 2000, **37**:348-357.

[53] Weiss MM, Hermsen MA, Meijer GA, van Grieken NC, Baak JP, Kuipers EJ, van Diest PJ: **Comparative genomic hybridisation.** *Mol Pathol* 1999, **52**:243-251.

[54] Snijders AM, Pinkel D, Albertson DG: **Current status and future prospects of array-based comparative genomic hybridisation.** *Brief Funct Genomic Proteomic* 2003, **2**:37-45.

[55] Birney E, Andrews TD, Bevan P, Caccamo M, Chen Y, Clarke L, Coates G, Cuff J, Curwen V, Cutts T, Down T, Eyras E, Fernandez-Suarez XM, Gane P, Gibbins B, Gilbert J, Hammond M, Hotz H, Iyer V, Jekosch K, Kahari A, Kasprzyk A, Keefe D, Keenan S, Lehvaslaiho H, McVicker G, Melsopp C, Meidl P, Mongin E, Pettett R, Potter S, Proctor G, Rae M, Searle S, Slater G, Smedley D, Smith J, Spooner W, Stabenau A, Stalker J, Storey R, Ureta-Vidal A, Woodwark KC, Cameron G, Durbin R, Cox A, Hubbard T, Clamp M: **An overview of Ensembl.** *Genome Res* 2004, **14**:925-928.

[56] Curwen V, Eyras E, Andrews TD, Clarke L, Mongin E, Searle SMJ, Clamp M: **The Ensembl automatic gene annotation system.** *Genome Res* 2004, **14**:942-950.

[57] Hinrichs AS, Karolchik D, Baertsch R, Barber GP, Bejerano G, Clawson H, Diekhans M, Furey TS, Harte RA, Hsu F, Hillman-Jackson J, Kuhn RM, Pedersen JS, Pohl A, Raney BJ, Rosenbloom KR, Siepel A, Smith KE, Sugnet CW, Sultan-Qurraie A, Thomas DJ, Trumbower H, Weber RJ, Weirauch M, Zweig AS, Haussler D, Kent WJ: **The UCSC Genome Browser Database: update 2006.** *Nucleic Acids Res* 2006, **34**:D590-8.

[58] Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, Geer LY, Kapustin Y, Khovayko O, Landsman D, Lipman DJ, Madden TL, Maglott DR, Ostell J, Miller V, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Sirotkin K, Souvorov A, Starchenko G, Tatusov RL, Tatusova TA, Wagner L, Yaschenko E: **Database resources of the National Center for Biotechnology Information.** *Nucleic Acids Res* 2007, **35**:D5-12.

[59] **The Apache HTTP Server Project** [http://httpd.apache.org/]

[60] Stalker J, Gibbins B, Meidl P, Smith J, Spooner W, Hotz H, Cox AV: **The Ensembl Web site: mechanics of a genome browser.** *Genome Res* 2004, **14**:951-955.

[61] Stabenau A, McVicker G, Melsopp C, Proctor G, Clamp M, Birney E: **The Ensembl core software libraries.** *Genome Res* 2004, **14**:929-933.

[62] Potter SC, Clarke L, Curwen V, Keenan S, Mongin E, Searle SMJ, Stabenau A, Storey R, Clamp M: **The Ensembl analysis pipeline.** *Genome Res* 2004, **14**:934-941.

[63] von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P: **Comparative assessment of large-scale data sets of protein-protein interactions.** *Nature* 2002, **417**:399-403.

[64] Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M: **From genomics to chemical genomics: new developments in KEGG.** *Nucleic Acids Res* 2006, **34**:D354-7.

[65] **KEGG Markup Language Manual** [http://www.genome.jp/kegg/docs/xml/]

[66] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet* 2000, **25**:25-29.

[67] Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, Foulger R, Eilbeck K, Lewis S, Marshall B, Mungall C, Richter J, Rubin GM, Blake JA, Bult C, Dolan M, Drabkin H, Eppig JT, Hill DP, Ni L, Ringwald M, Balakrishnan R, Cherry JM, Christie KR, Costanzo MC, Dwight SS, Engel S, Fisk DG, Hirschman JE, Hong EL, Nash RS, Sethuraman A, Theesfeld CL, Botstein D, Dolinski K, Feierbach B, Berardini T, Mundodi S, Rhee SY, Apweiler R, Barrell D, Camon E, Dimmer E, Lee V, Chisholm R, Gaudet P, Kibbe W, Kishore R, Schwarz EM, Sternberg P, Gwinn M, Hannick L, Wortman J, Berriman M, Wood V, de la Cruz N, Tonellato P, Jaiswal P, Seigfried T, White R: **The Gene Ontology (GO) database and informatics resource.** *Nucleic Acids Res* 2004, **32**:D258-61.

[68] Quackenbush J: **Computational analysis of microarray data.** *Nat Rev Genet* 2001, **2**:418-427.

[69] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: **Missing value estimation methods for DNA microarrays.** *Bioinformatics* 2001, **17**:520-525.

[70] Bolstad BM, Irizarry RA, Astrand M, Speed TP: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.** *Bioinformatics* 2003, **19**:185-193.

[71] Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP: **Exploration, normalization, and summaries of high density oligonucleotide array probe level data.** *Biostatistics* 2003, **4**:249-264.

[72] Affymetrix: **Statistical Algorithms Description Document**. Affymetrix, Santa Clara, CA, 2002 [http://www.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf]

[73] Yang Y, Dudoit S, Luu P, Speed T: **Normalization for cDNA microarray data**. In *Proceedings of SPIE*. Volume 4266. Bittner M, Chen Y, Dorsel A, Dougherty E. 2001:141-152.

[74] Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP: **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.** *Nucleic Acids Res* 2002, **30**:e15.

[75] Huber W, von Heydebreck A, Sültmann H, Poustka A, Vingron M: **Variance stabilization applied to microarray data calibration and to the quantification of differential expression.** *Bioinformatics* 2002, **18 Suppl 1**:S96-104.

[76] Huber W, von Heydebreck A, Sueltmann H, Poustka A, Vingron M: **Parameter estimation for the calibration and variance stabilization of microarray data.** *Stat Appl Genet Mol Biol* 2003, **2**:Artikel 3.

[77] Naef F, Lim DA, Patil N, Magnasco M: **DNA hybridization to mismatched templates:**

a chip study. *Phys Rev E Stat Nonlin Soft Matter Phys* 2002, **65**:40902.

[78] Tukey J: *Exploratory Data Analysis*; Reading Massachusetts: Addison-Wesley 1977.

[79] Irizarry RA, Bolstad BM, Collin F, Cope LM, Hobbs B, Speed TP: **Summaries of Affymetrix GeneChip probe level data.** *Nucleic Acids Res* 2003, **31**:e15.

[80] Wu Z, Irizarry R, Gentleman R, Martinez-Murillo F, Spencer F: **A model-based background adjustment for oligonucleotide expression arrays.** *Journal of the American Statistical Association* 2004, **99**:909-917.

[81] Wu Z, Irizarry RA: **Stochastic models inspired by hybridization theory for short oligonucleotide arrays.** *J Comput Biol* 2005, **12**:882-893.

[82] Affymetrix: **Guide to Probe Logarithmic Intensity Error (PLIER) Estimation.** Affymetrix, Santa Clara, CA, 2005 [http://www.affymetrix.com/support/technical/technotes/plier_technote.pdf]

[83] Katz S, Irizarry RA, Lin X, Tripputi M, Porter MW: **A summarization approach for Affymetrix GeneChip data using a reference training set from a large, biologically diverse database.** *BMC Bioinformatics* 2006, **7**:464.

[84] Seo J, Hoffman EP: **Probe set algorithms: is there a rational best bet?** *BMC Bioinformatics* 2006, **7**:395.

[85] Smyth GK, Speed T: **Normalization of cDNA microarray data.** *Methods* 2003, **31**:265-273.

[86] Goldstein D (Ed): *Normalization for two-color cDNA microarray data*: *2003*; . IMS Lecture Notes - Monograph Series; 2003.

[87] Kim SY, Lee JW, Sohn IS: **Comparison of various statistical methods for identifying differential gene expression in replicated microarray data.** *Stat Methods Med Res* 2006, **15**:3-20.

[88] Welch B: **The generalization of "student's" problem when several different population variances are involved** *Biometrika* 1947, **34**:28-35.

[89] Lönnstedt I, Speed T: **Replicated microarray data** *Statistica Sinica* 2002, **12**:31-46.

[90] Smyth GK: **Linear models and empirical bayes methods for assessing differential expression in microarray experiments.** *Stat Appl Genet Mol Biol* 2004, **3**:Article3.

[91] Cui X, Churchill GA: **Statistical tests for differential expression in cDNA microarray experiments.** *Genome Biol* 2003, **4**:210.

[92] Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing** *Journal of the Royal Statistical Society Series B* 1995, **57**:289-300.

[93] Benjamini Y, Yekutieli D: **The control of the false discovery rate in multiple testing under dependency** *Annals of Statistics* 2001, **29**:1165-1188.

[94] Storey JD: **A direct approach to false discovery rates** *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2002, **64**:279-498.

[95] Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proc Natl Acad Sci U S A* 1998, **95**:14863-14868.

[96] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR: **Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.** *Proc Natl Acad Sci U S A* 1999, **96**:2907-2912.

[97] Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares MJ, Haussler D: **Knowledge-based analysis of microarray gene expression data by using support vector machines.** *Proc Natl Acad Sci U S A* 2000, **97**:262-267.

[98] Raychaudhuri S, Stuart JM, Altman RB: **Principal components analysis to summarize microarray experiments: application to sporulation time series.** *Pac Symp Biocomput* 2000, 455-466.

[99] Khatri P, Drăghici S: **Ontological analysis of gene expression data: current tools, limitations, and open problems.** *Bioinformatics* 2005, **21**:3587-3595.

[100] Rivals I, Personnaz L, Taing L, Potier M: **Enrichment or depletion of a GO category within a class of genes: which test?** *Bioinformatics* 2007, **23**:401-407.

[101] Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FC, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A, Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, Vingron M: **Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.** *Nat Genet* 2001, **29**:365-371.

[102] **MGED Society** [http://www.mged.org/]

[103] Sansone S, Rocca-Serra P, Tong W, Fostel J, Morrison N, Jones AR: **A strategy capitalizing on synergies: the Reporting Structure for Biological Investigation (RSBI) working group.** *OMICS* 2006, **10**:164-171.

[104] Jones AR, Pizarro A, Spellman P, Miller M: **FuGE: Functional Genomics Experiment Object Model.** *OMICS* 2006, **10**:179-184.

[105] Whetzel PL, Brinkman RR, Causton HC, Fan L, Field D, Fostel J, Fragoso G, Gray T, Heiskanen M, Hernandez-Boussard T, Morrison N, Parkinson H, Rocca-Serra P, Sansone S, Schober D, Smith B, Stevens R, Stoeckert CJJ, Taylor C, White J, Wood A: **Development of FuGO: an ontology for functional genomics investigations.** *OMICS* 2006, **10**:199-204.

[106] Ball CA, Brazma A: **MGED standards: work in progress.** *OMICS* 2006, **10**:138-144.

[107] Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M, Swiatek M, Marks WL, Goncalves J, Markel S, Iordan D, Shojatalab M, Pizarro A, White J, Hubley R, Deutsch E, Senger M, Aronow BJ, Robinson A, Bassett D, Stoeckert CJJ, Brazma A: **Design and implementation of microarray gene expression markup language (MAGE-ML).** *Genome Biol* 2002, **3**:RESEARCH0046.

[108]Whetzel PL, Parkinson H, Causton HC, Fan L, Fostel J, Fragoso G, Game L, Heiskanen M, Morrison N, Rocca-Serra P, Sansone S, Taylor C, White J, Stoeckert CJJ: **The MGED Ontology: a resource for semantics-based description of microarray experiments.** *Bioinformatics* 2006, **22**:866-873.

[109]**MAGE Software Toolkit** [http://mged.sourceforge.net/software/MAGEstk.php]

[110]Durinck S, Allemeersch J, Carey VJ, Moreau Y, De Moor B: **Importing MAGE-ML format microarray data into BioConductor.** *Bioinformatics* 2004, **20**:3641-3642.

[111]Rayner TF, Rocca-Serra P, Spellman PT, Causton HC, Farne A, Holloway E, Irizarry RA, Liu J, Maier DS, Miller M, Petersen K, Quackenbush J, Sherlock G, Stoeckert CJJ, White J, Whetzel PL, Wymore F, Parkinson H, Sarkans U, Ball CA, Brazma A: **A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB.** *BMC Bioinformatics* 2006, **7**:489.

[112]Zeeberg BR, Riss J, Kane DW, Bussey KJ, Uchio E, Linehan WM, Barrett JC, Weinstein JN: **Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics.** *BMC Bioinformatics* 2004, **5**:80.

[113]Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, Abeygunawardena N, Holloway E, Kapushesky M, Kemmeren P, Lara GG, Oezcimen A, Rocca-Serra P, Sansone S: **ArrayExpress–a public repository for microarray gene expression data at the EBI.** *Nucleic Acids Res* 2003, **31**:68-71.

[114]Parkinson H, Kapushesky M, Shojatalab M, Abeygunawardena N, Coulson R, Farne A, Holloway E, Kolesnykov N, Lilja P, Lukk M, Mani R, Rayner T, Sharma A, William E, Sarkans U, Brazma A: **ArrayExpress–a public database of microarray experiments and gene expression profiles.** *Nucleic Acids Res* 2007, **35**:D747-50.

[115]Barrett T, Troup DB, Wilhite SE, Ledoux P, Rudnev D, Evangelista C, Kim IF, Soboleva A, Tomashevsky M, Edgar R: **NCBI GEO: mining tens of millions of expression profiles–database and tools update.** *Nucleic Acids Res* 2007, **35**:D760-5.

[116]Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI.** *Nucleic Acids Res* 2007, **35**:D26-31.

[117]**Java Platform, Enterprise Edition** [http://java.sun.com/javaee/]

[118]**JBoss Application Server** [http://www.jboss.org/]

[119]**GlassFish Application Server** [http://java.sun.com/javaee/community/glassfish/]

[120]**Enterprise JavaBeans** [http://java.sun.com/products/ejb/]

[121]**Java Servlet Technology** [http://java.sun.com/products/servlet/index.html]

[122]**Java ServerFaces Technology** [http://java.sun.com/javaee/javaserverfaces/]

[123]Gamma E, Helm R, Johnson R, Vlissides J: *Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software*; Addison-Wesley 2004

[124]Reenskaug T: **Models-Views-Controllers**. XEROX PARC, 1979 [http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html]

[125]**GEPAT an der Universität Würzburg** [gepat.bioapps.biozentrum.uni-wuerzburg.de]

[126]**Java Technology** [http://java.sun.com]

[127]Johnson R: *Expert One-on-one J2EE Design and Development.*; Wiley & Sons 2003

[128]**Apache Tomcat** [http://tomcat.apache.org/]

[129]**Distributed Resource Management Application Api** [http://drmaa.org/]

[130]**Sun Grid Engine** [http://gridengine.sunsource.net/]

[131]**Java Universal Network/Graph Framework** [http://jung.sourceforge.net]

[132]**JavaServer Pages Standard Tag Library** [http://java.sun.com/products/jsp/jstl/]

[133]**Jakarta Taglibs** [http://jakarta.apache.org/taglibs/]

[134]**JavaMail** [http://java.sun.com/products/javamail/]

[135]**JavaBeans Activation Framework** [http://java.sun.com/products/javabeans/jaf/]

[136]**Jakarta Commons** [http://jakarta.apache.org/commons/]

[137]**Java Database Connectivity** [http://java.sun.com/javase/technologies/database/]

[138]**MySQL Connector/J** [http://www.mysql.de/products/connector/j/]

[139]**log4j** [http://logging.apache.org/log4j/]

[140]**JUnit** [http://www.junit.org/]

[141]**Cern Colt Scientific Library** [http://dsd.lbl.gov/~hoschek/colt/]

[142]**Ensj Java API to Ensembl.** [http://www.ensembl.org/info/software/java/index.html]

[143]**rJava** [http://www.rforge.net/rJava/]

[144]**Java Native Interface** [http://java.sun.com/j2se/1.5.0/docs/guide/jni/index.html]

[145]Shneiderman B: *Designing the User Interface: Strategies for Effective Human-Computer Interaction.*; Addison-Wesley Longman, Amsterdam 2004

[146]**mySQL** [http://www.mysql.com]

[147]**GenePix File Formats** [http://www.moleculardevices.com/pages/software/gn_genepix_file_formats.html]

[148]**Asynchronous HTML and HTTP** [http://microformats.org/wiki/rest/ahah]

[149]**NCBI UniGene** [www.ncbi.nlm.nih.gov/UniGene]

[150]**Genechip Exon Array System** [http://www.affymetrix.com/products/arrays/exon_application.affx]

[151]Letunic I, Copley RR, Pils B, Pinkert S, Schultz J, Bork P: **SMART 5: domains in the context of genomes and networks.** *Nucleic Acids Res* 2006, **34**:D257-60.

[152]Finn RD, Mistry J, Schuster-Böckler B, Griffiths-Jones S, Hollich V, Lassmann T,

Moxon S, Marshall M, Khanna A, Durbin R, Eddy SR, Sonnhammer ELL, Bateman A: **Pfam: clans, web tools and services.** *Nucleic Acids Res* 2006, **34**:D247-51.

[153]Pruitt KD, Tatusova T, Maglott DR: **NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins.** *Nucleic Acids Res* 2007, **35**:D61-5.

[154]Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL: **GenBank.** *Nucleic Acids Res* 2007, **35**:D21-5.

[155]**The Newick format definition** [http://evolution.genetics.washington.edu/phylip/newicktree.html]

[156]Hartigan JA, Wong MA: **A K-means clustering algorithm** *Applied Statistics* 1979, **28**:100-108.

[157]Knutsen T, Gobu V, Knaus R, Padilla-Nash H, Augustus M, Strausberg RL, Kirsch IR, Sirotkin K, Ried T: **The interactive online SKY/M-FISH & CGH database and the Entrez cancer chromosomes search database: linkage of chromosomal aberrations with the genome sequence.** *Genes Chromosomes Cancer* 2005, **44**:52-64.

[158]Gentleman R, Carey V, Huber W, Irizarry R, Dudoit R: *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*; Springer 2005

[159]**The OBO Flat File Format Specification, version 1.2** [http://www.geneontology.org/GO.format.obo-1_2.shtml]

[160]**The Gene Ontology** [http://www.geneontology.org]

[161]Kotera M, Okuno Y, Hattori M, Goto S, Kanehisa M: **Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions.** *J Am Chem Soc* 2004, **126**:16487-16498.

[162]von Mering C, Jensen LJ, Kuhn M, Chaffron S, Doerks T, Krüger B, Snel B, Bork P: **STRING 7–recent developments in the integration and prediction of protein interactions.** *Nucleic Acids Res* 2007, **35**:D358-62.

[163]Fruchterman TM, Reingold EM: **Graph Drawing by Force-directed Placement** *Software - Practice and Experience* 1991, **21**:1129-1164.

[164]Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J: **DrugBank: a comprehensive resource for in silico drug discovery and exploration.** *Nucleic Acids Res* 2006, **34**:D668-72.

[165]Sethupathy P, Corda B, Hatzigeorgiou AG: **TarBase: A comprehensive database of experimentally supported animal microRNA targets.** *RNA* 2006, **12**:192-197.

[166]Honold E: **Erste Evaluierung einer computergestützten Datenbank zur Analyse von B-Zell-Lymphomen**. PhD thesis. Julius-Maximilians-Universität Würzburg 2005.

[167]Riede UN, Werner M, Schaefer H: *Allgemeine und spezielle Pathologie*; Thieme Verlag, Stuttgart 5. Auflage (2003)

[168]Kumar V, Cotran RS, Robbins SL: *Robbins Basic Pathology*; W.B. Saunders Company Philadelphia 2003 (7. Auflage)

[169]Polo JM, Dell'Oso T, Ranuncolo SM, Cerchietti L, Beck D, Da Silva GF, Prive GG, Licht JD, Melnick A: **Specific peptide interference reveals BCL6 transcriptional and oncogenic mechanisms in B-cell lymphoma cells.** *Nat Med* 2004, **10**:1329-1335.

[170]Dyomin VG, Rao PH, Dalla-Favera R, Chaganti RS: **BCL8, a novel gene involved in translocations affecting band 15q11-13 in diffuse large-cell lymphoma.** *Proc Natl Acad Sci U S A* 1997, **94**:5728-5732.

[171]Lee JW, Yoo NJ, Soung YH, Kim HS, Park WS, Kim SY, Lee JH, Park JY, Cho YG, Kim CJ, Ko YH, Kim SH, Nam SW, Lee JY, Lee SH: **BRAF mutations in non-Hodgkin's lymphoma.** *Br J Cancer* 2003, **89**:1958-1960.

[172]Shin MS, Kim HS, Kang CS, Park WS, Kim SY, Lee SN, Lee JH, Park JY, Jang JJ, Kim CW, Kim SH, Lee JY, Yoo NJ, Lee SH: **Inactivating mutations of CASP10 gene in non-Hodgkin lymphomas.** *Blood* 2002, **99**:4094-4099.

[173]Sigal S, Ninette A, Rechavi G: **Microarray studies of prognostic stratification and transformation of follicular lymphomas.** *Best Pract Res Clin Haematol* 2005, **18**:143-156.

[174]Greiner A, Müller KB, Hess J, Pfeffer K, Müller-Hermelink HK, Wirth T: **Up-regulation of BOB.1/OBF.1 expression in normal germinal center B cells and germinal center-derived lymphomas.** *Am J Pathol* 2000, **156**:501-507.

[175]The International Non-Hodgkin's Lymphoma Prognostic Factors Project: **A predictive model for aggressive non-Hodgkin's lymphoma.** *The New England journal of medicine* 1993, **14**:987-994.

[176]Alizadeh A, Eisen M, Davis RE, Ma C, Sabet H, Tran T, Powell JI, Yang L, Marti GE, Moore DT, Hudson JRJ, Chan WC, Greiner T, Weisenburger D, Armitage JO, Lossos I, Levy R, Botstein D, Brown PO, Staudt LM: **The lymphochip: a specialized cDNA microarray for the genomic-scale analysis of gene expression in normal and malignant lymphocytes.** *Cold Spring Harb Symp Quant Biol* 1999, **64**:71-78.

[177]Segal MR: **Microarray gene expression data with linked survival phenotypes: diffuse large-B-cell lymphoma revisited.** *Biostatistics* 2006, **7**:268-285.

[178]Simon R, Radmacher MD, Dobbin K, McShane LM: **Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification.** *J Natl Cancer Inst* 2003, **95**:14-18.

[179]Lossos IS: **Molecular pathogenesis of diffuse large B-cell lymphoma.** *J Clin Oncol* 2005, **23**:6351-6357.

[180]Wu G, Keating A: **Biomarkers of potential prognostic significance in diffuse large B-cell lymphoma.** *Cancer* 2006, **106**:247-257.

[181]Lossos IS, Morgensztern D: **Prognostic biomarkers in diffuse large B-cell lymphoma.** *J Clin Oncol* 2006, **24**:995-1007.

[182]Huang JZ, Sanger WG, Greiner TC, Staudt LM, Weisenburger DD, Pickering DL, Lynch JC, Armitage JO, Warnke RA, Alizadeh AA, Lossos IS, Levy R, Chan WC: **The t(14;18) defines a unique subset of diffuse large B-cell lymphoma with a germinal center B-cell gene expression profile.** *Blood* 2002, **99**:2285-2290.

[183]Iqbal J, Sanger WG, Horsman DE, Rosenwald A, Pickering DL, Dave B, Dave S, Xiao L, Cao K, Zhu Q, Sherman S, Hans CP, Weisenburger DD, Greiner TC, Gascoyne RD, Ott G, Müller-Hermelink HK, Delabie J, Braziel RM, Jaffe ES, Campo E, Lynch JC, Connors JM, Vose JM, Armitage JO, Grogan TM, Staudt LM, Chan WC: **BCL2 translocation defines a unique tumor subset within the germinal center B-cell-like diffuse large B-cell lymphoma.** *Am J Pathol* 2004, **165**:159-166.

[184]**Molmine J-Express** [http://www.molmine.com/]

[185]**GeneSifter** [http://www.genesifter.net]

[186]**Agilent Technologies GeneSpring Analysis Platform** [http://www.chem.agilent.com/Scripts/Generic.ASP?IPage=35082]

[187]**Jakarta Struts** [http://struts.apache.org/]

[188]**Apache MyFaces** [http://myfaces.apache.org/]

[189]**Biocarta Pathways** [http://www.biocarta.com/genes/]

[190] Salomonis N, Hanspers K,Zambon AC,Vranizan K,Lawlor SC, Dahlquist KD, Doniger SW, Stuart J, Conklin BR, Pico AR: **GenMAPP 2: new features and resources for pathway analysis.** *BMC Bioinformatics* 2007, **8**:217.

Eigene Publikationen

1. **Weniger M**, Engelmann JC, Schultz J.
   Genome Expression Pathway Analysis Tool - Analysis and visualization of microarray gene expression data under genomic, proteomic and metabolic context.
   BMC Bioinformatics. 2007 Jun 2;8:179.

2. Aus den Ergebnissen der Diplomarbeit:
   Rothganger J, **Weniger M**, Weniger T, Mellmann A, Harmsen D.
   Ridom TraceEdit: a DNA trace editor and viewer.
   Bioinformatics. 2006 Feb 15;22(4):493-4.

Danksagung

Bei Herrn Prof. Dr. Jörg Schulz möchte ich mich für die Betreuung der Doktorarbeit bedanken. Neben seinen Anregungen und Ideen gab er mir die Gelegenheit, meine eigenen Vorstellungen umsetzen zu können. Für diese Freiheiten und die Möglichkeit, die Ergebnisse auf verschiedenen Tagungen präsentieren zu können, bin ich sehr dankbar.

Herrn Prof. Dr. Rainer Spang möchte ich für seine Bereitschaft zur Übernahme des Zweitgutachtens und die freundliche Zusammenarbeit danken.

Mein weiterer Dank gilt Herrn Prof. Dr. Thomas Dandekar und allen Mitarbeitern des Lehrstuhls für Bioinformatik an der Universität Würzburg für ihre Motivation und Unterstützung während meiner Zeit dort. Besonders danken möchte ich Dr. Tobias Müller für die interessanten Diskussionen zur statistischen Auswertung von Microarraydaten und Julia Engelmann für die Hilfe bei der statistischen Programmierung. Philipp Seipel möchte ich danken für die Diskussionen rund um Programmierkonzepte, Java und den Rechencluster und für die Hilfe bei der Behebung von Sicherheitsrisiken. Mein Dank gilt auch allen studentischen Hilfskräften, Diplomanden und Praktikanten, die zu dieser Arbeit beigetragen haben.

Dem IZKF Würzburg danke ich für die Ermöglichung des Projekts und der interdisziplinären Zusammenarbeit. Besonders danken möchte ich Frau Dr. Susanne Kneitz für die Unterstützung in der Microarraydatenanalyse.

Bei meinen Freunden möchte ich mich für die Zerstreuung nach langen Arbeitstagen bedanken. Danke auch all denen, welche die Arbeit korrekturgelesen haben.

Ganz besonders möchte ich mich bei meiner Familie und Ulrike bedanken, ohne deren Unterstützung diese Arbeit nicht möglich gewesen wäre.

Lebenslauf

Geburtstag	13.02.1978
Geburtsort	Würzburg
Staatsbürgerschaft	deutsch

06.1997	Wirtschaftsgymnasium, Kaufmännische Schule Tauberbischofsheim Abitur
11.1997 - 08.1998	PzBtl. 363, Külsheim Wehrdienst
10.1998 – 06.2004	Julius Maximilians Universität Würzburg Studium der Informatik
10.2000	Vordiplom in Informatik
05.2004	Julius Maximilians Universität Würzburg Diplom in Informatik Diplomarbeit: Qualitätsbasierte Analyse von DNA-Traces
09.2000 - 02.2003	wissenschaftliche Hilfskraft Lehrstuhl für Informatik II (Programmiersprachen und Programmiermethodik), Universität Würzburg
03.2003 – 12.2003	wissenschaftliche Hilfskraft Lehrstuhl für Informatik VI (Künstliche Intelligenz und angewandte Informatik), Universität Würzburg
01.2004 – 12.2006	Software Entwicklung Ridom GmbH, Würzburg
07.2004 - 04.2007	Wissenschaftlicher Mitarbeiter Lehrstuhl für Bioinformatik, Universität Würzburg
seit 05.2007	Software Entwicklung methodpark Software AG, Erlangen

Erklärung

Hiermit erkläre ich ehrenwörtlich, dass ich die Dissertation selbständig angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe. Die Dissertation wurde bisher weder vollständig noch teilweise einer anderen Hochschule mit dem Ziel, einen akademischen Grad zu erwerben, vorgelegt. Ich erkläre weiterhin, dass ich außer meinem Diplom in Informatik an der Universität Würzburg keine weiteren akademischen Grade erworben habe oder zu erwerben versucht habe.

_____

Würzburg, den