Data-Warehouse- und Mapping-basierte Datenintegrationsplattformen in der Bioinformatik

Toralf Kirsten

Leipzig, Dezember 2007

Interdisziplinäres Zentrum für Bioinformatik (IZBI) Leipzig

Für meine Eltern, Karin und Dieter Kirsten,
sowie meine Lieben, Katja und Linus

Zusammenfassung

Neue experimentelle Technologien und Möglichkeiten des Austausches von Informationen haben den Bereichen der Biologie und Bioinformatik zu einer rasanten Entwicklung verholfen. Stand gegen Ende des letzten Jahrtausends noch die Sequenzierung ganzer Genome im Mittelpunkt, liegt der Fokus heute auf dem Verständnis der "molekularen Maschine", die durch die Wechselwirkungen und Interdependenzen zwischen den molekularen Objekten (z.B. Gene und Proteine) determiniert wird. Dazu werden Daten weltweit von verschiedensten Organisationen und Institutionen erzeugt, gesammelt und in unterschiedlichen Datenquellen verwaltet. Die unterschiedliche organisatorische und zeitliche Entwicklung führt zu einer Heterogenität der Datenquellen, die nicht nur auf deren Formate beschränkt bleibt, sondern auch deren Struktur und vor allem deren Semantik umfasst. Eine Datenintegration, die einerseits das Datenvolumen beachtet und andererseits die Heterogenität der Datenquellen überwindet, ist von grundlegender Bedeutung für eine umfangreiche und effiziente Analyse in diesen Bereichen.

Diese Dissertation beschäftigt sich mit der Konzeption und dem Aufbau von Plattformen zur Integration von Daten im Bereich der Bioinformatik, mit denen eine effiziente und zielgerichtete Datenanalyse unterstützt wird. Die Daten sind dabei einerseits das Resultat verschiedener molekularbiologischer Experimente und andererseits Inhalt verschiedenster Datenquellen.

Im *Genetic Data Warehouse* (*GeWare*) werden experimentelle Daten zentral zusammengefasst, die mit Hochdurchsatz-Technologien zur Untersuchung der Genexpression erzeugt wurden. Assoziiert zu diesen experimentellen Daten, kann *GeWare* Metadaten speichern, die das Experiment aus aufbau- und ablauforganisatorischer Sicht beschreiben und damit nachvollziehbar und reproduzierbar machen. Dazu bietet die Plattform so genannte *Annotation Templates*, die eine Menge von Kategorien strukturieren, für die die atomaren Annotationswerte in Bezug auf die experimentelle Beschreibung erfasst werden. Das Konzept der *Templates* kombiniert eine für den Benutzer größtmögliche Flexibilität bei der Definition, Modifikation und Nutzung der Templates mit einem für die Plattformadministration vernachlässigbaren Aufwand. Mit ihnen können (selbst nachträgliche) Anpassungen an ex-

perimentspezifische Erfordernisse vorgenommen werden, ohne dass Änderungen am zugrunde liegenden Datenmodell notwendig werden. Darüber hinaus eignet sich das Konzept der *Annotation Templates* auch zur Aufnahme klinischer Parameter, z.B. aus einem Studienverwaltungssystem. Zusätzlich ist die *GeWare*-Plattform mit einem Mediator gekoppelt, der Daten aus ausgewählten Quellen virtuell integriert und damit eine kombinierte und iterative Analyse der experimentellen Daten mit denen der öffentlich verfügbaren Quellen ermöglicht. In diese Integrationslösung ist die Software SRS (Sequence Retrieval System) eingebunden, die auf Basis einer umfangreichen Wrapper-Bibliothek vor allem den Zugriff auf die angebundenen Datenquellen sicherstellt. Kern der Integrationslösung ist eine zentrale Mapping-Datenbank, die Mengen von Korrespondenzen (Mappings) zwischen den Objekten/Instanzen der integrierten Quellen aufnimmt und der effizienten Anfrageverarbeitung dient.

Die *BioFuice*-Plattform nutzt Mappings, um Daten aus privaten und frei verfügbaren Datenquellen sowie Ontologien im Bereich der Bioinformatik zu integrieren. Die Mappings repräsentieren hierbei Korrespondenzen zwischen Objekten spezifischer Typen (z.B. Gen, Protein), die sowohl innerhalb einer Datenquelle als auch zwischen unterschiedlichen Quellen bestehen. Mengenbasierte Operatoren übernehmen die Ausführung der Mappings und können in Skripten zur Abbildung von ad-hoc Workflows zusammengefasst werden. *BioFuice* bietet ein mächtiges GUI, in der Anfragen verschiedenartig formuliert und ausgeführt werden. Dazu zählen neben der freien Skriptprogrammierung und der Abarbeitung von parametrisierten Skripten insbesondere die Formulierung und Ausführung einer Stichwortsuche sowie modellbasierter Anfragen. Letztere erfordern eine automatische Transformation in ausführbare Skripte. Darüber hinaus bietet *BioFuice* einen Datenexport in für die Bioinformatik spezifische Datenformate und eine Schnittstelle zur statistischen Software R, mit der die integrierten Daten einer statistischen Analyse zugeführt werden können.

Mit *GeWare* und *BioFuice* wurden auf Basis unterschiedlicher Anforderungen zwei Plattformen konzipiert und aufgebaut, die im Bereich der Bioinformatik Daten aus unterschiedlichen Quellen integrieren und für umfangreiche Analysen nutzbar machen. Die Plattformen wurden in verschiedenen Projekten verwendet und konnten die dort gestellten Anforderungen abdecken.

Leipzig, Dezember 2007 Toralf Kirsten

Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit am Interdisziplinären Zentrum für Bioinformatik (IZBI) der Universität Leipzig in den Jahren 2002 bis 2007. Mein Dank gilt vor allem meinem Mentor, dem Leiter der Arbeitsgruppe *Datenbanken und Datenintegration* am IZBI, Prof. Dr. Erhard Rahm, für die mir in jeder Hinsicht zuteil gewordene Unterstützung. Er gab mir die Möglichkeit zur Promotion. Den Herren Prof. Dr. Markus Löffler, wissenschaftlicher Leiter des IZBI, und PD Dr. Hans Binder, Geschäftsführer des IZBI, danke ich für das mir entgegengebrachte Vertrauen und die finanzielle Unterstützung während meiner Zeit am IZBI. Letzterer verhalf mir auf Basis gemeinsamer Untersuchungen zu Einblicken in die Geheimnisse der sequenzspezifischen Messung von RNA-Konzentrationen auf der Grundlage der Microarray-Technologie. Für diese äußerst interessante Zeit in einem kreativen Umfeld bin ich ihm sehr dankbar.

Dank sagen möchte ich auch den Mitarbeitern des IZBI für das kollegiale Miteinander. Insbesondere möchte ich die Mitarbeiter der Arbeitsgruppe *Datenbanken und Datenintegration* Dr. Hong-Hai Do, Michael Hartung und Jörg Lange hervorheben, mit denen ich ein Arbeitszimmer teilte und an gemeinsamen Projekten arbeitete. Vor allem durch und mit Dr. Hong-Hai Do habe ich in der langjährigen Zusammenarbeit am IZBI viel Neues erlernt. Frau Christine Körner und Herrn Jörg Lange danke ich für die Implementierung von wichtigen Teilen des *GeWare*-Systems, die Gegenstand ihrer Diplomarbeiten waren. Mit letzterem verbindet mich zudem eine zweijährige Zusammenarbeit mit dem Ziel der Anwendung der *GeWare*-Plattform bei der Untersuchung molekularer Mechanismen in zwei deutschlandweiten klinischen Studien. Herrn Dr. Jörg Galle danke ich für die Zusammenarbeit im Bereich der Genexpressionsanalyse unter Nutzung der *GeWare*-Plattform, die für die Forschungsgruppen um Frau Prof. Dr. Gabriele Aust (Anatomie) und Frau Dr. Anja Saalbach / Herr Dr. Ulf Anderegg (Hautklinik) der Medizinischen Fakultät an der Universität Leipzig durchgeführt wurde. Den Herren Dr. Ulf-Dietrich Braumann und Dr. Jens-Peer Kuska danke ich für ihre Unterstützung bei Fragen und Problemen rund um LaTeX, den Herren Patrick Scheibe und Nico Scherf für Ihre moralische Unterstützung.

Mein ebenso herzlichster Dank gilt den Mitarbeitern des Lehrstuhls Datenbanken an der Universität Leipzig unter Leitung von Herrn Prof. Dr. Erhard Rahm. Insbesondere möchte ich mich bei den Herren Andreas Thor, David Aumüller und Nick Golovin für die kreative und tolle Zusammenarbeit im Projekt *iFuice* und den daraus resultierenden Anwendungen bedanken. Auch den Mitarbeitern des Lehrstuhls Bioinformatik an der Universität Leipzig unter Leitung von Herrn Prof. Dr. Peter F. Stadler gilt mein Dank. Ferner danke ich den Herren PD Dr. Knut Krohn und Dr. Markus Eszlinger vom Interdisziplinären Zentrum für klinische Forschung (IZKF) und der Medizinischen Fakultät der Universität Leipzig für biologische Einblicke in die Welt der Genexpressionsanalyse. Nicht vergessen möchte ich an dieser Stelle, die Frauen Andrea Hesse und Petra Pregel sowie Herrn Jens Steuck, die mir bei administrativen Aufgaben hilfreich zur Seite standen.

Mein tiefster Dank gilt meinen Eltern Karin und PD Dr. Dieter Kirsten, meiner Lebensgefährtin Katja Rudert und meinem Sohn Linus, denen ich diese Arbeit widme. Sie standen mir auf meinem bisherigen Weg stets hilfreich zur Seite, obwohl sie viele Einschnitte und Wochenenden ohne meine Anwesenheit hinnehmen mussten. Ohne ihre Unterstützung sowohl in moralischer als auch finanzieller Hinsicht hätte ich diesen Weg wohl kaum beschreiten können.

Letztlich danke ich herzlich den Gutachtern der Dissertation und ihren zahlreichen Hinweisen, die zu einer stetigen Verbesserung der Arbeit geführt haben.

Leipzig, Dezember 2007 Toralf Kirsten

Inhaltsverzeichnis

TEIL I EINFÜHRUNG UND MOTIVATION 1

***Kapitel* 1 Datenintegration in der Bioinformatik 2**

1.1 Motivation 2
1.2 Offene Probleme 4
1.3 Wissenschaftlicher Beitrag der Arbeit 10
1.4 Gliederung der Arbeit 13

***Kapitel* 2 Einordnung und generelle Formen der Datenintegration 16**

2.1 Überblick 16
2.2 Datenintegration mit und ohne Verwendung einer homogenisierten Sicht 18
2.3 Arten der Instanzdatenintegration 31
2.4 Zusammenfassung 34

TEIL II DATA-WAREHOUSE-BASIERTE DATENINTEGRATION 37

***Kapitel* 3 Grundlagen der Genexpressionsanalyse 38**

3.1 Biologische Grundlagen 38
3.2 Microarray-basierte Genexpressionsanalyse 41
3.3 Chip-basierte Mutationsanalyse 44
3.4 Eigenschaften resultierender Genexpressions- und Mutationsdaten 45
3.5 Zusammenfassung 46

Kapitel 4 Vergleich von datenbankgestützten Analyseplattformen für Microarray-Experimente 48

4.1 Motivation . . . 48
4.2 Evaluierungskriterien . . . 49
4.3 Systemevaluierung . . . 56
4.4 Zusammenfassung . . . 67

Kapitel 5 Die Datenintegrations- und Analyseplattform GeWare 68

5.1 Motivation . . . 68
5.2 Systemarchitektur . . . 69
5.3 System Workflows . . . 73
5.4 Data Warehouse Schema . . . 74
5.5 Integration experimenteller Metadaten . . . 77
5.6 Microarray-basierte Genexpressionsanalyse . . . 82
5.7 Analyseintegration . . . 88
5.8 Abgrenzung zu verwandten Systemen . . . 90
5.9 Zusammenfassung . . . 91

Kapitel 6 Sequenzbasierte Analysen von Oligo-Intensitäten auf Basis der GeWare-Plattform 93

6.1 Motivation . . . 93
6.2 Besondere Projekt- und Analyseanforderungen . . . 95
6.3 Integration von Sequenzdaten und Analyseroutinen . . . 96
6.4 Ausgewählte Analyseergebnisse . . . 98
6.5 Zusammenfassung . . . 106

Kapitel 7 Die GeWare-Plattform im Anwendungsbereich klinischer Studien 108

7.1 Motivation . . . 108
7.2 Projektumgebung und spezifische Anforderungen . . . 110
7.3 Plattformarchitektur . . . 113
7.4 Integration von patientenbezogenen Annotationsdaten . . . 115
7.5 Übergreifende Analysen . . . 117
7.6 Zusammenfassung . . . 118

TEIL III MAPPING-BASIERTE DATENINTEGRATION 119

***Kapitel* 8 Hybride Integration molekularbiologischer Annotationsdaten 120**

8.1 Motivation . . . 120
8.2 Analyseszenarien . . . 122
8.3 Architektur im Überblick . . . 123
8.4 Metadatenverwaltung . . . 126
8.5 Anfragebearbeitung im Query-Mediator . . . 129
8.6 Ausgewählte Performanzanalysen . . . 133
8.7 Zusammenfassung . . . 137

***Kapitel* 9 Semantische Peer-to-Peer-artige Datenfusion: Der iFuice-Ansatz 139**

9.1 Motivation . . . 139
9.2 Ein beispielhaftes Szenario . . . 140
9.3 Mappings und Mapping-Erzeugung . . . 142
9.4 Konzeptuelle Strukturen . . . 143
9.5 Operatoren . . . 146
9.6 Skriptbasierte Analyse . . . 151
9.7 Zusammenfassung . . . 153

***Kapitel* 10 BioFuice: iFuice in der Bioinformatik 154**

10.1 Motivation . . . 154
10.2 Systemarchitektur . . . 155
10.3 Das *RiFuice*-Paket zur statistischen Analyse . . . 158
10.4 Interaktive Anfragen mit BioFuice Query . . . 161
10.5 Ausgewählte Anwendungsszenarien . . . 164
10.6 Zusammenfassung . . . 169

***Kapitel* 11 Verwandte Integrationsansätze 170**

11.1 Überblick . . . 170
11.2 Columba . . . 173
11.3 GenMapper . . . 174
11.4 Sequence Retrieval System . . . 175
11.5 Das BioFast-Projekt . . . 178
11.6 Zusammenfassung . . . 179

TEIL IV ZUSAMMENFASSUNG 181

***Kapitel* 12 Fazit und Ausblick** **182**
12.1 Fazit und Beitrag der Arbeit 182
12.2 Ausblick 185

Anhang **189**
A Evaluierte Microarray-Plattformen 189
B Daten zur sequenzbasierten Analyse von Oligo-Intensitäten . . 190
C Anfrageformulierung und -transformation von Annotationsanalysen in *GeWare* 207
D Anfrageformulierung und -transformation in *BioFuice* 211
E Document Type Definitionen zum XML-basierten Datenaustausch mit *BioFuice* 217

Literaturverzeichnis **219**

Stichwortverzeichnis **247**

Lebenslauf und wissenschaftlicher Werdegang des Verfassers **249**

Dissertationsbzogene bibliographische Daten **251**

Selbständigkeitserklärung **253**

Abbildungsverzeichnis

1.1 Probleme der Datenverwaltung und -integration im Bereich der Bioinformatik 4

2.1 Klassifikation von Integrationsformen 18
2.2 Verwendung eines applikationsspezifischen globalen Schemas . 19
2.3 Nutzung eines generischen globalen Schemas 22
2.4 Datenintegration auf Basis einer globalen Ontologie 24
2.5 Schemaintegration mit Verzicht auf ein globales Schema am Beispiel von PDMS 28
2.6 Arten der Integration von Instanzdaten 32

3.1 Grundlegende Bestandteile einer eukaryotischen Zelle 39
3.2 Schematische Darstellung der Genexpression in einer eukaryotischen Zelle 40
3.3 Schematischer Ablauf eines Microarray Experiments 43
3.4 Multidimensionalität von Genexpressions- und Mutationsdaten 46

4.1 Evaluierungskriterien 49
4.2 Formen der Analyseintegration 55

5.1 *GeWare* Systemarchitektur im Überblick 70
5.2 Abstrakte Import- und Analyseprozesse in *GeWare* 73
5.3 Multidimensionales Data Warehouse Schema im Überblick . . 75
5.4 Generisches Schema zur Verwaltung der experimentellen Metadaten 79
5.5 Spezifikation und Analyse der experimentellen Metadaten . . . 81
5.6 Ausgewählte Visualisierungsformen von Analyseergebnissen . . 86
5.7 Analyse der Genexpression unter Nutzung des ausgewählten molekularbiologischen Netzwerkes "Jak Stat Pathway from IL6 Rezeptor" 87
5.8 Transparente Analyseintegration von BioConductor Funktionen 89

6.1 Vergleich von Sequenzen und gemessenen Intensitäten von PM und MM von Oligos eines Affymetrix Microarrays 94
6.2 Häufigkeit von Oligos in Bezug auf die Anzahl der Nukleotide in den Oligo-Sequenzen 102
6.3 Häufigkeit von Oligos in Bezug auf die Position der Nukleotide in den Oligo-Sequenzen 103
6.4 Oligo-Intensitäten (PM/MM) in Abhängigkeit von der Mittelbase 105
6.5 Durchschnittliche Intensitäten bezogen auf das Mitteltripel . . 107

7.1 Projektumgebung und resultierende Daten 111
7.2 Kopplung der Systeme eRN und *GeWare* 114
7.3 Definition, Transfer und Auswertung von patientenbezogenen Annotationsdaten 116
7.4 Kombinierte Analysen 118

8.1 Annotations- und Mappingdaten in LocusLink 121
8.2 Analyseszenarien 123
8.3 Integrationsansatz und Komponenten im Überblick 124
8.4 Metadatenverwaltung in ADM- und Mapping-Datenbank . . . 127
8.5 Anfrageformulierung auf der automatisch generierten Web-Oberfläche 130
8.6 Schritte zur Erstellung des Anfrageplans 132
8.7 Ergebnisse für Projektions- und Selektionsanfragen 133
8.8 Performanz von Selektionsanfragen in SRS und MySQL 135
8.9 Performanz von Projektions- und Selektionsanfragen in SRS und MySQL 136
8.10 Performanz von Selektionsanfragen an die Datenquelle Ensembl 137

9.1 Beispielhaftes Analyseszenario im Bereich der Bioinformatik . 141
9.2 Das Source-Mapping-Modell und das semantische Domänenmodell für das dargestellte Szenario 144
9.3 *iFuice*-Datenstrukturen als Grundlage der operatorgesteuerten Verarbeitung 147

10.1 Systemarchitektur von BioFuice 156
10.2 Interaktive Anfrageformulierung mit BioFuice Query 162
10.3 Metadaten-Modelle im Bereich der Genexpression 165
10.4 Metadaten-Modelle zur Integration von Proteininteraktionen . 166
10.5 Metadaten-Modelle im Analysebereich von ncRNA, miRNA und snRNA 168

11.1 Einordnung von verwandten Integrationsansätzen 171

C.1 Syntaxdiagramm der erzeugten SRS-Projektionsanfrage 208
C.2 Syntaxdiagramm der erzeugten SRS-Selektionsanfrage 209

Tabellenverzeichnis

2.1 Gegenüberstellung von Integrationsformen mit und ohne Schemaintegration 30
2.2 Vergleich von Arten zur Integration von Instanzdaten 33

4.1 Relevante Datenarten und deren Charakteristik 50
4.2 Technische Implementierung 57
4.3 Unterstützung von Bild- und Expressionsdaten 58
4.4 Spezifikation und Speicherung experimenteller Metadaten . . . 59
4.5 Integrierte Genannotationen öffentlicher Datenquellen 60
4.6 Unterstützte Normalisierungsstrategien 61
4.7 Abfrage- und Berichtsmöglichkeiten 62
4.8 Implementierte Data Mining und statistische Methoden 63
4.9 Visualisierung 64
4.10 Wichtige Vor- und Nachteile der Systeme 66

6.1 Ergebnis des paarweisen Oligo-Sequenzvergleiches für den Chiptyp HG-U95Av2 99
6.2 Häufigkeit von Oligo-Äquivalenzklassen für den Chiptyp HG-U95Av2 101

9.1 Quellenspezifische Operatoren 149
9.2 Operatoren zur Navigation und Aggregation 150
9.3 Generische Operatoren 151

10.1 Verbindungs- und Ausführungsfunktionen im Überblick 158
10.2 Funktionen zum Metadaten-Management im Überblick 159
10.3 Import- und Exportfunktionen im Überblick 160
10.4 Mengengerüst zu Proteinen und Proteininteraktionen für die Spezies Homo Sapiens in BIND, DIP und MINT 167

A.1 Evaluierte datenbankgestützte Systeme für Microarray-Daten . 189
B.2 Überblick zu den Microarrays des Latin-Square-Experiments . 190

B.3 Gemessene Intensitäten äquivalenter Oligos 191
B.4 Oligo-Sequenzen der ausgewählten Äquivalenzklassen 191
B.5 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U95Av2 192
B.6 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U133A 193
B.7 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U133_Plus_ 2 194
B.8 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp MG-U74Av2 195
B.9 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp MOE430A 196
B.10 Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp Mouse430_2 197
B.11 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U95Av2 198
B.12 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U133A 199
B.13 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U133_Plus_2 200
B.14 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp MG-U74Av2 201
B.15 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp MOE430A 202
B.16 Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp Mouse430_2 203
B.17 PM/MM-Missverhältnis 204
B.18 Standardisierte Mitteltripel-Intensitäten 205

Teil I
Einführung und Motivation

Im Fokus dieser Arbeit stehen Ansätze und Plattformen für die Integration unterschiedlicher Daten im Bereich der Bioinformatik. Die Lösung des Datenintegrationsproblems in dieser Domäne ist von besonderer Bedeutung, um komplexe Analysen und deren effiziente Durchführung zu ermöglichen.

Der einführende Teil besteht aus zwei Kapiteln. Kapitel 1 motiviert die Notwendigkeit der Datenintegration in der Bioinformatik und zeigt die offenen und bislang unzureichend gelösten Probleme auf. Bezug nehmend darauf werden die wichtigsten Beiträge dieser Arbeit dargestellt. Kapitel 2 ordnet die Datenintegration in generelle Integrationsansätze ein und gibt einen Überblick zu allgemeinen Datenintegrationsansätzen.

Kapitel 1
Datenintegration in der Bioinformatik

1.1 Motivation

Die strukturellen Eigenschaften der Desoxyribonukleinsäure (DNS), die alle für die Vererbung notwendigen Informationen kodiert, beschrieben James Watson und Francis Crick [WC53] beruhend auf den Arbeiten von Wilkins, Stokes und Wilson [WSW53] sowie Franklin und Gosling [FG53] bereits im Jahr 1953. Diese Arbeiten waren ein Ausgangspunkt für weitreichende Forschungsarbeiten, die zum Ende des letzten Jahrtausends mit der Sequenzierung ganzer Genome (vgl. beispielsweise [FAW+95, ACH+00, VAM+01]) einen besonderen Höhepunkt erreichten. Insbesondere die stetig verbesserten Technologien, die zur Analyse des aus Zellen extrahierten genomischen Erbmaterials eingesetzt werden, machten eine solche rasante Entwicklung möglich. Dabei nahmen die Analysen durch den Einsatz von verbesserten Messmethoden und -techniken nicht nur an Genauigkeit zu. Vielmehr erlaubt die heutige Technologie umfangreichere Analysen. Beispielsweise wurde bis in die 1990er Jahre allein die Expression, d.h. die Aktivität, von einzelnen, spezifisch ausgewählten Genen quantifiziert. Dazu wurden Techniken wie die Polymerasekettenreaktion (engl. polymerase chain reaction - PCR) [KIHS92] verwendet. Dagegen ermöglicht die Microarray-Technologie [SSDB95, MLB+96] den Schritt von der einzelgenbasierten zur genomweiten Genexpressionsanalyse, in die gleichzeitig Tausende Gene einbezogen werden.

Die Anwendung solcher neuen Technologien macht elaborierte Analysen möglich, die beispielsweise der Erforschung von Wechselwirkungen zwischen molekularbiologischen Objekten (Gene, Proteine etc.), deren funktionalen Abhängigkeiten sowie deren Verhalten unter verschiedenen Rahmenbedingungen (z.B. bei Krankheiten bzw. speziellen Krankheitsstadien) dient. Diese Art von Grundlagenforschung ist für viele Bereiche bedeutsam; zum Beispiel kann sie im Bereich der Biomedizin eine an den Patienten individuell angepasste Medikamentation zur Behandlung von Krankheiten oder eine verbesserte Therapiesteuerung zum Ziel haben.

Neben dem Voranschreiten der Biotechnologie kommt der Entwicklung und dem Ausbau des Internets, getrieben durch dessen intensive Nutzung als globales, weltumspannendes Computernetzwerk, eine besondere Bedeutung zu. Insbesondere die dezentrale Bereitstellung von Informationen, die mit dem Internet vernetzt werden, macht neben vorhandenem Wissen auch neuartige Erkenntnisse einer breiten Schicht von Forschern und Interessierten schnell und einfach zugänglich. Die auf dieser Art und Weise veröffentlichten Daten, die biologische Systeme, deren Elemente und Wechselwirkungen beschreiben, spielen nicht nur bei der Interpretation und Validierung von erzielten Analyseergebnissen eine Rolle, z.B. für einen Vergleich von eigenen mit Ergebnissen anderer Forschergruppen. Vielfach kommen diese Daten auch bei der Planung von Experimenten zum Einsatz, sei es um die Ergebnisse anderer Gruppen durch eigene Nachforschungen zu überprüfen oder auf diesen aufbauend neue biologische Fragestellungen zu klären.

Aus diesen Entwicklungen resultiert ein erheblich gewachsenes Datenvolumen, das zum einen durch die im Labor ermittelten experimentellen Ergebnisse, die so genannten Primärdaten, und zum anderen sowohl durch die eigenen erzielten als auch durch die im Internet frei verfügbaren Analyseergebnisse (Sekundärdaten) und Beschreibungen determiniert wird. Das führt in Hinsicht auf eine effiziente Datenanalyse zu zwei großen Herausforderungen. Einerseits sind neue Analyseansätze und -algorithmen notwendig, die nicht nur spezifische Analyseziele verfolgen, sondern auch das große Volumen an Daten mit ihren vielfältigen Abhängigkeiten und damit gegenseitigen Beeinflussungen berücksichtigen. Andererseits führt insbesondere die Menge an unterschiedlichen Daten zu veränderten und neuen Anforderungen bezüglich der Datenverwaltung und -integration, die Gegenstand des nächsten Abschnittes sind.

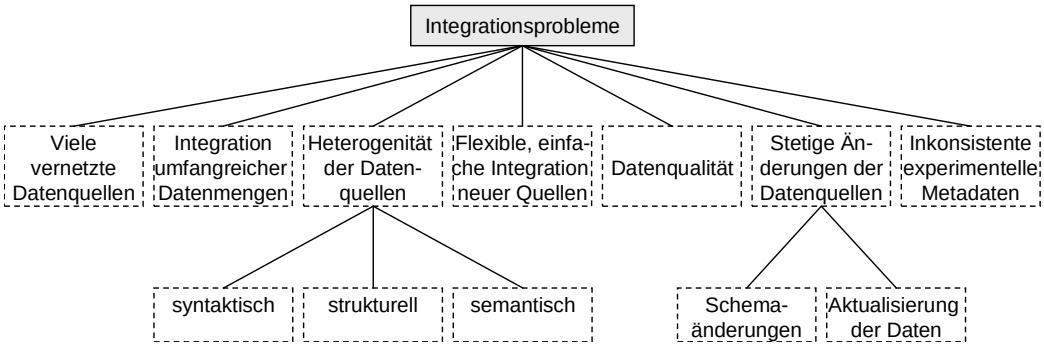


Abbildung 1.1: Probleme der Datenverwaltung und -integration im Bereich der Bioinformatik

1.2 Offene Probleme

Die Abbildung 1.1 zeigt offene Probleme im Überblick, die sich aus der Notwendigkeit der Datenverwaltung und -integration insbesondere für den Bereich der Bioinformatik ergeben.

Viele vernetzte Datenquellen. Die Forschungsbemühungen unterschiedlicher Institutionen und Organisationen haben dazu geführt, dass Daten in separaten, voneinander unabhängig entwickelten Datenquellen verwaltet werden. Das Journal "Nucleic Acids Research" listet einmal jährlich alle registrierten und inhaltlich beschriebenen Datenquellen auf; im Jahr 2006 enthielt die Aufstellung mehr als 850 Datenquellen [Gal06]. Viele dieser Datenquellen bieten verschiedenartige Anfragemöglichkeiten auf Basis einer Web-Schnittstelle, mit der auf die Daten frei zugänglich zugegriffen werden kann. Darüber hinaus sind viele Quellen zur umfassenden lokalen Auswertung als Kopie erhältlich.

Die Datenquellen enthalten entsprechend ihrer Ausrichtung unterschiedliche Arten von Daten. Ensembl [HAC+05], NCBI Entrez [MOPT05] und UCSC Genome Browser [KBD+03] sind repräsentative Genom-Datenquellen, die Gene, Transkripte und Proteine u.a. von unterschiedlichen Spezies beschreiben. Die Datenquellen UniProt [Con07] und die Protein Data Bank (PDB) [BWF+00] enthalten Daten über Proteine und Proteinstrukturen. BIND [BDW+01], MINT [ZMPQ+02] und DIP [XSD+02] fokussieren auf Proteininteraktionen, während in KEGG [KG00] vor allem Daten über molekulare Netzwerke (so genannte Pathways) gespeichert sind. Eine andere Gruppe von Datenquellen, zu denen beispielsweise Medline/PubMed[1] gehören, bein-

[1] `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed`

haltet eine große Anzahl von Publikationen, die vor allen Dingen mit medizinischer und biologischer Ausrichtung in unterschiedlichen Journalen erschienen sind. Eine zunehmend wichtiger werdende Art von Datenquellen sind Ontologien, die mit einem definierten Vokabular (so genannte Konzepte) die semantische und eindeutige Beschreibung der Eigenschaften von molekularbiologischen Objekten (z.B. Gene, Proteine) ermöglichen. Eine große Anzahl von Ontologien werden unter der Initiative "Open Biomedical Ontologies"[2] (OBO) zusammengefasst. Ein repräsentativer Vertreter ist die GeneOntology [HCI+04], mit der molekulare Funktionen, biologische Prozesse und zelluläre Komponenten beschrieben werden können.

Eine wichtige Eigenschaft dieser Datenquellen besteht darin, dass ihre enthaltenen Objekte (Instanzdaten) auf Objekte anderer Datenquellen verweisen. Diese Art der Objekt-Referenzierung basiert auf der Verwendung von Objekt=Identifikatoren, die für eine Quelle eindeutig sind. Beispielsweise ist das Protein-Objekt mit der ID *ENSP00000171757* der Datenquelle Ensembl mit dem Protein-Objekt mit der ID *O00398* (Putative P2Y purinoceptor 10) der Quelle SwissProt verbunden. Zugleich ist es mit den molekularen Funktionen *GO:0001584* (rhodopsin-like receptor activity) und *GO:0045028* (purinergic nucleotide receptor activity, G-protein coupled) der Datenquelle GeneOntology assoziiert. Die Semantik der Objektassoziationen ist in vielen Fällen nicht Teil der Datenquellen, sondern wird statisch auf der Web-Oberfläche angegeben, auf der die Objekt-Referenzen gruppiert als Web-Links dargestellt werden. Die Objekt-Referenzen tragen dazu bei, dass die relativ hohe Anzahl umfassender und registrierter Datenquellen im Bereich der Bioinformatik hochgradig miteinander vernetzt sind.

Integration umfangreicher Datenmengen. Die registrierten und öffentlich zugänglichen Datenquellen umfassen Daten für eine zunehmend große Anzahl von molekularbiologischen Objekten. Beispielsweise beinhaltet Ensembl Daten zu derzeit 27 unterschiedlichen Spezies[3]. Allein für die Spezies "Homo Sapiens" sind mehr als 31.100 Gene vorhanden, die mit mehr als 51.900 Transkripten assoziiert sind und die wiederum zu mehr als 43.600 Translationen (Proteine) führen. Neben diesen Typen von Objekten verwaltet Ensembl mehr als 277.900 Exons, die mit den Translationen und Transkripten assoziiert sind sowie Daten für viele andere Objekttypen, wie z.B. spezielle genetische Marker, unterschiedliche Protein-Features und Sequenz-Alignments.

Neben den öffentlich verfügbaren Datenquellen enthalten auch vermehrt

[2] `http://obo.sourceforge.net/main.html`
[3] Eine Auflistung der Genome findet sich unter `http://www.ensembl.org`.

lokale Quellen große Datenmengen, die durch die Anwendung neuer experimenteller Technologien entstehen. Dazu gehört beispielsweise die Hochdurchsatz-Technologie in Form von Microarrays (Chips) zur genomweiten Untersuchung der Genexpression (vgl. Kapitel 3). Ein aktueller Chip (Typ: HG-U133 Plus 2) des Herstellers Affymetrix enthält mehr als 600.000 unterschiedliche Sequenzen, die mit mehr als 50.000 Transkripten und mehr als 30.000 Genen assoziiert sind. Mit jedem Chip, der in eine solche Expressionsstudie einbezogen wird, fallen für jede dieser Sequenzen Werte an, die in nachgelagerten Prozessen normalisiert und zur Menge der Transkripte und Gene aggregiert werden, bevor eine umfangreiche Analyse starten kann. Die Anwendung unterschiedlicher Normalisierungs- und Analyseverfahren erhöht das produzierte und zu verwaltende Datenvolumen weiter.

Das vorhandene Datenvolumen von öffentlichen Quellen sowie die in experimentellen Untersuchungen produzierte Datenmenge muss sowohl bei der Integration als auch bei der Verwaltung dieser Daten beachtet werden, das als Voraussetzung für eine effiziente Analyse gilt.

Heterogenität der Datenquellen. Die getrennte organisatorische Entwicklung der Datenquellen ist ein Grund für deren Heterogenität, die sich nach [She98, VSSV02, LLB+05] vor allem in einer unterschiedlichen Syntax, Struktur und Semantik ausdrückt.

- **Syntax.** Die syntaktische Heterogenität resultiert aus unterschiedlichen Formaten, mit denen Daten beschrieben und gespeichert werden können. Das Spektrum reicht hierbei von relationalen Datenbanken, in denen die Daten strukturiert vorliegen, über die semi-strukturierte Speicherung in XML bis hin zu spezifischen proprietären Formaten und separierten flachen Dateien (engl. flat files/comma separated files). In der Bioinformatik haben sich unterschiedliche Formate etabliert. Das FASTA-Format wird meist für Sequenzdaten verwendet; andere proprietäre Formate, wie Genbank, sind quellenspezifisch und wurden vor allem durch die vermehrte Nutzung der Quelle populär. Für den Austausch der experimentellen Metadaten wird vielfach MAGE-ML, ein auf XML basierendes Format, verwendet. Relationale Datenbanken werden vielmehr für die Verwaltung der Daten als für deren Austausch genutzt. Eine Integration von Daten aus verschiedenen Formaten bedingt eine Datenkonvertierung unter Nutzung von speziellen Tools, wie z.B. XML-Parsern etc.

- **Struktur.** Neben der Verwendung unterschiedlicher Formate resultiert die getrennte organisatorische Entwicklung in verschiedenen Schemata,

die zur Beschreibung gleicher Realwelt-Objekte dienen. Die Schemata unterscheiden sich nicht nur in ihrem Typ (z.B. relational vs. hierarchisch), sondern auch in den verwendeten Schemaelementen (z.B. Tabellen und Attribute in einem relationalen Schema). Die Wahl eines Schemas bzw. die Schemabildung (Datenmodellierung) ist von vielen Faktoren abhängig. Dazu zählen beispielsweise Art und Umfang der zu speichernden Daten; während in einer Organisation die Verwaltung aggregierter Daten ausreichend ist, kann in einer anderen die atomare Speicherung aller Daten notwendig werden. Ebenso können die Objekte in Abhängigkeit der verfügbaren Daten in verschiedenen Institutionen mit unterschiedlichen Attributen beschrieben werden. Letztlich ist die verwendete Struktur auch vom biologischen Verständnis und den Präferenzen des Modellierers bei der Modellbildung abhängig. Eine Integration von Quellen, die verschiedene Schemata verwenden und damit über eine unterschiedliche Struktur verfügen, macht eine Integration der Schemata notwendig.

- **Semantik.** Semantische Interoperabilität setzt einen Konsens in Hinsicht auf das verwendete Vokabular in der Anwendungsdomäne voraus. Eine solche Abstimmung ist im Bereich der Bioinformatik noch nicht erreicht. Grundlegende Begriffe wie "Gen" werden in unterschiedlicher Weise interpretiert bzw. verwendet [Sch98]. Darüber hinaus finden für die Beschreibung der Objekte sowohl auf Schemaebene als auch auf Instanzebene unterschiedliche Namen für gleiche Sachverhalte Anwendung, genauso wie gleiche Namen für unterschiedliche Sachverhalte verwendet werden. Während die Namen der Schemaelemente (z.B. Namen der Tabellen und Attribute im relationalen Schema) den Kontext der gespeicherten Daten wiedergeben, repräsentieren die beschriebenen Eigenschaften (Instanzebene) den Kontext der molekularbiologischen Objekte. Hierbei kommen vermehrt kontrollierte Vokabulare, Taxonomien und Ontologien zur Anwendung, die eine semantische Interoperabilität sicherstellen helfen. Die Beachtung der Semantik der Objekte in den zu integrierenden Datenquellen ist unabdingbar, um den Integrationsprozess (z.B. zur Bildung eines globalen Schemas) erfolgreich zu gestalten.

Eine erweiterte Klassifikation von Ursachen und Formen der Heterogenität von Datenquellen wird in [LN07] unabhängig vom Bereich der Bioinformatik vorgestellt und diskutiert.

Datenqualität. Die Qualität der Daten, die z.B. aus Labor-Experimenten, Analysen und deren Interpretationen resultieren, wird von mehreren Faktoren beeinflusst. Neben inhaltlichen Fehlern, die beispielsweise aus Fehlern

im Versuchsaufbau und -durchführung innerhalb eines Experiments sowie Fehlern im Analyseprozess resultieren können, bestimmen oftmals fehlende und gegensätzliche Werte (von Objektbeschreibungen zwischen zwei Datenquellen) die Datenqualität. Mehrfach repräsentierte Objekte (Duplikate) innerhalb einer Datenquelle beeinflussen die Datenqualität ebenso wie die Übernahme von Daten aus anderen Datenquellen.

Die Datenqualität bestimmt im bedeutendem Maße die auf sie aufbauenden Interpretationen sowie die Ergebnisse von Analysen. Daher beschäftigen einige Organisationen, die ihre Datenquellen für die Aufnahme von Daten anderer Forschergruppen öffnen, so genannte *Kuratoren*, die die eingegangenen Daten auf Basis von Regeln sowie ihrer Kenntnisse und Erfahrungen auf Richtigkeit überprüfen. Damit wird nicht nur eine höhere Datenqualität angestrebt. Vielmehr kann daraus eine höhere Akzeptanz und Vertrauenswürdigkeit auf Seiten der Benutzer resultieren. Daher ist es notwendig, die Herkunft der integrierten Daten aufzuzeigen [BKT00, BKT01].

Stetige Änderungen von Daten und Schemata. Die Daten der verschiedenen Quellen repräsentieren den zu einem Zeitpunkt geltenden Stand der Forschung. Neue Erkenntnisse führen zu Änderungen der Daten, die sowohl in Korrekturen der vorhandenen Daten als auch im Hinzufügen neuer Daten bestehen können. Die Frequenz der Aktualisierung der Daten ist spezifisch für eine Quelle. Öffentliche Datenquellen, wie beispielsweise GeneOntology oder NCBI Entrez Gene unterliegen täglichen Änderungen während andere einem Release-Konzept folgend (z.B. NetAffx [CST+04] im Drei-Monats-Rhythmus) Aktualisierungen vornehmen.

Zusätzlich ziehen neue oder veränderte Rahmenbedingungen und die sich daraus ergebenden Anforderungen eine Adaption der Datenquelle nach sich, die in einer Schemaänderung münden kann. Sowohl die häufige Änderung der Daten als auch die auftretende Schemaevolution sind bei der Datenintegration zu beachten.

Flexible und einfache Integration neuer Quellen. Untersuchungen im Bereich der Bioinformatik haben ihren Ursprung in biologischen Fragestellungen. Zur Lösung dieser biologisch motivierten Probleme tragen u.a. Analysen bei, für die Daten aus verschiedenen Quellen integriert werden müssen, sei es weil sie die Daten als Eingabe erwarten oder die Daten zur nachfolgenden Ergebnisdarstellung und Interpretation gebraucht werden. Die Anforderungen, d.h. welche Daten aus welchen Quellen notwendig sind, hängen von der verfolgten Fragestellung und den durchzuführenden Analysen ab und können deshalb sehr stark variieren. Eine Integration aller Quellen ist nicht

nur auf Grund des enormen Aufwandes unrealistisch, sondern auch deshalb unmöglich, weil evtl. nicht alle Datenquellen (z.B. private Quellen) zum Zeitpunkt einer Integration zur Verfügung stehen. Somit wird eine nachträgliche Integration von Datenquellen erforderlich, die einfach und flexibel, d.h. mit möglichst wenig Aufwand verbunden sein sollte.

Inkonsistente Erfassung experimenteller Metadaten. Die experimentellen Untersuchungen bedingen eine umfassende Beschreibungen des Experiments sowohl aus ablauf- als auch aufbauorganisatorischer Sicht, um eine Nachvollziehbarkeit und Reproduzierbarkeit zu gewährleisten. Typischerweise dient ein Laborbuch diesem Dokumentationszweck. Eine elektronische Erfassung hat darüber hinaus den Vorteil, dass die spezifizierten Daten rechnergestützte Auswertungen ermöglichen. Ziel dieser Auswertungen kann es einerseits sein, relevante Experimente oder interessante Teile daraus zu identifizieren, die im Weiteren Grundlage eigener Experimente und Analysen sind. Beispielsweise wird eine Gruppierung der experimentellen Daten möglich, zu deren Erstellung der Benutzer Bedingungen in Hinsicht auf die erfassten Untersuchungsbedingungen angibt (z.B. alle Teile eines Experiments, bei denen die Zellen aus dem untersuchten Gewebe einen bestimmten Status besitzen). Die gruppierten Daten können im Anschluss Eingang in verschiedene Analysen finden, die im einfachsten Fall die gruppierten Daten miteinander vergleichen. Andererseits erlauben die erfassten experimentellen Metadaten eine Interpretation der erzielten Analyseergebnisse. Insbesondere kann damit geklärt werden, inwieweit die Ergebnisse und speziell anormale Resultate (so genannte Ausreißer) von experimentellen Fehlern beeinflusst wurden und damit von diesen abhängen.

Die Art und Weise der Erfassung experimenteller Metadaten hat entscheidenden Einfluss auf ihre spätere Auswertung. Die verbale Beschreibung in Form von komplexeren Textkorpora, die unter Nutzung eines einzelnen Freitextfeldes spezifiziert wurde, bietet zwar größtmögliche Flexibilität in Bezug auf die aufzunehmende Beschreibung. Jedoch wird die Analyse der in dieser Form aufgenommenen Metadaten unnötig erschwert. Insbesondere kann keine Vergleichbarkeit hinsichtlich der zu beschreibenden Merkmale und der zu verwendenden Merkmalswerte sichergestellt werden. Daher sollte eine Menge von Parametern definiert werden, die das Experiment bzw. die Experimentserie einheitlich und atomar beschreiben. Zusätzlich sollte auf vordefinierte Wertebereiche zurückgegriffen werden, die terminologische Variationen bei der Verwendung von Begriffen soweit wie möglich reduzieren. Dazu können kontrollierte Vokabulare, Taxonomien oder Ontologien dienen, die frei verfügbar sind oder lokal erstellt werden.

1.3 Wissenschaftlicher Beitrag der Arbeit

Fokussiert auf die gezeigten offenen Probleme und den sich daraus ergebenden Anforderungen enthält diese Arbeit eine Menge von Beiträgen, die sich in die folgenden drei Bereiche zusammenfassen lassen.

Data-Warehouse-basierte Integrations- und Analyseplattform für Daten Microarray-basierter Experimente sowie deren Metadaten

Basierend auf den Anforderungen lokaler Forschungsgruppen, die vorrangig Genexpressionsanalysen unter Nutzung der Microarray-Technologie vornehmen, und den Ergebnissen einer durchgeführten Evaluierung zu bestehenden Integrationsplattformen für diese Art von experimentellen Daten wurde das Genetic Data Warehouse System (*GeWare*) konzipiert und entwickelt. Hierbei sind die folgenden Aspekte wesentlich:

- **Evaluierung ausgewählter Integrationsplattformen**: Getrieben durch unterschiedliche lokale Anforderungen wurden verschiedene Integrationsplattformen mit der Zielstellung entwickelt, die in aufwändigen Microarray-basierten Experimenten ermittelten Daten zu verwalten und ihre Analyse zu unterstützen. Um einen Überblick über die verwendeten Ansätze bzgl. der Datenverwaltung, -integration und -analyse in solchen Systemen zu bekommen, wurden acht publizierte Systeme anhand definierter Anforderungen untersucht. Zusätzlich diente die Evaluierung dazu, Problembereiche und Ansatzpunkte für Forschungsaktivitäten zu identifizieren.

- **Integration experimenteller Daten mit dem *GeWare*-System**: *GeWare* folgt dem Data Warehouse Ansatz [JLVV03] und verwendet ein multidimensionales Schema, um zentralisiert umfangreiche Genexpressions- und Mutationsdaten zu speichern, die auf Basis von Hochdurchsatz-Technologien erzeugt werden. Das multidimensionale Schema unterstützt flexibel zielgerichtete Analysen, z.B. durch die Selektion relevanter Experimente anhand ihrer Metadaten, und ist erweiter- und skalierbar.

- **Konsistente und autonome Erfassung experimenteller Metadaten**: *GeWare* nutzt so genannte *Annotation Templates*, in denen einheitlich alle relevanten Kategorien zusammengefasst sind, zu denen ein Benutzer Annotationswerte spezifiziert. Anhand der Template-Definition werden automatisch Web-Seiten generiert, so dass eine autonome, web-basierte Dateneingabe vorgenommen werden kann. Die bestehenden *Annotation Templates* können für neue Experimente wieder

verwendet und jederzeit erweitert werden. Dazu dient ein generisches Datenmodell als Teil des *GeWare*-Systems, mit dem sowohl die Template-Definitionen als auch die spezifizierten Annotationswerte gespeichert werden.
Vordefinierte Vokabulare dienen der konsistenten Verwendung von Begriffen und Bezeichnungen und sichern eine hohe Datenqualität der spezifizierten experimentellen Metadaten. Sie können in einem Template mit einzelnen Kategorien assoziiert werden. Das erleichtert eine spätere Analyse dieser Daten, z.B. zur Suche nach Chips, die unter gleichen Bedingungen verarbeitet wurden.

- **Integration klinischer Daten**: Die *Annotation Templates* bieten die notwendige Flexibilität, um ausgewählte patientenbezogene, klinische Daten aufzunehmen. Solche Daten werden in zwei deutschlandweiten klinischen Studien aus einem Studienverwaltungssystemen in ein vordefiniertes *Template* importiert. Eine Mapping-Tabelle, die die Korrespondenzen zwischen den verwendeten Patienten IDs des Studienverwaltungssystems und Chip IDs von *GeWare* enthält, erlaubt die Verbindung von patientenbezogenen und experimentellen Daten beim Datenimport. Damit sind übergreifende Analysen möglich, die experimentelle, Microarray-basierte Daten zusammen mit den erfassten klinischen Daten zu kombinieren.

- **Analysekopplung auf Basis zentraler Objektgruppen** *GeWare* verwendet sowohl Datenbank-Funktionen als auch existierende Software, um die integrierten Daten zu analysieren; für die meisten Analysemethoden wird auf die statistische Software R unter Nutzung des BioConductor-Paketes zurückgegriffen. Um die unterschiedlich integrierten Analysemethoden miteinander zu koppeln, nutzt *GeWare* abstrakte Objektgruppen als Kollektionen von Objekten (Gene, Clone und Chips) sowie Matrizen (Expressions- & Mutationsmatrizen). Damit können die Ergebnisse einer Analyse unmittelbar als Eingabe einer folgenden Analyse verwendet werden.

Hybride Integration öffentlicher Annotationsdaten und Anbindung an das *GeWare*-System

Ein weiterer wesentlicher Beitrag besteht in einem hybriden Integrationsansatz, mit dem Annotationsdaten zu molekularbiologischen Objekten aus öffentlichen Datenquellen für datenintensive Expressionsanalysen verwendbar gemacht werden. Der Ansatz weist die folgenden besonderen Merkmale auf:

- Die Expressionsdaten sind zusammen mit den experimentellen Metadaten und ausgewählten klinischen Daten physisch in dem Data Warehouse *GeWare* integriert und unterstützen schnelle Auswertungen.

- Die öffentlichen Annotationsdaten werden virtuell über einen Mediatoransatz integriert und bedarfsgesteuert für Analysen abgerufen. Für die einheitliche Anbindung der Datenquellen wird das vielfach angewendete Integrationssystem SRS (Sequence Retrieval System) [EHB03] der Fa. BioWisdom[4] genutzt, das hierfür eine umfangreiche Wrapper-Bibliothek bietet. Die Kopplung zwischen dem Data Warehouse *GeWare* und SRS erfolgt über einen Query-Mediator unter Nutzung einer Mapping-Datenbank, die Mengen von Korrespondenzen (Mappings) zwischen den Objekten (Instanzen) der integrierten Datenquellen beinhaltet.

- Im Kern des Ansatzes werden die Datenquellen sternförmig um eine ausgewählte zentrale Quelle angeordnet. Die Mapping-Datenbank enthält alle Mappings zwischen der zentralen Quelle und den Datenquellen. Die Mappings entstammen dabei einerseits den Datenquellen, in denen sie oftmals als Web-Links enthalten sind. Andererseits finden auch vorberechnete Mapping-Kompositionen Verwendung, die die Objekte zwischen zwei Quellen unter Nutzung von temporären Datenquellen abbilden. Alternative Mappings zwischen zwei Quellen führen zu multiplen Mapping-Pfaden. Damit wird eine flexible und dennoch effiziente Berechnung der Join-Operationen durch a) Benutzer-wählbare Mapping-Pfade und b) Vorberechnung ausgewählter Mapping-Kompositionen der Quellen zu einer zentralen Quelle erreicht, wodurch Wege zwischen zwei Quellen mit einer maximalen Länge von 2 garantiert werden.

Dieser Integrationsansatz erweitert das bestehende *GeWare*-System, in dem es exemplarisch Daten der öffentlichen Datenquellen GeneOntology, LocusLink und Ensembl in Analysen verfügbar macht. Darüber hinaus wurden verschiedene Performanzmessungen durchgeführt, die die Leistungsfähigkeit des Ansatzes aufzeigen.

[4] Bis zum Jahr 2006 war die Software SRS im Produktportfolio der Fa. LION bioscience.

Mapping-basierte Datenintegration mit *BioFuice*

Die *BioFuice*-Plattform basiert auf dem *iFuice*-Ansatz[5] [RTA+05] und ermöglicht die Integration von Daten aus privaten und öffentlichen Datenquellen sowie Ontologien im Bereich der Bioinformatik. Dazu dienen Mappings, die Mengen von Korrespondenzen zwischen molekularbiologischen Objekten repräsentieren. Mappings können sowohl innerhalb einer als auch zwischen mehreren Quellen bestehen. Gegenüber dem *iFuice*-Ansatz weist *BioFuice* die folgenden Erweiterungen auf.

- Die Mappings können einerseits statisch vorliegen, d.h. in Form von in den Datenquellen enthaltenen Instanz-Korrespondenzen. Andererseits können Mappings dynamisch auf Basis von Berechnungen erzeugt werden; beispielsweise kann ein Sequenz-Alignment von ausgewählten DNA-Sequenzen in Bezug auf eine selektierte Genom-Sequenz mit der Software BLAST [AGM+90] durchgeführt werden, dessen Ergebnis ein Mapping darstellt.

- *BioFuice* bietet ein mächtiges GUI, in der Anfragen verschiedenartig formuliert und ausgeführt werden können. Dazu zählen neben der freien Skritprogrammierung unter Nutzung der von *iFuice* verwendeten Operatoren auch die Abarbeitung von parametrisierten Skripten. Darüber hinaus können Anfragen in Form einer Stichwortsuche sowie auf Basis der von *iFuice* verwendeten Metadaten-Modelle formuliert werden. Diese beiden Möglichkeiten erfordern eine automatische Transformation der in dem GUI formulierten Anfragen in ausführbare Skripte.

- Die kommandozeilenorientierte Ausführung von *BioFuice* erlaubt eine Einbindung in Analyse-Workflows, für die *BioFuice* die notwendigen Datenintegrationsaufgaben übernimmt. Dazu ermöglicht *BioFuice* einen Datenexport in für die Bioinformatik spezifische Datenformate, wie z.B. das FASTA-Format zum Austausch annotierter genetischer Sequenzen. Ferner wurde eine Schnittstelle zur statistischen Software R entwickelt, das *RiFuice*-Paket, mit der auf Basis von *BioFuice* integrierte Daten einer statistischen Analyse zugeführt werden können.

1.4 Gliederung der Arbeit

Die Arbeit gliedert sich in vier Teile. Im bisherigen Teil 1 wurde die Notwendigkeit der Datenintegration in der Bioinformatik motiviert, offene Probleme

[5] Der Integrationsansatz *iFuice* wurde in Zusammenarbeit mit den Herren E. Rahm, A. Thor, A. Aumüller und N. Golovin entwickelt.

und Anforderungen in dieser Domäne dargestellt und die Beiträge der Arbeit in Hinsicht auf die aufgezeigten Probleme (Kapitel 1) dargelegt. Im Weiteren zeigt Teil 1 generelle Datenintegrationsformen (Kapitel 2), die sich auch in anderen Domänen herausgebildet haben und dort zur Anwendung kommen.

Teil 2 widmet sich der Data-Warehouse-basierten Datenintegration. Aufbauend auf biologischen Grundlagen der Genexpressionsanalyse (Kapitel 3) und den Ergebnissen einer durchgeführten Evaluierung bestehender Datenintegrationsplattformen (Kapitel 4) wird das *GeWare*-System (Kapitel 5) vorgestellt. Neben der Systemarchitektur und dem verwendeten multidimensionalen Datenmodell wird eine Möglichkeit aufgezeigt, experimentelle Metadaten und klinische Daten flexibel, umfangreich und konsistent zu integrieren, die für die Dokumentation und die Reproduzierbarkeit des Experiments unabdingbar sind. Im Anschluss werden zwei Anwendungen des *GeWare*-Systems vorgestellt, die in der sequenzbasierten Untersuchung von Oligo-Intensitäten (Kapitel 6) und der Analyse molekularbiologischer Daten im Anwendungsbereich klinischer Studien (Kapitel 7) bestehen.

Im Mittelpunkt von Teil 3 stehen zwei Mapping-basierte Ansätze zur Datenintegration. Beginnend wird ein hybrider Integrationsansatz (Kapitel 8) beschrieben, mit dessen Hilfe Annotationsdaten, die in öffentlichen Datenquellen verfügbar sind, einer Analyse im *GeWare*-System zugänglich gemacht werden. Im Weiteren wird auf *BioFuice* (Kapitel 10) fokussiert, das auf dem *iFuice*-Ansatz aufbaut. Notwendige Grundlagen des *iFuice*-Ansatzes (Kapitel 9) werden ebenso vorgestellt wie ausgewählte auf *BioFuice* basierende Anwendungen. Abschließend werden ausgewählte verwandte Integrationsansätze und -systeme (Kapitel 11) diskutiert.

Teil 4 beschließt diese Arbeit. Es stellt die wichtigsten Beiträge in zusammenfassender Form dar (Kapitel 12) und gibt einen Ausblick für mögliche zukünftige Arbeiten.

Teile dieser Arbeit wurden als begutachtete Beiträge auf Konferenzen, Workshops und in Journals publiziert und präsentiert. Im Bereich der Data-Warehouse-basierten Datenintegration betrifft dies die Evaluierung Microarray-basierter Datenbanken zur Genexpressionsanalyse in [DKR03] und die Präsentation des *GeWare*-Ansatzes in [KDR03, RKL07]. Zwei technische Berichte [KDR04a, KDR04b] beschreiben den *GeWare*-Ansatz und die darauf aufbauende Datenintegrations- und Analyseplattform ausführlicher. Die Anwendung dieser Plattform in zwei deutschlandweiten klinischen Studien und die Kopplung mit einem System zur Verwaltung und Koordinierung klinischer Studien wird in [KLR06, LKR06] gezeigt. Daneben wurde die Plattform zur sequenzbasierten Analyse von Oligo-Intensitäten verwendet, deren Ergebnisse u.a. in [BKLS04, BKH$^+$04, BPK05] vorgestellt werden. In

[KKDR05, KDKR05] wird der hybride Integrationsansatz als Mapping-basierte Datenintegration beschrieben. Der *iFuice*-Ansatz wird in [RTA+05] präsentiert. *BioFuice* wird in [KR06] beschrieben, dessen Anwendung u.a. zu Ergebnissen bei der Analyse von miRNA [LKS07] geführt hat.

Kapitel 2
Einordnung und generelle Formen der Datenintegration

2.1 Überblick

Eine Datenintegration strebt die Vereinigung von Daten aus unterschiedlichen Quellen an, die räumlich verteilt und hinsichtlich Syntax, Struktur und Semantik heterogen sein können. Mit ihr wird es einem Benutzer in einheitlicher Art und Weise möglich, Anfragen zu formulieren, zu deren Beantwortung die relevanten Daten aus den integrierten Quellen beitragen. Dazu verfolgen Datenintegrationsansätze vielfach den Aufbau einer einheitlichen Sicht – in [Hae02] wird sie als "homogenisierte Sicht" bezeichnet – auf die integrierten Datenbestände. Sie resultiert aus einer Integration der Schemata oder relevanter Schemafragmente der zu integrierenden Datenquellen, mit denen die Daten (so genannte Instanzdaten) in den Quellen beschrieben werden. Im Ergebnis dieser Schemaintegration kann die "homogenisierte Sicht" als applikationsspezifisches globales Schema, generisch oder als globale Ontologie repräsentiert werden. Die Elemente eines applikationsspezifischen globalen Schemas sind auf Realwelt-Objekte ausgerichtet, d.h. die Semantik von Tabellen und Attributen ist festgelegt, was durch eine entsprechende Benennung dieser Schemaelemente repräsentiert wird. Damit bleibt das Einsatzgebiet des applikationsspezifischen globalen Schemas auf eine Domäne und/oder Appli-

kation begrenzt. Dagegen ist eine generische Repräsentation der homogenisierten Sicht von einer Domäne und einer speziellen Applikation unabhängig und kann somit verschiedenartig eingesetzt werden.

Ontologien werden nach [Gru93] als "... explcit specification of a conceptualization ..." betrachtet und ermöglichen anhand von definierten Begriffen, den so genannten Konzepten, eine semantische Beschreibung der Daten in den zu integrierten Quellen. Sie können einerseits im Prozess der Schemaintegration, z.B. zum Aufbau eines applikationsspezifischen globalen Schemas [DL04], eingesetzt werden. Andererseits kann eine Ontologie ebenso die Rolle der "homogenisierten Sicht" übernehmen, an die der Benutzer Anfragen richtet. Im letzteren Fall ist die globale Ontologie ähnlich wie ein applikationsspezifisches globales Schema zumeist auf ein Einsatzgebiet begrenzt; jedoch agiert sie auf einem abstrakteren Niveau als ein Schema[6].

Neben den genannten Formen, die den Aufbau einer "homogenisierten Sicht" a priori verfolgen, existieren Datenintegrationsansätze, die auf eine solche Sicht verzichten. Sie vermeiden nicht nur den zur Erstellung der homogenisierten Sicht notwendigen Aufwand, sondern versuchen auch der Dynamik von auftretenden neuen Fragestellungen und der damit möglicherweise verbundenen Integration von neuen Datenquellen zu begegnen. Ansätze, die eine solche Integration verfolgen, nutzen vielfach Korrespondenzen zwischen den Objekten der zu integrierenden Datenquellen. Im Bereich der Bioinformatik sind diese Objektkorrespondenzen oftmals Bestandteil der Datenquellen. Sie ermöglichen eine Navigation zwischen den in Beziehung stehenden Objekten, die sowohl innerhalb einer Datenquelle als auch aus verschiedenen Datenquellen stammen können. In ihrer einfachsten Form werden die Objektkorrespondenzen als Web-Links repräsentiert. Andere, anspruchsvollere Ansätze nutzen die Objektkorrespondenzen, um Mengen von Objekten auf die anderer Datenquellen abzubilden.

Neben der Klassifikation von Datenintegrationsansätzen, die eine Schemaintegration unter Nutzung einer "homogenisierten Sicht" verfolgen und solchen, die auf eine solche Sicht a priori verzichten, kann eine Einteilung nach der Art der Instanzdatenintegration vorgenommen werden. Hiernach lassen sich die Ansätze in drei Kategorien einteilen, je nachdem, ob sie eine virtuelle, physische oder hybride Integration der Instanzdaten verfolgen. Während die Daten bei einer virtuellen Instanzdatenintegration in den Quellen verbleiben, resultiert aus einer physischen Integration eine neue Datenquelle, in der die relevanten Daten aus den integrierten Quellen materialisiert werden. Ein hybrider Ansatz koppelt Merkmale der virtuellen mit denen einer physischen

[6] Als Schema seien hier die explizit vorhandenen Metadaten (z.B. Bezeichnungen, Strukturen) bezeichnet, die zu den Daten einer Datenquelle existieren (vgl. [GMUW02, Hae02]).

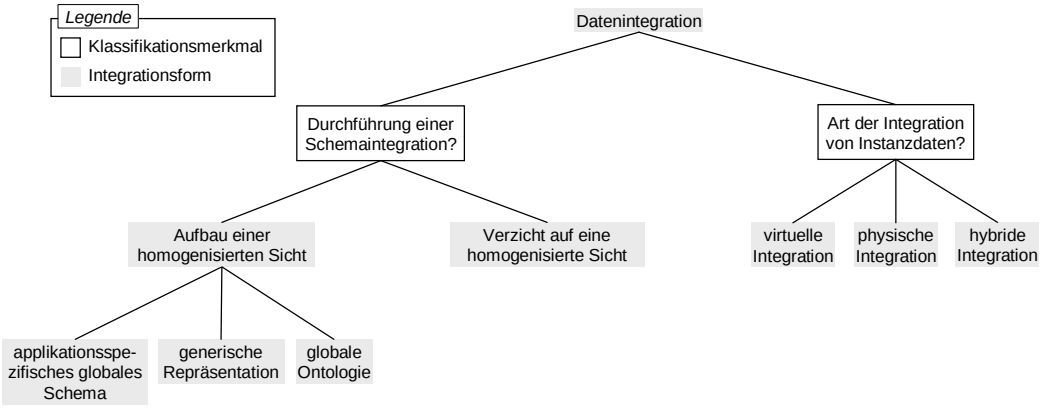


Abbildung 2.1: Klassifikation von Integrationsformen

Instanzdatenintegration.

Die Abbildung 2.1 fasst die genannten Integrationsformen zusammen. Beide Merkmale, die Durchführung einer Schemaintegration und die Art der Instanzdatenintegration, sind orthogonal zu betrachten. Beispielsweise können Integrationsansätze ein applikationsspezifisches globales Schema verwenden und gleichzeitig eine virtuelle Integration verfolgen. Jedoch soll weder unterstellt werden, dass jede Kombinationsmöglichkeit erstrebenswert ist, noch dass zu jeder Kombination bisher Integrationsansätze verfügbar sind. Die Verwendung anderer Merkmale kann eine modifizierte oder erweiterte Klassifikation nach sich ziehen, wie sie beispielsweise in [DD99, JMP02, RAD+06, LN07] angeführt und diskutiert werden.

2.2 Datenintegration mit und ohne Verwendung einer homogenisierten Sicht

Im Folgenden werden die im vorherigen Abschnitt benannten Datenintegrationsansätze – sowohl solche, die den Aufbau einer homogenisierte Sicht a priori verlangen, als auch solche, die darauf zumindest a priori verzichten – diskutiert.

2.2.1 Datenintegration mit einem applikationsspezifischen globalen Schema

Die Abbildung 2.2a zeigt die Verwendung eines globalen Schemas, das an domänen- und anwendungsspezifischen Gegebenheiten bzw. Rahmenbedingungen ausgerichtet ist. Dazu beinhaltet das globale Schema Schemaelemen-

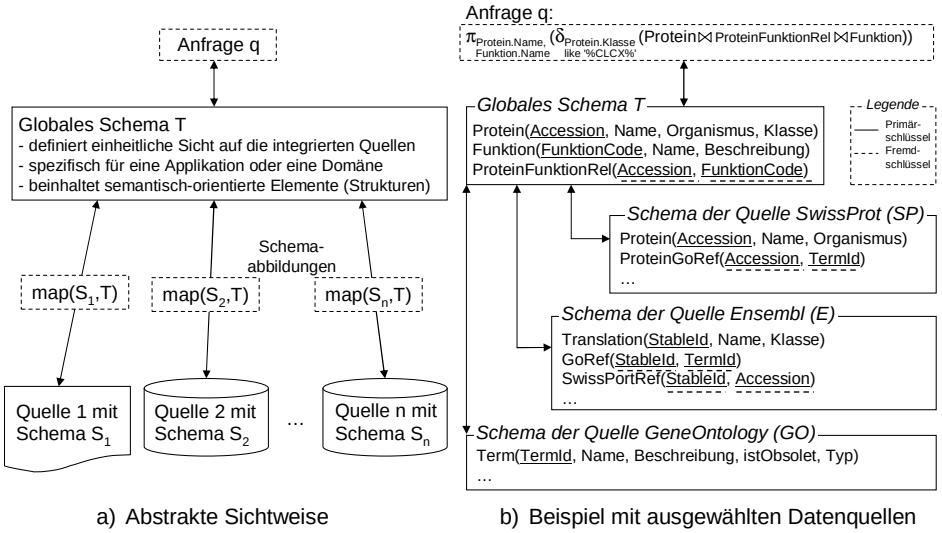


Abbildung 2.2: Verwendung eines applikationsspezifischen globalen Schemas

te (z.B. Relationen im relationalen Modell), die die abgegrenzte Miniwelt der Domäne oder der Applikation semantisch beschreiben. Im Prozess der Schemaintegration entstehen Metadaten, die die Beziehung der Schemaelemente der zugrunde liegenden Datenquellen zu denen des globalen Schemas wiedergeben. Diese Beziehungen münden in schemaspezifischen Abbildungen (engl. schema mappings) $map(S_i, T)$ zwischen den Schemata der Quellen $S_1, \ldots, S_n$ mit $1 \leq i \leq n$ und dem globalen Schema T. Neben einem applikationsspezifischen globalen Schema, das im Zuge der Schemaintegration neu erstellt wird, kann alternativ auch ein Quellenschema S_j $(1 \leq j \leq n)$ als globales Schema verwendet werden. In diesem Fall werden Abbildungen $map(S_i, S_j)$ zwischen den Schemata der Quellen $\{S|S \in \{S_1, \ldots, S_n\} \setminus S_j\}$ und dem als globalen Schema verwendeten Quellenschema S_j erzeugt. Diese Metadaten sind Grundlage für die Integration der Instanzdaten. Damit setzt eine Schemaintegration voraus, dass jede der zu integrierenden Datenquellen ein Schema aufweist, d.h. die Daten dieser Quellen liegen in einer für ein Datenverwaltungssystem oder einer sonstigen Anwendung bekannten Struktur vor [Hae02].

Die Abbildung 2.2b zeigt exemplarisch das Ergebnis der Schemaintegration von Fragmenten dreier ausgewählter Datenquellen aus dem Bereich der Bioinformatik, nämlich GeneOntology (GO), SwissProt (SP) und Ensembl (E), in ein selbst gewähltes applikationsspezifisches globales Schema T. Zur vereinfachten Darstellung werden ausschließlich relationale Schemata verwendet. Die Datenquelle GO beinhaltet Ontologien zur Beschreibung von Eigenschaften molekularbiologischer Objekte (z.B. Protein). Die Quellen SwissProt

und Ensembl enthalten sowohl Daten über Proteine als auch Zuordnungen zu molekularen Funktionen, die in einer Ontologie in GO zusammengefasst sind. Die Schemata der drei Datenquellen beinhalten verschiedene Relationen[7], wobei für eine Integration die Relationen PROTEIN, PROTEINGOREF (beide SP), TRANSLATION, TRANSLATIONGOREF (beide E) sowie TERM (GO) relevant seien. Darüber hinaus beinhaltet die Datenquelle Ensembl Korrespondenzen (Relation SWISSPROTREF) zwischen den beiden Protein-Datenquellen. Damit können äquivalente Proteindaten identifiziert werden, d.h. Daten eines Proteins, die in beiden Quellen ggf. unterschiedlich repräsentiert werden.

Wie in Abbildung 2.2b zu sehen ist, überlappen sich die Schemata der beiden Protein-Datenquellen. Einerseits beinhalten beide Quellen einen Proteinnamen und einen Identifikator, der jedoch quellenspezifisch ist (Ensembl: StableId, SwissProt: Accession). Andererseits werden in ihnen die Beziehungen zwischen den Proteindaten und den Konzepten der GO Ontologien zur funktionalen Beschreibung gespeichert. Damit ist es notwendig, äquivalente Protein-Objekte zwischen beiden Datenquellen zu identifizieren, um Duplikate zu vermeiden und Widersprüche in Hinsicht auf die zugeordneten funktionalen Beschreibungen aufzuspüren. Diese Problematik ist typischerweise Gegenstand der Instanzdatenintegration (vgl. Abschnitt 2.3).

Das globale Schema T in Abbildung 2.2b dient der gemeinsamen Darstellung und Analyse von Proteinen, zu denen Funktionen assoziiert sind. Dazu besteht es aus den Relationen PROTEIN, FUNKTION und PROTEINFUNKTIONREL, die über Fremdschlüsselbeziehungen (unter Annahme von Namensgleichheit der Attribute) miteinander verbunden sind. Es ergeben sich die folgenden Schemaabbildungen zwischen den Datenquellen und dem globalen Schema:

$$
\begin{aligned}
map(GO,T) := \{&(Term.TermId \rightarrow Funktion.FunktionCode),\\
&(Term.Name \rightarrow Funktion.Name),\\
&(Term.Beschreibung \rightarrow Funktion.Beschreibung)\}\\
map(E,T) := \{&(Translation.StableId \rightarrow Protein.Accession),\\
&(Translation.Name \rightarrow Protein.Name),\\
&(Translation.Klasse \rightarrow Protein.Klasse),\\
&(GoRef.StableId \rightarrow ProteinFunktionRel.Accession),\\
&(GoRef.TermId \rightarrow ProteinFunktionRel.FunktionCode)\}
\end{aligned}
$$

[7] Die Relationen wurden zu Darstellungszwecken stark vereinfacht.

$$
\begin{aligned}
map(SP,T) := \{&(Protein.Name \rightarrow Protein.Name),\\
&(Protein.Organismus \rightarrow Protein.Organismus),\\
&(ProteinGoRef.Accession \rightarrow ProteinFunktionRel.Accession),\\
&(ProteinGoRef.TermId \rightarrow ProteinFunktionRel.FunktionCode)\}
\end{aligned}
$$

Anfragen, wie beispielsweise "*Selektiere alle Proteine mit der Klasse 'CXCL' (spezielle Klasse von Chemokine Liganden), und projiziere den Proteinnamen sowie die assoziierten Funktionen*" werden direkt und ausschließlich an das globale Schema gerichtet. In Abbildung 2.2b wird die vorgenannte, exemplarische Anfrage als Ausdruck der relationalen Algebra dargestellt. Die weitere Anfrageverarbeitung richtet sich nach der Art der Integration der Instanzdaten (siehe Abschnitt 2.3).

2.2.2 Generische Repräsentation eines globalen Schemas

Die Abbildung 2.3a zeigt die Nutzung einer generischen Repräsentation. Sie zeichnet sich dadurch aus, dass mit ihr die homogenisierte Sicht unabhängig von der Anwendungsdomäne oder einer konkreten Applikation repräsentiert werden kann. Der *Entity-Attribute-Value*-Ansatz (EAV) [NB98], das *Generic Annotation Model* (GAM) [DR04] und das *uniforme Datenmodell*, z.B. bekannt aus [JMvG95] und [EH98, Hei02], sind Ansätze, mit denen die Elemente einer homogenisierten Sicht unter Verwendung generischer Strukturen in einem relationalen Schema gespeichert werden können. Diese Ansätze bieten den Vorteil, dass die Hinzunahme neuer Quellen sowie Schemaänderungen der integrierten Datenquellen keine Änderungen an den Strukturen des generischen Schemas nach sich ziehen, wie sie beispielsweise bei einem applikationsspezifischen globalen Schema notwendig werden können. Demgegenüber ist eine Erweiterung oder Adaption der Schemaabbildungen zwischen den Quellen und dem globalen Schema jedoch ggf. weiterhin nötig. Darüber hinaus sind Anfragen an ein generisches Schema komplexer gegenüber solchen, die für ein applikationsspezifisches globales Schema formuliert werden. Mit dieser höheren Komplexität der Anfragen ist oftmals eine längere Zeit der Anfrageverarbeitung verbunden. Nachteilig kann ebenso die fehlende Transparenz der Daten sein: Im Unterschied zum applikationsspezifischen globalen Schema sind bei Kenntnis der Schemaelemente und -strukturen keine Rückschlüsse auf die Struktur der Daten in den Quellen möglich. Daher muss ein Benutzer neben den Elementen des generischen Schemas die Struktur der

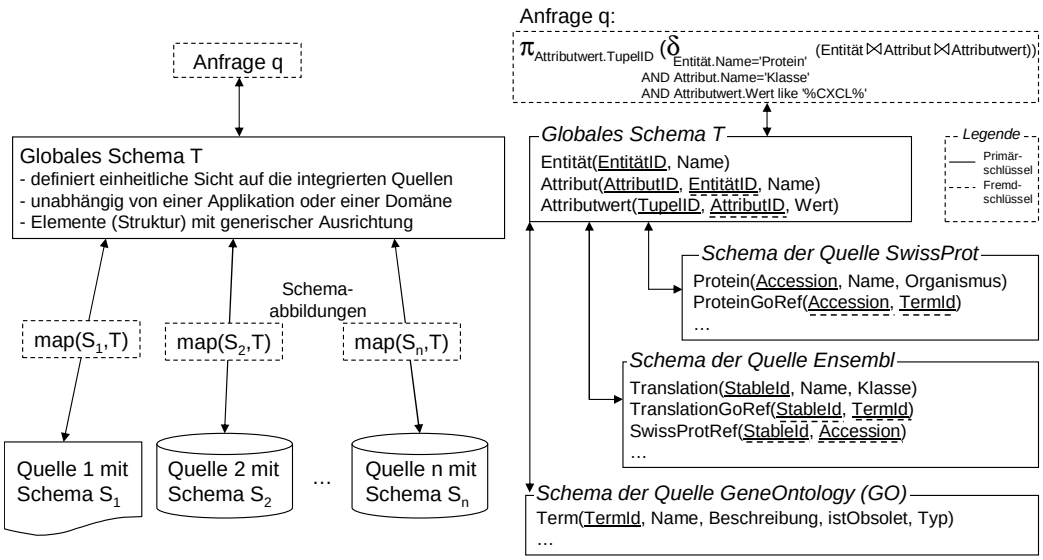


Abbildung 2.3: Nutzung eines generischen globalen Schemas

Daten kennen, um eine Anfrageformulierung und -verarbeitung erfolgreich zu gestalten.

Die Abbildung 2.3b illustriert exemplarisch die Schemaintegration unter Nutzung eines generischen globalen Schemas. Dazu werden die Datenquellen verwendet, für die bereits eine Integration in Abbildung 2.2b gezeigt wurde; für eine vereinfachte Darstellung beschränkt sich die Darstellung auf relationale Schemata. Das (vereinfachte) generische Schema folgt dem EAV-Ansatz bestehend aus den Relationen ENTITÄT, ATTRIBUT und ATTRIBUTWERT. Eine Besonderheit dieses Schemas besteht darin, dass die Relationen ENTITÄT und ATTRIBUT die Elemente der erzeugten homogenisierten Sicht als Instanzdaten enthalten, die den Namen der Relationen und Attribute des applikationsspezifischen globalen Schemas in Abbildung 2.2b entsprechen. Beispielsweise enthält die Relation ENTITÄT die Tupelmenge

$$\{(1, \text{Protein}), (2, \text{ProteinFunktionRel}), (3, \text{Funktion})\}$$

und korrespondierend dazu die Relation ATTRIBUT

$$\{(1, 1, \text{Accession}), (2, 1, \text{Name}), \ldots, (5, 2, \text{FunktionCode}), \ldots\}$$

Damit können neue Elemente der homogenisierten Sicht flexibel hinzugefügt werden; es müssen lediglich in beiden Tabellen ENTITÄT und ATTRIBUT die entsprechenden Tupel eingefügt werden. Ein Änderung des Schemas,

wie im Falle eines applikationsspezifischen globalen Schemas, ist damit nicht notwendig.

Im Gegensatz dazu beinhaltet die Relation ATTRIBUTWERT die im Sinne des applikationsspezifischen globalen Schemas verwendeten Instanzdaten. Dabei bildet jedes Tupel in dieser Relation genau einen atomaren Attributwert ab. Alle Attributwerte, die im applikationsspezifischen globalen Schema zusammenhängend ein Tupel bilden, werden über das Attribut TupelId identifiziert. Beispielsweise führen die beiden Tupel der Relation PROTEIN im applikationsspezischen globalen Schema der Abbildung 2.2b

(ENSP00000226317, Cytokine B6 precursor, Homo Sapiens, CLCX6)
(ENSP00000306512, Interleukin-8 precursor, Homo Spaiens, CLCX8)

zu den folgenden Tupeln der Relation ATTRIBUTWERT in der generischen Repräsentation

(1, 1, ENSP00000226317)
(1, 2, Cytokine B6 precursor)
(1, 3, Homo Sapiens)
. . .
(2, 1, ENSP00000306512)
(2, 2, Interleukin-8 precursor)
(2, 3, Homo Sapiens)
. . .

Damit ist nicht nur ein erhöhter Speicherbedarf verbunden, sondern insbesondere auch eine wesentlich komplexere und kompliziertere Anfrageformulierung und -verarbeitung. Für jeden Zugriff auf einen Attributwert der in Abbildung 2.2b gezeigten Anfrage werden Selektions-, Projektions- und Join-Operationen im generischen globalen Schema notwendig. Die in Abbildung 2.3b gezeigte Anfrage bildet lediglich einen Teil der Anfrage aus Abbildung 2.2b ab, nämlich den, der zur Selektion von Genen der Chemokine CXCL-Klasse notwendig ist. Aus einer solchen komplexeren und umfassenderen Anfrageformulierung resultieren längere Abarbeitungszeiten der Anfrage. Der Einsatz von materialisierten Sichten, die die Daten redundant in Form eines applikationsspezifischen globalen Schemas (oder ausgewählter Fragmente) speichern und jederzeit aus den generischen Strukturen abgeleitet werden können, verspricht Performanzverbesserungen.

Entgegen der komplexeren Anfrageverarbeitung werden die Schemaabbildungen vereinfacht, da alle Instanzdaten der zu integrierenden Datenquellen

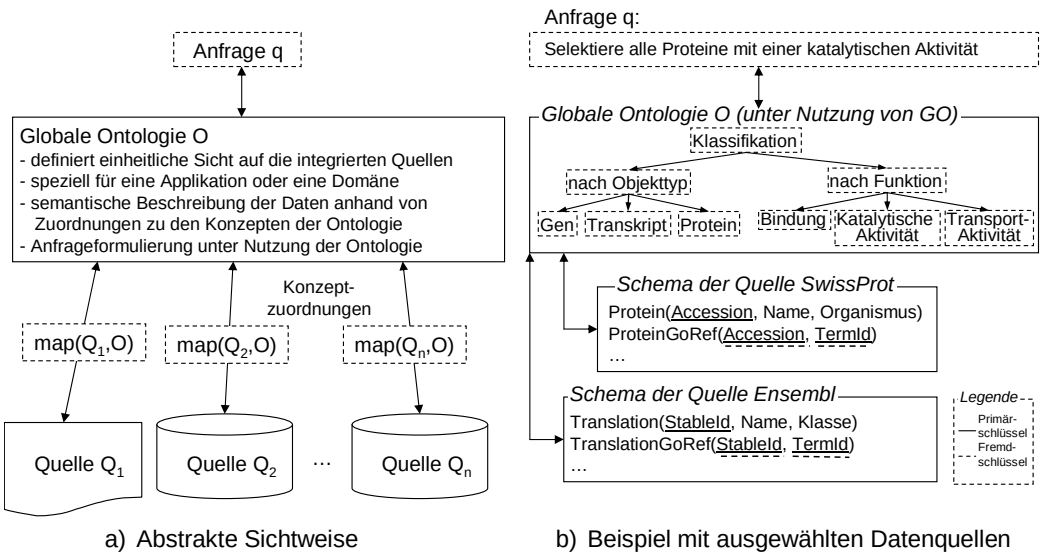


Abbildung 2.4: Datenintegration auf Basis einer globalen Ontologie

ausschließlich in die Relation ATTRIBUTWERT des generischen globalen Schemas Eingang finden.

2.2.3 Datenintegration unter Nutzung einer globalen Ontologie

Mit Ontologien können Realwelt-Objekte konsistent beschrieben werden und machen damit deren Semantik explizit. Um terminologische Variationen so weit wie möglich zu reduzieren, verfügen sie über ein definiertes und kontrolliertes Vokabular von Termen, den so genannten Konzepten. Zwischen den Konzepten bestehen Beziehungen, wie etwa "is a" und "part of", so dass eine hierarchische Organisation in Bäumen resultiert oder komplexe Graphen (z.B. gerichtete azyklische Graphen) entstehen.

Die Ausnutzung einer solchen semantischen Beschreibung von Objekten der einzelnen Datenquellen ermöglicht eine Datenintegration, in der die verwendete Ontologie als homogenisierte Sicht[8] fungiert (vgl. Abbildung 2.4a). Dazu sind alle zu integrierenden Datenquellen mit der Ontologie derart verbunden, dass deren Konzepte die Objekte (z.B. Gene, Proteine) in den Datenquellen beschreiben und strukturieren. Damit wird die Erstellung von Abbildungen (Mappings) von der Schema- auf die Instanzebene verlagert. Sofern auf eine bereits bestehende Ontologie und Assoziationen zwischen Instanzen

[8] Gegenüber einem applikationsspezifischen globalen Schema agiert die globale Ontologie auf einem wesentlich abstrakteren Niveau.

und Konzepten zurückgegriffen werden kann, bleibt der Integrationsaufwand gering; muss stattdessen eine globale Ontologie erst aufgebaut werden, kann der Aufwand sehr hoch sein. Insbesondere die Zuordnung der Instanzen zu den Konzepten kann aufwändiger sein als die Integration mit einem applikationsspezifischen globalen Schema, da die Anzahl der Instanzen vielfach die Anzahl der Schemaelemente übersteigt.

Ähnlich wie ein applikationsspezifisches globales Schema strebt eine globale Ontologie eine semantische Homogenität an. Jedoch begegnet sie im Gegensatz zum applikationsspezifischen globalen Schema nicht der strukturellen Heterogenität, da die Objekte lediglich kategorisiert werden. Eine Abbildung der Quellenschemata findet nicht statt, z.B. welche Schemaelemente zur Beschreibung eines Objekts zwischen zwei Datenquellen ähnlich sind. Damit kann ein Konzept lediglich mit Objekten einer Datenquelle verbunden werden oder der Benutzer nimmt Duplikate sowohl in den Instanzdaten als auch in den beschreibenden Attributen in Kauf.

Die Abbildung 2.4b illustriert exemplarisch die Integration der beiden Datenquellen Ensembl und SwissProt unter Verwendung einer selbst gewählten globalen Ontologie. Die Ontologie dient einerseits der typspezifischen Klassifikation von molekularbiologischen Objekten und andererseits zur funktionalen Beschreibung, wobei hierzu auf die existierende GO-Ontologie "Molekulare Funktionen" zurückgegriffen wird[9]. Dazu sind die in den beiden Datenquellen gespeicherten Objekte (Relation in Ensembl: TRANSLATION, Relation in SwissProt: PROTEIN) mit ausgewählten Konzepten der globalen Ontologie assoziiert, die auf Basis der Relationen TRANSLATIONONTREF (Ensembl) und PROTEINONTREF (SwissPort) gespeichert werden. Anfragen werden unter Nutzung der globalen Ontologie formuliert, für die zumeist proprietäre Sprachen verwendet werden. Daher soll sie im vorliegenden Beispiel mit einer natürlich sprachlichen Formulierung nur angedeutet werden.

Abweichend von der hier gezeigten Darstellung werden in [WVV+01] drei mögliche ontologiebasierte Integrationsformen unterschieden. Sie bestehen in der Nutzung einer einzelnen globalen Ontologie, verbundenen multiplen lokalen Ontologien sowie einem hybriden Ontologie-Ansatz. Während der Einsatz einer einzelnen globalen Ontologie dem in Abbildung 2.4 gezeigten Ansatz entspricht, wird mit den beiden letzteren Ansätzen auf die Erstellung und Nutzung einer homogenisierten Sicht verzichtet.

[9] In der Abbildung 2.4b wird die Ontologie nur angedeutet, da eine umfassende Darstellung an dieser Stelle nicht möglich und sinnvoll ist.

2.2.4 Datenintegration mit Verzicht auf eine homogenisierte Sicht

Obwohl die Schemaintegration in den letzten Jahren ein Feld aktiver Forschungsbemühungen (für einen Überblick siehe z.B. [RB01, Do06]) war, geht die Konstruktion einer homogenisierten Sicht je nach Komplexität der Schemata bzw. der relevanten Schemafragmente der zugrunde liegenden Datenquellen mit einen hohen Aufwand einher. Dies gilt insbesondere für die Bildung und Nutzung eines applikationsspezifischen globalen Schemas sowie einer globalen Ontologie. Die für diese beiden Integrationsformen notwendigen Schemaabbildungen und Konzeptzuordnungen sind nicht vollautomatisiert mit einem korrekten Ergebnis zu erlangen. Um den Integrationsprozess zu flexibilisieren und den notwendigen Aufwand zur Integration zu reduzieren, haben sich Ansätze gebildet, die auf die Bildung einer homogenisierte Sicht a priori verzichten. Solche Ansätze nutzen vielfach die in den Datenquellen vorhandenen Objektkorrespondenzen aus, orientieren sich an Architektur und Funktionsweise von Peer-to-Peer-Systemen oder kombinieren Merkmale aus beiden. Andere Ansätze, die ebenso auf ein globales Schema verzichten, z.B. Multidatenbanken [HM85, LMR90], werden hier nicht weiter betrachtet.

Objektkorrespondenzen verbinden die Objekte (Instanzen) innerhalb einer Datenquelle oder zwischen verschiedenen Datenquellen. Abgegrenzte Mengen dieser Objektkorrespondenzen werden als *Mappings* (engl. Abbildungen) bezeichnet und können unter Nutzung von speziellen Merkmalen gebildet werden, z.B. Objektkorrespondenzen zwischen zwei ausgewählten Datenquellen oder solche, die zusätzlich über dieselbe Semantik (z.B. Orthologie- vs. Paralogiebeziehung zwischen Genen) verfügen. Im Bereich der Bioinformatik sind die Mappings Bestandteil von Datenquellen oder werden mit speziellen Hilfsmitteln (z.B. Programmen) erzeugt. Sie erlauben ausgehend von einer Objektmenge eine Navigation zu in Beziehung stehenden Objekten und gelten als Grundlage einer *Mapping-basierten Datenintegration*.

Im Bereich der Bioinformatik sind die Objektkorrespondenzen vielfach Teil der Datenquellen (vgl. Kapitel 1); nicht vorhandene Korrespondenzen sind oftmals das Ziel von komplexen Analysen oder können auf Basis von Sequenzvergleichen ermittelt werden. Da viele der öffentlich zugänglichen Quellen eine Web-Schnittstelle anbieten, werden die Objektkorrespondenzen zu Zwecken der Anfrageformulierung zu Web-Links transformiert, mit denen zu assoziierten Objekten navigiert werden kann. Jedoch bleibt die Navigation zumeist auf ein Objekt beschränkt, für das dem Web-Link gefolgt werden kann. Damit gestalten sich Anfragen und Analysen sehr aufwändig, in die viele Objekte einbezogen werden sollen. Andere Ansätze, z.B. das *Sequence Retrival System* (SRS) [EHB03] und das *Distributed Annotation System* (DAS)

[DJD+01, PBC+06] verwenden ebenso Web-Links, setzen aber verschiedene Techniken ein, mit denen auch mengenbasierte Anfragen sowie weitergehende Operationen möglich sind[10].

Peer Data Management Systeme (PDMS) [BGK+02, HIMT03, HHNR05, LN07] orientieren sich an Peer-to-Peer-Systemen, die nach [SvHSS06] alle Systeme charakterisiert, die eine dezentralisierte Architektur verwenden und einzelnen Teilnehmern (Peers) erlaubt, ohne zentrale Steuerung und Kontrolle sowohl Ressourcen zur Verfügung zu stellen als auch an ihnen zu partizipieren[11]. Demgemäß können die Datenquellen, die im PDMS als Peers agieren, zwei Aufgaben übernehmen [LN07]. Zum einen agieren sie als Datenquellen, die Daten speichern und zur Verfügung stellen. Zum anderen übernehmen sie die Rolle eines Mediators, der Anfragen entgegen nimmt und auf Basis eigener Daten beantwortet oder die Anfrage (oder Teile davon) an andere Peers weiterreicht. Dazu sind die Peers miteinander verbunden; die Art der Verbindung kann unterschiedlich gestaltet sein und von der expliziten Ermittlung, Speicherung und Nutzung von Schema-Mappings zwischen den Peers bis hin zur Verwendung von bestehenden instanzbasierten Mappings (als Ausdruck für abgegrenzte Mengen von Objektkorrespondenzen) reichen.

Die Abbildung 2.5a illustriert ein PDMS, das aus einer Menge von mit Mappings verbundenen Peers besteht. Jeder Peer repräsentiert der P2P-Idee folgend eine Menge von Datenquellen - im Spezialfall ist jedem Peer eine Datenquelle zugeordnet. Im Unterschied zu den Ansätzen, die eine homogenisierte Sicht verlangen, können Anfragen an jeden Peer gestellt werden, der sie auf Basis eigener Daten beantwortet oder an andere Peers weiterleitet. Die Heterogenität der Datenquellen und Peers wird mit Hilfe der Mappings überwunden. Sie können einerseits dazu dienen, auf Basis von angegebenen Schemaabbildungen die weiterzuleitenden Anfragen umzuformulieren, so dass dem die Anfrage empfangende Peer eine Anfrageverarbeitung und Beantwortung möglich ist. Andererseits ermöglichen Mappings auf Basis von Objektkorrespondenzen mengenbasierte Operationen, z.B. selektiere alle Proteine der Quelle SwissProt, die zu denen einer zuvor selektierten Menge von Proteinen der Datenquelle Ensembl korrespondieren. Werden die Peers als Knoten und die Mappings als Kanten aufgefasst, ergibt sich ein Netzwerk, dessen Struktur dynamisch ist, d.h. es kann einer Änderung unterworfen sein. Neue Knoten können dem Netzwerk hinzugefügt werden, während es andere verlassen. Jedoch ist die Dynamik typischerweise weit weniger stark ausgeprägt als bei reinen P2P-Systemen.

[10] In Kapitel 11 wird SRS vorgestellt und zu den eigenen entwickelten Mapping-basierten Ansätzen abgegrenzt.

[11] In [LN07] wird eine gute Darstellung von PDMS sowie Abgrenzungen zu anderen Integrationsansätzen gegeben.

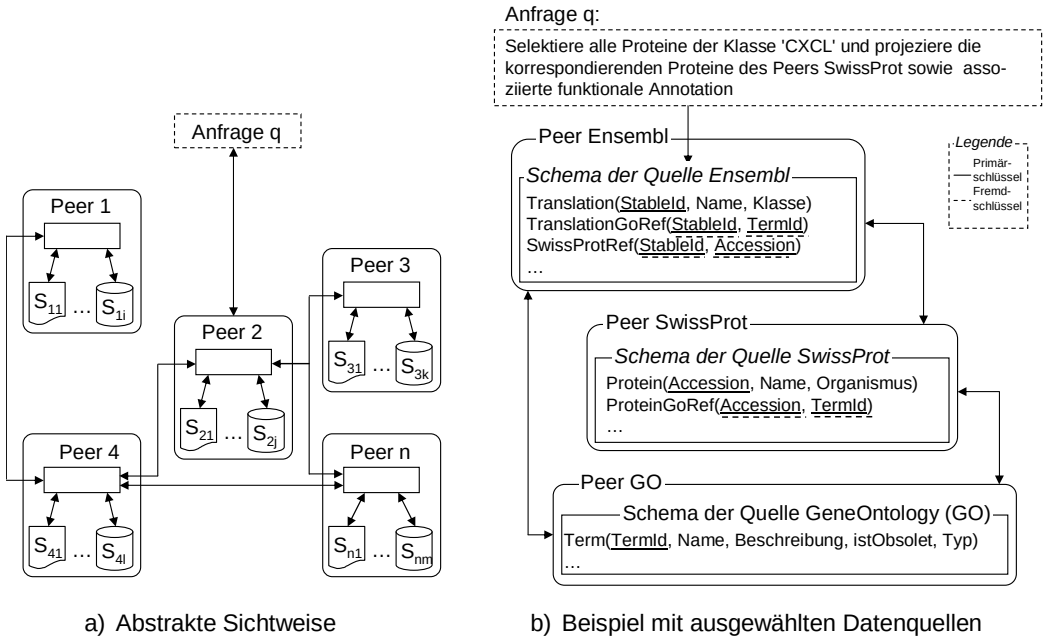


Abbildung 2.5: Schemaintegration mit Verzicht auf ein globales Schema am Beispiel von PDMS

Die Abbildung 2.5b zeigt ein vereinfachtes Beispiel für die Datenintegration auf Basis eines PDMS. Jede der drei Datenquellen, Ensembl, SwissProt und GO, repräsentiert einen Peer im PDMS. Mengen von Objektkorrespondenzen zwischen den Datenquellen bilden Mappings und verbinden damit die Datenquellen. Beispielsweise enthält die Relation SwissProtRef der Quelle Ensembl das Mapping zwischen den Translationen der Quelle Ensembl und den Proteinen der Quelle SwissProt. Schemaabbildungen bestehen implizit zwischen den Objekt-Identifikatoren der Datenquellen (z.B. StableId für Ensembl); komplexe Schemaabbildungen werden zur vereinfachten Darstellung vernachlässigt. Eine Anfrage, die im gezeigten Beispiel in natürlicher Sprache angegeben ist, soll alle Proteine der Klasse 'CXCL' zurück liefern, die mit den Identifikatoren der Datenquelle SwissProt sowie funktionaler Annotation der Quelle GO assoziiert sind. Eine vereinfachte Anfrageverarbeitung könnte wie folgt aussehen. Zuerst werden in der Quelle Ensembl alle Proteine der Klasse 'CXCL' identifiziert. Auf Basis der Objektkorrespondenzen in der Relation SwissProtRef (Quelle: Ensembl) werden die korrespondierenden Proteindaten in der Quelle SwissProt gefunden, zu denen unter Nutzung der Objektkorrespondenzen in der Relation ProteinGoRef (Quellen: SwissProt) die funktionale Annotation assoziiert werden kann. Neue Peers und damit neue Datenquellen können jederzeit in das PDMS eingebunden wer-

den. Hierzu wird ein Mapping benötigt, das die neue Datenquelle mit einer bereits im PDMS integrierten Datenquelle verbindet.

2.2.5 Gegenüberstellung von Integrationsformen mit und ohne homogenisierter Sicht

Die Tabelle 2.1 stellt die oben angeführten Integrationsformen, die den Aufbau einer homogenisierten Sicht a priori verlangen oder auf eine solche a priori verzichten, unter Nutzung ausgewählter Kriterien gegenüber. Die a priori erstellte homogenisierte Sicht kann in einem applikationsspezifischen globalen Schema, einer generischen Repräsentation und einer globalen Ontologie bestehen. Während das applikationsspezifische globale Schema eine sehr gute Analyseunterstützung bietet, ist dies bei einer generischen Repräsentation auf Grund der komplexeren Anfrageformulierung zumeist nicht ohne den Einsatz weiterer Techniken gewährleistet, die beispielsweise in der Nutzung materialisierter Sichten bestehen kann. Jedoch ist der Aufwand, ein applikationsspezifisches globales Schema zu etablieren, und die Abhängigkeit von den Quellenschemata gegenüber der Nutzung einer generischen Repräsentation sehr hoch, da es nicht nur der Konstruktion des globalen Schemas bedarf, sondern auch der Schemaabbildungen zwischen den Quellen und dem globalen Schema. Letztere sind im Vergleich zu einem applikationsspezifischen globalen Schema mit geringem Aufwand zu erstellen oder anzupassen.

Obwohl die globale Ontologie eine homogenisierte Sicht auf die Daten bereitstellt, findet eine Schemaintegration, wie sie für die Nutzung eines applikationsspezifischen globalen Schemas und einer generischen Repräsentation notwendig werden, nicht statt. Vielmehr werden Assoziationen benötigt, die die Instanzen den Konzepten der globalen Ontologie zuordnen. Auftretende Änderungen der Quellenschemata haben nur einen geringen Einfluss auf die globale Ontologie und den Mappings zu ihr; lediglich die Anfragefunktionalität auf die Quellen (z.B. in Form von Wrappern) muss wiederhergestellt werden. Jedoch kann der Integrationsaufwand den für die Erstellung eines applikationsspezifischen globalen Schemas übersteigen, sofern nicht auf eine existierende Ontologie und vorhandene Korrespondenzen zwischen Instanzdaten und den Konzepten der verwendeten Ontologie zurückgegriffen wird. Gerade bei einer komplexen Domäne mit vielen unterschiedlichen Objekttypen (z.B. Gene, Proteine) sowie den umfangreichen Datenbeständen, wie sie in der Bioinformatik vorzufinden sind (vgl. Kapitel 1), kann die Erstellung einer neuen und umfassenden Ontologie einen hohen Aufwand erfordern.

Mapping-basierte Ansätze und PDMS verzichten a priori auf die Erstel-

[12] Dazu liegen bisher wenig Erkenntnisse vor (siehe Seite 30).

Tabelle 2.1: Gegenüberstellung von Integrationsformen mit und ohne Schemaintegration

	Verwendung einer homogenisierten Sicht			**keine homogenisierte Sicht**
	applikationsspezifisches globales Schema	**generische Repräsentation**	**globale Ontologie**	**Mapping-basierte Ansätze & PDMS**
Zeitpunkt der Schemaintegration	a priori		Schemaintegration bleibt zumeist aus	nicht a priori
Aufwand zur Integration	hoch: Aufbau des globalen Schemas & Def. der Schemaabbildungen	niedrig: wiederverwendbares generisches Schema & Aufbau der Schemaabbildungen	Aufwand abhängig von Existenz einer Ontologie und Assoziationen zwischen Objekten und Konzepten der globalen Ontologie	moderat: Bestimmung von Mappings (Objektkorrespondenzen & Schemaabbildungen
Abhängigkeit von Schemaänderungen in den Quellen	hoch: Adaption des globalen Schema & der Schemaabbildungen	mittel: Adaption der Schemaabbildungen	niedrig: Anpassung der Anfragefunktionalität bei Durchgriff auf Quellen	moderat: evtl. Anpassung von wiederverwendbaren Schemaabbildungen
Analyseunterstützung	sehr gut	schlechte Performanz ohne Einsatz weiterer Techniken	abhängig von der Mächtigkeit der eingesetzten Anfragesprache	moderate bis potentiell gute Unterstützung[12]
Skalierbarkeit (Anzahl von Quellen)	gering: sehr hoher Aufwand zur Integration neuer Quellen	mittel: ein globales Schema für viele Quellen	potentiell gering[12]	potentiell gut[12]

lung einer homogenisierten Sicht. Solche Integrationsansätze nutzen vielfach die in Form von Objektkorrespondenzen gegebenen Mappings zwischen den Instanzdaten zweier Quellen für eine Integration. Oftmals finden diese Mappings in explorativen Analysen Anwendung, in denen anhand der gegebenen Daten Hypothesen generiert oder überprüft werden können[13]. Mapping-basierte Ansätze benötigen einen niedrigen bis moderaten Aufwand, der aus der Beschreibung der Mappings (bezogen auf die verfügbaren Objektkorrespondenzen) sowie der evtl. expliziten Angabe von Schemaabbildungen resultiert. Im Gegensatz zu Ansätzen mit einem applikationsspezifischen globalen Schema bieten sie potentiell eine moderate Analyseunterstützung. Bei PDMS wird sie durch die komplexe Anfrageverarbeitung unter Einbeziehung mehrerer Peers beeinflusst, da Anfragen oftmals umformuliert und Daten transformiert werden müssen. Das kann letztlich einen Informationsverlust zur Folge haben (vgl. [LN07]).

2.3 Arten der Instanzdatenintegration

In Hinsicht auf die Integration von Instanzdaten wird in die virtuelle, physische und hybride Integration unterschieden. Im Folgenden werden die ersten beiden Arten vorgestellt. Eine hybride Instanzdatenintegration kombiniert Merkmale einer virtuellen und physischen (materialisierten) Integration.

Virtuelle Datenintegration Die Abbildung 2.6a illustriert eine virtuelle Integration. Die Daten verbleiben in den originären Quellen, auf die auf Anforderung, d.h. zur Laufzeit der Anfrageverarbeitung, zugegriffen wird. Das setzt voraus, dass die Datenquellen zur Laufzeit verfügbar sind und den relevanten Teil der Anfrage verarbeiten können.

Ansätze, die eine virtuelle Integration verfolgen, bestehen oftmals in förderierten Datenbanken (FDBS) [SL90, TTC+90, BHP92, Rah94] oder Mediatoren [Wie92] und verwenden ein globales Schema, das ebenso wie die Abbildungen zwischen den Schemata der zu integrierenden Datenquellen und dem globalen Schema in einem Metadaten-Repository gespeichert ist. Anfragen werden an das globale Schema gestellt. Eine solche Anfrage wird im Zuge der Anfrageverarbeitung durch das FDBS / den Mediator auf Basis der vorhandenen Schemaabbildungen in einzelne Anfragen an die relevanten originären Datenquellen separiert. Diese quellenspezifischen Anfragen werden anschließend an die jeweiligen Datenquellen gesendet und dort verarbei-

[13] Die Validierung von Hypothesen beschränkt sich auf die Auswertung des vorhandenen Datenmaterials. Für eine tiefgreifendere Überprüfung sind oftmals weitergehende Laborexperimente notwendig.

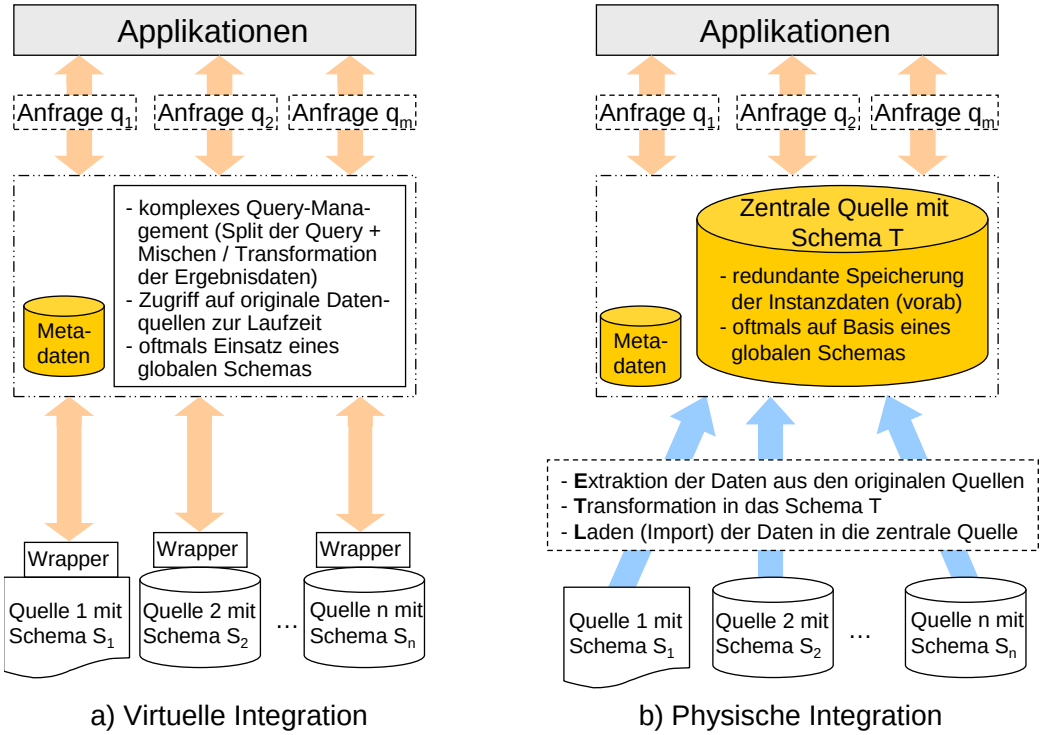


Abbildung 2.6: Arten der Integration von Instanzdaten

tet. Dabei wird der Zugriff auf die Daten und die Abarbeitung der Anfrage vielfach durch so genannte "*Wrapper*" sichergestellt. Die Ergebnisdaten, die sich aus den einzelnen Anfragen an die originären Datenquellen ergeben, werden im anschließenden Schritt vom FDBS / Mediator gemischt, so dass die relevante Ergebnismenge in Bezug auf die Anfrage an das globale Schema entsteht. Soweit Aspekte, die die Datenqualität betreffen, nicht bereits bei der Anfragetransformation (Formulierung von Anfragen an die zugrunde liegenden Datenquellen) beachtet wurden, z.B. bei inhaltlichen Überlappungen der Quellen, ist ein weiterer Schritt notwendig, um Dateninkonsistenzen zu beheben.

Physische Datenintegration Die Abbildung 2.6b illustriert eine physische Integration. Im Gegensatz zur virtuellen Integration wird auf die relevanten Daten nicht in den originären Quellen zugegriffen. Vielmehr werden sie in einer neuen Datenquelle zusammengefasst und somit redundant gespeichert. Diese neue Datenquelle dient der Verarbeitung von Anfragen. Somit muss zur Laufzeit der Anfrageverarbeitung nicht auf die originären Quellen zugegriffen werden, was einerseits zu einer Performanzsteigerung führen kann

Tabelle 2.2: Vergleich von Arten zur Integration von Instanzdaten

	virtuelle Integration	**physische Integration**
Zeitpunkt der Integration	zur Laufzeit	a priori
Abhängigkeit der Anfrageausführung von Datenquellen	abhängig von Zugreifbarkeit und Leistungsfähigkeit der Quellen während der Anfrage-ausführung	zum Zeitpunkt des ETL-Prozesses
Aktualität der Daten	stets aktuell	abhängig vom Ladezyklus
Hardware-anforderungen	leistungsfähiger Server	hochleistungsfähiger Server mit viel Speicher
Analyse-performanz	weitestgehend abhängig von Leistungsfähigkeit der Quellen (bzw. deren Wrapper)	abhängig von zentraler Datenhaltung (z.B. Datenbank)
Eignung für große Datenmengen in Anfragen	weniger geeignet; größeres Datenvolumen verlangt höhere Bandbreite oder mehr Zeit zur Übertragung	gut geeignet
Datenqualität	Konfliktlösung zur Laufzeit; evtl. nur eingeschränkt möglich	Konfliktlösung a priori im ETL-Prozess

und andererseits eine Unabhängigkeit von den originären Quellen gewährleistet. Letzteres wirkt sich insbesondere dann positiv aus, wenn die originären Quellen, z.B. wegen Wartungsarbeiten, nicht verfügbar sind oder Anfragen auf Grund von Überlastung, die typischerweise zu so genannten *timeouts* führen, nicht verarbeiten kann.

Ein Ansatz, der eine physische (a priori) Integration der Daten verfolgt, ist das Data Warehouse [Inm92, JLVV03]. Es verwendet ein globales Schema, das zusammen mit den Schemaabbildungen in einem Metadaten-Repository gespeichert ist. Dabei dienen die Schemaabbildungen vor allem für die Vorabintegration der Daten aus den Quellen in dem so genannten ETL-Prozess (ETL steht für **e**xtraction, **t**ransformation, **l**oading) [JLVV03]. In diesem ETL-Prozess werden die relevanten Daten aus den Quellen extrahiert (Schritt 1: extraction), auf Basis der vorhandenen Schemaabbildungen in das globale Schema transformiert (Schritt 2: transformation) und anschließend in das zentrale Data Warehouse geladen (Schritt 3: loading). Bestehende Datenkonflikte, Inkonsistenzen und sonstige Aspekte, die die Datenqualität beeinflussen, werden im Schritt der Datentransformation behandelt [RD00]. Im Gegensatz zur virtuellen Integration mit FDBS und Mediatoren finden

diese Transformationen vorab statt und sind somit nicht Teil der Anfrageverarbeitung. Jedoch bedingen Änderungen der Daten in den einzelnen Quellen im Gegensatz zur virtuellen Integration eine Aktualisierung der Daten im Data Warehouse oder allgemeiner der physisch integrierten Daten. Das kann zu häufigen Ladezyklen führen, um eine stetige Aktualität der Daten im Data Warehouse zu gewährleisten. Dabei kann ein solcher Ladevorgang in Abhängigkeit von der Komplexität der durchzuführenden Datentransformation und der Anzahl der integrierten Quellen umfangreiche Zeit in Anspruch nehmen.

Vergleich der Arten zur Integration von Instanzdaten. Die Tabelle 2.2 vergleicht die beiden oben gezeigten Arten der Integration von Instanzdaten. Wesentlich für die virtuelle Integration ist, dass mit ihr zur Laufzeit der Anfrage auf die originären Quellen zugegriffen wird. Damit werden stets die aktuellen Daten in eine Analyse einbezogen. Jedoch wird die Analyse von den Eigenschaften der Quellen bzw. der angewendeten Wrapper beeinflusst. Dagegen wird im Zuge der physischen Integration eine neue Datenquelle erzeugt, die die relevanten Daten aus den Quellen redundant speichert. Je nach Art und Umfang der Analysen und der aufzunehmenden Daten ist der Einsatz von hochleistungsfähiger Hard- und Software notwendig, um eine leistungsfähige Analyse zu gewährleisten.

2.4 Zusammenfassung

Im Mittelpunkt dieses Kapitels standen generelle Formen der Datenintegration. Es können Integrationsformen danach unterschieden werden, ob sie a priori den Aufbau einer homogenisierten auf die zu integrierenden Daten verlangen oder auf eine solche verzichten. Die homogenisierte Sicht kann in einem applikationsspezifischen globalen Schema, einer generischen Repräsentation oder einer globalen Ontologie bestehen. Ansätze, die auf eine homogenisierte Sicht a priori verzichten, nutzen vielfach Objektkorrespondenzen aus, die im Bereich der Bioinformatik Bestandteil der Datenquellen sind oder mit entsprechenden Analysen (z.B. Sequenz-Alignments) erzeugt werden können. Mengen dieser Objektkorrespondenzen bilden Mappings. Daher werden Ansätze, die diese Objektkorrespondenzen explizit verwenden, im Sinne dieser Arbeit als Mapping-basierte Ansätze bezeichnet. Ein anderes Klassifikationsmerkmal betrifft die Art der Instanzdatenintegration, mit der die Integrationsansätze unterschieden werden können, je nachdem ob sie eine virtuelle, physische oder hybride Datenintegration verfolgen. Abschließend wurden die sich aus den Klassifikationsmerkmalen ergebenden Integrations-

formen mit einander verglichen. Die genannten Integrationsformen sollen einerseits dem Verständnis der in dieser Arbeit vorgestellten Datenintegrationsplattformen in den Kapiteln 5, 8, und 10 dienen und andererseits bei der Einordnung von verwandten Arbeiten in Kapitel 11 helfen.

Teil II

Data-Warehouse-basierte Datenintegration

Im Mittelpunkt des zweiten Teils steht die Integration von Daten, die von biologischen Experimenten unter Nutzung der Microarray-Technologie produziert werden. Dazu führt Kapitel 3 in die notwendigen biologischen Grundlagen ein, zeigt schematisch den Ablauf von Microarray-basierten Experimenten zur Genexpressionsanalyse und beschreibt die Eigenschaften der daraus resultierenden Daten.

Kapitel 4 gibt die Anforderungen an eine datenbankgestützte Verwaltung dieser experimentellen Daten wieder, anhand derer eine Evaluierung ausgewählter Microarray-Datenbanken vorgenommen wurde. Aufbauend auf den Evaluierungsergebnissen wurde die *GeWare*-Plattform, einem Data Warehouse zur Unterstützung der Genexpressionsanalyse, konzipiert, die in Kapitel 5 vorgestellt wird.

Abschließend folgen zwei ausgewählte Anwendungszenarien der Plattform. Das Kapitel 6 zeigt die Anwendung der *GeWare*-Plattform zur sequenzbasierten Analyse von Microarray-Daten, deren Ergebnisse zur Entwicklung einer neuen Normalisierungsmethode geführt haben. Das Kapitel 7 beschreibt den Einsatz der *GeWare*-Plattform zur Untersuchung molekularer Mechanismen in zwei deutschlandweiten klinischen Studien.

Kapitel 3
Grundlagen der Genexpressionsanalyse

3.1 Biologische Grundlagen

Zellen repräsentieren die kleinsten strukturellen und funktionalen Einheiten eines Organismus [Pas94] unabhängig davon, ob es sich um einzellige Viren und Bakterien oder höhere Lebensformen wie die Vertebraten handelt, zu denen auch der Homo Sapiens gehört. In Hinsicht auf den Aufbau von Zellen lassen sich zwei Arten unterscheiden, eukaryotische und prokaryotische Zellen. Im Gegensatz zu prokaryotischen Zellen, die zumeist bei Einzellern vorzufinden sind, besitzen eukaryotische Zellen einen Zellkern und sind Bestandteil aller höheren Lebensformen. Der Aufbau eukaryotischer Zellen gleicht sich innerhalb eines Organismus, auch wenn die Zellen in unterschiedlichen Geweben lokalisiert sind und verschiedene Aufgaben wahrnehmen. Die Abbildung 3.1 skizziert die wichtigsten Komponenten einer eukaryotischen Zelle. Dazu zählen u.a. das Mitochondrium, in der für den Stoffwechsel wichtige Moleküle abgebaut werden, der Zellkern und das Ribosom. Die Komponenten sind innerhalb der Zelle vom Zellplasma umgeben und werden von der Zellwand begrenzt, die den Abschluss der Zelle markiert.

Nahezu alle Zellen eines Organismus enthalten die gleiche genetische Information, die fundamentalen Charakter insbesondere für Aufbau, Gestalt, Wirkungsweise und Entwicklung des Organismus besitzt. Beispielsweise beschreibt sie Funktionen wie Synthese, Transport und Abbau von Molekülen,

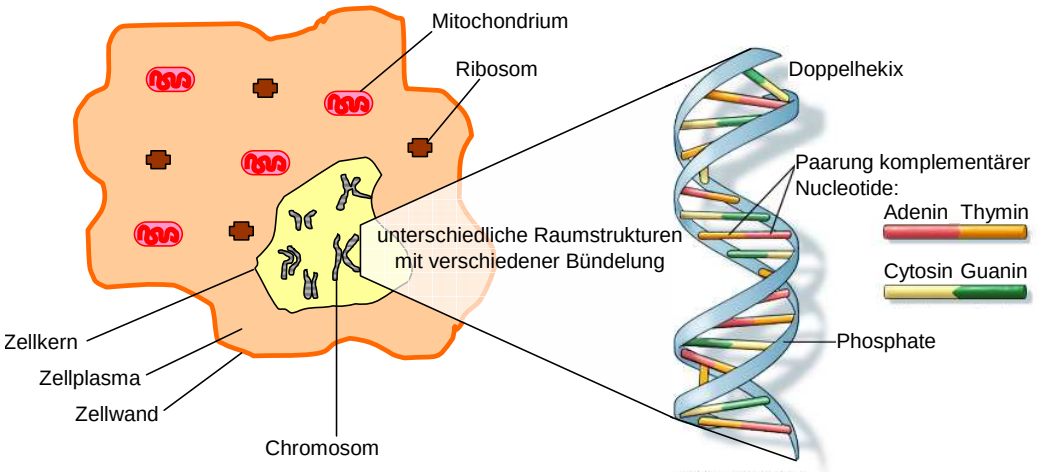


Abbildung 3.1: Grundlegende Bestandteile einer eukaryotischen Zelle

die Grundlage für den Stoffwechsel und die Energiegewinnung aus der aufgenommenen Nahrung sind, aber auch die Zellteilung und -differenzierung, die Wachstum und Entwicklung eines Organismus beeinflussen [Pas94]. Träger der genetischen Information ist die Desoxyribonukleinsäure (DNS/DNA[14]), die bei eukaryotischen Zellen im Zellkern, in den Chromosomen, lokalisiert ist und in ihrer einfachsten Form gemeinhin die Struktur einer gewundenen Doppelhelix aufweist (vgl. Abbildung 3.1).

Die DNA enthält die genetische Information in verschlüsselter Form. Dazu werden die vier chemischen Verbindungen, die so genannten Nukleotide, Adenin (A), Cytosin (C), Guanin (G) und Thymin (T) verwendet. Sie sind in zwei komplementären Strängen von Sequenzen organisiert, die die Doppelhelix bilden. Während Adenin komplementär zu Thymin ist, ergänzen sich Cytosin und Guanin. Die Reihenfolge der Nukleotide in der DNA-Sequenz bestimmt nicht nur molekularbiologische Eigenschaften eines Individuums, sondern hat darüber hinaus auch Einfluss auf dessen phänotypischen Charakter, d.h. beobachtbare, körperliche Eigenschaften des Individuums, wie z.B. Haut- und Haarfarbe.

Nach der Sequenzierung verschiedener Organismen, wie z.B. Homo Sapiens (Mensch), Mus Musculus (Maus) und Drosophila Melanogaster (Fruchtfliege), liegt der Fokus gegenwärtig in der Erforschung von funktionalen Einheiten der DNA, deren Interdependenzen und Abhängigkeiten sowie deren Verhalten unter verschiedenen Bedingungen (z.B. gesunder Organismus vs. verschiedene Krankheiten und deren Stadien). Dazu liefert die Analyse der Genexpression einen Beitrag. Die Genexpression ist ein mehrstufiger Prozess

[14] Die Abkürzung DNA folgt aus der englischen Bezeichnung *desoxyribonuclein acid*.

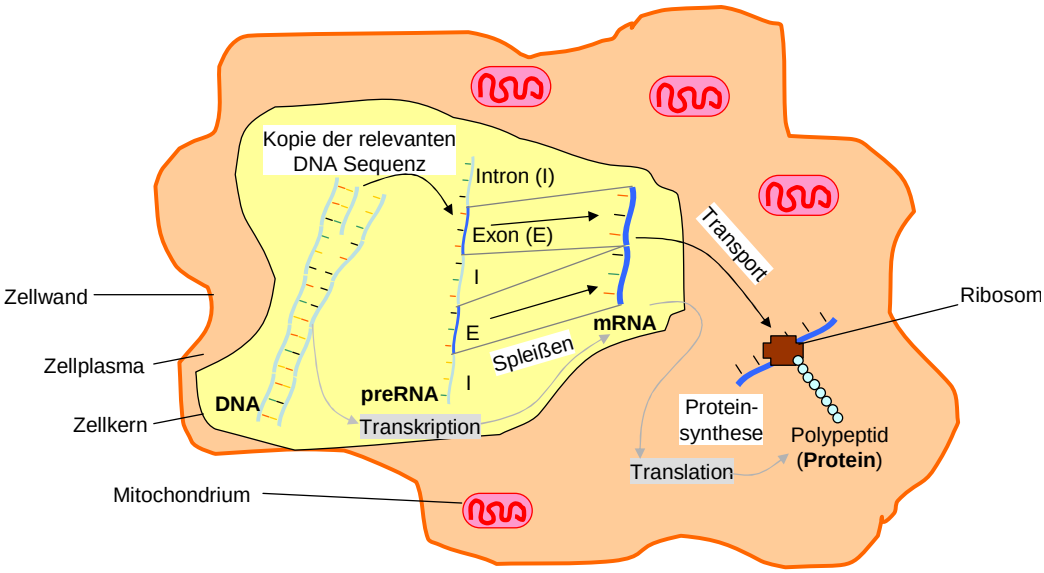


Abbildung 3.2: Schematische Darstellung der Genexpression in einer eukaryotischen Zelle

(vgl. Abbildung 3.2), in dem ausgehend von speziellen und zusammengehörigen Abschnitten der DNA-Sequenz Proteine gebildet werden, die für die meisten Funktionen und Prozesse inner- und außerhalb der Zelle notwendig sind. Diese grundlegenden DNA-Abschnitte werden als Gene bezeichnet.

Der mehrstufige Prozess der Genexpression wird in Abbildung 3.2 skizziert und ist grob in die beiden Teilprozesse Transkription und Translation gegliedert, die sich ihrerseits wiederum in einzelne Teilprozesse zerlegen lassen. Ziel der Transkription ist die Ableitung der einsträngigen Ribonukleinsäure (RNS/RNA[15]) aus einer spezifischen DNA-Sequenz, während die Translation die Proteinbildung zum Ziel hat. Dazu wird im Prozess der Transkription, gesteuert durch einen Enzym-Komplex (spezielle Proteine), die Doppelhelix an einer spezifischen Stelle geöffnet. Im Anschluss wird die relevante DNA-Sequenz (z.B. eines Genes) kopiert, in dem die komplementäre Sequenz zur Sequenz des zum relevanten DNA-Abschnitt gehörenden Gegenstranges erstellt wird. Im Ergebnis dieses Prozesses entsteht die gegenüber der originären DNA-Sequenz leicht veränderte RNA-Sequenz, die preRNA.

Die transkribierte preRNA-Sequenz eines Gens besteht allgemein aus zwei Arten von Teilsequenzen, den Exons und Introns. Während die Exons direkt in die Proteinbildung eingehen, ist dies bei Introns nicht der Fall[16]. Die Länge

[15] Die Abkürzung RNA folgt aus der englischen Bezeichnung *ribonuclein acid*.
[16] Die Funktion der Introns ist derzeit noch weitgehend unerforscht.

und Anordnung von Exons und Introns innerhalb eines Transkripts ist spezifisch für die transkribierte Gensequenz [Pas94]. Im darauf folgenden Schritt, auch Spleißen genannt, werden die Introns, d.h. die nicht proteinkodierenden Abschnitte entfernt. Im Ergebnis des Spleiß-Prozesses wird die so genannte *Boten-RNA* (mRNA[17]) erzeugt. Varianten der mRNA für ein und dieselbe transkribierte Gensequenz entstehen, wenn auch Exons im Spleiß-Prozess entfernt werden; es wird dann von alternativen/differentiellen Spleißen (Prozess) und von Spleiß-Varianten (mRNA) gesprochen [Pas94, SGH+98]. In der anschließenden Translation werden die mRNA-Sequenzen zu einem Ribosom transportiert. Dort werden im Prozess der Proteinsynthese aus der mRNA Polypeptide erzeugt, die aus einer Sequenz von Aminosäuren bestehen und Proteine bilden.

3.2 Microarray-basierte Genexpressionsanalyse

Im Mittelpunkt der Genexpressionsanalyse steht die Frage, welche Gene oder allgemeiner welche DNA-Sequenzen unter welchen Bedingungen und Zeitpunkten im oben skizzierten Prozess der Genexpression aktiv sind und welche nicht. In der Vergangenheit wurden unterschiedliche Technologien entwickelt, um die Genexpression unter verschiedenen Bedingungen differenziert nach Organen und Spezies zu untersuchen. Beispielhaft dafür stehen *Northern Blotting* [AKS77], *Differential Display* [LP92, LP97], *Reverse Transcription-Polymerase Chain Reaction* (RT-PCR) [SWMB95], *Expressed Sequence Tag (EST) Clustering* [VEB+98], *Serial Analysis of Gene Expression* (SAGE) [VZVK95] und *Microarrays* [SSDB95, LDB+96, LFGL99]. Insbesondere die Microarray-Technologie hat sich in den letzten Jahren zu einem dominanten experimentellen Ansatz in Hinsicht auf die Genexpressionsanalyse entwickelt. Sie ermöglicht es im Gegensatz zu anderen Technologien, gleichzeitig die Expression von mehreren Tausenden Genen zu messen.

Prinzip der Microarray-Technologie Microarrays nutzen ein einfaches molekularbiologisches Prinzip: die Bindung von zwei komplementären Strängen von Nukleotidsequenzen. Auf einem Microarray, der auch als Chip bezeichnet wird, werden bekannte Sequenzen an einer definierten Position, den so genannten *spots*, befestigt. Unbekannte Sequenzen werden identifiziert, in dem sie auf den Microarray gegeben werden. Dort binden sie in einem längeren chemischen Prozess (Hybridisierung) an die auf dem Chip befestigten

[17] Die Abkürzung entstammt der englischen Bezeichnung *messenger ribonuclein acid*.

Sequenzen, so dass komplementäre Paare zwischen den aufgebrachten und zugegebenen Sequenzen entstehen. Mit der Analyse der gebundenen Sequenzen kann festgestellt werden, welche Sequenzen zu identifizieren waren. In Abhängigkeit von Typ und Länge der auf dem Chip befestigten Sequenzen sind zwei Varianten von Microarrays zu unterscheiden.

- **cDNA-Microarrays**. Diese ältere Technik wurde an der Stanford Universität entwickelt und verwendet komplementäre DNA-Sequenzen (cDNA[18]) mit einer Länge von ca. 500 - 5000 Nukleotiden. Jede dieser Sequenzen repräsentiert ein Gen und wird mit der Oberfläche des Objektträgers fest verbunden.

- **Oligonukleotid-Arrays**. Diese Microarrays nutzen kurze Sequenzen, die so genannten *Oligos*, mit einer Länge von ca. 20 - 80 Nukleotiden. Im Gegensatz zu den cDNA-Microarrays wird ein Gen durch eine Menge von Oligos repräsentiert. Diese Technik wurde maßgeblich von der Fa. Affymetrix Inc.[19] entwickelt und propagiert.

Zusätzlich wird in Abhängigkeit von der Anzahl an Proben (Zellsegmente), aus denen die zu untersuchenden Sequenzen stammen, nach Ein- und Zweiproben-Microarrays unterschieden. Alternativ haben sich auch die Bezeichnungen ein- und zwei-/mehrfarbige Microarrays etabliert, wobei letztere den Zweiproben-Microarrays entsprechen, da sie zwei verschiedene Farbstoffe zur Kennzeichnung der Sequenzen aus den beiden Gewebeproben verwenden. Die Zweiproben-Microarrays ermöglichen einen direkten Vergleich des Expressionsverhaltens von Genen aus zwei verschiedenen Proben im Experiment. Damit wird sofort eine Überexpression von Genen der einen Probe farblich sichtbar und kann direkt aus den Ergebnissen abgelesen werden. Im Gegensatz zu den Einproben-Microarrays verringern sie die experimentelle Fluktuationen, die beispielsweise von unterschiedlichen Rahmenbedingungen während der Hybridisierung (z.B. Dauer und Temperatur) herrühren. Jedoch ist ihr Einsatz mit neuen Fehlerquellen behaftet; insbesondere die unterschiedliche Menge und Qualität der beiden verwendeten Proben führt zu verschiedenen Ergebnissen. Derzeit werden vorrangig einfarbige Microarrays verwendet.

Experimenteller Ablauf Die Abbildung 3.3 gibt einen Überblick zum Ablauf der Genexpressionsanalyse bei Ein- und Zweiproben-Microarrays. Die Analyse beruht auf der Annahme, dass aktivierte Gene in einer Zelle im Prozess der Genexpression eine Menge von mRNA-Sequenzen erzeugen. Diese

[18] Die Abkürzung folgt aus der englischen Bezeichnung *complementary DNA*.
[19] `http:\\www.affymetrix.com`

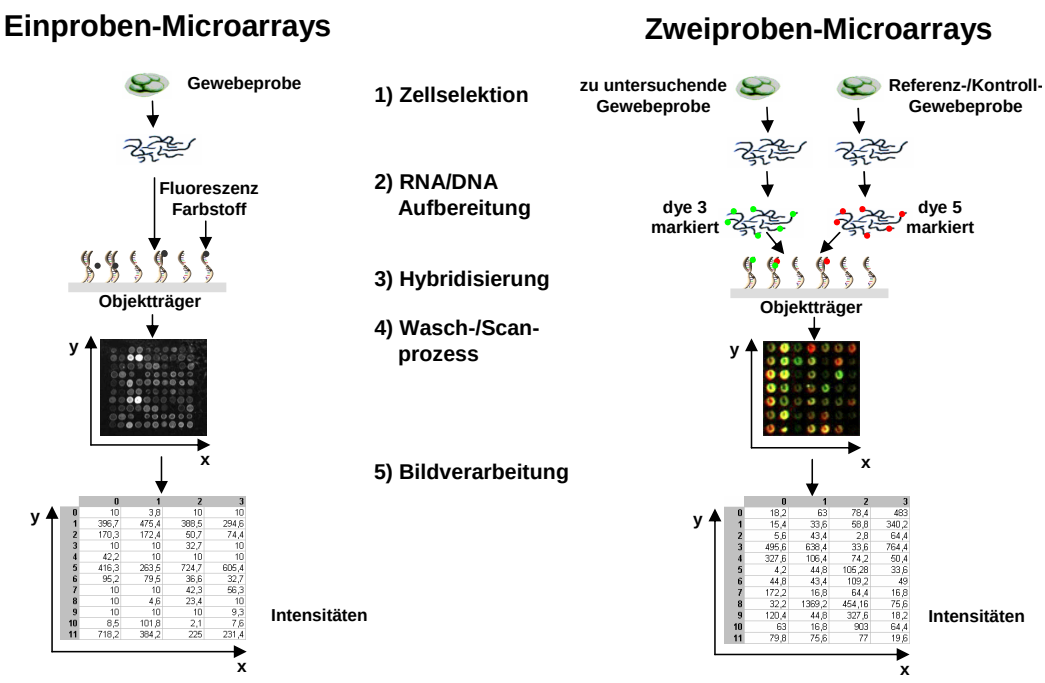


Abbildung 3.3: Schematischer Ablauf eines Microarray Experiments

Sequenzen müssen daher in einem ersten Schritt aus einer Zelle der ausgewählten Gewebeprobe (Schritt 1) extrahiert werden. Nachfolgend (Schritt 2) werden die Sequenzen fragmentiert und mit einem Fluoreszenz-Farbstoff markiert. Im Gegensatz zu Einproben-Microarrays werden bei Zweiproben-Microarrays verschiedene Farbstoffe verwendet, um die gebundenen Sequenzen beider Proben in der späteren Auswertung unterscheiden zu können. Hiernach werden die farblich markierten Sequenzen in einer speziellen Mixtur auf den Microarray gegeben, so dass anschließend - unter Einhaltung spezieller Bedingungen, wie z.B. der Temperatur - der Bindungsprozess (Schritt 3) stattfinden kann. Nach Ablauf der Hybridisierung werden nicht gebundene Sequenzfragmente in einem Waschvorgang entfernt. Hiernach wird der gesäuberte Microarray gescannt (Schritt 4). Das Ergebnis besteht in einem Bild mit verschiedenen Farb- und Helligkeitsintensitäten, die anzeigen, wie viel Sequenzen mit Fluoreszenz-Farbstoffen gebunden haben.

In der in Schritt 5 stattfindenden Bildverarbeitung entsteht für jede auf dem Microarray befestigte Sequenz ein numerischer Wert, der die korrespondierende positionsspezifische Helligkeit im Bild und damit die Menge an gebundenen Sequenzen anzeigt. Während das Ergebnis von Einproben-Microarrays aus absoluten Werten besteht, resultieren aus dem Einsatz von Zweiproben-Microarrays relative Werte, die das Verhältnis zwischen beiden

Proben wiedergeben. Im Gegensatz zu cDNA-Arrays, wo jeder Wert als Grad der Aktivität eines Genes interpretiert werden kann, resultiert im Falle von Oligonukleotid-Arrays je ein Wert für ein Oligo, die noch zu einem Expressionswert für das korrespondierende Gen aggregiert werden müssen. Diese Expressionswerte sind Grundlage der sich anschließenden Auswertung, die mit verschiedenen Methoden und Verfahren durchgeführt werden kann.

3.3 Chip-basierte Mutationsanalyse

Die Verfügbarkeit einer speziesspezifischen Genomsequenz in verschiedenen Datenquellen, wie z.B. Ensembl, UniGene und UCSC Genome, impliziert die Annahme, dass eine einzige Genomsequenz existiert, die allen Individuien einer Spezies zugrunde liegt. Jedoch besitzt jedes Individuum (bis auf eineiige Zwillinge) eine einzigartige DNA-Sequenz, die zu ca. 99,9 % mit der Genomsequenz in den frei verfügbaren Datenquellen übereinstimmt [KN01]. Obwohl der Unterschied (0.1 %) sehr gering ist, bestimmt diese Variation der Genomsequenz die individuellen, genetisch bedingten Merkmale eines Individuums. Dazu zählen sowohl Eigenschaften, wie beispielsweise die Haar- oder Augenfarbe, als auch die Anfälligkeit für eine bzw. das Vorhandensein einer genetisch bedingten Krankheit. Gerade letztere werden durch Sequenzvariationen in den kodierenden und regulatorischen Abschnitten eines Genes beeinflusst [GCS00]. Sequenzvariationen beruhen auf Mutationen, d.h. auf Einschüben (Insertion) neuer Sequenzen sowie Ersetzungen (Substitution), Löschungen (Deletion) und Verschiebungen von Sequenzen. Positionsspezifische Mutationen, d.h. spezielle Positionen im Genom, an denen mehrere Nukleotide vorkommen können, sind die häufigste Art genetischer Variation [WFS+98]; längere, blockweise Mutationen sind häufig mit genetisch bedingten Krankheiten (z.B. Trisonomie 21 [Pas94]) verbunden.

Eine Mutationsanalyse untersucht die genetische Variabilität eines Individuums in Bezug zur propagierten (Referenz-) Genomsequenz, wie sie in öffentlichen Datenquellen vorhanden ist. Dazu wurden verschiedene Techniken entwickelt. Einige von ihnen basieren auf der Chip-Technologie - ähnlich der Microarrays in der Genexpressionsanalyse - und ermöglichen somit eine genomweite Analyse. Zu diesen Chip-basierten Techniken zählen beispielsweise Matrix-CGH-Arrays[20] [KKS+92, SLS+97] und SNP-Arrays[21]. Das experimentelle Vorgehen für diese beiden Chip-Techniken ist dem der Genexpressionsanalyse unter Verwendung von Microarrays sehr ähnlich (vgl. hierzu Abschnitt 3.2). Beide Chip-Techniken verwenden bekannte Sequenzen an die in

[20] Die Abkürzung CGH steht für "Comparative Genomic Hybridization" (engl.).
[21] Das Akronym SNP steht für "Single Nucleotide Polymorphism" (engl.).

einem Hybridisierungsprozess aus einer Zelle extrahierte Sequenzen binden. Die Zielstellung und die Interpretation der erzielten Ergebnisse unterscheiden sich jedoch sehr stark. Mit Matrix-CGH-Arrays lässt sich die Anzahl von speziell ausgewählten DNA-Sequenzen, den so genannten Clonen, im Genom feststellen. Auf Basis der ermittelten Kopienanzahl kann analysiert werden, ob Mutationen dazu geführt haben, dass eine DNA-Sequenz mehrfach oder nicht mehr im Genom des untersuchten Individuums vertreten ist. Somit kann die Mutationsanalyse unter Verwendung von Matrix-CGH-Arrays dazu genutzt werden, um die ermittelten Werte der Genexpressionsanalyse zu validieren und zu interpretieren. Dagegen zielt die Analyse unter Nutzung von SNP-Arrays darauf, einzelne, positionsspezifische Unterschiede von Nukleotiden zwischen einer ausgewählten DNA-Sequenz des zu untersuchenden Individuums und der (Referenz-) DNA-Sequenz in öffentlichen Datenquellen zu identifizieren [CAI+99, GCS00, KN01].

3.4 Eigenschaften resultierender Genexpressions- und Mutationsdaten

Die Expressionsdaten eines Microarrays zeigen die Aktivität der untersuchten Gen-Sequenz zu einem bestimmten Zeitpunkt (insbesondere bei Zeitreihenanalysen) oder unter einer speziellen Bedingung, die sich beispielsweise auf die Herkunft der Proben (Organe, Patienten etc.) beziehen oder eine spezielle Krankheit bzw. Krankheitsstadium eines Patienten reflektieren können. Mutationsanalysen untersuchen die genetische Diversität von DNA-Sequenzen, den Clonen und SNP-Sequenzen. Ähnlich wie Expressionsanalysen werden sie für spezielle Zellgewebe, Patienten etc. gemessen. Daraus wird deutlich, dass Expressions- und Mutationsdaten multidimensionaler Natur sind: Jeder Messwert kann als Punkt im n-dimensionalen Raum R^n aufgefasst werden. Die Abbildung 3.4 zeigt beispielhaft die drei Dimensionen Analysemethoden, Gene/Clone/SNP und Experimente, die einen Genexpressions-/Mutationswert determinieren (vgl. auch [MT01]).

- **Gene/Clone/SNP**: Diese Dimension beschreibt Objekte, wie Gene und Clone, die von auf einem Chip befestigten Sequenzen repräsentiert werden. Damit sind vergleichende Analysen zwischen zwei oder mehreren Genen (Clone/SNP) bezogen auf deren Expressionswerte (Mutationswerte) möglich.

- **Experimente**: Die Experiment-Dimension subsumiert alle in den Experimenten hybridisierten und gemessenen Chips. Sie stellt damit die

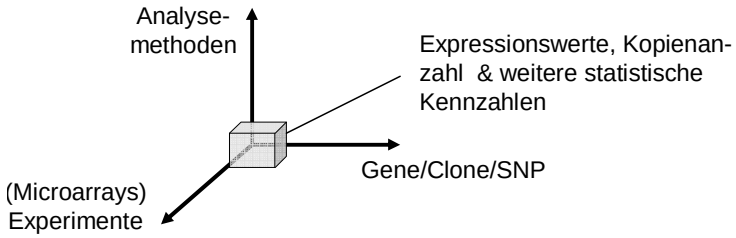


Abbildung 3.4: Multidimensionalität von Genexpressions- und Mutationsdaten

Verbindung zu den zu untersuchenden Proben (z.B. Zellgewebe von Patienten) her, die mit jedem verwendeten Chip assoziiert werden.

- **Analysemethoden**: Die Dimension Analysemethode beinhaltet alle Methoden, die für eine Auswertung der Expressionsdaten relevant sind. Dazu zählen vorverarbeitende Methoden, wie sie beispielsweise zur Normalisierung eingesetzt werden, aber auch elaborierte Methoden, z.B. zum Auffinden co-exprimierter Gene, die auf den Ergebnissen von vorverarbeitenden Methoden aufbauen.

Der numerische Charakter der Expressions- und Mutationsdaten macht multidimensionale Analysen möglich, wie sie bereits aus dem Data Warehousing und OLAP [JLVV03] bekannt sind und in anderen Domänen eingesetzt werden. Mit der Definition von Gruppen innerhalb der verschiedenen Dimensionen, wie z.B. Gen-/Clone-Gruppen und Gruppen von Microarrays (Chips), können verschiedene Aggregationsfunktionen (z.B. $min()$, $max()$, $avg()$) in der Analyse eingesetzt werden. Damit sind typische aus dem OLAP bekannte Analysefunktionalitäten möglich, z.B. *Drill down* und *Roll up*.

3.5 Zusammenfassung

Zellen sind die kleinsten funktionalen Einheiten in einem Organismus, die nahezu alle einen gleichen Aufbau bzw. über die gleichen Bestandteile verfügen. In eukaryotischen Zellen ist die genetische Information in Form von DNA-Sequenzen im Zellkern in den Chromosomen lokalisiert. Gene bilden zusammengehörige Abschnitte auf der DNA, aus denen in einem mehrstufigen Prozess, der Genexpression, Proteine gebildet werden. Dieser Prozess besteht grob aus zwei Teilprozessen, der Transkription und der Translation. Während ein Translation die Proteinsynthese zum Ziel hat, entstehen im Ergebnis der Transkription mRNA-Sequenzen, deren Menge bei der Microarray-

basierten Genexpressionsanalyse gemessen wird. Microarrays ermöglichen gegenüber anderen Technologien ein Studium des Genexpressionsverhaltens, in das mehrere Tausend Gene gleichzeitig einbezogen werden können. Damit sind Analysen möglich, die beispielsweise Abhängigkeiten und Interdependenzen zwischen den Genen sowie den betroffenen molekularen Funktionen und zellulären Prozessen aufzeigen ohne dass eine Beschränkung auf einzelne ausgewählte Gene vorliegt. Mutationsanalysen untersuchen die genetische Diversität eines Zellgewebes, d.h. inwieweit die DNA-Sequenz des untersuchten Zellgewebes von einer Referenz-Sequenz abweicht. Die durch Anwendung dieser experimentellen Technologie entstehenden numerischen Expressions- und Mutationsdaten sind primär durch ihr massives Volumen gekennzeichnet und weisen einen multidimensionalen Charakter auf.

Kapitel 4

Vergleich von datenbankgestützten Analyseplattformen für Microarray-Experimente

4.1 Motivation

Mit der Entwicklung und Anwendung der Microarray-Technologie und dem daraus resultierenden großen Datenvolumen wird eine datenbankgestützte Analyseplattform notwendig, die die anfallenden Daten aufnimmt, verwaltet und deren effiziente Analyse ermöglicht. Verschiedene Plattformen wurden für diesen Zweck entwickelt. Mit dem Fokus auf die zur Verfügung stehenden Analysemöglichkeiten werden einige von ihnen in [GGL01] evaluiert. Hierbei werden jedoch solchen Anforderungen, die aus dem Aufbau, der Wartung und Nutzung einer datenbankgestützten Plattform erwachsen, nicht genügend Rechnung getragen. Ferner sind viele der evaluierten Plattformen weder öffentlich zugreifbar noch in wissenschaftlichen Publikationen beschrieben.

Um einen besseren Überblick über den gegenwärtigen Stand von Analyseplattformen zu bekommen, die die Datenbanktechnologie zur Unterstützung der Genexpressionsanalyse einsetzen, werden in diesem Kapitel acht verfügbare Systeme evaluiert, für die zum Zeitpunkt der Evaluierung (2003) mindes-

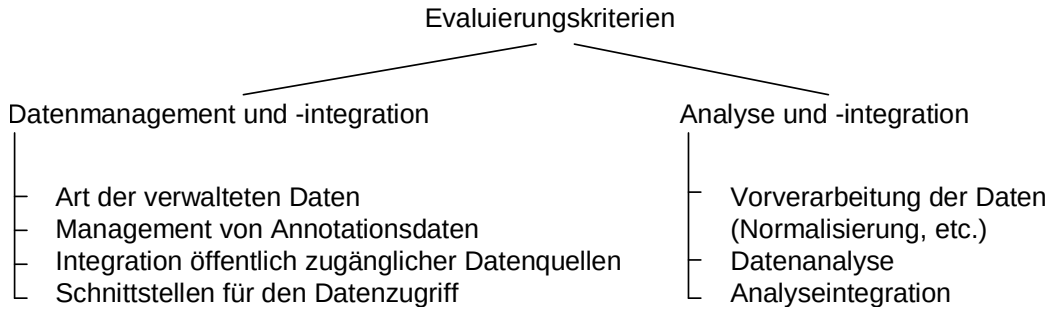


Abbildung 4.1: Evaluierungskriterien

tens eine wissenschaftliche Veröffentlichung verfügbar war. Zu diesem Zweck werden die wichtigsten Anforderungen, die an die Verwaltung von Microarray-Daten gestellt werden, diskutiert. Insbesondere wird hierbei auf datenbankbezogene Sachverhalte eingegangen, wie z.B. Performanzaspekte, Datenintegration und die Kopplung verschiedener Analyse- und Data-Mining-Methoden mit der Datenbank, die in [GGL01] nicht betrachtet wurden.

4.2 Evaluierungskriterien

Die Abbildung 4.1 zeigt die gewählten Evaluierungskriterien im Überblick. Es werden Kriterien des Datenmanagements und -integration sowie der Analyse und -integration unterschieden. Im Folgenden werden die Kriterien vorgestellt; eine Darstellung der Kriterien *Management von Annotationsdaten* und *Integration öffentlich zugänglicher Datenquellen* wird nicht vorgenommen, da die Kapitel 1 und 2 bereits allgemeine Eigenschaften von Datenquellen im Bereich der Bioinformatik, die Notwendigkeit einer Datenintegration sowie generelle Integrationsansätze aufgezeigt haben.

4.2.1 Art der verwalteten Daten

In Genexpressionsanalysen werden verschiedene Arten von Daten verwendet. Für nachfolgende Betrachtungen wird zwischen Bild-, Expressions-, Annotations- und klinischen Daten unterschieden. Die Tabelle 4.1 fasst die Charakteristik der verschiedenen Datenarten und ihre Nutzung in der Genexpressionsanalyse zusammen.

Bilddaten. Die Bilddaten sind das Ergebnis eines Scan-Prozesses von Microarrays und werden in großen Dateien (typischerweise > 10 MB) gespeichert. Sie sind die Grundlage für die im Prozess der Bildverarbeitung ent-

Tabelle 4.1: Relevante Datenarten und deren Charakteristik

Datenart		**Quelle**	**Typ**	**Charakteristik**	**Nutzung**
Bilddaten		Scannen des Arrays	binär	große Dateien	Generierung von Expressionsdaten
Expressionsdaten		Bildanalyse	numerisch	schnell wachsendes Volumen	statistische Analyse, Data Mining, Visualisierung
Annotationsdaten	Gene	externe öffentliche Datenquellen	Text	regelmäßige Aktualisierung	Interpretation von Analyseergebnissen
	experimentelle Metadaten	Benutzereingabe		manuelle Eingabe, oftmals verbale Beschreibung	
klinische Daten		Studienverwaltungssysteme	oftmals Text	unterschiedliche Parameter / Attribute	

stehenden Expressionsdaten. Jedoch kann dieser Bildverarbeitungsprozess Änderungen unterliegen, die sich u.a. mit der Entwicklung leistungsfähigerer Algorithmen mit dem Ziel einer höheren Bildqualität ergeben. Daher sollten die Bilddaten zusätzlich zu den daraus abgeleiteten Expressionsdaten gespeichert werden. Möglichkeiten zur Speicherung der Bilddaten bestehen innerhalb der Datenbank (z.B. als binäres Objekt unter Verwendung des Datentyps BLOB) oder als eigenständige Datei im Dateisystem, auf die die Plattform eine Referenz (z.B. in der zugrunde liegenden Datenbank) besitzt.

Expressionsdaten. Die Expressionsdaten sind numerische Messwerte, die aus den Bilddaten durch bildverarbeitende Methoden entstehen und die Aktivität von Genen widerspiegeln (vgl. Kapitel 3). Mit jedem neuen Microarray entstehen sie in großer Menge; allein dem von Affymetrix vertriebenen Microarray mit der Bezeichnung HG-U133Plus2 liegen mehr als 1.300.000 Oligos zu Grunde, die zu mehr als 47.000 Transkripten und ca. 39.000 Genen korrespondieren (vgl. Kapitel 1). Das Wachstum der Expressionsdaten ist proportional zu dem der Bilddaten. Gegenüber den Bilddaten verwenden Analysen sehr häufig die Expressionsdaten. Auf Grund ihres großen Volumens stellen sie besondere Anforderungen an die Performanz der Analyse; insbesondere in den vielfach verwendeten interaktiven Analysen werden kurze Antwortzeiten

erwartet. Dazu kann der Einsatz von fortgeschrittenen Datenbanktechniken beitragen, die beispielsweise in materialisierte Sichten, Indizierung und Parallelverarbeitung bestehen.

Annotationsdaten. Annotationen sind Metadaten, die dem Benutzer oder Experimentator die Interpretation der gemessenen Expressionswerte und Analyseresultate erleichtern. Es können Genannotationen und experimentelle Metadaten unterschieden werden. Die Genannotationen sind mit den an der Oberfläche eines Microarrays befestigten Sequenzen als Bestandteil von Gensequenzen einer Spezies assoziiert. Sie bestehen in diversen Beschreibungen (Annotation), wie beispielsweise Gennamen (zzgl. der Synonyme), Symbole, bereits bekannte assoziierte Genfunktionen und Krankheiten sowie die Lokalisation im Genom. Diese Daten tragen zur Interpretation der gemessenen Expressionswerte bei. Die experimentellen Metadaten dokumentieren sowohl den Untersuchungsgegenstand (untersuchtes Zellgewebe) als auch den experimentellen Prozess. Mit ihnen wird das Experiment nachvollziehbar bzw. reproduzierbar. Sie ermöglichen es, signifikante Veränderung des Expressionsniveaus aller Genen durch mögliche Fehler im experimentellen Prozess zu erklären. Eine Empfehlung, welche experimentellen Parameter zu dokumentieren sind, gibt die MIAME-Richtlinie[22] [BHQ+01].

Klinische Daten. Klinische Daten umfassen klinische und pathologische Befunde sowie Daten über Patienten. Solche Daten werden in klinischen Studien erfasst und typischerweise in atomarer Form in Studienverwaltungssystemen gespeichert. Je nach Zielstellung der klinischen Studien unterscheiden sich die Parameter, zu denen die Daten erhoben werden. Daraus resultiert, dass die klinischen Daten verschiedener Studien inhaltlich sehr heterogen sein können (vgl. Kapitel 7). Dies sollte bei deren Integration berücksichtigt werden, so dass ebenso wie die Annotationsdaten die klinischen Daten zur Interpretation und dem Vergleich von Analyseergebnissen herangezogen werden können.

4.2.2 Schnittstellen für den Datenzugriff

Schnittstellen, die einen Datenzugriff ermöglichen, können sich zum einen auf den Datenaustausch und zum anderen auf die Benutzerschnittstelle beziehen.

Datenaustausch. Microarray-basierte Experimente werden fortwährend von verschiedenen Benutzern und in unterschiedlichen Laboratorien durchgeführt. Deshalb sollte eine datenbankgestützte Analyseplattform einerseits einen Da-

[22] Eine Diskussion der MIAME-Richtlinie erfolgt in Abschnitt 5.5.

tenimport in Hinsicht auf eine einheitliche Datenverwaltung und -analyse ermöglichen. Andererseits bedingt eine Datenanalyse mit externen Programmen – das sind Programme, die nicht in das System integriert sind – einen Datenexport. Daher wird sowohl ein Datenimport als auch ein -export benötigt.

Ein Datenaustausch wird auf der Grundlage von definierten Datenformaten vollzogen, in die die Daten exportiert oder aus denen die Daten importiert werden. Ein einfaches und häufig verwendetes Format ist das von CSV-Dateien, in denen die Daten tabellenartig und mit einem ausgewählten Zeichen getrennt enthalten sind. Dieses Datenformat kommt auch für die Expressionsdaten zur Anwendung, wobei diese Daten in Form einer Matrix, der so genannten Expressionsmatrix [BHQ+01], organisiert sind. Eine solche Matrix enthält die Expressionswerte für verschiedene Gene, die die Zeilen bezeichnen, und Microarrays, die die Spalten kennzeichnen. Nachteilig für die Verwendung dieses Datenformats ist, dass weder ausgewählte Genannotationen noch experimentelle Metadaten zum Untersuchungsgegenstand und dem experimentellen Prozess enthalten sind, sondern ausschließlich die experimentellen Daten. Ein Lösungsansatz wird mit dem Einsatz von XML-basierten Datenformaten gegeben, die die Expressions- und Annotationsdaten semistrukturiert aufnehmen. Verschiedene propagierte Ansätze für ein standardisiertes XML-basiertes Datenaustauschformat in Hinsicht auf Microarray-basierte Expressionsdaten stellen beispielsweise MAGE-ML[23], GEML[24] und GeneXML[25] dar.

Zugriffskontrolle. Typischerweise unterliegen die gemessenen Expressionsdaten und Analyseresultate einem restriktiven Zugriff. Dies liegt zum einen daran, dass die Experimentatoren und beteiligten Forschungsgruppen den durch das Experiment vermeintlich gewonnenen Forschungsvorsprung ausnutzen und zuerst die Daten vollständig auswerten wollen, bevor andere sich derer bedienen. Zum anderen sind gerade Microarray-Experimente mit nicht unwesentlichen Kosten verbunden, so dass die erzielten Ergebnisse zuerst in Journalen oder als Konferenzbeiträge publiziert werden. Vielfach geht eine Publikation der Analyseergebnisse mit einer Veröffentlichung der zugrunde liegenden Daten einher, wie es von führenden Journalen [BSP+02a, BSP+02b, BSP+02c] für eine Nachvollziehbarkeit und Validierung der Ergebnisse gefordert wird. Daher ist eine Limitierung des Benutzerzugriffs ein wichtiges Kriterium für die Akzeptanz solcher Systeme.

[23] http://www.mged.org/Workgroups/MAGE/mage-ml.html
[24] http://www.rosettabio.com/products/conductor/geml/default.html
[25] http://www.ncgr.org/genex/genexml.html

4.2.3 Vorverarbeitung der Daten

Die Expressionsdaten beinhalten experimentelle Fehler, die sich z.B. aus unterschiedlichen Umgebungsbedingungen von Proben und Parametereinstellungen der benutzten Hard- und Software vor, während und nach dem Laborexperiment ergeben. Daher ist eine Normalisierung der Daten unumgänglich, um eine Vergleichbarkeit der experimentellen Ergebnisse sicherzustellen. Darüber hinaus enthalten die Daten auf Grund von vielfältigen Einflüssen während des Experiments und der experimentellen Technik selbst ein "Rauschen", das für jeden Microarray verschieden sein kann und somit korrigiert werden muss.

Einige der gegenwärtigen Methoden zur Vorverarbeitung sind auf einen einzelnen ausgewählten Array ausgerichtet; andere hingegen beziehen eine Menge von Arrays in die Verarbeitung ein. Eine Beschreibung und Evaluierung von verschiedenen Methoden zur Vorverarbeitung von Expressionsdaten basierend auf cDNA- und Oligo-Arrays wird beispielsweise in [SBM+00, HBW+01, HS06] gegeben. Jedoch existiert für die Vorverarbeitung der Daten noch kein abgestimmtes Standardverfahren. Daher sollten dem Benutzer mehrere repräsentative Methoden zur Verfügung stehen. Neben den resultierenden Daten sollten auch die Rohdaten gespeichert werden, um eine erneute Vorverarbeitung unter Nutzung einer anderen Methode bzw. veränderter Parametereinstellung zu ermöglichen.

4.2.4 Datenanalyse

Eine Datenanalyse verwendet die aus der Vorverarbeitung resultierenden Daten mit dem Ziel, das Verhalten der Gene in Abhängigkeit vom Untersuchungsgegenstand und deren Bedingungen (z.B. spezielle Krankheiten, Krankheitsstadien) zu studieren. Viele Methoden, die in der Genexpressionsanalyse angewendet werden, wurden bereits in anderen Gebieten genutzt und erprobt. Es kann zwischen den folgenden Gruppen von Analyseansätzen unterschieden werden.

Berichte und Abfragen. Für häufig verwendete Abfragen sollten vordefinierte Berichte unterstützt werden, in denen die Attributmengen für Projektion und Selektion festgelegt sind. Solche Berichte können vom Benutzer zu jeder Zeit und mit verschiedenen Eingabewerten in Hinsicht auf die vorgegebenen Selektionsattribute ausgeführt werden. Damit kann der Benutzer einen Überblick über die Daten und deren Eigenschaften gewinnen und z.B. potentielle Ausreißer erkennen. Solche Berichte nutzen oftmals einfache Metriken der deskriptiven Statistik wie Durchschnitt, Standardabweichung oder

Varianz. Die ermittelten Ausreißer können anschließend einer vertieften Analyse zugeführt werden, um Erklärungen für ihr Auftreten zu finden. Hierfür eignet sich ebenso die Assoziation mit verfügbaren Annotationsdaten. In anderen Analysen können die erkannten Ausreißer ausgeschlossen und etwaige Verzerrungen (so genannte Artefakte) in den Daten auf Grund von Mess- oder experimenteller Fehler vermieden werden.

Statistik. Fortgeschrittene statistische Analysen werden vor allem dann notwendig, wenn Daten von verschiedenen Microarrays einbezogen werden, um beispielsweise signifikante, differentiell exprimierte Gene zu finden. Eine weit verbreitete Analyse ist die Varianzanalyse ANOVA [Lin74, KMC00]. Weitere statistische Techniken bestehen in Signifikanztests, wie z.B. dem Permutationstest und der p-Value Adjustierung [WY93, GDS03], dem t-Test sowie dem Wilcoxon-Test [Geh65, TOTZ01, DYCS02].

Data Mining. Mit Methoden des Data Mining können interessante und bislang unbekannte Muster in großen Datenmengen erkannt werden [WF00]. Es wird zwischen überwachten (engl. supervised) und nicht überwachten (engl. unsupervised) Analyseansätzen unterschieden. Zu den überwachten Analyseansätzen gehören beispielsweise Klassifikationsmethoden, während die Clusteranalyse (engl. Clustering) ein typischer Vertreter von unüberwachten Analyseansätzen ist. Auf beide wird im Folgenden kurz eingegangen.

- *Clusteranalyse*: Eine Clusteranalyse [KR90], angewendet auf die Expressionsdaten, hat zum Ziel, Objekte, d.h. Gene und Microarrays, mit gleichen oder ähnlichen Eigenschaften zu gruppieren. Verschiedene Algorithmen, wie z.B. das hierarchische Clustering, K-means und selbstorganisierende Karten (SOM) [Koh97], wurden bereits auf die Expressionsdaten angewendet [ABN+99, BDSY99]. Typischerweise werden Cluster von Genen gebildet, um co-regulierte und funktional verwandte oder in Beziehung stehende Gene zu identifizieren [Knu02].

  Eine Clusteranalyse wird oftmals von Methoden mit dem Ziel der Dimensionsreduzierung begleitet, um wenig informative Dimensionen identifizieren oder entfernen zu können. Das ermöglicht es, die Clusteranalyse auf eine relevante Teilmenge von Daten anzuwenden und führt auch damit zur Verbesserung der Laufzeit der Analyse. Typische Methoden der Dimensionsreduzierung umfassen *Multidimensional Scaling* (MDS), *Principal Component Analysis*[26] (PCA) und *Single Value Decomposition* (SVD).

[26] Hauptkomponentenanalyse

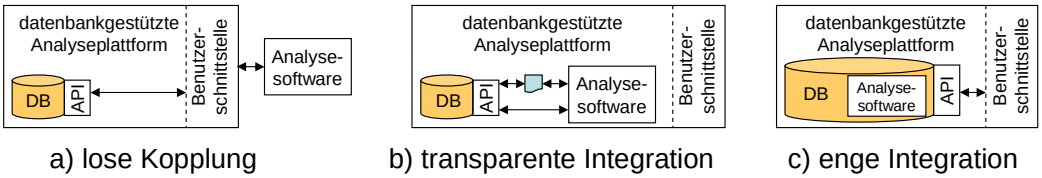


Abbildung 4.2: Formen der Analyseintegration

- *Klassifikation*: Im Gegensatz zur Clusteranalyse zielt die Klassifikation auf die Zuordnung der Objekte zu bereits bekannten Klassen (Cluster oder Gruppen) ab. Dazu werden die Daten in zwei repräsentative Gruppen, den Trainings- und Testdaten, separiert. Die Trainingsdaten dienen den Klassifikationsmethoden, um eine Klassifizierung zu erlernen, deren Güte im Anschluss unter Nutzung der Testdaten validiert wird. Gegenwärtige Klassifizierungsmethoden umfassen beispielsweise die Diskriminanzanalyse [BB89], Entscheidungsbäume [WF00] und *Support Vector Machines* (SVM) [Vap95, BGL+00, GWBV02].

Visualisierung. Um die erzielten Ergebnisse dem Benutzer zu präsentieren, bedarf es einer visuellen Datenaufbereitung. Neben der tabellarischen Darstellung sind unterschiedliche Diagrammformen notwendig, wie beispielsweise Punkt-, Linien- und Balkendiagramme sowie Dendrogramme. Letztere visualisieren das Ergebnis einer hierarchischen Clusteranalyse. So genannte "heat maps" repräsentieren die Werte einer Expressionsmatrix auf Grund einer vorgegebenen farbigen Skala, die definierten Wertebereichen Farben zuordnet. Damit kann der Benutzer visuell Gruppen von ähnlich exprimierten Genen oder Microarrays mit einem ähnlichen Expressionsverhalten identifizieren.

4.2.5 Analyseintegration

Gewöhnlich werden die Daten mit verschiedenen Methoden und in Interaktion mit dem Benutzer analysiert. Viele der oben angeführten Analyse- und Visualisierungsmethoden sind bereits in verschiedenen leistungsstarken Programmsystemen implementiert. Diese sollten zusammen mit dem Auswertungssystem für die Analyse der Microarray-Daten genutzt werden und helfen den Entwicklungsaufwand zu reduzieren. Die Abbildung 4.2 zeigt drei Integrationsformen, nach denen eine Analyseintegration unterschieden werden kann.

Lose Kopplung. In diesem Szenario exportiert der Benutzer eine für ihn interessante Teilmenge der Daten aus dem Auswertungssystem in eine Datei, die er im Anschluss in das Analyseprogramm importiert. Ein Nachteil

dieser Integrationsform besteht darin, dass die Annotationen zumeist nicht in das Analyseprogramm übernommen werden können, das eine gemeinsame Darstellung von Analyseergebnissen und Annotationen verhindert und damit eine Interpretation erschwert.

Transparente Integration. Dieser Ansatz wird angewendet, um Analyseprogramme, die ein API für den Zugriff auf ihre Analysefunktionen anbieten, zu integrieren. Dazu wird eine Benutzerschnittstelle aufgebaut, die die verschiedenen Schritte der Expressionsanalyse unter Nutzung von unterschiedlichen Analyseprogrammen abdeckt. Für den Datenaustausch zwischen den Analyseprogrammen und der Datenbank kann einerseits das Datenbank-API herangezogen oder dateibasiert, d.h. Export der Daten in Dateien mit anschließendem Import in das Analyseprogramm, durchgeführt werden. Dabei sind dem Benutzer die Datenaustauschprozesse transparent.

Enge Integration. Eine enge Analyseintegration zielt auf den Einsatz von Analysemethoden, die direkt im Datenbank-Management-System (DBMS) ausgeführt werden können. Damit kann direkt auf die Daten in der Datenbank zugegriffen werden. Dadurch wird einerseits ein zeitaufwändiger Export der Daten aus der Datenbank vermieden, was vor allem eine kürzere Gesamtlaufzeit nach sich zieht. Andererseits geht der Einsatz von in das DBMS integrierten Analysemethoden mit einer effizienteren Speichernutzung einher, da die Daten nicht von den datenbankinternen in von einer Programmiersprache abhängige Datentypen konvertiert werden müssen. Jedoch impliziert diese Integrationsform einen hohen Implementierungsaufwand, da bereits implementierte Analysealgorithmen als neue Datenbankapplikationen oder als Teil des DBMS (z.B. gespeicherte Prozeduren, spezielle Typerweiterungen) reimplementiert werden müssen.

4.3 Systemevaluierung

4.3.1 Ausgewählte Systeme

Für die im Folgenden dargestellte Evaluierung wurden die Plattformen *ArrayDB* [ERP+98], *ExpressDB* [ARC00], *GeneX* [MSZ+01], *GIMS* [PKH+00, CPW+01, CPH+03], *M-CHIPS* [FHB+01, FHB+02], *RAD2* [SPM+01], *SMD* [SHBK+01] und *YMD* [CWH+02] ausgewählt. Alle diese Systeme waren zum Zeitpunkt der Evaluierung (2003) öffentlich zugänglich und durch mindestens eine wissenschaftliche Publikation ausreichend beschrieben. Anhang A gibt einen Überblick über die Herkunft der evaluierten System und listet die ver-

Tabelle 4.2: Technische Implementierung

Systeme	Open Source	DBMS	Programmiersprache	GUI
ArrayDB	ja	Sybase	Perl, Java	Web
ExpressDB			Perl, JavaScript	
GeneX		PostgreSQL, Sybase	Perl, Java, R	
GIMS	nein	POET	Java	Java Applet
M-CHIPS		PostgreSQL	C, Perl, MatLab	Web
RAD2		Oracle	Perl, Java	
SMD	ja		Perl, C	
YMD	nein		Perl	

wendeten Web-Adressen für den Zugriff über das Internet auf.

Weitere Plattformen sind beispielsweise ArrayExpress [BPS+03, PSS+05], Gene Expression Omnibus (GEO) [EDL02, BST+05], Gene Expression Atlas [SCC+02], HugeIndex [HWB+02], Yeast Microarray Global Viewer (yMGV) [CDJM02] und Riken Expression Array Database (READ) [BKHO02]. Jedoch boten sie zum Zeitpunkt der Evaluierung (2003) entweder einen eingeschränkten Zugriff, waren im Aufbau begriffen oder nicht ausreichend in einer verfügbaren Publikation beschrieben, so dass sie nicht in die folgende Evaluierung einbezogen wurden. ArrayExpress und GEO haben hiernach als zentrale Repositories Expressionsdaten am European Bioinformatics Institute und NCBI Geltung erlangt. In den folgenden Abschnitten werden ausgewählte Evaluierungsergebnisse vorgestellt; [DKR03] enthält eine vollständige Aufstellung der erzielten Ergebnisse.

4.3.2 Technische Implementierung

Die Tabelle 4.2 zeigt die technische Implementierung der betrachteten Systeme. Obwohl alle Systeme über das Internet erreichbar sind, wurden sie dazu angelegt, die in den lokalen Laboratorien produzierten Microarray-Daten zu speichern und auszuwerten. Trotzdem bieten sie die Möglichkeit für externe Benutzer, auf publizierte Daten zuzugreifen und anhand von definierten Abfragen Daten zu selektieren. Einige Systeme (ArrayDB, ExpressDB, GeneX und SMD) sind als softwaretechnische Lösung öffentlich verfügbar und erlauben damit eine lokale Installation. Alle evaluierten Systeme verwenden ein Datenbank-Management-System (DBMS), wobei vielfach ein relationales DBMS eingesetzt wird. Einzig GIMS nutzt das objektorientierte DBMS

Tabelle 4.3: Unterstützung von Bild- und Expressionsdaten

Systeme	Bilddaten	Expressionsdaten
ArrayDB	im Dateisystem	cDNA
ExpressDB	keine Unterstützung	cDNA, Oligo, SAGE
GeneX		
GIMS		cDNA
M-CHIPS		cDNA, Oligo, SAGE
RAD2		
SMD	im Dateisystem	cDNA
YMD		cDNA, Oligo

POET. Ebenso verwenden alle betrachteten Datenbanken, außer GIMS, eine web-basierte Benutzerschnittstelle, die mit der gegenwärtig zur Verfügung stehenden Web-Technologie und überwiegend durch die Programmiersprachen Perl, Java und JavaScript implementiert wurden. Für die Analyse der Daten greifen einige Systeme auf externe Analyseprogramme zurück, die beispielsweise in der Statistiksoftware R und dem wissenschaftliche Softwarepaket MatLab bestehen.

4.3.3 Art der verwalteten Daten

Die Tabelle 4.3 zeigt, inwieweit die evaluierten Systeme Bild- und die aus verschiedenen Technologien resultierenden Expressionsdaten verwalten. ArrayDB, SMD und YMD unterstützen die Speicherung von Bilddaten, wobei keines der drei genannten Systeme die Bilddaten mit dem DBMS verwaltet. Vielmehr werden die Bilddaten im Dateisystem eines Servers abgespeichert. Das DBMS enthält lediglich Referenzen auf die Bilddaten im Dateisystem.

Expressionsdaten, die unter Nutzung von cDNA-Microarrays entstehen, können von allen Systemen verwaltet werden. Darüber hinaus sind einige Systeme in der Lage, Expressionsdaten weiterer Technologien aufzunehmen. Dazu zählen insbesondere Oligo-basierte Microarrays, wie sie in großen Mengen von der Fa. Affymetrix hergestellt und vertrieben werden. Die Systeme ExpressDB, GeneX, M-Chips und RAD2 unterstützen zusätzlich Expressionsdaten der SAGE-Technologie. SMD repräsentiert das System mit dem größten Datenvolumen. Es enthält Expressionsdaten von mehr als 25 Tausend Microarrays unterschiedlicher Typen, denen insgesamt mehr als 538 Millionen Expressionswerte zugeordnet sind. Trotz des großen Datenvolumens und den Anforderungen an die Abfragegeschwindigkeit, existieren keine Erfahrungs-

Tabelle 4.4: Spezifikation und Speicherung experimenteller Metadaten

Systeme	Spezifikation	Speicherung
ArrayDB	keine kontrollierten Vokabulare	parameterspezifische Attribute im DB-Schema
ExpressDB		
GeneX	lokale Vokabulare	
GIMS	–	–
M-CHIPS	lokale Vokabulare	generisch (EAV)
RAD2	Standardvokabulare	parameterspezifische Attribute im DB-Schema
SMD	lokale Vokabulare	hybrid
YMD	–	–

berichte, die den Einsatz von Datenbanktechniken zur Performanzsteigerung wie Indizes, materialisierte Sichten und Parallelverarbeitung betreffen.

4.3.4 Management von Annotationsdaten

Die Tabelle 4.4 zeigt, wie die betrachteten Systeme eine konsistente Spezifikation und flexible Verwaltung von experimentellen Metadaten realisieren. Vielfach werden zusätzliche Textfelder zur Verfügung gestellt, in denen eine verbale Beschreibungen vorgenommen werden kann. ExpressDB benutzt keine spezifischen Parameter; es steht ein einzelnes Textfeld zur umfassenden Beschreibung zu Verfügung. Im Gegensatz dazu nutzt M-CHIPS umfangreiche Listen von hierarchisch definierten Parametern, mit denen eine einheitliche Annotation über mehrere Experimente hinweg vorgenommen werden kann. Die möglichen Annotationswerte sind ebenso Bestandteil der lokal definierten Listen, so dass eine konsistente Erfassung der experimentellen Metadaten sichergestellt wird. Andere Systeme versuchen die negativen Effekte von Freitextfeldern zu minimieren, in dem sie deren Einsatz reduzieren und stattdessen kontrollierte Vokabulare verwenden. Während in GeneX und SMD die Vokabulare durch die lokalen Benutzer entwickelt und modifiziert werden, nutzt RAD2 primär externe Quellen, wie z.B. NCBI Taxonomie, MGD Maus Anatomy und die in KEGG definierten Krankheiten.

Typischerweise nutzen die betrachteten Systeme eine Standardrepräsentation mit vordefinierten Attributen zur Aufnahme der experimentellen Metadaten. Nur M-CHIPS und SMD folgen dem EAV-Ansatz und sind damit flexibler gegenüber strukturellen Änderungen. Im Gegensatz zu M-CHIPS folgt SMD jedoch einer hybriden Lösung, d.h. grundlegende Attribute sind

Tabelle 4.5: Integrierte Genannotationen öffentlicher Datenquellen

Systeme	Web-Links	förderierte Integration	materialisierte Integration	autom. Aktualisierung
ArrayDB	dbEST, GenBank KEGG, UniGene	keine	keine	–
ExpressDB	BIGED		MIPS	nein
GeneX	dbEST, GenBank, KEGG, MGD, SGD, SwissProt		keine	–
GIMS	keine		MIPS	nein
M-CHIPS	keine		GeneOntology	nein
RAD2	AllGenes, GenBank, KEGG		keine	–
SMD	dbEST, GenMap, LocusLink, SwissProt		GeneOntology, WormDB, UniGene	ja
YMD	DRAGON, SOURCE, UniGene		keine	–

Bestandteil des DB-Schemas, während der EAV-Ansatz die notwendige Flexibilität für neue Parameter sicherstellt.

4.3.5 Datenintegration

Die Tabelle 4.5 zeigt die externen, öffentlichen Datenquellen je System, die unter Nutzung einer speziellen Datenintegrationsform integriert werden. Die Nutzung von Web-Links ist die am häufigsten verwendete Integrationsform. Trotz der Limitierung, dass die Daten aus den externen Quellen nicht programmtechnisch in die Analyse der Expressionsdaten einbezogen werden können, bleibt der Integrationsaufwand in vielen Fällen gering. Somit stehen Systeme, die diesem Ansatz folgen, schnell für eine Benutzung zur Verfügung. Web-Links werden vor allem zu den Datenquellen UniGene, GenBank, Swissprot und KEGG eingesetzt.

Im Gegensatz zur Integration auf Basis von Web-Links setzt kein System den förderierten Ansatz zur virtuellen Integration ein. Der materialisierte Ansatz wird ebenso nur wenig angewendet; er kommt für die Integration ausgewählter Genannotationen zur Anwendung. Dazu werden die relevanten Daten in die lokale Datenbank importiert, wobei der Importprozess wie in ExpressDB und GIMS einmalig bzw. unregelmäßig durchgeführt wird. Ein

Tabelle 4.6: Unterstützte Normalisierungsstrategien

Systeme	Normalisierungsmethoden	Expressionsdaten
ArrayDB	keine	–
ExpressDB		–
GeneX	Norm. bzgl. eines Kontrollexperiments und durchschn. Expression	Roh- & normalisierte Daten
GIMS	keine	–
M-CHIPS	robuste affine-linear Regression bzgl. eines Kontrollexperiments	Roh- & normalisierte Daten
RAD2	Norm. bzgl. Gesamtexpression und Kontrollexperiment	
SMD	Strategien mit Skalierungsfaktoren	
YMD	keine	–

Mechanismus, um die integrierten Quellen zu aktualisieren, wird dagegen von vielen Systemen nicht zur Verfügung gestellt. Einzig SMD nutzt unter den evaluierten Systemen eine materialisierte Integrationsform für ausgewählte Genannotationen, die automatisch aktualisiert werden.

4.3.6 Schnittstellen für den Datenzugriff

Der Datenaustausch bleibt zumeist auf die Expressionsdaten (Gen-Ebene) reduziert. Dazu werden vielfach CSV-Dateien verwendet. Darüber hinaus erlauben SMD und YMD den Datenimport von proprietären Dateiformaten, die von der Bildverarbeitungssoftware Genepix und Scanalize genutzt werden. Ebenso unterstützen sie den Datenexport in Formate, die von externen Analysetools benötigt werden, wie z.B. TreeView, CLUSTER und Excel. Dagegen zeigt GeneX erste Bestrebungen, XML als Datenaustauschformat zu etablieren. Mit GeneXML können sowohl Expressionsdaten als auch Annotationsdaten ausgetauscht werden.

4.3.7 Vorverarbeitung der Daten

Die Tabelle 4.6 zeigt die von den evaluierten Systemen unterstützten Normalisierungsstrategien. ArrayDB, ExpressDB und GIMS verfügen über keine Funktionen zur Normalisierung der Daten. Sie bieten dem Benutzer lediglich die Möglichkeit, Daten in das System zu laden und anschließend zu analysieren. Eine Normalisierung bleibt evtl. unter Hinzuziehung externer Tools dem

Tabelle 4.7: Abfrage- und Berichtsmöglichkeiten

Systems	Software Tools	Integration	Funktionalität
ArrayDB	ArrayViewer, MultiExperiment-Viewer	enge Integration	Selektion und Filterung von Experimenten; Filterung von Genen
ExpressDB	Web-Browser		manuelle Selektion von Experimenten; Filterung von Genen
GeneX			manuelle Selektion und Filterung von Experimenten
GIMS	proprietär (Java Applet)		vordefinierte, parametrisierte Abfragen
M-CHIPS	Web-Browser		Filterung von Genen
RAD2			manuelle Selektion von Experimenten; Filterung von Genen; vordefinierte, parametrisierte Abfragen
SMD			manuelle Selektion von Experi-
YMD			menten; Filterung von Genen

Benutzer überlassen. Somit liegt es im Kenntnisbereich des Benutzers, welche der gespeicherten Daten bereits normalisiert sind und welche Methode hierzu verwendet wurde.

Im Gegensatz dazu integrieren SMD, M-CHIPS und RAD2 verschiedene Normalisierungsmethoden, die es dem Benutzer erlauben, die in das System geladenen Expressionsdaten mit verschiedenen Strategien und Parameterwerten zu transformieren. Diese Systeme speichern sowohl die Roh- als auch die normalisierten Daten und ermöglicht damit eine erneute Normalisierung der Daten unter Nutzung einer anderen Methode oder anderen Parameterwerten.

4.3.8 Datenanalyse und Applikationsintegration

In diesem Abschnitt wird untersucht, welche Möglichkeiten die Systeme in Hinsicht auf die Datenanalyse bieten. Insbesondere wird auf die Abfrage- und Berichterstellung, Data Mining und Statistik sowie die Visualisierung eingegangen. Keines der untersuchten Systeme nutzt die OLAP-Technologie.

Abfrage- und Berichterstellung. Die Tabelle 4.7 zeigt die Funktionalitäten der einzelnen Systeme in Hinsicht auf die Möglichkeiten der Abfrage- und Berichterstellung. Die meisten Systeme bieten eine web-basierte Benut-

Tabelle 4.8: Implementierte Data Mining und statistische Methoden

Systeme	Software Tools	Integration	Methoden
ArrayDB	keine	–	keine
ExpressDB	proprietär	lose Kopplung	Clustering unter Nutzung des Pearson-Korrelations-Koeffizienten
GeneX	CyberT, Eisen, RClust	transparente Integration	hierarchisches Clustering, PCA, K-means, Bonferonni Korrektur, Bayesianische Varianzschätzung
GIMS	keine	–	keine
M-CHIPS	proprietär	enge Integration	Korrespondenzanalyse, hierarchisches Clustering
RAD2	keine	–	keine
SMD	XCluster	transparente Integration	hierarchisches Clustering, K-means, SOM, SVD
YMD	keine	–	keine

zerschnittstelle, um Abfragen zu erstellen und auszuführen. Diese Schnittstelle nutzt direkt das Datenbank-API, um kurze Antwortzeiten sicherzustellen. Ein vielfach genutzter Ansatz, dem z.B. ExpressDB, SMD, YMD und RAD2 folgen, besteht darin, dass ein Benutzer zuerst gezielt Experimente/Microarrays auswählt oder nach solchen sucht. Dazu dienen die spezifizierten experimentellen Metadaten. Im Anschluss werden relevante Gene selektiert. Hierzu kann der Benutzer unter Nutzung von Suchworten (z.B. Namen, Organismen und Krankheiten) eine Suche in der integrierten Genannotation durchführen. Darüber hinaus nutzen einige Abfragen und Berichte die Angabe von Grenzwerten oder Intervallen, um auf die relevanten Expressionsdaten zu fokussieren. Im Gegensatz zu den einfachen HTML-basierten Benutzeroberflächen anderer Systeme, verfügt ArrayDB über ein umfangreiches und integriertes Grafiktool, das weitere interaktive Operationen für benutzerspezifische Abfragen zulässt.

Data Mining und Statistik. Die Tabelle 4.8 zeigt die wichtigsten in den evaluierten Systemen implementierten Data-Mining- und statistischen Methoden. GeneX, SMD und M-CHIPS bieten ein umfangreiches Spektrum an solchen Methoden, die dem Benutzer beispielsweise erlauben, die Expressionsdaten mit verschiedenen Clustering-Methoden (z.B. hierarchisches agglomeratives Clustering und K-means) zu analysieren. M-CHIPS setzt dedizierte Analysetools ein, die spezifisch für diesen Zweck entwickelt wurden und direkt

Tabelle 4.9: Visualisierung

Systeme	Software Tools	Integration	Funktionalität
ArrayDB	ArrayViewer, MultiExperiment-Viewer	enge Integration	Spot-Maps, Intensitätsgraphen
ExpressDB	MS Excel	lose Kopplung	Cluster Map
GeneX	RClust, Eisen	transparente Integration	anklickbare Dendrogramme, Cluster-Maps
GIMS	proprietär (Java)	enge Integration	graphische Navigation für Protein-Interaktionen
M-CHIPS	proprietär	enge Integration	Diagramm zur Korrespondenzanalyse
RAD2	keine	–	–
SMD	XCluster, TreeView	transparente Integration	Spot-Maps, anklickbare Cluster-Maps
YMD	keine	–	–

auf der Datenbank operieren (enge Integration). Dagegen integrieren GeneX und SMD transparent existierende Tools für das Clustering der Daten unter einer einheitlichen web-basierten Benutzerschnittstelle. Der Benutzer selektiert beispielsweise zuerst über das vorhandene Abfragemodul eine Teilmenge an Daten, die dann Eingang in eine ausgewählte Data-Mining-Methode findet und mit dieser analysiert werden. Dabei werden die Daten zumeist aus der Datenbank in eine Datei exportiert und anschließend in das Data-Mining-Tool importiert. ExpressDB verfügt ebenso über ein Analysetool für das Clustering, das jedoch nur lose gekoppelt ist. Für eine Analyse müssen die Daten manuell aus der Datenbank exportiert, evtl. transformiert und anschließend in das Analysetool importiert werden. ArrayDB, GIMS und YMD integrieren keine Data-Mining-Methoden.

Visualisierung. Die Tabelle 4.9 zeigt nennenswerte Besonderheiten der Systeme in Hinsicht auf die Visualisierung von Expressionsdaten oder Analyseergebnissen. Typischerweise verfügen die integrierten Analysetools für das Clustering über eine eigenständige Visualisierung der erzielten Ergebnisse. In GeneX und SMD ist die Visualisierung der Ergebnisse von Cluster-Analysen interaktiv gestaltet, so dass der Benutzer direkt von den Ergebnissen zu ausgewählten Genen und ihrer Annotation navigieren kann. ExpressDB verwendet MS Excel, um die Clustering-Ergebnisse zu visualisieren. GIMS bringt eine Java-basierte Benutzerschnittstelle, das die Navigation entlang

von Protein-Protein-Interaktionen in einem Netzwerk erlaubt. Nur in ArrayDB und SMD ist eine Visualisierung der gescannten Microarrays möglich. Das erlaubt dem Benutzer die Ansicht der einzelnen Spots und damit die Verifizierung von berechneten aggregierten Intensitätswerten sowie die Ansicht relevanter Spot-bezogener Annotation (z.B. Sequenzen).

4.3.9 Vergleichende Bewertung

Die Tabelle 4.10 fasst die wichtigsten Vor- und Nachteile der evaluierten Systeme zusammen. Einige Systeme unterstützen Expressionsdaten verschiedener experimenteller Technologien; ArrayDB und SMD verwalten ausschließlich Daten von cDNA-Arrays. Die Spezifikation und Nutzung von Annotationsdaten ist bei jedem System unterschiedlich. Die experimentellen Metadaten werden zumeist durch Freitextfelder erfasst. ExpressDB stellt ein einzelnes Textfeld zur Verfügung, das eine umfassende verbale Beschreibung aufnehmen kann. Dagegen setzen andere Systeme kontrollierte Vokabulare ein. M-CHIPS nutzt hierzu lokal entwickelte Listen, die sowohl die zu annotierenden Parameter als auch deren möglichen Werte beinhalten. Dagegen verwendet RAD2 öffentliche Vokabulare. Zur Speicherung der experimentellen Metadaten werden parameterspezifische Attribute im DB-Schema verwendet. Der flexiblere EAV-Ansatz wird nur von M-CHIPS und SMD unterstützt. SMD materialisiert Genannotationen von verschiedenen öffentlichen Datenquellen, die in bestimmten Zeitabständen aktualisiert werden. GeneX, M-CHIPS, RAD2 und YMD verwenden primär Web-Links zu externen Quellen.

Alle Systeme verwenden CSV-Dateien zum Datenaustausch. Zusätzlich unterstützt GeneX einen Datenaustausch auf Basis von GeneXML. Dateien in diesem Format können Expressionsdaten und Annotationen enthalten.

Letztlich differieren die evaluierten Systeme in den integrierten Methoden zur Analyse und Visualisierung. ArrayDB verfügt über ein umfangreiches graphisches Abfragetool, dass der interaktiven Genexpressionsanalyse dient. Dagegen setzen andere Systeme eine relativ einfache web-basierte Benutzerschnittstelle ein. Nur GIMS und RAD2 bieten vordefinierte, parametrisierbare Abfragen. ArrayDB, ExpressDB, GIMS, RAD2 und YMD unterstützen kein Clustering oder statistische Methoden zur Datenanalyse. Im Gegensatz dazu bieten GeneX und SMD umfangreiche Analysemöglichkeiten, insbesondere durch spezielle Analysetools, die transparent integriert sind.

Tabelle 4.10: Wichtige Vor- und Nachteile der Systeme

Systeme	Vorteile	Nachteile
ArrayDB	⊕ umfangreiches graphische Abfragetool	⊖ Fokussierung auf cDNA-Microarray Daten ⊖ keine materialisierte Genannotationen ⊖ keine integrierte Cluster- und statistische Analyse
ExpressDB	–	⊖ Freitextfeld und damit limitierte Analyse experimenteller Metadaten ⊖ keine integrierte Datenanalyse
GeneX	⊕ transparente Integration von Analysefunktionalitäten (Clustering & Statistik) ⊕ XML (GeneML) als Austauschformat	⊖ keine materialisierte Genannotationen
GIMS	⊕ umfassende Bibliothek von parametrisierten Abfragen	⊖ keine integrierte Cluster- und statistische Analyse
M-CHIPS	⊕ Eingang lokal definierter und kontrollierter Vokabulare für die Spezifikation experimenteller Annotationsdaten ⊕ flexible (EAV) Verwaltung der experimentellen Metadaten	⊖ keine materialisierte Genannotationen
RAD2	⊕ Integration verschiedener öffentlicher Vokabulare für die Erfassung von experimentellen Metadaten ⊕ parametrisierte Abfragen	⊖ keine materialisierte Genannotationen
SMD	⊕ transparente Integration der Clusteranalyse ⊕ materialisierte Integration von Genannotationen mit automatischer Aktualisierung	⊖ Fokussierung auf cDNA-Microarray Daten
YMD	–	⊖ keine integrierte Cluster- und statistische Analyse ⊖ keine materialisierte Genannotationen

4.4 Zusammenfassung

Gegenstand dieses Kapitels waren die Anforderungen an die Verwaltung und Analyse von Microarray-Daten. Dazu wurden verschiedene Datenarten vorgestellt sowie Anforderungen, die sich aus der Datenverwaltung, -integration und -analyse ergeben. Im Anschluss wurden die identifizierten Anforderungen für eine Evaluierung von acht bestehenden Systemen verwendet, die zur Zeit der Evaluierung (2003) mit mindestens einer wissenschaftlichen Publikation beschrieben und öffentlich zugänglich waren. Basierend auf der durchgeführten Evaluierung der Systeme waren die folgenden wichtigen Problembereiche zu identifizieren [DKR03]:

- **Datenintegration**: Eine Datenintegration auf Basis von Web-Links, wie sie von den meisten der untersuchten Systeme verfolgt wird, reicht für eine effiziente Unterstützung der Genexpressionsanalyse nicht aus. Fortgeschrittene Ansätze, beispielsweise durch den Einsatz von förderierte Datenbanken, Mediatoren und die materialisierte Integration, versprechen eine umfassendere Analyse der Microarray-Daten. Ihre Implementierung birgt signifikante Herausforderungen in Hinsicht auf die Schema- und Datenintegration, das Schema-Matching und Data-Cleaning. Obwohl Datenintegrationslösungen anderer Domänen hierzu Beiträge liefern können, muss auf die Charakteristik der Bioinformatik eingegangen werden, die beispielsweise in den begrenzten Anfragemöglichkeiten der öffentlichen Quellen, den überlappenden Datenquellen und der Nutzung verschiedener Vokabulare besteht (vgl. Kapitel 1).

- **Datenanalyse**: Die untersuchten Systeme verfügen über einfache Analyseansätze, die die Annotationsdaten nur unwesentlich mit einbeziehen. Die Multidimensionalität der Expressionsdaten (vgl. Kapitel 3) und die hierarchische Struktur der (Annotations-) Dimensionen werden nur unzureichend ausgenutzt; die OLAP-Technologie kommt nicht zur Anwendung

- **Optimierung der Performance**: Um eine hohe Performanz der Analysen zu gewährleisten, die die großen Mengen von Expressionsdaten verarbeiten, ist der Einsatz fortgeschrittener Datenbanktechniken notwendig. Insbesondere für die interaktiven Analysen sind solche Techniken anzuwenden, um eine kurze Antwortzeit zu erreichen.

Kapitel 5
Die Datenintegrations- und Analyseplattform GeWare

5.1 Motivation

Um den Limitierungen der vorhandenen Ansätze und Systeme zu begegnen und gleichzeitig den Anforderungen von Genexpressionsanalysen in verschiedenen Forschungskooperationen gerecht zu werden, wurde eine Integrations- und Analyseplattform mit dem Namen *Genetic Data Warehouse* (*GeWare*[27]) konzipiert und entwickelt. Die wichtigsten Aspekte dieser Plattform sind die Folgenden.

- *GeWare* folgt dem Data-Warehouse-Ansatz [JLVV03], um zentralisiert alle relevanten Daten zu integrieren. Zu diesen Daten zählen die experimentellen Rohdaten, die normalisierten Daten und die Analyseergebnisse aber auch die experimentellen Metadaten, die das Experiment durch geeignete Parameter dokumentieren. Der Data-Warehouse-Ansatz verspricht signifikante Vorteile, weil alle Daten direkt und unmittelbar für Analysen zugänglich sind und somit eine gute Performanz

[27] Das Akronym hat sich auf Grund der Aufnahme weiterer experimenteller Daten vom *Gene Expression Warehouse zum oben angegebenen Namen gewandelt.*

erlangt werden kann. *GeWare* verwendet ein multidimensionales Schema, mit dem die voluminösen experimentellen Daten von verschiedenen Perspektiven und Bedingungen analysiert werden können.

- *GeWare* integriert sowohl öffentlich verfügbare Daten als auch experimentelle Metadaten und klinische Daten. Diese Daten werden intern in einem generischen Format verwaltet, um dem hohen Grad an Heterogenität zu begegnen und offen für neue Arten von Annotationen zu sein. Die experimentellen Metadaten können manuell durch den Benutzer spezifiziert werden, während klinische Daten aus einem Studienverwaltungssystem importiert werden. Öffentlich verfügbare Daten werden unter Nutzung eines Mediators [KDKR05] (vergl. Kapitel 8) integriert.

- *GeWare* verfügt über verschiedene Methoden, um Microarray-basierte Daten zu normalisieren bzw. einer Vorverarbeitung zu unterziehen und zu analysieren. Diese Analysemethoden können auf einfache Art und Weise miteinander gekoppelt werden, nämlich durch den Austausch von Treatment- und Gengruppen sowie Expressionsmatrizen. Somit sind die Ergebnisse einer Analyse unmittelbar als Eingabe einer folgenden Analyse verwendbar.

Das *GeWare* System wurde in verschiedenen Leipziger Forschungsprojekten angewendet. Dazu zählen beispielsweise Genexpressionsanalysen für verschiedene Arten von Schilddrüsenknoten [EKB+05] (Dr. Eszlinger/Dr. Krohn), für CD97 überexprimierte Tumorzellen (Prof. Aust) und für Fibroblastenkulturen (Dr. Anderegg/Dr. Saalbach). Daneben fand das System zur Untersuchungen molekularer Mechanismen der Herz-Kreislauf-Physiologie bei Mäusen (Dr. Briest/Dr. Deten) Anwendung. Darüber hinaus wird die Plattform in zwei deutschlandweiten klinischen Studien (vgl. Kapitel 7) zur Untersuchung der molekularen Mechanismen von malignen Lymphomen und Gliomen verwendet, für die erste Ergebnisse in [HBB+06] publiziert wurden. Für diese und weitere Projekte verwaltet *GeWare* derzeit die Daten auf Basis von mehr als 2.100 Microarrays.

5.2 Systemarchitektur

Die Abbildung 5.1 zeigt die Systemarchitektur der *GeWare*-Plattform im Überblick. Die Daten werden von verschiedenen Quellen importiert und in einer so genannten *Staging Area* transformiert, bevor sie in die zentrale Data-Warehouse-Datenbank geladen und dort gespeichert werden. Mit Hilfe von Data Marts [JLVV03] kann auf spezielle Analyseanforderungen eingegangen

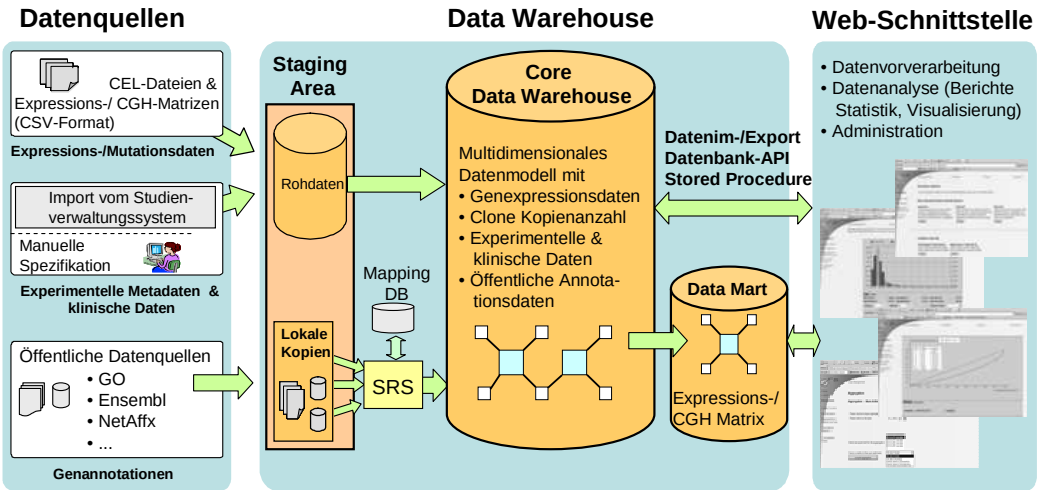


Abbildung 5.1: *GeWare* Systemarchitektur im Überblick

werden. Sie enthalten entweder abgeleitete Daten auf Basis der Daten des Data Warehouse oder werden benutzt, um Resultate ausgewählter Analysemethoden für die spätere Visualisierung oder Wiederverwendung in weiteren Analysen zu speichern. Alle administrativen und Analysefunktionen von *GeWare* sind über die Web-Schnittstelle mit einem Browser zugreifbar.

Das Data Warehouse *GeWare* integriert die folgenden Daten:

- **Expressionsdaten**. Unter dem Einfluss lokaler Anforderungen fokussiert *GeWare* gegenwärtig auf Expressionsdaten, die durch Microarrays der Fa. Affymetrix produziert werden. Diese Oligonukleotid-Microarrays (vgl. Kapitel 3) verwenden kurze Sequenzen (Oligos) mit einer Länge von 25 Nukleotiden. Ein Gen wird durch ein so genanntes Probeset repräsentiert, das sich aus 11 - 20 Oligos zusammensetzt. Mit einem Microarray wird die Expression von Millionen von Oligos gemessen, von denen die Probeset-Intensitäten abgeleitet werden. *GeWare* bietet Import-Schnittstellen für die Oligo- und Probeset-Expressionsdaten. Die Oligo-Intensitäten werden aus den so genannten CEL-Dateien importiert, während die Probeset-Intensitäten in CSV-Dateien in Form von Expressionsmatrizen vorliegen können. Obwohl die Oligo-Intensitäten weitaus voluminöser sind, erlauben sie die Anwendung von verschiedenen Methoden zur Vorverarbeitung, z.B. zur Normalisierung und Hintergrundbereinigung sowie zur Aggregation und damit zur Ableitung von Probeset-Intensitäten. Im Gegensatz dazu liegen die importierten Probeset-Intensitäten bereits präprozessiert vor.

- **Mutationsdaten**. Mutationsdaten beschreiben die genetische Diver-

sität. Gewöhnlich sind Gene an festen Positionen auf einem Chromosom lokalisiert und verwenden eine spezielle Sequenz. Jedoch können individuelle Mutationen (z.B. Einschübe, Löschungen, Ersetzungen und Verschiebungen) von Sequenzregionen einen signifikanten Einfluss auf die Genexpression und damit auf die assoziierten Genfunktionen haben. Das betrifft insbesondere große, blockweise Mutationen, z.B. Kopien und Verschiebungen langer Sequenzregionen zu anderen Chromosomen. Die Matrix-CGH-Arrays (Matrix-based comparative genomics hybridization) [KKS+92, SLS+97] ist eine gegenwärtig genutzte Technologie, um solche Mutationen zu messen (vgl. auch Abschnitt 3.3). Auf diesen Chips sind Sequenzen (Clone) aufgebracht, die jeweils eine ausgewählte Sequenzregion eines Chromosoms repräsentieren, für die die Häufigkeit ihres Auftretens im Genom (Kopienanzahl) gemessen wird. Clone mit einer hohen (mit keiner = 0) Kopienanzahl verweisen auf rekurrente (nicht verfügbare bzw. stark mutierte) Sequenzenregionen, so dass hohe (niedrige) Expressionswerte von Genen innerhalb dieser Regionen erklärt werden können. Ähnlich wie die Affymetrix Microarrays zur Genexpressionsanalyse generieren die Matrix-CGH-Arrays ein großes Datenvolumen. *GeWare* besitzt Schnittstellen, um die Daten solcher Chips zu importieren.

- **Experimentelle Metadaten und klinische Daten**. Diese Daten werden typischerweise von den Benutzern unter Nutzung der Web-Schnittstelle spezifiziert. Dazu dient ein zuvor definiertes *Annotation Template*. Auf Basis dieses Templates generiert *GeWare* automatisch die für die Spezifikation notwendigen Webseiten, die dem Benutzer einerseits erlauben, verbale Eingaben zu tätigen und andererseits mögliche Werte aus assoziierten kontrollierten Vokabularen auszuwählen. Relevante Mengen von klinischen Daten werden automatisch aus einem Studienverwaltungssystem importiert, das die patientenbezogenen Daten bereits in anonymisierter Form enthält.

- **Genannotationen**. *GeWare* integriert Daten aus verschiedenen öffentlich zugänglichen Quellen, wie z.B. NetAffx [LLS+03], GeneOntology [HCI+04], Ensembl [BAB+04] und LocusLink [PM01]. Dazu wird ein hybrider Integrationsansatz verwendet, der in Kapitel 8 näher vorgestellt wird.

Die importierten Daten werden zuerst einer Transformation unterzogen, bevor sie im Data Warehouse integriert gespeichert werden. Für die importierten Microarray-Rohdaten stehen verschiedene Normalisierung- und Aggregationsmethoden zur Verfügung, die die Rohdaten transformieren. Die

Verfügbarkeit mehrerer solcher Methoden stellt die notwendige Flexibilität in der Vorverarbeitung bereit, da sich noch keine allgemein akzeptierte Methode zur Vorverarbeitung durchgesetzt hat. Die vom Benutzer spezifizierten experimentellen Metadaten werden in einer generischen Struktur gespeichert (vgl. Abschnitt 5.5), das die Unabhängigkeit von verschiedenen, projektspezifischen Anforderungen sicherstellt.

Die zentrale Komponente von *GeWare*, das Data Warehouse, verwendet ein multidimensionales Datenmodell, auf das in Abschnitt 5.4 eingegangen wird. Für dessen Implementierung nutzt *GeWare* das relationale Datenbanksystem DB2 von IBM (auf einem High-end Linux Server), das die Verwaltung eines hohen Datenvolumens unterstützt und eine Vielzahl an Performanz-Tuning-Optionen zur Verfügung stellt, wie beispielsweise Indizierung, materialisierte Sichten und Partitionierung der Daten. Spezielle Datenmengen werden redundant in Data Marts gespeichert, die spezifisch auf eine hohe Performanz oder spezielle Analyseanforderungen ausgelegt sind. Zum Beispiel können Expressionsmatrizen für relevante Gen- und Treatment-Gruppen extrahiert und in einem Data Mart gespeichert werden, so dass die Werte schnell wiederverwendet und visualisiert werden können, ohne dass eine Neuberechnung notwendig ist.

Die Web-Schnittstelle von *GeWare* wurde unter Nutzung der Java Servlet Technologie implementiert. Damit werden ressourcenintensive Analyseaufgaben zentral auf dem Server durchgeführt. Zudem sind alle Funktionalitäten, einschließlich der Administration, des Datenim- und -exports, der Erfassung und Verwaltung der experimentellen Metadaten sowie der Expressionsanalysen, über eine einheitliche Schnittstelle, der Web-Schnittstelle, mit Hilfe eines Web-Browsers erreichbar. Der Zugriff auf *GeWare* ist nur authentifizierten Benutzern auf der Basis eines Benutzer-/Benutzergruppenkonzeptes möglich. Diesem Konzept folgend stellen die Benutzer einer Gruppe ihre Daten den anderen Benutzern der Gruppe frei zur Verfügung. Benutzer anderer Gruppen haben keinen Zugriff auf diese Daten. Das ermöglicht eine projektorientierte Forschungsarbeit. Neben dem Datenzugriff, reglementiert das Benutzergruppenkonzept den Zugriff auch auf spezielle Funktionen der Analyseplattform, die beispielsweise im Datenimport und -export aber auch in ausgewählten Analysefunktionen bestehen können. Die Webseiten des *GeWare*-Systems werden automatisch und in Abhängigkeit der für die verwendete Benutzergruppe definierten Rechte generiert und somit der Zugriff auf erlaubte Funktionalitäten und Daten sichergestellt.

5.3 System Workflows

Im Folgenden wird auf die Workflows der *GeWare* -Plattform eingegangen, die sich aus den Import- und den Analyseprozessen zusammensetzen.

5.3.1 Importprozesse

Die Abbildung 5.2a zeigt die von *GeWare* unterstützten Importprozesse. Grundlage für jeglichen Datenimport ist ein Experiment, das im System angelegt ist. Ein Experiment umfasst eine Menge von Microarrays, denen die Expressionsdaten in Form der Rohdaten (Oligo-Intensitäten), der vorverarbeiteten Daten (Probeset-Intensitäten) und Analyseergebnisse zugeordnet sind. Die Plattform verfügt über spezielle Import-Schnittstellen. Damit können einerseits die Microarray-Rohdaten aus den voluminösen CEL-Dateien geladen und in einem anschließenden Schritt einer Vorverarbeitung unterzogen werden. Andererseits bietet die Plattform Schnittstellen, um die mit Hilfe externer Routinen und Programmen vorverarbeiteten Probeset-Intensitäten und Matrix-CGH Daten zu importieren.

Unabhängig von den Importprozessen können die experimentellen Metadaten durch den Benutzer spezifiziert werden. Dazu generiert *GeWare* automatisch Webseiten auf Basis des gewählten *Annotation Templates* (vgl. Abschnitt 5.5).

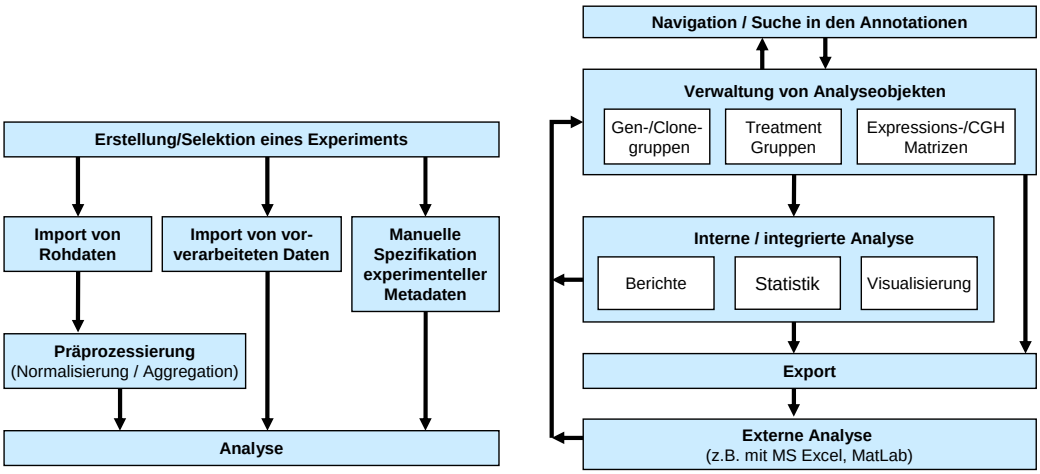


Abbildung 5.2: Abstrakte Import- und Analyseprozesse in *GeWare*

5.3.2 Analyseprozesse

Die Abbildung 5.2b zeigt die von der *GeWare*-Plattform unterstützten Analyseprozesse. Die flexible Kombination und Integration von verschiedenen Analysemethoden und Visualisierungen wird durch die einheitliche Verwendung von Gen-/Clone- und Treatment-Gruppen sowie Expressions-/CGH-Matrizen erreicht. Diese drei generischen Objekt-Container werden zentral von der Plattform verwaltet. Die Gen-/Clone- und Treatment-Gruppen entstehen a) durch die manuelle Auswahl und Spezifikation eines Benutzers von relevanten Genen/Clonen und Microarrays, sind b) Ergebnis von Analysen öffentlich verfügbarer Genannotationen sowie der experimentellen Metadaten und resultieren c) aus einer Analyse von Expressionsdaten.

Eine fokussierte Analyse von relevanten Expressionsdaten (Mutationsdaten) wird insbesondere durch Expressionsmatrizen (CGH-Matrizen) gewährleistet, die jeweils von einer bestehenden Gen- (Clone-) und Treatment-Gruppe determiniert werden. Die Gen-/Clone- und Treatment-Gruppen sowie die Expressions-/CGH-Matrizen werden für vielfältige Visualisierungen, statistische Berichte, Signifikanztests und Analysen verwendet. Das Ergebnis dieser Analysen, das typischerweise in einer Teilmenge der in sie eingehenden Gene/Clone besteht, kann wiederum als Gruppe gespeichert und in späteren Analysen und Visualisierungen wiederverwendet werden.

Für die Durchführung von Analysen mit externen Analyseroutinen bietet *GeWare* einen Export der Rohdaten, der vorverarbeiteten Daten und Analyseresultate und erlaubt somit eine Analyse mit einer vom Benutzer präferierten Software (z.B. MS Excel, R, MatLab). Die Ergebnisse solcher Analysen können im Anschluss in Form von Gen-/Clone- und Treatment-Gruppen sowie durch die Definition von Expressions-/CGH-Matrizen in die Plattform zurück fließen.

5.4 Data Warehouse Schema

GeWare benutzt zur Speicherung und Analyse der Expressions- und Mutationsdaten ein multidimensionales Schema, das in Abbildung 5.3 in abstrakter Form dargestellt ist. Dem multidimensionalen Modellierungsparadigma folgend besteht das Schema aus Dimensionen und Fakten. Während Fakten numerischen und additiven Charakter aufweisen, beschreiben die Dimensionen die Semantik der Fakten.

Gegenwärtig enthält das Schema die zwei Faktentabellen Probeset-Intensitäten und Clone-Intensitäten, die die Genexpressionsdaten und die Mutationsdaten (Kopienanzahl) der Clone repräsentieren. Die Fakten-

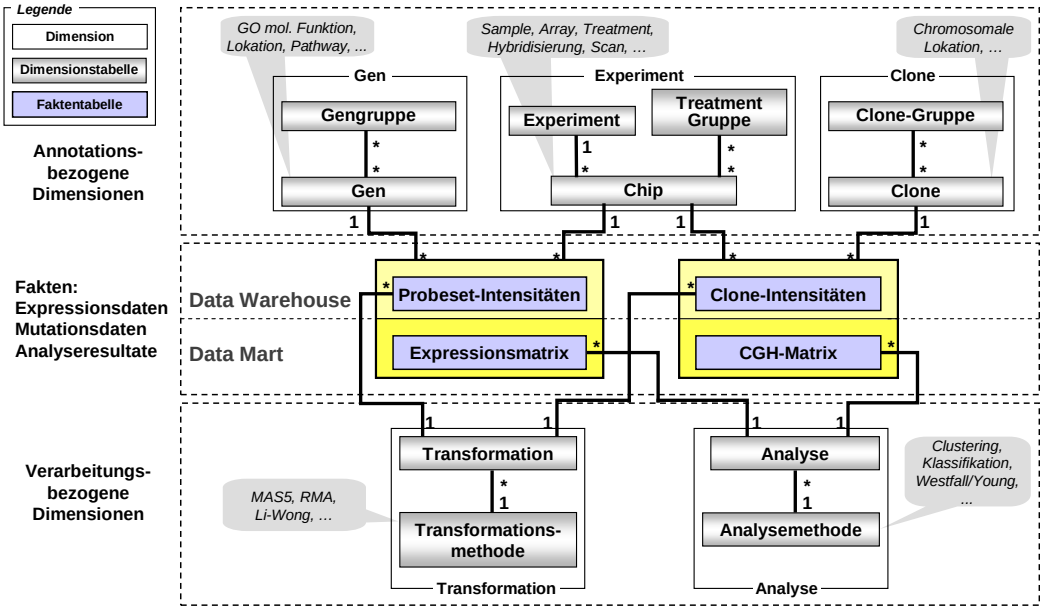


Abbildung 5.3: Multidimensionales Data Warehouse Schema im Überblick

tabelle Probeset-Intensitäten speichert sowohl die importierten, bereits vorverarbeiteten Expressionsdaten als auch die Probeset-Intensitäten, die anhand der integrierten Transformationsmethoden aus den CEL-Dateien im Schritt der Vorverarbeitung berechnet werden. Zusätzliche Faktentabellen existieren für Data Marts, wie z.B. den Expressionsmatrizen (CGH-Matrizen), die die Daten für ausgewählte Gene (Clone) und Microarrays (Matrix-CGH-Arrays) nach Anwendung von verschiedenen Analysemethoden speichern.

Die Dimensionen teilen sich in die annotations- und die verarbeitungsbezogenen Dimensionen, die in Abbildung 5.3 zusammen mit illustrierenden Beispielen dargestellt sind. Zu den annotationsbezogenen Dimensionen gehören *Experiment*, *Gen* und *Clone*, während die Dimensionen *Transformation* und *Analyse* zu den verarbeitungsbezogenen Dimensionen zählen. Die Experiment-Dimension beschreibt namentlich die Microarrays (Tabelle Chip), ihre Zugehörigkeit zu den Experimenten (Tabelle Experiment) sowie ihre Verwendung in den vom Benutzer spezifizierten oder aus Analysen resultierenden Treatment-Gruppen (Tabelle Treatment-Gruppe). Ferner sind die experimentellen Metadaten ein weiterer Bestandteil der Dimension, die die Experimente und die einzelnen Chips weiter beschreiben. Die Gen-Dimension enthält die auf einem Microarray durch Oligonukleotid-Sequenzen repräsentierten Probesets, die in Gengruppen zusammengefasst werden können. Äquivalent dazu beinhaltet die Clone-Dimension Clone von Matrix-

CGH-Arrays sowie ihre Gruppierungen.

Die verarbeitungsbezogenen Dimensionen enthalten namentlich spezielle Berechnungsmethoden (Tabelle TRANSFORMATIONSMETHODE), die der Transformation von Oligo-Intensitäten in Probeset-Intensitäten dienen sowie Methoden (Tabelle ANALYSEMETHODE), die bei der Analyse der Expressions- und Mutationsdaten eingesetzt werden. Bei jeder Ausführung einer dieser Methoden werden zusätzliche Metadaten erfasst, die die Parameter und deren vom Benutzer spezifizierten Werte umfassen. Auf diese Art und Weise wird sichergestellt, dass der Berechnungsprozess dokumentiert ist und damit die erzielten Ergebnisse für die Benutzer nachvollziehbar und reproduzierbar sind.

Typischerweise sind die Dimensionen hierarchisch organisiert. Die Hierarchien erlauben eine Generalisierung und Spezialisierung und ermöglichen damit verschiedene Ebenen der Abstraktion in der Analyse. Beispielsweise können die Probesets in der Gen-Dimension unter Verwendung der GeneOntology (GO) [HCI+04] nicht nur nach den zugeordneten molekularen Funktionen, biologischen Prozessen und zellulären Komponenten gruppiert werden. Vielmehr wird eine Expressionsanalyse unter Verwendung der GO-Hierarchien auf unterschiedlichen funktionalen Ebenen möglich. Darüber hinaus ermöglicht die Navigation in den Hierarchien, die aus dem Data Warehousing und dem OLAP in anderen Domänen bekannt ist [JLVV03], eine flexible Analyse der Expressionsdaten in Bezug auf die unterschiedlichen Detaillierungsstufen.

Ebenso wie die hierarchische Organisation der Dimensionen unterstützt die konzipierte Multidimensionalität des Data Warehouse Schemas eine flexible Genexpressionsanalyse. Während gegenwärtige Ansätze typischerweise eine komplette Expressionsmatrix verwenden, die die Expressionswerte aller auf einem Microarray repräsentierten Gene und aller Microarrays eines Experiments beinhalten, kann mit dem Schema auf individuelle und vergleichende Analysen unter Verwendung einer frei wählbaren Teilmenge an Expressionswerten fokussiert werden. Diese Menge an Expressionswerten kann anhand ausgewählter Werte einer speziellen Hierachieebene einer einzelnen Dimension ebenso bestimmt werden wie durch eine Kombination von Werten verschiedener Dimensionen.

Auch in Hinsicht auf Erweiterbarkeit und Skalierbarkeit besitzt das multidimensionale Schema Vorteile. Innerhalb jeder Dimension können neue Werte, wie z.B. neuartige Analysemethoden in der Dimension *Analyse* und Probesets (Gene) in der Dimension *Gen*, hinzugefügt werden, ohne dass das Änderungen an dem zugrunde liegende Data-Warehouse-Schema nach sich zieht. Ebenso ist das Schema um neue Datenbereiche erweiterbar, wie sie beispielsweise von den neuartigen Exon- und Tiling-Microarrays produziert

werden. Für die dazu notwendigen neu zu konzipierenden und zu implementierenden Faktenentitäten und evtl. Data Marts kann auf bereits bestehenden Dimensionen (z.B. Experiment, Analyse) zurückgegriffen werden.

5.5 Integration experimenteller Metadaten

5.5.1 Problemstellung

In Abhängigkeit vom biologischen Fokus können Microarray-Experimente auf unterschiedliche Art und Weise durchgeführt und dokumentiert werden. Beispielsweise ist die Spezifikation von Zeitpunkten in einem Zeitserien-Experiment notwendig, während sie für Vergleiche von molekularen Eigenschaften zwischen gesundem und krankem Gewebe nicht relevant sind. Ebenso erfordert die Untersuchung von Gewebe aus unterschiedlichen Organen eine differenzierte Annotation, die beispielsweise in der Gehirnregion bei Gliomen und der Knotenart und deren Größe bei Schilddrüsen besteht (vgl. Kapitel 7).

Die MIAME-Richtlinie[28] [BHQ+01] adressiert diese Problematik und gibt eine Empfehlung, welche Daten notwendig sind, um ein Microarray-Experiment zumindest in seinen Grundzügen zu beschreiben. Da die MIAME-Richtlinie sehr allgemein ist, wurde das Objektmodell MAGE-OM[29] entworfen, dass eine komplexe Klassenstruktur zur Beschreibung von Microarray-Experimenten besitzt. Die Klassen enthalten verschiedene Attribute, für die der Benutzer im Annotationsprozess Werte spezifizieren kann. Jedoch lassen MIAME und MAGE offen, welche Werte für die Annotation verwendet werden sollten. Das führt zu Inkonsistenzen in der Beschreibung eines Microarray-Experiments, wenn verschiedene Werte (z.B. "Homo sapiens", "H. sapiens", "Hsa", "Human" zur Angabe des Organismus) verwendet werden. Dem begegnet die MGED-Ontologie [WPC+06], die vordefinierte Begriffe zur Beschreibung eines Microarray-Experiments enthält. Die MGED-Ontologie wird fortwährend weiterentwickelt und ist in verschiedenen Versionen verfügbar.

Laborinformationssysteme (LIMS) und verschiedene Applikationen, z.B. MIAMExpress, erlauben dem Benutzer auf Basis des MAGE-OM und der MGED-Ontologie, eine Beschreibung von Microarray-basierten Experimenten vorzunehmen. Jedoch sind sie oftmals in ihrer Flexibilität begrenzt, sowohl neue Klassen und Attribute als auch Konzepte der MGED-Ontologie

[28] MIAME steht für *Minimal Information about a Microarray Experiment.*

[29] MAGE-OM steht für *Microarray Gene Expression Object Model* und wurde unter Nutzung von UML entwickelt.

hinzuzufügen, z.B. um den lokalen Anforderungen bei der Annotation zu begegnen. Solche Änderungen sind meist mit einer Aktualisierung der Applikation verbunden. Zudem dienen sie primär der Datenerfassung; eine Integration, die eine Analyse zusammen mit experimentellen Daten ermöglicht, unterbleibt bzw. ist nicht ihr Ziel.

5.5.2 Konzept

Die Vielzahl an unterschiedlichen experimentellen Designs sowie klinischer Parameter macht die Verwendung einer festen Schemastruktur, die direkt auf die aufzunehmenden Daten Bezug nimmt, unpraktisch. Vielmehr wird eine generische Repräsentation benötigt, so dass die verschiedenen experimentellen Metadaten und klinischen Daten einheitlich verwaltet werden. *GeWare* verwendet so genannte *Annotation Templates*, um die einheitliche Beschreibung eines Microarray-Experiments zu erreichen, und kontrollierte Vokabulare, mit denen mögliche Annotationswerte festgelegt werden.

Definition (kontrolliertes Vokabular)**.** *Unter einem kontrollierten Vokabular V wird in GeWare eine Sammlung (Menge) von Begriffen verstanden, die dem Benutzer beispielsweise zur Spezifikation von Werten bei der Erfassung der experimentellen Metadaten zur Verfügung stehen.*

Definition (Annotation Template)**.** *Ein Annotation Template ist ein Graph, der aus einem Kategoriebaum T_K, einer Menge von Vokabularen $\{V\}$ und einer Relation m_{T_K-V} besteht. Ein Kategoriebaum $T_K = (S, K, E)$ ist ein Graph in Form eines Baumes, dessen Knoten mit gerichteten Kanten $e, e' \in E$ der Form $e = (s, s)$ und $e' = (s, k)$ $(s \in S,\ k \in K)$ verbunden sind. Innere Knoten sind Seiten $s \in S$, auf denen Kategorien $k \in K$ (Blätter) angeordnet sind. Die Tiefe des Baumes ist ebenso wie seine Gestalt variabel, mindestens aber 3; er besteht mindestens aus einer automatisch generierten Index-Seite (Wurzelknoten) und einer Seite, die eine Kategorie enthält. Mit der Relation m_{T_K-V} werden den Kategorien $k \in K$ verfügbare Vokabulare V zugewiesen.*

Mit der Definition eines *Annotation Templates* werden Seiten, ihre hierarchische Einordnung sowie die auf ihnen enthaltenen Kategorien konstruiert. Kategorien besitzen spezielle Eigenschaften, die beispielsweise in ihrem Eingabetyp, ihrer Eingabeform und der Notwendigkeit der Eingabe (obligatorische vs. optionale Kategorien) bestehen. Eine Kategorie hat entweder den Eingabetyp Freitext oder ist mit einem kontrollierten Vokabular assoziiert. Das kontrollierte Vokabular umfasst den Wertevorrat, aus dem die zu selektierenden Werte im Annotationsprozess stammen. Die Eingabeform legt

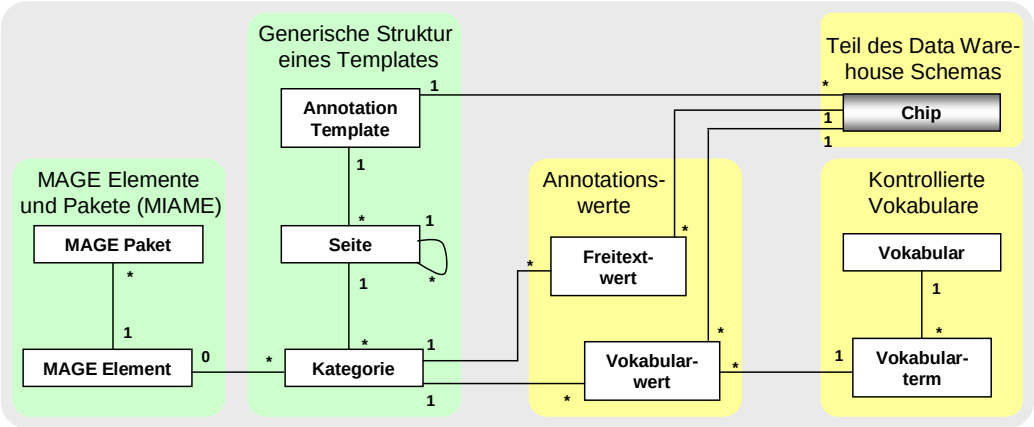


Abbildung 5.4: Generisches Schema zur Verwaltung der experimentellen Metadaten

das Erscheinungsbild auf der Web-Oberfläche (z.B. Checkboxen, Radioboxen, Auswahllisten) fest.

Das Kopieren eines bereits bestehenden *Annotation Templates* und dessen anschließende Modifikation erlaubt es, in kurzer Zeit ähnliche Templates zu konstruieren, was gerade bei umfangreichen Templates von großem Nutzen ist. Das kann notwendig werden, um projektspezifischen Anforderungen Rechnung zu tragen. *GeWare* verwendet sowohl lokale, von den Benutzern definierte kontrollierte Vokabulare als auch öffentlich verfügbare Ontologien. Zu Letzteren gehören die *NCBI Taxonomy* zur Angabe des Organismus und Teile der MGED-Ontologie. Einmal definiert oder importiert, kann ein kontrolliertes Vokabular in verschiedenen *Annotation Templates* Verwendung finden.

5.5.3 Management experimenteller Metadaten

Der wechselnde biologische Fokus, auf den ein Microarray-Experiment ausgerichtet sein kann, sowie die davon hervorgerufenen inhaltlichen und strukturellen Änderungen bedingen eine flexible Speicherung der Kategorien und der durch den Benutzer spezifizierten Annotationswerte. Die Abbildung 5.4 zeigt das Schema, das *GeWare* für die Verwaltung der experimentellen Metadaten verwendet. Ein *Annotation Template* wird in der Tabelle ANNOTATIONTEMPLATE beschrieben und besteht aus einer oder mehreren Seiten (Tabelle SEITE), die hierarchisch organisiert sein können. Auf jeder Seite können verschiedene Kategorien (Tabelle KATEGORIE) platziert werden, zu denen im Annotationsprozess entweder Freitexteingaben (Tabelle FREITEXTWERT)

oder Terme (Tabelle VOKABULARWERT) aus einem zugeordneten Vokabular (Tabellen VOKABULAR und VOKABULARTERM) spezifiziert werden.

Für einen Datenaustausch mit anderen Laboren kann auf das MAGE-OM zurückgegriffen werden, für das die auf XML basierende Sprache MAGE-ML existiert. In MAGE-ML werden die MAGE-OM Klassen als XML-Elemente (Tabelle MAGEELEMENT) repräsentiert, die zu so genannten Paketen (Tabelle MAGEPAKET) zusammengefasst werden. Eine Zuordnung der MAGE-ML Elemente zu den Kategorien eines *Annotation Templates* stellt dessen MIAME-Kompatibilität sicher und trägt zu einem Datenaustausch via MAGE-ML bei.

Die Konsistenz der experimentellen Annotation wird mit dem gezielten Einsatz eines *Annotation Template* für alle Microarrays eines Experiments und den assoziierten Vokabularen sichergestellt. Regelbasierte Eingabeüberprüfungen, wie sie beispielsweise auf der Ebene des DBMS (z.B. spezielle Constraints) und der Applikation implementiert werden können, werden nicht verfolgt, um den Aufwand zur Erstellung eines Template bewusst niedrig zu halten.

5.5.4 Spezifikation und Analyse experimenteller Metadaten

Ausgehend von einem definierten *Annotation Templates* generiert *GeWare* automatisch die Webseiten für die Benutzereingabe der experimentellen Metadaten. Der Benutzer navigiert durch die generierten Webseiten und spezifiziert dabei diese Metadaten, d.h. die Annotationswerte je Kategorie. Kategorien, deren Werte für alle Microarrays (Chips) eines Experiments gelten, können beim Annotieren des ersten Microarrays im Experiment bestimmt werden. Hiernach werden die Werte automatisch auf die anderen Microarrays übertragen. Ebenso lassen sich alle experimentellen Metadaten eines Microarrays auf einen anderen kopieren. Das ist gerade bei umfangreichen *Annotation Templates* hilfreich, wenn die Annotationswerte nur für wenige Kategorien differieren. Die Annotationswerte sind jederzeit änderbar. Eine Übersicht zeigt dem Benutzer, zu wie viel der vorhandenen Kategorien pro Seite bereits Annotationswerte spezifiziert wurden.

Die Abbildung 5.5a zeigt den Index aller Seiten für das existierende *Annotation Template* "Human Cell Culture". Damit hat der Benutzer die Möglichkeit, sich nicht nur sequentiell durch die Annotation zu bewegen, sondern kann direkt zu einer relevanten Seite navigieren.

Die Seite "Culture Conditions" aus dem oben genannten *Annotation Template* wird in Abbildung 5.5b gezeigt, auf der der Benutzer Annota-

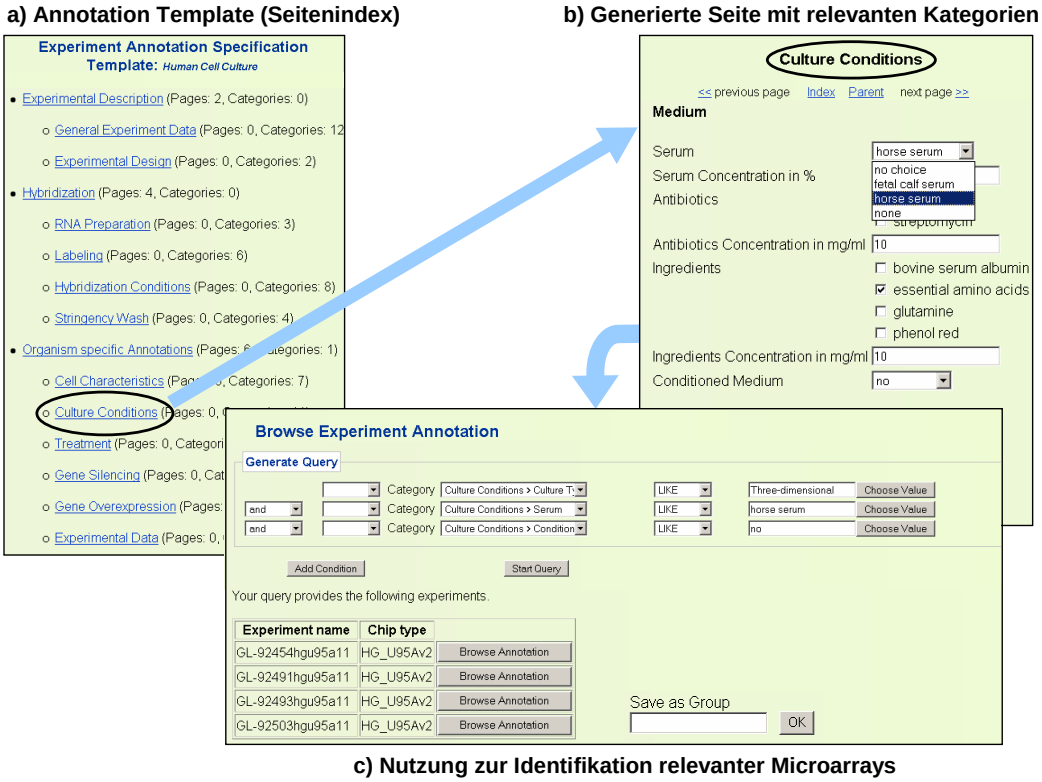


Abbildung 5.5: Spezifikation und Analyse der experimentellen Metadaten

tionswerte zu speziellen Kulturbedingungen in einem in-vitro Experiment spezifizieren kann. Die Werte derjenigen Kategorien, zu denen in der Definitionsphase des Templates ein kontrolliertes Vokabular zugeordnet wurde, stehen je nach der definierten Eingabeform in Checkboxen, Auswahlboxen etc. zur Verfügung.

Mit der Nutzung eines einzigen *Annotation Templates* wird eine einheitliche Beschreibung der Microarrays eines Experiments (und darüber hinaus evtl. mehrerer Experimente) unterstützt. Relevante Microarrays werden identifiziert, in dem die erfassten experimentellen Metadaten zur Suche verwendet werden. Wie die Abbildung 5.5c zeigt, bietet *GeWare* die Möglichkeit, mehrere Bedingungen zu spezifizieren, die mit den logischen Operatoren AND, OR und NOT beliebig kombiniert werden können. Jede Bedingung besteht dabei aus einer Kategorie, deren Werte mit einem ausgewählten Vergleichsoperator (Like, $<$, $>$) und einem spezifizierten Vergleichswert verglichen werden. Der Vergleichswert ist das Resultat einer Freitext-Eingabe des Benutzers; alternativ kann der Benutzer einen Wert aus dem Wertebereich aller zu dieser Kategorie annotierten Werte auswählen. Schließlich besteht die Möglichkeit,

die identifizierten Microarrays in einer Treatment-Gruppe zusammenzufassen, die in einer anschließenden bzw. späteren Analyse verwendet wird.

Spezielle Analysen, Berichte und Visualisierungen können ebenso auf die spezifizierten experimentellen Annotationswerte zurückgreifen, so dass eine kombinierte Auswertung von Expressions-/Mutationsdaten zusammen mit den experimentellen Annotationswerten ermöglicht wird. In Kapitel 7 (vgl. Abbildung 7.4) wird hierzu ein Beispiel aufgezeigt.

5.6 Microarray-basierte Genexpressionsanalyse

Im Folgenden wird eine Auswahl von verschiedenen Analysemethoden gezeigt, die von *GeWare* unterstützt werden. Dabei wird in Analogie zu Kapitel 4 in Methoden der Vorverarbeitung, statistischen Analysen und Berichten sowie Visualisierung getrennt.

5.6.1 Vorverarbeitung

Im Zuge der Vorverarbeitung werden die Rohdaten (Oligo-Intensitäten) zu Probeset-Intensitäten transformiert. Dabei sind zwei hauptsächliche Aufgaben zu bewältigen, die in der Normalisierung und Aggregation bestehen. Das Ziel einer Normalisierung ist es, die Oligo-Intensitäten von verschiedenen Microarrays vergleichbar zu machen. Dazu werden einerseits die Rohdaten um Rauschanteile korrigiert, die beispielsweise im Scan- und Bildverarbeitungsprozess (vgl. Abschnitt 3.2) auf Grund der durch die Messgeräte induzierten Erfassung eines Signalgrundpegels entstehen. Andererseits werden die Intensitäten so modifiziert, dass unterschiedliche Gerätekalibrierungen und sonstige experimentelle Rahmenbedingungen die Höhe der Oligo-Intensität nicht mehr beeinflussen. In der Phase der Aggregation werden die normalisierten Signalwerte zu Probeset-Intensitäten transformiert.

GeWare bietet dem Benutzer die Möglichkeit, gezielt die Methoden für jeden einzelnen Schritt (d.h. Normalisierung und Aggregation) auszuwählen. Zusätzlich verwendet *GeWare* eine Menge von Methoden, die beide Schritte in sich vereinen. Zu diesen Methoden, die ebenso von *GeWare* für eine Datenvorverarbeitung angeboten werden, zählen beispielsweise RMA [IBC+03, IHC+03], das Li-Wong-Modell [LW01] in zwei verschiedenen Formen, eine in der angrenzenden Arbeitsgruppe "Signaltransduktion und Genexpressionsanalyse" am IZBI Leipzig entwickelte Empfehlung und die vom Microarray-Hersteller Affymetrix propagierte MAS5-Methode [Aff02].

5.6.2 Statistische Analysen und Berichte

GeWare besitzt Analysefunktionen, die von einfachen parametrisierbaren Berichten bis hin zu elaborierten statistischen Analysen reichen. Die parametrisierbaren Berichte dienen einerseits der Auswertung, z.B. durch Filterung und Sortierung, sowie zur Anzeige von zuvor ermittelten Analyseergebnissen. Andererseits verwendet *GeWare* diese Art von Berichten zur Detektion von Ausreißern unter Nutzung empirischer Filter. Solche Ausreißer sind Probesets (Gene), die einen extremen (hohen oder niedrigen) Expressionswert für einen einzelnen oder eine Gruppe von Microarrays im Vergleich zu anderen Microarrays oder Genen besitzen. Ausgewählte empirische Filter, die in *GeWare* Verwendung finden, sind beispielsweise die Folgenden.

- Ein erster Bericht zeigt alle Probesets, deren Expressionswerte einen größeren Abstand zum Mittelwert (bezogen auf das Probeset) besitzen als die f-fache Standardabweichung. Der Bericht listet aus allen Probesets $1 \leq i \leq m$ (mit $i, m \in \mathbb{N}$) diejenigen auf, deren Expressionswerte $e_{i,j} \in \mathbb{R}$ in den ausgewählten Microarrays $1 \leq j \leq n$ (mit $j, n \in \mathbb{N}$) Elemente der Menge

$$\{e_{i,j} | e_{i,j} > \overline{e_i} + \sigma(e_i) \cdot f \cup e_{i,j} < \overline{e_i} - \sigma(e_i) \cdot f\} \tag{5.1}$$

  sind. Hierbei ist $f \in \mathbb{R}$ der vom Benutzer spezifizierte Faktor, mit dem die Standardabweichung der Expressionswerte $\sigma(e_i)$ multipliziert wird und entweder zum Durchschnitt der Expressionswerte $\overline{e_i}$ addiert (Supremum) oder von diesem subtrahiert (Infimum) wird.

- Ein weiterer Bericht zeigt alle Probesets, deren Expressionswerte in einer Menge von Microarrays einer überdurchschnittlichen großen Schwankung unterliegen. Der Bericht listet aus allen Probesets $1 \leq i \leq m$ (mit $i, m \in \mathbb{N}$) diejenigen auf, deren Probeset-bezogene Standardabweichung $\sigma(e_i)$ der Expressionswerte $e_{i,j} \in \mathbb{R}$ größer ist als der um einen vom Benutzer spezifizierten Faktor $f \in \mathbb{R}$ vervielfachte Probeset-bezogene Durchschnitt $\overline{e_i}$, oder kurz: $\sigma(e_i) > \overline{e_i} \cdot f$.

- Mit einem weiteren Bericht werden alle Probesets aufgelistet, deren Expressionswerte in einer Gruppe von Microarrays im Vergleich zu einer anderen Gruppe immer größer/kleiner sind.

In diesen und anderen parametrisierten Berichten werden vor allem Treatment- und Gengruppen verwendet, um in einfacher Weise auf die relevanten Expressionsdaten zu fokussieren. Die Gruppen werden auch von verschiedenen Korrelationsanalysen verwendet, die einerseits die Korrelation zwischen den Probesets und andererseits zwischen Microarrays berechnen. Der

Benutzer analysiert mit ihnen in Abhängigkeit von der zugrunde liegenden Fragestellung und ausgehend von einem spezifizierten Probeset (Microarray) sowie einer Treatment-Gruppe (Gengruppe) die Probesets (Microarrays) mit den höchsten/niedrigsten Korrelationskoeffizienten. Das Ergebnis der Analyse besteht in einer Liste von Probesets (Microarrays), deren Korrelationskoeffizienten aufsteigend/absteigend[30] sortiert sind. Ein zu spezifizierender Grenzwert, der die relevanten Probesets (Microarrays) anhand ihres Korrelationskoeffizienten einschließt, gewährleistet eine zielgerichtete Analyse.

Weitere statistische Analysen sind beispielsweise die Lorenzkurve [Gas71, Gas72, PW97], der t-Test mit anschließender p-Wert Adjustierung nach Westfall [SKR04] und verschiedene auf *Resampling* basierende Analysemethoden nach Westfall/Young [WY93, GDS03, Lä05]. Die Lorenzkurve ist zusammen mit dem Gini-Koeffizienten eine aus der Volkswirtschaft stammende Analysemethode, um die Verteilung bzw. Konzentration von Werten auf ihre Merkmale zu untersuchen. Mit ihr lassen sich Aussagen formulieren wie "Die Top-10% der Probesets (in Hinsicht auf ihren Expressionswert) vereinen 80% aller gemessenen Expression.". Der t-Test[31] ist ein gängiger statistischer Hypothesentest, um Mittelwertunterschiede zwischen zwei Testreihen (z.B. die Expression eines Genes im gesunden vs. kranken Gewebe) oder einer Testreihe und der Grundgesamtheit aufzudecken. Jedoch ist die Expression eines Genes von verschiedenen Faktoren abhängig, die zumeist nicht erfasst und damit für eine Erklärung der berechneten Signifikanzwerte nicht zur Verfügung stehen. Dadurch können aus einer zweiten Testreihe/Stichprobe (unter sonst gleichen Bedingungen) andere Signifikanzwerte resultieren, die den Aussagen des ersten Signifkanztest widerspechen. Die Adjustierung der p-Werte soll die Effekte dieses Multiplizitätsproblems verringern. Auch die Resampling-Verfahren verfolgen das Ziel, falsche aber als signifikant erkannte bzw. klassifizierte Gene (sog. falsch positive Zuordnungen) zu vermeiden. Dazu werden in einer rechenintensiven Simulation auf Basis von ausgewählten Expressionsdaten iterativ verschiedene Gengruppen gebildet, deren berechnete Test-Statistik miteinander verglichen werden. Das Ergebnis besteht in einer Menge von als signifikant erachteten Genen.

5.6.3 Visualisierung

Obwohl die tabellarische Ausgabe der Ergebnisse die präferierte Präsentationsform in *GeWare* ist, existieren verschiedene zumeist auf einen Analysezweck zugeschnittene Visualisierungen, z.B. in Form von Linien-, Balken-

[30] Die Sortierung ist abhängig davon, ob die höchsten oder niedrigsten Korrelationskoeffizienten relevant sind.

[31] In der Statistik bekannt als Signifikanztest auf Basis einer t-Statistik.

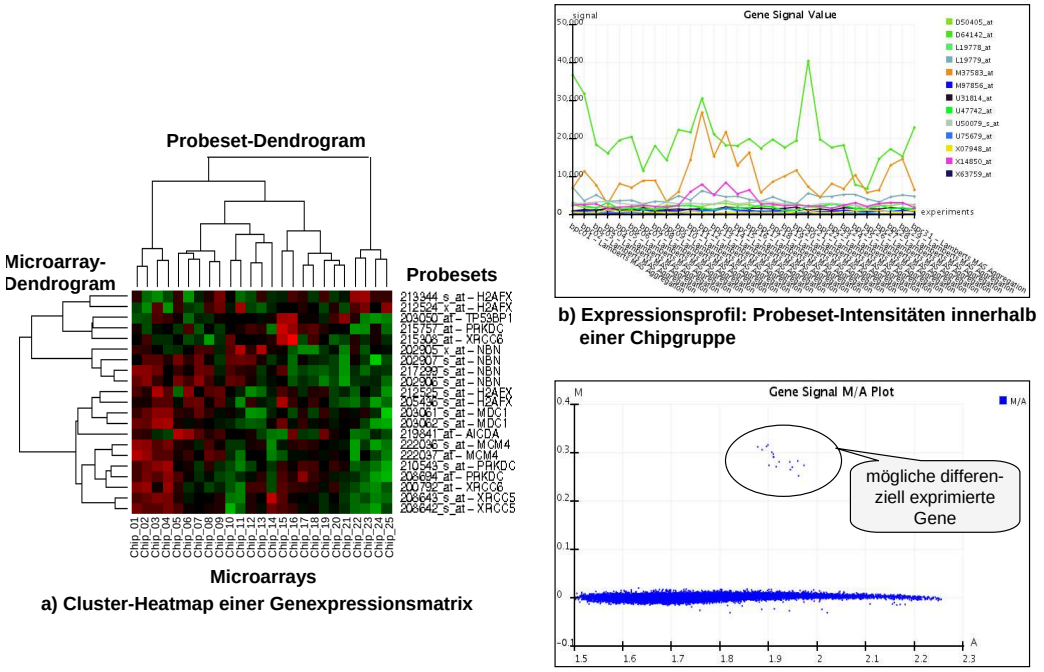
und Punktdiagrammen. Die Abbildung 5.6a zeigt eine derzeit populäre grafische Präsentationsform einer Genexpressionsmatrix, deren Probeset-Intensitäten durch eine ausgewählte Menge von Probesets (Gene) in den Zeilen und Microarrays in den Spalten determiniert wird. Dabei wurde sowohl die Menge der Probesets als auch die der Microarrays einer hierarchischen Clusteranalyse unterzogen, deren Ergebnis das Probeset-Dendrogram (links) und das Microarray-Dendrogram (oben) darstellt. Probesets (Microarrays) eines Clusters im hierarchisch organisierten Dendrogram weisen ein ähnliches Expressionsprofil auf. Bevor die Expressionswerte $e_{i,j} \in \mathbb{R}$ der korrespondierenden Probesets $1 \leq i \leq m$ (mit $i, m \in \mathbb{N}$) und Microarrays $1 \leq j \leq n$ (mit $j, n \in \mathbb{N}$) farblich visualisiert werden, unterliegen sie einer Standardisierung, die auf der folgenden Formel (vgl. [BB89]) basiert.

$$z_{i,j} = \frac{e_{i,j} - \overline{e}}{\sigma(e)} \tag{5.2}$$

Hierbei ist $\overline{e}$ der Durchschnitt der Expressionswerte und $\sigma(e)$ deren Standardabweichung. Im Ergebnis entsteht eine Menge von standardisierten Expressionswerten $z_{i,j}$, die um ihren Durchschnitt $\overline{z} = 0$ mit einer Standardabweichung $\sigma(z) = 1$ schwanken. Die farbliche Repräsentation der standardisierten Expressionswerte erfolgt in den Farben schwarz, grün und rot mit diversen farblichen Abstufungen. Dabei kennzeichnet die Farbe schwarz alle Werte, die sich innerhalb eines Toleranzbandes um den Durchschnitt befinden und damit gegenüber der durchschnittlichen Expression keine signifikanten Abweichungen aufweisen. Die Farbe rot (grün) wird allen Expressionswerten zugewiesen, die eine positive (negative) signifikante Abweichung von der durchschnittlichen Expression und damit eine Überexpression (verringerte Expression) darstellen.

Die Abbildung 5.6b zeigt die Expressionsprofile für eine ausgewählte Menge von Probesets in Form eines Liniendiagramms. Jede Linie verbindet die Expressionswerte (y-Achse) eines Probesets für die gewählte Menge an Microarrays (x-Achse). Somit lassen sich visuell Rückschlüsse auf das Expressionsverhalten der einzelnen assoziierten Gene in Hinsicht auf die Untersuchungsbedingungen (Microarrays) und im Vergleich zu anderen Genen (z.B. co-exprimierte Gene) ziehen.

Eine weitere derzeit häufig genutzte Visualisualisierungsform, das M/A-Diagramm, zeigt Abbildung 5.6c. Das Diagramm dient zur visuellen Identifikation von differentiell exprimierten Probesets (Genen), die sich aus dem Vergleich der Probeset-bezogenen Expressionswerte zweier präprozessierter Microarrays ergeben. Dazu wird für jedes Probeset (Gen) das Verhältnis nach der folgenden Formel bestimmt und als Punkt $p_{x,y}$ in das M/A-Diagramm



a) Cluster-Heatmap einer Genexpressionsmatrix

b) Expressionsprofil: Probeset-Intensitäten innerhalb einer Chipgruppe

c) M/A-Diagramm zur Identifikation differenzieller Gene

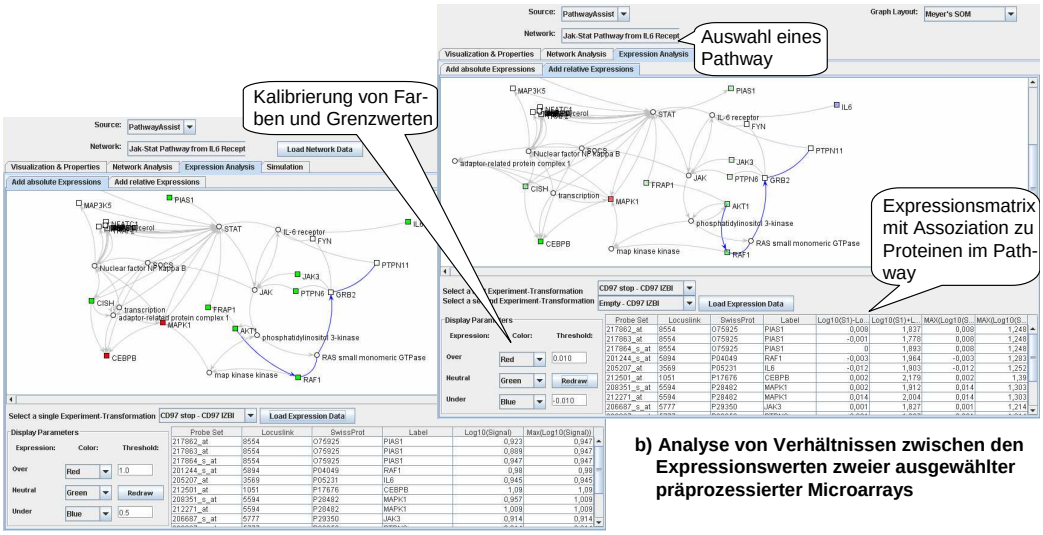
Abbildung 5.6: Ausgewählte Visualisierungsformen von Analyseergebnissen

eingetragen.

$$p_{x,y} = \frac{\frac{1}{2}\log_2(e_{i,1} \cdot e_{i,2})}{\log_2(\frac{e_{i,1}}{e_{i,2}})} = \frac{\frac{1}{2}(\log_2(e_{i,1}) + \log_2(e_{i,2}))}{\log_2(e_{i,1}) - \log_2(e_{i,2})} = \frac{A}{M} \tag{5.3}$$

Das in Abbildung 5.6c gezeigte M/A-Diagramm enthält eine Menge von Probesets, die eine auffällige differenzielle Expression aufweisen. Im Gegensatz zu anderen Genen sind deren Werte zwischen den beiden gewählten Microarrays unterschiedlich (große Differenz) und erreichen in der Summe einen hohen Wert, so dass aus der visuellen Analyse eine signifikante Expression unterstellt werden kann. *GeWare* bietet die Möglichkeit, diese identifizierten Probesets in einer Gengruppe abzuspeichern, die in einer weiteren Analyse einer genaueren Untersuchung unterzogen werden kann.

Darüber hinaus bietet *GeWare* eine Möglichkeit, die Expressionsdaten im Kontext von verschiedenen biologischen Netzwerken zu analysieren. Hierzu wurden die Daten ausgewählter biologischer Netzwerke auf Grund von Anwenderpräferenzen aus einem kommerziellen System in *GeWare* integriert. Diese biologischen Netzwerke bestehen aus verschiedenen molekularbiologischen Objekten (z.B. Proteine, Enzyme), die als Knoten repräsentiert und durch Kanten unterschiedlicher Semantik (z.B. Bindung, Regulation, enzy-



a) Analyse der Genexpressionwerte eines präprozessierten Microarrays im Kontext eines biologischen Netzwerkes

b) Analyse von Verhältnissen zwischen den Expressionswerten zweier ausgewählter präprozessierter Microarrays

Abbildung 5.7: Analyse der Genexpression unter Nutzung des ausgewählten molekularbiologischen Netzwerkes "Jak Stat Pathway from IL6 Rezeptor"

matische Aktivität) verbunden werden. Auf Basis von Mappings, die Korrespondenzen zwischen den Netzwerk-Objekten und den Probesets der Microarrays umfassen, ist eine Analyse der Expressionsdaten unter Nutzung der Netzwerke möglich. Die Abbildung 5.7a zeigt eine solche Analyse für das biologische Netzwerk "Jak Stat Pathway from IL6 Rezeptor" und den Expressionsdaten eines ausgewählten Microarrays. Die Höhe der Expressionswerte hat Einfluss auf eine farblich abgestufte Kennzeichnung (vgl. farbliche Kennzeichnung des Heatmaps in Abbildung 5.6a) von assoziierten Objekten im Netzwerk. Zusätzlich können Ergebnisse einer Netzwerkanalyse, z.B. kürzeste Pfade [Dij59] zwischen zwei Knoten (biologische Objekte), in die Analyse einfließen. Die Netzwerkanalyse wird in der Applikation (Applet) durchgeführt.

Die Abbildung 5.7b zeigt die Analyse der Expressionswerte eines ausgewählten Microarrays in Bezug zu einem anderen. Ähnlich wie bei der absoluten Analyse von Expressionswerten (vgl. Abbildung 5.7a) können mit der Spezifikation von Grenzwerten Überexpressionen und verringerte Expressionen farblich gekennzeichnet werden. Diese Art der Visualisierung der Expressionsdaten im Kontext von biologischen Netzwerken versetzt den Benutzer in die Lage, im Gegensatz zur tabellarischen Auswertung eine kontextbezogene Analyse zu betreiben.

5.6.4 Datenexport

Um eine Datenanalyse und -visualisierung mit Programmen zu unterstützen, die nicht in die Plattform integriert sind, besitzt *GeWare* eine zentrale Export-Schnittstelle. Diese Schnittstelle erlaubt sowohl den Zugriff auf die in den CEL-Dateien vorliegenden Oligo-Intensitäten (Rohdaten) als auch den Export von abgeleiteten Daten, z.B. in Form von Expressionsmatrizen. Zusätzlich stehen alle Analyseergebnisse, die beispielsweise in tabellarischer Darstellung auf der Web-Oberfläche angezeigt werden, in komprimierter Form für einen Export zur Verfügung. Dazu werden die aus einer Analyse/Anfrage resultierenden Daten in Form von CSV-Dateien im Dateisystem temporär gespeichert, von wo sie nach einer festgelegten Dauer automatisch gelöscht werden. Ebenso werden die in einer Analyse generierten Abbildungen und Diagramme in temporären Dateien gespeichert. Den dynamisch erzeugten Web-Oberflächen werden automatisch Verweise auf diese Dateien hinzugefügt, so dass der Benutzer unter Nutzung der Funktionalität des Web-Browsers Zugriff auf diese Daten erlangt.

Der Zugriff auf die CEL-Dateien sowie die in temporären Dateien gespeicherten Analyseresultate ist abhängig von den für den Benutzer und seine Benutzergruppe geltenden Berechtigungen.

5.7 Analyseintegration

GeWare verwendet verschiedene Techniken, um die im vorherigen Abschnitt beschriebenen Analysemethoden zu integrieren. Die parametrisierten Berichte sowie die Korrelationsanalysen verwenden datenbankinterne Funktionen und Prozeduren des DBMS DB2. Für spezielle aber einfache Analysen (vgl. Kapitel 6) wurden dem DBMS benutzerdefinierte Funktionen hinzugefügt. Datenbankeigene und benutzerdefinierte DB-Funktionen nutzen die Vorteile der engen Kopplung (vgl. Abschnitt 4.2.5) und insbesondere die des multidimensionalen Data-Warehouse-Schemas aus. Somit wird ein großes Spektrum an Anfragen gepaart mit kurzen Antwortzeiten ermöglicht.

Für andere Analysemethoden, insbesondere solche, die eine elaborierte Datenanalyse ermöglichen und nicht mit den vom Datenbank-Hersteller mitgelieferten Funktionen und Prozeduren durchgeführt werden können, greift *GeWare* auf bereits bestehende Programme und Java-Bibliotheken zurück. Damit wird eine aufwendige Reimplementierung vermieden. Zu diesen Analysemethoden zählen beispielsweise die Test- und Resampling-Verfahren aber auch die Visualisierungen, die über eine tabellenartige Darstellung hinaus gehen. Letztere sind zumeist auf Basis einer Java-Bibliothek realisiert, die

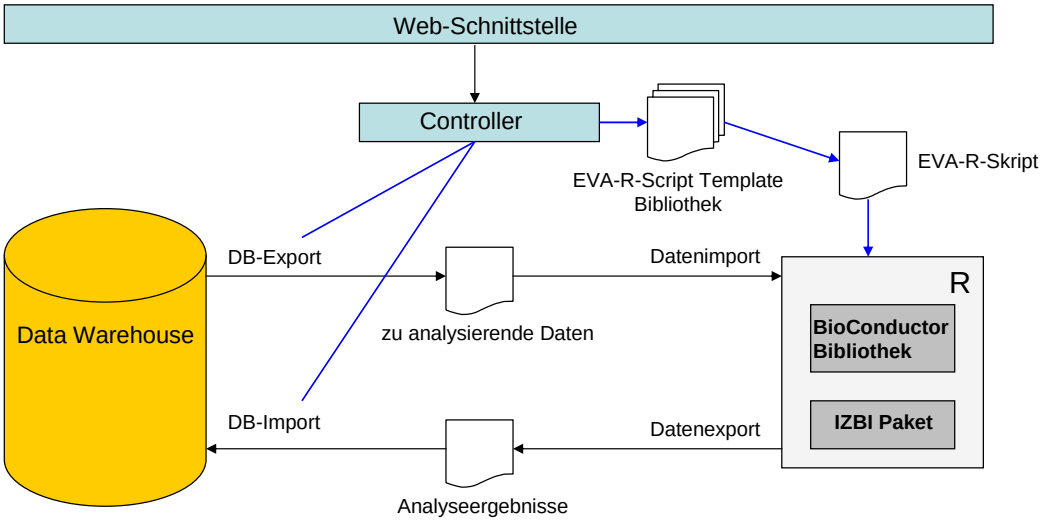


Abbildung 5.8: Transparente Analyseintegration von BioConductor Funktionen

die Daten unter Nutzung des Datenbank-API direkt verarbeiten und das Ergebnis in Bilddateien temporär speichern. Diese Bilddateien werden im Anschluss in die generierte Ergebnis-Webseite integriert.

Für viele elaborierte Analysemethoden verwendet *GeWare* die frei verfügbare statistischen Analysesoftware *R* [Dev06, Dal02]. Diese Software kann um spezifische Analysemethoden, z.B. für die Microarray-basierte Genexpressionsanalyse, erweitert werden, die in so genannten Paketen zusammengefasst sind. An der Entwicklung und Implementierung der Analysemethoden sind weltweit verschiedene Forschergruppen beteiligt, die die resultierenden Pakete zur freien Verfügung stellen. Eine Gruppe von Paketen, die auf die Analyse von Affymetrix Microarray-basierten Expressionsdaten fokussieren, werden im *BioConductor* Projekt [GCB+04, GCH+05] entwickelt und in einem namensgleichen Paket zusammengefasst. Darüber hinaus sind alle am IZBI entwickelten Analysemethoden in gesonderten Paketen enthalten. Da diese Software datenbankunabhängig ist und keine Zugriffsfunktionen auf eine DB2-Datenbank bietet, wie sie das *GeWare*-Data-Warehouse nutzt, ist ein Datenexport aus der Datenbank und anschließender Import in diese Analysesoftware notwendig.

Die Abbildung 5.8 zeigt die Arbeitsweise der für den Benutzer transparent integrierten Analysesoftware R inkl. notwendiger Pakete. Ein Controller nimmt die auf der Web-Oberfläche vom Benutzer vorgenommenen Eingaben entgegen, die Grundlage des durch den Controller veranlassten und gesteuerten Datenexports aus der Datenbank in temporäre Dateien ist. Da alle

zur Analyse notwendigen Methoden als Funktionen in zusätzlich installierten Paketen vorliegen, folgt die skriptgesteuerte Verarbeitung der aus der Datenbank exportierten Expressionsdaten in R dem EVA-Prinzip (EVA = Eingabe, Verarbeitung, Ausgabe). Dazu wählt der Controller in Abhängigkeit von den Benutzereingaben ein so genanntes *EVA-R-Skript-Template* aus den in einer Bibliothek zur Verfügung stehenden Skripten aus und ersetzt die definierten Eingabeparameter. Im Anschluss sendet der Controller das erzeugte R-Skript (als Instanz des ausgewählten Templates) an die R Software, wo es zur Ausführung kommt. Da das erzeugte Skript bereits die Ausgabe der Analyseergebnisse in eine temporäre Datei beinhaltet, ist abschließend lediglich der Import der Ergebnisdaten in die Datenbank notwendig. Nach Abschluss der Analyse stehen die importierten Analyseergebnisse zur Auswertung unter Nutzung verschiedener Darstellungsformen bereit.

Eine lose Kopplung, wie sie in Abschnitt 4.2.5 beschrieben wurde, wird in *GeWare* durch die Bereitstellung der Export-Schnittstellen unterstützt.

5.8 Abgrenzung zu verwandten Systemen

In Kapitel 4 wurden verschiedene relevante Plattformen zur Verwaltung und Analyse von Genexpressionsdaten benannt, von denen acht ausgewählte Systeme anhand von speziellen Kriterien vorgestellt wurden; eine detaillierte Evaluierung der Systeme enthält [DKR03]. Auf die Nachteile der dort beschriebenen Systeme wurde bereits eingegangen. In [GGL01] wird eine Gegenüberstellung von Microarray-Plattformen aus biologischer und Anwendersicht vorgenommen.

Eine weitere Plattform, die zur Verwaltung und Analyse von Daten dient, die mit Hochdurchsatz-Technologien erzeugt wurden, wird in [NAP04] beschrieben. Das System basiert wie *GeWare* auf dem Data-Warehouse-Ansatz und kann neben Expressions- und Mutations-Daten auch große Mengen chipbasierter Proteindaten aufnehmen. Letztere werden von *GeWare* nicht unterstützt. Klinische und pathologische Daten können aus Vorsystemen geladen werden und stehen für übergreifende Analysen zur Verfügung. Die Beschreibung der Experimente mit experimentellen Metadaten folgt der MIAME-Richtlinie. Jedoch unterstützt die Plattform nicht eine flexible Beschreibung, wie sie in *GeWare* auf Basis der Annotation Templates vorgenommen werden kann. Wie *GeWare* integriert die Plattform Annotationsdaten aus öffentlich verfügbaren Datenquellen. Jedoch werden sie im Gegensatz zu *GeWare* unter Nutzung eines applikationsspezifischen globalen Schemas in der Plattform materialisiert.

Das Ziel der caBIG/caGRID-Initiative [CHS+03, Bue05, SOH+06] des

NCICB (National Cancer Institute, Center for Bioinformatics) besteht darin, ein Netzwerk von Krebsforschungszentren und Laboratorien zu etablieren, um gegenseitige Stärken und Expertise besser ausnutzen zu können. Dazu wurden Standards, Vorgehensweisen, gemeinsame Applikationen und eine Grid-Infrastruktur geschaffen, die von jedem beteiligten Institut verwendet werden kann. Auf Basis dieser Infrastruktur können verschiedene Daten integriert und analysiert werden, z.B. Microarray-Daten für Genexessions- und Mutationsanalysen, Daten klinischer Studien sowie öffentlicher Quellen. Ein Zugriff auf diese Daten erfolgt virtuell über die Grid-Infrastruktur unter Nutzung spezieller Grid-Services [FKNT02, TCF+]. Im Gegensatz dazu setzt *GeWare* keine Grid-Infrastruktur ein. Für eine Spezifikation von experimentellen Metadaten sowie der Analyse von biologischen Objekten stehen zwei umfangreiche kontrollierte Vokabulare zur Verfügung. Der NCI Meta-Thesaurus basiert auf dem umfangreichen Unified Medical Language System (UMLS) [HL93, Bod04], das einerseits um nicht krebsspezifische Konzepte bereinigt wurde und dem andererseits Vokabulare anderer Quellen hinzugefügt wurden[32]. Der NCI Thesaurus[33] ist eine Eigenentwicklung des NCI und fokussiert ebenso wie der NCI Meta-Thesaurus auf die Terminologie der Krebsforschung. Die in *GeWare* integrierten Vokabulare sind dagegen weit weniger umfangreich als beide NCI Thesauri; der NCI Thesuarus umfasst etwa 25.000 Konzepte mit etwa 70.000 Termen (inkl. Synonymen). Auch für den Zugriff auf die NCI Thesauri stehen spezielle Grid-Services zur Verfügung.

Letztlich differieren alle Systeme und damit auch *GeWare* in den integrierten Analysemethoden. Sie sind vielfach von lokalen Entwicklungen und Präferenzen beeinflusst. Beispielsweise ermöglicht M-Chips eine Korrespondenzanalyse [FHB+01], während die in [NAP04] beschriebene Plattform vor allem durch eine Analyse unter Nutzung eines aus Literaturdaten gewonnenen Gen-Netzwerkes und Cross-Spezies-Vergleichen von Expressions- und Mutationsdaten sowie assoziierter Annotationsdaten auffällt.

5.9 Zusammenfassung

Das *GeWare*-System ist eine Datenintegrations- und Analyseplattform, die Expressions- und Mutationsdaten für eine Vielzahl von Chip-basierten Experimenten (Expressions- und Matrix-CGH-Arrays), experimentelle Metada-

[32] Eine Liste der integrierten Quellen kann unter `http://ncimeta.nci.nih.gov` eingesehen werden.

[33] `http://nciterms.nci.nih.gov/NCIBrowser/Connect.do?dictionary=NCI_Thesaurus&bookmarktag=1`

ten und klinische Daten integriert, verwaltet und analysiert. *GeWare* speichert alle relevanten Daten in einem multidimensionalen Schema. Die experimentellen Metadaten werden unter Nutzung von vordefinierten *Annotation Templates* und kontrollierten Vokabularen erfasst, integriert und verwaltet, die die benötigte Datenkonsistenz bei gleichzeitiger größtmöglicher Flexibilität sicherstellen. Grundlage hierfür ist einerseits das generische Schema, das eine einfache Erweiterbarkeit und Skalierbarkeit gewährleistet, und andererseits die Verwendung von kontrollierten Vokabularen. *GeWare* integriert verschiedene Methoden der Vorverarbeitung, statistische Analysen und Berichte sowie unterschiedliche Formen der Visualisierung. Die einheitlich in Treatment- und Gengruppen sowie Expressionsmatrizen zusammengefassten Analyseergebnisse können in neuen Analysen wiederverwendet werden und unterstützen damit sowohl eine iterative als auch eine vergleichende Analyse. Zur Optimierung der Performanz wurden verschiedene Datenbanktechniken eingesetzt, wie beispielsweise Indizierung, Materialisierung von oft in Analysen verwendeten, abgeleiteten Daten (vgl. auch Kapitel 6) und Implementierung von benutzerdefinierte Datenbankfunktionen. *GeWare* ist im operativen Zustand und wird bislang in verschiedenen Forschungsprojekten an der Universität Leipzig sowie in zwei deutschlandweiten klinischen Studien verwendet.

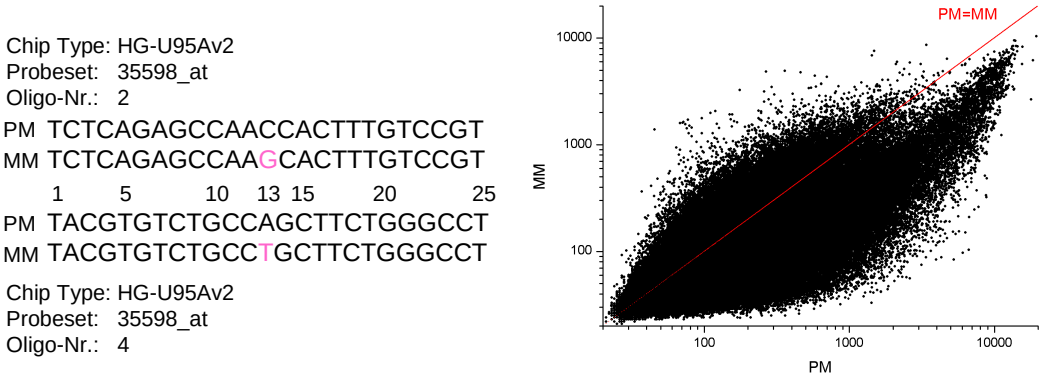
Kapitel 6

Sequenzbasierte Analysen von Oligo-Intensitäten auf Basis der GeWare-Plattform

6.1 Motivation

Auf einem Oligonukleotid-Microarray werden Gene mit mehreren voneinander verschiedenen Oligo-Sequenzen repräsentiert. Die zu einem Gen korrespondierenden Oligo-Sequenzen befinden sich nicht nebeneinander, sondern über den Chip verteilt, um möglichst zufällige Effekte und Verzerrungen auszuschließen. Ziel ist es, die Konzentration von mit Fluoreszenzfarbstoff markierten mRNA-Fragmenten zu messen, die während des stattfindenden Transkriptionsprozesses in einer Zelle erzeugt werden. Dazu werden die mRNA-Fragmente aus der Zelle extrahiert, fragmentiert und als Mixtur auf den Chip gegeben. Die Bindung der mRNA-Fragmente mit den auf dem Chip befestigten Oligo-Sequenzen soll letztlich Aufschluss darüber geben, welche zu diesen mRNA-Fragmenten korrespondierenden Gene als exprimiert gelten. Dabei kann die Messung durch verschiedene Faktoren beeinflusst werden, die das gemessene Signal "verrauschen" bzw. verzerren. Dazu zählt die Erkenntnis, dass die Bindungsstärke zwischen den mRNA Fragmenten und der entsprechenden Oligo-Sequenz sequenzabhängig ist. Weitere Faktoren be-

treffen u.a. den Beitrag unterschiedlich stark mit dem Fluoreszenzfarbstoff markierter mRNA-Fragmente und die nichtspezifische und konzentrationsabhängige Hybridisierung von konkurrierenden mRNA-Fragmenten mit den Oligo-Sequenzen [NM03, BKLS04].



a) PM/MM-Sequenz für einen ausgewählten Oligo

b) Visualisierung der PM-/MM-Intensitäten für einen ausgewählten Microarray

Abbildung 6.1: Vergleich von Sequenzen und gemessenen Intensitäten von PM und MM von Oligos eines Affymetrix Microarrays

Um insbesondere Effekte der unspezifischen Bindung zu erkennen und damit eine nachträgliche Korrektur der Messwerte zu ermöglichen, wird auf den Microarrays der Fa. Affymetrix jedes Oligo durch zwei Sequenzen repräsentiert, der Perfect-Match (PM) und Mis-Match (MM) Sequenz. PM und MM sind jeweils 25 Basenpaare lang und unterscheiden sich lediglich in der mittleren Base (Position 13): Gegenüber PM verwendet MM an dieser Position die entsprechend den Watson-Crick Paaren (C-G, A-T/U) komplementäre Base (vgl. Beispiele in Abbildung 6.1a). Ziel ist es, den Beitrag der spezifischen Bindung aus den gemessenen Intensitäten für PM und MM, z.B. nach dem folgenden Hybridisierungs-Modell aus [NLPM02] (vgl. auch [LW01, IBC$^+$03, BKH$^+$04]), zu errechnen.

$$I_{PM} = I_S + I_{NS} + B \tag{6.1}$$

$$I_{MM} = (1 - \alpha) I_S + I_{NS} + B \tag{6.2}$$

$$I_{PM} - I_{MM} = \alpha I_S \tag{6.3}$$

Hierbei kennzeichnet I_{PM} (I_{MM}) die gemessene Intensität des PM (MM) für ein betrachtetes Oligo, das sich additiv aus einem Beitrag I_S der spezifischen Bindung zwischen den komplementären Sequenzen, einem Beitrag I_{NS} nichtspezifischer Bindungen sowie einem optisch bedingten Hintergrund B zusammensetzt. Der nichtnegative Faktor α verringert die spezifische Bindung auf Grund des Sequenzunterschiedes in der mittleren Base (Pos. 13) der

MM-Sequenz. Da die für PM und MM gemessenen Intensitäten nichtnegativ sind, erscheint die intuitive Annahme realistisch, dass der errechnete Beitrag der spezifischen Bindung als Differenz der gemessenen Intensitäten von PM und MM ebenso nichtnegativ ist (vgl. Formel 6.3). Jedoch zeigt sich, dass bei ca. 30% der Oligos die MM-Intensität die der PM-Intensität übersteigt [NLPM02], d.h. der Beitrag nichtspezifischer Bindungen und des Hintergrundes ist höher als die spezifische Bindung. Das führt dazu, dass der errechnete Beitrag der spezifischen Bindung für ausgewählte Oligos nach dem o.g. Hybridisierungsmodell negativ ist. Die Abbildung 6.1b illustriert dieses PM/MM-Missverhältnis für den Microarray Expt01R1 des Latin-Square-Experiments (vgl. Abschnitt 6.4) in logarithmischer Darstellung; diese Erkenntnis ließ sich auch an den Daten anderer Microarrays desselben Experiments nachweisen.

Bisherige Normalisierungsverfahren, z.B. MAS5 [Aff02], RMA [IBC+03] und Varianten des Li-Wong-Modells [LW01], tragen diesem Umstand nur unzureichend Rechnung[34]. Insbesondere beziehen sie keine Sequenzdaten in die Korrektur, Normalisierung und Berechnung der Probeset-spezifischen Intensität ein. Daher wurde eine umfangreiche Analyse vorgenommen, um die vorgenannten Effekte unter Beachtung der Oligo-Sequenzen (PM, MM) charakterisieren zu können. Die Ergebnisse der Untersuchung wurden u.a. in [BKLS04, BKH+04, BPK05, BP05, BP06] publiziert und dazu benutzt, ein verbessertes Normalisierungsverfahren vorzuschlagen.

6.2 Besondere Projekt- und Analyseanforderungen

Neben allgemeinen (vgl. Kapitel 1) und Anforderungen, die sich an datenbankgestützte Plattformen wie *GeWare* zur Verwaltung und Analyse von Microarray-Daten richten (vgl. Kapitel 4), ergeben sich oftmals spezielle Anforderungen im Umfeld eines Projektes, die die vorgenannten erweitern oder konkretisieren. Für die Untersuchung der Oligo-Intensitäten waren insbesondere die folgenden speziellen Anforderungen relevant.

- **Integration von Sequenzdaten.** Die Expressionsdaten, die Gegenstand der Untersuchung sind, werden in verschiedenen Microarray-Experimenten erzeugt und können unter Nutzung unterschiedlicher Module und Web-Schnittstellen in die *GeWare*-Plattform geladen werden. Dagegen sind Oligo- und Probeset-spezifische Sequenzdaten (DNA-Sequenzen) noch nicht Bestandteil dieser Analyseplattform, da sie in

[34] Die Methode *gcRMA* [WI04, WIG+04] kam erst nach der Publikation der ersten Ergebnisse der durchgeführten Analyse auf.

den bisherigen integrierten Analysen nicht benötigt wurden. Zur Untersuchung der Oligo-Intensitäten sind beide Datenarten notwendig, d.h. Expressions- und Sequenzdaten, was eine Integration von Daten beider Arten erfordert.

- **Entwicklung und Integration spezieller Analyseroutinen.** Die Analyse hat explorativen, wissenschaftlichen Charakter und ist deshalb in ihrer komplexen Form weder im zugrunde liegenden DBMS noch in einem anderen Analyseprogramm vorhanden. Vielmehr ist es notwendig, aufbauend auf theoretischen Überlegungen iterativ Analyseroutinen zu entwickeln und zu verbessern. Dabei sollten die Analyseroutinen die durch die Plattform integrierten Daten möglichst effizient nutzen, um einerseits eine gute (Haupt-)Speichernutzung und andererseits trotz des hohen Datenvolumens eine gute Performanz sicherzustellen. Darüber hinaus soll die Plattform die Ergebnisdaten in aggregierter Form zur Verfügung stellen. Zusätzlich zur tabellarischen Repräsentationsform der Ergebnisdaten auf entsprechenden dynamisch generierten Webseiten sollen alle Analyseergebnisse exportiert werden können, um sie ggf. mit einer externen Analysesoftware manuell und nach persönlichen Empfinden zu visualisieren und zu vergleichen.

6.3 Integration von Sequenzdaten und Analyseroutinen

Im Gegensatz zu den Expressionsdaten, deren Umfang sich mit jedem neuen Microarray erweitert, hat das Datenvolumen von Sequenzdaten einen statischen Charakter. Jeder Microarray eines Typs verwendet die gleichen Oligos und damit die gleichen DNA-Sequenzen. Die Oligo-Sequenzen sind Teil einer Probeset-Sequenz. Sowohl die Auswahl der Probeset- als auch Einteilung und Selektion der Oligo-Sequenzen wurde für die verwendeten Microarrays von der Fa. Affymetrix nach eigenen, nicht publizierten Kriterien durchgeführt. Jedoch werden sowohl die Sequenzen als auch relevante Annotationsdaten, wie z.B. korrespondierende Gene öffentlicher Datenquellen, von Affymetrix in verschiedenen Formaten zur Verfügung gestellt. Damit wird der MIAME-Richtlinie [BHQ+01], die auch die Beschreibung der Microarrays fordert, Rechnung getragen.

Die Offenlegung der für einen Typ von Microarray verwendeten DNA-Sequenzen ermöglicht eine Integration von Sequenz- und Expressionsdaten. Im multidimensionalen Datenmodell (vgl. Abschnitt 5.4) der *GeWare*-Plattform werden die Expressionsdaten und erzielte Analyseergebnisse in Fak-

tentabellen (PROBESET-INTENSITÄTEN, EXPRESSIONSMATRIX) gespeichert, die mit verschiedenen Dimensionstabellen assoziiert sind. Die Gen-Dimension beschreibt die Probesets und Oligos der verwendeten Microarrays. Mit der Erweiterung der Dimensionstabellen um spezifische Attribute können die Sequenzdaten für Probesets und Oligos in diese Dimension eingefügt werden und stehen für diverse Auswertungen innerhalb der Datenbank zur Verfügung.

Um den Performanz-Anforderungen spezieller Analysen gerecht zu werden, wurden fortgeschrittene Datenbanktechniken, z.B. Indizierung, Vorberechnung und Materialisierung von oft benötigten Daten, eingesetzt. Dies betrifft beispielsweise die Vorberechnung und Materialisierung der Basenanzahl von Adenin (A), Cytosin (C), Guanin (G) und Thymin (T) in den Oligo-Sequenzen. Damit wurde eine schnelle Analyse erreicht, z.B. eine Gruppierung und Zählung von Oligos hinsichtlich der Basenanzahl für die Nukleotide A, C, G, T (vgl. Analyseergebnisse in Abbildung 6.2); eine vergleichende Analyse, die den durch die vorgenommene Materialisierung erlangten Performanzgewinn belegen kann, wurde nicht durchgeführt.

Für die durchgeführte Untersuchung wurden verschiedene Analyseroutinen eingesetzt, die iterativ entwickelt und auf Basis der erlangten Zwischenergebnisse verbessert wurden. Um eine Analyse mit hoher Performanz zu gewährleisten, wurde das der Plattform zugrunde liegende DBMS um einfache und noch nicht vorhandene Funktionen erweitert. Diese benutzerdefinierten Datenbankfunktionen können direkt in die Anfrageformulierung via SQL einbezogen werden. Mit ihnen werden einerseits der vom DBMS verwendete Speicherbereich ausgenutzt und somit aufwändige Transformationen (hinsichtlich Zeit und Speicher) von Datenstrukturen zwischen DBMS und Analyseprogramm vermieden. Andererseits kann eine Anfrageverarbeitung als der Teil der Analyse, in dem die größte Datenmenge zu verarbeiten ist, zentral im Data Warehouse (d.h. direkt in der Datenbank) vorgenommen werden. Darüber hinaus können die Vorteile, die sich durch die multidimensionale Modellierung ergeben, in einer Anfrageverarbeitung und somit einer Analyse ausgenutzt werden. Zu den in Form von benutzerdefinierten Datenbankfunktionen integrierten Analyseroutinen zählen u.a. die Folgenden.

- **baseCount.** Die Skalarfunktion $baseCount(c, s)$ ermittelt die Anzahl eines Zeichens c in einer Zeichenkette s der Länge n. Dabei entsteht die Zeichenkette s durch Konkatenation von Zeichen $s = s_1 s_2 \ldots s_{n-1} s_n$. Die Funktion ist definiert als

$$baseCount(c, s) = \sum_{i=1}^{n=|s|} \begin{cases} 1 & , c = s_i \text{ mit } s = s_1 s_2 \ldots s_n \\ 0 & , \text{sonst} \end{cases} \tag{6.4}$$

- **sequenceComplement.** Die Skalarfunktion $sequenceComplement(s)$ bestimmt die zu der DNA/RNA-Sequenz s komplementäre Sequenz. Dazu wird jedes Nukleotid s_i der Sequenz $s = s_1 s_2 \dots s_n$ $(1 \leq i \leq n = |s|)$ durch das jeweils komplementäre Nukleotid $\overline{s_1}, \overline{s_2} \dots \overline{s_n}$ (entsprechend den Watson-Crick-Paaren) ersetzt.

- **matchCount.** Die Skalarfunktion $matchCount(s,t)$ berechnet für zwei gleichlange Zeichenketten s und t die Anzahl an übereinstimmenden Zeichen in s und t. Die Funktion ist definiert als

$$matchCount(s,t) = \sum_{i=1}^{n=|s|=|t|} \begin{cases} 1 & , s_i = t_i \text{ mit } s = s_1 s_2 \dots s_n \\ & \text{ und } t = t_1 t_2 \dots t_n \\ 0 & , \text{sonst} \end{cases} \tag{6.5}$$

  Auf Basis dieser Funktion kann eine Ähnlichkeit $sim_{matchCount}(s,t)$ zwischen den zwei gleichlangen Zeichenketten s und t berechnet werden. Sie ist wie folgt definiert.

$$sim_{matchCount}(s,t) = \frac{matchCount(s,t)}{|s|} \text{ mit } |s| = |t| \tag{6.6}$$

Weitere Skalar- und Aggregationsfunktionen, die für die Untersuchung relevant waren, z.B. substr()[35], avg()[36], stddev()[37], sind bereits Bestandteil des DBMS DB2, das für die Data-Warehouse-Plattform eingesetzt wird.

Eine Visualisierung und/oder weitere spezielle Analysen können in externen Analysetools auf der Grundlage von vorberechneten und evtl. aggregierten Daten der *GeWare* Plattform erfolgen. Damit wird eine hohe Flexibilität der Analyse erreicht und somit dem explorativen Charakter der Untersuchung Rechnung getragen. Für besonders häufig durchgeführte Analysen wurde die Web-Schnittstelle der Plattform erweitert. Dabei werden die im Ergebnis der durchgeführten Analyse und mit der Web-Schnittstelle dargestellten Daten auch in komprimierter Form als Kopie für eine lokale Analyse angeboten.

6.4 Ausgewählte Analyseergebnisse

6.4.1 Analyseumgebung

Die durchgeführten Analysen basieren einerseits auf den von Affymetrix zur Verfügung gestellten Sequenzdaten für Probesets und Oligos. Andererseits

[35] Die Funktion substr() extrahiert eine Zeichenkette als Teil einer gegebenen Zeichenkette.
[36] Die Funktion avg() ermittelt das arithmetische Mittel.
[37] Die Funktion stddev() errechnet die Standardabweichung.

sind die Expressionsdaten des von Affymetrix durchgeführten Latin-Square-Experiments Grundlage der Untersuchung. Dieses Experiment hat zum Ziel, die Genexpression unter verschiedenen Konzentrationen (0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 pM) der zugegebenen Mixtur zu studieren. Das Experiment besteht aus 42 Microarrays des Chiptyps HG-U133A, die sich in 14 Gruppen zu jeweils drei Replikaten (Wiederholungen) unterteilen[38].

Die Sequenz- und Expressionsdaten wurden für die Analyse in der *GeWare*-Plattform integriert und mit den implementierten Analysefunktionen analysiert. Eine Visualisierung sowie die Vergleiche von Analyseresultaten wurden mit der Software Origin der Fa. OriginLab[39] durchgeführt.

6.4.2 Sequenzanalysen

Zu Beginn der Untersuchung wurden Analysen der Oligo-Sequenzen vorgenommen, die sich in zwei Komplexe mit unterschiedlichen Zielstellungen teilen. Während im ersten Analysekomplex die Oligo-Sequenzen untereinander mit dem Ziel analysiert wurden, Oligos mit einer sehr ähnlichen oder identischen PM-Sequenz zu ermitteln, konzentriert sich der zweite Komplex von Analysen auf die Zusammensetzung der Oligo-Sequenzen.

Tabelle 6.1: Ergebnis des paarweisen Oligo-Sequenzvergleiches für den Chiptyp HG-U95Av2

$matchCount(s_1, s_2)$	$sim_{matchCount}$	**Anzahl**
20	0,80	1.109
21	0,84	87
22	0,88	86
23	0,92	96
24	0,96	158
25	1.00	1.732

Ausgehend von der theoretischen Überlegung, dass der Beitrag einer spezifischen Bindung bei identischen oder sehr ähnlichen Sequenzen nicht korrekt ermittelt werden kann, wurde ein paarweises Alignment der Oligo-Sequenzen unter Zuhilfenahme der oben definierten Funktionen *matchCount* und $sim_{matchCount}$ vorgenommen. Ziel war es, Oligos mit einer identischen oder sehr ähnlichen Sequenz zu erkennen und zu markieren, um diese evtl. von

[38] Eine detaillierte Aufstellung der verwendeten Microarrays wird im Anhang B.1 gegeben.
[39] Weitere Informationen sind unter `http://www.originlab.com` (letzter Zugriff: 22.10.2006) erhältlich.

der Verwendung in einem späteren Normalisierungsverfahren auszuschließen. Die Tabelle 6.1 zeigt die Häufigkeit von gefundenen Sequenzpaaren für den Chiptyp HG-U95Av2 mit einer Ähnlichkeit größer gleich 0,8 ($matchCount \geq 20$)[40]. Danach ist die Anzahl von Oligo-Paaren mit einer identischen Sequenz (1.732) signifikant höher (> 10fach) als die von sehr ähnlichen Sequenzen, z.B. mit $sim_{matchCount} = 0.96$ (158). Daher sind die Oligos mit identischen Sequenzen von besonderem Interesse. Insbesondere sind sie Gegenstand der Äquivalenzrelation R über der Menge S von Oligo-Sequenzen des ausgewählten Chiptyps.

$$R \subseteq S \times S \text{ und es gilt für } \forall x, y, z \in S$$

$$\begin{aligned}
\text{Reflexivität: } & sim_{matchCount}(x, x) = 1 \\
\text{Symmetrie: } & sim_{matchCount}(x, y) = 1 \Leftrightarrow sim_{matchCount}(y, x) = 1 \\
\text{Transitivität: } & sim_{matchCount}(x, y) = 1 \text{ und } sim_{matchCount}(y, z) = 1 \\
& \Rightarrow sim_{matchCount}(x, z) = 1
\end{aligned}$$

Aufbauend auf der Äquivalenzrelation R können Äquivalenzklassen[41] gebildet werden, die jeweils unterschiedliche Oligos[42] mit einer identischen Sequenz beinhalten. Die Tabelle 6.2 zeigt die Häufigkeit solcher Äquivalenzklassen in Abhängigkeit von der Anzahl Oligos je Klasse. Es ist zu sehen, dass eine Klasse mit sieben Oligos vorhanden ist, aber bereits acht Klassen mit jeweils sechs Oligos, die eine identische Sequenz besitzen. Viele Oligos besitzen nur ein Duplikat (1.285 Klassen). Das im Anhang B.2 gegebene Beispiel soll belegen, dass die gemessene Intensität für Oligo-Sequenzen einer Äquivalenzklasse annähernd gleich ist, auch wenn die Oligos verschiedenen Probesets zugeordnet sind.

Im Mittelpunkt des zweiten Analysekomplexes stand die Zusammensetzung der Oligo-Sequenzen. Insbesondere galt es zu klären, ob die Anzahl und das Auftreten der Nukleotide (Adenin - A, Cytosin - C, Guanin - G, Thymin - T) in den verwendeten Oligo-Sequenzen einer bestimmten Verteilung folgen. Eine erste Analyse, deren Ergebnisse die Abbildungen 6.2a-f[43] illustrieren, untersucht die Häufigkeit von Oligos in Abhängigkeit von der Anzahl eines speziellen Nukleotids in den Oligo-Sequenzen für jeweils drei verschiedene Chiptypen der Gattungen *Homo Sapiens* und *Mus Musculus*.

[40] In die Untersuchung wurden nur die PM-Sequenzen eingeschlossen; für die MM-Sequenzen ergibt sich auf Grund der ausgetauschten Mittelbase ein äquivalentes Ergebnis.

[41] Es werden nur solche Äquivalenzklassen K betrachtet, deren Mächtigkeit $|K| > 1$ ist.

[42] Mit Unterschied ist hier sowohl die Bezeichnung als auch die Zugehörigkeit zum korrespondierenden Probeset gemeint.

[43] Die zu den Abbildungen korrespondierenden Daten sind im Anhang B.3 enthalten.

Tabelle 6.2: Häufigkeit von Oligo-Äquivalenzklassen für den Chiptyp HG-U95Av2

Anzahl Klassen	**Anzahl Oligos pro Klasse**
1	7
8	6
6	5
18	4
46	3
1.285	2

Dabei repräsentieren die Chiptypen HG-U95Av2 (Homo Sapiens, Abbildung 6.2a), HG-U133A (Homo Sapiens, Abbildung 6.2b), HG-U133_Plus_2 (Homo Sapiens, Abbildung 6.2c), MG-U74Av2 (Mus Musculus, Abbildung 6.2d), MOE430A (Mus Musculus, Abbildung 6.2e) und Mouse430_2 (Mus Musculus, Abbildung 6.2f) in der angegebenen Reihenfolge die chronologische Entwicklung der einzelnen Microarray-Typen.

Die Diagramme zeigen, dass die Häufigkeitsverteilung abhängig vom verwendeten Chiptyp ist. Das bedeutet, dass die Verteilung sowohl von der Spezies (Homo Sapiens vs. Mus Musculus) als auch von der Chip-Generation determiniert wird. Zum Beispiel sind die Kurvenverläufe der Nukleotide A und T sowie C und G für den Chiptyp HG-U95Av2 (MG-U74Av2) sehr ähnlich, während die Häufigkeitskurven der Nukleotide A und C sowie G und T für den Chiptypen HG-U133A und HG-U133_Plus_2 (MOE430A und Mouse430_2) eine hohe Ähnlichkeit aufweisen. Dagegen sind die geringeren Unterschiede zwischen den Chiptypen HG-U133A und HG-U133_Plus_2 (MOE430A und Mouse430_2) dadurch zu erklären, dass letzterer eine neuere Chip-Generation darstellt, in die die meisten der Oligo-Sequenzen des Chiptyps HG-U133A (MOE430A) eingehen.

Die Abbildungen 6.3a-f[44] zeigen Häufigkeitsverteilungen von Oligos in Abhängigkeit von der Position eines einzelnen Nukleotids in der Oligo-Sequenz. Analog zur vorherigen Analyse wurden mehrere Chiptypen (HG-U95Av2, Abbildung 6.3a; HG-U133A, Abbildung 6.3b; HG-U133_Plus_2, Abbildung 6.3c; MG-U74Av2, Abbildung 6.3d; MOE430A, Abbildung 6.3e; Mouse430_2, Abbildung 6.3f) in die Untersuchung einbezogen, um evtl. Gemeinsamkeiten oder Unterschiede sowohl zwischen den Chip-Generationen einer Spezies als auch zwischen den Chiptypen zweier Spezies festzustellen. Die Abbildungen illustrieren, dass sich die Häufigkeitskurven deutlich zwischen ers-

[44] Die korrespondierenden Datentabellen sind im Anhang B.4 aufgeführt.

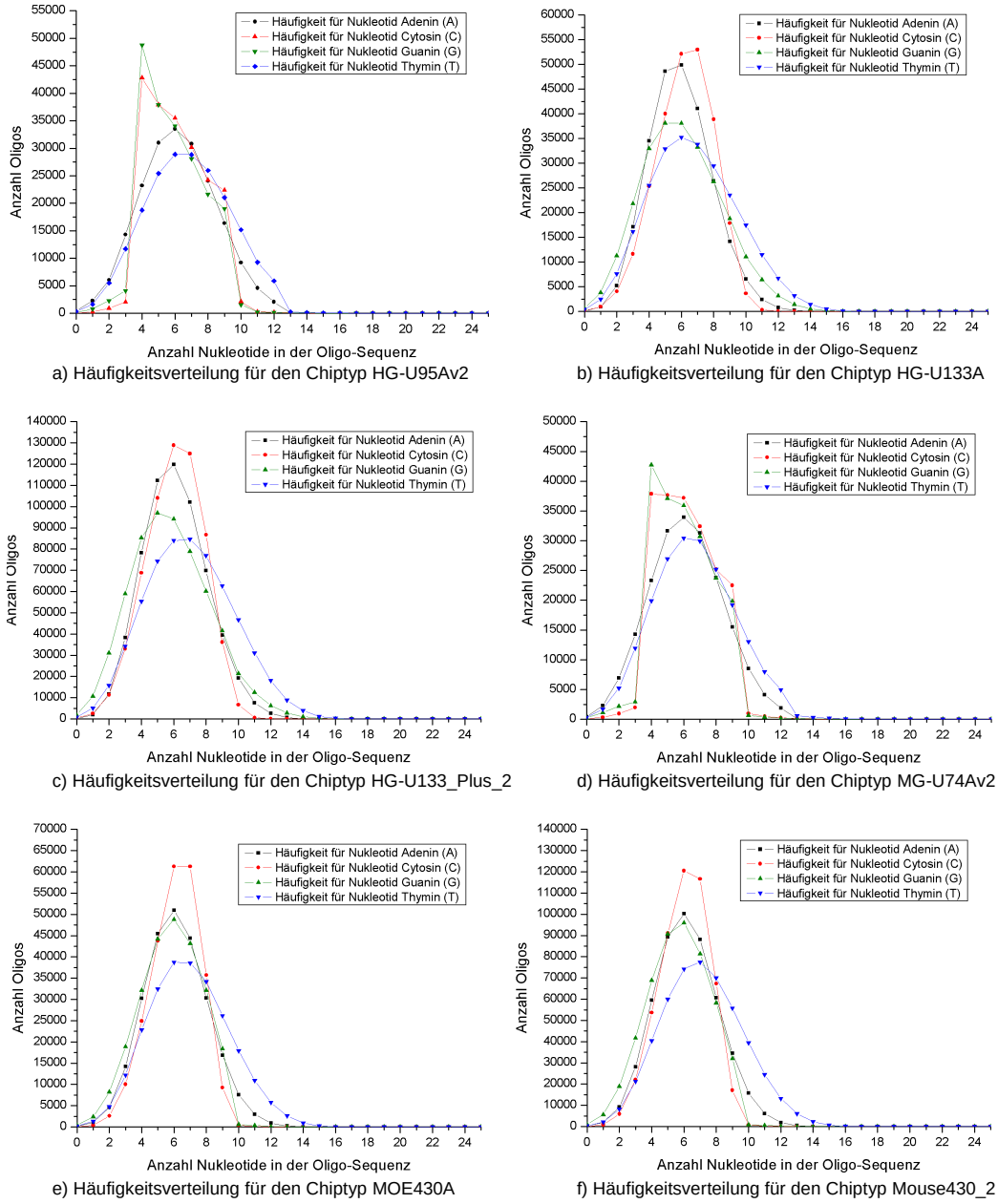
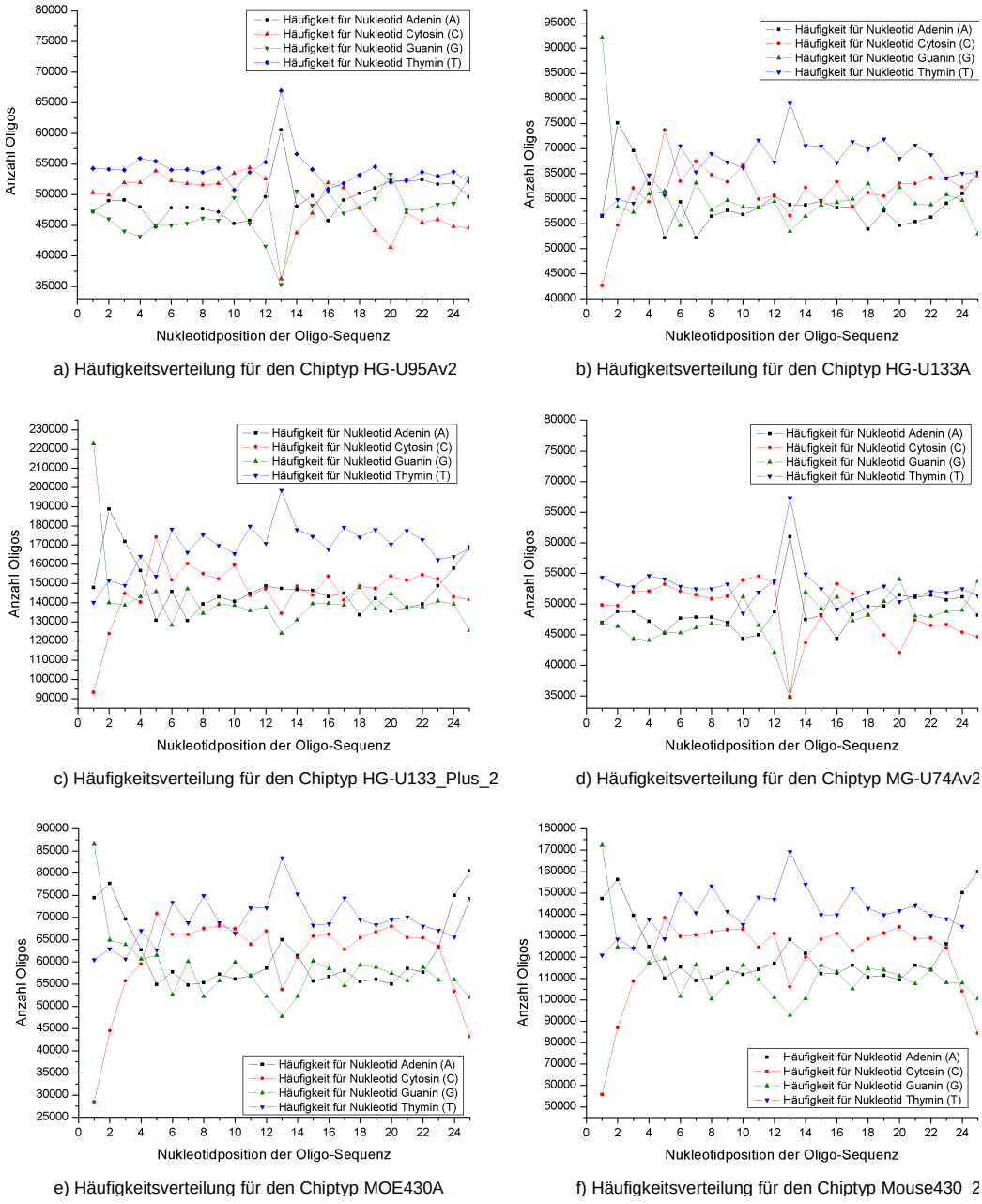


Abbildung 6.2: Häufigkeit von Oligos in Bezug auf die Anzahl der Nukleotide in den Oligo-Sequenzen

ter (HG-U95Av2, MG-U74Av2) und zweiter (HG-U133A, MOE430A) sowie dritter Chip-Generation (HG-U133_Plus_2, Mouse430_2) unterscheiden. Während die erste Chip-Generation einen signifikanten Unterschied der Nukleotidhäufigkeit in der mittelsten Sequenzposition (Position 13) aufweist, ist



a) Häufigkeitsverteilung für den Chiptyp HG-U95Av2

b) Häufigkeitsverteilung für den Chiptyp HG-U133A

c) Häufigkeitsverteilung für den Chiptyp HG-U133_Plus_2

d) Häufigkeitsverteilung für den Chiptyp MG-U74Av2

e) Häufigkeitsverteilung für den Chiptyp MOE430A

f) Häufigkeitsverteilung für den Chiptyp Mouse430_2

Abbildung 6.3: Häufigkeit von Oligos in Bezug auf die Position der Nukleotide in den Oligo-Sequenzen

dies bei der zweiten und dritten Generation jeweils am Anfang und Ende der Oligo-Sequenz der Fall.

6.4.3 Sequenzabhängige Analyse der Oligo-Intensität

Unabhängig von den einzelnen Ergebnissen der Sequenzanalysen wurde die Oligo-Intensität in Hinsicht auf die Oligo-Sequenz untersucht. Ausgangspunkt der Analyse bildete die Erkenntnis, dass das PM/MM-Missverhältnis (vgl. Abbildung 6.1b) das in Abschnitt 6.1 gezeigte Standard-Hybridisierungs-Modell negativ beeinflusst. Damit ist die Frage verbunden, inwieweit die Oligo-Intensitäten von der Mittelbase und dem Mitteltripel abhängt bzw. inwieweit spezielle Mittelbasen und Mitteltripel vom PM/MM-Missverhältnis betroffen sind. Dazu wurden die gemessenen Oligo-Intensitäten $I_{\text{Oligo}}^{(PM|MM)}$ (getrennt für PM und MM) jedes einzelnen Microarrays des Latin-Square-Experiments der folgenden Standardisierung (vgl. Formel 6.7 - 6.9) unterzogen.

$$\varnothing I_{\text{Oligo, Probeset}}^{(PM|MM)} = \frac{1}{N_{\text{Oligos im Probeset}}} \sum_{i=1}^{N_{\text{Oligos im Probeset}}} I_{\text{Oligo i}}^{(PM|MM)} \tag{6.7}$$

$$\sigma(I_{\text{Oligo}}^{(PM|MM)})_{\text{Probeset}} = \sqrt{\frac{\sum_{i=1}^{N_{\text{Oligos im Probeset}}} (I_{\text{Oligo i}}^{(PM|MM)} - \varnothing I_{\text{Oligo, Probeset}}^{(PM|MM)})^2}{N_{\text{Oligos im Probeset}}}} \tag{6.8}$$

$$Z_{\text{Oligo, Chip}}^{(PM|MM)} = \frac{I_{\text{Oligo}}^{(PM|MM)} - \varnothing I_{\text{Oligo, Probeset}}^{(PM|MM)}}{\sigma(I_{\text{Oligo}}^{(PM|MM)})_{\text{Probeset}}} \tag{6.9}$$

Hierbei bezeichnen $\varnothing I_{\text{Oligo, Probeset}}^{(PM|MM)}$ und $\sigma(I_{\text{Oligo i}}^{(PM|MM)})_{\text{Probeset}}$ die durchschnittlich gemessene Oligo-Intensität sowie deren Standardabweichung bezogen auf das korrespondierende Probeset. Im Ergebnis entstehen die standardisierten Intensitäten $Z_{\text{Oligo, Chip}}^{(PM|MM)}$ jedes Oligo eines Chips, die sich bezüglich ihrer Mittelbase X (Nukleotid X an der Position 13) der Oligo-Sequenz, $X \in \{A, C, G, T\}$ filtern und gegenüberstellen lassen. Für die Werte einer Gruppe von Microarrays (Replikatgruppen) des Latin-Square-Experiments wurde im Anschluss der Durchschnitt berechnet.

$$\varnothing Z_{\text{Oligo, Gruppe}}^{(PM|MM)} = \frac{1}{N_{\text{Chips der Gruppe}}} \sum_{i=1}^{N_{\text{Chips der Gruppe}}} Z_{\text{Oligo, Chip i}}^{(PM|MM)} \tag{6.10}$$

$$\varnothing Z_{\text{Oligo}}^{(PM|MM)} = \frac{1}{N_{\text{Gruppen}}} \sum_{i=1}^{N_{\text{Gruppen}}} Z_{\text{Oligo, Gruppe i}}^{(PM|MM)} \tag{6.11}$$

Die Abbildung 6.4 illustriert das PM/MM-Missverhältnis in Abhängigkeit von der Mittelbase auf der logarithmischen Skala (unter Nutzung der gemessenen Oligo-Intensitäten $I_{\text{Oligo}}^{(PM|MM)}$) für den Microarray Expt01R1 des Latin-

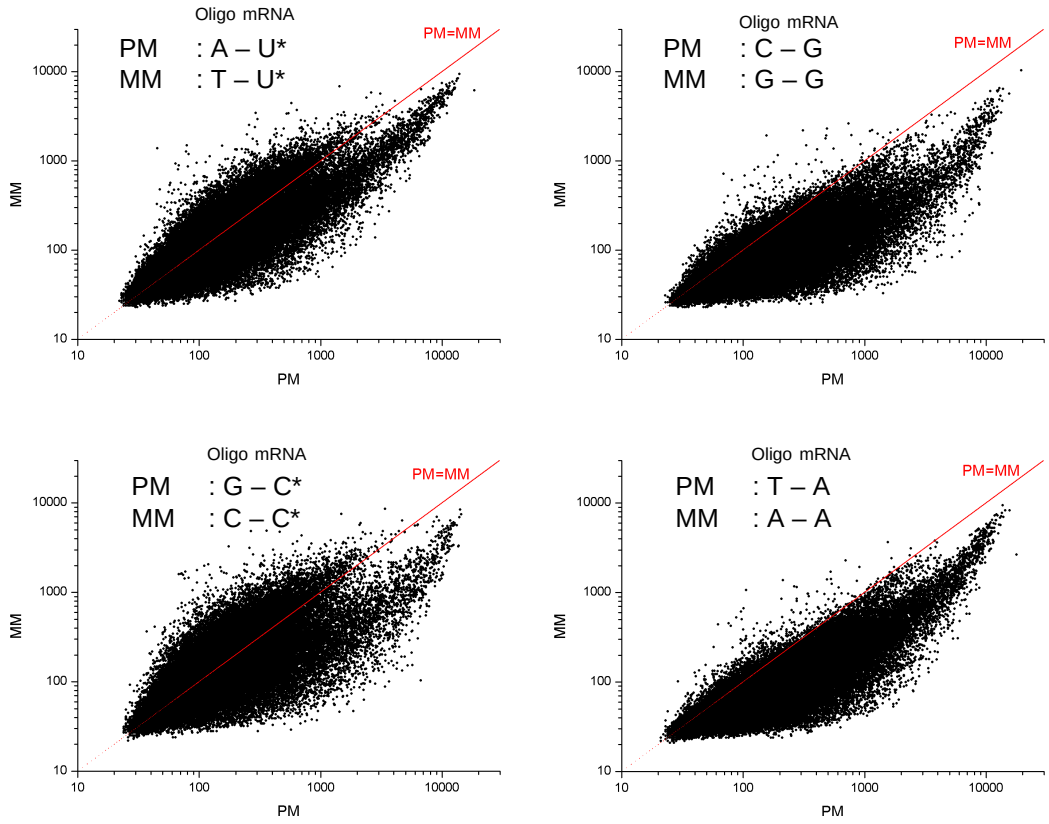


Abbildung 6.4: Oligo-Intensitäten (PM/MM) in Abhängigkeit von der Mittelbase

Square-Experiments. Unter der Annahme, dass die Oligos mit den zur PM-Sequenz komplementären Sequenzen binden, ergeben sich die in der Abbildung angegebenen Watson-Crick Paare für die mittlere Basenposition. Die mit dem Fluoreszenzfarbstoff versehenen Basen C und U in den Sequenzen der zugegebenen Mixtur sind mit einem "*" gekennzeichnet. Im Ergebnis zeigt sich für die mittleren Basen A (77,32%[45]) und G (68,44%) ein deutliches PM/MM-Missverhältniss, während es für die mittleren Basen C (31,71%) und T (34,51%) vergleichsweise gering ausfällt.

Im Weiteren wurde die Abhängigkeit der Oligo-Intensität vom Mitteltripel (Positionen 12-14) der Oligo-Sequenz untersucht. Dazu wurde ausgehend von der standardisierten Oligo-Intensität $Z^{(PM|MM)}_{\text{Oligo, Chip}}$ (vgl. Formel 6.9) die durchschnittliche Mitteltripel-Intensität $Z^{(PM|MM)}_{(XYZ),\text{Chip}}$ $(X, Y, Z \in \{A, C, G, T\})$

[45] Die Anzahl sowie der Anteil an den Oligos mit der jeweiligen Mittelbase sind im Anhang B.5 für jeden Microarray des Latin-Square-Experiments aufgeführt.

wie folgt berechnet.

$$Z^{(PM|MM)}_{(XYZ),\text{Chip}} = \frac{1}{N_{\text{Oligos mit XYZ}}} \sum_{i=1}^{N_{\text{Oligos mit XYZ}}} Z^{(PM|MM)}_{(XYZ),\text{Oligo } i} \tag{6.12}$$

$$\varnothing Z^{(PM|MM)}_{(XYZ)} = \frac{1}{N_{\text{Chips}}} \sum_{i=1}^{N_{\text{Chips}}} Z^{(PM|MM)}_{(XYZ),\text{Chip i}} \tag{6.13}$$

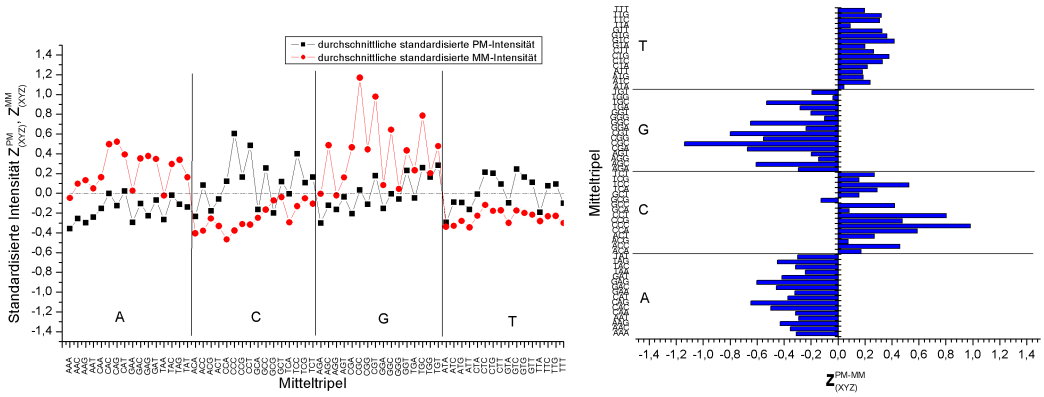
$$\varnothing Z^{(PM-MM)}_{(XYZ)} = \varnothing Z^{PM}_{(XYZ)} - \varnothing Z^{MM}_{(XYZ)} \tag{6.14}$$

Die Abbildung 6.5a stellt die standardisierten durchschnittlichen Intensitäten $\varnothing Z^{PM}_{(XYZ)}$ und $\varnothing Z^{MM}_{(XYZ)}$ für die Mitteltripel XYZ (vgl. Formel 6.11) gegenüber, wobei die Werte nach der Mittelbase sortiert sind. Es zeigt sich, dass die Mitteltripel hinsichtlich ihrer MM-Intensität in Abhängigkeit ihrer Mittelbase unter- (Mittelbasen C und T) und überdurchschnittlich (Mittelbasen A und G) repräsentiert sind. Darüber hinaus illustriert die Abbildung ein PM/MM-Missverhältnis ($\varnothing Z^{PM}_{(XYZ)} < \varnothing Z^{MM}_{(XYZ)}$) für einige Mitteltripel, wobei die größten Differenzen bei den Mittelbasen G (Mitteltripel CGC, CGT, CGA) und A (Mitteltripel CAG, GAG, CAC) auftreten. Die Differenzen $\varnothing Z^{(PM-MM)}_{(XYZ)}$ der standardisierten durchschnittlichen Intensitäten für die Mitteltripel sortiert nach deren Mittelbase werden in der Abbildung 6.5b dargestellt. Dabei wird deutlich, dass insbesondere die Mitteltripel mit den Mittelbasen A und G zum PM/MM-Missverhältnis beitragen, das dem Ergebnis der vorangegangenen Analyse (vgl. Diagramme der Abbildung 6.4) gleicht. Dagegen sind die PM/MM-Differenzen der Mitteltripel für die Mittelbasen C und T fast ausschließlich positiv; eine Ausnahme bildet lediglich das Palindrom GCG. Die größten PM/MM-Differenzen besitzen die Mitteltripel CGC ($\varnothing Z^{PM}_{(XYZ)} < \varnothing Z^{MM}_{(XYZ)}$) und CCC ($\varnothing Z^{PM}_{(XYZ)} > \varnothing Z^{MM}_{(XYZ)}$).

Aufbauend auf den Ergebnissen der vorangegangenen Analysen wurden elaboriertere Analysen [BKLS04, BKH+04, BPK05] mit dem Ziel durchgeführt, die Beiträge nichtspezifischer Bindungen exakter zu erfassen. Damit wurde eine Voraussetzung für die Entwicklung einer neuen Methode zur Vorverarbeitung von Expressionsdaten geschaffen, in der die gemessenen Intensitäten möglichst genau die Beiträge der spezifischen Bindungen wiedergeben.

6.5 Zusammenfassung

Im Mittelpunkt dieses Kapitels stand die Anwendung der *GeWare*-Plattform, um sequenzbasierte Analysen der Oligo-Intensitäten durchzuführen.



a) Standardisierte Oligo-Intensität (PM, MM) pro Mitteltripel

b) Differenz der PM- und MM-Intensität pro Mitteltripel

Abbildung 6.5: Durchschnittliche Intensitäten bezogen auf das Mitteltripel

Diese Art von übergreifenden Analysen wird durch die Integration von Sequenzdaten für Oligos und Probesets ermöglicht, die ebenso wie die Expressionsdaten auf der Grundlage des multidimensionalen Datenmodells in der Plattform gespeichert werden. Um dem explorativen, wissenschaftlichen Charakter der Analyse und der dadurch benötigten Flexibilität Rechnung zu tragen, wurden verschiedene Analyseroutinen in das der Plattform zugrunde liegende DBMS integriert, anhand derer Analyseergebnisse in aggregierter Form erzeugt werden, die nach ihrem Export mit weiteren externen Tools analysiert, verglichen oder visualisiert werden können. Die im Kapitel vorgestellten Analysefunktionen dienen vorrangig der Sequenzanalyse, z.B. zur Duplikatsuche von Oligos (in Bezug auf ihre Sequenz) oder den Häufigkeitsverteilungen von Oligos auf Basis ihrer Sequenz. Mit der Analyse der Oligo-Intensitäten in Abhängigkeit ihrer Sequenz konnte ein Verständnis der den Hybridisierungsprozess eines Microarrays beeinflussenden molekularen Faktoren erlangt werden. Damit wurde die Voraussetzung für die Entwicklung eines Verfahrens zur Vorverarbeitung von Microarray-Daten geschaffen, mit dem die gemessenen Intensitäten um Beiträge nichtspezifischer Bindungen und des Hintergrunds exakter korrigiert werden können.

Kapitel 7
Die GeWare-Plattform im Anwendungsbereich klinischer Studien

7.1 Motivation

Klinische Studien werden mit dem Ziel durchgeführt, den Heilungsverlauf und die Überlebensrate von erkrankten Patienten unter Anwendung von neuen oder adaptierten Therapien und Medikamenten zu studieren, um beispielsweise auf spezifische Arten von Krebs zu reagieren. Zu diesem Zweck werden viele patienten- und therapiebezogene Parameter in einer Studie aufgenommen, beobachtet und schließlich analysiert. Neben Analysen, die den Erfolg/Misserfolg von Therapien und Medikamenten widerspiegeln, wird oftmals versucht, klassifizierende Parameter zu finden, für die an der Studie teilnehmenden Patienten einen differenzierten Therapie- und Heilungsverlauf sowie -erfolg zeigen.

Weiterhin ist bekannt, dass Krankheits- und Therapieprozesse auf molekularbiologischer Ebene von Genen, Proteinen und ihren komplexen intra- und interzellulären Interaktionen beeinflusst werden. Beispielsweise unterliegen Krebszellen genetischen Mutationen, die eine gegenüber gesunden Zellen modifizierte Genexpression nach sich zieht und in fortgeschrittenen Krankheitsstadien umso weitreichendere Auswirkungen haben. Um diese Genotyp-Phänotyp-Beziehungen von Krankheiten und ihren Therapien besser zu ver-

stehen, wird es immer wichtiger, klinische und molekularbiologische Daten miteinander zu kombinieren. Damit wird es beispielsweise möglich, Beziehungen zwischen den pathologischen Klassifikationen und genomischen Disparitäten zu untersuchen [Cov03]. Dazu setzen diese Studien typischerweise Microarray-basierte Technologien ein, um eine patientenspezifische Genexpressionsanalyse durchzuführen [Kal05].

Aus der Notwendigkeit, klinische und molekularbiologische Daten zu kombinieren, resultieren spezifische Anforderungen hinsichtlich der Datenintegration. Diese unterschiedlichen Datenarten werden nicht nur in einer Vielzahl von unterschiedlichen Quellen verwaltet, sondern vielfach in verschiedenen, komplexen Systemen zur Datenverwaltung und -analyse gespeichert. An klinischen Studien sind typischerweise mehrere evtl. örtlich getrennte Institutionen beteiligt. Das führt zu komplexen, institutionsübergreifenden Arbeitsabläufen, die gewöhnlich mit kommerziellen Studienverwaltungssystemen unterstützt werden. Zu diesen Systemen zählen beispielsweise eResearch Network[46] (eRN), Oracle Clinical[47], und MACRO[48]. Viele dieser Studienverwaltungssysteme sind durch öffentliche Behörden zertifiziert, zu denen beispielsweise die Federal Drug Administration (FDA) in den USA und die European Medicines Agency (EMEA) in Europa zählen [KO03]. Dagegen werden experimentelle Daten, die unter Nutzung der Microarray-Technologie gewonnen wurden, typischerweise in speziell entwickelten Systemen verwaltet. In Kapitel 4 wurden bereits einige solcher Systeme vorgestellt. Klinische Parameter werden von ihnen nur unzureichend berücksichtigt.

Im Fokus dieses Kapitels steht die Anwendung und Erweiterung der *GeWare*-Plattform, um klinische und molekularbiologische Daten aus derzeit zwei großen kooperativen Studien der Krebsforschung zusammenzuführen und damit je Studie eine übergreifende Analyse zu ermöglichen. Eine erste Studie zielt auf die molekularen Mechanismen maligner Lymphome[49]; die zweite untersucht die molekularen Mechanismen von Gliomen[50] (Gehirntumor). Ein erstes Ergebnis [HBB+06] besteht in einer Klassifikation und damit in einer typspezifischen Unterscheidung von Lymphomen anhand klinischer Parameter, die auf Basis der experimentellen, molekularbiologischen Daten validiert wurden.

[46] http://www.ert.com
[47] http://www.oracle.com/industries/life_sciences/clinical.html
[48] http://www.infermed.com/macro/
[49] http://www.lymphome.de/en/Projects/MMML/index.jsp
[50] http://www.gliomnetzwerk.de/

7.2 Projektumgebung und spezifische Anforderungen

Im Mittelpunkt dieses Abschnittes stehen die Projektumgebung sowie -anforderungen, die ursächlich zu der entwickelten Lösung geführt haben. Begonnen wird mit einer Beschreibung der Projektumgebung und den verwendeten Daten.

7.2.1 Projektumgebung und resultierende Daten

Klinische Studien sind typischerweise mit komplexen Arbeitsabläufen (Workflows) verbunden, an denen verschiedene z.T. auch örtlich getrennte Organisationen beteiligt sind. Die Abbildung 7.1 zeigt einen Ausschnitt aus solchen Arbeitsprozessen in einer klinischen Studie, wobei auf die wichtigsten Schritte der Datenaquisitation fokussiert wird. Es beginnt mit der Definition von Kriterien, anhand derer relevante Patienten für die Teilnahme an der Studie ausgewählt werden. Die Kriterien bedürfen einer sorgfältigen Definition, da mit ihnen einerseits die zielgerichtete Auswahl an Patienten für die der Studie zugrunde liegenden Fragestellung erfolgt. Andererseits muss sichergestellt werden, dass Patienten in ausreichender Anzahl an der Studie teilnehmen, um eine statistisch valide Analyse zu ermöglichen. Die ausgewählten Patienten werden mit ihren *persönlichen Daten* erfasst. Zu diesen Daten zählen beispielsweise Alter, Geschlecht, Familienstand und die Unterscheidung in Raucher/Nichtraucher. Einige dieser persönlichen Eigenschaften reflektieren Eigenheiten und Gewohnheiten der Patienten, die einen großen Einfluss auf die spätere Analyse haben können, z.B. wenn die Daten getrennt nach Raucher und Nichtraucher klassifiziert werden.

Ein *klinischer Befund* wird erstellt, wenn einer der ausgewählten Patienten von einem Arzt untersucht wird. Diese Befunderstellung kann dabei einerseits einem regulären Untersuchungsplan folgen, z.B. durch regelmäßige Untersuchungen. Andererseits resultieren diese Befunde auch aus abnormalen Ereignissen, z.B. wenn der Patient auf Grund einer Erkrankung in ein Krankenhaus eingeliefert und dort untersucht wird. In beiden Fällen beschreibt der klinische Befund den gegenwärtigen Status des Patienten, so dass eine Rückverfolgung des Therapiestatus anhand präzis definierter Parameter möglich wird. Typischerweise werden solche klinischen Befunddaten in einem Studienverwaltungssystem erfasst und gespeichert.

Zusätzlich wird es oftmals notwendig, krankhaftes Gewebematerial, z.B. während einer Operation, zu extrahieren, das im Anschluss von Pathologen unter Nutzung verschiedener experimenteller Techniken, wie beispielsweise

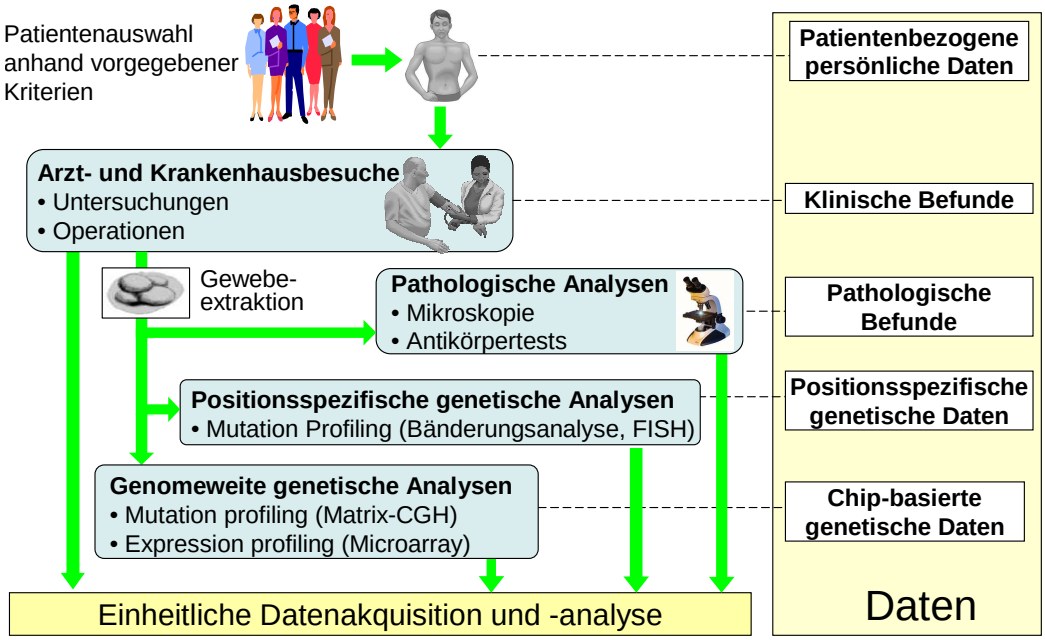


Abbildung 7.1: Projektumgebung und resultierende Daten

Lichtmikroskopie und Antikörpertests, genauer untersucht wird. Im Ergebnis dieser Untersuchungen beschreiben die Pathologen die Eigenschaften des extrahierten Gewebematerials und erstellen damit einen *pathologischen Befund*. Dieser Befund kann Einfluss auf die weiteren Entscheidungen der behandelnden Ärzte im Therapieprozess des Patienten haben.

Zusätzlich können Teile des extrahierten Gewebematerials verwendet werden, um spezielle Eigenschaften auf der genetischen Ebene experimentell zu eruieren. Dazu werden oftmals Expressions- und Mutationsprofile verwendet. Expressionsprofile zeigen das Expressionsverhalten bzw. die Aktivität der Gene unter spezifischen Bedingungen (z.B. gesundes vs. krankes Gewebe) auf und werden im Umfeld klinischer Studien zunehmend auf Basis von Microarrays (vgl. Abschnitt 3.2) erstellt.

Mutationsprofile fokussieren auf die genetische Diversität des Gewebematerials. Zu den experimentellen Technologien, die die genetische Mutation messen, zählen beispielsweise die Bänderungsanalyse [CZJM70], die Fluoreszent in situ Hybridisierung (FISH) [Mec95] und die Matrix-basierte vergleichende genomische Hybridisierung (Matrix-CGH) [KKS+92, SLS+97]. Die ersten beiden Techniken untersuchen die Mutation in einem spezifischen Bereich im Genom. Aus ihnen resultiert ein relativ kleines Datenvolumen oder lediglich eine vom Experimentator vorgenommene Beschreibung. Dagegen operieren die Matrix-CGH Experimente genomweit und produzieren wie Mi-

croarrays zur Genexpressionsanalyse eine enorme Menge an Daten. Darüber hinaus werden die Bänderungs- und FISH-Analysen dezentral in den einzelnen an der Studie teilnehmenden Krankenhäusern und Laboren durchgeführt, während die Matrix-CGH und Microarrays in einem für die Studie zentralen und spezialisierten Labor prozessiert werden.

7.2.2 Projektspezifische Anforderungen

Die dargestellte Projektumgebung einer klinischen Studie benötigt einen standardisierten Ansatz, um die Daten verschiedener Typen zu integrieren und darauf aufbauend eine effiziente Datenanalyse auszuführen. Neben bereits in den Kapiteln 1 und 4 diskutierten Anforderungen waren die Folgenden relevant.

- **Einheitliche Spezifikation der Daten**: Die anfallenden Daten werden zumeist in unterschiedlichen Krankenhäusern, Institutionen und Laboratorien, z.B. auf der Basis von Untersuchungen und voneinander unabhängigen Ärzten und Pathologen, generiert. Daher sind definierte Regeln, die die Dateneingabe betreffen, genauso wichtig wie standardisierte Formate zum Datenaustausch, um eine Vergleichbarkeit der spezifizierten Daten zu gewährleisten. Das betrifft nicht nur die Menge an Kategorien (Parameter), für die die Werte zu erheben sind, sondern auch die Werte selbst. Der Einsatz kontrollierter Vokabulare, Taxonomien und Ontologien ist ein probates Mittel, um den Wertebereich, der in Hinsicht auf eine Kategorie zur Verfügung steht, einzuschränken bzw. vorzugeben.

- **Autonome Dateneingabe**: Die direkte Dateneingabe in ein Studienverwaltungssystem sollte der manuellen Spezifikation der Daten auf formularbasierten Papierbögen vorgezogen werden. Damit findet die Dateneingabe an dem Ort statt, an dem die Daten erzeugt werden. Web-basierte Formulare unterstützen eine solche autonome Eingabe und helfen Fehlinterpretationen durch missverständliche Spezifikationen zu vermeiden. Das Studienverwaltungssystem kann die eingegebenen Daten zentral speichern. Zusätzlich sollten verschiedene Routinen mit dem Ziel der Validierung der eingegebenen Daten ausgeführt werden, um eine hohe Datenqualität sicherzustellen.

- **Zentrale molekularbiologische Experimente**: Jeder Typ von molekularbiologischen Experimenten sollte in einem zentralen Labor durchgeführt werden. Das ermöglicht die Ausführung der Experimente unter einheitlichen Laborbedingungen und gleichen Geräteeinstellungen. Das

ist Voraussetzung, um den experimentellen Fehler gering zu halten, und erlaubt eine vergleichende Analyse, in die die Daten aus unterschiedlichen Microarrays einfließen.

- **Datenschutzaspekte**: Gesetzliche Rahmenbedingungen erfordern den Schutz der Privatsphäre des Patienten. Insbesondere Daten, die den Patienten identifizieren, z.B. die Nummer des Personalausweises und der Sozialversicherung sowie der Name des Patienten, dürfen nicht zusammen mit den anderen klinischen und molekularbiologischen Daten gespeichert werden.

- **Verwendung bestehender Informationssysteme**: Typischerweise werden die patientenbezogenen persönlichen Daten sowie die unterschiedlichen Befunddaten in Studienverwaltungssystemen aufgenommen und organisiert, wohingegen die verschiedenen genomischen Datenbanken die großen Mengen an Microarray-basierten Expressions- und Matrix-CGH-basierten Mutationsdaten speichern. Um Zeit und Kosten bei der Entwicklung und Etablierung einer Integrations- und Analyseplattform zu sparen, sollte auf vorhandene Systeme zurückgegriffen werden. Deren Kopplung ist der Konzeption eines neuen Systems vorzuziehen.

7.3 Plattformarchitektur

Um den Anforderungen, die Gegenstand des letzten Abschnitts waren, in zwei großen Verbundprojekten im Bereich der Krebsforschung zu begegnen, wurde eine umfangreiche Integrations- und Analyseplattform an der Universität Leipzig aufgebaut. Die Plattform verbindet zwei an der Universität Leipzig bestehende Systeme, das *GeWare*-System (vgl. Kapitel 5) und das Studienverwaltungssystem eRN. Sowohl *GeWare* als auch eRN integrieren Daten aus unterschiedlichen Quellen, stellen autorisierten Benutzern Web-Oberflächen für eine interaktive Dateneingabe zur Verfügung und unterstützen die Analyse ihrer Daten. Die Abbildung 7.2 zeigt die Architektur der Plattform und die Kopplung der beiden Systeme im Überblick.

Das Studienverwaltungssystem eRN erlaubt den Anwendern von den an der Studie teilnehmenden Institutionen unabhängig patientenbezogene persönliche Daten sowie klinische und pathologische Befunddaten einzugeben. Dazu stehen vordefinierte Web-Oberflächen zur Verfügung. Eine *technische Patienten ID* wird unabhängig von beiden Systemen für jeden an der Studie teilnehmenden Patienten erzeugt. Eine Eingabe der Daten wird nur in

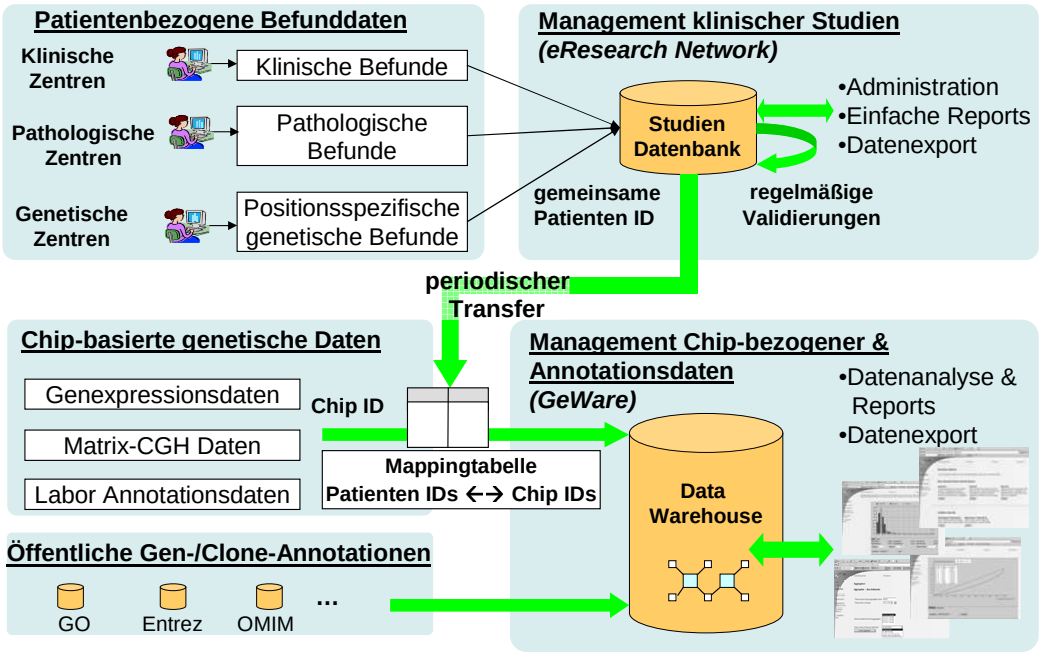


Abbildung 7.2: Kopplung der Systeme eRN und *GeWare*

Hinsicht auf eine Patienten ID vorgenommen, wobei alle Daten von einer Eingabe ausgeschlossen werden, die eine Identifikation des Patienten ermöglichen könnten. Die Korrespondenzen zwischen der Patienten ID und den identifizierenden Daten eines Patienten, wie z.B. dessen Name, Sozialversicherungsnummer etc., wird außerhalb der Plattform durch Dritte sicher verwahrt und genügt damit den rechtlichen Datenschutzaspekten und -richtlinien.

Das System verfügt über verschiedene regelbasierte Eingabefelder sowie Routinen zur Prüfung der Konsistenz und Kreuzvalidierung, um eine hohe Datenqualität zu gewährleisten. Damit werden lediglich Eingaben zugelassen, die den definierten Regeln entsprechen (z.B. Geburtsdatum $<$ aktuelles Datum). Spezielle Berichte zeigen gegensätzliche bzw. inkonsistente und fehlende Eingaben auf Basis der durchgeführten Validierungen, so dass der Benutzer Korrekturen vornehmen kann, bevor die Daten vom System akzeptiert werden und für die Analyse verfügbar sind. Alle Analysen des Studienverwaltungssystems sind über die Web-Schnittstelle zugreifbar. Jedoch beziehen sie sich ausschließlich auf die eingegebenen patientenbezogenen Daten und ermöglichen lediglich einfache statistische Auswertungen, wie z.B. die Anzahl der untersuchten Personen zu den verschiedenen Stati der Therapie.

Während das eRN-System die patientenbezogenen Daten verwaltet, sind die unter Nutzung von Microarrays und Matrix-CGH generierten Expressions- und Mutationsdaten Gegenstand des *GeWare*-Systems. In beiden Stu-

dien werden diese Daten in einem zentralen Labor produziert. Das eRN-System bietet keine Möglichkeit diese Daten zu speichern, da sie weitaus umfangreicher als die patientenbezogenen Daten sind.

Eine übergreifende Analyse, in die sowohl patientenbezogene als auch die voluminösen chipbasierten Daten einbezogen werden, setzt die Integration dieser Daten voraus. Aus diesem Grund importiert *GeWare* eine relevante Teilmenge der patientenbezogenen Daten aus eRN. Auswahl und Umfang der Daten sind abhängig vom Forschungsprojekt. Derzeit werden ca. 100 bis 130 Parameterwerte pro Patient übertragen. Während die patientenbezogenen Daten mit der anonymen Patienten ID assoziiert sind, verwenden die chipbasierten Daten in *GeWare* eine eindeutige, technisch generierte *Chip ID*, die keine Gemeinsamkeiten mit der *Patienten ID* aufweist. Keine der beiden IDs kann von der jeweils anderen abgeleitet werden. Daher wird eine zentrale *Mapping-Tabelle* verwendet, die jede *Patienten ID* mit den korrespondierenden *Chip IDs* assoziiert. Die Zuordnungen in dieser *Mapping-Tabelle* sind Grundlage für die korrekte Kombination der klinischen, pathologischen und experimentellen Daten und damit Basis für eine übergreifende Analyse. Externe, in öffentlichen Datenquellen verfügbare Daten integriert *GeWare* auf Basis eines Query-Mediators (vgl. Kapitel 8).

7.4 Integration von patientenbezogenen Annotationsdaten

In Abhängigkeit vom klinischen Fokus werden für eine Studie unterschiedliche Daten erhoben, die zur Dokumentation und Auswertung herangezogen werden. Beispielsweise beinhalten die pathologischen Befunde in einer Studie mit dem Ziel der Untersuchung der molekularen Mechanismen von Lymphomen eine Beschreibung der krankhaften Krebsknoten. Diese Beschreibung wird durch Parametern wie Knotengröße und -typ aber auch dem Status der Lymphknoten determiniert, da dieser eine wichtige Rolle im Stoffwechselprozess inne hat. Dagegen ist in einer Studie mit dem Fokus auf Gliomen die Gehirnregion, in der das Gliom lokalisiert ist, ein wichtiger Parameter. Somit differieren die Parameter je Studie, für die die Werte in den einzelnen Institutionen erfasst werden. *GeWare* verwendet *Annotation Templates*, um solche variablen Daten flexibel zu speichern (vgl. Kapitel 5).

Die Abbildung 7.3 illustriert den Prozess, um die patientenbezogenen Daten vom Studienverwaltungssystem eRN nach *GeWare* zu transferieren. Mit der Initiierung einer klinischen Studie werden in Abhängigkeit vom klinischen Fokus die notwendigen Kategorien definiert sowie dazu korrespondierend die

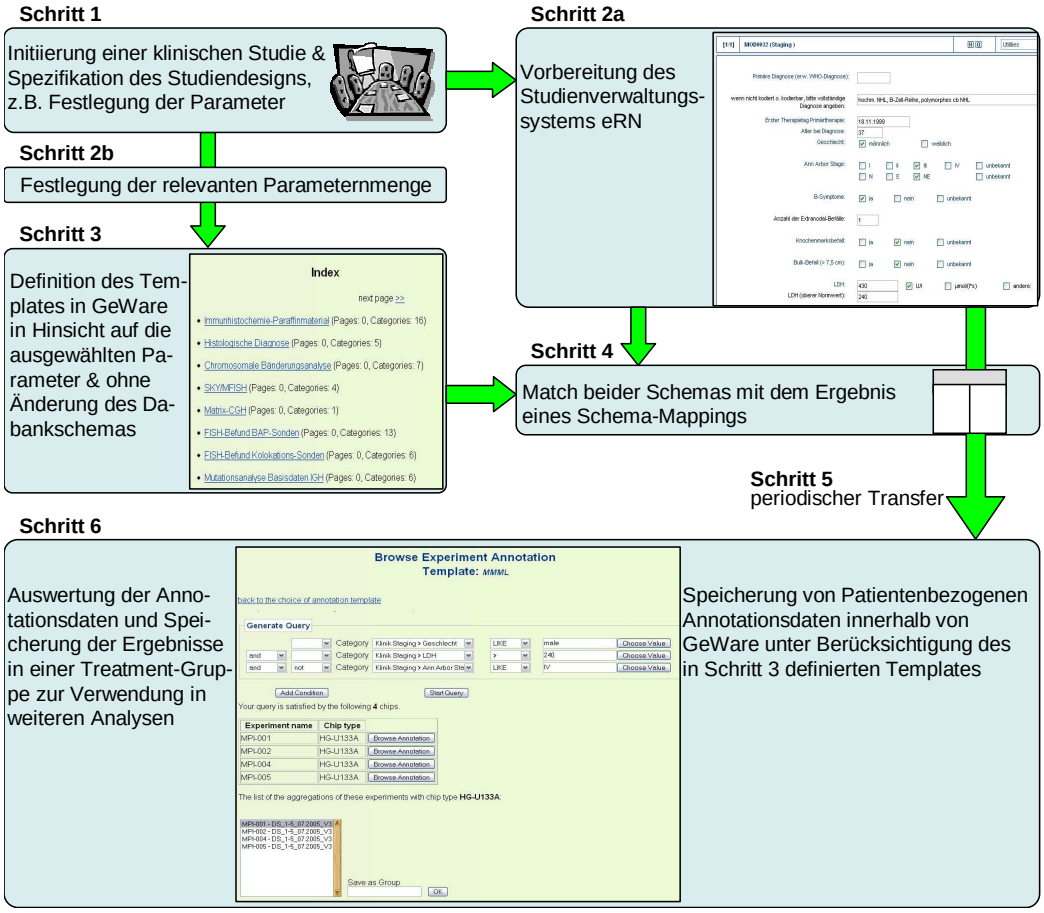


Abbildung 7.3: Definition, Transfer und Auswertung von patientenbezogenen Annotationsdaten

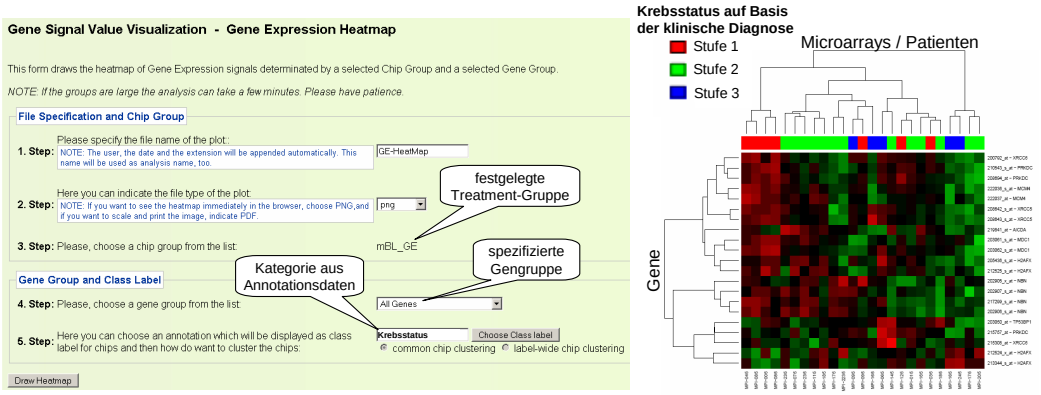
Wertebereiche festgelegt (Schritt 1). Anschließend wird das Studienverwaltungssystem eRN konfiguriert (Schritt 2a). Aus diesen Einstellungen werden Web-Oberflächen erzeugt, mit denen die Benutzer die Werte zu den definierten Kategorien erfassen können. Gleichzeitig kann die Teilmenge an Kategorien festgelegt werden, die für eine übergreifende Auswertung in *GeWare* zur Verfügung stehen soll (Schritt 2b). Basierend auf dieser Teilmenge wird in *GeWare* ein neues *Annotation Template* angelegt (Schritt 3), das die ausgewählten Kategorien in hierarchisch organisierten Seiten einordnet und die später zu transferierenden Daten aus dem eRN-System aufnimmt. Die generische Speicherung der *Annotation Templates* in *GeWare* macht keine Änderungen und Anpassungen des zugrunde liegende Datenbank-Schemas notwendig. Da beide Systeme eine unterschiedliche Repräsentation zur Speicherung der patientenbezogenen Daten benutzen, ist ein Schema-Mapping zwischen dem

Datenbank-Schema von eRN und den hierarchisch organisierten Kategorien in *GeWare* nötig (Schritt 4), bevor ein Datentransfer erstmalig durchgeführt werden kann. Im Ergebnis dieses Schema-Mappings wird jedem relevanten Element des eRN Datenbank-Schemas (d.h. die kategoriespezifischen Attribut- und Tabellennamen) ein korrespondierendes Zielelement in *GeWare* zugeordnet. Relevante Elemente des eRN Datenbank-Schemas orientieren sich an der identifizierte und zu übernehmenden Teilmenge an Kategorien. Da die Elemente in *GeWare* hierarchisch in den *Annotations Templates* organisiert sind, besteht das Zielelement in einem kategoriespezifischen Pfad, der ausgehend von der Index-Seite alle Seiten- und den Kategorie-Namen einschließt. Das Schema-Mapping wird derzeit manuell erstellt; auf semi-automatische Ansätze unter Nutzung von Schema-Matching Algorithmen [RB01] wird gegenwärtig nicht zurückgegriffen. Das Schema-Mapping ist die Grundlage für den täglichen Transfer der patientenbezogenen Daten aus dem eRN in das *GeWare*-System. Somit stehen diese Daten den in *GeWare* verfügbaren Analysen zur Verfügung (Schritt 6).

7.5 Übergreifende Analysen

GeWare verfügt über verschiedene Möglichkeiten der Datenanalyse (vgl. Abschnitt 5.6), die sich von parametrisierbaren Berichten und einfachen statistischen Analysen bis hin zu elaborierten Analysen und verschiedenen Arten der Visualisierung erstrecken. Diese Analyse- und Visualisierungsmethoden stehen grundsätzlich auch für die molekularbiologischen Daten den Verbundprojekten zur Verfügung. Jedoch sind sie zumeist auf die Analyse der experimentellen Daten ausgerichtet, in die keine klinischen und Annotationsdaten einbezogen werden. Deshalb wurden weitere Analyse- und Visualisierungsmethoden hinzugefügt bzw. bestehende angepasst. Zu diesen zählt die graphische Repräsentation der Expressionsdaten in einem Heatmap, wie sie bereits in Abbildung 5.6a gezeigt wurde.

Die Abbildung 7.4 zeigt die adaptierte Version des Heatmaps, in dem zusätzlich klinische Daten eingefügt wurden. Hierzu spezifiziert der Benutzer neben einer Treatment- und Gengruppe, die die Menge der Expressionsdaten begrenzen, auch eine Kategorie, die aus dem patientenbezogenen Annotationsprozess stammt. Im Beispiel, für das die Abbildung 7.4a die Spezifikation der Visualisierungsparameter zeigt, wurde der Krebsstatus als Annotationsdatum gewählt, der in der klinischen Diagnose als Klassifizierer fungiert. Jeder Krebsstatus entspricht einer Klasse, die die Häufigkeit von befallenen Lymphknoten repräsentiert. Dieser klinische Klassifizierer ist als Farbbalken über dem Heatmap in Abbildung 7.4b sichtbar. Dabei werden unterschied-



a) Spezifikation der Eingabeparameter

b) Heatmap mit ausgewähltem klinischen Parameter

Abbildung 7.4: Kombinierte Analysen

liche Krebsstati durch verschiedene Farben wiedergegeben. Damit kann der Benutzer visuell untersuchen, ob eine Korrelation zwischen der hierarchischen Ordnung, die aus der Clusteranalyse der Expressionsdaten resultiert, und der farblichen Fragmentierung des Farbbalkens, die die annotierten Krebsstati repräsentieren.

Andere Analysemethoden werden derzeit nicht von *GeWare* angeboten, da die Datenanalyse auf Grund von projektspezifischen Vereinbarungen in einer anderen Institution und unabhängig von der angebotenen Plattform stattfindet.

7.6 Zusammenfassung

Im Mittelpunkt dieses Kapitels stand die Anwendung und Erweiterung der *GeWare*-Plattform für zwei große, deutschlandweit durchgeführte, klinische Studien. Die Plattform integriert sowohl klinische als auch molekularbiologische Daten. Dazu wurde das *GeWare*-System mit dem Studienverwaltungssystem eRN gekoppelt. Das eRN-System gewährleistet die einheitliche und autonome Spezifikation von patientenbezogenen Daten, wobei alle eingegebenen Daten mit einer anonymisierten technischen *Patienten ID* assoziiert werden. *GeWare* importiert ausgewählte klinische Daten aus eRN, die hiernach zusammen mit Daten aus den in einem zentralen Labor durchgeführten Experimenten analysiert werden können. Dabei werden die importierten Annotationen in *GeWare* generisch verwaltet, so dass Annotationsdaten von klinischen Studien mit einem unterschiedlichen klinischen Fokus unterstützt werden. Das *GeWare*-System ist in klinischen Studien mit einer vergleichbaren Umgebung und Anforderung anwendbar.

Teil III

Mapping-basierte Datenintegration

Der ditte Teil fokussiert auf die flexible Integration von molekularbiologischen Daten. Dazu werden die vielfach in den Datenquellen in Form von Web-Links enthaltenen Objektkorrespondenzen ausgenutzt. Die Korrespondenzen zwischen zwei biologischen Objekttypen (z.B. Gene, Proteine) zweier ausgewählter Datenquellen formen ein so genanntes Mapping. Mappings bilden die Grundlage der beiden Ansätze, dem hybriden Integrationsansatz und *BioFuice*.

Kapitel 8 beschreibt einen hybriden Integrationsansatz. Der Ansatz nutzt eine zentrale Mapping-Datenbank, die eine Auswahl von Mappings enthält. Die Mappings entstammen öffentlich verfügbaren Datenquellen. Der Ansatz wird in der *GeWare*-Plattform (vgl. Kapitel 5) verwendet, um die Analyse und Interpretation von Analyseergebnissen zu unterstützen. Ausgewählte Performanzuntersuchungen belegen die Anwendbarkeit des Ansatzes.

Der *iFuice*-Ansatz in Kapitel 9 verwendet Mappings zur Peer-to-Peer-artigen Integration von öffentlich verfügbaren Datenquellen, privaten Daten und Ontologien. Kapitel 10 beschreibt die für Bioinformatik-Anwendungen entwickelten Erweiterungen, aus denen der *BioFuice*-Prototyp resultiert.

Letztlich charakterisiert Kapitel 11 ausgewählte verwandte Integrationsansätze und -systeme sowie deren Abgrenzung zum hybriden Integrationsatz und *BioFuice*.

Kapitel 8
Hybride Integration molekularbiologischer Annotationsdaten

8.1 Motivation

Viele Applikationen, wie z.B. für Genexpressions- und Mutationsanalysen, erfordern eine Integration zahlreicher Annotationsdaten aus unterschiedlichen Datenquellen. Zur Verdeutlichung der zu betrachtenden Arten von Daten zeigt Abbildung 8.1 einen Ausschnitt eines Gen-Eintrages der Referenzquelle LocusLink[51]. Der Eintrag enthält Beschreibungen zu dem Gen mit dem quellenspezifischen Identifikator 15, die in Annotations- und Mappingdaten unterschieden werden. Erstere sind quellenspezifische Attribute wie *Product*, *Alternate Symbols* etc., die durch eine verbale Beschreibung gekennzeichnet sind. Die Mappingdaten umfassen Web-Links zu anderen Datenquellen, welche durch die Identifikatoren (Accessions) aus den entsprechenden Quellen gekennzeichnet sind, z.B. 2.3.1.87 (Enzyme[52]) oder Hs.431417 (UniGene[53]).

Definition (Mapping)**.** *Unter einem Mapping wird eine Menge von Korrespondenzen zwischen den Objekten zweier Datenquellen verstanden. Die Se-*

[51] `http://www.ncbi.nlm.nih.gov/projects/LocusLink`
[52] `http://www.expasy.org./enzyme/`
[53] `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=engine`

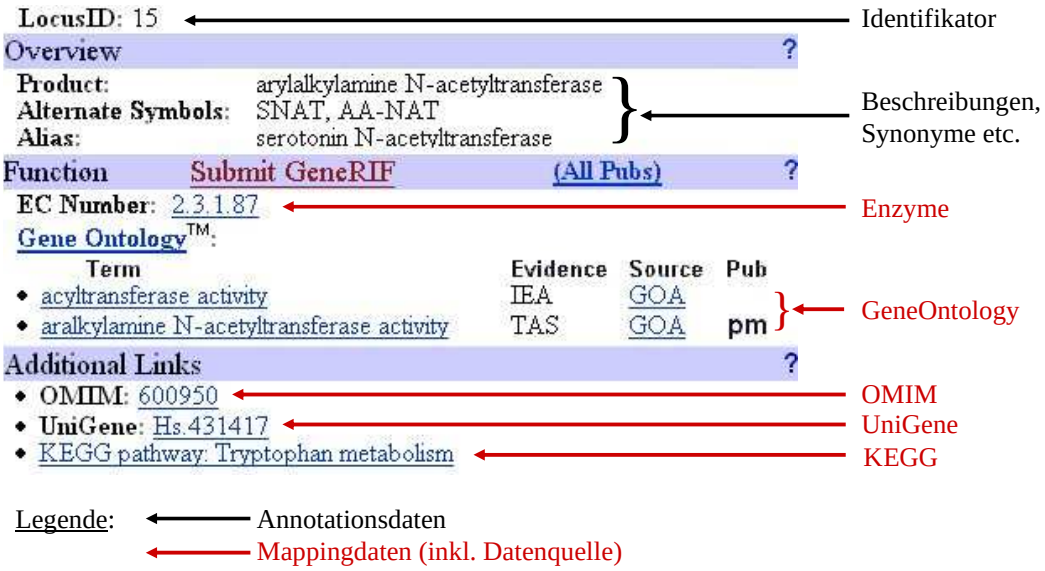


Abbildung 8.1: Annotations- und Mappingdaten in LocusLink

mantik eines Mappings kann explizit bei der Mapping-Erstellung vorgegeben werden und wird vom Integrationssystem beim Integrationsprozess verwendet oder verbleibt im Verantwortungsbereich des Benutzers.

Im o.g. Beispiel existieren Mappings zwischen der Quelle LocusLink und Enzyme, LocusLink und GeneOntology usw. Die Mappings versetzen den Benutzer in die Lage, zwischen den betreffenden Quellen zu navigieren und deren Daten zu verknüpfen, ggf. über mehrere Zwischenstationen hinweg. Im Beispiel können damit bei der Analyse des Gens auch Annotationen der referenzierten Objekte aus Enzyme und GeneOntology herangezogen werden.

Die Nutzung von Web-Links stellt einen einfachen Weg zur Datenintegration dar, welcher bereits weit verbreitet in den einzelnen Datenquellen benutzt wird. Allerdings wird damit nur die interaktive Analyse einzelner Objekte unterstützt, nicht jedoch die gleichzeitige und umfassende Auswertung für zahlreiche Objekte.

Dieses Kapitel präsentiert einen Ansatz zur Integration von Annotationsdaten aus öffentlichen Datenquellen, der zur Unterstützung von Expressionsanalysen in *GeWare* angewendet wird. Dabei sind die Expressionsdaten zusammen mit den experimentellen Metadaten physisch in dem Data Warehouse *GeWare* integriert. Die öffentlichen Annotationsdaten werden virtuell über einen Mediator integriert, wofür das verbreitete Tool SRS (Sequence Retrieval System) der Fa. BioWisdom[54] eingesetzt wird. Die Kopplung zwi-

[54] Die Software war bis 2006 ein Produkt der Fa. LION bioscience.

schen dem Warehouse und SRS erfolgt über einen eigens implementierten Query-Mediator. Die wesentlichen Beiträge sind:

- Eine materialisierte Integrationsform wird mit einer virtuellen Datenintegration kombiniert, um die Vorteile beider Ansätze in einem neuartigen hybriden Ansatz zu vereinen. Einerseits kann für komplexe Analysen experimenteller Daten im Warehouse eine hohe Performanz erreicht werden. Andererseits können über den Mediator bedarfsgesteuert aktuelle Annotationsdaten für die Auswertung herangezogen werden.
- Öffentliche Datenquellen werden einheitlich durch das bewährte Mediator-Tool SRS eingebunden, welches Schnittstellen zu zahlreichen öffentlichen (sowohl dateibasierten als auch relationalen) Datenquellen bereitstellt. Dadurch wird eine redundante Implementierung von Importfunktionen vermieden, und es können zahlreiche Datenquellen einfach in die Integrationsplattform eingebunden werden.
- Mappings werden explizit aus den Datenquellen extrahiert und in einer eigenen Datenbank, der Mapping-Datenbank, materialisiert. Die explizite Trennung der Mappings von anderen Daten erlaubt, Join-Wege zwischen Datenquellen flexibel zu bestimmen und zur Performanzoptimierung ggf. vorauszuberechnen.

8.2 Analyseszenarien

Die Abbildung 8.2 stellt zwei typische Analyseszenarien aktueller Bioinformatik-Anwendungen dar, die Expressions- und Annotationsanalyse. Eine Expressionsanalyse ermittelt und vergleicht mit Hilfe unterschiedlicher Technologien, wie z.B. Microarrays, den Aktivitätsgrad von Genen oder Proteinen unter unterschiedlichen Bedingungen der Zelle, beispielsweise im normalen und kranken Gewebe. Das Ziel ist i.d.R., Gruppen von Genen/Proteinen mit ähnlichem Expressionsmuster zu identifizieren. Beispielsweise können Gene, die eine hohe Aktivität in den Krebszellen, nicht aber in den gesunden Zellen aufweisen, für die unkontrollierte Teilung der Krebszellen verantwortlich sein. Die Analyse der Annotationen einzelner Gene der Gruppe kann nun Aufschluss darüber geben, ob eventuell auch ein gemeinsames Muster mit ähnlichen molekularen Funktionen vorhanden ist. Umgekehrt können anhand der Suche in Annotationsdaten Gruppen von Genen oder Proteinen mit ähnlichen funktionalen Eigenschaften identifiziert werden. Für diese Gen-/Proteingruppen kann anschließend eine Expressionsanalyse durchgeführt werden, um Einblicke in das Expressionsmuster zu gewinnen.

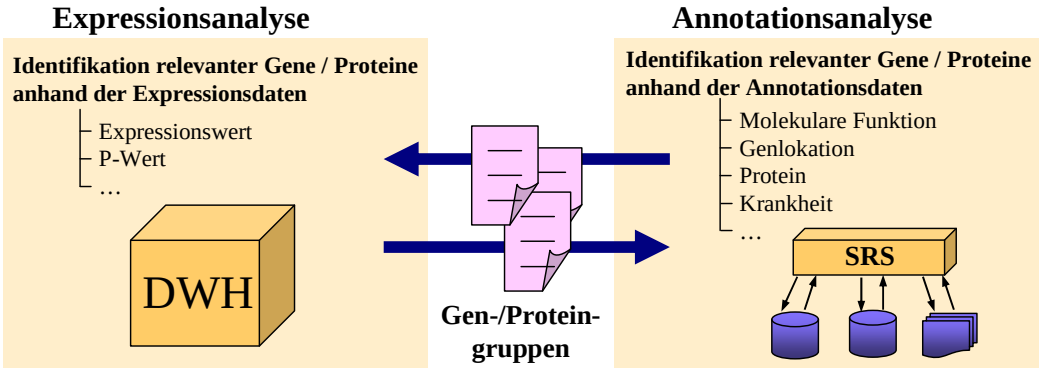


Abbildung 8.2: Analyseszenarien

8.3 Architektur im Überblick

8.3.1 Komponenten

Aufgrund der im letzten Abschnitt beschriebenen Integrationsanforderungen wurde eine hybride Integrationslösung entwickelt. Die Abbildung 8.3a zeigt die Architektur des Integrationsansatzes. Die beteiligten Komponenten sind:

- **GeWare** (vgl. Kapitel 5) dient als Integrationsplattform für den hybriden Ansatz. Die Annotationsdaten aus öffentlichen Datenquellen werden bei der Interpretation von Analyseresultaten eingesetzt, wie z.B. bei der Suche nach Gemeinsamkeiten von als aktiviert (exprimiert) identifizierten Genen.

- **SRS** dient als Mediator zu verschiedenen öffentlichen Datenquellen. Zur Zeit werden die Annotationsdaten aus LocusLink [PM01], GeneOntology [HCI+04] und Ensembl [BAB+04, PCC+04, HAC+05] integriert. Darüber hinaus sind die Identifikatoren (Accessions) der Datenquellen UniGene [PWS04, WCE+04, WBB+06] und NetAffx [LLS+03, CST+04] verfügbar.

- Der **Query-Mediator** dient zur Kopplung von GeWare und SRS. Seine Aufgabe besteht darin, die Benutzeranfragen in eine oder mehrere SRS-Abfragen zu übersetzen, sie durch SRS ausführen zu lassen und die Ergebnisse im Anschluss zu kombinieren.

- Die **Mapping-Datenbank** speichert die Mappings zwischen den Objekten der integrierten Quellen zur Join-Berechnung. Um den Bestand der Mappings klein zu halten, wird eine sternförmige Anordnung der

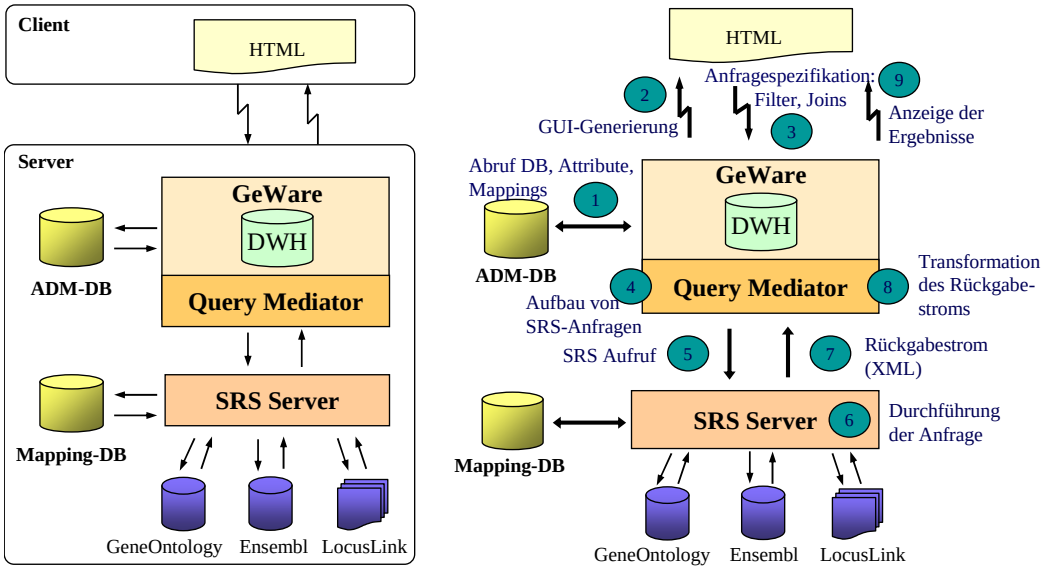


Abbildung 8.3: Integrationsansatz und Komponenten im Überblick

Quellen um eine zentrale Quelle vorgenommen. Die Mapping-Datenbank wird von SRS als Datenquelle eingebunden.

- Die **ADM-Datenbank** (ADM=Administration) speichert die Metadaten über die integrierten Quellen, z.B. die Namen der Quellen, deren Attribute und die verfügbaren Mappings. Anhand dieser Metadaten wird die Web-Oberfläche zur Anfrageformulierung automatisch generiert.

Der Aufbau der einzelnen Komponenten wird im Abschnitt 8.4 (Metadatenverwaltung in der Mapping- und ADM-Datenbank) sowie im Abschnitt 8.5 (Anfragebearbeitung im Query-Mediator) diskutiert. Die beiden folgenden Abschnitte beschreiben das Zusammenspiel der Komponenten in zwei Nutzungsprozessen: die Anbindung der Datenquellen und die Anfrageverarbeitung.

8.3.2 Anbindung der Datenquellen

Die umfangreiche Bibliothek verfügbarer Wrapper von SRS gestattet es, fast jede Datenquelle einzubinden. Aus Performanzgründen werden die Datenquellen i.d.R. als lokale Kopien angelegt, auf die über die Wrapper zugegriffen wird. In unserer Testinstallation sind die dateibasierte Quelle LocusLink sowie zwei relationale (MySQL) Datenbanken, Ensembl und GeneOntology,

derartig integriert. Ferner werden Metadaten über die Quellen, insbesondere die Namen der Datenquellen und deren Attribute erfasst und in der ADM-Datenbank von GeWare gespeichert.

Um den Aufwand für die Join-Anfragen gering zu halten, werden die Datenquellen in einem sternförmigen Graphen organisiert. Eine Datenquelle wird dabei als zentrale Datenquelle identifiziert, von der zu jeder anderen Quelle ein oder mehrere Mappings zur Verfügung stehen bzw. berechnet werden können. In der Genexpressionsanalyse ist LocusLink eine etablierte Referenzquelle für Genannotationen und wird deshalb als zentrale Quelle in unserer Integrationslösung verwendet. Für den Aufbau der Mapping-Datenbank werden die Mappings zwischen LocusLink und den anderen Quellen, wie Ensembl und GeneOntology benötigt. Diese werden (unter Nutzung des GenMapper-Tools [DR04]) aus den Quellen extrahiert und in die Mapping-Datenbank importiert. Für jede Verbindung zwischen der zentralen Quelle und einer anderen können mehrere Mappings importiert werden. Sie werden in der ADM-Datenbank erfasst und stehen in GeWare als alternative Join-Pfade zur Verfügung.

Neben den Quellen, die für Auswertungen zur Verfügung stehen sollen, wird die Mapping-Datenbank durch die relationale Schnittstelle in SRS registriert und integriert. Das hat den Vorteil, das die Integration aller Quellen - und damit auch der Mapping-Datenbank - einheitlich mit SRS erfolgt. Einerseits wird zusätzlicher Programmieraufwand vermieden, der sich aus einer direkten Verwendung der Mapping-Datenbank im Query-Mediator ergibt. Andererseits beschränkt sich die Aufgabe des Query-Mediators auf die Anfrage- und Ergebnistransformation (Formatierung); eine Datenmanipulation bleibt dem SRS-Server vorbehalten.

8.3.3 Anfrageverarbeitung

Die Abbildung 8.3b zeigt den allgemeinen Ablauf der Anfrageverarbeitung und wird im Abschnitt 8.5 detaillierter beschrieben. Im ersten Schritt (1) werden Metadaten zu den verfügbaren Quellen, deren Attribute und Mappings von der ADM-Datenbank abgerufen. Diese werden dazu genutzt, um eine Web-Oberfläche automatisch zu generieren (2). Auf der Web-Oberfläche kann der Benutzer Anfragen formulieren, indem er die relevanten Attribute und Datenquellen auswählt sowie Filter- und Join-Bedingungen spezifiziert (3). Die Anfragen werden an den Query-Mediator weitergegeben. Dieser interpretiert die Anfragen und generiert daraus einen Anfrageplan (4), welcher in eine oder mehrere SRS-spezifischen Abfragen umgesetzt wird. Der SRS-Server wird aufgerufen, um die generierten Abfragen zu bearbeiten (5). Die jeweiligen Selektionen und Projektionen für die ausgewählten Attribute wer-

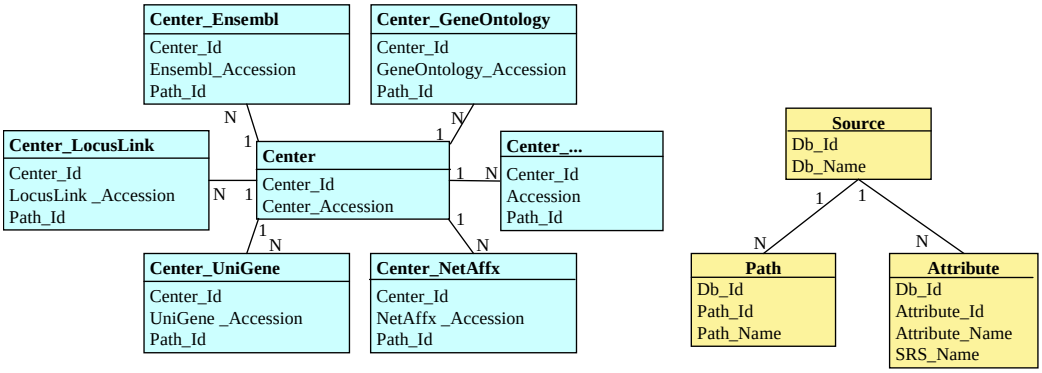
den in den Datenquellen ausgeführt, während die Join-Operationen an die Mapping-Datenbank zur Bearbeitung geschickt werden (6). Die Ergebnisse werden von SRS in einem XML-Datenstrom zurückgeliefert (7). Dieser Datenstrom wird vom Query-Mediator entgegengenommen, dessen Daten extrahiert (8) und in ein Ausgabeformat, z.B. HTML für die Anzeige im Web-Browser oder CSV für den Download, konvertiert (9).

8.4 Metadatenverwaltung

8.4.1 Aufbau und Wirkungsweise der Mapping-Datenbank

Integrationssysteme wie SRS ermitteln korrespondierende Objekte zweier Datenquellen durch eine Mehrwege-Join-Operation entlang des kürzesten Pfades zwischen den entsprechenden Quellen. Dabei sind verschiedene Probleme zu beobachten. Der kürzeste Weg ist nicht immer der sinnvollste für bestimmte Benutzer oder Anwendungen. Das folgt daraus, dass die Semantik des kürzesten Weges sich von der anderer Wege unterscheiden kann. Darüber hinaus kann ein solcher Weg zu Datenqualitätsproblemen führen, wenn der Weg über veraltete Datenquellen führt. Ferner können die Pfade immer noch sehr lang sein, was zu Performanzproblemen bei einer Komposition zur Laufzeit führen kann. Mappings zwischen beliebigen Quellen können zwar zur Performanzoptimierung vorberechnet und materialisiert werden. Jedoch führt diese Vorgehensweise zu einem nicht mehr handhabbaren Datenbestand; es ergibt sich eine Speicherkomplexität von $O(n) = n^2$ Mappings bei n Quellen, wobei jedes dieser Mappings eine Menge von Objektkorrespondenzen besitzt. Diese Probleme werden a) durch die Unterstützung alternativer Pfade, die in Zusammenarbeit mit den Benutzern ausgewählt werden, und b) durch die Vorberechnung der Mappings zu einer zuvor ausgewählten zentralen Quelle adressiert.

Die Datenquellen werden in diesem Ansatz ähnlich wie in Columba [RMT+04] sternförmig (multidimensional) miteinander verbunden. Die Mapping-Datenbank verwendet das in Abbildung 8.4a gezeigte Schema. Es benutzt den Identifikator einer zentralen Quelle (Center_Accession in der Tabelle Center), der mit den einzelnen quellenspezifischen Mapping-Tabellen über den Fremdschlüssel Center_Id verbunden ist. Jede Mapping-Tabelle beinhaltet die Mappings zwischen den Identifikatoren der zentralen Quelle und einer bestimmten Annotationsquelle. Verschiedene Beziehungen zwischen denselben Instanzen der beiden Quellen werden anhand des Mapping-Pfades (Path_Id) unterschieden. Diese Pfade, deren Bezeichnungen sich in



a) Schema der Mapping-Datenbank **b) Schema der ADM-Datenbank**

Abbildung 8.4: Metadatenverwaltung in ADM- und Mapping-Datenbank

der Tabelle PATH der ADM-Datenbank (siehe Abschnitt 5.2) wieder finden, ergeben sich einerseits direkt aus den Mapping-Daten einer Datenquelle und andererseits aus der Komposition von Mappings über verschiedene Datenquellen hinweg. Eine solche Mapping-Komposition wird beim Import der Mapping-Daten ausgeführt und damit vorberechnet in der jeweiligen Mapping-Tabelle gespeichert.

Damit beispielsweise Annotationsdaten aus UniGene und Ensembl miteinander verknüpft werden können, ist ein Mapping zwischen beiden Datenquellen notwendig. Jedoch verfügt weder UniGene noch Ensembl über Korrespondenzen zur jeweils anderen Datenquelle. Damit ist kein direktes Mapping ableitbar und es muss ein Join-Weg, der weitere Quellen einschließt, gewählt werden. Ein möglicher Join-Weg besteht in UniGene → LocusLink → NetAffx → Ensembl. Da im hybriden Integrationssystem LocusLink als zentrale Quelle gewählt wurde, enthält die Mapping-Tabelle CENTER_UNIGENE das direkte Mapping LocusLink → UniGene. Analog dazu existiert die Mapping-Tabelle CENTER_ENSEMBL. Da zwischen LocusLink und Ensembl kein direktes Mapping besteht, beinhaltet diese Mapping-Tabelle vorberechnete Mappings zwischen beiden Quellen (u.a. auch LocusLink → NetAffx → Ensembl), die anhand des Pfades (Join-Weg) unterschieden werden. Durch die sternförmige Anordnung der Mapping-Tabellen innerhalb der Mapping-Datenbank ist die Anzahl der zu materialisierenden Mappings linear beschränkt, auch wenn zusätzliche alternative Mappings unterstützt werden ($k \cdot n$ Mappings bei n Quellen und durchschnittlich k alternativen Mappings pro Quelle). Die Verknüpfung von Objekten zweier beliebiger Datenquellen kann günstig durch die Verknüpfung mit dem zentralen Identifikator durchgeführt werden. Die Auswahl der für die Verknüpfung zu verwendenden Mappings wird durch den Benutzer mit der Angabe des Mapping-Pfades auf der

Web-Oberfläche getroffen.

Eine neue Annotationsquelle kann flexibel hinzugefügt werden, indem mindestens ein Mapping zwischen der neu hinzuzufügenden Datenquelle und der zentralen Quelle erstellt oder abgeleitet wird. Dieses Mapping wird in eine neu zu erstellende Mapping-Tabelle innerhalb der Mapping-Datenbank eingefügt. Dadurch werden Annotationsquellen integriert, ohne dass dies zu einer Erhöhung der Laufzeitkomplexität führt, da der Join-Weg in der Mapping-Datenbank zwischen zwei Quellen nie mehr als 2 beträgt (Source → Center → Source). Umgekehrt können für nicht mehr benötigte Annotationsquellen die jeweiligen Mapping-Tabellen ohne großen Aufwand entfernt werden. Die Speicherung der Mapping-Daten je Datenquelle in eigenen Mapping-Tabellen erleichtert zudem die Administration; insbesondere können die Mappingdaten je nach Aktualisierungszyklen der Quellen rasch erneuert werden. Im Aktualisierungsprozess werden die alten Mapping-Daten gelöscht und anschließend die neuen importiert. Die lokalen Kopien der Annotationsquellen können unabhängig davon erneut repliziert werden.

Die Integration einer neuen Datenquelle setzt voraus, dass mindestens ein Mapping zwischen dem Identifikator der zentralen Quelle und der neuen Quelle existiert, unabhängig davon, ob es ein direktes Mapping ist oder aus einer Komposition abgeleitet und vorberechnet wurde. Deshalb hat die Auswahl der zentralen Quelle und somit die Bildung des zentralen Identifikators maßgeblichen Einfluss auf diese Art der Datenintegration. Neben Aktualität, Redundanzgrad, und Akzeptanz der Objekte ist die Anzahl der in der Quelle des zentralen Identifikators enthaltenen Mappings zu berücksichtigen. Solche können in die Mapping-Datenbank importiert und zur Verknüpfung mit anderen Quellen benutzt werden. Beispielsweise ist LocusLink eine Referenz-Datenquelle für Gendaten, während sich SwissProt für proteinbezogene Daten eignet.

8.4.2 Verwendung der ADM-Datenbank

Abbildung 8.4b zeigt einen Ausschnitt aus dem Schema der ADM-Datenbank, der zur Verwaltung von Metadaten der integrierten Datenquellen verwendet wird. Zur Zeit werden diese Metadaten teilweise manuell, teilweise automatisch durch Datenbankskripte aus den entsprechenden Datenquellen extrahiert und importiert. Die Quellen werden in der Tabelle SOURCE mit einer eindeutigen Nummer (Db_Id) und einem Namen verwaltet. Zusätzlich werden die Attribute der einzelnen Quellen für deren Auswahl auf der Web-Oberfläche gespeichert. Dazu dient die Tabelle ATTRIBUTE, in der die jeweiligen Attribute der Datenquellen mit einer eindeutigen Nummer (Attribute_Id), einem dem Benutzer verständlichen Namen, dem SRS-internen

Bezeichner sowie der zugehörigen Quelle enthalten sind. Die Pfade, die zur Berechnung der in die Mapping-Datenbank importierten Mappings benutzt wurden, beinhaltet die Tabelle PATH. Der Pfadname setzt sich aus den Namen der beteiligten Quellen zusammen, so dass der Benutzer später zwischen alternativen Mappings unterscheiden sowie sich über die Semantik einzelner Mappings informieren kann. Der Abschnitt 8.5.2 diskutiert, wie Web-Oberflächen aus diesen Metadaten für die Anfragespezifikation automatisch generiert werden.

8.5 Anfragebearbeitung im Query-Mediator

8.5.1 Anfragetypen

Ausgehend von den spezifischen Anforderungen der Expressions- und Annotationsanalysen unterscheidet der Query-Mediator derzeit zwei Typen von Anfragen, nämlich Projektions- und Selektionsanfragen:

- **Projektionsanfragen** unterstützen die Expressionsanalyse und erlauben die gemeinsame Darstellung der durch den Benutzer spezifizierten Attribute verschiedener Annotationsquellen für eine Gruppe von Genen.
- **Selektionsanfragen** unterstützen die Annotationsanalyse und suchen Gene mit bestimmten funktionalen Eigenschaften anhand von spezifizierten Filterbedingungen. Das Ergebnis mündet in einer Gengruppe, die dann in GeWare zur Analyse des Expressionsverhaltens weiter verwendet werden kann.

Die zwei Anfragetypen unterscheiden sich nur in den Ein- und Ausgaben. Projektionsanfragen benötigen eine Gengruppe als Eingabe, während Selektionsanfragen eine Gengruppe als Ausgabe erzeugen. Die Verarbeitung ist in beiden Fällen ähnlich und konzentriert sich auf die Assoziation der Gene mit den Attributen aus entsprechenden Datenquellen. Im folgenden Abschnitt wird eine Anfrageformulierung ausführlicher dargelegt; im Anhang C werden die beiden Anfragetypen in Form von Syntaxdiagrammen formaler dargestellt.

8.5.2 Anfrageformulierung

Die Anfragen werden im Web-Browser durch die Auswahl der relevanten Attribute (Projektion) und die Spezifikation der Filterbedingungen (Selektion)

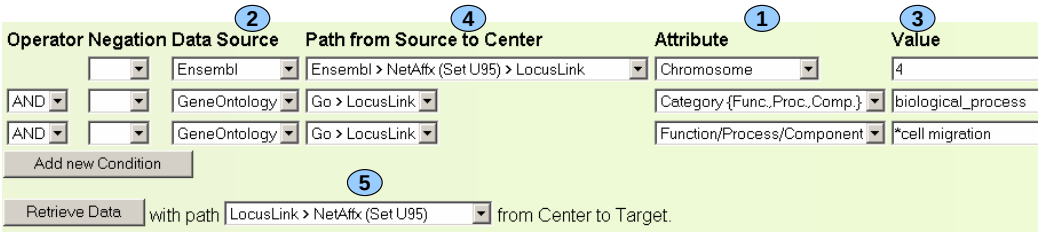


Abbildung 8.5: Anfrageformulierung auf der automatisch generierten Web-Oberfläche

formuliert. Aus den Metadaten der ADM-Datenbank (siehe Abschnitt 5.2) wird die Web-Oberfläche, wie sie die Abbildung 8.5 für eine einfache Selektionsanfrage zeigt, zur Formulierung der Anfrage automatisch generiert. Gesucht sind beispielsweise alle Gene der Datenquelle NetAffx (Set U95), d.h. Gene des Microarray-Sets U95, die bestimmte Eigenschaften besitzen: sie befinden sich auf dem Chromosom *vier* und sind mit dem biologischen Prozess *cell migration* assoziiert.

Auf der Web-Oberfläche können beliebig viele Filterbedingungen zur Selektion bzw. Attribute zur Anzeige von Annotationen zu den Genen spezifiziert werden. Die Bedingungen bestehen aus einem Attribut (1), einer Datenquelle (2), für das ein Filterwert (3) für eine exakte oder Ähnlichkeitssuche angegeben werden kann. Ferner ist ein Mapping zwischen der gewählten Datenquelle und der zentralen Quelle, hier LocusLink, durch einen entsprechenden Pfad (4) auszuwählen. Die einzelnen Bedingungen können beliebig mit den logischen Operatoren OR, AND und NOT kombiniert werden, wobei OR die niedrigste und NOT die höchste Priorität bei der Evaluation der Anfrage hat. Letztlich ist ein Mapping von der zentralen Quelle zur Zielquelle, im Beispiel NetAffx (Set U95), durch einen entsprechenden Pfad festzulegen (5).

Unabhängig von der Zielquelle, von der die relevanten Objekte in der Ergebnismenge enthalten sein sollen, können beliebige Attribute verschiedener Datenquellen abgefragt werden. Dies ist eine wichtige Verbesserung gegenüber SRS, das nur Filterbedingungen für Attribute einer einzelnen Quelle unterstützt. Ferner wird auch die Projektion verschiedener Attribute aus unterschiedlichen Quellen in einer gemeinsamen Anfrage unterstützt, was mit SRS in dieser flexiblen Art noch nicht möglich ist. Eine weitere Abgrenzung zwischen dem hybriden Ansatz unter Verwendung der Mapping-Datenbank und SRS wird in Abschnitt 11.4 diskutiert.

8.5.3 Generierung des Anfrageplans

Anhand der Benutzerspezifikation auf der Web-Oberfläche (siehe Abbildung 8.5) generiert der Query-Mediator eine SRS-spezifische Abfrage, welche in Abbildung 8.6 (Schritt 3) gezeigt wird. Diese Abfrage wird anschließend durch SRS abgearbeitet, wobei deren Performanz vom Anfrageplan abhängt. Der Query-Mediator optimiert bei der Erstellung der SRS-spezifischen Abfrage den Anfrageplan mit den folgenden Schritten:

1. **Blockbildung**: Die Filterbedingungen zur Selektion werden in Hinsicht auf den logischen Operator OR in einzelne Blöcke unterteilt. Innerhalb dieser Blöcke können anschließend Optimierungen der Abfrage durchgeführt werden. In unserem Beispiel sind die einzelnen Filterbedingungen einheitlich durch den logischen Operator AND verknüpft. Deshalb wird, wie in Abbildung 8.6 (Schritt 1) zu sehen, nur ein einziger Block gebildet, der aus den drei Attributen *Chromosome*, *Category* und *Process* besteht. Im Gegensatz dazu bilden die ausgewählten Attribute zur Projektion immer einen Block.

2. **Zusammenfassung quellenspezifischer Attribute**: Für jeden resultierenden Block werden die Attribute und Filterbedingungen nach Datenquellen sowie nach den zu verwendenden Mappings sortiert. Damit ist es möglich, Attribute bzw. Filter, die dieselbe Datenquelle und dasselbe Mapping benutzen, so zusammenzufassen, dass sie in einer gemeinsamen Anfrage an die Datenquelle verwendet werden können. Im Beispiel stammen die beiden Attribute *Category* und *Process* aus der Quelle GeneOntology und benutzen dasselbe Mapping zum zentralen Identifikator. Abbildung 8.6 (Schritt 2) zeigt die Zusammenfassung dieser beiden Attribute; das Attribut *Chromosome* der Quelle Ensembl wird nicht mit anderen Attributen zusammengefasst.

3. **Zusammensetzung der SRS-Abfrage(n)**: Die Namen der Quellen und Attribute werden durch die internen SRS-Bezeichner ersetzt. Die ausgewählten Mappings werden mit den Identifikatoren aus der Mapping-Datenbank versehen. Im Beispiel ist die zweite und dritte auf der Web-Oberfläche spezifizierte Filterbedingung (siehe Abbildung 8.5) in der Zeile 3 der in Abbildung 8.6 (Schritt 3) gezeigten SRS-Abfrage zu sehen. Die Quelle GeneOntology sowie die Attribute *Category* und *Process* wurden in die SRS-internen Namen *GoTerm* und *typ* bzw. *tna* übersetzt. Ebenfalls wurde die Nummer des Mappings (Nummer 1 im Beispiel) zwischen GeneOntology und LocusLink identifiziert. Die Anfrage kann nun zur Abarbeitung an SRS durch den Aufruf des Interpreters "getz" geschickt werden.

1. Schritt: Blockbildung

Block	*Pfad*	*Quelle*	*Attribut*	*Filterwert*
1	Ensembl>NetAffx(Set U95)>LocusLink	Ensembl	Chromosome	4
1	GeneOntology>LocusLink	GeneOntology	Category	biological_process
1	GeneOntology>LocusLink	GeneOntology	Process	*cell migration

2. Schritt: Zusammenfassung quellspezifischer Attribute

Block	*Zsfg.*	*Pfad*	*Quelle*	*Attribut*	*Filterwert*
1	a	Ensembl>NetAffx(Set U95)>LocusLink	Ensembl	Chromosome	4
1	b	GeneOntology>LocusLink	GeneOntology	Category	biological_process
1	b	GeneOntology>LocusLink	GeneOntology	Process	*cell migration

3. Schritt: Zusammensetzung der SRS-Abfrage

```
1 getz -vf "accession" "([Mapping-pid:5]
2   < (Center < ([Mapping-pid:2]<([EnsemblGene-cnm:4]))
3   < ([Mapping-pid:1]<([GoTerm-typ: biological_process] & [GoTerm-tna:*cell migration])))
```

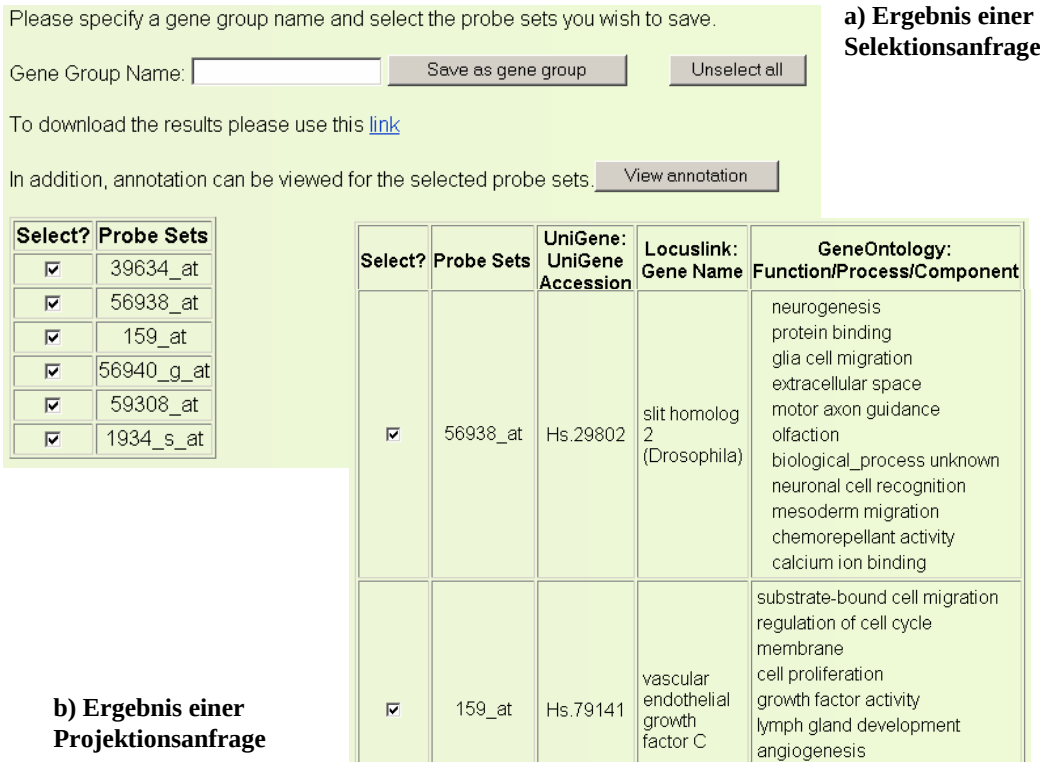
Abbildung 8.6: Schritte zur Erstellung des Anfrageplans

Aus der SRS-Abfrage in Abbildung 8.6 (Schritt 3) geht hervor, dass die Objekte von EnsemblGene bzw. GoTerm zuerst mit den jeweiligen Filtern selektiert (Zeilen 2 und 3) und anschließend auf die Center_Id (zentraler Identifikator) abgebildet werden (Zeile 2). Die resultierenden zentralen Identifikatoren werden anschließend unter Nutzung des Mappings zwischen der zentralen Quelle und der Zielquelle (Mapping Nr. 5, Zeile 1) den Objekten der Datenquelle NetAffx zugeordnet. Im Ergebnis wird eine Liste von Accessions (Identifikatoren) der entsprechenden NetAffx-Objekte zurückgeliefert.

8.5.4 Extraktion und Transformation der Ergebnisse

SRS liefert als Antwort für jede Abfrage (Aufruf des "getz" Interpreters) einen XML-Datenstrom zurück, aus dem die notwendigen Daten durch den Query-Mediator extrahiert werden. Komplexe Anfragen, wie z.B. Projektionen mit vielen Attributen aus unterschiedlichen Datenquellen, können in mehrere SRS-Abfragen unterteilt werden, woraus ebenso viele XML-Datenströme als Antwort resultieren. Die extrahierten Daten werden anschließend im Query-Mediator zusammengesetzt. Ferner werden fehlende Anfragefunktionen einiger Quellen, die in SRS noch nicht berücksichtigt sind, wie z.B. der Mengendurchschnitt in MySQL, durch zusätzliche Transformationen im Query-Mediator übernommen. Das Ergebnis wird anschließend in das HTML-Format konvertiert und ausgegeben. Eine Gengruppe als Ergebnis einer Anfrage kann als Eingabe neuer Anfragen verwendet werden. Ebenso kann das Ergebnis einer Projektionsanfrage um weitere Attribute von verschiedenen Datenquellen erweitert werden. Dies ermöglicht eine iterative Analyse.

Die Abbildung 8.7a zeigt das Ergebnis der in Abschnitt 8.5.2 eingeführten Selektionsanfrage, die eine Menge von Genen (Affymetrix Probesets) iden-



a) Ergebnis einer Selektionsanfrage

b) Ergebnis einer Projektionsanfrage

Abbildung 8.7: Ergebnisse für Projektions- und Selektionsanfragen

tifiziert, die auf dem Chromosom *vier* lokalisiert und mit dem biologischen Prozess *cell migration* entsprechend der GeneOntology Klassifikation assoziiert sind. GeWare bietet die Möglichkeit, eine benutzerspezifische Auswahl dieser Gene in einer Gengruppe unter Angabe eines Gruppennamens abzuspeichern. Mit dieser Gruppe kann anschließend, z.B. eine Projektionsanfrage, durchgeführt werden, deren Ergebnis in Abbildung 8.6b gezeigt ist. Insbesondere wurden die korrespondierenden Identifikatoren von UniGene, die Gennamen von LocusLink sowie die assoziierten Funktionsnamen von GeneOntology abgefragt. Interessante Gene können hier ebenfalls ausgewählt und in einer neuen Gengruppe für weiterführende Analysen abgelegt werden.

8.6 Ausgewählte Performanzanalysen

Im folgenden Abschnitt werden zwei ausgewählte Analysen vorgestellt, die zur Abschätzung des Performanzverhaltens der Integrationslösung dienen.

8.6.1 Testumgebung

Den Performanzmessungen lag eine Intel-Server-Plattform mit der folgenden Konfiguration als Testumgebung zugrunde.

Hardware	
CPU:	4 x Intel Xeon 2.5 GHz
Hauptspeicher:	8 GB DDR-RAM
Software	
Betriebssystem:	Linux, Fedora 2.4.22
Datenbanken:	IBM DB2 8.1.0
	MySQL, Version 4.0.17-max
SRS-Server:	SRS Relational 7.3.1 für Linux
Java:	Java 2 SUN-Plattform, Standard Edition Version 1.4.2

Das Data Warehouse wie auch die ADM- und die Mapping-Datenbank benutzen das relationale System DB2 von IBM in der angegebenen Version. Die Programmlogik des *GeWare*-Systems sowie des Query-Mediators wurden auf Basis der Java 2 SUN-Plattform implementiert. In SRS wurden die Mapping-Datenbank (IBM DB2, RDBMS) sowie die Annotationsquellen LocusLink (Datei), Ensembl (MySQL, RDBMS) und GeneOntology (MySQL, RDBMS) integriert.

Da *GeWare* von den Benutzern in unregelmäßigen Abständen genutzt wird, verteilt sich die Belastung sehr unterschiedlich zwischen Spitzenzeiten und Ruhephasen. Alle Tests wurden unter geringer Belastung ausgeführt und unter der Gewährleistung von mindestens 90% freier Prozessorkapazität.

8.6.2 Messergebnisse und Interpretation

Aus einer Reihe von Performanzmessungen fokussieren wir hier auf zwei Auswertungen, die die Abarbeitungszeiten von Anfragen in Abhängigkeit zur Anzahl resultierender Datensätze untersuchen. Die Abbildungen 8.8 und 8.9 zeigen die Ergebnisse der ersten Messreihe. Diese Messreihe soll klären, inwieweit Performanzeinbußen von SRS gegenüber RDBMS, wie beispielsweise MySQL, bei der Selektion und Projektion auftreten. Dazu wurden jeweils 15 Anfragen an die lokale Kopie der Annotationsquelle Ensembl, deren Daten in einer relationalen Datenbank MySQL vorliegen, untersucht. Alle Anfragen benutzen einheitlich das Attribut *des* (Genbeschreibung), unterscheiden sich jedoch im angegebenen Filterwert und dadurch in der Anzahl der resultierenden Datensätze. Darüber hinaus wird zwischen Selektionsanfragen und einer Kombination von Selektion und Projektion unterschieden. Während erstere lediglich den Identifikator von Ensembl zurückgeben, liefern letztere zusätzlich zum Identifikator das gefilterte Attribut zurück. Jeder Mess-

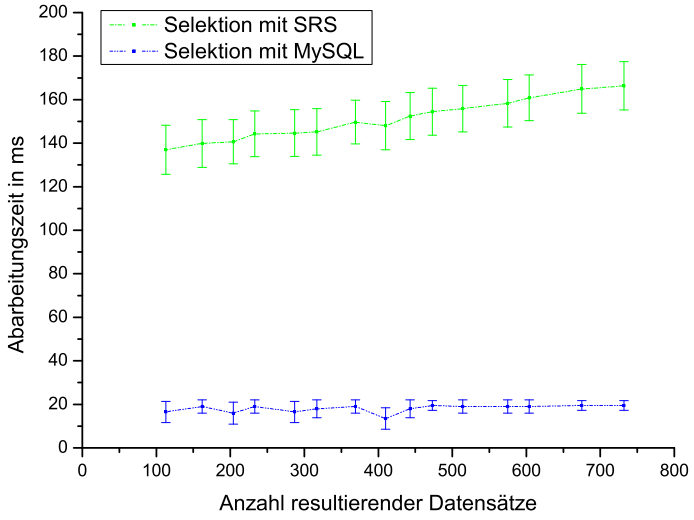


Abbildung 8.8: Performanz von Selektionsanfragen in SRS und MySQL

punkt repräsentiert den Mittelwert aus 20 Wiederholungen einer Anfrage; ein Zwischenspeichern (Cache) der Ergebnisse wurde ausgeschlossen. Die Standardabweichung ist zu jedem Messpunkt als Fehlerbalken aufgetragen. Jede Abarbeitungszeit wurde unabhängig von der Web-Oberfläche evaluiert, um störende Faktoren weitestgehend auszuschließen und um eine Vergleichbarkeit herzustellen.

Die Abbildungen 8.8 und 8.9 stellen die Abarbeitungszeiten für die untersuchten Anfragearten, die Selektion sowie die Kombination von Selektion und Projektion, jeweils für SRS und MySQL in Abhängigkeit zur Größe der Ergebnismenge dar. Dabei zeigt sich, dass sowohl SRS als auch MySQL sehr geringe Abarbeitungszeiten (< 200 ms) zur Selektion benötigen, die mit zunehmender Anzahl an resultierenden Datensätzen nur schwach linear ansteigen (siehe Abbildung 8.9). Auf die Abarbeitungszeiten für die Kombination von Selektion und Projektion in MySQL trifft dies ebenso zu. Dagegen führt die Kombination von Selektion und Projektion in SRS zu einem starken Anstieg der Abarbeitungszeit, die zudem mit dem Umfang der Ergebnismenge linear zunimmt. Der Grund hierfür liegt in der Art und Weise, wie SRS Anfragen an relationale Datenbanken verarbeitet. Für diese Art von Quellen verwendet SRS so genannte Hub-Tabellen, die mit zusätzlichen Annotationstabellen der Quelle verbunden sind. SRS führt zuerst die Selektion aus und liefert anschließend die relevante Objektmenge zurück. Da die Objekt-

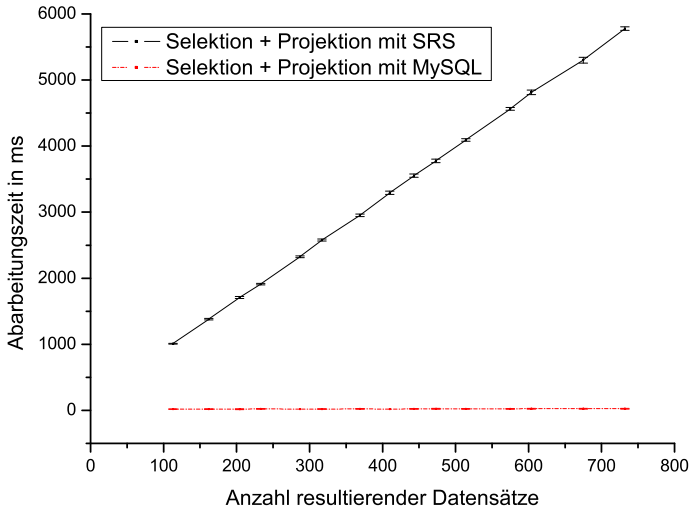


Abbildung 8.9: Performanz von Projektions- und Selektionsanfragen in SRS und MySQL

Identifikatoren in der Hub-Tabelle enthalten sind bildet SRS den kürzesten Weg zwischen der Annotationstabelle, die für die Selektion benutzt wurde, und der Hub-Tabelle. Für die anschließende Projektion verbindet SRS wiederum die relevanten Annotationstabellen, aus der die benötigten Attribute stammen, mit der Hub-Tabelle. Dazu werden die zwischengespeicherten Objektidentifikatoren zur Selektion in der Hub-Tabelle verwendet. Dieser zweistufige Prozess [EHB03] erklärt die deutlich schlechtere Performanz von SRS gegenüber MySQL bei einer Kombination von Projektion- und Selektionsanfragen.

Abbildung 8.10 stellt die Ergebnisse der zweiten Messreihe zur Performanzbewertung der hybriden Integrationslösung dar. Diese basieren auf 11 Selektionsanfragen an die Annotationsquelle Ensembl, für die die Abarbeitungszeiten nach jedem einzelnen Schritt des Anfrageplans (Selektion Ensembl, Mapping Ensembl → LocusLink, Mapping LocusLink → NetAffx, Projektion des NetAffx Identifikators) gemessen wurden. Die Anfragen verwenden einheitlich das Attribut *des* (Genbeschreibung) der Datenquelle Ensembl zur Selektion mit verschiedenen Filterwerten. Wie bei der ersten Messreihe repräsentiert jeder Messpunkt den Mittelwert aus 20 Wiederholungen, wobei die Abarbeitungszeiten unabhängig von der Web-Oberfläche evaluiert wurden. Die Fehlerbalken charakterisieren die Standardabweichung.

Abbildung 8.10 stellt die Abarbeitungszeiten für einzelne Schritte des An-

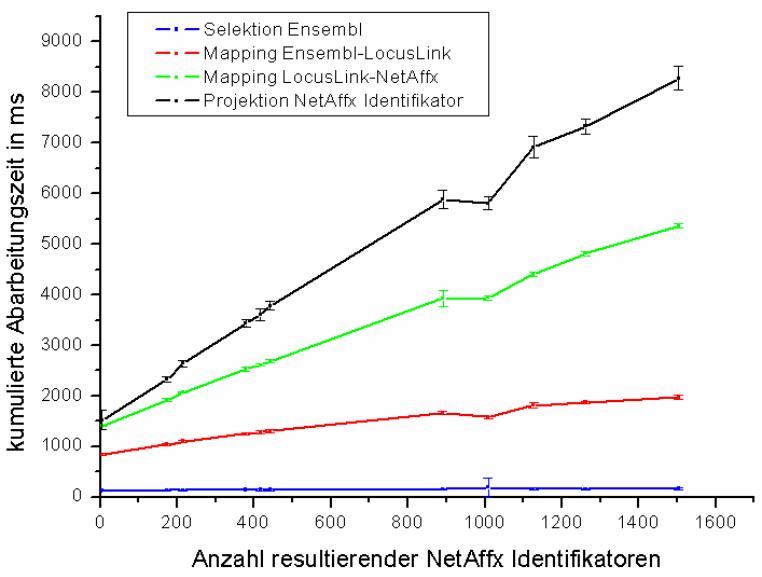


Abbildung 8.10: Performanz von Selektionsanfragen an die Datenquelle Ensembl

frageplans kumulativ dar. Das bedeutet, die Abarbeitungszeit eines betrachteten Schrittes beinhaltet bereits die Zeiten aller vorherigen Schritte und die Gesamtzeit ergibt sich mit der Projektion der NetAffx Identifikatoren. Insgesamt steigt die Bearbeitungszeit wiederum linear mit der Anzahl der Ergebnissätze und liegt für typische Größenordnungen in einem akzeptablen Bereich. Die Selektion der Ensembl-Identifikatoren benötigt die kürzeste Zeit. Dagegen ist die Abbildung dieser Ergebnismenge auf die Mapping-Datenbank noch sehr zeitaufwendig und deutet auf ein Potential zur Performanzverbesserung. Gegenwärtig wird dieser Schritt bzw. alle mit der Mapping-Datenbank korrespondierenden Schritte durch SRS übernommen. Einerseits soll in der neuen SRS Version 8 die Anbindung von relationalen Datenquellen bedeutend verbessert worden sein (allerdings nicht für DB2). Andererseits könnten sich weitere Verbesserungen erzielen lassen, wenn der Query-Mediator direkt auf die Mapping-Datenbank über ein definiertes API zugreift, um die Mapping-Komposition Datenquelle → LocusLink → Datenquelle zur Laufzeit in einem Schritt auszuführen.

8.7 Zusammenfassung

Im Mittelpunkt dieses Kapitels stand ein hybrider Integrationsansatz, um Annotationsdaten von molekularbiologischen Objekten wie Genen, Proteinen

und Pathways aus öffentlichen Datenquellen für datenintensive Expressionsanalysen in *GeWare* verwendbar zu machen. Während die experimentellen Daten physisch in *GeWare* integriert sind, um schnelle Auswertungen zu unterstützen, werden die öffentlichen Annotationsdaten virtuell über einen Mediatoransatz integriert und bedarfsgesteuert für Analysen abgerufen. Für die einheitliche Anbindung der Datenquellen wird das verbreitete Tool SRS (Sequence Retrieval System) genutzt. Die Kopplung zwischen *GeWare* und SRS erfolgt über den Query-Mediator. Die vorhandenen Mappings, die zwischen den einzelnen Datenquellen bestehen, werden aus den Quellen extrahiert und in einer eigenen Datenbank zur Integration der Annotationsdaten gespeichert. Dieser hybride Integrationsansatz wird für die Einbindung verschiedener öffentlicher Datenquellen in Annotations- und Expressionsanalysen eingesetzt. Die ausgewählten Performanzanalysen zeigen die Praktikabilität des Ansatzes, jedoch auch die Abhängigkeit von der Bearbeitungsgeschwindigkeit von SRS. Zwar ist die Integrationslösung nicht auf das Anwendungsfeld der Genexpression beschränkt, wurde aber hauptsächlich in diesem Bereich verwendet und konnte die dort gestellten Anforderungen abdecken.

Kapitel 9
Semantische Peer-to-Peer-artige Datenfusion: Der iFuice-Ansatz

9.1 Motivation

Mit der hybriden Integrationsform im vorangegangenen Kapitel wird eine flexible Anbindung und Integration von frei verfügbaren Datenquellen erreicht. Jedoch wird oftmals eine Kombination von Daten aus privaten und frei verfügbaren Datenquellen sowie Ontologien notwendig, deren Integration auf Basis des vorgestellten Ansatzes noch zu statisch ist. Während SRS für viele frei verfügbaren Datenquellen Wrapper zur Verfügung stellt, sind diese für private Quellen neu zu erstellen. Darüber hinaus verfolgt der Integrationsansatz keine semantische Integration; sie verbleibt im Verantwortungsbereich des Benutzers.

Eine semantische Integration ist das Ziel vieler traditioneller Datenintegrationsansätze, wie dem Data Warehousing [Inm92, JLVV03] und Mediatoren [Wie92] (vgl. Kapitel 2). Diese Ansätze sind zwar oft anwendbar, benötigen aber eine lange Entwicklungszeit und unterstützen evtl. nur unzureichend eine explorative Analyse der Daten. Typischerweise wird die semantische Integration auf der Grundlage eines einheitlichen globalen Schemas erreicht, das eine konsistente Sicht auf die Daten aus unterschiedlichen Quellen zur Verfügung stellt. Jedoch geht die Erstellung eines solchen globalen Schemas

mit einem immensen personellen Aufwand einher, wenn mehr als ein paar Quellen integriert werden sollen, da sich diese in der Bioinformatik durch eine hohe Diversität, Komplexität sowie häufige Änderungen auszeichnen (vgl. Kapitel 1). Jede zusätzliche neue Quelle, die integriert werden soll, bedingt evtl. sowohl die Adaption des globalen Schemas als auch die Anwendungen, die auf Basis des globalen Schemas und der integrierten Daten agieren.

Eine Alternative zu den traditionellen Data-Warehousing- und Mediator-Lösungen unter Nutzung eines globalen Schemas stellen die so genannten Peer Data Management Systeme (PDMS) [BGK+02, HIMT03] dar. In Kapitel 2 wurden die Grundzüge solcher Systeme vorgestellt: Sie verwenden oftmals bidirektionale Mappings, die die autonomen Datenquellen (Peers) miteinander verknüpfen, anstatt die Datenquellen auf das erstellte globale Schema abzubilden. Für die Integration einer neuen Datenquelle wird lediglich ein Mapping notwendig, das die neue mit einer bereits integrierten Quelle verbindet. Damit wird der Aufwand vermieden, der bei den traditionellen Ansätzen für die Adaption des globalen Schemas sowie der Erstellung des Mappings zwischen dem Schema der neuen Quelle und dem globalem Schema notwendig ist.

Ein Ansatz, der der Peer-to-Peer Integration folgt, ist *iFuice* (information fusion using instance correspondences and peer mappings) [RTA+05]. *iFuice* verwendet Mappings, d.h. Mengen von Objektkorrespondenzen, um Informationen von verschiedenen Quellen zu kombinieren oder zu fusionieren. Dazu werden die Quellen und Mappings einem Domänenmodell zugeordnet, das eine semantische Integration und Fusion der Daten unterstützt. In die *iFuice*-Architektur[55] ist ein Mapping-Mediator eingebettet, mit dem sowohl ein interaktiver als auch skriptgetriebener, workflowartiger Zugriff auf die Quellen möglich ist und die Ausführung von Mappings durchgeführt wird. *iFuice* ist ein generischer Integrationsansatz; der Ansatz ist nicht auf die Bioinformatik beschränkt, sondern kann in verschiedenen Domänen angewendet werden.

Gegenstand dieses Kapitels ist eine Darstellung des *iFuice*-Konzepts, das Grundlage der domänenspezifischen Anwendung *BioFuice* im Bereich der Bioinformatik (Kapitel 10) ist. Details zum *iFuice*-Konzept werden in [Tho07] erläutert.

9.2 Ein beispielhaftes Szenario

Zur Illustration des Integrationsansatzes dient das folgende Beispiel, das die Abbildung 9.1 zeigt. In der Ausgangssituation liegt eine Menge von menschli-

[55] Die Darstellung der *iFuice*-Architektur erfolgt zusammen mit der für die Bioinformatik spezifischen Erweiterung *BioFuice*.

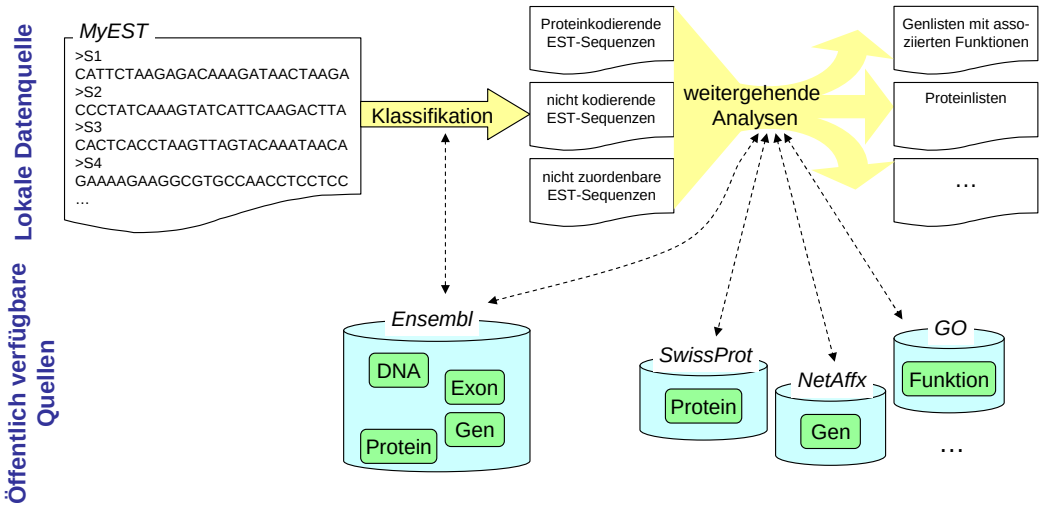


Abbildung 9.1: Beispielhaftes Analyseszenario im Bereich der Bioinformatik

chen EST-Sequenzen vor, die in der lokalen, privaten Datenquelle *MyEST* in Form einer Fasta-Datei[56] gespeichert sind. EST (engl. Expressed Sequence Tags) sind typischerweise kurze DNA-Sequenzen eines speziellen Organismus, die aus einem Sequenzierungsverfahren, z.B. der *shot gun* Sequenzierung, resultieren. Für eine spätere fokussierte Analyse spezieller EST-Sequenzen besteht die Aufgabe, die gegebene Menge an EST-Sequenzen in drei Klassen einzuteilen. Die Klassen sind durch die drei folgenden Fragen charakterisiert.

Frage 1 (Q1): *Für welche der gegebenen EST kann keine passende DNA-Sequenz im Genom gefunden werden?*

Frage 2 (Q2): *Welche der gegebenen EST sind mit proteinkodierender DNA assoziiert?*

Frage 3 (Q3): *Welche der gegebenen EST können mit einer DNA-Sequenz im Genom assoziiert werden, die jedoch nicht proteinkodierend sind?*

Die Klassifikation basiert auf einem Vergleich (in der Bioinformatik auch als Alignment bezeichnet) zwischen den EST-Sequenzen und der DNA-Sequenz des menschlichen Genoms. Solche Sequenzen stehen in umfangreichen, öffentlich zugänglichen Datenquellen, z.B. Ensembl [HAC+05], NCBI Entrez Gene [MOPT05] und UCSC Genome Browser [KSF+02, KBD+03], für unterschiedliche Spezies zur Verfügung. Das Sequenz-Alignment wird notwendig,

[56] **Fasta** ist ein in der Bioinformatik häufig verwendetes Format, um Sequenzdaten in Dateien zu speichern (vgl. [Rau01]).

da die gegebenen EST-Sequenzen lediglich einem selbst generierten Identifikator zugeordnet sind, jedoch die exakte Lokation im Genom fehlt, aus der die Klassifikation abgeleitet werden kann. Die Klassifikation ist Ausgangspunkt für vielfältige Analysen, die die Integration weiterer öffentlich verfügbarer oder privater Datenquellen verlangen. Die Zielstellung solcher Analysen werden beispielhaft anhand der folgenden Fragestellungen charakterisiert.

Frage 4 (Q4): *Welche Proteine sind zu den proteinkodierenden EST-Sequenzen assoziiert?*

Frage 5 (Q5): *Welche Gene (Probesets) korrespondieren zu den proteinkodierenden EST?*

Die Klassifikation der EST-Sequenzen sowie die darauf aufbauenden Analysen stellen ein typisches Datenintegrationsproblem dar, da die gegebene Menge von EST-Sequenzen mit anderen molekularbiologischen Daten über Gene und Proteine aus anderen Quellen kombiniert werden müssen. Dies ist Grundlage, um einerseits zu entscheiden, welche EST-Sequenz welcher der vorgenannten Klassen zugeordnet wird, und andererseits den beispielhaft aufgeführten oder sich ad-hoc ergebenden Fragestellungen nachzugehen.

9.3 Mappings und Mapping-Erzeugung

Mappings nehmen eine zentrale Rolle bei einer Datenintegration mit dem *iFuice*-Ansatz ein. Ein Mapping umfasst eine Menge von Korrespondenzen zwischen den Objekten zweier Datenquellen (vgl. Kapitel 2 und 8). Im Bereich der Bioinformatik sind diese Objektkorrespondenzen vielfach in Form von Querverweisen oder Web-Links in den Datenquellen gespeichert. In Kapitel 8 wurde bereits ein Beispiel (vgl. Abbildung 8.1) für Mappings auf Basis von Web-Links gezeigt. Vielfach werden diese Web-Links auf Basis von auf den Webseiten spezifizierten Abfragen angezeigt. Darüber hinaus kann oftmals über ein zur Verfügung gestelltes Datenbank-API auf die Daten zugegriffen werden, in dem in Form von standardisierten Sprachen (z.B: SQL, XQuery) Abfragen formuliert werden. Andere Mappings sind wiederum in XML- und CSV-Dateien enthalten, auf die einfach zugegriffen werden kann. Alternativ können Mengen von Objektkorrespondenzen und damit Mappings unter Nutzung von Programmen erzeugt werden. Die Eingabedaten, die die Programme verarbeiten, können von *iFuice* zur Verfügung gestellt werden; evtl. erzeugte Ergebnisdaten der Programme können in *iFuice* in weiteren Schritten verwendet werden. Um dies zu ermöglichen, müssen die Ausgabedaten der verwendeten Programme ein Mapping umfassen, in dem die Objekte der Eingabemenge zu Objekten der Ergebnismenge zugeordnet sind.

In dem oben beschriebenen Szenario enthält die lokale Datenquelle MyEST lediglich EST-Sequenzen, denen jeweils ein technisch (selbst-) generierter Identifikator (S1, S2, ...) zugeordnet wurde. Ein Mapping, das den EST-Sequenzen (Datenquelle MyEST) die korrespondierenden DNA-Abschnitte (Datenquelle Ensembl) im menschlichen Genom zuordnet, ist weder in der Datenquelle MyEST noch in der Datenquelle Ensembl vorhanden. Es kann unter Nutzung eines BLAST (-ähnlichen) Softwaretools [AGM+90] erzeugt werden, das eine Abbildung der EST-Sequenzen auf die Genom-Sequenz (Sequenz-Alignment) vornimmt. Die Mapping-Erzeugung kann dabei zur Laufzeit von *iFuice* oder separat erfolgen. Im ersten Fall ist es notwendig, die Eingabedaten (hier: EST-Sequenzen) in einem bestimmten Format dem Programm zur Verfügung zu stellen, das Programm auszuführen und anschließend das Ergebnis für eine mögliche Weiterverarbeitung zu parsen und zu importieren. Im zweiten Fall wird die Datenbereitstellung und Programmausführung weitestgehend manuell ausgeführt; für *iFuice* werden lediglich die Ergebnisdaten in Form des materialisierten Mappings zur Verfügung gestellt.

Allgemein verwendet *iFuice* verschiedene Ausführungsdienste (vgl. Abbildung 10.1 in Kapitel 10), um auf Mappingdaten zuzugreifen und zu nutzen. Vor allem die Nutzung standardisierter Abfragesprachen, z.B. SQL und XQuery, machen eine sonst mühsame Entwicklung von quellenspezifischen Wrappern unnötig. Damit können neue Quellen einfach und schnell integriert werden.

9.4 Konzeptuelle Strukturen

Die Abbildung 9.2a zeigt vier physische Datenquellen und assoziierte Mappings, die für das aufgezeigte Datenintegrationsszenario verwendet werden. Dazu zählen die drei öffentlich verfügbaren Datenquellen *Ensembl*, *NetAffx* [LLS+03, CST+04] und *SwissProt* [BA00] sowie die lokale Quelle *MyEST*, die die Menge an EST-Sequenzen enthält. Diese Quellen werden als physische Datenquellen bezeichnet.

Definition (Physische Datenquelle)**.** *Physische Datenquellen (PDS* [57]*) sind lokale, private oder frei verfügbare Datenquellen und Ontologien, die unter Nutzung eines großen Spektrums an Formaten gespeichert sind. Auf physische Datenquellen kann rechnergestützt zugegriffen werden.*

Eine physische Datenquelle kann Objekte mehrerer unterschiedlicher Typen beinhalten. Beispielsweise beinhaltet die Datenquelle *Ensembl* neben Genen auch Exons und Objekte anderer Typen (vgl. Abbildung 9.2a). Die Daten

[57] Das Akronym PDS steht für die engl. Bezeichnung *Physical Data Source*.

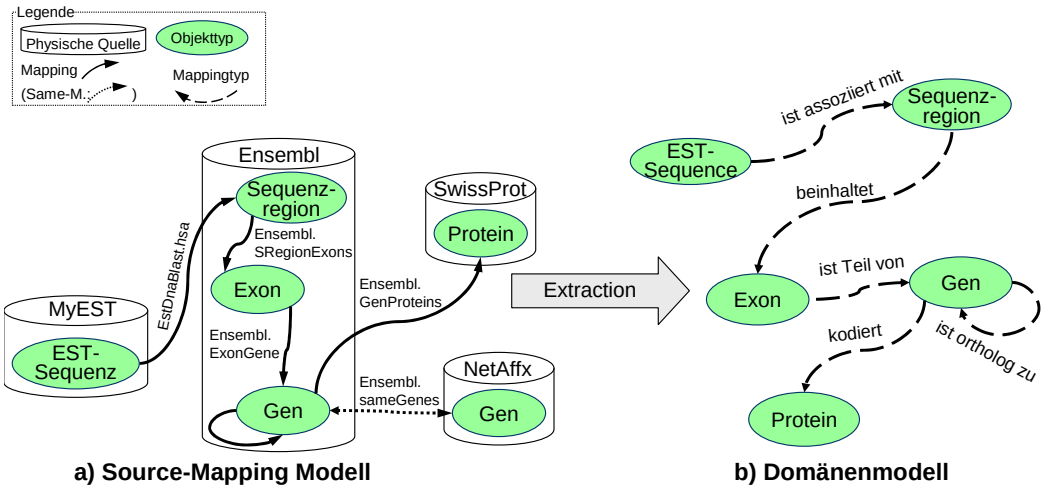


Abbildung 9.2: Das Source-Mapping-Modell und das semantische Domänenmodell für das dargestellte Szenario

eines solchen Objekttyps einer PDS werden als logische Datenquelle zusammengefasst.

Definition (Logische Datenquelle)**.** *Eine logische Datenquelle (LDS* [58]*) ist der Teilbereich einer physischen Quelle, der Objekte eines Typs repräsentiert. Der Objekttyp spiegelt die Semantik der Objekte wider.*

Die Notation *<Objekttyp>@<Physische Datenquelle>* wird verwendet, um spezifische LDS, z.B. Gen@Ensembl oder Protein@SwissProt, zu kennzeichnen. Jede LDS besitzt eine Menge an Attributen, zu denen Daten existieren können, anhand derer die Objekte beschrieben werden. Ein Attribut aus dieser Menge ist das identifizierende Attribut (ID Attribut).

Ein Mapping assoziiert die Objekte von zwei LDS und verbindet somit auf abstrakter Ebene zwei LDS. Sowohl jedes Mapping als auch jede LDS ist durch einen eindeutigen Namen gekennzeichnet. Zusammen bilden Mappings und LDS das so genannte *Source-Mapping-Modell* (SMM), das in Abbildung 9.2a für das oben beschriebene Integrationsszenario beispielhaft illustriert wird.

Definition (Source-Mapping-Modell)**.** *Ein Source-Mapping-Modell (SMM) ist ein gerichteter, beschrifteter Graph*

$$\overrightarrow{G_{SMM}} = (\Sigma_{LDS}, \Sigma_{Map}, V_{LDS}, E_{Map}, L_{LDS}, L_{Map}),$$

[58] Das Akronym LDS steht für die engl. Bezeichnung *Logical Data Source*.

in dem die Menge der Knoten V_{LDS} die logischen Datenquellen eines Integrationsszenarios repräsentieren. Jeder Knoten (LDS) $v_{LDS} \in V_{LDS}$ besitzt auf Basis der bijektiven Abbildung $L_{LDS} : \Sigma_{LDS} \rightarrow V_{LDS}$ einen eindeutigen Namen $\sigma_{LDS} \in \Sigma_{LDS}$. Die Kanten E_{Map} bilden eine Menge geordneter Paare von LDS und entsprechen den Mappings im Integrationsszenario. Jedes Mapping $e_{Map} \in E_{Map}$ trägt auf Grund der bijektiven Abbildung $L_{Map} : \Sigma_{Map} \rightarrow E_{Map}$ einen eindeutigen Namen $\sigma_{Map} \in \Sigma_{Map}$.

$\overrightarrow{G_{SMM}}$ wird zu einem Multigraph $\overrightarrow{G'_{SMM}}$, wenn mindestens zwei Mappings $e_a = (v, w, \sigma_a) \in E_{Map}$ und $e_b = (s, t, \sigma_b) \in E_{Map}$ $(v, w, s, t \in V_{LDS})$ dasselbe geordnete Paar zweier LDS ($v = s$ und $w = t$) repräsentieren, deren Namen $\sigma_a, \sigma_b \in \Sigma_{Map}$ sich unterscheiden ($\sigma_a \neq \sigma_b$).

Eine LDS (Mapping) ist mit einem Objekttyp (Mappingtyp) assoziiert. Dagegen kann ein Objekttyp (Mappingtyp) mehreren LDS (Mappings) zugeordnet sein. Objekt- und Mappingtypen repräsentieren die Semantik der LDS und Mappings anhand von Bezeichnungen wie *Gen*, *Protein* (beides Objekttypen), *homologe Gene* und *Proteinfunktionen* (beides Mappingtypen). Zusammen bilden Objekt- und Mappingtypen das abstrakte *Domänenmodell* eines Integrationsszenarios. Die Abbildung 9.2b zeigt ein solches Domänenmodell für das oben beschriebene Integrationsszenario.

Definition (Domänenmodell)**.** *Ein Domänenmodell (DM) ist ein gerichteter, beschrifteter Graph*

$$\overrightarrow{G_{DM}} = (\Sigma_{OT}, \Sigma_{MT}, V_{OT}, E_{MT}, L_{OT}, L_{MT}).$$

Die Menge der Knoten V_{OT} umfasst alle Objekttypen eines Integrationsszenarios, die auf Basis der bijektiven Abbildung $L_{OT} : \Sigma_{OT} \rightarrow V_{OT}$ eindeutig bezeichnet sind. Die Mappingtypen eines Integrationsszenarios stellen die Menge der Kanten E_{MT} dar, die je zwei Objekttypen gerichtet verbinden. Sie sind auf Grund der bijektiven Abbildung $L_{MT} : \Sigma_{MT} \rightarrow E_{MT}$ eindeutig benannt.

Vergleichbar mit einem SMM-Graph wird $\overrightarrow{G_{DM}}$ zu einem Multigraph $\overrightarrow{G'_{DM}}$, wenn mindestens zwei Mappingtypen $e_a = (v, w, \sigma_a) \in E_{MT}$ und $e_b = (s, t, \sigma_b) \in E_{MT}$ $(v, w, s, t \in V_{OT})$ über dasselbe geordnete Paar von Objekttypen $((v, w) = (s, t))$ verfügen, deren Namen $\sigma_a, \sigma_b \in \Sigma_{MT}$ sich unterscheiden ($\sigma_a \neq \sigma_b$).

Ein Domänenmodell kann aus einem Source-Mapping-Modell abgeleitet werden; auf die Angabe einer Transformationsvorschrift soll an dieser Stelle verzichtet werden. Gegenüber einem globalen Schema erlaubt das Domänenmodell, Datenquellen und Mappings semantisch auf einem weitaus höherem

konzeptuellen Niveau zu klassifizieren. Da den Objekttypen keine Attribute zugeordnet sind, lässt sich ein großes Spektrum von unterschiedlichen Datenquellen aufnehmen. Für viele Anwendungsszenarien, wie das oben beschriebene, sollten kleine Mengen von Objekt- und Mappingtypen ausreichend sein. Neue Objekt- und Mappingtypen können bedarfsgesteuert mit der Integration einer neuen Quelle (oder neuer Teilbereiche einer Quelle) hinzugefügt werden. Damit können sowohl das Source-Mapping-Modell als auch das Domänenmodell inkrementell erweitert werden.

Mappings haben allgemein einen assoziativen Charakter, d.h. sie ordnen die Objekte einer LDS den Objekten einer zweiten LDS zu und verbinden somit zwei LDS. Schlingen [Die06], mit denen Objekte einer LDS auf Objekte derselben LDS abgebildet werden, sind ebenso möglich. Jedoch sind zwei Fälle zu unterscheiden. Einerseits repräsentieren die assoziierten Objekte unterschiedliche Realwelt-Objekte. Das ist der Fall, wenn ein Mapping die Objekte von zwei unterschiedlichen Objekttypen verbindet, z.B. Genobjekte mit Proteinen. Ebenso kann ein Mapping zwischen zwei LDS bestehen, deren Objekte den gleichen Objekttyp besitzen, beispielsweise Genobjekte einer LDS, die zu funktional verwandten Genen in Beziehung gesetzt werden. Solche Mappings werden als *Assoziations-Mappings* bezeichnet. Andererseits kann ein Mapping die Objekte zweier LDS aufeinander abbilden, die dieselben Realwelt-Objekte beschreiben. In diesem Fall bildet das Mapping eine Gleichheits- oder Äquivalenzbeziehung ab. Daher werden Mappings dieser Art als *Same-Mappings* bezeichnet. Same-Mappings sind Grundlage einer Fusion von Daten, die dasselbe Realwelt-Objekt beschreiben. In Abbildung 9.2a ist *SameGenes* ein solches Same-Mapping, das die Genobjekte der PDS Ensembl und NetAffx assoziiert und deren Fusion ermöglicht.

9.5 Operatoren

iFuice verwendet eine Menge von Operatoren, mit denen Anfragen und Mappings ausgeführt werden können. Die Operatoren operieren typischerweise auf einer Menge von Eingabeobjekten, z.B. alle oder spezielle, d.h. selektierte, Objekte einer LDS. Sie generieren eine Menge von Ausgabeobjekten, die wiederum als Eingabe von weiteren Operatoren verwendet werden können. Die Operatoren sind Bestandteil der *iFuice*-Skriptsprache, für die in [Tho07] Syntaxdiagramme angegeben sind. Die *iFuice*-Skriptsprache erlaubt eine Kombination verschiedener Operatoren. Neben der sequentiellen Anordnung können die Operatoren auch geschachtelt werden, d.h. das Ergebnis einer Operation fungiert unmittelbar als Eingabe einer anderen Operation.

Die Menge der Operatoren teilt sich in quellenspezifische Operatoren,

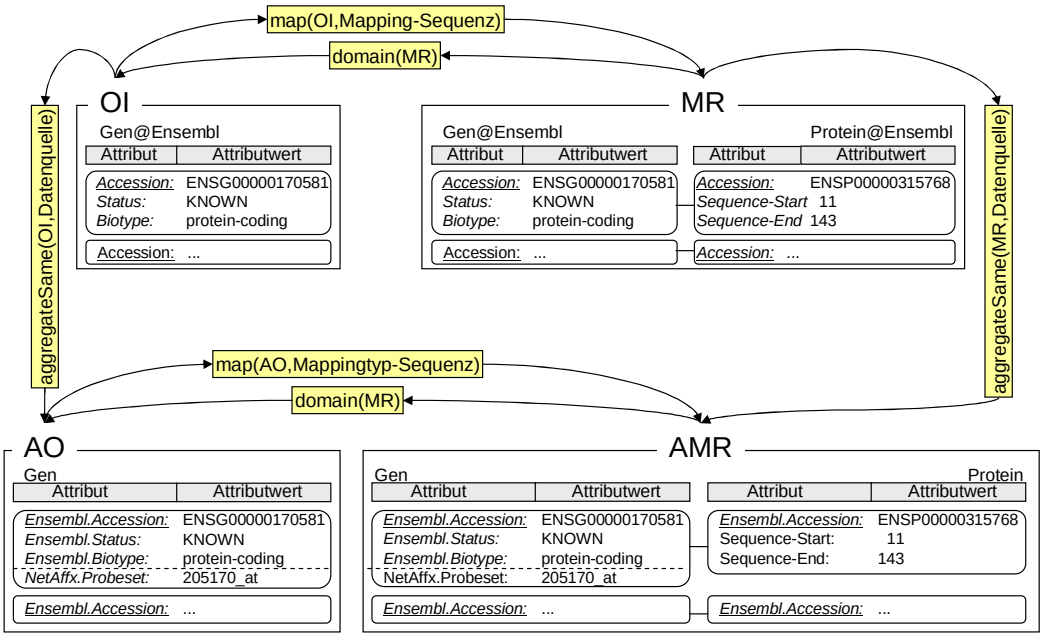


Abbildung 9.3: *iFuice*-Datenstrukturen als Grundlage der operatorgesteuerten Verarbeitung

Operatoren zur Navigation und Aggregation sowie mengenmanipulierende Operatoren. Dieser Einteilung folgend werden im Folgenden ausgewählte Operatoren kurz charakterisiert. Zuvor werden spezielle Datenstrukturen charakterisiert, die von den Operatoren genutzt und erzeugt werden. Eine ausführlichere Darstellung und Abgrenzung der Datenstrukturen findet sich ebenso in [Tho07].

9.5.1 Ausgewählte Datenstrukturen

In der *iFuice*-Skriptsprache werden primär vier Datenstrukturen verwendet und erzeugt: *Objektinstanzen*, *Mapping-Resultate*, *aggregierte Objekte* und *aggregierte Mapping-Resultate*. Die Abbildung 9.3 illustriert beispielhaft die Datenstrukturen und ihre Transformation unter Nutzung von Operatoren an ausgewählten Objekten aus dem Bereich der Bioinformatik.

Objektinstanzen. Die Datenstruktur Objektinstanzen (OI) repräsentiert eine Menge von Objekten/Instanzen einer LDS ($OI \subseteq O_{LDS}$). Jedes Objekt $o \in OI$ wird dabei durch eine Menge von Attributen bzw. deren zugeordneten Werte beschrieben, wovon eines ein ID-Attribut ist. Das ID-Attribut ist für alle Objekte aller OI einer LDS gleich; der Attributwert je Objekt ist

eindeutig. Es dient der Identifikation der Objekte einer LDS.

Mapping-Resultat Ein Mapping-Resultat (MR) ist eine Menge von Korrespondenzen, die zwischen den Objekten zweier LDS bestehen ($MR : OI \times OI$). Es kann mit der Ausführung eines Mappings für eine gegebene Eingabemenge von Objekten (d.h. ein OI) entstehen. Das MR umfasst damit die Teilmenge von Objektkorrespondenzen des ausgeführten Mappings, die für die Eingabemenge OI relevant sind.

Aggregierte Objekte. Die Datenstruktur der aggregierten Objekte (AO) stellt eine Menge von Elementen dar, wovon ein jedes eine Menge von Objekten zusammenfasst ($AO \subseteq Fusion(OI_1, \ldots, OI_n)$). Die zusammengefasste Objektmenge kann beispielsweise mit der Fusion gleichartiger Objekte unterschiedlicher LDS (Objekte, die dasselbe Realwelt-Objekt beschreiben) und unter Nutzung von verfügbaren Same-Mappings entstehen. Eine Fusion ist nur für Objekte möglich, die über den gleichen Objekttyp verfügen; daher besitzen alle Elemente von AO denselben Objekttyp.

Aggregiertes Mapping-Resultat. Die Datenstruktur aggregiertes Mapping-Resultat (AMR) umfasst eine Menge $AMR : AO \times AO$ von Korrespondenzen, die die Elemente zweier AO assoziieren. Die Korrespondenzen können mit der Anwendung von verschiedenen Mappings eines Mappingtyps auf ein AO entstehen.

9.5.2 Quellenspezifische Operatoren

Die Tabelle 9.1 gibt einen Überblick über die quellenspezifischen Operatoren *queryInstances*, *searchInstances* und *getInstances*, zeigt deren Signaturen und gibt eine prägnante Beschreibung. Die Operatoren sind polymorph, d.h. sie verwenden jeweils unterschiedliche Signaturen ohne ihre grundlegende Semantik zu ändern. Sowohl *queryInstances* als auch *searchInstances* selektieren aus einer gegebenen Menge von Objekten unter Nutzung einer spezifizierten Bedingung (Operator *queryInstances*) oder angegebenen Suchworten (Operator *searchInstances*) relevante Objekte. Eine solche Bedingung besteht aus einem Attributnamen, einem Vergleichsoperator (z.B. $=, \neq, <, >$) und einem Vergleichswert. Damit wird eine Anfrage mit dem Operator *queryInstances* gezielt für das angegebene Attribut ausgeführt. Dagegen werden mit dem Operator *searchInstances* alle Attribute der Objekte in der Ausgangsmenge durchsucht, ob sie die spezifizierten Suchwörter beinhalten. Die

Tabelle 9.1: Quellenspezifische Operatoren

Signatur	Beschreibung
OI = queryInstances(LDS, Bedingung)	Selektiert aus einer LDS die der Bedingung entsprechenden OI
OI = queryInstances(OI, Bedingung) AO = queryInstances(AO, Bedingung)	Selektiert aus den OI (AO) die der Bedingung genügende Teilmenge OI (AO)
OI = searchInstances(LDS, {Schlüsselwörter})	Selektiert aus einer LDS die Objekte, die die angegebenen Schlüsselwörter enthalten
OI = searchInstances(OI, {Schlüsselwörter}) AO = searchInstances(AO, {Schlüsselwörter})	Selektiert aus einer OI (AO) diejenige Teilmenge OI (AO), deren Objekte die Schlüsselwörter beinhalten
OI = getInstances(OI)	Fügt allen Objekten einer OI weitere Attribute aus der korrespondierenden LDS hinzu

Verbindung der Suchwörter kann dabei mit der Angabe von logischen Operatoren AND und OR erfolgen, was zu einer Ausweitung oder Eingrenzung der Ergebnismenge führt. Im Gegensatz zu *queryInstances* und *searchInstances* führt der Operator *getInstances* ein Mapping aus - sofern es vorhanden ist, um den Objekten, die nur das ID-Attribute besitzen, weitere beschreibende Attribute der korrespondierenden LDS hinzuzufügen[59].

9.5.3 Operatoren zur Navigation und Aggregation

Die Tabelle 9.2 zeigt ausgewählte Operatoren und ihre Signaturen zur Navigation und Aggregation. Der Operator *traverse* wendet auf ein OI der Ausgangs-LDS ein Mapping an, so dass ein OI der Ziel-LDS entsteht. Damit wird im Graphen des *Source-Mapping-Modells* traversiert. Der Operator *map* führt ebenso ein Mapping für eine gegebene Menge von Objekten aus. Im Unterschied zum Operator *traverse* besteht das Ergebnis von *map* in einer Menge von Korrespondenzen zwischen den Eingangs- und Zielobjekten, dem MR. Alternativ zu einem Mapping verwenden beide Operatoren auch eine Sequenz von Mapping-Namen oder ein MR (AMR), um innerhalb des *Source-Mapping-Modells* zu navigieren.

Dem gegenüber transformiert der Operator *aggregate* OI in AO sowie MR in AMR. In diesen Datenstrukturen können ähnliche bzw. zusammengehörige Objekte gruppiert und fusioniert werden (vgl. Abbildung 9.3).

[59] Welche Attribute hinzugefügt werden, bestimmt das auszuführende Mapping.

Tabelle 9.2: Operatoren zur Navigation und Aggregation

Signatur	**Beschreibung**
OI = traverse(OI, Mappingsequenz)	Traversiert im SMM von den OI einer LDS zu denen einer anderen durch die Anwendung einer Mappingsequenz
MR = map(OI, Mappingsequenz) AMR = map(AO, Mappingsequenz)	Stellt die Objekte des OI (AO) den korrespondierenden Zielobjekten gegenüber, die mit der Anwendung einer Mappingsequenz entstehen
MR = compose(MR, MR) AMR = compose(AMR, AMR)	Komponiert zwei MR (AMR), wobei die LDS des Wertebereiches des ersten MR (AMR) mit der des Definitionsbereiches des zweiten MR (AMR) übereinstimmen muss
AO = aggregate(OI) AMR = aggregate(MR)	Transformiert Objekte eines OI (MR) zu denen eines AO (AMR)
AO = aggregateSame(OI, Mappingtyp)	Fusioniert Objekte eines OI mit denen einer anderen LDS durch Ausführung aller Mappings eines bestimmten Typs

9.5.4 Mengenmanipulierende Operatoren

Die Tabelle 9.3 zeigt drei ausgewählte mengenmanipulierende Operatoren *union*, *intersect* und *diff* sowie ihre Signaturen. Sie bilden die typischen Mengenoperationen Vereinigung (union), Durchschnitt (intersect) und Differenz (diff) ab. Die Operatoren sind auf die Datenstrukturen OI, AO, MR und AMR ohne Einschränkung und ohne Änderung ihrer Semantik anwendbar.

Die Operatoren *domain* und *range* extrahieren den Definitions- (domain) und Wertebereich (range) eines MR/AMR.

Tabelle 9.3: Generische Operatoren

Signatur	Beschreibung
OI = union(OI, OI) AO = union(AO, AO)	Bildet die Vereinigungsmenge zweier Objektmengen OI (AO)
MR = union(MR, MR) AMR = union(AMR, AMR)	Bildet die Vereinigungsmenge zweier Korrespondenzmengen MR (AMR)
OI = intersect(OI, OI) AO = intersect(AO, AO)	Bildet den Durchschnitt zweier Objektmengen OI (AO)
MR = intersect(MR, MR) AMR = intersect(AMR, AMR)	Bildet den Durchschnitt zweier Korrespondenzmengen MR (AMR)
OI = diff(OI, OI) AO = diff(AO, AO)	Bildet die Differenz zwischen zwei Objektmengen OI(AO) in der angegebenen Reihenfolge
MR = diff(MR, MR) AMR = diff(AMR, AMR)	Bildet die Differenz zwischen zwei Korrespondenzmengen MR (AMR) in der angegebenen Reihenfolge
OI = domain(MR) AO = domain(AMR)	Gibt den Definitionsbereich eines MR (AMR) wieder
OI = range(MR) AO = range(AMR)	Gibt den Wertebereich eines MR (AMR) wieder

9.6 Skriptbasierte Analyse

Das Klassifikationsproblem der EST-Sequenzen (Q1-Q3) aus Abschnitt 9.2 kann mit dem folgenden einfachen *iFuice*-Skript gelöst werden, das die EST-Sequenzen den drei Klassen zuordnet.

Skript 9.1 *iFuice*-Skript zur Klassifikation von EST-Sequenzen

```
$allEstOI := queryInstances(EST-Sequenz@MyEST, "");
$alignedEstMR := map( $allEstOI, {EstDnaBlast.hsa} );
$unalignedEstOI := diff( $allEstOI, domain ( $alignedEstMR ));
$codingEstMR := compose ( $alignedEstMR,
                map( range($alignedEstMR ), {Ensembl.SRegionExons}));
$proteinCodingEstOI := domain ( $codingEstMR );
$nonCodingEstOI:= diff( domain($alignedEstMR), $proteinCodingEstOI);
```

Im ersten Schritt werden die in der lokalen Datenquelle MyEST gegebenen EST-Sequenzen abgefragt und der Variablen *allEstOI* zugeordnet; eine Bedingung wird nicht spezifiziert, da für alle EST-Sequenzen ohne Einschränkungen die folgende Analyse durchgeführt werden soll. Die EST-Sequenzen fungieren als Eingabe, um im zweiten Schritt die korrespondieren-

den DNA-Sequenzregionen des menschlichen Genoms in der Datenquelle Ensembl zu ermitteln. Dazu führt der Operator *map* das Mappings *EstDnaBlast.hsa* aus. Dieses Mapping ist das (mit Tabulatoren getrennte) Ergebnis eines Sequenz-Alignments unter Nutzung er Software *blastall* (BLAST). Das Ergebnis der Anweisung wird in der Variablen *alignedEstMR* gespeichert, die damit für alle zuordenbaren EST-Sequenzen die entsprechenden Korrespondenzen zur DNA-Sequenz enthält. Die Menge an EST-Sequenzen, für die kein Gegenstück in der DNA-Sequenz des menschlichen Genoms gefunden werden kann (Ergebnismenge zu Q1), entspricht der Differenz zwischen der Menge aller EST-Sequenzen in der Quelle MyEST und den zuvor ermittelten zuordenbaren Sequenzen (im Definitionsbereich des Mapping-Ergebnisses der Variablen *alignedEstMR*). In Schritt zwei wird diese Differenzermittlung durchgeführt und das Ergebnis in der Variablen *unalignedEstOI* gespeichert. Um darüber hinaus zwischen proteinkodierenden und nicht proteinkodierenden EST-Sequenzen zu unterscheiden, muss auf das biologische Basiswissen zurückgegriffen werden, da Gensequenzen typischerweise aus mehreren wechselseitig auftretenden Intron- und Exon-Sequenzabschnitten bestehen. Die Intron-Sequenzen werden im Spleiß-Prozess aus der zu translatierenden DNA-Sequenz herausgetrennt, bevor eine Translation erfolgt. Damit kodieren die Intron-Sequenzen keine Proteine. Dagegen sind die Exon-Sequenzen gewöhnlich in den Prozess der Proteinbildung involviert (vgl. Kapitel 3). In Schritt vier kommt das Mapping *Ensembl.SRegionExons* zur Ausführung, um die Exons zu ermitteln, die den DNA-Sequenzen aus dem zweiten Schritt zugeordnet werden können. Der Definitionsbereich dieser Mapping-Komposition beinhaltet damit alle EST-Sequenzen, die mit an der Proteinbildung beteiligten DNA-Sequenzen assoziiert sind (Schritt fünf; Ergebnismenge zu Q2). Schließlich wird im Schritt sechs die Menge an zuordenbaren EST-Sequenzen abgeleitet, die nicht zur Proteinbildung beitragen. Diese Menge (Ergebnismenge zu Q3) berechnet sich aus der Differenz aus allen zugeordneten und in den Proteinbildungsprozess involvierten EST-Sequenzen.

Dieses kleine Beispiel zur Klassifikation von gegebenen EST-Sequenzen zeigt die Mächtigkeit der mengenorientierten Operatoren, um Daten aus verschiedenen Quellen miteinander zu kombinieren. Die Operatoren machen es ebenso einfach, aufbauend auf den ermittelten Sequenzklassen weiterführende Analysen auszuführen, die sich ad-hoc ergeben können, so dass schnell auf neue Anforderungen reagiert werden kann. Beispielsweise ist es möglich, die proteinkodierenden EST-Sequenzen sowohl den Proteinen der Quelle SwissProt (siehe Q4) als auch den Genen der Quellen Ensembl und NetAffx zuzuordnen (siehe Q5), um eine anschließende fokussierte Genexpressionsanalyse zu unterstützen. Das wird durch die folgende Skripterweiterung erreicht.

Aus der ersten Anweisung resultiert ein Mapping-Resultat, das die pro-

Skript 9.2 *iFuice*-Skript zur explorativen Analyse der proteinkodierenden EST-Sequenzen aus dem vorangegangenen Szenario

```
$codingEstProteinMR = compose( $codingEstMR, map(
        range($codingEstMR), {Ensembl.ExonGene, Ensembl.GeneProteins}));
$codingEstGeneOI := traverse( range($codingEstMR), {Ensembl.ExonGene});
$fusedGeneAO := aggregateSame($codingEstGeneOI, NetAffx);
```

teinkodierenden EST-Sequenzen mit ihren korrespondierenden Proteinen in SwissProt beinhaltet (Ergebnismenge zu Q4). Im Ergebnis der zweiten Anweisung werden alle proteinkodierenden EST-Sequenzen mit den Genen der Datenquelle Ensembl assoziiert. Diese Menge an Genen kann zusätzlich mit Gendaten der Quelle NetAffx fusioniert werden. Dazu ist lediglich die Ausführung des vorhandenen Same-Mappings notwendig. Die fusionierten Gene werden im Anschluss durch die Attribute von beiden LDS, sowohl Gene@Ensembl als auch Gene@NetAffx, beschrieben.

9.7 Zusammenfassung

Im Mittelpunkt dieses Kapitels stand die Mapping-basierte Integration auf Basis des *iFuice*-Ansatzes. *iFuice* folgt einer Peer-to-Peer-artigen Integration von lokalen und privaten sowie öffentlich verfügbaren Datenquellen. Dazu werden die Quellen anhand von Mappings miteinander verknüpft. Die Datenquellen und Mappings werden mit semantischen Typen assoziiert, die in ein ausdrucksstarkes Domänenmodell eingehen und somit eine semantische Interoperabilität unterstützen. Anfragen und Mappings werden durch spezielle Operatoren ausgeführt, die in Skripten kombiniert zusammengefasst werden können.

Kapitel 10
BioFuice: iFuice in der Bioinformatik

10.1 Motivation

Der *iFuice*-Ansatz ermöglicht es, Daten aus unterschiedlichen, heterogenen Quellen unter Nutzung eines semantischen Modells zu integrieren. Obwohl der Ansatz generisch ist und damit in verschiedenen Domänen Anwendung finden kann, sind domänenspezifische Anforderungen zu berücksichtigen, die von diesem Ansatz nicht abgedeckt werden. Dazu gehören beispielsweise die Unterstützung spezieller Datenformate sowie Schnittstellen zu vorhandenen Analyseprogrammen. Zusätzlich zu denen von *iFuice* besitzt *BioFuice* die folgenden zentralen Eigenschaften.

- **Unterstützung spezieller Datenformate**: *BioFuice* ermöglicht den Zugriff auf Daten, die in verschiedenen, für den Bereich der Bioinformatik spezifischen Formaten vorliegen können. Dazu zählen proprietäre Datenformate von Quellen, wie beispielsweise Prosite [HSS+04], HUGO [EDTPS+06] und Enzyme [Bai00]. Darüber hinaus ermöglicht *BioFuice* einen Datenexport in das Fasta-Format[60] , das im Bereich der Bioinformatik sehr häufig zum Austausch von Sequenzdaten verwendet wird.
- **Einbindung in Analyseprogramme**: Die statistische Basissoftware R [Dev06] dient oftmals zur Analyse von experimentellen Daten,

[60] Ein CSV- und XML-Format werden ebenso unterstützt.

wie Genexpressionsdaten. Mit dem *RiFuice*-Paket stellt *BioFuice* eine Schnittstelle zur Verfügung, um innerhalb der Software R skriptgetriebene *iFuice*-Workflows auszuführen, z.B. um Datenintegrationsaufgaben zu bewältigen, deren Daten anschließend in R für eine weitergehende Analyse oder Kombination mit anderen Daten verfügbar gemacht werden. Damit wird der Analyseprozess flexibilisiert.

- **Unterschiedliche Formen der Anfrageformulierung**: *BioFuice* unterstützt neben der freien Skriptprogrammierung und der Ausführung parametrisierter Skripte auch spezielle Suchanfragen und Anfragen, die auf Basis der Metadaten-Modelle in einem GUI formuliert werden. Anfragen der letzten beiden Formen der Anfrageformulierung transformiert *BioFuice* in ausführbare *iFuice*-Skripte.

BioFuice wird in verschiedenen Projekten zur Datenintegration und zur Abbildung von Analyse-Workflows verwendet. Auf ausgewählte Szenarien wird in einem separaten Abschnitt dieses Kapitels eingegangen. Im Folgenden werden ausgehend von der *BioFuice*-Architektur ausgewählte Benutzer-Schnittstellen zur interaktiven Analyse sowie die Kopplung mit der Statistiksoftware R vorgestellt.

10.2 Systemarchitektur

Die Abbildung 10.1 gibt einen Überblick über die Systemarchitektur von *BioFuice*. Sie besteht aus mehreren Komponenten, wobei zwei davon, der *iFuice-Kernel* und *BioFuice base* für eine Anfrageverarbeitung, grundlegend sind. Mit Hilfe des generischen *iFuice-Kernels* werden Skripte ausgeführt und Daten aus verschiedensten Quellen fusioniert. Die Komponente *BioFuice base* stellt Basisdienste zur Verfügung, die dem Austausch von Skripten, Ergebnisdaten sowie Metadaten dienen. Ebenso zählen verschiedene Export-Schnittstellen zu den Basisdiensten, um die Ergebnisdaten in spezielle dateibasierte Formate (z.B. CSV, XML, FASTA) zu exportieren. Aufbauend auf der Komponente *BioFuice base* operieren weitere Komponenten und Schnittstellen, die eine skriptbasierte Abarbeitung von der Kommandozeile (Kommandozeilen-Schnittstelle), eine interaktive Analyse (*BioFuice Query*) sowie die Kopplung mit der Software R (unter Nutzung des Pakets *RiFuice*) zur statistischen Datenanalyse verfolgen.

Der *iFuice-Kernel* besteht wiederum aus einzelnen Komponenten. Dazu gehören generische Mapping-Ausführungsdienste, ein zentrales Datenfusionsmodul, ein Repository und Mediator-Schnittstellen. Jedes Mapping ist mit einem Mapping-Ausführungsdienst assoziiert, mit dem eine Definition und

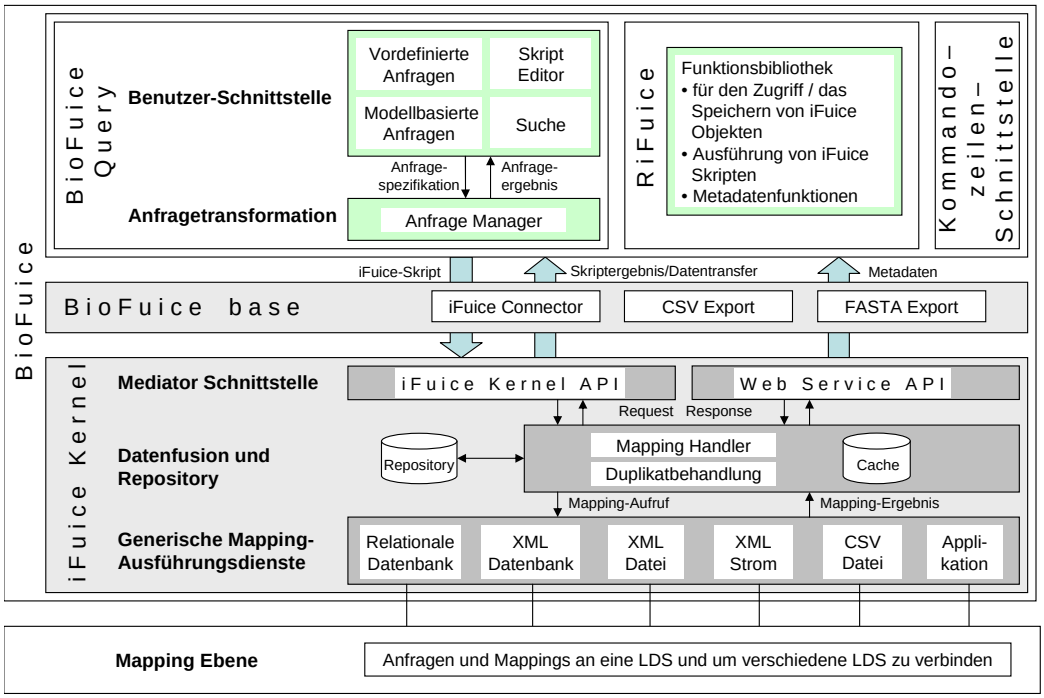


Abbildung 10.1: Systemarchitektur von BioFuice

Beschreibung des Mappings ermöglicht wird. Damit erlangt der Mediator Zugriff auf die Mapping-Daten und kann das Mapping im Bedarfsfall ausführen. Ein Mapping, das in einer relationalen Datenbank gespeichert ist, wird unter Nutzung des Ausführungsdienstes "Relationale Datenbank" definiert. Die Definition schließt die Angabe einer SQL-Anfrage sowie zusätzliche Parameter ein, um einerseits Zugriff auf die Datenbank zu erlangen und andererseits die Mappingdaten abzufragen. Das Spektrum an unterschiedlichen Mapping-Ausführungsdiensten ermöglicht die Definition von Mappings für Quellen, die in verschiedenen Formaten vorliegen, wie z.B. relationale Datenbanken, XML-basierte Quellen und Java Applikation.

Das *iFuice*-Repository enthält alle Metadaten des Source-Mapping- und des Domänenmodells. Dazu speichert es nicht nur Beschreibungen zu allen LDS, sondern auch zu allen verfügbaren Mappings und den semantischen Objekt- und Mappingtypen. Die Mediator-Schnittstellen gewährleisten den Zugriff auf die Funktionalität von *iFuice*. Es stehen sowohl eine Programmschnittstelle (API) als auch spezielle Web-Service-Methoden zur Verfügung. Die Web-Service-Methoden ermöglichen es, dass der *iFuice-Kernel* auf einem separaten Server initialisiert und ausgeführt wird.

Im Prozess der Skriptverarbeitung dienen beide Mediator-Schnittstellen

dazu, das auszuführende Skript entgegen zu nehmen. Der *iFuice*-Kernel leitet es an den Mapping-Handler (Bestandteil des Datenfusionsmoduls) weiter, der die im Skript enthaltenen Anweisungen parst und anschließend operatorgesteuert Anfragen und Mappings zur Ausführung bringt sowie die Variablenzuweisung vornimmt. Die Ausführung von Anfragen und Mappings basiert auf den Mapping-Definitionen und -Beschreibungen. Dazu werden zur Laufzeit die Anfragen, z.B die SQL-Anfrage auf Basis des Mapping-Ausführungsdienstes "Relationale Datenbank", um Attributwerte der Eingabeobjekte erweitert, die anschließend an die definierte Datenbank (in der das Mapping gespeichert ist) zur Ausführung übergeben wird. Zwischen- und Endergebnisse können in einem Cache bei Bedarf temporär gespeichert werden. Mit dem Abschluss der Skriptverarbeitung stehen die Ergebnisse zum Abruf bereit; mit speziellen Methoden der Mediator-Schnittstellen kann auf die Daten zugegriffen bzw. können diese transferiert werden.

Wie der *iFuice-Kernel* sind auch die Komponenten *BioFuice base* und *BioFuice Query* modular aufgebaut. *BioFuice base* stellt Grundfunktionalitäten zur Verfügung, die es in einer verteilten Umgebung ermöglichen, auf den *iFuice-Kernel* zuzugreifen (insbesondere dann, wenn die Web-Service-Schnittstelle verwendet wird). Dazu beinhaltet es das Modul *iFuice Connector*, das dem Verbindungsauf- und abbau zu einem *iFuice-Kernel* dient und Zugriff auf die *iFuice*-Metadaten sowie Ergebnisdaten gewährt. Ferner enthält es drei Schnittstellen, mit denen die Ergebnisdaten eines *iFuice*-Skripts in das CSV-, Fasta- und ein XML-Format[61] exportiert werden können. Damit kann *BioFuice* auf der Kommandozeile, z.B. auf einem Server, gestartet werden, so dass die Kopplung mit anderen Analysetools flexibilisiert wird.

Dagegen stellt *BioFuice Query* eine interaktive Benutzerschnittstelle zur Verfügung, mit der nicht nur Anfragen spezifiziert werden können, sondern in der auch die Ergebnisdaten dargestellt werden. Das Spektrum von unterstützten Anfragetypen umfasst neben der freien Skriptprogrammierung die Ausführung vordefinierter, parametrisierter Skripte und zwei Formen der Stichwortsuche. Darüber hinaus kann eine Anfrage unter Nutzung der *iFuice*-Metadaten-Modelle formuliert werden, den so genannten modellbasierten Anfragen. Da die Anfragen zur Stichwortsuche sowie die modellbasierten Anfragen in einem GUI formuliert werden, bedarf es einer Transformation der spezifizierten Anfrage in ein ausführbares *iFuice*-Skript, das im Anschluss an den *iFuice-Kernel* gesendet und dort abgearbeitet wird.

Das *RiFuice*-Paket dient der Kopplung von *BioFuice* mit der statistischen Software R. Damit werden die von *BioFuice* integrierten Daten direkt in den Analysen verwendbar, die mit R durchgeführt werden. Ebenso wie *Bio-*

[61] Im Anhang E werden die DTD zum XML-basierten Datenaustausch aufgezeigt.

Tabelle 10.1: Verbindungs- und Ausführungsfunktionen im Überblick

Funktionsname	Beschreibung
void=connect(URI Mediator,URI Konf.-datei)	Aufbau einer Verbindung zum *iFuice*-Mediator, der an der Adresse 'URI Mediator' zu finden ist und für den die 'Konfigurationsdatei' bei der Initialisierung benutzt werden soll
void=disconnect()	Abbruch der bestehenden Verbindung zum *iFuice*-Mediator
void=executeCommand(*iFuice*-Operation)	Ausführung der angegebenen *iFuice*-Operation
void=executeScript(URI Skriptdatei)	Ausführung des spezifizierten *iFuice*-Skripts

Fuice Query verwendet das *RiFuice*-Paket Module und Schnittstellen von *BioFuice base*, um auf die Funktionen des *iFuice-Kernels* und dessen Daten zuzugreifen. Das *RiFuice*-Paket und die interaktive Anfrageverarbeitung mit *BioFuice Query* sind Inhalt der nächsten Abschnitte.

10.3 Das *RiFuice*-Paket zur statistischen Analyse

Das optionale *RiFuice*-Paket ist eine umfassende Funktionsbibliothek, mit der die statistische Software R um Datenintegrationsfunktionalitäten erweitert werden kann. Dazu greift das Paket *RiFuice* auf die Basisdienste von *BioFuice base* zurück, mit dem der Zugriff auf das Funktionsspektrum des *iFuice*-Kernels sichergestellt wird. Die Nutzung des *RiFuice*-Pakets ermöglicht es, Daten von ad-hoc integrierten Quellen in die statistische Analyse einzubeziehen.

Die Funktionsbibliothek des *RiFuice*-Pakets gliedert sich in drei Gruppen, den Verbindungs- und Ausführungsfunktionen, den Metadaten-Funktionen sowie den Import- und Exportfunktionen. Alle Funktionen nutzen die Funktionalitäten des Pakets SJava[62], das den Zugriff auf Java-Objekte und deren Methoden in R sowie den Datenaustausch zwischen R und den Java-Objekten sicherstellt.

[62] `http://www.omegahat.org/RSJava`

Tabelle 10.2: Funktionen zum Metadaten-Management im Überblick

Funktionsname	Beschreibung
data.frame=getLdsNames()	Rückgabe von Namen und Beschreibungen aller logischen Datenquellen, die der *iFuice*-Mediator benutzen kann
data.frame=getMappingNames()	Rückgabe der Namen aller Mappings, über die der *iFuice*-Mediator verfügt
data.frame=getOperatorNames()	Rückgabe der Namen, Beschreibungen und Signaturen aller *iFuice*-Operatoren
data.frame=getVariableNames()	Rückgabe von Namen aller verfügbaren Variablen des *iFuice*-Mediators, auf die zugegriffen werden kann
void=deleteVariable(Variablenname)	Löschung der angegebenen Variablen im *iFuice*-Mediator
void=deleteVariable()	Löschung aller Variablen im *iFuice*-Mediator

10.3.1 Verbindungs- und Ausführungsfunktionen

Die Tabelle 10.1 zeigt die Verbindungs- und Ausführungsfunktionen des *RiFuice*-Pakets im Überblick. Die Funktion *connect()* stellt die Verbindung zum angegebenen *iFuice*-Mediator her, der lokal installiert oder als Web-Service verfügbar sein kann. Eine solche Verbindung ist grundlegend, um *iFuice*-Skripte und Operationen sowie alle Metadatenfunktion und Im-/Exportfunktionen auszuführen.

Die Funktionen *executeCommand()* und *executeScript()* dienen der Ausführung einer einzelnen *iFuice*-Operation, z.B. einer Anfrage oder eines Mappings, sowie eines gesamten Skripts, das lokal gespeichert ist.

10.3.2 Metadaten-Management-Funktionen

Die Tabelle 10.2 zeigt die Funktionen, um Metadaten des *iFuice*-Mediators zurückzugeben. Anhand dieser Funktionen gewinnt der Benutzer einerseits einen Überblick über die im *iFuice*-Mediator verfügbaren logischen Datenquellen und Mappings als Komponenten des Source-Mapping-Modells, aus dem das Domänenmodell mit seinen semantischen Objekt- und Mappingtypen erzeugt werden kann. Andererseits zeigen die Funktionen *getOperatorNames()* und *getVariableNames()* die verfügbaren Operatoren und deren Beschreibung, die zum Erstellen von Skripten Verwendung finden, sowie die im Mediator vorhandenen Variablen, die Daten referenzieren. Mit den Funktionen *deleteVariable()* und *deleteVariables()* können selektiv einzelne oder alle Variablen im Cache des Mediators gelöscht werden.

Tabelle 10.3: Import- und Exportfunktionen im Überblick

Funktionsname	Beschreibung
List=getVariableData(Variablenname)	Rückgabe der Daten vom *iFuice*-Mediator, die vom angegebenen Variablennamen referenziert werden
void=setOI(Variablenname, OI)	Transfer der Objektinstanzen zum *iFuice*-Mediator, in dem es unter dem angegebenen Variablennamen gespeichert wird
void=setMR(Variablenname, MR)	Transfer des Mapping-Ergebnisses zum *iFuice*-Mediator, in dem es unter dem angegebenen Variablennamen gespeichert wird

Die resultierenden Metadaten werden einer der Software R eigenen Datenstruktur, den so genannten *data frames* gespeichert.

10.3.3 Import- und Exportfunktionen

Die Tabelle 10.3 zeigt die Funktionen, mit denen ein Datentransfer zwischen einem R-Programm und dem *iFuice*-Mediator durchgeführt werden kann. Während die Funktion *getVariableData()* die Daten der spezifizierten Variablen für alle möglichen Datenstrukturen des *iFuice-Kernels* (OI, MR, AO, AMR etc.) in der Software R verfügbar machen, bleibt der Datentransfer zum Mediator auf Grund ihrer Komplexität auf die Datenstrukturen OI und MR beschränkt.

10.3.4 Anwendungsmöglichkeiten

Mit der Anwendung des *RiFuice*-Pakets wird das Spektrum an Funktionen, das *BioFuice* zur Verfügung stellt (z.B. zum Zwecke der Datenintegration), für eine statistische Datenanalyse in der Software R nutzbar. Dazu zählen Methoden der deskriptiven Statistik, der induktiven Statistik mit verschiedenen Tests aber auch komplexere Analysemethoden, wie die des Data Mining. Solche Methoden sind in großer Zahl in R und seinen erweiternden Paketen vorhanden. Eine populäre Auswertungsstrategie im Bereich der Genexpressionsanalyse ist die Erstellung und Validierung von funktionalen Profilen (engl. functional profiling). Ein solches Profil entsteht aus der Zuordnung von aus der Genexpressionsanalyse resultierenden signifikanten, differentiell exprimierten oder co-exprimierten Genen zu funktionalen Beschreibungen. Für letztere werden vorzugsweise die Subontologien der GeneOntology verwen-

det. Mit dem erstellten Profil können Konzepte oder Gruppen von Konzepten abgegrenzt werden, die gegenüber allen anderen Konzepten eine signifikante Anzahl von zugeordneten auffälligen Genen aufweisen. Anstatt viele unterschiedliche Softwaretools zu nutzen, die für eine solche Analyse entwickelt wurden, kann die Analyse umfassend in R durchgeführt werden. Die Nutzung dieser Software hat darüber hinaus den Vorteil, dass algorithmische oder methodische Neuentwicklungen schnell, mit der Möglichkeit die Software um Pakete zu ergänzen, hinzugefügt werden können. Damit wird eine aufwändige Reimplementierung in bestehende Programmsysteme vermieden.

Das *RiFuice*-Paket konnte in kleineren Anwendungsfällen seine Praxistauglichkeit zeigen; ein umfangreicherer Einsatz ist geplant.

10.4 Interaktive Anfragen mit BioFuice Query

BioFuice Query stellt verschiedene Möglichkeiten der Anfrageformulierung in Hinsicht auf eine explorative Datenanalyse zur Verfügung. Zusätzlich zu der von *iFuice* bekannten freien Skriptprogrammierung und -ausführung unterstützt *BioFuice* vordefinierte, parametrisierte Anfragen, Anfragen auf Basis der *iFuice*-Metadaten-Modelle sowie eine Stichwortsuche. Eine vordefinierte, parametrisierte Anfrage besteht aus einem *iFuice*-Skript, das an bestimmten Stellen Parameter anstatt realer Werte aufweist. Die Parameter können spezielle Bedingungen oder Vergleichswerte zur Filterung relevanter Daten sein. Die Parameterwerte werden vor der Skriptausführung in *BioFuice* durch den Benutzer spezifiziert und ersetzen die Parameter im *iFuice*-Skript.

Da die vordefinierten, parametrisierten Anfragen in manchen Fällen zu statisch sind, das Schreiben von Skripts für den Endbenutzer aber vielfach zu komplex ist, unterstützt *BioFuice* Anfragen auf Basis der *iFuice*-Metamodelle und eine Stichwortsuche. Erstere, die so genannten modellbasierten Anfragen, nutzen das Source-Mapping- und das Domänenmodell zur Anfrageformulierung. Die Abbildung 10.2a zeigt die interaktive Benutzer-Schnittstelle für diese Art der Anfragespezifikation. Beide Metadaten-Modelle werden als Graphen im linken Teil der Benutzeroberfläche illustriert (oben: Domänenmodell, unten: Source-Mapping-Modell). Die Knoten repräsentieren Objekttypen (LDS) und die Kanten symbolisieren die Mappingtypen (Mappings) im Domänenmodell (Source-Mapping-Modell).

Auf der rechten Seite des GUI in Abbildung 10.2a kann der Benutzer relevante Quellen auswählen und Anfragebedingungen in Form von Schlüssel-

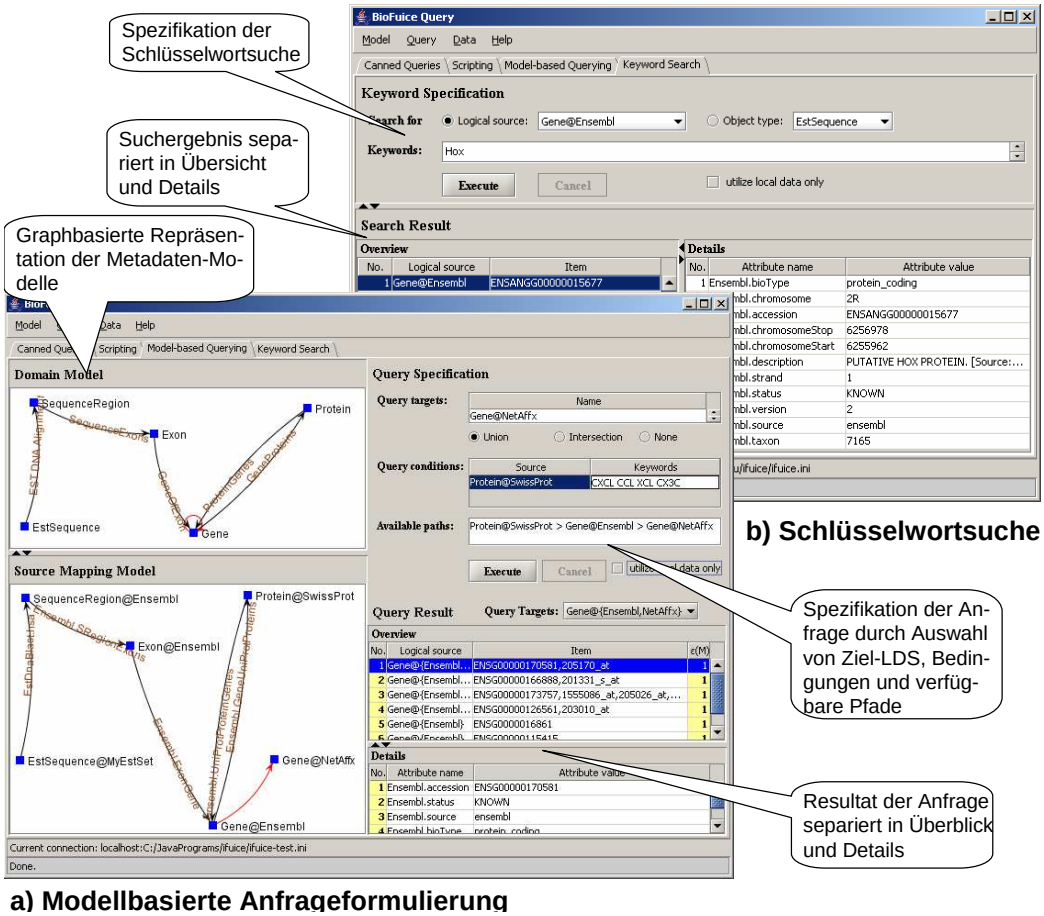


Abbildung 10.2: Interaktive Anfrageformulierung mit BioFuice Query

wörtern für spezielle LDS spezifizieren. Darüber hinaus muss die Ziel-LDS der Anfrage vom Benutzer spezifiziert werden. Solch eine Ziel-LDS besteht in einer Menge von LDS, für die die Objektinstanzen schließlich wiedergegeben werden sollen. Objektinstanzen eines Objekttypes werden dabei unter Nutzung von vorhandenen Same-Mappings aggregiert. Auf Basis des Source-Mapping-Modells identifiziert *BioFuice* hiernach automatisch die verfügbaren Pfade, die die LDS, denen die Anfragebedingungen zugeordnet sind, mit den Ziel-LDS verbinden. Die Pfade werden ebenso auf der rechten Seite des GUI dargestellt, von denen der Benutzer die für ihn relevanten markieren muss.

Die Abbildung 10.2a zeigt exemplarisch eine solche modellbasierte Anfragespezifikation für eine einfache explorative Analyse. Das Ziel der Analyse besteht darin, alle Gene der Quelle NetAffx zu finden, die zu den Chemokin-Proteinen – Proteine, die für Zell-Zell Interaktionen verantwortlich sind –

korrespondieren. Ausgehend vom Source-Mapping-Modell soll die Datenquelle SwissProt verwendet werden, um die relevanten Proteine zu identifizieren, zu denen im Anschluss die korrespondierenden Gene in der Quelle Ensembl und NetAffx wiedergegeben werden. *BioFuice* übersetzt die interaktiv spezifizierte Anfrage in das folgende *iFuice*-Skript:

Skript 10.1 Erzeugtes *iFuice*-Skript auf Basis einer modellbasierten Anfrage

```
$zzz_0:=searchInstances(Protein@SwissProt,{"CXCL","CCL","XCL","CX3C"});
$zzz_1:=map($zzz_0,{Ensembl.ProtGenes});
$zzz_2:=aggregateSame(range($zzz_1),NetAffx);
```

Im ersten Schritt wird die LDS Protein@SwissProt benutzt, um die Chemokin-Proteine zu suchen. In [TBY+05] werden diese Proteine systematisch in die vier Gruppen CXC, CC, XC und CX3C klassifiziert. Diese Gruppennamen können für die Suche der relevanten Proteine benutzt werden, da sie Bestandteil des Proteinnamens und der korrespondierenden Beschreibung sind. Im zweiten Schritt werden die identifizierten Proteine mit den Genen der Quelle Ensembl assoziiert, die im dritten Schritt mit den Genen der Quelle NetAffx auf Basis des definierten Same-Mappings fusioniert werden. Eine formale Beschreibung der Skripterzeugung sowie eine Abgrenzung zum Umfang der *iFuice*-Skriptsprache gibt Anhang D.

Mit einer Stichwortsuche fokussiert der Benutzer eine relevante Objektmenge anhand spezifizierter Stichwörter. Es werden zwei Formen der Stichwortsuche unterschieden. Eine LDS-spezifische Stichwortsuche sucht relevante Objekte in einer ausgewählten LDS anhand der spezifizierten Wortmenge (Stichwörter). Dagegen werden relevante Objekte in einer Stichwortsuche für einen ausgewählten Objekttyp in allen mit dem Objekttyp assoziierten LDS gesucht. Die Objekte der resultierenden Mengen (je LDS eine resultierende Objektmenge) werden im Anschluss unter Nutzung von Same-Mapping fusioniert und die entstehenden Mengen vereinigt.

Die Abbildung 10.2b zeigt die Benutzerschnittstelle für eine LDS-spezifische Stichwortsuche, in der der Benutzer an dem Auffinden von HOX-Genen in der Quelle Ensembl interessiert ist. Im Zuge der Anfrageformulierung wird die LDS Gene@Ensembl ausgewählt und das Stichwort "Hox" spezifiziert. Aus dieser Anfrage resultiert das folgende *iFuice*-Skript, das der Anfrage-Manager erstellt. Eine formale Beschreibung der Skripterzeugung für LDS- und Objekttyp-spezifische Anfragen zur Stichwortsuche sowie eine Abgrenzung zur *iFuice*-Skriptsprache ist im Anhang D enthalten.

Skript 10.2 Erzeugtes *iFuice*-Skript zur Stichwortsuche

```
$zzz_0:=searchInstances(Gene@Ensembl,{"Hox"});
```

10.5 Ausgewählte Anwendungsszenarien

BioFuice wurde in verschiedenen Analyseszenarien im Bereich der Bioinformatik eingesetzt und dabei auf Anwendbarkeit überprüft. Zu diesen Analyseszenarien zählen insbesondere Integrationsprozesse zur Unterstützung der Genexpressionsanalyse, die Integration von Proteininteraktionen sowie die Datenintegration als Grundlage der Analyse von verschiedenen Typen von RNA-Sequenzen. In jedem dieser Analyseszenarien wird mit der Auswahl und Beschreibung relevanter Datenquellen und Mappings begonnen. Damit kann der Benutzer eine Analyse auf die notwendigen Daten fokussieren, während er gleichzeitig in der Lage ist, neue Quellen und Mappings bedarfsgesteuert hinzuzufügen. Im Folgenden werden die drei genannten Analyseszenarien im Überblick vorgestellt.

10.5.1 Datenintegration in der Genexpressionsanalyse

Eine Genexpressionsanalyse untersucht das Expressionsverhalten von Genen unter verschiedenen zugrunde liegenden Bedingungen (vgl. Kapitel 3). Zur Interpretation der Ergebnisse werden oftmals weitere Daten zu den untersuchten Genen notwendig. Dazu gehören beispielsweise die im Translationsprozess entstehenden Proteine oder funktionale Eigenschaften der Gene. Diesem Ansatz folgend bestand die Zielstellung in diesem Analyseszenario einerseits darin, die Gene einer Gengruppe mit Proteinen der Quellen SwissProt, Trembl, UniProt [BA00] und Ensembl [BAB+04, HAC+05] zu assoziieren sowie zugeordnete funktionale Eigenschaften unter Verwendung der Ontologie GeneOntology (GO) [HCI+04] aufzuzeigen. Die Gengruppen waren das Ergebnis einer vorherigen Analyse von Microarray-Daten und wurden teilweise auf Basis der *GeWare*-Plattform erzeugt oder von Kooperationspartnern in Dateiform zur Verfügung gestellt. Andererseits dienten die genannten Quellen dazu, relevante Gene zu finden, für die eine fokussierte Analyse der gemessenen Expressionswerte durchgeführt wurde. Das Analyseszenario umfasste unter anderem die folgenden, ausgewählten Fragestellungen.

- Welche Proteine (aus den ausgewählten Datenquellen Ensembl, SwissProt und Trembl) werden von den Genen einer gegebenen Gengruppe kodiert?
- Welche Proteine werden von Genen kodiert, die zusammen mit Stat-3 (Protein) regulierenden Genen co-exprimiert werden und damit in einer Gengruppe

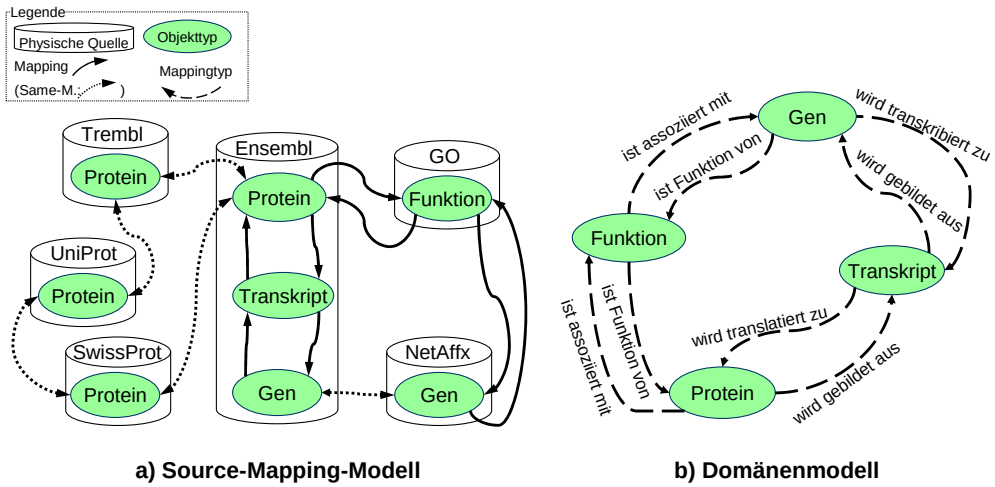


Abbildung 10.3: Metadaten-Modelle im Bereich der Genexpression

vertreten sind?

- Welche Gene der Datenquelle NetAffx und einem Microarray vom Typ (Chiptyp) HG-U133A sind mit Chemokine-Liganden assoziiert?
- Welche Gene der zur Verfügung gestellten Gengruppe kodieren Chemokine-Rezeptoren oder sind mit molekularen Funktionen/biologischen Prozessen assoziiert, denen Chemokine-Rezeptoren zugeordnet sind ?
- Welche Gene der Datenquelle NetAffx (Chiptyp HG-U133A) haben ähnliche funktionale Eigenschaften wie die Hox-Gene unter Ausnutzung der is-a und part-of Beziehungen zwischen den Konzepten der Subontologien molekulare Funktionen und biologische Prozesse der GeneOntology?

Die durchgeführten Analysen waren beispielsweise Grundlage für Simulationen von Zell-Zell-Wechselwirkungen von Kooperationspartnern [GSH+06][63]. Die Abbildung 10.3 zeigt exemplarisch die Metadaten-Modelle (Source-Mapping-Modell und Domänenmodell) zur Unterstützung der Genexpressionsanalyse. Es unterstützt sowohl explorative Analysen, etwa um differenziell exprimierte Gene den korrespondierenden Proteinen und GO-Funktionen zuzuordnen und zu sortieren, als auch um relevante Gene durch eine Selektion von Proteinen zu identifizieren, die später einer gesonderten Betrachtung in der statistischen Analyse unterliegen.

Für die Quellen SwissProt, Trembl, Ensembl und GO wurden bereits vorhandene lokale Kopien verwendet. Die Mappings zwischen den Datenquellen entstammen weitestgehend den Quellen Ensembl und NetAffx [LLS+03,

[63] Auf die Darstellung einzelner Ergebnisse soll an dieser Stelle verzichtet werden.

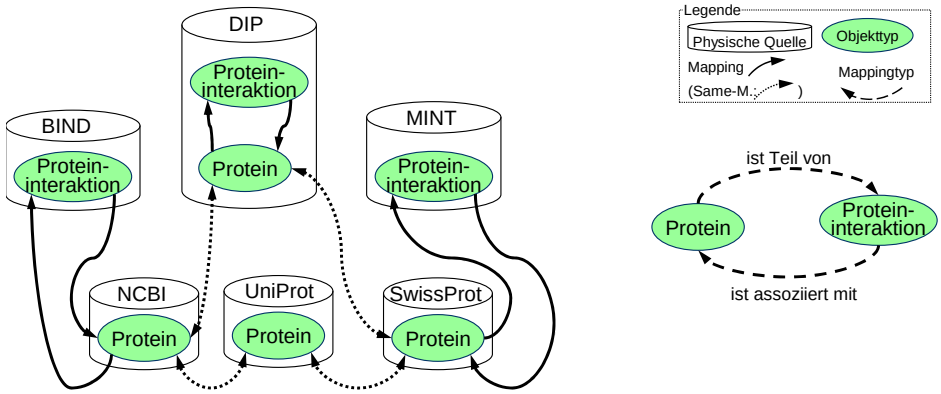


Abbildung 10.4: Metadaten-Modelle zur Integration von Proteininteraktionen

CST+04]; letztere stellt auch die Verbindung zwischen Genen und den auf einem Microarray befindlichen Sequenzen her, für die die Expressionswerte ermittelt und analysiert werden. Die Gengruppen standen als private Datenquellen von Kooperationspartnern zur Verfügung und wurden auf Basis von CSV-Dateien integriert. Die Gen-Objekte in diesen Dateien verwenden einen Identifikator der öffentlichen Quelle NetAffx; damit ist ihnen bereits das Mapping zwischen der privaten Datenquelle (Datei) und NetAffx inhärent. Eine Kopplung mit der *GeWare*-Plattform wurde bisher nicht verfolgt.

10.5.2 Integration von Proteininteraktionen

Ein weiteres Anwendungsgebiet von *BioFuice* besteht in der Integration von Proteininteraktionen aus unterschiedlichen Quellen. Im Gegensatz zum vorgenannten Analyseszenario, für das eine Integration für jeweils eine relevante Objektmenge (Gen, Gengruppe, Protein) vorgenommen wurde, ist eine Integration für alle Objekte der Quellen notwendig. Damit soll eine Analyse großer zusammenhängender Interaktionsnetzwerke unterstützt werden. Ein typisches Ziel solcher Analysen besteht darin, relevante Muster zwischen den Eigenschaften von speziesspezifischen Netzwerken aufzudecken. Solche Eigenschaften sind beispielsweise Eigenwerte, Cluster-Koeffizienten und die Skalenfreiheit der Netzwerke. Das erlaubt eine Charakterisierung des Zusammenspiels zwischen Verhalten, Struktur und Funktion, wie es in [BO04] beschrieben wird.

Die Abbildung 10.4 zeigt die Metadaten-Modelle zur Integration und Fusion von Proteininteraktionen aus den Quellen BIND [BDW+01], DIP

Tabelle 10.4: Mengengerüst zu Proteinen und Proteininteraktionen für die Spezies Homo Sapiens in BIND, DIP und MINT

Datenquelle	# Proteine	# Proteininteraktionen
BIND	11.959	2.375.139
DIP	1.025	1.555
MINT	2.076	5.632

[XSD+02] und MINT [ZMPQ+02], die großen Mengen von Proteininteraktionen verschiedener Spezies beinhalten. Die Proteininteraktionen werden in diesen Quellen mit verschiedenen Attributen beschrieben. Da in *BioFuice* lediglich LDS mit Attributen beschrieben werden können (für ein Mapping stehen lediglich die beiden vorgegebenen Attribute confidence und occurrence zur Verfügung), wurde ein separater Objekttyp Proteininteraktion verwendet, der die Beziehung zwischen zwei Proteinen charakterisiert. Während BIND und MINT direkt die Proteinidentifikatoren (ID Attribut) der Quellen NCBI Protein respektive SwissProt verwenden, benutzt DIP einen eigenen Identifikator. Jedoch bietet DIP Mappings zu beiden Quellen, NCBI Protein und SwissProt. Somit ist eine Integration der Proteininteraktionen möglich, in der die Proteindaten aus den unterschiedlichen Quellen auf Basis der vorhandenen Mappings fusioniert werden.

Die Analyse der Proteininteraktionen ergab, dass die drei Quellen nicht nur heterogen in Bezug auf ihren Aufbau sind, sondern sich auch in Hinsicht auf die enthaltenen Daten stark unterscheiden. Während BIND Proteininteraktionen zu sechs Spezies (Mensch Maus, Ratte, Hefe, Wurm, Fruchtfliege) enthält, speichert MINT Interaktionen zu über 240 Spezies. Jedoch fokussiert MINT vor allem auf spezielle Viren; Proteininteraktionen der Spezies Mensch, Maus und Ratte sind nur zu einem kleinen Teil enthalten. DIP beinhaltet Proteininteraktionen zu acht Spezies (u.a. Mensch, Maus, Ratte, Hefe, Fruchtfliege), die in mehreren Experimenten nachgewiesen wurden. Dadurch ist die Anzahl der Proteininteraktionen sehr begrenzt. Die Tabelle 10.4 zeigt das Mengengerüst der drei Datenquellen hinsichtlich der enthaltenen Protein-Objekte und Interaktionen für die Spezies Homo Sapiens. Es ist zu sehen, dass sich die Anzahl der Proteine und Interaktionen in den Datenquellen signifikant unterscheidet. Die Überlappung der Datenquellen bleibt gering; DIP und MINT sind disjunkt. Dies kann auch dem nicht perfekten Same-Mapping zwischen den Protein-Datenquellen SwissProt und NCBI Protein geschuldet sein, das aus letzterer Datenquelle über die HTML-Schnittstelle extrahiert wurde. Ein Same-Mapping, das evtl. zusätzlich die Vergleichsergebnisse der Proteinsequenzen beider Datenquellen ausnutzt, könnte bessere

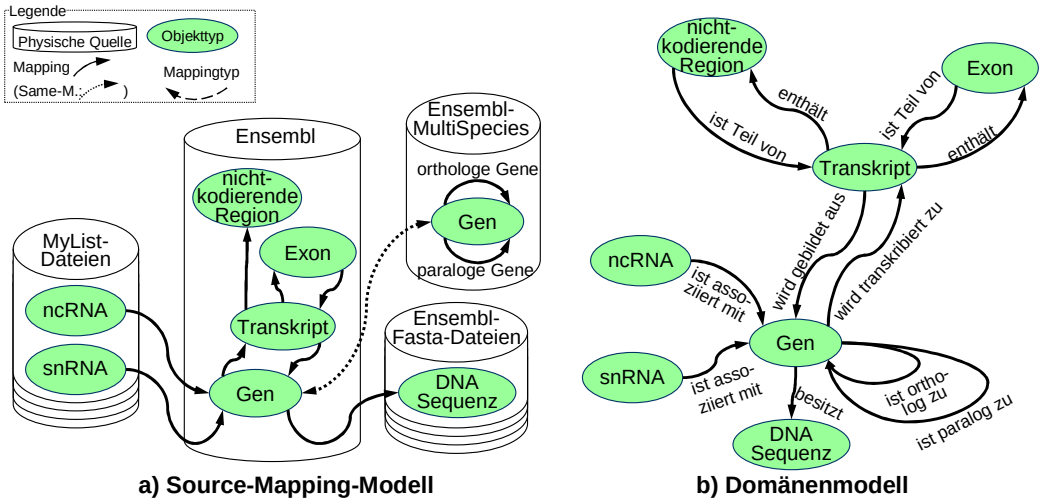


Abbildung 10.5: Metadaten-Modelle im Analysebereich von ncRNA, miRNA und snRNA

Ergebnisse liefern. Die weitere Analyse der Proteininteraktionen mit Fokus auf die Netzwerkeigenschaften wurde von einem Kooperationspartner (MPI Mathematik in den Naturwissenschaften) durchgeführt.

10.5.3 Integrationsprozesse zur Untersuchung nicht kodierender RNA

Nicht kodierende RNA (engl. ncRNA - non-coding RNA) sind spezielle Moleküle, die nicht in Proteine übersetzt werden. Die ncRNA umfassen verschiedene Typen, z.B. snRNA (engl. small nuclear RNA), snoRNA (engl. small nucleolar RNA), microRNA oder tRNA (engl. transfer RNA). Die Aufgaben und Funktionen dieser ncRNA sind vielfältig; sie sind Gegenstand intensiver Forschung. Vielfach kommt ihnen im Transkriptionsprozess von Genen, die im Weiteren in Proteine translatiert werden, eine bestimmte Funktion zu. Analysen in diesem Bereich haben beispielsweise das Auffinden homologer ncRNA und ihre Einordnung im Evolutionsprozess (Phylogonie) zum Ziel. Daraus können Gemeinsamkeiten und Unterschiede zwischen verschiedenen Spezies und damit des Expressionsprozesses abgeleitet werden.

Die Abbildung 10.5 zeigt Metadaten-Modelle, die in diesem Analysebereich typischerweise zur Anwendung kommen. Ausgangspunkt bilden private, lokale Dateien, die interessante und bereits vorselektierte ncRNA für verschiedene Spezies allgemein oder differenziert nach ihrem jeweiligen Typ enthalten. Diese ncRNA werden auf die bekannten Gene im Genom abgebildet.

Die Komposition dieses Mappings mit Mappings zwischen Genen und Transkripten, nicht kodierenden Regionen (so genannte Introns - vgl. Kapitel 3) oder homologen Genen in anderen Spezies ermöglicht eine gezielte Auswertung der gegebenen Menge von ncRNA. Diese Mappings und resultierenden Objekte sind Teil der Datenquelle Ensembl sowie weiterer Quellen, die beispielsweise die Genomsequenz oder die Genhomologien beinhalten. In keiner dieser Quellen waren nicht kodierende Regionen von Genen vorhanden; sie sind zwischen den Exon-Abschnitten eines Gens lokalisiert und können aus beiden, den Gen- und Exon-Annotationen berechnet werden. Um eine ausreichende Performanz zu erreichen, wurden sie im Vorfeld der Analyse berechnet und zusätzlich in der lokalen Kopie der Datenquelle Ensembl materialisiert. Ausgewählte Ergebnisse der durchgeführten Analysen, die mit anderen Analysetools weiterverarbeitet und analysiert wurden, werden in [LKS07] dargestellt.

10.6 Zusammenfassung

Im Mittelpunkt dieses Kapitels stand das *BioFuice*-System, um Daten aus dezentralisierten privaten und öffentlichen Quellen sowie Ontologien zu integrieren. *BioFuice* erweitert den Peer-to-Peer-artigen *iFuice*-Ansatz für Bioinformatik-Anwendungen. *BioFuice* unterstützt eine explorative Analyse auf Basis einer umfangreichen Benutzer-Schnittstelle, mit der strukturierte Anfragen bis hin zu unstrukturierten Suchanfragen spezifiziert werden können, aus denen anschließend automatisch *iFuice*-Skripte erzeugt werden. Die Ergebnisdaten können in verschiedenen dateibasierten Formaten gespeichert werden. Darüber hinaus wird die Kopplung von *BioFuice* mit der statistischen Software R über das eigens entwickelte *RiFuice*-Paket vollzogen, so dass die mit *BioFuice* integrierten Daten direkt in den Analysen unter Nutzung von R verwendbar werden. *BioFuice* wurde in verschiedenen Analyseszenarien der Bioinformatik angewendet, die beispielsweise in der Genexpressionsanalyse, der Analyse von Proteininteraktionen und der Untersuchung von ncRNA bestehen.

Kapitel 11
Verwandte Integrationsansätze

11.1 Überblick

Eine Übersicht repräsentativer Ansätze und Systeme zur Integration von öffentlich verfügbaren Datenquellen im Bereich der Lebenswissenschaften wird in [LC03, HK04] gegeben. Im Gegensatz zu diesen beiden Arbeiten, die eher einer informatiknahen Sichtweise folgen, zeigt [Ste03] aus biologischer Sicht einen Überblick über aktuelle Datenintegrationsansätze sowie ausgewählte Systeme. Auch in [LN07] werden ausgewählte Integrationsansätze und -systeme vorgestellt, die im Bereich der Bioinformatik zur Anwendung kommen.

Eine mögliche Klassifikation von Ansätzen und Systemen zur Datenintegration wurde in Kapitel 2 vorgestellt. Dieser Klassifikation folgend zeigt die Abbildung 11.1 eine Einordnung von ausgewählten Datenintegrationsansätzen und -systemen, die mit den in den Kapiteln 8, 9 und 10 beschriebenen Integrationsansätzen Gemeinsamkeiten aufweisen. Annonda [PC05] und Bis (Biological Integration System) [LBE03] verwenden eine typische Mediator-Architektur, um Daten im Bereich der Bioinformatik (z.B. zur Genexpressionsanalyse) zu integrieren. In beiden Systemen kommt ein applikationsspezifisches globales Schema zur Anwendung, wobei in Bis sich das globale Schema aus den Schemata der Quellen und deren Verbindungen ergibt. Darüber hinaus propagiert Bis das Konzept von abgeleiteten Wrap-

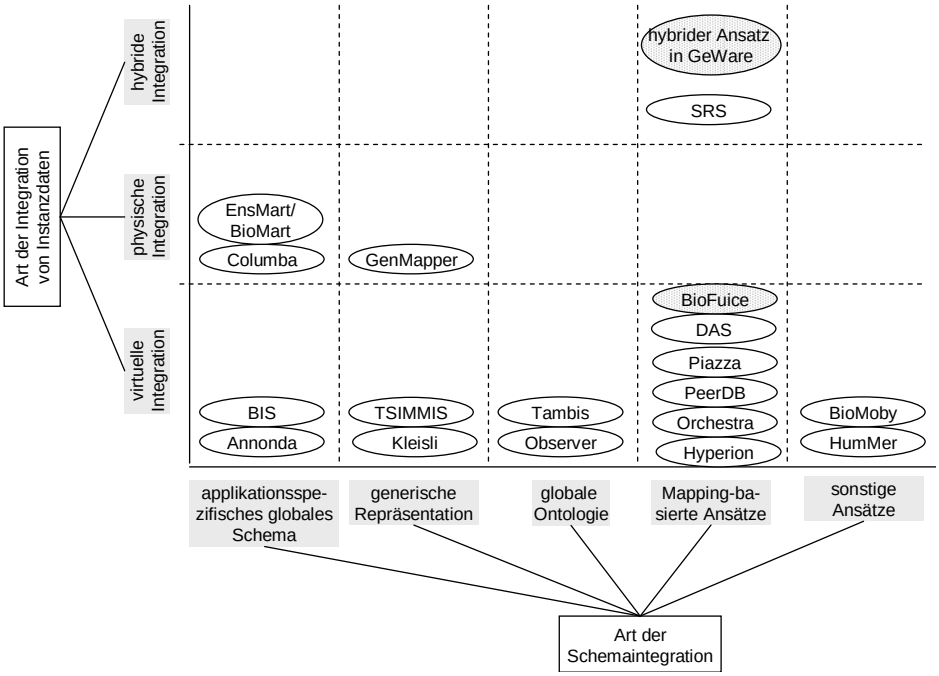


Abbildung 11.1: Einordnung von verwandten Integrationsansätzen

pern, mit denen die Ergebnisse spezieller Anfragen an Datenquellen sowie von Applikationen integriert werden können. EnsMart/BioMart [KKS+04] und Columba [RMT+04] verwenden ebenso ein applikationsspezifisches globales Schema. Jedoch verfolgen beide Systeme eine physische Integration von verschiedenen molekularbiologischen Objekten (EnsMart/BioMart) und Proteinen (Columba) in einer eigenen Datenquelle (Data Warehouse). In dieser Datenquelle werden die Daten der verschiedenen Objekttypen (EnsMart/BioMart) und Quellen (Columba) um einen zentralen Objekttyp bzw. eine zentrale Datenquelle multidimensional gespeichert.

Kleisli [Won00, CCW03] und Tsimmis [GHI+95, GMPQ+04] verfolgen einen Mediator-basierten Ansatz, zur Integration von Datenquellen und Applikationen. In beiden werden die Daten intern generisch repräsentiert; Kleisli verwendet ein komplexes Objekttyp-System, während Tsimmis auf dem OEM (Object Exchange Model) basiert. Tsimmis unterstützt eine Fusion von Daten aus verschiedenen Quellen [PAG96]. Anfragen werden in den speziell entwickelten Sprachen Collection Programming Language (Kleisli) und OEM-QL (Tsimmis) formuliert. GenMapper [DR04] verfolgt eine physische Integration von molekularbiologischen Daten auf Basis des GAM (Generic Annotation Model), mit dem die integrierten Daten generisch in einer neuen Datenbank repräsentiert werden. Zur Optimierung der Anfrageverar-

beitung werden komplexe Annotationssichten aus der GAM-Repräsentation materialisiert, die hiernach direkt zur Beantwortung von Anfragen genutzt werden können.

Observer [MIKS00] und Tambis [BGB+99, GSN+01, SGP+03] verwenden globale Ontologien, um einen transparenten Zugriff auf Datenquellen zu gewährleisten. Konzepte der globalen Ontologie sind mit den Instanzen der Quellen assoziiert. Anfragen werden unter Nutzung der Konzepte der globalen Ontologie formuliert (z.B. in Form von Beschreibungslogik). Je nach Umfang und Anzahl der integrierten Datenquellen kann die globale Ontologie sehr umfangreich sein; die Ontologie von Tambis enthält ca. 1800 Konzepte [LN07].

Das Distributed Annotation System (DAS) [DJD+01, PBC+06] und das Sequence Retrieval System (Srs) [EA93, EUA96, EHB03] sind Systeme, die eine Mapping-basierte Integration verfolgen. Das Das ermöglicht die Integration von dezentral, von verschiedenen Forschungsgruppen entwickelten Datenquellen, auf die unter einem einheitlichen GUI, dem Genome-Viewer, zugegriffen werden kann. Im Gegensatz zu anderen Ansätzen werden alle Objekte (Gene, Transkripte, Exons etc.) auf ihre Sequenzregion zurückgeführt. Objekte sind assoziiert, wenn sie die gleiche Sequenzregion teilen (oder einschließen). Damit werden semantische Konflikte (z.B. Gendefinition zweier Quellen) vermieden. Jedoch verlangt dieser Ansatz eine Neuannotierung oder zumindest eine Anpassung bzw. Neuberechnung der Angaben der zugeordneten Sequenzregion, sobald eine veränderte Genomsequenz vorliegt. Im Gegensatz zur virtuellen Integration der beiden vorgenannten Systeme verfolgt Srs eine hybride Strategie. Die zu integrierenden Quellen müssen zumindest als Kopie lokal vorliegen. Eine mengenbasierte Navigation kann unter Nutzung von den in den Quellen vorhandenen Objektkorrespondenzen erfolgen.

Eine Mapping-basierte Integration wird auch von Piazza [HIMT03], PeerDB [NOTZ03, OTZ+03], Orchestra [IKKC05] sowie von Hyperion [AKK+03, RGKG+05] verfolgt. Diese Systeme integrieren Datenquellen P2P-artig. PeerDB und Orchestra basieren jeweils auf einem P2P-System, in dem die Datenquellen die Peers darstellen, zwischen denen Anfragen und Anfrageresultate ausgetauscht werden. Piazza ist ein PDMS, in dem auf Basis von Schema-Mappings Anfragen reformuliert werden, wenn sie zwischen Quellen ausgetauscht werden. Hyperion verwendet Mapping-Ausdrücke und Mapping-Tabellen (vgl. [KAM03]), um Anfragen zu transformieren und Datenquellen miteinander zu verbinden.

Im Folgenden werden die Ansätze und Systeme Columba, GenMapper und Srs sowie das BioFast-Projekt auf Grund ihrer Beziehung zu den in dieser Arbeit beschriebenen Ansätzen und Systemen näher charakteri-

siert. Darüber hinaus wird eine Abgrenzung zum hybriden Integrationsansatz und/oder *BioFuice* gegeben.

11.2 Columba

Columba [RMT+04] verfolgt einen physischen Integrationsansatz, um Proteinannotationen aus verschiedenen Quellen (u.a. PDB, SwissProt, SCOP, CATH, DSSP) mit Publikationsdaten (u.a. PubMed) und Ontologien (u.a. GeneOntology) in einer lokalen Datenbank zusammenzuführen. Bei der Konstruktion des globalen Schemas werden die Schemata der Quellen weitestgehend übernommen, um den Aufwand der Schemaintegration gering zu halten. Dabei wird eine Datenquelle, die Protein Data Bank (PDB), als zentrale Quelle verwendet; die anderen Quellen sind mit dieser zentralen Quelle jeweils mit einem Mapping verbunden. Damit können Join-Operationen zwischen zwei beliebigen Datenquellen günstig über die zentrale Quelle berechnen werden.

Abgrenzung zum hybriden Integrationsansatz[64]

Ähnlich wie die Mapping-Datenbank des hybriden Integrationsansatzes sind die integrierten Datenquellen in Columba sternförmig um eine zentrale Quelle organisiert. Dazu materialisiert die Mapping-Datenbank die aus den Datenquellen extrahierten Mapping-Daten; die anderen Daten werden virtuell und bedarfsgesteuert aus den lokalen Kopien der Quellen integriert. Dagegen übernimmt Columba die ausgewählten Quellen und speichert sie physisch in einer gemeinsamen Datenbank. Diese Datenbank ist Grundlage für alle Auswertungen in Columba. Der hybride Integrationsansatz verwendet SRS für einen Zugriff auf die Datenquellen.

Zur Integration der Datenquellen werden in Columba Mappings eingesetzt. Anstatt wie in Columba jeweils nur ein Mapping zwischen zwei Quellen zu übernehmen oder vorauszuberechnen, unterstützt der hybride Integrationsansatz alternative Mappings für unterschiedliche Join-Pfade.

Begünstigt durch die klare Fokussierung auf die Integration von Proteindaten, bindet Columba verschiedene Analysetools in die Web-Schnittstelle ein. So werden beispielsweise auf Basis des JMol-Viewer [RJSS00] Proteinstrukturen in 3D visualisiert. Eine solche Einbindung von externen Visualisierungs-Tools wird derzeit mit dem hybriden Integrationsansatz nicht verfolgt.

[64] Auf eine Abgrenzung zu *BioFuice* wird auf Grund der starken Unterschiedlichkeit beider Ansätze verzichtet.

11.3 GenMapper

GenMapper [DR04] verwendet ein generisches Datenmodell, das Generic Annotation Model (GAM), um Annotationsdaten verschiedener Quellen physisch zu integrieren. Im GAM werden sowohl die Beziehungen zwischen Objekten einer Quelle, wie sie beispielsweise bei Ontologien auftreten, als auch zwischen den Objekten verschiedener Quellen gespeichert. Mit Hilfe von abstrakten Operatoren werden aus der GAM-Repräsentation Annotationssichten für unterschiedliche Analysezwecke generiert.

Abgrenzung zum hybriden Integrationsansatz

Mit dem GAM bleibt GenMapper auf Mappingdaten beschränkt. Komplexere Daten, z.B. geometrische Daten der Proteinfaltung oder Genomsequenzen, werden nicht mit einbezogen. Dagegen verwendet der hybride Integrationsansatz lokale Kopien der zu integrierenden Datenquellen, deren Daten in vollem Umfang zur Beantwortung von Anfragen zur Verfügung stehen. Die Mapping-Daten werden ebenso wie bei GenMapper in einer Mapping-Datenbank materialisiert. Obwohl Ziel und Funktion der Mapping-Datenbank und GenMapper sich sehr ähnlich sind, unterscheiden sich beide Ansätze in ihrem Aufbau. Während die Mapping-Datenbank des hybriden Integrationsansatzes ein Stern-Schema verwendet, das bei Bedarf um weitere Mapping-Tabellen erweitert werden kann, verfolgt GenMapper mit dem GAM eine generische Repräsentation von Objekten und deren Beziehungen. Um den evtl. Performanzeinbußen des generischen Schemas entgegenzuwirken, können erstellte Mapping-Kompositionen in der GenMapper-Datenbank materialisiert werden. Dagegen wird eine Mapping-Komposition in der Mapping-Datenbank des hybriden Ansatzes mit jeder neuen Anfrage erneut ausgeführt; mit dem gewählten Stern-Schema wird höchstens eine Mapping-Komposition notwendig.

Abgrenzung zu *BioFuice*

Ebenso wie der vorgestellte hybride Integrationsansatz ermöglicht *BioFuice* die Integration und damit den Zugriff auf komplexere Daten (s.o.), was mit GenMapper nicht oder nur sehr eingeschränkt möglich ist. Ferner werden die Daten mit *BioFuice* nicht vorab physisch (wie mit GenMapper), sondern bedarfsgesteuert zur Laufzeit der Anfrage integriert. Die Anfrageergebnisse werden von *BioFuice* in einem Cache zwischengespeichert, so dass ähnlich zu den erzeugten Annotationssichten im GenMapper kurze Antwortzeiten für die erneute Ausführung der Anfrage (oder ähnlicher Anfra-

gen) zu erwarten sind. Für Anfragen verwendet *BioFuice* u.a. die *iFuice*-Skriptsprache, mit der operatorgesteuert eine Integration vorgenommen werden kann. GenMapper benutzt ebenso Operatoren zur Datenintegration, die jedoch nicht in Skripts zusammengefasst werden können, sondern über ein GUI implizit aufgerufen und ausgeführt werden.

Gegenüber *BioFuice* wird die Semantik der materialisierten Objekte und Mappings im GenMapper nicht explizit gespeichert und verbleibt damit in der Verantwortung des Benutzers. Darüber hinaus kann GenMapper lediglich ein direktes Mapping zwischen zwei Datenquellen verwalten, da ein Mapping im GAM implizit alle verfügbaren Objektkorrespondenzen zwischen zwei Quellen umfasst; eine explizite Erfassung des Mappings ist nicht möglich. In *BioFuice* können beliebig viele direkte Mappings zwischen zwei Quellen existieren, die sowohl im Umfang ihrer Objektkorrespondenzen als auch hinsichtlich ihres Namens und Typs abgegrenzt sind.

11.4 Sequence Retrieval System

Das Sequence Retrieval System (Srs) [EA93, EUA96, EHB03] verfolgt eine Kopplung lokal installierter und vor allem dateibasierter Datenquellen; für den Zugriff auf relationale Datenbanken ist ein Zusatzprodukt notwendig. Der Zugriff auf Datenquellen erfolgt unter Nutzung von Wrappern, die in einer umfangreichen Wrapper-Bibliothek zur Verfügung gestellt werden. Die Bibliothek wird sowohl herstellerseitig als auch von Anwendern gepflegt und erweitert, so dass Wrapper für eine Vielzahl von Datenquellen in hoher Qualität bestehen. Die quellenspezifischen Wrapper werden unter Nutzung der eigens entwickelten Sprache *Icarus* erstellt und enthalten sowohl das Schema der Quelle als auch eine Attributliste, für die eine Indizierung durchgeführt wird. Die resultierenden Indizes werden zur Anfrageverarbeitung verwendet.

Abgrenzung zum hybriden Integrationsansatz

Mit der hybriden Integrationslösung in Kapitel 8 wird den funktionalen Limitierungen von SRS begegnet. Eine solche Limitierung besteht darin, dass SRS in der verwendeten Version (7.3.1) ausschließlich Anfragen an eine Datenquelle unterstützt. Damit sind Filterbedingungen (Selektion) nur für Attribute einer Quelle möglich. Mit der hybriden Integrationslösung können dagegen unabhängig von der Zielquelle, von der die relevanten Objekte in der Ergebnismenge enthalten sein sollen, beliebige Attribute verschiedener Datenquellen in eine Selektionsanfrage einbezogen werden. Ferner unterstützt die hybride Integrationslösung eine gleichzeitige Projektion von Attributen un-

terschiedlicher Datenquellen, was mit SRS in dieser flexiblen Art noch nicht möglich ist. Während der Query-Mediator für eine Selektionsanfrage durch geschickte Umformulierung und Nutzung der Mapping-Datenbank eine SRS-Anfrage erzeugt, müssen für Projektionsanfragen mehrere SRS-Anfragen erstellt und verarbeitet werden, um die notwendige Flexibilität zu bieten.

Weitere Limitierungen von SRS bestehen bei der mengenbasierten Traversierung entlang von Navigationspfaden. Zwar bildet SRS eine Objektmenge (Objekte einer Datenquelle) auf die Objekte anderer Datenquellen ab. Jedoch wird zur Eingabemenge ausschließlich eine Ausgabemenge erzeugt; eine Rückgabe von Objekt-Korrespondenzen zwischen der Ein- und Ausgabemenge unterbleibt. Dieser Informationsverlust wirkt sich besonders in Projektionsanfragen nachteilig aus, z.B. wenn zu der aus einer Analyse resultierenden Menge von Probesets Attribute verschiedener Datenquellen angezeigt werden sollen. Dies setzt eine Zuordnung der Attributwerte zu den einzelnen Probesets voraus. Der Query-Mediator behebt diesen Nachteil durch die Nutzung der Objektkorrespondenzen in der Mapping-Datenbank und internen Datenstrukturen (Hash-basierte Speicherstrukturen), die einen flexiblen und schnellen Zugang gewährleisten.

Darüber hinaus verwendet SRS zur Traversierung einer Objektmenge zwischen Datenquellen, die über kein direktes Mapping miteinander verbunden sind, den kürzesten Weg (sofern nicht ein Pfad explizit spezifiziert wurde) zwischen den Quellen. Das kann eine reduzierte Datenqualität der Ergebnismenge zur Folge haben, z.B. wenn die Datenquellen und Mappings entlang des Pfades veraltet sind. Die Mapping-basierte Integrationslösung verwendet dagegen explizit in der Mapping-Datenbank gespeicherte Mappings, die der Benutzer zur Formulierung von Anfragen auswählt; damit bleibt die vorgenommene Mapping-Komposition transparent.

Im Gegensatz zu SRS bleibt der vorgestellte Mapping-basierte Ansatz auf die Komposition der verfügbaren Mappings in der Mapping-Datenbank beschränkt. Zur Verwendung neuer Mappings bedarf es der Extraktion der Objektkorrespondenzen aus den Quellen und deren Import in die Mapping-Datenbank. Ferner wird für das Mapping von Objekten zweier Datenquellen, die beide nicht die zentrale Quelle sind, eine Mapping-Komposition (Source → Center → Source) notwendig. Damit bleiben direkte Mappings zwischen zwei Datenquellen bei der Anfrageverarbeitung unberücksichtigt, was evtl. in einer schlechteren Datenqualität des aus der Komposition resultierenden Mappings (gegenüber dem direkten Mapping) münden kann.

Abgrenzung zu *BioFuice*

SRS verfolgt eine sehr lose Form der Datenintegration [LN07]. Die Objekte der lokal installierten Quellen (bzw. Kopien von Quellen) sind implizit mit einem semantischen Typ assoziiert. Quellen mit Objekten unterschiedlicher semantischer Typen, die den logischen Datenquellen in *BioFuice* entsprechen, werden nicht unterstützt; sie müssen in mehrere typspezifische Partitionen aufgeteilt werden. Ähnlich wie *BioFuice* greift SRS auf Mengen von Objektkorrespondenzen (Mappings) zurück, um die Objekte verschiedener Quellen aufeinander abzubilden. Im Gegensatz zu *BioFuice* wird einem Mapping jedoch weder ein Name noch ein semantischer Typ zugeordnet. Damit kann zwischen zwei Datenquellen lediglich ein Mapping bestehen, dessen Semantik im Verantwortungsbereich des Benutzers verbleibt. Eine semantische Datenintegration wird daher ebenso wie eine Datenfusion von SRS nicht verfolgt.

Ähnlich wie *BioFuice* kann SRS Anfragen nur an eine Datenquelle stellen, deren Ergebnismenge auf Objekte anderer Quellen abgebildet werden kann. Beide Ansätze nutzen eine proprietäre Anfragesprache[65]. *BioFuice* führt Anfragen und Mappings operatorgesteuert aus, die in Skripts kombiniert werden können. Dagegen erfolgt die Anfrageformulierung in SRS unter Nutzung spezieller Ausdrücke. Diese Ausdrücke beziehen sich im Wesentlichen auf die Angabe von Bedingungen und der Ausführung von Mappings unter Verwendung der erzielten Ergebnisse von Anfragen. Spezielle Operationen, wie sie etwa *BioFuice* mit der Aggregation zur Datenfusion u.a.m. verfolgt, werden von der Anfragesprache in SRS nicht unterstützt. Darüber hinaus bleibt SRS im Unterschied zu *BioFuice* auf die Integration von lokal installierten Quellen (oder Kopien) begrenzt.

Die von SRS zur Integration verwendeten Wrapper basieren auf der proprietären Sprache *Icarus*. Dagegen nutzt *BioFuice* XML zur Beschreibung von Mappings, wobei Standard-Sprachen wie SQL und XQuery zur Ausführung der Mappings eingesetzt werden. Das erleichtert einerseits eine mögliche Modifikation einer Mapping- bzw. Quellenbeschreibung/Wrapper, da der Benutzer keine neue Anfragesprache erlernen muss. Andererseits kann bei Mappings, die die generischen Ausführungsdienste RDBMS und XML-DBMS nutzen, mit dem Einsatz von Datenbanktechniken (z.B. Indizierung, materialisierte Sichten) eine Performanzsteigerung erreicht werden, sofern ausreichend Zugriffsrechte vorhanden sind und das zugrunde liegende DBMS die Techniken unterstützt.

[65] Eine Gegenüberstellung beider Sprachen wird hier nicht verfolgt.

11.5 Das BioFast-Projekt

Das BioFast-Projekt [BLM+04, LMNR04b, LMNR04a] hat den Aufbau einer Infrastruktur zum Ziel, mit deren Hilfe ein Benutzer eine explorative Analyse effizient durchführen kann. Verschiedene Probleme, z.B. Anfragesprachen zur Datenexploration, Semantik und Eigenschaften von Mappings und -kompositionen sowie Anfrageoptimierung, werden in diesem Projekt untersucht bzw. adressiert. Grundlage bildet ein Integrationsszenario, in dem die Datenquellen, ihre enthaltenen Objekte sowie ihre Beziehungen untereinander auf physischer und logischer Ebene in verschiedenen Graph-Strukturen repräsentiert werden. Zu diesen Strukturen zählt ein Quellen-Graph (physische Ebene), der relevante Datenquellen als Knoten und Mappings zwischen ihnen als Kanten enthält. Auf logischer Ebene werden die Objekte der Datenquellen mit einem semantischen Typ assoziiert. Diese semantischen Objekttypen sind mit typgleichen Korrespondenzmengen verbunden und finden in einem logischen Graphen als Knoten (Objekttypen) und Kanten (Typ von Korrespondenzmengen) Eingang. In [LMNR04b, LMNR04a] sowie [BKN+06] werden spezielle Eigenschaften der Graphen in Bezug auf die o.g. Probleme untersucht.

Abgrenzung zu *BioFuice*[66]

BioFuice basiert auf zwei Metadaten-Modellen, einem Source-Mapping-Modell (SMM) und einem Domänenmodell, die ebenso wie die Untersuchungen im BioFast-Projekt auf Graph-Strukturen aufbauen (vgl. Kapitel 9). Während ein Quellen-Graph (physische Ebene) in *BioFuice* zwar existiert (bzw. aus dem SMM abgeleitet werden kann) aber keine explizite Verwendung findet, ist der logische Graph im BioFast-Projekt dem Domänenmodell von *BioFuice* sehr ähnlich. In beiden Graphen (auch das Domänenmodell ist ein Graph) werden die semantischen Objekttypen eines Integrationsszenarios als Knoten repräsentiert, die auf Basis spezieller Typen von Mappings miteinander verbunden sind. Im Gegensatz zum BioFast-Projekt steht eine Anfrageoptimierung durch die geschickte Auswahl von Mappings und Mapping-Pfaden (Mapping-Kompositionen) nicht im Vordergrund. Vielmehr wurde die *iFuice*-Skriptsprache sowie verschiedene GUI entwickelt, mit der Anfragen formuliert und anschließend ausgeführt werden können.

[66] Eine Abgrenzung zur hybriden Integration unterbleibt auf Grund der Unterschiedlichkeit beider Ansätze.

11.6 Zusammenfassung

Mit diesem Kapitel wurde ein Überblick zu verwandten Ansätzen und System gegeben. Dazu wurde die in Kapitel 2 vorgestellte Klassifikation verwendet, in die die Ansätze und Systeme eingeordnet wurden. Darüber hinaus charakterisierte das Kapitel die Integrationsansätze und -systeme COLUMBA, GENMAPPER und SRS sowie das BIOFAST-Projekt näher. Zu diesen Ansätzen und Systemen wurden die wesentlichsten Gemeinsamkeiten und Unterschiede zum hybriden Integrationsansatz und *BioFuice* diskutiert.

Teil IV
Zusammenfassung

Dieser Teil beschließt die Arbeit. Wichtige Beiträge und Erkenntnisse dieser Dissertation werden in einer Zusammenfassung aufgezeigt. Ein Ausblick stellt Erweiterungen, Verbesserungen und Anwendungen der in dieser Arbeit enthaltenen Datenintegrationsplattformen vor, die lohnenswert erscheinen, weiter verfolgt zu werden.

Kapitel 12
Fazit und Ausblick

12.1 Fazit und Beitrag der Arbeit

Eine Datenintegration ist von grundlegender Bedeutung für viele Bereiche der Wirtschaft, Wissenschaft und Forschung. In der Bioinformatik ermöglicht sie, komplexe und übergreifende Analysen und ist Ausgangspunkt für deren effiziente Durchführung. Im Fokus dieser Dissertation standen Ansätze und Plattformen für die Integration molekularbiologischer und klinischer Daten, die einerseits durch verschiedene Laborexperimente erzeugt werden und andererseits Inhalt von lokalen und öffentlichen Datenquellen sowie Ontologien sind. Insbesondere lieferte diese Arbeit die folgenden Beiträge.

Data-Warehouse-basierte Integrations- und Analyseplattform *GeWare*

Auf Basis einer durchgeführten Evaluierung Microarray-basierter Datenbanken wurde das Genetic Data Warehouse (*GeWare*) als Integrations- und Analyseplattformen konzipiert und entwickelt. *GeWare* speichert zentral sowohl große Mengen an experimentellen Rohdaten, die unter Nutzung von Hochdurchsatz-Technologien, wie z.B. Microarrays und Matrix-CGH-Arrays, erzeugt werden, als auch normalisierte Daten und Analyseergebnisse, die in umfangreichen Genexpressions- und Mutationsanalysen entstehen und Ver-

wendung finden. Dazu verwendet *GeWare* ein multidimensionales Schema bestehend aus mehreren hierarchisch organisierten Dimensionen, die die numerischen Genexpressions- und Mutationsdaten (Fakten) beschreiben. Das Modell unterstützt zielgerichtete Analysen und kann bei Bedarf um neue Dimensionen und Fakten flexibel erweitert werden. Die Analyseplattform verfügt über verschiedene Methoden, um die Microarray-basierten Expressionsdaten einer Vorverarbeitung (z.B. Normalisierung) zu unterziehen und hiernach ebenso wie die Matrix-CGH-Daten zu analysieren und zu visualisieren. Die Analysen, Visualisierungen und parametrisierten Berichte verwenden Gen-/Clone- und Treatment-Gruppen sowie Genexpressions- und Matrix-CGH-Matrizen als Eingabe, die zentral von *GeWare* verwaltet werden und Ergebnis einer vorangegangenen Analyse sein können. Damit wird eine fokussierte und iterative Analyse ermöglicht.

Ein weiterer Schwerpunkt der Plattform liegt in der flexiblen Integration von Metadaten, die einerseits durch den Experimentator manuell mit dem Ziel erfasst werden, das Experiment aus aufbau- und ablauforganisatorischer Sicht konsistent und umfassend zu beschreiben. Andererseits können klinische Daten, die beispielsweise in klinischen Studien erhoben und in einem Studienverwaltungssystem gespeichert werden, in die Plattform eingebracht und in den Analysen verwendet werden. Die einheitliche Erfassung der experimentellen Metadaten wird durch so genannte *Annotation Templates* sichergestellt, die einen definierten Umfang von Kategorien in hierarchisch angeordneten Seiten beinhalten. Den Kategorien können dabei Werte aus kontrollierten Vokabularen zugewiesen werden, die einen abgegrenzten Wertebereich definieren und somit eine konsistente Verwendung sicherstellen helfen. Auf Basis der spezifizierten experimentellen Annotation und der importierten klinischen Daten können Treatment-Gruppen sowie Genexpressions- und Matrix-CGH-Matrizen erstellt werden, die den Ausgangspunkt für eine fokussierte Analyse der Expressions- und Mutationsdaten darstellen.

Die *GeWare*-Plattform wurde von verschiedenen medizinischen und biologischen Anwendern zur Datenverwaltung und -analyse ihrer Genexpressionsdaten verwendet. Darüber hinaus findet die Plattform in zwei deutschlandweiten klinischen Studien mit dem Ziel der Erforschung von molekularen Mechanismen verschiedener Krebsarten Anwendung. Das System beinhaltet derzeit Daten von mehr als 2.100 Microarrays.

Hybride Integration öffentlicher Annotationsdaten

Um öffentlich verfügbare Daten zu molekularbiologischen Objekten wie Genen, Proteinen und deren Funktionen in den Analysen verfügbar zu machen, wurde ein hybrider Integrationsansatz konzipiert und entwickelt. Nach die-

sem sind die experimentellen Daten im *GeWare*-Data-Warehouse physisch gespeichert, während die öffentlichen Annotationsdaten über einen Mediator bedarfsgesteuert abgerufen werden. Für die einheitliche Anbindung der Datenquellen wird die kommerzielle Software SRS eingesetzt, die durch den Mediator und unter Nutzung einer Mapping-Datenbank mit der *GeWare*-Plattform gekoppelt ist. Die Mapping-Datenbank enthält die Beziehungen zwischen den Instanzen der öffentlichen Datenquellen, die dort oftmals als Web-Links repräsentiert sind.

Das besondere Merkmal des Integrationsansatzes besteht darin, dass die Datenquellen sternförmig um eine zentrale Quelle angeordnet sind. Dieser Ansatz wird in der Mapping-Datenbank mit einem sternförmigen Schema repräsentiert, mit dem sowohl direkte Mappings zwischen der zentralen Quelle und den Datenquellen als auch Mapping-Komposition gespeichert werden können. Zudem sind alternative Mappings und somit multiple Mapping-Pfade zwischen zwei Quellen möglich. Mit diesem Aufbau der Mapping-Datenbank wird eine flexible und dennoch effiziente Berechnung der Join-Operationen durch a) Benutzer-wählbare Mapping-Pfade und b) der Vorberechnung ausgewählter Mapping-Komposition der Quellen zu einer zentralen Quelle erreicht, wodurch Wege mit einer maximalen Länge von 2 garantiert werden.

Die Kopplung dieses Ansatzes mit der *GeWare*-Plattform ermöglicht es dem Benutzer einerseits, Gen- und Clone-Gruppen als Ergebnis einer Analyse (z.B. Suche) der mit diesem Ansatz integrierten Daten zu erstellen. Andererseits kann die Annotation der Gen- und Clone-Gruppen untersucht werden, die aus der Genexpressions- und CGH-basierten Mutationsanalyse resultieren. Derzeit werden mit diesem Ansatz die öffentlich verfügbaren Datenquellen GeneOntology, LocusLink und Ensembl integriert. Durchgeführte Performanzanalysen belegen die Leistungsfähigkeit des Ansatzes.

Mapping-basierte und semantische Integration mit *BioFuice*

Im Gegensatz zu üblichen Datenintegrationsansätzen, z.B. Data Warehouse, die ein globales Schema einsetzen, um eine einheitliche und konsistente Sicht über die unterschiedlichen heterogenen Quellen bereitzustellen, folgt *BioFuice* einer Peer-to-Peer-artigen Datenintegration. Damit können ad-hoc neue Datenquellen, z.B. die lokale Genliste, integriert und in eine Anfrage involviert werden. Das gelingt, in dem die einzelnen Datenquellen mit Hilfe von Mappings (auf Grundlage der Beziehungen zwischen den Instanzen zweier Quellen) verbunden werden, so dass für die Integration einer neuen Quelle lediglich ein Mapping zwischen den bereits integrierten Quellen und der neu zu integrierenden Quelle notwendig wird. Die Semantik von Quellen und Mappings repräsentiert ein Metadaten-Modell, das so genannte Domänen-

modell, mit dem abgegrenzte und benutzerspezifizierte Objekttypen sowie deren Beziehungen (Mappingtypen) untereinander modelliert werden. Anfragen und Mappings werden mit Hilfe von abstrakten Operatoren ausgeführt, die in Skripten zusammengefasst werden können.

BioFuice fußt auf dem *iFuice*-Ansatz und beinhaltet zahlreiche domänenspezifische Erweiterungen für den Bereich der Bioinformatik. Eine Schnittstelle zur frei verfügbaren statistischen Software R, die weit verbreitet für die Genexpressionsanalyse eingesetzt wird, stellt das R Paket *RiFuice* bereit. Mit dieser Erweiterung können Daten aus verschiedenen heterogenen Quellen in die statistischen Analysen einbezogen werden, die beispielsweise die Verteilung von als exprimiert erkannten Genen auf die assoziierten Funktionen und Prozesse (gemäß der GeneOntology) untersuchen. *BioFuice Query* unterstützt insbesondere eine explorative Datenanalyse und stellt verschiedene interaktive Abfrage- und Suchfunktionalitäten zur Verfügung. Darüber hinaus enthält BioFuice die Möglichkeit, die gesammelten und integrierten Daten in verschiedene und für die Bioinformatik spezifische Formate zu exportieren. So wird beispielsweise das Fasta-Format verwendet, um genomische Sequenzen zusammen mit frei wählbaren Annotationen in Dateien abzuspeichern.

BioFuice wurde in verschiedenen Analyseszenarien zur Datenintegration erfolgreich eingesetzt, die von der Genexpressionsanalyse, der Analyse von Proteininteraktionen bis zur Untersuchung verschiedener Typen von RNA-Sequenzen reichen und die Anwendbarkeit des Ansatzes belegen.

12.2 Ausblick

Neben den genannten Beiträgen, die in den entwickelten Ansätzen und Plattformen zur Integration molekularbiologischer Daten bestehen, soll diese Dissertation Verbesserungen und interessante Fragestellungen aufzeigen, die Inhalt von zukünftigen Arbeiten sein kann.

GeWare als technologieübergreifende Plattform

Derzeit integriert *GeWare* Daten von zwei experimentellen Technologien: Microarrays zur Untersuchung der Genexpression (fokussiert auf Microarrays der Fa. Affymetrix) und Matrix-CGH-Arrays zur Analyse der Mutation (Anzahl der Kopien). Die Analyse dieser Daten erfolgt noch getrennt, obwohl einerseits technologieübergreifende Fragestellungen (z.B. zur Validierung der gemessenen Werte und gewonnenen Ergebnisse aus der Anwendung der jeweils anderen Technologie) existieren und interessant erscheinen. Andererseits besteht ein aus biologischer Sicht begründbarer Zusammenhang zwi-

schen den Daten beider Technologien. Daher erscheint es lohnenswert, die ausgebrachten Daten beider Technologien zu verknüpfen. Das erfordert, den Zusammenhang zwischen Probesets (Affymetrix Microarrays) und Clonen (Matrix-CGH-Arrays) genauer zu untersuchen, so dass daraus ein Mapping zwischen den Instanzen beider Objekttypen abgeleitet werden kann. Mit der Integration des Mappings in die *GeWare*-Plattform sind im Anschluss technologieübergreifende Berichte, Visualisierungen und Analysen möglich, die noch einer genaueren Spezifikation bedürfen.

Andere experimentelle Technologien zur Untersuchung von Genexpression und Mutation, die ebenfalls auf der Hochdurchsatz-Technologie aufbauen, bestehen beispielsweise in Exon-, Tiling- und SNP-Arrays und sind bereits in Anwendung. Gegenüber den Expressions-Arrays und Matrix-CGH-Arrays wird mit ihnen ein noch größeres Datenvolumen erzeugt, so dass sich bei Integration dieser Daten die Vorteile der *GeWare*-Plattform weiter erschließen.

Scientific Workflow Management

Die Analysen im Bereich der Bioinformatik sind wie in vielen anderen Forschungsbereichen datengetrieben. Eine Datenintegration stellt eine notwendige Voraussetzung für effiziente Analysen dar, für die vielfältige Algorithmen zur Anwendung kommen können. Dazu sind die Algorithmen in verschiedenen Analysetools implementiert, die zum einen frei verfügbar oder eigens für die Analyse erstellt werden. Eine Zusammenfassung der einzelnen Schritte, die von der Datenintegration bis hin zur Ausgabe des Analyseergebnisses reichen, bleibt derzeit zumeist auf die manuelle Durchführung jedes einzelnen Schrittes oder die programmtechnische Kodierung des Analyseablaufes beschränkt. Dabei könnten gerade generische Workflow-Ansätze der Analyse zu mehr Flexibilität und kürzerer Entwicklungszeit verhelfen, wie sie beispielsweise von Kepler [ABJ+04, LLB+05, LLB+05] und Taverna [OAF+04, HWS+06] vorgeschlagen werden. Ebenso kann die skriptgesteuerte Ausführung von Anfragen und Mappings in *BioFuice* nicht nur Datenintegrationsaufgaben übernehmen, sondern zur Abarbeitung von komplexen Workflows verwendet werden, in denen die Datenintegration und -analyse miteinander kombiniert sind.

Ontologie-Matching

Ontologien dienen der semantischen Beschreibung und Klassifikation von Objekten. Obwohl viele Datenquellen im Bereich der Bioinformatik bereits mit Ontologien assoziiert sind, bleiben Beziehungen zwischen Ontologien bislang ungenutzt. Im Unterschied zu anderen Domänen, wie dem e-Business,

drücken solche Beziehungen zwischen Ontologien nicht immer eine Gleichheit aus. Vielmehr können Krankheiten zu molekularen Prozessen zugeordnet und somit auf abstrakter Ebene als Ursache-Wirkungs-Beziehungen dargestellt werden. Die Abbildungen zwischen Ontologien können mit verschiedenen Match-Ansätzen realisiert werden. Metadaten-basierte Ansätze [MB04, OCAM+04, BAB05], die ausschließlich Metadaten (Konzeptnamen, -beschreibungen etc.) der Ontologien verwenden, reichen zumeist nicht aus, um ein qualitativ hochwertiges Match-Ergebnis zu erzielen. Mit ihnen werden linguistische Beziehungen in semantische Assoziationen zwischen den Konzepten zweier Ontologien umgewandelt; jedoch basieren nicht alle semantischen Assoziationen auf linguistischen Beziehungen. Im Gegensatz dazu nutzen instanzbasierte Ansätze [KFD+03, KSB04, BB05] die Beziehungen der Konzepte zu assoziierten molekularbiologischen Objekten (Instanzen). Die Grundidee dieser Ansätze ist, dass Konzepte mit gleichen/ähnlichen Mengen assoziierter Instanzen in Beziehung stehen. Jedoch hängt die Qualität des erzielten Mappings einerseits vom Anteil der Konzepte je Ontologie ab, denen mindestens eine Instanz zugeordnet ist, und andererseits vom Grad der Überlappung der assoziierten Instanzmengen. Daher sind weitere elaborierte Lösungsansätze notwendig, die einerseits die Metadaten und Strukturen der Ontologien sowie die assoziierten Instanzdaten besser ausnutzen. Andererseits könnten kombinierte Match-Strategien (im einfachsten Fall Durchschnitt und Vereinigung) bessere Ergebnisse versprechen. Letztlich ist die Evaluierung der Match-Ergebnisse für große Ontologien ein noch nicht vollständig gelöstes Problem. Erste Lösungsansätze hierzu werden in [TKR07] für ein e-Business Szenario und in [KTR07] für den Bereich der Bioinformatik aufgezeigt.

Aufdeckung von Datenfehlern

Oftmals sind molekularbiologische Daten in den verschiedenen Quellen mit Fehlern behaftet oder unvollständig [LN07]. In der Bioinformatik sind Fehler nicht nur experimentell bedingt, d.h. sie entstehen nicht nur auf Grund der eingesetzten experimentellen Technik. Vielmehr ermöglicht die weitgehend manuelle Erfassung der Daten von verschiedenen dezentralen Forschergruppen Datenfehler, die ihre Erkenntnisse auf Grund von vorgenommenen Untersuchungen in Form von Beschreibungen darlegen, ohne dass eine begleitende oder nachträgliche Validierung stattfindet. Auch die Übernahme von Daten aus anderen Quellen beeinflusst die Datenqualität einer Datenquelle. Eine klassifizierende Aufstellung von möglichen Datenfehlern und ihren Ursachen geben [RD00, LN07].

Die Peer-to-Peer-artige Integration mit *BioFuice* ermöglicht es, gezielt vorhandene Annotationsdaten aus verschiedenen Datenquellen gegenüberzu-

stellen. Somit können insbesondere fehlende und gegensätzliche Werte in den einzelnen Quellen erkannt werden, das den Ausgangspunkt für eine Verbesserung der Datenqualität, z.B. in Zusammenarbeit mit dem Betreiber der Datenquelle, darstellt.

Anhang

A Evaluierte Microarray-Plattformen

Tabelle A.1: Evaluierte datenbankgestützte Systeme für Microarray-Daten

System	**vollständiger Systemname/Organisation/URL**
ArrayDB	Array Database National Human Genome Research Institute (NHGRI), USA `http://genome.nggri.nih.gov/arraydb`
ExpressDB	Expression Database Harvard University, USA `http://arep.med.harvard.edu/ExpressDB`
GeneX	GeneX National Center for Genome Resources (NCGR), USA `http://genebox.ncgr.org/genex`
GIMS	Genome Information Management System University of Manchester, UK `http://www.cs.man.ac.uk/~norm/gims`
M-CHIPS	Multi-Conditional Hybridization Intensity Processing System Deutsches Krebsforschungszentrum (DKFZ), Deutschland `http://www.mchips.de`
RAD2	RNA Abundance Database Version 2 University of Pennsylvania, USA `http://www.cbil.upenn.edu/RAD2`
SMD	Stanford Microarray Database Stanford University, USA `http://genome-www4.stanford.edu/MicroArray/SMD`
YMD	Yale Microarray Database Yale University, USA `http://info.med.yale.edu/microarray`

B Daten zur sequenzbasierten Analyse von Oligo-Intensitäten

B.1 Verwendete Microarray-Daten im Überblick

Die folgende Tabelle B.2 gibt einen Überblick über die Microarrays des Latin-Square-Experiments, das von der Fa. Affymetrix durchgeführt wurde. Alle Microarrays verwenden den Chiptyp HG-U133A. Die Daten und weitere Informationen sind auf der Webseite `http://www.affymetrix.com/support/technical/sample_data/datasets.affx` (letzter Zugriff: 20.10.2006) öffentlich zugänglich. Die in der zugegebenen Mixtur enthaltenen unterschiedlichen Konzentrationen von Genen werden hier nicht wiedergegeben. Eine solche Aufstellung ist Bestandteil der von Affymetrix zur Verfügung gestellten Beschreibung des Latin-Square-Experiments.

Tabelle B.2: Überblick zu den Microarrays des Latin-Square-Experiments

Replikatgruppe	**Microarray Bezeichnungen**
Gruppe 1	Expt01R1, Expt01R2, Expt01R3
Gruppe 2	Expt02R1, Expt02R2, Expt02R3
Gruppe 3	Expt03R1, Expt03R2, Expt03R3
Gruppe 4	Expt04R1, Expt04R2, Expt04R3
Gruppe 5	Expt05R1, Expt05R2, Expt05R3
Gruppe 6	Expt06R1, Expt06R2, Expt06R3
Gruppe 7	Expt07R1, Expt07R2, Expt07R3
Gruppe 8	Expt08R1, Expt08R2, Expt08R3
Gruppe 9	Expt09R1, Expt09R2, Expt09R3
Gruppe 10	Expt10R1, Expt10R2, Expt10R3
Gruppe 11	Expt11R1, Expt11R2, Expt11R3
Gruppe 12	Expt12R1, Expt12R2, Expt12R3
Gruppe 13	Expt13R1, Expt13R2, Expt13R3
Gruppe 14	Expt14R1, Expt14R2, Expt14R3

B.2 Oligo-Intensitäten ausgewählter Äquivalenzklassen

Die folgende Tabelle B.3 zeigt beispielhaft die gemessenen Intensitäten von Oligos zweier ausgewählter Äquivalenzklassen, die sich hinsichtlich der Oligo-Sequenz ergeben. Die Daten basieren auf dem Microarray Expt01R1 des Latin-Square-Experiments, das unter Nutzung des Chiptyps HG-U95Av2 durchgeführt wurde.

Tabelle B.3: Gemessene Intensitäten äquivalenter Oligos

Klasse	Probeset-Name	Oligo-Nr.	PM Intensität	MM Intensität
1	31954_f_at	11	1.256,00	1.275,00
1	33680_f_at	12	1.217,00	1.272,00
1	31960_f_at	11	1.200,00	1.243,00
1	31953_f_at	11	1.248,50	1.242,30
1	37065_f_at	14	1.270,00	1.235,30
1	33671_f_at	12	1.216,00	1.216,00
1	31498_f_at	14	1.155,00	1.081,00
2	37065_f_at	2	945,50	330,00
2	31498_f_at	2	1.132,30	324,30
2	33680_f_at	2	1.121,80	296,00
2	31960_f_at	2	1.009,00	288,50
...				

Die Oligos einer Äquivalenzklasse besitzen die gleiche Oligo-Sequenzen. Die Tabelle B.4 beinhaltet die Oligo-Sequenzen für die in Tabelle B.3 dargestellten Äquivalenzklassen.

Tabelle B.4: Oligo-Sequenzen der ausgewählten Äquivalenzklassen

Klasse	Oligo-Sequenz
1	CAACACCTGAAGAAGGGGAACCAGC
2	CGGACTCTTTTTCCTCTACTGAGAT

B.3 Oligo-Häufigkeitsverteilung in Abhängigkeit von der Nukleotidanzahl

Chiptyp HG-U95Av2

Die folgende Tabelle B.5 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2a) für einen Affymetrix Microarray vom Typ HG-U95Av2.

Tabelle B.5: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U95Av2

Anzahl	**Anzahl Oligos für Nukleotid**			
Nukleotide	**A**	**C**	**G**	**T**
0	399	24	161	315
1	2.276	261	795	1.640
2	6.054	881	2.297	5.501
3	14.306	2.057	4.071	11.728
4	23.227	42.837	48.790	18.760
5	31.010	37.936	37.935	25.419
6	33.515	35.498	34.041	28.901
7	30.802	30.178	28.113	28.833
8	24.070	24.205	21.629	25.967
9	16.374	22.388	19.009	21.036
10	9.218	2.132	1.562	15.178
11	4.589	292	223	9.284
12	2.084	178	143	5.889
13	98	77	104	237
14	36	26	51	142
15	7	2	32	87
16	6	0	9	44
17	1	0	6	9
18	0	0	2	2
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

Chiptyp HG-U133A

Die folgende Tabelle B.6 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2b) für einen Affymetrix Microarray vom Typ HG-U133A.

Tabelle B.6: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U133A

Anzahl Nukleotide	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
0	73	127	637	407
1	893	959	3.842	2.446
2	5.220	4.048	11.261	7.626
3	17.104	11.633	21.856	16.171
4	34.541	25.331	32.979	25.491
5	48.645	40.028	38.146	32.876
6	49.867	52.127	38.134	35.240
7	41.049	52.982	33.264	33.846
8	26.415	38.890	26.254	29.419
9	14.178	17.853	18.838	23.523
10	6.565	3.631	11.062	17.511
11	2.383	323	6.384	11.516
12	781	22	3.165	6.689
13	200	8	1.389	3.203
14	45	3	530	1.401
15	6	0	180	473
16	0	0	41	104
17	0	0	3	23
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

Chiptyp HG-U133_Plus_2

Die folgende Tabelle B.7 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2c) für einen Affymetrix Microarray vom Typ HG-U133_Plus_2.

Tabelle B.7: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp HG-U133_Plus_ 2

Anzahl	**Anzahl Oligos für Nukleotid**			
Nukleotide	**A**	**C**	**G**	**T**
0	154	360	1.893	884
1	1.953	2.576	10.672	5.063
2	11.734	11.332	31.023	15.855
3	38.275	33.065	58.972	34.181
4	78.290	68.773	85.307	55.450
5	11.2328	104.102	96.985	74.333
6	119.879	128.898	94.209	83.943
7	102.086	124.917	78.942	84.611
8	69.803	86.720	60.133	76.896
9	39.476	36.167	41.538	62.640
10	19.244	6.697	21.457	46.781
11	7.556	610	12.451	31.155
12	2.635	30	6.310	18.097
13	692	8	2.840	8.902
14	133	3	1.075	3.893
15	17	0	360	1.234
16	3	0	84	279
17	0	0	7	58
18	0	0	0	3
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

Chiptyp MG-U74Av2

Die folgende Tabelle B.8 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2d) für einen Affymetrix Microarray vom Typ MG-U74Av2.

Tabelle B.8: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp MG-U74Av2

Anzahl Nukleotide	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
0	421	65	254	275
1	2.274	334	1.103	1.700
2	6.914	954	2.188	5.257
3	14.258	1.968	2.921	11.922
4	23.290	37.877	42.748	19.900
5	31.637	37.640	37.112	26.979
6	33.909	37.204	35.908	30.439
7	31.309	32.409	30.720	29.960
8	23.779	25.182	23.697	25.187
9	15.522	22.490	19.853	19.178
10	8.508	976	648	13.058
11	4.126	484	343	7.989
12	1.872	263	206	4.955
13	104	113	143	575
14	47	34	85	319
15	15	0	31	181
16	8	0	22	79
17	0	0	5	33
18	0	0	6	7
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

Chiptyp MOE430A

Die folgende Tabelle B.9 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2e) für einen Affymetrix Microarray vom Typ MOE430A.

Tabelle B.9: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp MOE430A

Anzahl	**Anzahl Oligos für Nukleotid**			
Nukleotide	**A**	**C**	**G**	**T**
0	143	34	446	200
1	1.064	382	2.360	1.332
2	4.615	2.612	8.215	4.783
3	14.243	10.054	18.874	12.152
4	30.263	24.931	32.139	22.870
5	45.435	43.817	44.262	32.497
6	50.944	61.292	48.838	38.717
7	44.353	61.297	43.187	38.560
8	30.341	35.723	32.124	34.176
9	16.886	9.256	18.416	26.161
10	7.544	413	585	17.956
11	2.961	123	341	10.919
12	881	23	125	5.750
13	237	1	38	2.626
14	43	0	7	955
15	4	0	1	255
16	1	0	0	40
17	0	0	0	9
18	0	0	6	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

Chiptyp Mouse430_2

Die folgende Tabelle B.10 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Anzahl der Nukleotide (A, C, G, T) in den Oligo-Sequenzen (Daten der Abbildung 6.2b) für einen Affymetrix Microarray vom Typ Mouse430_2.

Tabelle B.10: Anzahl Oligos in Abhängigkeit von der Nukleotidanzahl für den Chiptyp Mouse430_2

Anzahl	**Anzahl Oligos für Nukleotid**			
Nukleotide	**A**	**C**	**G**	**T**
0	271	69	1.035	342
1	2.031	51	5.669	2.220
2	9.212	5.973	18.851	8.241
3	28.083	22.117	41.836	21.149
4	59.555	53.708	68.891	40.413
5	89.195	91.166	90.702	60.086
6	100.366	120.462	95.999	74.155
7	88.158	116.630	81.435	77.512
8	60.718	67.313	58.228	70.033
9	34.477	17.154	32.055	55.821
10	15.810	764	909	39.538
11	6.127	214	561	24.514
12	1.894	42	217	13.229
13	469	4	63	6.109
14	87	1	16	2.319
15	13	0	1	662
16	2	0	0	104
17	0	0	0	20
18	0	0	0	1
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

B.4 Oligo-Häufigkeitsverteilung in Abhängigkeit von der Nukleotidposition

Chiptyp HG-U95Av2

Die folgende Tabelle B.11 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ HG-U95Av2.

Tabelle B.11: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U95Av2

Nukleotid-position	Anzahl Oligos für Nukleotid			
	A	C	G	T
1	47.166	50.337	47.287	54.284
2	48.978	49.929	46.017	54.160
3	49.124	51.914	44.031	54.015
4	47.987	52.022	43.173	55.903
5	44.734	53.897	45.002	55.451
6	47.821	52.259	44.973	54.031
7	47.896	51.810	45.297	54.108
8	47.681	51.590	46.153	52.660
9	45.282	51.808	45.806	54.302
10	45.282	53.496	49.512	50.794
11	45.782	54.385	45.268	53.649
12	49.638	52.584	41.550	55.312
13	60.564	36.248	35.301	66.971
14	48.106	43.761	50.570	56.640
15	49.779	46.963	48.225	54.117
16	45.761	51.929	50.483	50.911
17	49.084	51.181	46.984	51.835
18	50.139	47.875	47.806	53.210
19	51.051	44.138	49.369	53.210
20	52.323	41.420	53.343	51.998
21	52.208	47.059	47.472	52.345
22	52.431	45.506	47.465	53.682
23	51.703	45.926	48.402	53.053
24	51.939	44.817	48.582	53.746
25	49.611	44.574	52.763	52.136

Chiptyp HG-U133A

Die folgende Tabelle B.12 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ HG-U133A.

Tabelle B.12: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U133A

Nukleotid-position	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
1	56.481	42.646	92.145	56.693
2	75.145	54.695	58.366	59.759
3	69.586	62.075	57.267	59.037
4	62.930	59.365	60.949	64.721
5	52.174	73.680	61.455	60.656
6	59.325	63.484	54.630	70.526
7	52.142	67.401	63.084	65.338
8	56.509	64.757	57.708	68.991
9	57.660	63.335	59.656	67.314
10	56.788	66.631	58.322	66.224
11	58.159	59.884	58.265	71.657
12	60.543	60.676	59.472	67.274
13	58.806	56.625	53.523	79.011
14	58.724	62.197	56.466	70.578
15	59.497	59.306	58.700	70.462
16	58.167	63.329	59.257	67.212
17	58.390	58.298	59.937	71.340
18	53.909	61.167	62.956	69.933
19	57.558	60.511	58.036	71.860
20	54.683	63.052	62.200	68.030
21	55.375	63.009	58.956	70.625
22	56.260	64.163	58.744	68.798
23	59.060	63.984	60.781	64.140
24	60.988	62.298	59.646	65.033
25	65.197	64.556	52.991	65.221

Chiptyp HG-U133_Plus_2

Die folgende Tabelle B.13 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ HG-U133_Plus_2.

Tabelle B.13: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp HG-U133_Plus_2

Nukleotid-position	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
1	147.941	93.289	222.809	140.219
2	188.835	123.887	139.956	151.580
3	171.771	144.807	138.808	148.872
4	156.687	140.288	143.202	164.081
5	130.734	174.143	145.711	153.670
6	145.789	151.839	128.368	178.262
7	130.580	160.298	147.121	166.259
8	139.254	155.140	134.468	175.396
9	142.945	152.355	139.157	169.801
10	140.617	159.595	138.542	165.504
11	144.761	143.785	136.004	179.708
12	148.630	147.180	137.708	170.740
13	147.282	134.446	124.062	198.468
14	146.751	148.499	130.931	178.077
15	146.368	143.933	139.385	174.572
16	143.205	153.696	139.650	167.707
17	144.963	141.283	138.700	179.312
18	133.587	148.689	147.835	174.147
19	142.087	147.386	136.799	177.986
20	135.620	153.782	144.461	170.395
21	137.575	151.672	137.509	177.502
22	139.265	154.471	137.863	172.659
23	148.742	152.337	140.792	162.387
24	158.015	143.005	139.288	163.950
25	168.963	141.488	125.496	168.311

Chiptyp MG-U74Av2

Die folgende Tabelle B.14 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ MG-U74Av2.

Tabelle B.14: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp MG-U74Av2

Nukleotid-position	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
1	46.973	49.848	46.789	54.383
2	48.790	49.720	46.376	53.107
3	48.790	51.976	44.388	52.839
4	47.172	52.086	44.084	54.651
5	45.278	53.239	45.381	54.095
6	47.707	52.095	45.297	52.894
7	47.847	51.533	46.151	52.462
8	47.831	50.839	46.822	52.501
9	46.954	51.270	46.536	53.233
10	44.355	53.922	51.176	48.540
11	44.962	54.560	46.523	51.948
12	48.711	53.433	42.100	53.749
13	60.991	34.910	34.779	67.313
14	47.445	43.720	51.928	54.900
15	48.207	48.013	49.306	52.467
16	44.372	53.281	51.154	49.186
17	48.312	51.678	47.285	50.718
18	49.604	48.274	48.193	51.922
19	49.695	44.946	50.436	52.916
20	51.467	42.092	54.032	50.402
21	51.111	47.385	48.089	51.408
22	51.463	46.487	47.995	52.048
23	50.677	46.625	48.789	51.902
24	51.153	45.390	48.990	52.460
25	48.194	44.670	53.720	51.409

Chiptyp MOE430A

Die folgende Tabelle B.15 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ MOE430A.

Tabelle B.15: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp MOE430A

Nukleotid-position	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
1	74.458	28.464	86.547	60.489
2	77.640	44.499	64.898	62.921
3	69.623	55.721	63.963	60.651
4	62.719	59.555	60.634	67.050
5	54.887	70.877	61.456	62.738
6	57.736	66.189	52.677	73.356
7	54.802	66.195	60.125	68.836
8	55.326	67.502	52.247	74.883
9	57.201	68.128	55.782	68.847
10	56.157	67.452	59.931	66.418
11	56.845	64.019	56.903	72.191
12	58.575	66.932	52.287	72.164
13	64.984	53.781	47.745	83.448
14	61.392	60.989	52.274	75.303
15	55.700	65.785	60.156	68.317
16	56.642	66.224	58.508	68.584
17	58.022	62.810	54.719	74.407
18	55.644	65.458	59.273	69.583
19	56.045	66.744	58.844	68.325
20	54.994	68.000	57.472	69.492
21	58.529	65.478	55.849	70.102
22	57.706	65.407	58.799	68.046
23	63.383	63.515	55.942	67.118
24	74.966	53.371	56.023	65.598
25	80.453	43.174	52.005	74.326

Chiptyp Mouse430_2

Die folgende Tabelle B.16 zeigt die Häufigkeitsverteilung von Oligos in Hinsicht auf die Nukleotidpositionen in den Oligo-Sequenzen (Daten der Abbildung 6.3) für einen Affymetrix Microarray vom Typ Mouse430_2.

Tabelle B.16: Anzahl Oligos in Abhängigkeit von der Nukleotidposition für den Chiptyp Mouse430_2

Nukleotid-position	**Anzahl Oligos für Nukleotid**			
	A	**C**	**G**	**T**
1	147.355	55.782	172.329	121.002
2	156.285	86.998	124.818	128.367
3	139.539	108.617	124.208	124.104
4	124.852	116.860	117.221	137.535
5	110.171	138.410	119.285	128.602
6	115.366	129.663	101.736	149.703
7	109.041	130.362	116.381	140.684
8	110.701	131.891	100.538	153.338
9	114.416	132.816	107.864	141.372
10	111.942	133.154	116.159	135.213
11	114.206	124.691	109.512	148.059
12	117.197	130.969	101.185	147.117
13	128.199	106.126	92.928	169.215
14	121.797	119.990	100.587	154.094
15	112.214	128.281	116.124	139.849
16	112.474	131.014	113.107	139.873
17	116.197	122.888	105.211	152.172
18	110.582	128.434	114.661	142.791
19	111.482	131.257	113.904	139.825
20	109.377	134.060	111.284	141.747
21	116.174	128.635	107.636	144.023
22	114.166	128.841	113.921	139.540
23	126.203	124.258	108.053	137.954
24	150.147	104.061	107.886	134.374
25	159.922	84.388	100.696	151.462

B.5 Mittelbasenabhängige PM/MM-Missverhältnisse

Die folgende Tabelle B.17 zeigt das absolute und relative PM/MM-Missverhältnis für die Replikatgruppen des Latin-Square-Experiments. Das relative PM/MM-Missverhältnis wird dabei auf der Basis der Anzahl von Oligos mit der jeweiligen Mittelbase berechnet. Die Anzahl von Oligos mit der Mittelbase A, C, G, T ist in der Tabelle B.12 (vgl. Zeile für Nukleotidposition 13) angegeben.

Tabelle B.17: PM/MM-Missverhältnis

Replikat-	∅ Anzahl PM<MM				∅ Anteil PM<MM in %			
gruppe	**A**	**C**	**G**	**T**	**A**	**C**	**G**	**T**
Gruppe 1	45.121,67	17.559,67	36.035,33	26.206,33	76,73	31,01	67,33	33,17
Gruppe 2	44.555,00	17.100,00	35.799,67	25.401,67	75,77	30,20	66,89	32,15
Gruppe 3	46.010,67	18.281,67	36.956,00	27.792,33	78,24	32,29	69,05	35,18
Gruppe 4	45.095,33	17.993,00	36.584,00	27.224,00	76,68	31,78	68,35	34,46
Gruppe 5	44.560,33	17.746,33	36.523,33	26.876,33	76,78	31,34	68,24	34,02
Gruppe 6	44.560,33	16.846,33	35.629,00	25.150,67	75,77	29,75	66,57	31,83
Gruppe 7	46.548,67	18.731,33	37.338,67	28.778,33	79,16	33,08	69,76	36,42
Gruppe 8	46.258,00	19.173,00	37.446,33	29.779,00	78,66	33,86	69,96	37,69
Gruppe 9	46.213,67	18.394,33	37.732,00	28.322,67	78,59	32,48	70,50	35,85
Gruppe 10	45.346,00	17.718,67	36.540,67	26.397,67	77,11	31,29	68,27	33,41
Gruppe 11	44.728,67	16.981,67	35.707,33	25.605,33	76,06	29,99	66,71	32,41
Gruppe 12	45.808,00	18.240,33	36.819,00	28.303,00	77,90	32,21	68,79	35,82
Gruppe 13	45.964,00	19.002,67	37.371,67	29.282,33	78,16	33,56	69,82	37,06
Gruppe 14	45.184,67	17.615,67	36.420,33	26.622,67	76,84	31,11	68,05	33,69
∅	45.467,59	17.956,05	36.635,95	27.267,31	77,32	31,71	68,44	34,51

B.6 Standardisierte Mitteltripel-Intensitäten

Die folgende Tabelle B.18 zeigt die durchschnittlichen standardisierten Mitteltripel-Intensitäten für das Latin-Square-Experiment (Daten der Abbildung 6.5).

Tabelle B.18: Standardisierte Mitteltripel-Intensitäten

Mittel base	**Mittel tripel**	$\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\varnothing Z^{MM}_{(\mathrm{XYZ})}$	$\sigma\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\sigma\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\varnothing Z^{PM-MM}_{(\mathrm{XYZ})}$
A	AAA	-0,358	-0,048	0,854	0,977	-0,31
A	AAC	-0,257	0,096	0,852	1,027	-0,353
A	AAG	-0,299	0,131	0,841	1,007	-0,43
A	AAT	-0,242	0,049	0,878	1,004	-0,291
A	CAA	-0,153	0,162	0,909	1,059	-0,315
A	CAC	-0,002	0,497	0,974	1,178	-0,499
A	CAG	-0,125	0,521	0,926	1,202	-0,646
A	CAT	0,021	0,392	0,992	1,14	-0,371
A	GAA	-0,293	0,026	0,843	0,96	-0,319
A	GAC	-0,105	0,352	0,884	1,128	-0,457
A	GAG	-0,226	0,376	0,848	1,109	-0,602
A	GAT	-0,07	0,346	0,939	1,106	-0,416
A	TAA	-0,265	-0,023	0,879	0,968	-0,242
A	TAC	-0,02	0,295	0,968	1,123	-0,315
A	TAG	-0,112	0,337	0,936	1,122	-0,449
A	TAT	-0,139	0,161	0,926	1,063	-0,3
C	ACA	-0,235	-0,406	0,876	0,705	0,171
C	ACC	0,081	-0,379	0,993	0,672	0,46
C	ACG	-0,179	-0,256	0,904	0,748	0,077
C	ACT	-0,059	-0,331	0,953	0,749	0,272
C	CCA	0,121	-0,468	1,045	0,664	0,589
C	CCC	0,605	-0,378	1,194	0,712	0,983
C	CCG	0,165	-0,313	1,092	0,688	0,478
C	CCT	0,485	-0,318	1,159	0,753	0,803
C	GCA	-0,164	-0,248	0,894	0,753	0,084
C	GCC	0,255	-0,166	1,108	0,802	0,421
C	GCG	-0,199	-0,072	0,885	0,788	-0,127
C	GCT	0,118	-0,039	1,049	0,89	0,157
C	TCA	-0,004	-0,295	0,998	0,791	0,291
C	TCC	0,398	-0,13	1,144	0,897	0,528
C	TCG	0,106	-0,051	1,034	0,879	0,157
C	TCT	0,163	-0,107	1,071	0,903	0,27
G	AGA	-0,3	-0,006	0,826	0,954	-0,294
G	AGC	-0,122	0,485	0,923	1,174	-0,607

Mittel base	**Mittel tripel**	$\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\varnothing Z^{MM}_{(\mathrm{XYZ})}$	$\sigma\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\sigma\varnothing Z^{PM}_{(\mathrm{XYZ})}$	$\varnothing Z^{PM-MM}_{(\mathrm{XYZ})}$
G	AGG	-0,164	-0,021	0,883	0,954	-0,143
G	AGT	-0,038	0,16	0,936	1,038	-0,198
G	CGA	-0,206	0,465	0,893	1,19	-0,671
G	CGC	0,032	1,17	1,001	1,299	-1,138
G	CGG	-0,11	0,443	0,934	1,178	-0,553
G	CGT	0,179	0,978	1,053	1,288	-0,799
G	GGA	-0,154	0,083	0,877	1,001	-0,237
G	GGC	-0,007	0,643	0,939	1,229	-0,65
G	GGG	-0,058	0,043	0,936	0,975	-0,101
G	GGT	0,232	0,433	1,057	1,166	-0,201
G	TGA	-0,049	0,232	0,984	1,115	-0,281
G	TGC	0,257	0,786	1,098	1,247	-0,529
G	TGG	0,164	0,202	1,028	1,074	-0,038
G	TGT	0,284	0,477	1,078	1,175	-0,193
T	ATA	-0,293	-0,339	0,841	0,765	0,046
T	ATC	-0,09	-0,329	0,919	0,729	0,239
T	ATG	-0,093	-0,281	0,939	0,77,	0,188
T	ATT	-0,164	-0,346	0,903	0,736	0,182
T	CTA	-0,009	-0,228	1,016	0,849	0,219
T	CTC	0,212	-0,119	1,089	0,872	0,331
T	CTG	0,202	-0,178	1,099	0,818	0,38
T	CTT	0,092	-0,173	1,039	0,869	0,265
T	GTA	-0,099	-0,3	0,92	0,753	0,201
T	GTC	0,244	-0,174	1,077	0,821	0,418
T	GTG	0,164	-0,199	1,022	0,767	0,363
T	GTT	0,11	-0,217	1,022	0,801	0,327
T	TTA	-0,192	-0,283	0,925	0,821	0,091
T	TTC	0,077	-0,233	1,016	0,813	0,31
T	TTG	0,094	-0,229	1,032	0,811	0,323
T	TTT	-0,102	-0,301	0,946	0,776	0,199

C Anfrageformulierung und -transformation von Annotationsanalysen in *GeWare*

Im Rahmen der Annotationsanalyse (vgl. Abschnitt 8.2) unterscheidet *GeWare* in Projektions- und Selektionsanfragen. Beide Anfragetypen kann der Benutzer auf Basis eines web-basierten GUI formulieren.

C.1 Projektionsanfragen

Ziel einer Projektionsanfrage, die im Rahmen der Annotationsanalyse in *GeWare* ermöglicht wird, ist es, für eine gegebene Objektmenge (Eingabe) die Werte einer ausgewählten Attributmenge (Ausgabe) anzuzeigen. Die Objektmenge wird zu Beginn einer Projektionsanfrage automatisch übernommen, z.B. aus einer unmittelbar zuvor durchgeführten Analyse oder einem Bericht. Jedes Objekt besitzt einen in der Objektmenge eindeutigen Identifikator. Eine solche Projektionsanfrage Q_P ist allgemein von folgender Form.

$$Q_P = (S, T, O_S, A_T, P_{S-c}, P_{c-T})$$

Hierbei ist S eine Menge von Datenquellen[67], aus denen die Objektmengen $O_{s_1}, O_{s_2}, \ldots, O_{s_n} \in O_S$ mit $s_1, s_2, \ldots, s_n \in S$ stammen, die als Eingabe für die Projektionsanfrage dienen. *GeWare* bildet bisher nur den Spezialfall $|S| = 1$ ab, d.h. es gibt genau eine Objektmenge O_s der Datenquelle s. Für O_S werden die Werte einer selektierten Attributmenge A_T projiziert. Diese Attributmenge A_T ist eine Teilmenge der in den integrierten Datenquellen[68] $T = \{t_1, \ldots, t_m\}$ verfügbaren Attributmenge $A_{total} = \{a_{1,t_1}, \ldots, a_{n,t_1}, a_{1,t_2}, \ldots, a_{n,t_m}\}$. Die Spezifikation von A_T kann iterativ erfolgen, d.h. neue Attribute a_{i,t_j} $(1 \leq i \leq n, 1 \leq j \leq m)$ werden einzeln zur Attributmenge A_T hinzugefügt. Die Menge P_{S-c} charakterisiert die selektierten Pfade $p_{s_i-c} \in P_{S-c}$ zwischen den Datenquellen s_i $(1 \leq i \leq |S|)$ und der zentralen Datenquelle c (c = Center), anhand der die Objektmenge O_{s_i} auf die Objekte der zentralen Datenquelle abgebildet wird. Die Menge P_{c-T} enthält alle spezifizierten Pfade $p_{c-t_i} \in P_{c-T}$ zwischen der zentralen Datenquelle und den Datenquellen $t_i \in T$ $(1 \leq i \leq |T|)$, für die zu projizierende Attribute ausgewählt wurden. Die beiden Mengen von Pfaden P_{S-c} und P_{c-T} sind als bidirektionale Mappings in der Mapping-DB gespeichert.

Der Query-Mediator transformiert die formulierte Projektionsanfrage in eine Sequenz von SRS-Anfragen, um den Limitierungen von SRS zu begegnen. Da SRS innerhalb einer Anfrage keine Join-Operation durchführen kann,

[67] Das Symbol S steht für Sources.
[68] Das Symbol T steht für Targets.

sondern lediglich eine mengenmäßige Traversierung zwischen den Datenquellen ermöglicht, kann eine SRS-Anfrage nur die zu projizierenden Attribute einer Datenquelle enthalten. Aus diesem Grund bildet der Query-Mediator auf Basis der spezifizierten Projektionsanfrage eine Menge von SRS-Anfragen (spezifisch für die Ziel-Datenquelle). Die Abbildung C.1 zeigt ein Syntaxdiagramm einer erzeugten SRS-Anfrage. Die Ergebnisse der einzelnen SRS-Anfragen werden nach deren Verarbeitung vom Query-Mediator zusammengesetzt.

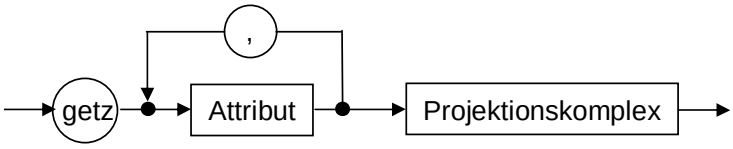


Abbildung C.1: Syntaxdiagramm der erzeugten SRS-Projektionsanfrage

Die erzeugten SRS-Anfragen basieren auf der im Folgenden dargestellten Grammatik, die in der Erweiterten Backus-Naur-Form (EBNF) angegeben ist.

```
<Projektionskomplex> ::= "(" <Center-Target-Mapping> <Join> "(" <Center-DB-Name> <Join>
                           "(" <Source-Center-Mapping> <Join> "("
                           <Source-Datenquelle-Name> "-" <Source ID-Attribut> ":"
                           <Attribut-Wert> ")" ")" ")" ")"

<Source-Center-Mapping> ::= <Mapping-Selektion>
<Center-Target-Mapping> ::= <Mapping-Selektion>
<Mapping-Selektion>     ::= "[" <Mapping-DB-Name> "-" <Path-Id> ":" <ID> "]"
<Join>                  ::= "<"

<Source-Datenquellen-Name> ::= <alphanumWert>
<Source ID-Attribut>       ::= <alphanumWert>
<Attribut-Wert>            ::= <alphanumWert>

<Center-DB-Name>     ::= "Center"
<Mapping-DB-Name>    ::= "Mapping"
<Path-ID>            ::= "pid"
<ID>                 ::= <natZahl>

<alphaNumWert> ::= (<natZahl> | <Zeichenkette>)+
<natZahl>      ::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")+
<Zeichenkette> ::= ("a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
                    "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
                    "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
                    "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z")+
```

C.2 Selektionsanfragen

Ziel einer Selektionsanfrage, die im Rahmen einer Annotationsanalyse in *GeWare* durchgeführt werden kann, ist es, eine Objektmenge zu erhalten, für

die eine Menge von spezifizierten Filterbedingungen gilt. Eine solche Selektionsanfrage Q_S kann allgemein mit dem folgenden Tupel beschrieben werden.

$$Q_S = (S, t, C_{A_S}, P_{S-c}, p_{c-t})$$

Hierbei ist S eine Menge von Datenquellen, für die Bedingungen C_{A_S} spezifiziert werden. Die Angabe einer Bedingung $c_{a_{i,s_j}}$ erfolgt attributweise, d.h. für ein Attribut a_i $(1 \leq i \leq |A|)$ der Attributmenge A_{s_j} $(1 \leq j \leq |S|)$ der Datenquelle $s_j \in S$. Jede Bedingung ist von der Form (a,o,w), wobei das quellenspezifische Attribut a unter Nutzung eines Vergleichsoperators o mit einem Wert w verglichen wird. Die Menge P_{S-c} enthält alle Pfade zwischen den Datenquellen s_i und der zentralen Datenquelle c, mit denen die aus den Bedingungen resultierenden Objektmengen auf Objektmengen der zentralen Datenquelle abgebildet werden können. Der spezifizierte Pfad p_{c-t} verbindet die zentrale Datenquelle c und eine ausgewählte Ziel-Datenquelle t, deren Objektmenge auf Grund der durchgeführten Selektion zurückgeliefert werden soll. Ein Beispiel zur Formulierung einer Selektionsanfrage unter Nutzung der Web-Schnittstelle zeigt die Abbildung 8.5 in Abschnitt 8.5.2.

Der Query-Mediator transformiert die formulierte Selektionsanfrage in eine SRS-Anfrage. Im Gegensatz zu Projektionsanfragen wird eine einzelne SRS-Anfrage erstellt, was mit der Nutzung der Mapping-DB als SRS-Datenquelle möglich wird; ohne diese Mapping-DB wären quellenspezifische Anfragen notwendig, da SRS nicht gleichzeitig eine Selektion in zwei verschiedenen Datenquellen ausführen kann. Die Abbildung C.2 illustriert ein Syntaxdiagramm einer solchen erzeugten SRS-Anfrage.

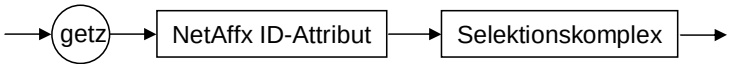


Abbildung C.2: Syntaxdiagramm der erzeugten SRS-Selektionsanfrage

Die erzeugten SRS-Anfragen basieren auf der im Folgenden dargestellten Grammatik, die in der Erweiterten Backus-Naur-Form (EBNF) angegeben ist.

```
<Selektionskomplex> ::= "(" <Center-Target-Mapping> <Join> "(" <Center-DB-Name> <Join>
                           (<Bedingungskomplex>)+ ")" ")"
<Bedingungskomplex> ::= "(" <Source-Center-Mapping> <Join> "(" <Filterbedingungen> ")" ")"
<Filterbedingungen> ::= <Bedingung> |
                        <Bedingung> "&" <Filterbedingungen> |
                        <Bedingung> "|" <Filterbedingungen>
                        "(" <Bedingung> "&" <Filterbedingungen> ")" |
                        "(" <Bedingung> "|" <Filterbedingungen> ")"
<Bedingung> ::= <Datenquelle-Name> "-" <Attribut-Name> ":" <Attribut-Wert>

<Source-Center-Mapping> ::= <Mapping-Selektion>
```

```
<Center-Target-Mapping> ::= <Mapping-Selektion>
<Mapping-Selektion>     ::= "[" <Mapping-DB-Name> "-" <Path-Id> ":" <ID> "]"
<Join>                  ::= "<"

<Datenquellen-Name>   ::= <alphanumWert>
<Attribut-Name>       ::= <alphanumWert>
<Attribut-Wert>       ::= (<alphanumWert> | <Jokerzeichen>)+
<NetAffx ID-Attribut> ::= <alphanumWert>
<Jokerzeichen>        ::= "*"|"?"

<Center-DB-Name>     ::= "Center"
<Mapping-DB-Name>    ::= "Mapping"
<Path-ID>            ::= "pid"
<ID>                 ::= <natZahl>

<alphanumWert> ::= (<natZahl> | <Zeichenkette>)+
<natZahl>      ::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")+
<Zeichenkette> ::= ("a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
                    "o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|"A"|
                    "B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|"N"|
                    "O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z")*
```

Die Terminale, die die Nicht-Terminale <Datenquellen-Name>, <Attribut-Name> (siehe Nicht-Terminal <Bedingung>) und <NetAffx ID-Attribut> ersetzen, entstammen einer Tabelle der ADM-Datenbank, die die Namen der integrierten Datenquellen und ihrer Attribute sowohl in Klartext als auch in der von SRS verwendeten Form beinhaltet. Das Terminal, das das Nicht-Terminal <Attribut-Wert> substituiert, wird zur Zeit der Anfrageformulierung vom Benutzer unter Nutzung der Web-Schnittstelle spezifiziert.

D Anfrageformulierung und -transformation in *BioFuice*

BioFuice erlaubt unterschiedliche Arten der Anfrageformulierung. Für die Stichwortsuche und Anfragen auf Basis der Metadaten-Modelle (Source-Mapping- und Domänenmodell), die so genannten modellbasierten Anfragen, wird eine Transformation der formulierten Anfrage in ausführbare *iFuice*-Skripte notwendig. Im Folgenden soll diese Transformation charakterisiert werden.

D.1 Anfragen zur Stichwortsuche

Es sind zwei Arten von Anfragen zur Stichwortsuche zu unterscheiden, die LDS- und Objekttyp-spezifische Stichwortsuche. Sie resultieren aus verschiedenen Eingaben und ziehen eine unterschiedliche Anfragetransformation in ausführbare *iFuice*-Skripte nach sich.

LDS-spezifische Stichwortsuche. Eine Anfragen zur LDS-spezifischen Stichwortsuche $Q_{KW_{LDS}}$ kann formal wie folgt definiert werden.

$$Q_{KW_{LDS}} = (s_{LDS}, W)$$

Bei Anfragen dieser Art wird eine Menge von Stichwörtern W für eine LDS s_{LDS} spezifiziert, wobei W nicht leer sein darf. Aus diesen Angaben generiert *BioFuice* ein ausführbares *iFuice*-Skript. Eine Grammatik der dazu notwendigen Sprache $L_{KW_{LDS}}$ wird in der Erweiterten Backus-Naur-Form (EBNF) angegeben.

```
<Skript>          ::= <Search-Anweisung>
<Search-Anweisung>::= <VariableName> <Zuweisungsoperator> <Search-Operator> <Anweisungsende>

<VariableName>    ::= "zzz_" <natZahl>
<Zuweisungsoperator> ::= ":="
<Search-Operator> ::= "searchInstances (" <LDS> "," <W> ")"
<LDS>             ::= <Objekttyp> "@" <PDS>
<PDS>             ::= <Zeichenkette>
<Objekttyp>       ::= <Zeichenkette>
<Anweisungsende>  ::= ";"
<W>               ::= <alphaNumWert> (<alphaNumWert> | " " | "," | ";")*
<alphaNumWert>    ::= (<natZahl> | <Zeichenkette>)+
<natZahl          ::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")+
<Zeichenkette>    ::= ("a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
                       "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
                       "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
                       "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z")+
```

Hierbei entspricht <LDS> der ausgewählten LDS und <W> der spezifizierten Wortmenge. Das Skript besteht somit aus lediglich einer Anweisung und wird zur Abarbeitung an den iFuice-Mediator übergeben.

Objekttyp-spezifische Stichwortsuche. Eine Anfrage zur Objekttyp-spezifischen Stichwortsuche $Q_{KW_{OT}}$ kann wie folgt definiert werden.

$$Q_{KW_{OT}} = (S_{LDS_{OT}}, W)$$

Mit Anfragen dieser Art werden die spezifizierten Stichwörter (Wortmenge W) in der Menge der LDS $S_{LDS_{OT}}$ gesucht, die dem ausgewählten Objekttyp OT besitzen. Im Anschluss werden die Suchtreffer (Objekte) fusioniert und vereinigt, sofern Same-Mappings zur Verfügung stehen. Mit diesen Angaben wird ein *iFuice*-Skript generiert. Ein solches generiertes Skript basiert auf der Sprache $L_{KW_{OT}}$ für das eine Grammatik in der Erweiterten Backus-Naur-Form (EBNF) angegeben ist.

```
<Skript>              ::= <Search-Anweisung> | <OT-Anweisungsblock>
<OT-Anweisungsblock>  ::= <OT-Anweisungsblock> <LDS-Anweisungsblock> <Union-Anweisung> |
                          <LDS-Anweisungsblock> <LDS-Anweisungsblock> <Union-Anweisung>
<LDS-Anweisungsblock> ::= <Search-Anweisung> <Aggregate-Anweisung>
                          (<Aggregate-Anweisung> <Union-Anweisung>)*

<Search-Anweisung>    ::= <VariableName> <Zuweisungsoperator> <Search-Operator>
                          <Anweisungsende>
<Aggregate-Anweisung> ::= <VariableName> <Zuweisungsoperator> <Aggregate-Operator>
                          <Anweisungsende>
<Union-Anweisung>     ::= <VariableName> <Zuweisungsoperator> <Union-Operator>
                          <Anweisungsende>

<VariableName>        ::= "zzz_" <natZahl>
<Zuweisungsoperator>  ::= ":="
<Search-Operator>     ::= "searchInstances (" <LDS> "," <W> ")"
<Aggregate-Operator>  ::= "aggregate (" <VariableName> "," <MappingSequenz> ")"
<Union-Operator>      ::= "union (" <VariableName> "," <VariableName> ")"
<Anweisungsende>  ::= ";"

<MappingSequenz>  ::= <MappingSequenz> "," <MappingName> | <MappingName>
<MappingName>     ::= <Zeichenkette>

<LDS>             ::= <Objekttyp> "@" <PDS>
<PDS>             ::= <Zeichenkette>
<Objekttyp>       ::= <Zeichenkette>
<W>               ::= <alphaNumWert> (<alphaNumWert> | " " | "," | ";")*
<alphaNumWert>    ::= (<natZahl> | <Zeichenkette>)+
<natZahl          ::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")+
<Zeichenkette>    ::= ("a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
                       "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
                       "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
                       "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z")+
```

Abgrenzung der verwendeten Sprachen

Es folgen Betrachtungen, die die Sprachen $L_{KW_{LDS}}$, $L_{KW_{OT}}$ sowie die *iFuice* -Skriptsprache L_{iFuice} gegeneinander abgrenzen.

Lemma 1 ($L_{KW_{LDS}} \subset L_{KW_{OT}}$:)**.** *Die Sprache $L_{KW_{LDS}}$, auf der die im Ergebnis von LDS-spezifischen Stichwort-Anfragen entstehenden Skripte basieren,*

bildet einen Teil der Sprache $L_{KW_{OT}}$, die zur Erzeugung von Skripten für eine Objekttyp-spezifische Stichwortsuche verwendet wird.

Beweis:
(1) $L_{KW_{LDS}} \rightarrow L_{KW_{OT}}$: Jede LDS besitzt genau einen Objekttyp. Dagegen kann ein Objekttyp von verschiedenen LDS repräsentiert werden. Dem *iFuice*-Ansatz folgend ist die Suche nach einer Wortmenge in einer einzelnen LDS ein Spezialfall einer Suche nach derselben Wortmenge in mehreren LDS, die über denselben Objekttyp verfügen. Damit kann erstere nicht komplexer sein als letztere. Das repräsentieren die oben angeführten EBNF von $L_{KW_{LDS}}$ und $L_{KW_{OT}}$, auf der die erzeugten Skripte für beide Anfragearten basieren. Die Skripte LDS-spezifischer Stichwort-Anfragen bestehen aus einer Search-Anweisung. Auch die Skripte Objekttyp-spezifischer Suchwort-Anfragen können eine einzige Anweisung, eine Search-Anweisung, enthalten. Das ist genau dann der Fall, wenn der selektierte Objekttyp genau einer LDS zugeordnet ist. Damit können alle Skripte auf Basis der Sprache $L_{KW_{LDS}}$ auch mit Skripten der Sprache $L_{KW_{OT}}$ abgedeckt werden.

(2) $L_{KW_{LDS}} \leftarrow L_{KW_{OT}}$: Auf Grund der oben angegebenen Kardinalitätsbeziehung (1:N) zwischen Objekttyp und LDS, muss ein Skript auf Basis der Sprache $L_{KW_{OT}}$ mehrere Search-Anweisungen vornehmen können, deren Ergebnisse hiernach fusioniert und vereinigt werden. Gerade aus den zuletzt genannten Operationen (Fusion + Vereinigung) resultieren weitere Anweisungen unter Nutzung weiterer Operatoren, die in Skripten auf Basis der Sprache $L_{KW_{LDS}}$ nicht möglich sind. Damit ist die Sprache $L_{KW_{OT}}$ umfangreicher als $L_{KW_{LDS}}$.

Aus (1) und (2) folgt $L_{KW_{LDS}} \subset L_{KW_{OT}}$.

Lemma 2 ($L_{KW_{OT}} \subset L_{iFuice}$:)**.** *Die Sprache $L_{KW_{OT}}$ ist Teil der iFuice-Skriptsprache L_{iFuice}.*

Beweis:
(1) $L_{KW_{OT}} \rightarrow L_{KW_{iFuice}}$: Die Sprache $L_{KW_{OT}}$ basiert auf der *iFuice*-Skriptsprache L_{iFuice} (vgl. die EBNF in [Tho07]), d.h. sie verwendet ausgewählte Syntaxelemente der Sprache L_{iFuice}, die zu Anweisungen unter Nutzung spezieller Operatoren führen. Sowohl zusätzliche Konstrukte bei der Anfragetransformation als auch Syntaxelemente, die nicht in L_{iFuice} vorhanden sind, finden keine Anwendung. Damit sind Skripte auf Basis von $L_{KW_{OT}}$ gleichzeitig Skripte im Sinne der *iFuice*-Skriptsprache L_{iFuice}.

(2) $L_{KW_{OT}} \leftarrow L_{KW_{iFuice}}$: Die *iFuice*-Skriptsprache L_{iFuice} besitzt gegenüber der Sprache $L_{KW_{OT}}$ weitere Syntaxelemente. Dazu zählen Operatoren, wie

beispielsweise *queryInstances*, *traverse* und *diff* (vgl. Kapitel 9 und [Tho07]). Damit ist die Sprache L_{iFuice} umfangreicher als die Sprache $L_{KW_{OT}}$.

Aus (1) und (2) folgt $L_{KW_{OT}} \subset L_{KW_{iFuice}}$.

Zusammenfassend ergibt sich aus den beiden vorangegangenen Lemmas:

$$L_{KW_{LDS}} \subset L_{KW_{OT}} \subset L_{KW_{iFuice}}.$$

D.2 Anfragen auf Basis der *iFuice*-Metadaten-Modelle

BioFuice ermöglicht Anfragen Q_{MM} auf Basis der *iFuice*-Metadaten-Modelle, die mit dem folgenden Tupel formalisiert werden können.

$$Q_{MM} = (S, T, C_S, P, m)$$

Dabei wird auf Basis der Menge von LDS des Integrationsszenarios zwei Teilmengen S und T gebildet. S enthält alle LDS $s \in S$, für die mindestens eine Bedingung $c_{s,j} \in C_S$ $(j \in N)$ spezifiziert wurde. T ist die Menge der Ziel-LDS, für die relevante Objekte zurückgeliefert werden. Die relevanten Objekte für jede LDS $t \in T$ ergeben sich auf Grund der angegebenen Selektionsbedingungen C_S, wobei die LDS $s \in S$ mit den LDS $t \in T$ mit Mapping-Pfaden $p \in P$ verbunden sind. Ein Mapping-Pfad ist ein Pfad im Graphen des Source-Mapping-Modells $\overrightarrow{G_{SMM}}$ (vgl. Kapitel 9), der zwei LDS v_1 und v_n $(v_1, v_n \in V)$ über evtl. temporäre LDS $v \in V$ miteinander verknüpft. Die Abbildung zwischen den selektierten Objekten in $s \in S$ und $t \in T$ entsteht durch eine Komposition der Mappings entlang eines ausgewählten Mapping-Pfades $p \in P$. Die evtl. resultierenden multiplen Mengen von Zwischenergebnissen werden unter Nutzung der spezifizierten Funktion m gemischt. Als Misch-Funktion stehen die Vereinigung und der Durchschnitt zur Auswahl.

Anhand der formulierten Anfrage wird ein Skript generiert, das zur Ausführung an den *iFuice*-Mediator übergeben wird. Das generierte Skript basiert auf der Sprache L_{MM}, für die im Folgenden eine Grammatik in der Erweiterten Backus-Naur-Form (EBNF) angegeben ist.

```
<Skript>              ::= <Search-Anweisung> (<Search-Anweisung> <Mengen-Operation>)* |
                          <OT-Anweisungsblock> (<OT-Anweisungsblock> <Mengen-Operation>)*
<OT-Anweisungsblock>  ::= <OT-Anweisungsblock> <LDS-Anweisungsblock> <Mengen-Operation> |
                          <LDS-Anweisungsblock> <LDS-Anweisungsblock> <Mengen-Operation>
<LDS-Anweisungsblock> ::= <LDS-Anweisungsblock> <Search-Anweisung> <Map-Anweisung>
                          <Aggregate-Anweisung> <Mengen-Operation> |
                          <Search-Anweisung> <Map-Anweisung> <Aggregate-Anweisung>

<Aggregate-Anweisung> ::= <VariableName> <Zuweisungsoperator> <Aggregate-Operator>
                          <Anweisungsende>
<Intersection-Anweisung> ::= <VariableName> <Zuweisungsoperator> <Intersection-Operator>
                          <Anweisungsende>
<Map-Anweisung>       ::= <VariableName> <Zuweisungsoperator> <Map-Operator>
                          <Anweisungsende>
<Union-Anweisung>     ::= <VariableName> <Zuweisungsoperator> <Union-Operator>
                          <Anweisungsende>
<Search-Anweisung>    ::= <VariableName> <Zuweisungsoperator> <Search-Operator>
                          <Anweisungsende>
<Mengen-Operation>    ::= (<Union-Anweisung> | <Intersection-Anweisung>)

<Aggregate-Operator>  ::= "aggregate (" <VariableName> "," <MappingSequenz> ")"
<Intersection-Anweisung> ::= "intersect (" <VariablenName> "," <VariableName> ")"
```

```
<Union-Operator>      ::= "union (" <VariableName> "," <VariableName> ")"
<Search-Operator>     ::= "searchInstances (" <LDS> "," <W> ")"
<VariableName>        ::= "zzz_" <natZahl>
<Zuweisungsoperator>  ::= ":="

<MappingSequenz>  ::= <MappingSequenz> "," <MappingName> | <MappingName>
<MappingName>     ::= <Zeichenkette>

<LDS>             ::= <Objekttyp> "@" <PDS>
<PDS>             ::= <Zeichenkette>
<Objekttyp>       ::= <Zeichenkette>
<W>               ::= <alphaNumWert> (<alphaNumWert> | " " | "," | ";")*
<alphaNumWert>    ::= (<natZahl> | <Zeichenkette>)+
<natZahl          ::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")+
<Zeichenkette>    ::= ("a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
                       "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
                       "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
                       "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z")+
```

Abgrenzung der verwendeten Sprache

Lemma 3 ($L_{KW_{MM}} \subset L_{iFuice}$:)**.** *Die Sprache $L_{KW_{MM}}$ ist Teil der iFuice-Skriptsprache L_{iFuice}.*

Beweis:
(1) $L_{KW_{MM}} \rightarrow L_{KW_{iFuice}}$: Die Sprache $L_{KW_{MM}}$ basiert auf der *iFuice*-Skriptsprache L_{iFuice} (vgl. die EBNF in [Tho07]), d.h. sie verwendet ausgewählte Syntaxelemente der Sprache L_{iFuice}, die zu Anweisungen unter Nutzung spezieller Operatoren führen. Weder zusätzliche Konstrukte bei der Anfragetransformation noch andere Syntaxelemente, die nicht in L_{iFuice} vorhanden sind, finden Anwendung. Damit sind Skripte auf Basis von $L_{KW_{MM}}$ gleichzeitig Skripte im Sinne der *iFuice*-Skriptsprache L_{iFuice}.

(2) $L_{KW_{MM}} \leftarrow L_{KW_{iFuice}}$: Die *iFuice*-Skriptsprache L_{iFuice} besitzt gegenüber der Sprache $L_{KW_{MM}}$ weitere Syntaxelemente. Dazu zählen Operatoren, wie beispielsweise *queryInstances*, *traverse* und *diff* (vgl. Kapitel 9 und [Tho07]). Damit ist die Sprache L_{iFuice} umfangreicher als die Sprache $L_{KW_{MM}}$.

Aus (1) und (2) folgt $L_{KW_{MM}} \subset L_{KW_{iFuice}}$.

E Document Type Definitionen zum XML-basierten Datenaustausch mit *BioFuice*

E.1 DTD zum Datenaustausch von Objektinstanzen

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ObjectInstances (Object*)>
<!ATTLIST ObjectInstances
  name        CDATA     #REQUIRED
>

<!ELEMENT Object (Attribute+)>
<!ATTLIST Object
  type        CDATA     #REQUIRED
  source      CDATA     #REQUIRED
  confidence  CDATA     #REQUIRED
  idAttribute CDATA     #REQUIRED
>

<!ELEMENT Attribute (#PCDATA)>
<!ATTLIST Attribute
  name        CDATA     #REQUIRED
>
```

E.2 DTD zum Datenaustausch von aggregierten Objekten

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT AggregatedObjects(AggregatedObject*)>
<!ATTLIST AggregatedObjects
  name        CDATA     #REQUIRED
>

<!ELEMENT AggregatedObject(Object+)>

<!ELEMENT Object (Attribute+)>
<!ATTLIST Object
  type        CDATA     #REQUIRED
  source      CDATA     #REQUIRED
  confidence  CDATA     #REQUIRED
  idAttribute CDATA     #REQUIRED
>

<!ELEMENT Attribute (#PCDATA)>
<!ATTLIST Attribute
  name        CDATA     #REQUIRED
>
```

E.3 DTD zum Datenaustausch von Mapping-Resultaten

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT MappingResult (Correspondence+)>
<!ATTLIST MappingResult
  name        CDATA     #REQUIRED
>

<!ELEMENT Correspondence (Domain, Range)>
<!ATTLIST Correspondence
  confidence  CDATA     #REQUIRED
  occurrence  CDATA     #REQUIRED
>

<!ELEMENT Domain (Object)>
<!ELEMENT Range (Object)>
<!ELEMENT Object (Attribute+)>
<!ATTLIST Object
  type        CDATA     #REQUIRED
  source      CDATA     #REQUIRED
  confidence  CDATA     #REQUIRED
  idAttribute CDATA     #REQUIRED
>

<!ELEMENT Attribute (#PCDATA)>
<!ATTLIST Attribute
  name        CDATA     #REQUIRED
>
```

E.4 DTD zum Datenaustausch von aggregierten Mapping-Resultaten

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT AggregatedMappingResult (Correspondence*)>
<!ATTLIST AggregatedMappingResult
  name        CDATA     #REQUIRED
>

<!ELEMENT Correspondence (Domain, Range)>
<!ATTLIST Correspondence
  confidence  CDATA     #REQUIRED
  occurrence  CDATA     #REQUIRED
>

<!ELEMENT Domain (Object+)>
<!ELEMENT Range (Object+)>
<!ELEMENT Object (Attribute+)>
<!ATTLIST Object
  type        CDATA     #REQUIRED
  source      CDATA     #REQUIRED
  confidence  CDATA     #REQUIRED
  idAttribute CDATA     #REQUIRED
>

<!ELEMENT Attribute (#PCDATA)>
<!ATTLIST Attribute
  name        CDATA     #REQUIRED
>
```

Literaturverzeichnis

[ABJ+04] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher und S. Mock. Kepler: An extensible system for design and execution of scientific workflows. In *Proc. of the 16th International Conference on Scientific and Statistical Database Management*, 2004.

[ABN+99] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack und A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[ACH+00] Mark D. Adams, Susan E. Celniker, Robert A. Holt, Cheryl A. Evans, Jeannine D. Gocayne, Peter G. Amanatides, Steven E. Scherer, Peter W. Li, Roger A. Hoskins, Richard F. Galle, Reed A. George, Suzanna E. Lewis, Stephen Richards, Michael Ashburner, Scott N. Henderson et al. The genome sequence of drosophila melanogaster. *Science*, 287(5461):2185–2195, 2000.

[Aff02] Affymetrix. Statistical algorithms description document. White Paper, 2002.

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers und David J. Lipmanl. Basic local alignment search tool. *Journal of Molecular Biology*, 215(2):403–410, 1990.

[AKK+03] Marcelo Arenas, Vasiliki Kantere, Anastasios Kementsietsidis, Iluju Kiringa, Renee J. Miller und John Mylopoulos. The Hyperion project: From data integration to data coordination. *ACM SIGMOD Record*, 32(3):53–58, 2003.

[AKS77] James C. Alwine, David J. Kemp und George R. Stark. Method for detection of specific RNAs in agarose cells by transfer to diazobenzyloxymethyl-paper and hybridization with DNA

probes. *Proceedings of the National Academy of Sciences*, 74(12):5350–5354, 1977.

[ARC00] John Aach, Wayne Rindone und George M. Church. Systematic management and analysis of yeast gene expression data. *Genome Research*, 10(4):431–445, 2000.

[BA00] Amos Bairoch und Rolf Apweiler. The SWISS–PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45–48, 2000.

[BAB+04] Ewan Birney, T. Daniel Andrews, Paul Bevan, Mario Caccamo, Yuan Chen, Laura Clarke, Guy Coates, James Cuff, Val Curwen, Tim Cutts, Thomas Down, Eduardo Eyras, Xose M. Fernandez-Suarez, Paul Gane, Brian Gibbins, James Gilbert, Martin Hammond, Hans-Rudolf Hotz, Vivek Iyer, Kerstin Jekosch, Andreas Kahari, Arek Kasprzyk, Damian Keefe, Stephen Keenan, Heikki Lehvaslaiho, Graham McVicker, Craig Melsopp, Patrick Meidl, Emmanuel Mongin, Roger Pettett, Simon Potter, Glenn Proctor, Mark Rae, Steve Searle, Guy Slater, Damian Smedley, James Smith, Will Spooner, Arne Stabenau, James Stalker, Roy Storey, Abel Ureta-Vidal, K. Cara Woodwark, Graham Cameron, Richard Durbin, Anthony Cox, Tim Hubbard und Michele Clamp. An overview of Ensembl. *Genome Research*, 14:925–928, 2004.

[BAB05] O. Bodenreider, M. Aubry und A. Bugrun. Non-lexical approaches to identifying associative relations in the Gene Ontology. In *Proc. of the Pacific Symposium on Biocomputing*, 2005.

[Bai00] Amos Bairoch. The ENZYME database in 2000. *Nucleic Acids Research*, 28(1):304–305, 2000.

[BB89] Günter Bamberg und Franz Baur. *Statistik*. Oldenbourg Verlag, 6. Auflage, 1989.

[BB05] O. Bodenreider und A. Bugrun. Linking the Gene Ontology to other biological ontologies. In *Proc. ISMB Meeting on Bio-Ontologies*, 2005.

[BDSY99] Amir Ben-Dor, Ron Shamir und Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[BDW+01] Gary D. Bader, Ian Donaldson, Cheryl Wolting, B. F. Francis Ouellette, Tony Pawson und Christopher W. V. Hogue. BIND– The biomolecular interaction network database. *Nucleic Acids Research*, 29(1):242–245, 2001.

[BGB+99] P.G. Baker, C.A. Goble, S. Bechhofer, N.W. Paton, R. Stevens und A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6):510–520, 1999.

[BGK+02] Philip A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini und I. Zaihrayeu. Data management for peer–to–peer computing: A vision. In *Proc. of the 5th International Workshop on the Web and Databases (WebDB)*, 2002.

[BGL+00] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares und David Haussler. Knowledge–based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267, 2000.

[BHP92] M.W. Bright, A.R. Hurson und Simin H. Pakzad. A taxonomy and current issues in multidatabase systems. *Computer (IEEE)*, 25(3):50–60, 1992.

[BHQ+01] Alvis Brazma, Pascal Hingamp, John Quackenbush, Gavin Sherlock, Paul Spellman, Chris Stoeckert, John Aach, Wilhelm Ansorge, Catherine A. Ball, Helen C. Causton, Terry Gaasterland, Patrick Glenisson, Frank C. P. Holstege, Irene F. Kim, Victor M. Markowitz, John C. Matese, Helen Parkinson, Alan Robinson, Ugis Sarkans, Steffen Schulze-Kremer, Jason Stewart, Ronald Taylor, Jaak Vilo und Martin Vingron. Minimum information about a microarray experiment (MIAME) – Towards standards for microarray data. *Nature Genetics*, 29:365–371, 2001.

[BKH+04] Hans Binder, Toralf Kirsten, Ivo L. Hofacker, Peter F. Stadler und Markus Löffler. Interactions in oligonucleotide hybrid duplexes on microarrays. *Journal Physical Chemistry in the Biology*, 108(46):18015–18025, 2004.

[BKHO02] Hidemasa Bono, Takeya Kasukawa, Yoshihide Hayashizaki und Yasushi Okazaki. READ: RIKEN expression array database. *Nucleic Acids Research*, 30(1):211–213, 2002.

[BKLS04] Hans Binder, Toralf Kirsten, Markus Löffler und Peter F. Stadler. Sensitivity of microarray oligonucleotide probes: Variability and effect of base composition. *Journal Physical Chemistry in the Biology*, 108(46):18003–18014, 2004.

[BKN+06] Jens Bleiholder, Samir Khuller, Felix Naumann, Louiqa Raschid und Yao Wu. Query planning in the presence of overlapping sources. In *Proc. of the International Conference on Extending Database Technology (EDBT)*, 2006.

[BKT00] Peter Buneman, Sanjeev Khanna und Wang-Chiew Tan. Data provenance: Some basic issues. In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (FST TCS)*, 2000.

[BKT01] Peter Buneman, Sanjeev Khanna und Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Proceedings of the 8th International Conference on Database Theorie (ICDT)*, 2001.

[BLM+04] Jens Bleiholder, Zoe Lacroix, Hyma Murthy, Felix Naumann, Louiqa Raschid und Maria-Esther Vidal. BioFast: Challenges in exploring linked life science sources. *SIGMOD Record*, 33(2):72–77, June 2004.

[BO04] Albert-László Barabási und Zoltán N. Oltvai. Network Biology: Understanding the cell's functional organization. *Nature Review Genetics*, 5:101–113, 2004.

[Bod04] Oliver Bodenreider. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32 (Database Issue):D267–D270, 2004.

[BP05] Hans Binder und Stephan Preibisch. Specific and non specific hybridization of oligonucleotide probes on microarrays. *Biophysical Journal*, 89(1):337–352, 2005.

[BP06] Hans Binder und Stephan Preibisch. GeneChip microarray-signal intensities, RNA concentration and probe sequences. *Journal Physics: Condensed Matter*, 18(18):537–566, 2006.

[BPK05] Hans Binder, Stephan Preibisch und Toralf Kirsten. Base pair interactions and hybridization isotherms of matched and mismatched oligonucleotide probes on microarrays. *Langmuir*, 21(20):9287–9302, 2005.

[BPS+03] Alvis Brazma, Helen Parkinson, Ugis Sarkans, Mohammadreza Shojatalab, Jaak Vilo, Niran Abeygunawardena, Ele Holloway, Misha Kapushesky, Patrick Kemmeren, Gonzalo Garcia Lara, Ahmet Oezcimen, Philippe Rocca-Serra und Susanna-Assunta Sansone. ArrayExpress – A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31(1):68–71, 2003.

[BSP+02a] C. A. Ball, G. Sherlock, H. Parkinson, P. Rocca-Sera, C. Brooksbank, H. C. Causton, D. Cavalieri, T. Gaasterland, P. Hingamp, F. Holstege, M. Ringwald, P. Spellman, C. J. Stoeckert, J. E. Stewart, R. Taylor, A. Brazma und J. Quackenbush. An open letter to the scientific journals. *Science*, 298(5593):539, 2002.

[BSP+02b] C. A. Ball, G. Sherlock, H. Parkinson, P. Rocca-Sera, C. Brooksbank, H. C. Causton, D. Cavalieri, T. Gaasterland, P. Hingamp, F. Holstege, M. Ringwald, P. Spellman, C. J. Stoeckert, J. E. Stewart, R. Taylor, A. Brazma und J. Quackenbush. An open letter to the scientific journals. *Bioinformatics*, 18(11):1409, 2002.

[BSP+02c] C. A. Ball, G. Sherlock, H. Parkinson, P. Rocca-Sera, C. Brooksbank, H. C. Causton, D. Cavalieri, T. Gaasterland, P. Hingamp, F. Holstege, M. Ringwald, P. Spellman, C. J. Stoeckert, J. E. Stewart, R. Taylor, A. Brazma und J. Quackenbush. An open letter to the scientific journals. *The Lancet*, 360:1019, 2002.

[BST+05] Tanya Barrett, Tugba O. Suzek, Dennis B. Troup, Stephen E. Wilhite, Wing-Chi Ngau, Pierre Ledoux, Dmitry Rudnev, Alex E. Lash, Wataru Fujibuchi und Ron Edgar. NCBI GEO: Mining Millions of Expression Profiles - Database and Tools. *Nucleic Acids Research*, 33 (Database Issue):D562–D566, 2005.

[Bue05] Kenneth H. Buetow. Cyberinfrastructure: Empowering a third way in biomedical research. *Science*, 308(5723):821–824, 2005.

[BWF+00] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov und P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.

[CAI+99] Michele Cargil, David Altshuler, James Ireland, Pamela Sklar, Kristin Ardlie, Nila Patil, Charles R. Lane, Esther P. Lim, Nilesh Kalyanaraman, James Nemesh, Liuda Ziaugra, Lisa Friedland, Alex Rolfe, Janet Warrington, Robert Lipshutz, George Q. Daley und Eric S. Lander. Characterization of single-nucleotide polymorhisms in coding regions of human genes. *Nature Genetics*, 22(3):231–238, 1999.

[CCW03] Jing Chen, Su Yun Chung und Limsoon Wong. The Kleisli query system as a backbone for bioinformatics data integration and analysis. In Zoe Lacroix und Terence Critchlow (Hrsg.), *Bioinformatics – Managing Scientific Data*, Seiten 147–187. Morgan Kaufmann Publishers, 2003.

[CDJM02] Stéphane Le Crom, Frédéric Devaux, Claude Jacq und Philippe Marc. yMGV: Helping biologists with yeast microarray data mining. *Nucleic Acids Research*, 30(1):76–79, 2002.

[CHS+03] Peter A. Covitz, Frank Hartel, Carl Schaefer, Sherri De Coronado, Gilberto Fragoso, Himanso Sahni, Scott Gustafson und Kenneth H. Buetow. caCORE: A common infrastructure for cancer informatics. *Bioinformatics*, 19(18):2404–2412, 2003.

[Con07] The UniProt Consortium. The Universal Protein Resource. *Nucleic Acids Research*, 35 (Database Issue):D193–D197, 2007.

[Cov03] P. A. Covitz. Class struggle: expression profiling and categorizing cancer. *The Pharmacogenomics Journal*, 3(5):257–260, 2003.

[CPH+03] Michael Cornell, Norman W. Paton, Cornelia Hedeler, Paul Kirby, Daniela Delneri, Andrew Hayes und Stephen G. Oliver. GIMS: An integrated data storage and analysis environment for genomic and functional data. *Yeast*, 20(15):1291–1306, 2003.

[CPW+01] Mike Cornell, Norman W. Paton, Shengli Wu, Carole A. Goble, Crispin J. Miller, Paul Kirby, Karen Eilbeck, Andy Brass, Andrew Hayes und Stephen G. Oliver. GIMS – A data warehouse for storage and analysis of genome sequence and functional data. In *Proc. of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering (BIBE)*, 2001.

[CST+04] Jill Cheng, Shaw Sun, Adam Tracy, Earl Hubbell, Joseph Morris, Venu Valmeekam, Andrew Kimbrough, Melissa S. Cline,

Guoying Liu, Ron Shigeta, David Kulp und Michael A. Siani-Rose. NetAffx Gene Ontology mining tool: A visual approach for microarray data analysis. *Bioinformatics*, 20(9):1462–1463, 2004.

[CWH+02] Kei-Hoi Cheung, Kevin White, Janet Hager, Mark Gerstein, Valerie Reinke, Kenneth Nelson, Peter Masiar, Ranjana Srivastava, Yuli Li, Ju Li, Hongyu Zhao, Jinming Li, David B. Allison, Michael Snyder, Perry Miller und Kenneth Williams. YMD: A microarray database for large–scale gene expression analysis. In *Proc. of the American Medical Informatics Association 2002 Annual Symposium*, Seiten 140–144, 2002.

[CZJM70] T. Caspersson, L. Zech, C. Johansson und E. J. Modest. Identification of human chromosomes by DNA–binding fluorescent agents. *Chromosoma*, 30(2):215–227, 1970.

[Dal02] Peter Dalgaard. *Introductory Statistics with R.* Springer Verlag, 2002.

[DD99] Ruxandra Domenig und Klaus R. Dittrich. An overview and classification of mediated query systems. *ACM SIGMOD Record*, 28(3):63–72, 1999.

[Dev06] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, 2006. ISBN 3-900051-07-0.

[Die06] Reinhard Diestel. *Graphentheorie.* Springer Verlag, 2006.

[Dij59] Edsger Wybe Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269—-271, 1959.

[DJD+01] Robin D. Dowell, Rodney M. Jokerst, Allen Day, Sean R. Eddy und Lincoln Stein. The Distributed Annotation System. *BMC Bioinformatics*, 2(7), 2001.

[DKR03] Hong-Hai Do, Toralf Kirsten und Erhard Rahm. Comparative evaluation of microarray–based gene expression databases. In *Proc. of the 10th Conference on Database Systems for Business, Technology and Web (BTW)*, 2003.

[DL04] E. Dragut und R. Lawrence. Composite mappings between schemas using a reference ontology. In *Proc. International Conference Ontologies, Databases and Applications of Semantics (ODBASE)*, 2004.

[Do06] Hong-Hai Do. *Schema matching and mapping–based data integration.* Dissertation, Interdisciplinary Center for Bioinformatics Leipzig, University of Leipzig, 2006.

[DR04] Hong-Hai Do und Erhard Rahm. Flexible integration of molecular–biological annotation data: The GenMapper approach. In *Proc. of the Conference on Extended Database Technology (EDBT)*, 2004.

[DYCS02] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow und Terence P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistika Sinica*, 12(2), 2002.

[EA93] Thure Etzold und Patrick Argos. SRS – An indexing and retrieval tool for flat file data libraries. *Bioinformatics*, 9(1):49–57, 1993.

[EDL02] Ron Edgar, Michael Domrachev und Alex E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[EDTPS+06] Tina A. Eyre, Fabrice Ducluzeau, Sue Povey Tam P. Sneddon, Elspeth A. Bruford und Michael J. Lush. The HUGO Gene Nomenclature Database, 2006 updates. *Nucleic Acids Research*, 34:(Database Issue):D319–D321, 2006.

[EH98] Dieter Ehrenberg und Peter Heine. Konzept zur Datenintegration für Management Support Systeme auf der Basis uniformer Datenstrukturen. *Wirtschaftsinformatik*, 40(6):503–512, 1998.

[EHB03] Thure Etzold, Howard Harris und Simon Beaulah. SRS: An integration platform for databanks and analysis tools in bioinformatics. In Zoe Lacroix und Terence Critchlow (Hrsg.), *Bioinformatics – Managing Scientific Data*, Seiten 109–146. Morgan Kaufmann Publishers, 2003.

[EKB+05] Markus Eszlinger, Knut Krohn, K. Berger, J. Läuter, S. Kropf, M. Beck, D. Führer und R. Paschke. Gene expression analysis reveals evidence for increased expression of cell cycle-associated genes and G_q-protein-Protein kinase C signaling in cold thyorid nodules. *The Journal of Clinical Endocrinology & Metabolism*, 90(2):1163–1170, 2005.

[ERP+98] Olga Ermolaeva, Mohit Rastogi, Kim D. Pruitt, Gregory D. Schuler, Michael L. Bittner, Yidong Chen, Richard Simon, Paul Meltzer, Jeffrey M. Trent und Mark S. Boguski. Data management and analysis for gene expression arrays. *Nature Genetics*, 20:19–23, 1998.

[EUA96] T. Etzold, A. Ulyanov und P. Argos. SRS : Information retrieval system for molecular biology data banks. *Methods in Enzymology*, 266:114–128, 1996.

[FAW+95] Robert D. Fleischmann, Mark D. Adams, Owen White, Rebecca A. Clayton, Ewen F. Kirkness, Anthony R. Kerlavage, Carol J. Bult, Jean-Francois Tomb, Brian A. Dougherty, Joseph M. Merrick, Keith McKenney, Granger Sutton, Will Fitzhugh, Chris Fields, Jeannie D. Gocyne, John Scott, Robert Shirley, Li-Ing Liu, Anna Glodek, Jenny M. Kelley, Janice F. Weidman, Cheryl A. Phillips, Tracy Spriggs, Eva Hedblom, Matthew D. Cotton, Teresa R. Utterback, Michael C. Hanna, David T. Nguyen, Deborah M. Saudek, Rhonda C. Brandon, Leah D. Fine, Janice L. Fritchman, Joyce L. Fuhrmann, N. S. M. Geoghagen, Cheryl L. Gnehmand Lisa A. McDonald, Keith V. Small, Claire M. Fraser, Hamilton O. Smith und J. Craig Venter. Whole–genome random sequencing and assembly of haemophilus influenzae Rd. *Science*, 269(5223):496–498,507–512, 1995.

[FG53] R.E: Franklin und R.G. Gosling. Molecular configuration in sodium thymonucleate. *Nature*, 171:740–741, 1953.

[FHB+01] Kurt Fellenberg, Nicole C. Hauser, Benedikt Brors, Albert Neutzner, Jörg D. Hoheisel und Martin Vingron. Correspondence analysis applied to microarray data. *Proceedings of the National Academy of Sciences*, 98(19):10781–10786, 2001.

[FHB+02] Kurt Fellenberg, Nicole C. Hauser, Benedikt Brors, Jörg D. Hoheisel und Martin Vingron. Microarray data warehouse al-

lowing for inclusion of experiment annotations in statistical analysis. *Bioinformatics*, 18(3):423–433, 2002.

[FKNT02] I. Foster, C. Kesselman, J.M. Nick und S. Tuecke. Grid services for distributed system integration. *Computer (IEEE)*, 35(6):37–46, 2002.

[Gal06] Michael Y. Galperin. The molecular biology database collection: 2006 update. *Nucleic Acids Research*, 34 (Database issue):D3–D5, 2006.

[Gas71] Joseph L. Gastwirth. A general defintion of the Lorenz curve. *Econometrica*, 39(6):1037–1039, 1971.

[Gas72] Joseph L. Gastwirth. The estimation of the lorenz curve and Gini coefficient. *The Review of Economics and Statistics*, 54(3):306–316, 1972.

[GCB+04] Robert C Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Y. H. Yang und Jianhua Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[GCH+05] Robert Gentleman, Vince Carey, Wolfgang Huber, Rafael Irizarry und Sandrine Dudoit (Hrsg.). *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer Verlag, 2005.

[GCS00] Ian C. Gray, David A. Campbell und Nigel K. Spurr. Single nucleotide polymorphisms as tools in human genetics. *Human Molecular Genetics*, 9(16):2403–2408, 2000.

[GDS03] Yongchao Ge, Sandrine Dudoit und Terence P. Speed. Resampling-based multiple testing for microarray data analysis. Technical Report, Department of Statistics, University of California, Berkeley, 2003.

[Geh65] Edmund A. Gehan. A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1-2):203–223, 1965.

[GGL01] Margaret Gardiner-Garden und Timothy Littlejohn. A comparison of microarray databases. *Briefings in Bioinformatics*, 2(2):143–158, 2001.

[GHI+95] Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman und Jennifer Widom. Accessing and integrating heterogeneous information sources in TSIMMIS. In *Proc. of the AAAI Symposium on Information Gathering*, 1995.

[GMPQ+04] Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, Vasilis Vassalos und Jennifer Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 2004.

[GMUW02] Hector Garcia-Molina, Jeffrey Ullman und Jennifer Widom. *Database Systems – The Complete Book*. Prentice Hall, 2002.

[Gru93] Thomas Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[GSH+06] J. Galle, D. Sittig, I. Hanisch, M. Wobus, E. Wandel, M. Löffler und G. Aust. Individual cell-based models of tumor-environment interactions: Multiple effects of CD97 on tumor invasion. *American Journal of Pathology*, 169(5):1802–1811, 2006.

[GSN+01] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim und A. Brass. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal*, 40(2):532–551, 2001.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill und Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[HAC+05] T. Hubbard, D. Andrews, M. Caccamo, G. Cameron, Y. Chen, M. Clamp, L. Clarke, G. Coates, T. Cox, F. Cunningham,

V. Curwen, T. Cutts, T. Down, R. Durbin, X. M. Fernandez-Suarez, J. Gilbert, M. Hammond, J. Herrero, H. Hotz, K. Howe, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, S. Keenan, F. Kokocinsci, D. London, I. Longden, G. McVicker, C. Melsopp, P. Meidl, S. Potter, G. Proctor, M. Rae, D. Rios, M. Schuster, S. Searle, J. Severin, G. Slater, D. Smedley, J. Smith, W. Spooner, A. Stabenau, J. Stalker, R. Storey, S. Trevanion, A. Ureta-Vidal, J. Vogel, S. White, C. Woodwark und E. Birney. Ensembl 2005. *Nucleic Acids Research*, 33 (Database Issue):D447–D453, 2005.

[Hae02] Theo Haerder. Editorial zum Themenheft: Datenintegration. *Informatik Forschung und Entwicklung*, 17:99–100, 2002.

[HBB+06] Michael Hummel, Stefan Bentink, Hilmar Berger, Wolfram Klapper, Swen Wessendorf, Thomas F.E. Barth, Heinz-Wolfram Bernd, Sergio B. Cogliatti, Judith Dierlamm, Alfred C. Feller, Martin-Leo Hansmann, Eugenia Haralambieva, Lana Harder, Dirk Hasenclever, Michael Kühn, Dido Lenze, Peter Lichter, Jose Ignacio Martin-Subero, Peter Möller, Hans-Konrad Müller-Hermelink, German Ott, Reza M. Parwaresch, Christiane Pott, Andreas Rosenwald, Maciej Rosolowski, Carsten Schwaenen, Benjamin Stürzenhofecker, Monika Szczepanowski, Heiko Trautmann, Hans-Heinrich Wacker, Rainer Spang, Markus Löffler, Lorenz Trümper, Harald Stein und Reiner Siebert. A biological definition of Burkitt's Lymphoma from transcriptional and genomic profiling. *The New England Journal of Medicine*, 354(23):2419–2430, 2006.

[HBW+01] Andrew A Hill, Eugene L. Brown, Maryann Z. Whitley, Greg Tucker-Kellogg, Craig P. Hunter und Donna K. Slonim. Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls. *Genome Biology*, 2(12):055.1–055.13, 2001.

[HCI+04] M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L.

Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, R. White und Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32 (Database Issue):258–261, 2004.

[Hei02] Peter Heine. *Unternehmensweite Datenintegration. Modular–integrierte Datenlogistik in betrieblichen Informationssystemen*. Teubner Verlag, 2002.

[HHNR05] Ralf Heese, Sven Herschel, Felix Naumann und Armin Roth. Self-extending peer data management. In *Proc. of the 11th Conference on Database Systems in Business, Technology and Web (BTW)*, 2005.

[HIMT03] Alon Y. Halevy, Zachary G. Ives, Peter Mork und Igor Tatarinov. Piazza: Data management infrastructure for semantic web applications. In *Proc. of the 12th International Conference on World Wide Web*, 2003.

[HK04] Thomas Hernandez und Subbarao Kambhampati. Integration of biological sources: Current systems and challenges ahead. *SIGMOD Record*, 33(3):51–60, 2004.

[HL93] B. L. Humphreys und D. A. Lindberg. The UMLS project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170–177, 1993.

[HM85] Dennis Heimbiegner und Dennis McLeod. A federated architecture for information management. *ACM Transactions on Information Systems*, 3(3):253–278, 1985.

[HS06] Bettina Harr und Christian Schlötterer. Comparison of algorithms for the analysis of Affymterix microarray data as evaluated by co-expression of genes in known operons. *Nucleic Acids Research*, 34(2):e8, 2006.

[HSS+04] Nicolas Hulo, Christian J. A. Sigrist, Virginie Le Saux, Petra S. Langendijk-Genevaux, Lorenza Bordoli, Alexandre Gattiker,

Edouard De Castro, Philipp Bucher und Amos Bairoch. Recent improvements to the PROSITE database. *Nucleic Acids Research*, 32: (Database Issue):D134–D137, 2004.

[HWB+02] Peter M. Haverty, Zhiping Weng, Nathan L. Best, Kenneth R. Auerbach, Li-Li Hsiao, Roderick V. Jensen und Steven R. Gullans. HugeIndex: A database with visualization tools for high-density oligonucleotide array data from normal human tissues. *Nucleic Acids Research*, 30(1):214–217, 2002.

[HWS+06] Duncan Hull, Katy Wolstencroft, Robert Stevens, Carole Goble, Mathew R. Pocock, Peter Li und Tom Oinn. Taverna: A tool for building and running workflows of services. *Nucleic Acids Research*, 34 (Web Server Issue):W729–W732, 2006.

[IBC+03] Rafael A. Irizarry, Benjamin M. Bolstad, Francois Collin, Leslie M. Cope, Bridget Hobbs und Terence P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research*, 31(4):e15, 2003.

[IHC+03] Rafael A. Irizarry, Bridget Hobbs, Francois Collin, Yasmin D. Beazer-Barclay, Kristen J. Antonellis, Uwe Scherf und Terence P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[IKKC05] Zachary Ives, Nitim Khandelwal, Aneesh Kapur und Murat Cakir. Orchestra: Rapid, collaborative sharing of dynamic data. In *Proc. of the Conference on Innovative Data Systems Research (CIDR)*, 2005.

[Inm92] William H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, Wellesley, MA, USA, 1992.

[JLVV03] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou und Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer Verlag, 2003.

[JMP02] A. D. Jhingran, N. Mattos und H. Pirahesh. Information integration: A research agenda. *IBM Systems Journal*, 41(4):555–562, 2002.

[JMvG95] Manoj Jain, Anurag Mendhekar und Dirk van Gucht. A uniform data model for relational data and meta-data query processing. Technical Report TR436, Dept. of Computer Science, Indiana University, 1995.

[Kal05] O. Kallioniemi. Dissection of molecular pathways of cancer by high–throughput biochip technologies and RNA interference. *Breast Cancer Research*, 7:43, 2005.

[KAM03] Anastasios Kementsietsidis, Marcelo Arenas und Renee Miller. Mapping data in Peer-to-Peer systems: Semantics and algorithmic issues. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, Seiten 325–336, 2003.

[KBD+03] D. Karolchik, R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. T. Lu, K. M. Roskin, M. Schwartz, C. W. Sugnet, D. J. Thomas, R. J. Weber, D. Haussler und W. J. Kent. The UCSC Genome Browser Database. *Nucleic Acids Research*, 31(1):51–54, 2003.

[KDKR05] Toralf Kirsten, Hong-Hai Do, Christine Körner und Erhard Rahm. Hybrid integration of molecular–biological annotation data. In *Proc. of the 2nd International Workshop on Data Integration in the Life Sciences (DILS)*, 2005.

[KDR03] Toralf Kirsten, Hong-Hai Do und Erhard Rahm. A multidimensional data warehouse for gene expression analysis. In *Proc. of the German Conference on Bioinformatics (GCB), Poster Abstract Band*, 2003.

[KDR04a] Toralf Kirsten, Hong-Hai Do und Erhard Rahm. A data warehouse for multidimensional gene expression analysis. Technical Report 01-04, Interdisciplinary Centre for Bioinformatics, University of Leipzig, November 2004.

[KDR04b] Toralf Kirsten, Hong-Hai Do und Erhard Rahm. The IZBI gene expression analysis platform – A status report. Technical Report 02–04, Interdisciplinary Center for Bioinformatics, University of Leipzig, January 2004.

[KFD+03] O. D. King, R. E. Foulger, S. S. Dwight, J. V. White und F. P. Roth. Predicting gene function from patterns of annotation. *Genome Research*, 13(5):896–904, 2003.

[KG00] Minoru Kanehisa und Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[KIHS92] T. Kinoshita, J. Imamura, H.Nagai und K. Shimotohno. Quantification of gene expression over a wide range by the polymerase chain reaction. *Anal. Biochem.*, 206(2):231–235, 1992.

[KKDR05] Christine Körner, Toralf Kirsten, Hong-Hai Do und Erhard Rahm. Hybride Integration von molekularbiologischen Annotationsdaten. In *Proc. of the 11th Conference on Database Systems for Business, Technology and Web (BTW)*, 2005.

[KKS+92] A. Kallioniemi, O.P. Kallioniemi, D. Sudar, D. Rutovitz, J.W. Gray, F. Waldman und D. Pinkel. Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, 258(5083):818–821, 1992.

[KKS+04] Arek Kasprzyk, Damian Keefe, Damian Smedley, Darin London, William Spooner, Craig Melsopp, Martin Hammond, Philippe Rocca-Serra, Tony Cox und Ewan Birney. EnsMart: A generic system for fast and flexible access to biological data. *Genome Research*, 14(1):160–169, 2004.

[KLR06] Toralf Kirsten, Jörg Lange und Erhard Rahm. An integrated platform for analyzing molecular–biological data within clinical studies. In *Proc. of the International EDBT–Workshop on Information Integration in Healthcare Applications (IIHA)*, 2006.

[KMC00] Kathleen Kerr, Mitchell Martin und Gary A. Churchill. Analysis of variance for microarray gene expression data. *Journal of Computational Biology*, 7(6):819–837, 2000.

[KN01] Leonid Kruglyak und Deborah A. Nickerson. Variation is the spice of life. *Nature Genetics*, 27(3):234–236, 2001.

[Knu02] S. Knudsen. *A Biologist Guide to Analysis of DNA Microarray Data*. John Wiley & Sons, 2002.

[KO03] W. Kuchinke und C. Ohmann. "eTrials" werden zur Routine. *Deutsches Ärzteblatt*, 100(47):3081–3084, 2003.

[Koh97] Teuvo Kohonen. *Self Organizing Maps*. Springer Verlag, 1997.

[KR90] Leonard Kaufman und Peter J. Rousseeuw. *Finding Groups in Data – An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[KR06] Toralf Kirsten und Erhard Rahm. BioFuice: Mapping–based data intergation in bioinformatics. In *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences (DILS)*, 2006.

[KSB04] A. Kumar, B. Smith und C. Borgelt. Dependence relationships between Gene Ontology terms based on TIGR gene product annotations. In *Proc. 3rd International Workshop on Computational Terminology (CompuTerm)*, 2004.

[KSF+02] W. James Kent, Charles W. Sugnet, Terrence S. Furey, Krishna M. Roskin, Tom H. Pringle, Alan M. Zahler und David Haussler. The Human Genome Browser at UCSC. *Genome Research*, 12(6):996–1006, 2002.

[KTR07] Toralf Kirsten, Andreas Thor und Erhard Rahm. Instance-based matching of large life science ontologies. In *Proc. 4th International Workshop on Data Integration in the Life Sciences (DILS)*, 2007.

[Lä05] Jürgen Läuter. Hochdimensionale Statistik – Anwendung in der Genexpressionsanalyse. Leipzig Bioinformatics Working Paper (ISSN: 1860-2746) No. 7, Interdisciplinary Center for Bioinformatics, Leipzig, 2005.

[LBE03] Zoe Lacroix, Omar Boucelma und Mehdi Essid. The biological integration system. In *Proc. of the 5th ACM International Workshop on Web Information and Data Management*, 2003.

[LC03] Zoe Lacroix und Terence Critchlow. *Bioinformatics – Managing Scientific Data*. Morgan Kaufmann Publishers, 2003.

[LDB+96] David J. Lockhart, Helin Dong, Michael C. Byrne, Maximillian T. Follettie, Michael V. Gallo, Mark S. Chee, Michael Mittmann, Chunwei Wang, Michiko Kobayashi, Heidi Norton und Eugene L. Brown. Expression monitoring by hybridization to high–density oligonucleotide arrays. *Nature Biotechnology*, 14(13):1675–1680, 1996.

[LFGL99] Robert J. Lipshutz, Stephen P.A. Fodor, Thomas R. Gingeras und David J. Lockhart. High density synthetic oligonucleotide arrays. *Nature Genetics*, 21(1):20–24, 1999.

[Lin74] H. R. Lindman. *Analysis of variance in complex experimental designs*. W. H. Freeman & Co., 1974.

[LKR06] Jörg Lange, Toralf Kirsten und Erhard Rahm. An integrated analysis platform for experimental and clinical data in modern cancer research studies. In *Proceedings of the 51. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS)*, 2006.

[LKS07] Manja Lindemeyer, Toralf Kirsten und Peter Stadler. Molecular evolution of spliceosomal snRNA genes in Metazoan animals. *in submission*, 2007.

[LLB+05] Bertram Ludäscher, K. Lin, Shawn Bowers, E. Jaeger-Frank, B. Brodaric und C. Baru. Managing scientific data: From data integration to scientific workflow management. *GSA Today, Special Issue on Geoinformatics*, 2005.

[LLS+03] Guoying Liu, Ann E. Loraine, Ron Shigeta, Melissa Cline, Jill Cheng, Venu Valmeekam, Shaw Sun, David Kulp und Michael A. Siani-Rose. NetAffx: Affymetrix probesets and annotations. *Nucleic Acids Research*, 31(1):82–86, 2003.

[LMNR04a] Zoe Lacroix, Hyma Murthy, Felix Naumann und Louiqa Raschid. Links and paths through life science data sources. Technical Report, Humboldt-Universität zu Berlin, Institut für Informatik, 2004.

[LMNR04b] Zoe Lacroix, Hyma Murthy, Felix Naumann und Louiqa Raschid. Links and Paths through Life Sciences data sources. In *Proc. of the 1st International Workshop on Data Integration in the Life Sciences (DILS)*, 2004.

[LMR90] Witold Litwin, Leo Mark und Nick Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys (CSUR) – Special issue on heterogeneous databases*, 22(3):267–293, 1990.

[LN07] Ulf Leser und Felix Naumann. *Informationsintegration*. dpunkt.verlag, 2007.

[LP92] Peng Liang und Arthur B. Pardee. Differential display of eukaryotic messenger RNA by means of the polymerase chain reaction. *Science*, 257(5072):967–971, 1992.

[LP97] Peng Liang und Arthur B. Pardee. *Differential display methods and protocols*. Humana Press, 1997.

[LW01] Cheng Li und Wing Hung Wong. Model–based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences*, 98(1):31–36, 2001.

[MB04] Peter Mork und Philip Bernstein. Adapting a generic match algorithm to align ontologies of human anatomy. In *Proc. of the 20th International Conf. on Data Engineering (ICDE)*, 2004.

[Mec95] C. Mecucci. FISH (fluorescent in situ hybridization): The second youth of cytogenetics. *Haematologica*, 80(2):95–97, 1995.

[MIKS00] Eduardo Mena, Arantza Illarramendi, Vipul Kashyap und Amit P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre–existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[MLB+96] Glenn McGall, Jeff Labadie, Phil Brock, Greg Wallraff, Tiffany Nguyen und William Hinsberg. Light–directed synthesis of high–density oligonucleotide arrays using semiconductor photoresists. *Proceedings of the National Academy of Sciences*, 93:13555–13560, 1996.

[MOPT05] Donna Maglott, Jim Ostell, Kim D. Pruitt und Tatiana Tatusova. Entrez Gene: Gene-centered information at NCBI. *Nucleic Acids Research*, 33 (Database Issue):D54–D58, 2005.

[MSZ+01] H. Mangalam, J. Stewart, J. Zhou, K. Schlauch, M. Waugh, G. Chen, A. D. Farmer, G. Colello und J. W. Weller. GeneX: An open source gene expression database and integrated tool set. *IBM Systems Journal*, 40(1):552–569, 2001.

[MT01] Victor M. Markowitz und Thodoros Topaloglou. Applying data warehouse concepts to gene expression data management. In *Proc. of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, 2001.

[NAP04] Rakesh Nagarajan, Mushtaq Ahmed und Aditya Phatak. Database challenges in the integration of biomedical data sets. In *Proc. of the 30th International Conference on Very Large Data Bases (VLDB)*, 2004.

[NB98] Prakash M. Nadkarni und Cynthia Brandt. Data extraction and ad hoc query of an entity–attribute–value database. *Journal of American Medical Informatics Association*, 5(6):511–527, 1998.

[NLPM02] Felix Naef, Daniel A. Lim, Nila Patil und Marcelo Magnasco. DNA hybridization to mismatched templates: A chip study. *Physical Review E*, 65(4):040902–1–040902–4, 2002.

[NM03] Felix Naef und Marcelo O. Magnasco. Solving the riddle of the bright mismatches: Labeling and effective binding in oligonucleotide arrays. *Physical Review E*, 68(1):011906–1–011906–4, 2003.

[NOTZ03] Wee Siong Ng, Beng Chin Ooi, Kian Lee Tan und Aoying Zhou. PeerDB: A P2P-based system for distributed data sharing. In *Proc. of the 19th International Conference on Data Engineering (ICDE)*, 2003.

[OAF+04] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat und Peter Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[OCAM+04] P. V. Ogren, K. B. Cohen, G. K. Acquaah-Mensah, J. Eberlein und L. Hunter. The compositional structure of Gene Ontology terms. In *Proc. of the Pacific Symposium on Biocomputing*, 2004.

[OTZ+03] Beng Chin Ooi, Kian-Lee Tan, Aoying Zhou, Chin Hong Goh, Yingguang Li, Chu Yee Liau, Bo Ling, Wee Siong Ng, Yanfeng Shu, Xiaoyu Wang und Ming Zhang. PeerDB: Peering into personal databases. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2003.

[PAG96] Yannis Papakonstantinou, Serge Abiteboul und Hector Garcia-Molina. Object fusion in mediator systems. In *Proc. of the 22nd*

International Conference on Very Large Data Bases (VLDB), 1996.

[Pas94] Eberhard Passarge. *Taschenatlas der Genetik*. Georg Thieme Verlag, 1994.

[PBC+06] Andreas Prlic, Ewan Birney, Tony Cox, Thomas Down, Rob Finn, Stefan Gräf, David Jackson, Andreas Kahari, Eugene Kulesha, Roger Pettett, James Smith, Jim Stalker und Tim Hubbard. The Distributed Annotation System for integration of biological data. In *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences (DILS)*, 2006.

[PC05] Supawan Prompramote und Yi-Ping Phoebe Chen. ANNONDA: Tool for integrating molecular–biological annotation data. In *Proc. of the 21st International Conference on Data Engineering (ICDE)*, 2005.

[PCC+04] Simon C. Potter, Laura Clarke, Val Curwen, Stephen Keenan, Emmanuel Mongin, Stephen M.J. Searle, Arne Stabenau, Roy Storey und Michele Clamp. The Ensembl analysis pipeline. *Genome Research*, 14(5):934–941, 2004.

[PKH+00] Norman W. Paton, Shakeel A. Khan, Andrew Hayes, Fouzia Moussouni, Andy Brass, Karen Eilbeck, Carole A. Goble, Simon J. Hubbard und Stephen G. Oliver. Conceptual modelling of genomic information. *Bioinformatics*, 16(6):548–557, 2000.

[PM01] Kim D. Pruitt und Donna R. Maglott. RefSeq and LocusLink: NCBI gene–centered resources. *Nucleic Acids Research*, 29(1):137–140, 2001.

[PSS+05] H. Parkinson, U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, G. Garcia Lara, E. Holloway, M. Kapushesky, P. Lilja, G. Mukherjee, A. Oezcimen, T. Rayner, P. Rocca-Serra, A. Sharma, S. Sansone und A. Brazma. ArrayExpress – A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 33 (Database Issue):D553–D555, 2005.

[PW97] W. Piesch und S. With. Lorenz curve chords and weighted mean deviation sums. *Metron*, LV(3–4):3–19, 1997.

[PWS04] J.U. Pontius, L. Wagner und G.D. Schuler. UniGene: A unified view of the transcriptome. In: The NCBI Handbook. Technical Report, National Center for Biotechnology Information, 2004.

[RAD+06] Erhard Rahm, David Aumüller, Hong-Hai Do, Toralf Kirsten, Jörg Lange, Sabine Maßmann und Andreas Thor. Datenintegration in den Lebenswissenschaften – Ansätze und Werkzeuge. Vortrag auf dem Workshop zu Ontologien im Grid, März 2006.

[Rah94] Erhard Rahm. *Mehrrechner–Datenbanksysteme: Grundlagen der verteilten und parallelen Datenbankverwaltung.* Addison-Wesley, 1994.

[Rau01] Reinhard Rauhaut. *Bioinformatik. Sequenz – Struktur – Funktion.* John Wiley & Sons, 2001.

[RB01] Erhard Rahm und Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[RD00] Erhard Rahm und Hong-Hai Do. Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4):3–13, 2000.

[RGKG+05] Patricia Rodriguez-Gianolli, Anastasios Kementsietsidis, Maddalena Garzetti, Iluju Kiringa, Lei Jiang, Mehedi Masud, Renee J. Miller und John Mylopoulos. Data sharing in the Hyperion peer database system. In *Proc. of the 31st International Conference on Very Large Data Bases (VLDB), Demo Abstract Band*, 2005.

[RJSS00] Jan Reichert, Andreas Jabs, Peter Slickers und Jürgen Sühnel. The IMB Jena Image Library of biological macromolecules. *Nucleic Acids Research*, 28(1):246–249, 2000.

[RKL07] Erhard Rahm, Toralf Kirsten und Jörg Lange. The GeWare data warehouse platform for the analysis of molecular-biological and clinical data. *Journal of Integrative Bioinformatics*, 4(1):47, 2007.

[RMT+04] Kristian Rother, Heiko Müller, Silke Trissl, Ina Koch, Thomas Steinke, Robert Preissner, Cornelius Frömmel und Ulf Leser.

Columba: Multidimensional data integration of protein annotations. In *Proc. of the 1st International Workshop on Data Integration in the Life Sciences (DILS)*, 2004.

[RTA+05] Erhard Rahm, Andreas Thor, David Aumüller, Hong-Hai Do, Nick Golovin und Toralf Kirsten. iFuice – Information fusion utilizing instance-based peer mappings. In *Proc. of the 8th International Workshop on Web and Databases (WebDB)*, 2005.

[SBM+00] Johannes Schuchhardt, Dieter Beule, Arif Malik, Eryc Wolski, Holger Eickhoff, Hans Lehrach und Hanspeter Herzel. Normalization strategies for cDNA microarrays. *Nucleic Acids Research*, 28(10):E47–e47, 2000.

[SCC+02] Andrew I. Su, Michael P. Cooke, Keith A. Ching, Yaron Hakak, John R. Walker, Tim Wiltshire, Anthony P. Orth, Raquel G. Vega, Lisa M. Sapinoso, Aziz Moqrich, Ardem Patapoutian, Garret M. Hampton, Peter G. Schultz und John B. Hogenesch. Large–scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences*, 99(7):4465–4470, 2002.

[Sch98] Steffen Schulze-Kremer. Ontologies for molecular biology. In *Proc. of the 3rd Pacific Symposium on Biocomputing*, 1998.

[SGH+98] Wilhelm Seyfert, Hans Günter Gassen, Oswald Hess, Herbert Jäckle und Karl-Friedrich Fischbach (Hrsg.). *Lehrbuch der Genetik*. Gustav Fischer Verlag, 1998.

[SGP+03] Robert Stevens, Carol Goble, Norman W. Paton, Sean Bechofer, Gary Ng, Patricia Baker und Andy Brass. Complex Query Formulation over diverse Information Sources in Tambis. In Zoe Lacroix und Terence Critchlow (Hrsg.), *Bioinformatics – Managing Scientific Data*, Seiten 190–224. Morgan Kaufmann Publishers, 2003.

[SHBK+01] Gavin Sherlock, Tina Hernandez-Boussard, Andrew Kasarskis, Gail Binkley, John C. Matese, Selina S. Dwight, Miroslava Kaloper, Shuai Weng, Heng Jin, Catherine A. Ball, Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, David Botstein und J. Michael Cherry. The stanford microarray database. *Nucleic Acids Research*, 29(1):152–155, 2001.

[She98] A. Sheth. Changing focus on interoperability in information systems: From system, syntax, structur to semantics. In M. Goodchild, M. Egenhofer, R. Fegeas und C. Kottman (Hrsg.), *Interoperating Geographic Information Systems*, Seiten 5–30. Kluwer, 1998.

[SKR04] Ernst Schuster, Siegfried Kropf und Ingo Roeder. Micro array based gene expression analysis using parametric multivariate tests per gene – A generalized application of multiple procedures with data-driven order of hypothesis. *Biometrical Journal*, 46(6):687–698, 2004.

[SL90] Amit P. Shet und James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[SLS+97] Sabina Solinas-Toldo, Stefan Lampel, Stephan Stilgenbauer, Jeremy Nickolenko, Axel Benner, Hartmut Döhner, Thomas Cremer und Peter Lichter. Matrix-based comparative genomic hybridization: Biochips to screen for genomic imbalances. *Genes, Chromosomes and Cancer*, 20(4):399–407, 1997.

[SOH+06] Joel Saltz, Scott Oster, Shannon Hastings, Stephen Langella, Tashin Kurc, William Sanchez, Manav Kher, Arumani Mnisundaram, Krishnakant Shanbhag und Peter Covitz. caGRID: Design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics*, 22(15):1910–1916, 2006.

[SPM+01] C. Stoeckert, A. Pizarro, E. Manduchi, M. Gibson, B. Brunk, J. Crabtree, J. Schug, S. Shen-Orr, und G. C. Overton. A relational schema for both array–based and SAGE gene expression experiments. *Bioinformatics*, 17(4):300–308, 2001.

[SSDB95] M. Shena, D. Shalon, R.W. Davis und P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, 1995.

[Ste03] Lincoln Stein. Integrating Biological Databases. *Nature Review Genetics*, 4(5):337–345, 2003.

[SvHSS06] Heiner Stuckenschmidt, Frank van Harmelen, Wolf Siberski und Steffen Staab. Peer–to–Peer and Semantic Web. In Steffen

Staab und Heiner Stuckenschmidt (Hrsg.), *Peer–to–Peer and Semantic Web*, Seiten 1–17. Springer Verlag, 2006.

[SWMB95] R. Somogyi, X. Wen, W. Ma und J. L. Barker. Developmental kinetics of GAD family mRNAs parallel neurogenesis in the rat spinal cord. *Journal of Neuroscience*, 15(4):2575–2591, 1995.

[TBY+05] Tokiyushi Tanaka, Zhongbin Bai, Srinoulprasert Yuttana, Bogi Yang, Hayasaka Haruko und Miyasaka Masayuki. Chemokines in tumor progression and metastasis. *Cancer Science*, 96(6):317–322, 2005.

[TCF+] Steven Tuecke, Karl Czajkowski, Ian Foster, Jeffrey Frey, Steve Graham und Carl Kesselman. Grid Service Specification. http://esc.dl.ac.uk/WebServices/OGSA/gsspec.pdf.

[Tho07] Andreas Thor. *Automatische Mapping-Verarbeitung von Web-Daten*. Dissertation, Institut für Informatik, Universität Leipzig, 2007.

[TKR07] Andreas Thor, Toralf Kirsten und Erhard Rahm. Instance-based matching of hierarchical ontologies. In *Proc. of the 12th Conference on Database Systems for Business, Technology and Web (BTW)*, 2007.

[TOTZ01] Jeffrey G. Thomas, James M. Olson, Stephen J. Tapscott und Lue Ping Zhao. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research*, 11(7):1227–1236, 2001.

[TTC+90] Gomer Thomas, Glenn R. Thompson, Chin-Wan Chung, Edward Barkmeyer, Fred Carter, Marjorie Templeton, Stephen Fox und Berl Hartman. Heterogeneous distributed database systems for production use. *ACM Computing Surveys (CSUR) – Special issue on heterogeneous databases*, 22(3):237–263, 1990.

[VAM+01] J. Craig Venter, Mark D. Adams, Eugene W. Myers, Peter W. Li, Richard J. Mural, Granger G. Sutton, Hamilton O. Smith, Mark Yandell, Cheryl A. Evans, Robert A. Holt, Jeannine D. Gocayne, Peter Amanatides, Richard M. Ballew, Daniel H. Huson, Jennifer Russo Wortman et al. The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[VEB+98] George Vasmatzis, Magnus Essand, Ulrich Brinkmann, Byungkook Lee und Ira Pastan. Discovery of three genes specifically expressed in human prostate by expressed sequence tag database analysis. *Proceedings of the National Academy of Sciences*, 95(1):300–304, 1998.

[VSSV02] U. Visser, Heiner Stuckenschmidt, G. Schuster und T. Vögele. Ontologies for geographic information processing. *Computers & Geosciences*, 28(1):103–117, 2002.

[VZVK95] V. E. Velculescu, L. Zhang, B. Vogelstein und K. W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484–487, 1995.

[WBB+06] David L. Wheeler, Tanya Barrett, Dennis A. Benson, Stephen H. Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M. Church, Michael DiCuccio, Ron Edgar, Scott Federhen, Lewis Y. Geer, Wolfgang Helmberg, Yuri Kapustin, David L. Kenton, Oleg Khovayko, David J. Lipman, Thomas L. Madden, Donna R. Maglott, James Ostell, Kim D. Pruitt, Gregory D. Schuler, Lynn M. Schriml, Edwin Sequeira, Stephen T. Sherry, Karl Sirotkin, Alexandre Souvorov, Grigory Starchenko, Tugba O. Suzek, Roman Tatusov, Tatiana A. Tatusova, Lukas Wagner und Eugene Yaschenko. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 34 (Database Issue):D173–D180, 2006.

[WC53] James D. Watson und Francis H. Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

[WCE+04] David L. Wheeler, Deanna M. Church, Ron Edgar, Scott Federhen, Wolfgang Helmberg, Thomas L. Madden, Joan U. Pontius, Gregory D. Schuler, Lynn M. Schriml, Edwin Sequeira, Tugba O. Suzek, Tatiana A. Tatusova und Lukas Wagner. Database resources of the National Center for Biotechnology Information: update. *Nucleic Acids Research*, 32 (Database Issue):D35–D40, 2004.

[WF00] Ian H. Witten und Eibe Frank. *Data Mining – Practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann Publishers, 2000.

[WFS+98] David G. Wang, Jian-Bing Fan, Chia-Jen Siao, Anthony Berno, Peter Young, Ron Sapolsky, Ghassan Ghandour, Nancy Perkins, Ellen Winchester, Jessica Spencer, Leonid Kruglyak, Lincoln Stein, Linda Hsie, Thodoros Topaloglou, Earl Hubbell, Elizabeth Robinson, Michael Mittmann, Macdonald S. Morris, Naiping Shen, Dan Kilburn, John Rioux, Chad Nusbaumand Steve Rozen, Thomas J. Hudson, Robert Lipshutz, Mark Chee und Eric S. Lander. Large-scale identification, mapping and genotyping of single-nucleotide polymorphisms in the human genome. *Science*, 280(5366):1077–1082, 1998.

[WI04] Zhijin Wu und Rafael A Irizarry. Preprocessing of oligonucleotide array data. *Nature Biotechnology*, 22(6):656–658, 2004.

[Wie92] Gio Wiederhold. Mediators in the architecture of future information systems. *Computer (IEEE)*, 25(2):38–49, 1992.

[WIG+04] Zhijin Wu, Rafael A. Irizarry, Robert Gentleman, Francisco Martinez-Murillo und Forrest Spencer. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909–917, 2004.

[Won00] Limsoon Wong. Kleisli: A functional query system. *Journal of Functional Programming*, 10(1):19–56, 2000.

[WPC+06] Patricia L. Whetzel, Helen Parkinson, Helen C. Causton, Liju Fan, Jennifer Fostel, Gilberto Fragoso, Laurence Game, Mervi Heiskanen, Norman Morrison, Philippe Rocca-Serra, Susanna-Assunta Sansone, Chris taylor, Joseph White und Christian J. Stoeckert. The MGED ontology: A Resource for semantics-based description of microarray experiments. *Bioinformatics*, 22(7):866–873, 2006.

[WSW53] M.H. Wilkins, A.R. Stokes und H.R. Wilson. Molecular structure of deoxypentose nucleic acids. *Nature*, 171:738–740, 1953.

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann und S. Hübener. Ontology–based integration

of information – A survey of existing approaches. In *Proc. of the IJCAI Workshop on Ontologies and Information Sharing*, 2001.

[WY93] Peter H. Westfall und S. Stanley Young. *Resampling–Based Multiple Testing : Examples and Methods for p-Value Adjustment*. John Wiley & Sons, 1993.

[XSD+02] Ioannis Xenarios, Lukasz Salwinski, Xiaoqun Joyce Duan, Patrick Higney, Sul-Min Kim und David Eisenberg. DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.

[ZMPQ+02] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich und G. Cesareni. MINT: a Molecular INTeraction database. *FEBS Lett.*, 513(1):135–140, 2002.

Stichwortverzeichnis

A

Aggregierte Objekte ..... *siehe* AO
Aggregiertes Mapping-Resultat *siehe* AMR
AMR 148
Annotation Template ..... *siehe* GeWare
AO 148

B

BioFast 178
BioFuice 13, 154–169, 184
  Architektur 155
  BioFuice Query 155, 157, 161–163
  Datenstrukturen 147
  iFuice-Konzept 13, 139–153, 185
  Konzeptuelle Strukturen 143
  Operatoren 148–151
  RiFuice-Paket ... 13, 155, 157–161, 185
BLAST 13, 143, 152

C

Columba 126, 171, 173

D

DM 145
Domänenmodell ..... *siehe* DM

F

Fasta 6, 13, 141, 154, 155, 157, 185

G

Genexpressionsanalyse
  Ablauf 42
  Daten 45, 49, 70
  Datenanalyse 53, 62, 83, 117, 164
  Microarray 2, 6, 41, 70, 93
    Arten 42
    Prinzip 41
  Visualisierung 55, 64, 84, 117
  Vorverarbeitung 53, 61, 82
GenMapper 125, 171, 174
GeWare 10, 68–92, 96, 113, 182
  Analyseintegration 88
  Annotation Template .. 71, 77, 78, 115
  Architektur 69
  Data Warehouse Schema 74
  Expressionsanalyse 82–84, 117
  kontrolliertes Vokabular 78
  Metadaten-Management 71, 77
  Workflows
    Analyseprozesse 74
    Importprozesse 73

I

Instanzdatenintegration
  Integration, physisch 32
  Integration, virtuell 31
  Vergleich 34
Integrationsformen
  Data Warehouse 33, 170
  Mapping-basierte Integration . 26, 140, 172
  Mediator 170
  Mediatoren 31
  mit homogenisierter Sicht 16
    applikationsspezifisch . 16, 18–21, 170
    generisch 16, 21–24, 171
    globale Ontologie 17, 24–25, 172
  PDMS 27, 140, 172

K

klinische Daten 71, 110
  Integration 11, 90, 113, 183

M

Mapping 26, 120
Mapping-Datenbank 124, 126
Mapping-Resultat ..... *siehe* MR
MR 148
Mutationsanalyse 44, 111
  Chip-Technik 44
  Daten 45, 70

O

Objektinstanzen ..... *siehe* OI

OI 147
Ontologie 17
-Matching 186
globale ˜ . . . . . *siehe* Integrationsformen

S
SMM 144
Source-Mapping-Modell . . . . . *siehe* SMM
SRS 26, 121, 126, 175

Z
Zelle
Aufbau, Bestandteile 38
Prozesse
Genexpression 40
Transkription 40
Translation 40

Lebenslauf und wissenschaftlicher Werdegang des Verfassers

Allgemeine Angaben:

Geburtsdatum:	10.07.1972
Geburtsort:	Altdöbern
Nationalität:	deutsch
Staatsangehörigkeit:	BRD
Status:	ledig, 1 Kind
wohnhaft in:	Leipzig, Deutschland

Wissenschaftlicher Werdegang:

2002 –	wissenschaftlicher Mitarbeiter Interdisziplinäres Zentrum für Bioinfomatik Leipzig, Arbeitsgruppe Datenbanken und Datenintegration
1994 – 1999	Studium der Betriebswirtschaft Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH) Studienschwerpunkte: ⋆ Wirtschaftsinformatik ⋆ Materialwirtschaft/Logistik, Produktions- und Anlagenwirtschaft ⋆ Marketing Abschluss: Dipl.-Kaufmann (FH)
1992 – 1993	Fachoberschule Liebertwolkwitz, Fachrichtung Wirtschaft Abschluss: Fachhochschulreife
1989 – 1992	Berufsausbildung Datenverarbeitungszentrum Leipzig GmbH Abschluss: EDV-Kaufmann (IHK)
1979 – 1989	Polytechnische Oberschulen in Lübbenau und Leipzig Abschluss: 10. Klasse (entspricht: Mittlerer Reife)

Praktische Erfahrungen:

2001-01 –	Interdisziplinäres Zentrum für Bioinformatik Leipzig Forschungsleistungen zu Datenverwaltungs- und -integrationslösungen im Bereich der Bioinformatik
1999-02 – 2001-12	Mummert Consulting Konzeption und Implementierung von ⋆ Data Warehouse Architekturen ⋆ Datenmodellen ⋆ Datenintegration/-logistik ⋆ Analyseanwendungen, OLAP und Berichtswesen in verschiedenen Kundenprojekten und Branchen
1998-02 – 1998-08	Mummert Consulting Praktikum: Mitarbeit in verschiedenen SAP R/3 Kundenprojekten
1997-10 – 1998-01	Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH) Mitarbeit in Lehre und Forschung bei Prof. Klaus Kruczynski
1997-03 – 1997-09	Stadtwerke Düsseldorf AG Praktikum: Prozesskettenmodellierung und -analyse mit dem ARIS Toolset zur Umsetzung von automatisierten Workflows
1995-07 – 1997-02	Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH) Mitarbeit in Lehre und Forschung bei Prof. Klaus Kruczynski
1995-01 – 1995-07	Symdata GmbH Leipzig Tutor für verschiedene Softwaresysteme

Auszeichnungen:

- Ehrenmedaille der Bundeswehr in Bronze, 1994

Dissertationsbzogene bibliographische Daten

Toralf Kirsten

Data-Warehouse- und Mapping-basierte Datenintegrationsplattformen in der Bioinformatik

Dissertation, Fakultät für Mathematik und Informatik, Universität Leipzig
Fachgebiet Informatik (Bioinformatik)
2007

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder Dienstleistungen als solche gekennzeichnet.

_____

(Ort, Datum)

_____

(Unterschrift)