Universität Bielefeld
Technische Fakultät
Arbeitsgruppe Bioinformatik und Medizinische Informatik

Identifikation von potenziellen Transkriptionsfaktorbindestellen in Nukleotidsequenzen basierend auf einem Data-Warehouse-System

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

vorgelegt der Technischen Fakultät
der Universität Bielefeld

von: Dipl.-Inform. Klaus Hippe
geb. am: 05. Dezember 1980 in Melle

Bielefeld, Juli 2014

Dipl.-Inform. Klaus Hippe:
Identifikation von potenziellen Transkriptionsfaktorbindestellen in Nukleotidsequenzen basierend auf einem Data-Warehouse-System

Der Technischen Fakultät der Universität Bielefeld
am 14. Januar 2014 vorgelegt,
am 03. Juli 2014 verteidigt und genehmigt.

Gutachter:
Prof. Dr. Ralf Hofestädt, Universität Bielefeld
Prof. Dr. Thomas Dierks, Universität Bielefeld

Prüfungsausschuß:
apl. Prof. Dr. Karl Friehs, Universität Bielefeld
Prof. Dr. Ralf Hofestädt, Universität Bielefeld
Prof. Dr. Thomas Dierks, Universität Bielefeld
Dr.-Ing. Sebastian Wrede, Universität Bielefeld

235 Seiten
69 Abbildungen
39 Tabellen

Gedruckt auf alterungsbeständigem Papier (ISO 9706)

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltextverzeichnis

Abkürzungsverzeichnis

AFL	Academic Free License
AKID	Atomarität, Konsistenz, Isoliertheit, Dauerhaftigkeit
AnGEL	Annotation Grammar and Extraction Tool
ASF	Apache Software Foundation
AS	Aminosäure
BASE	Basically Available, Soft State, Eventually Consistent
BCNF	Boyce-Codd-Normalform
BED	Browser Extensible Data
Berkeley DB	Berkeley-Datenbank
BMBF	Bundesministerium für Bildung und Forschung
bp	Basenpaar
BRENDA	Braunschweig Enzyme Database
CAP	Consistency, Availability und Partition Tolerance
cDNA	complementary DNA
CD-ROM	Compact Disc Read-Only Memory
ChIP	Chromatin-Immunopräzipitation
CHO	Chinese Hamster Ovary
CLEAR	Coordinated Lysosomal Expression and Regulation
CMR	Comprehensive Microbial Resource
CRUD	Create, Read, Update, Delete
CSS	Core similarity score
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
C_T-Wert	Cycle Threshold
DAWIS-M.D.	Data Warehouse Information System for Metabolic Data
DB	Datenbank
DBMS	Datenbankmanagementsystem
DBS	Datenbanksystem
DES	Data Encryption Standard
DFG	Deutsche Forschungsgemeinschaft

DIN	Deutsche Institut für Normung
DNA	Deoxyribonucleic acid
DWH	Data-Warehouse
EA	Evolutionäre Algorithmen
EBI	European Bioinformatics Institute
E-box	Enhancer Box
EC-Nummer	Enzyme Commission numbers
EET	Enzymersatztherapie
EMSA	Electrophoretic Mobility Shift Assay
ERM	Entity-Relationship-Modell
ESA	Enhanced Suffix-Array
ETL	Extraktion, Transformation, Laden
EU	Europäische Union
FCFS	First-come, first-served
FDBS	Föderiertes Datenbanksystem
FGE	Sulfatase-modifying factor-1
FGF-2	Fibroblast growth factor-2
FURPS	Functionality, Usability, Reliability, Performance and Supportability
GAG	Glykosaminoglykane
GBO	Grafische Benutzeroberfläche
GMS	Genomatix Mining Station
GNN	Genome News Network
GO	Gene Ontology
GPL	GNU General Public License
HD	Hydrophile Domäne
HEK-293	Human Embryonic Kidney
HeLa	Henrietta Lacks
HIF-1-α	Hypoxia-inducible factor 1-α
HIF-2-α	Hypoxia-inducible factor 2-α
HKI	Hans-Knöll-Institut
HMM	Hidden Markov Model
HNF4	Hepatocyte nuclear factor 4
HPRD	Human Protein Reference Database
HQL	Hibernate Query Language
HS	Heparansulfat
HSPG	Heparansulfat-Proteoglykane
HTE	High Throughput Experimentation
HTML	Hypertext Markup Language
HTS	High-Throughput-Screening

HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IG	IntelliGenetics
IMD	Information Matrix Database
IS	Informationssystem
ISO	International Organization for Standardization
IUPAC	International Union of Pure and Applied Chemistry
JAK	Januskinase
Java EE	Java Platform, Enterprise Edition
JDBC	Java Database Connectivity
JSF	JavaServer Faces
JSP	JavaServer Pages
JVM	Java Virtual Machine
KEGG	Kyoto Encyclopedia of Genes and Genomes
KIS	Krankenhausinformationssystem
KMT	Knochenmarktransplantation
lac-Operon	Lactose-Operon
LAF	Look and Feel
LGP	Längstes gemeinsames Präfix
LIMS	Labor-Informations- und Management-System
LSK	Lysosomale Speicherkrankheit
MB	Megabyte
MBS	Mediatorbasiertes Integrationssystem & Wrapper
MEME	Multiple EM for Motif Elicitation
MPL	Mozilla Public License
mRNA	Messenger RNA
MSS	Matrix similarity score
MPS	Mukopolysaccharidose
MVC	Model View Controller
NAR	Nucleic Acids Research
NF-κB	Nuclear factor kappa-light-chain-enhancer of activated B-cells
NGS	Next Generation Sequencing
NKX2-2	Homeobox protein Nkx-2.2
ODBMS	Objektdatenbankmanagementsystem
ODS	Operational Data Store
OLAP	Online Analytical Processing
ORF	Open reading frame

OLTP	Online-Transaction-Processing
OMIM	Online Mendelian Inheritance in Man
OPM	Object Protocol Model
ORDBMS	Objektrelationales Datenbankmanagementsystem
OSS	Open-Source-Software
p53	Cellular tumor antigen p53
PAX6	Paired box protein Pax-6
PBS	Primerbindungstelle
PCR	Polymerase-Kettenreaktion
PDB	Protein Data Bank
PDF	Portable Document Format
PDMS	Peer-Daten-Management System
PFM	Position frequency matrix
PKU	Phenylketonurie
PPI	Protein-Protein-Interaktion
pre-mRNA	Precursor mRNA
PSSM	Position-specific scoring matrix
PTM	Posttranslationale Modifikation
qRT-PCR	quantitative Real-Time-PCR
RDBMS	Relationales Datenbankmanagementsystem
RI	Referentielle Integrität
RMI	Remote Method Invocation
RNA	Ribonucleic acid
RPC	Remote Procedure Call
rRNA	Ribosomale RNA
SBGN	Systems Biology Graphical Notation
SBML	Systems Biology Markup Language
SCOP	Structural Classification of Proteins
SE	Software-Ergonomie
Shh	Sonic hedgehog
SiTaR	Site Tracking and Recognition
SOAP	Simple Object Access Protocol
SRF	Serum-Response-Faktor
SRS	Sequence Retrieval System
SQL	Structured Query Language
STAT	Signal Transducers and Activators of Transcription
Sulf1	Sulfatase 1
Sulf2	Sulfatase 2
TAMBIS	Transparent Access to Multiple Bioinformatics

	Information Sources
TESS	Transcription Element Search System
TF	Transkriptionsfaktor
TFBS	Transkriptionsfaktorbindestelle
TFEB	Transcription factor EB
TGF	Transforming Growth Factor
TraBi	Transcription Factor Binding Site Prediction
tRNA	Transfer-RNA
TSP	Transkriptionsstartpunkt
TSV	Tab-Separated Values
UCSC	University of California, Santa Cruz
UML	Unified Modeling Language
VANESA	Visualization and Analysis of Networks in System Biology
VANTED	Visualization and Analysis of Networks containing Experimental Data
VEGF	Vascular Endothelial Growth Factor
vHNF1	Variant hepatic nuclear factor 1
Wnt	Wingless Int-1
WOTD-Algorithmus	Write-Only-Top-Down-Algorithmus
WT1	Wilms tumor protein
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

1 Einleitung

Das erste Kapitel gibt eine Übersicht über die Zielsetzung und die Struktur der vorliegenden Arbeit. Als erstes wird in Abschnitt 1.1 die Motivation und die Notwendigkeit der Integration von verteilten und heterogenen molekularbiologischen Daten in der Bioinformatik erläutert. Diese Thematik ist eine Voraussetzung für weitere Analysen, Simulationen und Vorhersagen durch bioinformatische Softwarelösungen, die ebenfalls in Abschnitt 1.1 thematisiert werden. Danach behandelt der Abschnitt 1.2 die Zielsetzung und die eigentliche Aufgabenstellung der Dissertation. Abschließend wird in Abschnitt 1.3 die Struktur der Arbeit beschrieben.

1.1 Motivation

Die unterschiedlichen Forschungsgebiete der Lebenswissenschaften erzeugen überwiegend durch *High-Throughput-Screening* (HTS)/*High Throughput Experimentation* (HTE) eine immense und vielschichtige Datenmenge. In der Regel werden Datenbanksysteme (DBS) für solche Datenbestände eingesetzt, weil auf diese Weise eine effiziente, persistente und widerspruchsfreie Speicherung, Manipulation und Verwaltung der Daten gewährleistet wird. Diese DBS sind häufig eine grundlegende Komponente eines webbasierten Informationssystems (IS), wodurch molekularbiologische Daten für die akademische und industrielle Grundlagenforschung global und frei verfügbar sind. Es sind aber nicht alle experimentellen Datenbestände und deren Erkenntnisse für die Öffentlichkeit frei zugänglich, weil auch kommerzielle Aspekte, Patentschutz und „Wissenschaftsspionage“ berücksichtigt werden müssen.

In den letzten Jahren ist die Anzahl der molekularbiologischen Datenbanken (DB) exponentiell angestiegen (siehe Abbildung 1.1). Das jährliche *Database Issue* der Zeitschrift *Nucleic Acids Research*[1] (NAR) listet derzeit etwa 1552 molekularbiologische DB, die Informationen aus unterschiedlichen Kategorien der Molekularbiologie bereitstellen [FSRG14]. Diese Informationen resultieren ursprünglich aus Experimenten und/oder der wissenschaftlichen Literatur und repräsentieren entweder den Genotyp oder den Phänotyp eines Individuums. Eine nachträgliche Analyse der Daten kann durch spezialisierte Anwendungssoftware erfolgen, welche die integrati-

[1] http://nar.oxfordjournals.org/

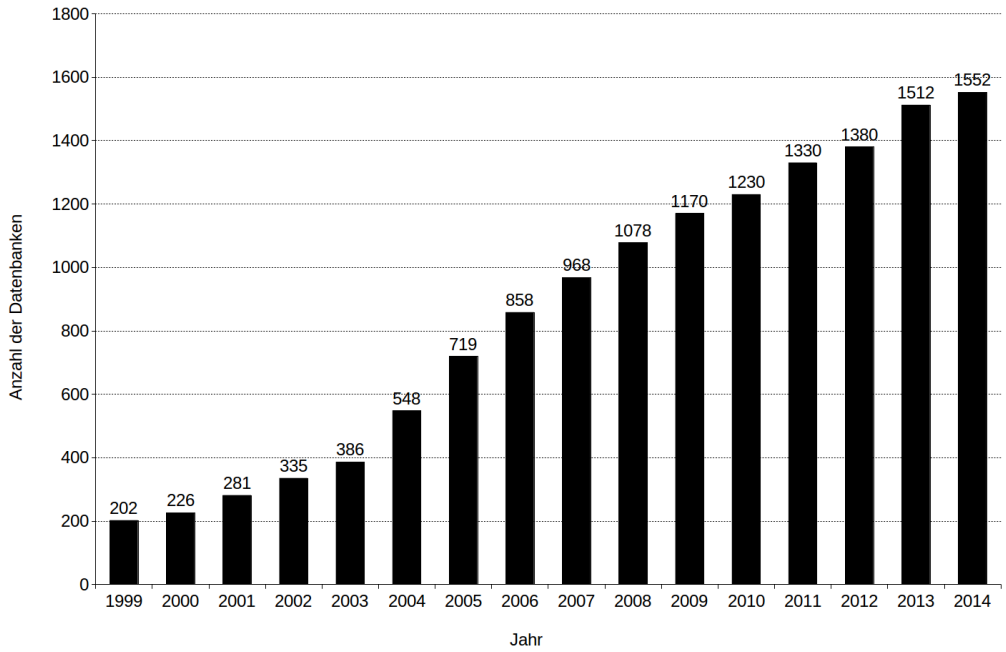


Abbildung 1.1: Entwicklung der Anzahl molekularbiologischer DB nach [FSRG14].

ve/praktische Bioinformatik zur Verfügung stellt. Darüber hinaus werden einige Informationen als zwei- oder dreidimensionales Netzwerk oder durch andere grafische Elemente dargestellt. Dabei handelt es sich häufig um Stoffwechselwege, Nukleotid- und Aminosäuresequenzen, Protein-Protein-Interaktionen (PPI), Proteinstrukturen oder chemische Strukturformeln.

Insbesondere die Deutsche Forschungsgemeinschaft[2] (DFG), das Bundesministerium für Bildung und Forschung[3] (BMBF) und die Europäische Union[4] (EU) als auch die freie Wirtschaft finanzieren fortwährend neue und bestehende Forschungsprojekte oder akademische Nachwuchsgruppen. Infolgedessen werden regelmässig neue molekularbiologische DB entwickelt oder bereits existierende Datenquellen aktualisiert und weiterentwickelt. Sobald die Projektfinanzierung eingestellt wird, ist die Pflege/Weiterentwicklung, die Verfügbarkeit/Erreichbarkeit als auch die Informationsqualität und -aktualität einer Datenquelle nicht mehr sichergestellt. Deshalb können zahlreiche molekularbiologische DB als obsolet bezeichnet werden. Daraus ergibt sich, dass die molekularbiologischen DB einer hohen Fluktuation unterliegen. Allerdings ist diese Problematik kein Phänomen, das ausschließlich molekularbiolo-

[2]http://www.dfg.de/
[3]http://www.bmbf.de/
[4]http://europa.eu/

gische DB betrifft, auch unzählige Softwarelösungen aus der (Bio-)Informatik sind davon betroffen.

Die Entwicklung der Anzahl der Datenbankeinträge wird in der Abbildung 1.2 für sieben populäre Datenquellen dargestellt. Hieran wird deutlich, dass die eigentlichen

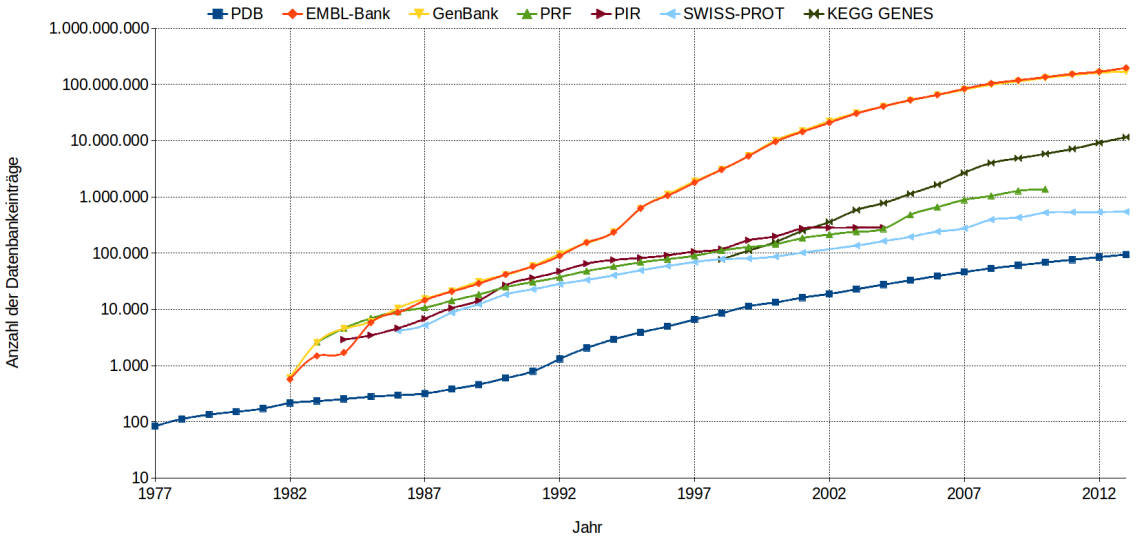


Abbildung 1.2: Entwicklung der Anzahl der Datenbankeinträge bei ausgewählten molekularbiologischen DB.

Datenbankeinträge ebenfalls exponentiell angestiegen sind. Dafür sind der technologische Fortschritt als auch die computergestützte Laborautomatisierung eine naheliegende Erklärung.

Die zahlreichen molekularbiologischen DB umfassen eine umfangreiche und variantenreiche Datenmenge und fokussieren auf verschiedene Kategorien der Molekularbiologie. Außerdem verfügen die Datenquellen selten über einheitliche und standardisierte Schnittstellen, sodass der direkte Datenzugriff und -austausch für externe Softwarelösungen nicht ohne Weiteres realisiert werden kann. Dadurch wird die wissenschaftliche Recherche für ganzheitliche Fragestellungen stark beeinträchtigt, weil unterschiedliche Datenquellen und deren Informationen berücksichtigt werden müssen. Insbesondere (in)direkte Wechselwirkungen und Abhängigkeiten zwischen Gen, Protein und Krankheit sowie genetische Regulationsmechanismen sind für die Fragestellungen der Lebenswissenschaften bedeutungsvoll. Dieses durchaus zeitintensive Recherchieren und Zusammenstellen der relevanten Informationen ist äußerst ineffizient und aufwendig, wodurch wichtige Zeit unnötig verschwendet wird. Abhilfe kann ein webbasiertes IS schaffen, das eine integrierte Sicht auf eine zugrundeliegende Datenbasis bereitstellt, die wiederum Datenbestände aus unterschiedlichen molekularbiologischen DB kombiniert. Eine zentrale, konsistente und strukturierte Datenbasis kann durch spezielle Strategien der Datenintegration angelegt werden, die eine grundlegende Thematik der Bioinformatik ist. Die molekularbiologischen DB variieren bei der Verteilung, Autonomie und Heterogenität und sind die drei Grund-

probleme der Datenintegration. Diese Probleme werden auch als die orthogonalen Dimensionen der Datenintegration bezeichnet [LN07]. Es gibt in der Bioinformatik entsprechende Softwarelösungen, die das Erstellen einer solchen Datenbasis ermöglichen und die drei Grundprobleme der Datenintegration durch geeignete Integrationsarchitekturen beseitigen. Allerdings verfügen die derzeitigen Softwarelösungen und deren Integrationsarchitekturen über gewisse Vor- und Nachteile, informationstechnische Restriktionen oder sind auf eine bestimmte Fragestellung oder Thematik spezialisiert.

In der Regel werden die klassischen Experimente der Lebenswissenschaften entweder *in vitro* oder *in vivo* durchgeführt. Diese beiden Vorgehensweisen sind zeit- und kostenaufwendig, verbrauchen Ressourcen und liefern im schlechtesten Fall keine eindeutigen Ergebnisse. Eine Alternative zur klassischen Vorgehensweise sind computergestützte Simulationen und Vorhersagen der Bioinformatik, die als *in silico* Experimente bezeichnet werden. Dafür bietet die Bioinformatik spezifische Anwendungssoftware, die kommerziell oder frei verfügbar ist und verschiedene Vor- und Nachteile besitzt. Durch diese Softwarelösungen können Simulationen und Vorhersagen durchgeführt oder existierende Daten nachträglich analysiert werden. Sofern die daraus resultierenden Ergebnisse vielversprechend sind, müssen diese erst durch Laborexperimente und letztendlich durch klinische Studien verifiziert werden. Ein traditionelles Forschungsgebiet der Bioinformatik ist die Analyse und die Interpretation der Genregulation, welche für die Steuerung der Genexpression zuständig ist. Insbesondere regulatorische Nukleotidsequenzen, die Transkriptionsfaktorbindestellen (TFBS), und spezielle Proteine, die Transkriptionsfaktoren (TF), sind verantwortlich für das Aktivieren oder Reprimieren der Transkription. Die computergestützte Vorhersage von potenziellen TFBS in Nukleotidsequenzen ist in der Bioinformatik eine Herausforderung, weil Zeit- und Platzkomplexität der Algorithmik wichtig ist und möglichst viele falsch positive Ergebnisse beseitigt werden müssen. Zudem ist eine konsistente und strukturierte Datenbasis, die alle erforderlichen Datenbestände zur Verfügung stellt, zwingend notwendig. Es gibt unterschiedliche Strategien und Softwarelösungen für die Identifikation von potenziellen TFBS in Nukleotidsequenzen, die aber einige Vor- und Nachteile aufweisen und anderen informationstechnischen Beschränkungen unterliegen.

1.2 Zielsetzung

Der Schwerpunkt der Dissertation liegt auf zwei verschiedenen und eigenständigen Ansätzen aus der Bioinformatik und deren Implementierung. Diese beiden Ansätze sollen hauptsächlich Wissenschaftler aus den Lebenswissenschaften aktiv bei deren unterschiedlichen Forschungsprojekten unterstützen. Insbesondere bei der Softwareentwicklung sind zwei naturwissenschaftliche Arbeitsgruppen der Universität Bie-

lefeld[5] beteiligt, deren molekularbiologische Expertise und Forschungsschwerpunkte bei der Realisierung der Software berücksichtigt werden. Die ersten beiden Prototypen der Software sollen durch die Arbeitsgruppe Biochemie I[6] und den Lehrstuhl für Zellbiologie[7] validiert werden. Durch diese Validierung der Software ist ein hohes Maß an Softwarequalität sichergestellt. Außerdem sollen für beide Arbeitsgruppen fachspezifische *in silico* Experimente durchgeführt werden, die als Grundlage für Laborexperimente fungieren, Hypothesen bestätigen und/oder neue Erkenntnisse liefern können.

Das Ziel der vorliegenden Arbeit ist der Entwurf und die Implementierung eines webbasierten IS zur computergestützten Vorhersage von potenziellen TFBS in Nukleotidsequenzen. Dabei müssen besonders die Zeit- und Platzkomplexität der Algorithmik beachtet werden, weil diese Kriterien die Grundlage für eine performante Software sind. Infolgedessen sollte ein Algorithmus mit einer linearen Laufzeit und einer effizienten Datenstruktur bevorzugt werden. Aufgrund der Komplexität sollte eine Lastverteilung möglich sein, sodass die eigentliche Vorhersage auf einem separaten Rechnerverbund durchgeführt werden kann. Die TFBS sind häufig in unmittelbarer Nähe der Gene lokalisiert, weshalb die 5'-Upstream-Region und die 3'-Downstream-Region der Gene von besonderen Interesse sind. Allerdings gibt es auch TFBS, die etliche Basenpaare (bp) von einem Gen entfernt sind. Deshalb sollten potenzielle TFBS in Nukleotidsequenzen identifiziert werden, die jeweils eine Länge von 2500 bp, 5000 bp oder 10000 bp repräsentieren und auf der 5'-Upstream-Region oder der 3'-Downstream-Region der jeweiligen Gene eines Organismus basieren. Der erste Prototyp der Software sollte ausschließlich drei eukaryotische Organismen berücksichtigen, welche für die aktuelle Grundlagenforschung eine besondere Signifikanz darstellen. Die erforderlichen Datensätze über Organismen, Gene, TF und TFBS sollte ein Data-Warehouse (DWH) zur Verfügung stellen, das einen umfangreichen molekularbiologischen Datenbestand beinhaltet. Die Ergebnisse einer Vorhersage müssen verständlich und strukturiert dargestellt werden. Ein Export der Ergebnisse in standardisierte Dateiformate wie Microsoft Excel sollte ebenfalls möglich sein.

Der erste Prototyp eines webbasierten Data-Warehouse-System für molekularbiologische Daten ist ebenfalls ein Bestandteil der vorliegenden Arbeit und soll mittels *Reengineering* und *Refactoring* sukzessive optimiert werden. Das zugrundeliegende DWH basierte ursprünglich auf Datenbeständen aus 11 unterschiedlichen molekularbiologischen Datenquellen wie EMBL-Bank [CAB+09], *Kyoto Encyclopedia of Genes and Genomes* (KEGG) [KGS+12] und *Universal Protein Resource* (UniProt) [The12b]. Zielsetzungen sind die Optimierung der Datenbankschemata, die Aktualisierung der Datenbestände und die Integration von weiteren molekularbiologischen Datenquellen wie die *Eukaryotic Promoter Database* (EPD) [SPPB06] und JASPAR [PCTK+09]. Außerdem sollte das DWH um eine zusätzliche DB ergänzt werden, die ausschließlich für die persistente Speicherung von Metadaten zuständig ist. Das

[5] http://www.uni-bielefeld.de/
[6] http://www.uni-bielefeld.de/chemie/bc1/
[7] http://web.biologie.uni-bielefeld.de/cellbiology/

webbasierte Data-Warehouse-System stützt sich auf ein Datenmodell, das in Folge der Aktualisierung und Erweiterung der Datenbasis vollständig überarbeitet werden sollte, weil neue Domänen und Beziehungen verfügbar sein werden. Die Beziehungen und Abhängigkeiten zwischen den unterschiedlichen Domänen müssen identifiziert und übersichtlich dargestellt werden. Des Weiteren sollte diese verbesserte Software flexible Suchformulare, eine dynamische Komponente zur Netzwerkvisualisierung und einen Export der Daten in standardisierte Dateiformate bereitstellen.

Darüber hinaus müssen diese beiden Softwarelösungen über das Internet frei verfügbar sein und nicht-funktionale Anforderungen wie Plattformunabhängigkeit, Benutzerfreundlichkeit, Skalierbarkeit, Wiederverwertbarkeit und Erweiterbarkeit gewährleisten. Ein weiteres Merkmal der beiden Softwarelösungen sollen interaktive und kollaborative Bestandteile sein, die besonders durch das Schlagwort *Web 2.0* repräsentiert werden.

1.3 Struktur der Arbeit

Die Struktur der vorliegenden Arbeit untergliedert sich in einen theoretischen und einen praktischen Teil, wobei der theoretische Teil durch die Kapitel 2 und 3 dargestellt wird. Im Gegensatz dazu repräsentieren die Kapitel 4, 5 und 6 den praktischen Teil der Dissertation.

Als erstes werden in Kapitel 2 die notwendigen Grundlagen der Molekularbiologie und der Informatik behandelt, welche zum besseren Verständnis der vorliegenden Arbeit wichtig sind. Allerdings werden Grundkenntnisse in der Molekularbiologie und der Informatik vorausgesetzt, weil die entsprechenden Sachverhalte nicht detailliert im Rahmen der vorliegenden Arbeit beschrieben werden können. Deswegen wird auf geeignete Literatur verwiesen, die diese Thematik ausführlich darstellt und eine angemessene Einführung bietet.

Danach werden in Kapitel 3 die verwandten Arbeiten und deren Funktionalität sowie die einzelnen Vor- und Nachteile thematisiert. Es werden in erster Linie populäre und aktuelle Softwarelösungen aus der Literatur erläutert, die den derzeitigen Stand der Forschung repräsentieren. Aufgrund der Zielsetzung werden ausschließlich Softwarelösungen aus der Bioinformatik berücksichtigt, die entweder Strategien und Techniken zur Integration von molekularbiologischen Datenquellen bereitstellen oder die Identifizierung von regulatorischen Elementen in Nukleotidsequenzen ermöglichen.

Durch die Analyse der verwandten Arbeiten in Kapitel 3 und die daraus resultierenden Ergebnisse konnten wichtige Erkenntnisse für den praktischen Teil identifiziert werden, die besonders bei Kapitel 4 und 5 berücksichtigt werden. Das Kapitel 4 beschreibt präzise die Anforderungsanalyse und die Systemarchitektur der Softwarelösungen, die während der Dissertation konzipiert und implementiert oder

kontinuierlich weiterentwickelt wurden. Darüber hinaus wird die Systemarchitektur einer plattformunabhängigen Software-Infrastruktur behandelt, die einen benutzerfreundlichen Funktionsumfang zur Integration von molekularbiologischen DB zur Verfügung stellt, der wiederum grundlegend für die Dissertation ist.

Anschließend wird in Kapitel 5 ausführlich die eigentliche Implementierung und das Design der jeweiligen Software thematisiert. Dabei werden besonders der Funktionsumfang, die Struktur und die Komponenten der einzelnen Software erörtert. Außerdem werden Programmbibliotheken/Programmierschnittstellen und zusätzliche Software und Technologien aus der Informatik dargestellt, die bei der Realisierung der Systemarchitektur und deren unterschiedlichen Schichten erforderlich sind.

Kapitel 6 erläutert einen Anwendungsfall aus der molekularbiologischen Grundlagenforschung, durch den die Funktionalität und der Anwendungsbereich der beiden Softwarelösungen in der Praxis verdeutlicht werden. Die molekularbiologische Thematik des Anwendungsfalls thematisiert einerseits die humanen Sulfatasen (siehe Abschnitt 6.1), andererseits einen spezifischen TF, der die Genexpression lysosomaler Hydrolasen, lysosomaler Membranproteine aber auch nicht-lysosomaler Proteine koordiniert (siehe Abschnitt 6.2). Diese Thematik ist die Grundlage für die *in silico* Experimente und die Laborexperimente, die in Kooperation mit der Arbeitsgruppe Biochemie I während des Graduiertenprogramms *Bioinformatics of Signaling Networks*[8] durchgeführt wurden. Die *in silico* Experimente und deren Ergebnisse sowie die daraus resultierenden Laborexperimente werden in Abschnitt 6.3 diskutiert.

Abschließend werden in Kapitel 7 die wesentlichen Aspekte der vorliegenden Arbeit zusammengefasst und zukünftige Weiterentwicklungen und Verbesserungen der einzelnen Software diskutiert.

Die Internetadressen der molekularbiologischen DB und der verwandten Arbeiten, die während der Dissertation verwendet oder analysiert wurden, werden im Anhang A dargestellt. Darüber hinaus werden einige Internetadressen von Softwarelösungen aufgelistet, die von der Arbeitsgruppe Bioinformatik und Medizinische Informatik[9] an der Universität Bielefeld entwickelt wurden. Der Anhang B zeigt den Quelltext von ausgewählten Klassen, Methoden oder Konfigurationsdateien. Abschließend zeigt der Anhang C zusätzliche Abbildungen der Softwarelösungen, die im Rahmen der vorliegenden Arbeit konzipiert und entwickelt wurden.

[8] http://www.cebitec.uni-bielefeld.de/index.php/graduate-programs/bioinformatics-of-signaling-networks

[9] http://www.techfak.uni-bielefeld.de/ags/bi/

2 Grundlagen

Das Kapitel 2 thematisiert die notwendigen Grundlagen der Molekularbiologie und der Informatik. In erster Linie werden dabei grundlegende Kenntnisse über Prozesse, Strukturen und Methoden aus der Molekularbiologie und Informatik vermittelt. Allerdings kann keine ausführliche Einführung in alle Aspekte gewährleistet werden, weshalb auf geeignete Literatur verwiesen wird.

Im Abschnitt 2.1 erfolgt eine Einführung in die relevanten Vorgänge der Molekularbiologie. Danach werden in Abschnitt 2.2 die erforderlichen Konzepte und Definitionen der Informatik vorgestellt. Abschließend erfolgt in Abschnitt 2.3 eine Zusammenfassung.

2.1 Grundlagen der Molekularbiologie

Mit Hilfe von Forschungsprojekten wie dem Humangenomprojekt [Int01] und dem 1000-Genome-Projekt [The10] konnten Prozesse, Interaktionen und Regulationsmechanismen innerhalb eines Organismus genauer interpretiert werden. Anhand dieser neuen Erkenntnisse konnten revolutionäre Methoden für die Grundlagenforschung und neue Therapien für die Medizin entwickelt werden, wobei die personalisierte Medizin als populäres Beispiel dafür zu nennen ist [SD05]. Trotz dieser bemerkenswerten Fortschritte sind viele Prozesse und Wechselwirkungen in einzelnen Zellen oder Organismen noch gänzlich unbekannt. In den folgenden Abschnitten werden die grundlegenden Prozesse und Strukturen der zellulären Molekularbiologie thematisiert. Die Einführung in diese Thematik ist zum Verständnis dieser Arbeit erforderlich. Für eine detaillierte Einführung in diese Thematik siehe [Kni06, KCS07].

2.1.1 Genome und Gene

Das Genom von Prokaryoten, Eukaryoten sowie von Viren wird auch als Erbgut bezeichnet und repräsentiert das gesamte Spektrum aller vererbbaren Informationen einer Zelle oder eines Viruspartikels. Diese Informationen sind für die Ontogenese und die spezifische Ausprägung eines Organismus essentiell. Bei allen Organismen

dient die *Deoxyribonucleic acid* (DNA) als Informationsträger der genetischen Information [AMM44]. Allerdings gibt es Viren, bei denen die *Ribonucleic acid* (RNA) als Informationsträger fungiert. Die Chromosomen sind Strukturen im eukaryotischen Zellkern, die sich aus DNA und Proteinen zusammensetzen und als linearer DNA-Doppelstrang organisiert sind. Im Gegensatz dazu haben Prokaryoten keinen Zellkern und verfügen auch über keine linearen Chromosomen. Vielmehr ist die DNA meist zirkulär im Cytoplasma organisiert. Mit Hilfe der DNA-Sequenzierung wurden zahlreiche Genome von unterschiedlichen Organismen erfolgreich sequenziert. Die Tabelle 2.1 präsentiert die Genomgröße und die Anzahl der Gene von bereits sequenzierten Organismen. Eine aktuelle und vollständige Übersicht bisher sequenzierter Organismen präsentiert das Genome News Network[1] (GNN). Die Grundbausteine der DNA und der RNA sind Nukleotide. In der Tabelle 2.2 werden die relevanten Nukleinbasen dargestellt. Anhand der Tabelle 2.2 wird deutlich, dass die

Organismus	**Genomgröße in Megabasenpaaren**	**Anzahl der Gene**
Archaeoglobus fulgidis	2,17	2493
E. coli	4,64	4397
H. influenzae	1,83	1791
S. cerevisiae (Hefe)	12,1	6548
Zea mays (Mais)	2500	≈20000
Homo sapiens (Mensch)	3300	≈25000

Tabelle 2.1: Genomgröße und Anzahl der Gene ausgewählter Organismen nach [KCS07].

Nukleinbase	**Abkürzung**	**Vorkommen**
Adenin	A	DNA und RNA
Cytosin	C	DNA und RNA
Guanin	G	DNA und RNA
Thymin	T	DNA
Uracil	U	RNA

Tabelle 2.2: Übersicht und Vorkommen relevanter Nukleinbasen.

Base Thymin nur in der DNA lokalisiert ist. Im Gegensatz dazu ist die Base Uracil nur in der RNA vorhanden. Des Weiteren ist der strukturelle Aufbau von DNA und RNA unterschiedlich. Die Struktur der DNA ist als Doppelhelix organisiert, wobei

[1] http://www.genomenewsnetwork.org/

die Basenpaare der einzelnen Stränge komplementär sind [JF53]. Im Allgemeinen ist die RNA als Einzelstrang strukturiert, während die Anzahl von möglichen dreidimensionalen Strukturen deutlich größer ist als bei der DNA. Zudem gibt es noch unterschiedliche Arten der RNA, die für verschiedene Prozesse wichtige Funktionen übernehmen. In den folgenden Abschnitten werden einige dieser notwendigen RNA-Typen behandelt.

Die einzelnen Bereiche auf der DNA werden generell in kodierende und nichtkodierende Nukleotidsequenzen eingeteilt. Die nötigen Grundinformationen zur Herstellung eines Genprodukts während der Genexpression werden durch Gene bereitgestellt. Ein Gen repräsentiert einen Bereich auf der DNA und verfügt grundsätzlich über regulatorische Elemente und ein Segment, das als *open reading frame* (ORF) bezeichnet wird. Im Abschnitt 2.1.2 wird der Prozess der Genexpression genauer vorgestellt.

Es gibt bei der Strukturierung der Gene zwischen Prokaryoten und Eukaryoten signifikante Unterschiede. Die Gene der Eukaryoten bestehen zumeist aus Exons (kodierende Nukleotidsequenzen) und Introns (nicht-kodierende Nukleotidsequenzen) [Gil78], welche die Exons separieren. Daher ist der ORF bei Eukaryoten nicht zusammenhängend. Während der Genexpression werden die Introns entfernt, was als Spleißen bezeichnet wird. Die Regulation der Genexpression erfolgt durch Promotoren, Silencer und Enhancer, die eine charakteristische Basenabfolge auf der DNA repräsentieren. In der Abbildung 2.1 wird die schematische Struktur eines Gens bei Eukaryoten dargestellt.

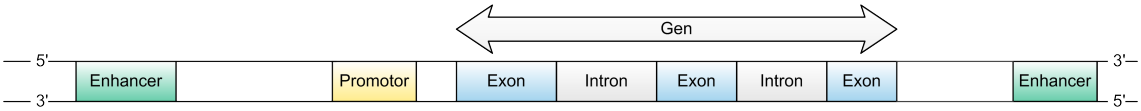


Abbildung 2.1: Schematische Struktur von einem Gen bei Eukaryoten.

Die Gene der Prokaryoten verfügen über keine Introns und beinhalten nur Exons. Dadurch ist es prinzipiell möglich, dass mehrere ORF unmittelbar aneinander grenzen und von einem regulatorischen Element reguliert werden. Diese zusammenhängenden Segmente werden als Operon bezeichnet. Dafür ist das Lactose-Operon (lac-Operon) ein populäres Beispiel, das in Abbildung 2.2 dargestellt ist und dessen schematische Struktur im Folgenden erläutert wird.

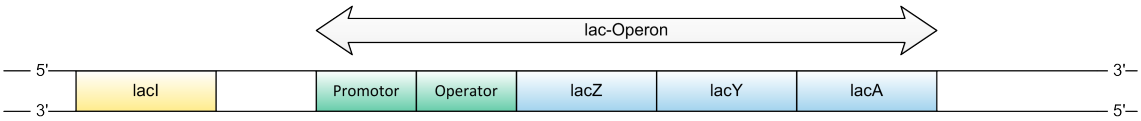


Abbildung 2.2: Schematische Struktur des lac-Operons nach [KCS07].

Das Gen lacI wird als Repressorgen bezeichnet, weil es ein Protein kodiert, das an

der Regulation beteiligt ist, einen so genannten Repressor. Sofern der Repressor mit dem Operator interagieren kann, wird die Transkription negativ beeinträchtigt. Als Strukturgene werden die Gene lacZ, lacY und lacA gekennzeichnet, weil diese Gene die Primärstruktur eines Enzyms kodieren. Der Promotor und der Operator repräsentieren die regulatorische Region des lac-Operons. Die regulatorische Einheit, das Repressorgen und die Strukturgene sind an der Steuerung des Lactosemetabolismus beteiligt. Die Regulation erfolgt bei Prokaryoten durch Promotor und Operator. Darüber hinaus sind bei Prokaryoten die regulatorischen Elemente Silencer und Enhancer nicht vorhanden. Anhand der Abbildungen 2.1 und 2.2 wird die unterschiedliche Strukturierung der Gene von Eukaryoten und Prokaryoten deutlich.

2.1.2 Transkription und Translation

Die Produktion eines Genprodukts basierend auf einer genetischen Information erfolgt während der Genexpression. Bis auf geringe Unterschiede ist dieser Prozess bei allen Organismen vergleichbar. Die signifikanten Unterschiede zwischen Prokaryoten und Eukaryoten bei der Genexpression werden in [Kni06] detailliert erläutert. Die Genexpression kann generell in die Schritte der Transkription und der Translation eingeteilt werden. Allerdings gibt es Proteine, die während oder nach der Translation durch entsprechende Prozesse modifiziert werden. Im Abschnitt 2.1.3 werden diese Prozesse thematisiert.

Die genetische Information von einem Gen auf der DNA wird innerhalb der Transkription identifiziert und komplementär als messenger RNA (mRNA) transkribiert. Des Weiteren wird die ebenfalls notwendige ribosomale RNA (rRNA) und transfer-RNA (tRNA) synthetisiert. Diese unterschiedlichen RNA-Typen werden im weiteren Prozess der Genexpression benötigt. Mit Hilfe der jeweiligen DNA-abhängigen RNA-Polymerase erfolgt die Synthese dieser verschiedenen Arten von RNA. Die rRNA als auch die tRNA sind nicht-kodierende RNA und werden nicht in ein Protein translatiert. Durch den Promotor kann die entsprechende DNA-abhängige RNA-Polymerase an der jeweiligen Position eine Interaktion mit der DNA realisieren und die Transkription initiieren. Eine charakteristische Basenabfolge auf der DNA, die als Terminator bezeichnet wird, beendet die Transkription. Bei Eukaryoten ist erst eine Precursor mRNA (pre-mRNA) im Zellkern verfügbar, die noch Introns und Exons enthält. Durch das Spleißen werden die Introns entfernt und die fertige mRNA synthetisiert. Anschließend wird die mRNA in das Cytoplasma transportiert, sodass eine Wechselwirkung mit den Ribosomen möglich ist. Der Export der mRNA ist bei Prokaryoten nicht notwendig, weil die Transkription im Cytoplasma erfolgt.

Die Translation ist ein wichtiger Schritt bei der Genexpression und erfolgt im Cytoplasma an den Ribosomen. Mit Hilfe der Ribosomen wird dabei die Nukleotidsequenz der mRNA in die Aminosäuresequenz eines Proteins translatiert. Dieser Schritt wird durch das Ribosom katalysiert. Die rRNA ist ein Struktur- und Funktionselement

der Ribosomen und somit essentiell für die Genexpression. Auf der mRNA kodieren drei aufeinanderfolgende Nukleotide eine Aminosäure (AS). Eine solche Abfolge wird als Codon oder Basentriplett bezeichnet. In der Tabelle 2.3 wird der genetische Code dargestellt. Anhand der Tabelle 2.3 wird deutlich, dass ein Codon eine Abfolge von

Erste Nukleinbase	Zweite Nukleinbase: U		Zweite Nukleinbase: C		Zweite Nukleinbase: A		Zweite Nukleinbase: G		Dritte Nukleinbase
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys	C
	UUA	Leu	UCA	Ser	UAA	Stopcodon	UGA	Stopcodon	A
	UUG	Leu	UCG	Ser	UAG	Stopcodon	UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg	C
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	A
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	C
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg	G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly	C
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly	G

Tabelle 2.3: Der genetische Code nach [Gra10].

drei Nukleotiden ist und dass der genetische Code auf vier Nukleotiden basiert. Daraus ergibt sich, dass es 64 mögliche Kombinationen für ein Codon gibt. Die Codons, die eine AS kodieren, können mit Hilfe der Tabellen 2.3 und 2.4 identifiziert werden. Es gibt für jedes Condon eine tRNA, die mit der entsprechenden AS beladen ist und über ein komplementäres Anticodon verfügt. Über die Codon-Anticodon-Paarung wird die AS am Ribosom entsprechend der mRNA-Sequenz aneinander gereiht und über Peptidbindungen zur Proteinprimärstruktur verknüpft. Das Codon AUG auf der mRNA kodiert die AS Methionin und wird auch als Startcodon bezeichnet. Der Abbruch der Translation wird durch drei Codons ermöglicht, für die keine komplementäre tRNAs bereitgestellt werden. Diese Codons auf der mRNA werden auch als Stopcodons bezeichnet. Die Verknüpfung von AS über Peptidbindungen produziert ein neues Protein, das anschließend durch den Prozess der Proteinfaltung eine komplexe Proteinstruktur erhält. Die Thematik der Prozesse und Strukturen der Proteine wird in Abschnitt 2.1.3 diskutiert.

Die DNA-abhängigen RNA-Polymerasen sind essentielle Enzyme bei dem Schritt der Transkription innerhalb der Genexpression. Allerdings sind zusätzliche Proteine notwendig um eine effiziente Transkription zu gewährleisten. Solche Proteine werden als TF bezeichnet und sind grundsätzlich bei der Initiierung und Regulation der

Transkription involviert. Die TF können mit der DNA interagieren und dabei als Aktivator oder Repressor für Promotoren dienen. Allerdings gibt es auch TF, die mit Hilfe zusätzlicher Proteine eine Interaktion mit der DNA realisieren. Ein TF wird in der Regel einer der folgenden zwei Arten zugeordnet:

1. **Allgemeine TF** identifizieren charakteristische Sequenzmotive auf der DNA wie etwa den Promotor und ermöglichen dann die Interaktion der jeweiligen DNA-abhängigen RNA-Polymerase. Dabei wird ein Proteinkomplex assembliert, der für die Initiierung der Transkription benötigt wird. Zudem kann dieser Proteinkomplex mit weiteren Proteinen interagieren. Diese Art der TF sind normalerweise nicht an der spezifischen Regulation der Gene beteiligt.

2. **Spezifische TF** regulieren unterschiedliche Gene und sind für deren Transkription verantwortlich. Dabei identifizieren diese TF ebenfalls charakteristische Sequenzmotive auf der DNA wie Enhancer oder Silencer und können dadurch als Aktivator oder Repressor dienen. Erst ein Proteinkomplex aus allgemeinen und spezifischen TF mit der DNA-abhängigen RNA-Polymerase ermöglicht letztendlich die Transkription des entsprechenden Gens.

Eine strukturierte Übersicht über experimentell verifizierte TF in den Organismen *Homo sapiens*, *Mus musculus* und *Rattus norvegicus* wird durch die *Transcription Factor Encyclopedia* [YBS+12] bereitgestellt. Die relevanten Informationen aus der Literatur werden durch Wissenschaftler zusammengefasst und in die Enzyklopädie integriert.

2.1.3 Prozesse und Struktur der Proteine

Der Prozess der Proteinbiosynthese ist für die Produktion biologisch aktiver Proteine in einem Organismus verantwortlich. Normalerweise setzen sich Proteine aus 100 - 800 AS zusammen. Sofern ein Protein über weniger als 100 AS verfügt, wird es als Polypeptid bezeichnet. Ein Enzym ist ebenfalls ein Protein und katalysiert essentielle biochemische Reaktionen. An der Signaltransduktion sind Proteine, als Enzyme in den unterschiedlichen Stoffwechselwegen und als Strukturprotein beim Aufbau der Struktur einer Zelle, essentiell beteiligt. Es gibt 20 natürliche AS, die in der Tabelle 2.4 dargestellt werden. Diese 20 natürlichen AS können durch die jeweilige Größe, Ladung und funktionelle Gruppe unterschieden werden. Als 21. und 22. AS werden die proteinogenen AS Selenocystein und Pyrrolysin bezeichnet, die aber den nichtkanonischen AS zugeordnet werden. Die 20 natürlichen AS und die AS Selenocystein werden im menschlichen Organismus benutzt, aber nur 12 der natürlichen AS werden durch den Organismus selbständig synthetisiert.

Der Prozess der Proteinfaltung ist verantwortlich für die dreidimensionale Struktur, die eine Voraussetzung für die korrekte biologische Funktion eines Proteins ist.

AS	Dreibuchstabencode	Einbuchstabencode
Alanin	Ala	A
Arginin	Arg	R
Asparagin	Asn	N
Asparaginsäure	Asp	D
Cystein	Cys	C
Glutamin	Gln	Q
Glutaminsäure	Glu	E
Glycin	Gly	G
Histidin	His	H
Isoleucin	Ile	I
Leucin	Leu	L
Lysin	Lys	K
Methionin	Met	M
Phenylalanin	Phe	F
Prolin	Pro	P
Serin	Ser	S
Threonin	Thr	T
Tryptophan	Trp	W
Tyrosin	Tyr	Y
Valin	Val	V

Tabelle 2.4: Die 20 natürlichen AS nach [LP98].

Grundsätzlich ist die Abgrenzung der Proteinstruktur in vier unterschiedliche hierarchische Strukturen möglich [LL52]. Die folgenden vier Strukturen eines Proteins werden dabei unterschieden:

1. Als **Primärstruktur** wird die lineare Anordnung von AS bezeichnet.
2. Die **Sekundärstruktur** kennzeichnet die konformationelle Anordnung der Polypeptidkette. Insbesondere α-Helices und β-Faltblätter sind dafür charakteristische und häufig auftretende Motive.
3. Die **Tertiärstruktur** ist unter Einbeziehung der Wechselwirkungen innerhalb des Moleküls die vollständige räumliche Anordnung der Sekundärstruktur.
4. Als **Quartärstruktur** wird die Anordnung zweier oder mehrerer Polypeptide mit Tertiärstruktur bezeichnet.

Als Konformationsänderung wird die Veränderung der räumlichen Proteinstruktur bezeichnet und ist unter anderem bei einem Enzym notwendig, um seine Funktionsfähigkeit zu gewährleisten. In der Regel erfolgt die Konformationsänderung durch die Interaktion mit einem Liganden, aber auch Chaperone sowie die Veränderung der

Temperatur oder des pH-Werts können diesen Mechanismus aktivieren. Allerdings ist es möglich, dass die Proteinfaltung nicht erfolgreich durchgeführt wird, sodass das Protein über keine korrekte biologische Funktion verfügt. Deswegen unterliegen Proteine einer Proteinqualitätskontrolle. Mit Hilfe dieser Kontrolle werden die fehlgefalteten Proteine identifiziert und anschließend proteolytisch abgebaut. Wenn dieser Abbau nicht erfolgt, können sogenannte Proteinfehlfaltungserkrankungen wie Alzheimer, Parkinson oder Chorea Huntington die Folge sein. Obwohl die Bioinformatik unterschiedliche Konzepte und Softwarelösungen zur Proteinfaltung zur Verfügung stellt, ist eine zuverlässige Vorhersage der Sekundärstruktur und der Tertiärstruktur basierend auf der Primärstruktur nur begrenzt möglich. Es gibt zahlreiche Forschungsprojekte wie Folding@home[2] , POEM@home[3] und Rosetta@home [DQR+07], die sich mit der Simulation, Vorhersage und Optimierung der Struktur und Faltung von Proteinen beschäftigen.

Die speziellen Modifikationen bei einigen Proteinen durch spezifische Enzyme können co- oder posttranslational erfolgen. Die posttranslationale Proteinmodifikation (PTM) erfolgt erst, wenn die Translation und die Proteinfaltung durchgeführt wurden. Im Gegensatz dazu findet die cotranslationale Modifikation eines Proteins während der Translation statt. Ein populäres Beispiel für eine PTM ist die O-Glykosylierung eines Glykoproteins, die in den Zisternen des Golgi-Apparates stattfindet [LP98]. Außerdem sind die Sulfatierung und die Phosphorylierung wichtige und charakteristische Beispiele für die PTM. Die N-Glykosylierung ist ein charakteristisches Beispiel für eine cotranslationale Modifikation eines Glykoproteins und erfolgt im Lumen des Endoplasmatischen Retikulum (ER) [LP98]. In [LP98] werden die N- und O-Glykosylierung sowie weitere Beispiele für die post- und cotranslationale Modifikationen eines Proteins ausführlich behandelt.

2.2 Grundlagen der Informatik

Die Informatik sowie deren Methoden und Definitionen sind inzwischen ein zentraler Bestandteil in der Gesellschaft und in der Wissenschaft. Insbesondere die aktuellen Fragestellungen in den Lebenswissenschaften benötigen die Informatik. Mit Hilfe der Informatik ist die erfolgreiche Analyse und Bewerkstelligung dieser Fragestellungen möglich. Vor allem die interdisziplinären Wissenschaften wie die Bioinformatik, Chemoinformatik oder Medizinische Informatik behandeln gegenwärtig verschiedene Probleme aus den Lebenswissenschaften.

Eine wichtige Thematik der Bioinformatik ist die Integration, Analyse und Vorhersage molekularbiologischer Daten. Die dafür notwendigen Grundlagen und Technologien aus der Informatik werden in diesem Abschnitt 2.2 thematisiert. Allerdings erfolgt keine detaillierte Einführung in diese Aspekte, weshalb auf entsprechende

[2] http://folding.stanford.edu/
[3] http://boinc.fzk.de/poem

Literatur verwiesen wird. Insbesondere zum Verständnis der Kapitel 4 und 5 ist eine Einführung erforderlich.

2.2.1 Algorithmen und Datenstrukturen

In der Informatik sind Algorithmen und Datenstrukturen ein grundlegender Aspekt und ermöglichen in Kombination, zeitintensive und komplexe Probleme zu bewerkstelligen. Mit Hilfe von Algorithmen und Datenstrukturen erfolgt auch in der Bioinformatik die Integration, Analyse und Vorhersage molekularbiologischer Daten aus den Lebenswissenschaften. Die Generierung solcher Daten kann durch *Next Generation Sequencing* (NGS), *Microarrays* oder Massenspektometrie erfolgen. Bei der Analyse dieser Daten und der Vorhersage neuer molekularbiologischer Daten können Algorithmen aus der Bioinformatik wie der Needleman-Wunsch-Algorithmus, der Smith-Waterman-Algorithmus oder der Neighbor-Joining-Algorithmus verwendet werden [MW03].

Eine Datenstruktur ist ein mathematisches Objekt, das zur Speicherung von Daten eingesetzt wird. Des Weiteren realisiert die Datenstruktur den Zugriff und die Verwaltung der Daten durch entsprechende Operationen. Insbesondere bei der Realisierung effizienter Algorithmen ist die Verwendung geeigneter und effektiver Datenstrukturen wichtig. Aufgrund dieser umfangreichen Thematik werden ausgewählte und für diese Arbeit notwendige Datenstrukturen im Folgenden thematisiert. Eine ausführliche Einführung in den Themenbereich der Algorithmen und Datenstrukturen erfolgt in [SS10]. Es gibt einige grundlegende Arten von Datenstrukturen in der Informatik, die sich bei den jeweiligen Operationen im Speicher- und Rechenaufwand unterscheiden können. Im Folgenden werden vier wichtige Datenstrukturen aus der Informatik präsentiert:

1. Der Datentyp und die Größe von einem **Array** wird zur Laufzeit definiert. Die Definition dieser statischen Datenstruktur kann entweder eindimensional oder mehrdimensional erfolgen. Das mathematische Gegenstück sind somit die Konstrukte Vektor und Matrix.

2. Die **verkettete Liste** kann eine zur Laufzeit unbekannte Anzahl von Elementen verwalten. Diese dynamische Datenstruktur erweitert oder reduziert die Kapazität automatisch zur Laufzeit. Außerdem ist jedes Element mit dem nächsten Element verknüpft.

3. Ein **Graph** ist ebenfalls eine Datenstruktur und verfügt über Knoten und Kanten. Mit Hilfe der Kanten können Beziehungen zwischen Knoten realisiert werden. Es gibt verschiedene Spezialisierungen von Graphen, die sich im Typ der Kante und in der Realisierung der Knoten und Kanten durch eine Datenstruktur unterscheiden. Die Repräsentation erfolgt in der Regel als Inzidenzmatrix, Adjazenzmatrix oder Adjazenzliste.

4. Die Darstellung hierarchischer Strukturen ist durch die Datenstruktur **Baum** möglich. In der Graphentheorie ist diese Datenstruktur ein spezifischer Graph und verfügt somit ebenfalls über Knoten und Kanten. Allerdings werden zur Realisierung hierarchischer Strukturen spezielle Eigenschaften der Knoten benötigt. In der Regel erfolgt die exakte Charakterisierung der Knoten durch die Schlüsselwörter Wurzel, innerer Knoten und Blatt. Insbesondere bei Indexstrukturen von DB sind Spezialisierungen dieser Datenstruktur relevant.

In der Regel gibt es von den grundlegenden Datenstrukturen noch weitere Spezialisierungen, die für spezielle Problemstellungen konstruiert wurden. Der folgende Abschnitt erläutert zwei dieser Spezialisierungen präziser.

2.2.1.1 Suffix-Bäume und Suffix-Arrays

Die effiziente Analyse von DNA-, RNA- oder Aminosäuresequenzen durch entsprechende Algorithmen ist ein Schwerpunkt der Bioinformatik. In der Regel werden diese Sequenzen als Zeichenketten interpretiert. Eine effektive Verarbeitung von Zeichenketten durch einen Algorithmus benötigt eine spezielle Datenstruktur. Solche Datenstrukturen sind Suffix-Bäume und -Arrays, die im Folgenden detaillierter thematisiert werden. Außerdem wird das Enhanced Suffix-Array (ESA) behandelt, das als Erweiterung der Suffix-Arrays gilt und die Vorteile der Suffix-Bäume und Suffix-Arrays kombiniert. Mit Hilfe dieser Datenstrukturen kann eine effiziente Suche nach Sequenzmotiven realisiert werden.

Ein Suffix-Baum ist ein gewurzelter Baum und verfügt über alle Suffixe einer Zeichenkette. Normalerweise ist die Konstruktion von solchen Suffix-Bäumen bei einer linearen Zeit- und Platzkomplexität möglich. Die einzelnen Algorithmen von Weiner, McCreight und Ukkonen benötigen eine Zeitkomplexität von $\mathcal{O}(n)$ zur Realisierung eines Suffix-Baums. In [GK97] werden die Unterschiede und Gemeinsamkeiten von diesen drei Algorithmen diskutiert. Außerdem gibt es den Write-Only-Top-Down-Algorithmus (WOTD-Algorithmus), dessen Zeitkomplexität bei der Suffix-Baum Konstruktion im *worst-case* $\mathcal{O}(n^2)$ und im *average-case* $\mathcal{O}(n\ log\ n)$ ist [GKS03]. In der Abbildung 2.3 wird ein Suffix-Baum für die Nukleotidsequenz ACAACAC dargestellt. Sobald diese Datenstruktur für eine Zeichenkette konstruiert wurde, ist die Suchzeit unabhängig von der Größe der Zeichenkette. Dadurch ist es möglich, ein Sequenzmotiv innerhalb dieser Zeichenkette in $\mathcal{O}(n)$ zu identifizieren. Allerdings ist der Speicherverbrauch der Suffix-Bäume in der Praxis zu intensiv, weshalb die Suffix-Arrays entwickelt wurden.

Die Suffix-Arrays sind eine Spezialisierung der Datenstruktur Array und verfügen über die einzelnen Suffixe einer Zeichenkette in lexikographischer Ordnung. Insbesondere der effiziente Speicherverbrauch dieser Datenstruktur ist ein wesentliches Merkmal im Vergleich mit dem Suffix-Baum. Prinzipiell gibt es zwei Möglichkeiten zur Konstruktion eines Suffix-Arrays für eine Zeichenkette. Der erste Ansatz kon-

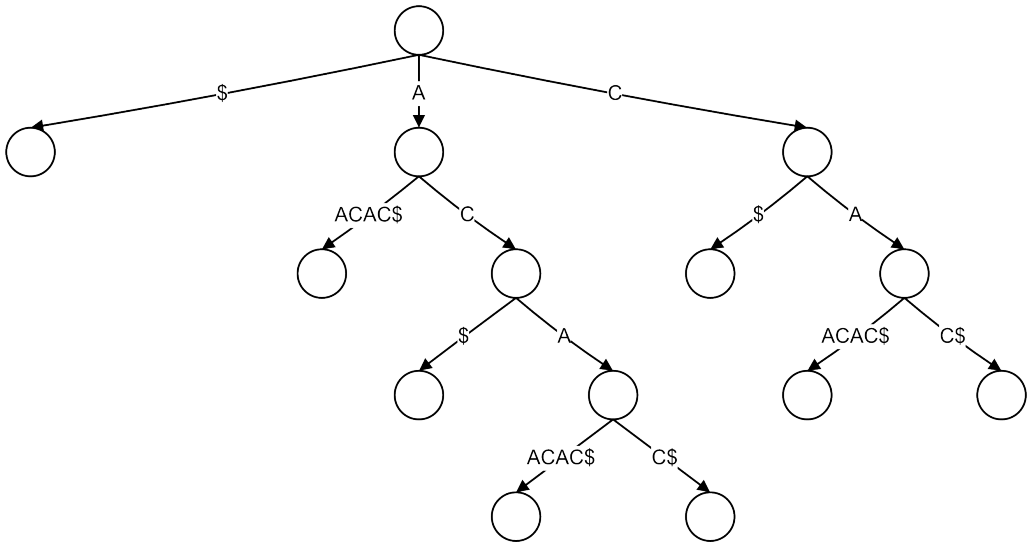


Abbildung 2.3: Darstellung eines Suffix-Baums für die Nukleotidsequenz ACAACAC. Die Konstruktion dieses Suffix-Baums erfolgte durch den WOTD-Algorithmus.

struiert für die Zeichenkette einen Suffix-Baum und anschließend wird mit Hilfe einer Tiefensuche das Suffix-Array realisiert. Die direkte Konstruktion durch Algorithmen wird als zweiter Ansatz bezeichnet und ist die bevorzugte Variante. Der Algorithmus von Ko und Aluru [KA03] und der Skew-Algorithmus [KS03] ermöglichen bei einer Zeit- und Platzkomplexität von $\mathcal{O}(n)$ die Konstruktion eines Suffix-Arrays. Sofern für eine Zeichenkette ein Suffix-Array existiert, kann mit Hilfe der binären Suche ein Sequenzmotiv innerhalb des Suffix-Arrays identifiziert werden. Allerdings benötigt diese Suche eine Zeitkomplexität von $\mathcal{O}(m\ log(n))$, weshalb eine Optimierung sinnvoll ist. Durch ein zusätzliches Datenfeld, das die Werte der längsten gemeinsamen Präfixe (LGP) für die jeweiligen Position repräsentiert, ist eine binäre Suche in der Zeit $\mathcal{O}(m + log(n))$ möglich. Außerdem kann jetzt durch das Suffix-Array und die LGP-Werte eine Baumstruktur realisiert werden, die als LGP-Intervall-Baum bezeichnet wird. Ein Aspekt der ESA ist die Erweiterung der Suffix-Arrays um die LGP-Werte. Das Ziel der ESA ist die Kombination der Vorteile von Suffix-Bäumen und Suffix-Arrays zu einer effizienten Datenstruktur, wodurch eine Transformation der Algorithmen von Suffix-Bäumen auf Suffix-Arrays möglich ist. Dafür wird jedoch ein weiteres Datenfeld benötigt, das als Child-Tabelle bezeichnet wird. Die Child-Tabelle verfügt über eine Verknüpfung zum nächsten Suffix aus einem anderen Teilbaum der Baumstruktur. Somit wird ein ESA durch das Suffix-Array, die LGP-Tabelle und der Child-Tabelle realisiert. Dadurch kann ein Sequenzmotiv innerhalb einer Zeichenkette in der gleichen Zeitkomplexität wie bei einem Suffix-Baum identifiziert werden. Zudem ist der Speicherverbrauch in der Praxis bei komplexen Anwendungen, die etwa vollständige Genome vergleichen, minimal [AKO04]. Die abstrakte Darstellung eines ESA für die Nukleotidsequenz ACAACAC wird in der Tabelle 2.5 dargestellt.

i	Suffix	suf[i]	lgp[i]	child[i]
0	AACAC$	2	0	8
1	ACAACAC$	0	1	4
2	ACAC$	3	3	3
3	AC$	5	2	4
4	CAACAC$	1	0	8
5	CAC$	4	2	6
6	C$	6	1	7
7	$	7	0	8

Tabelle 2.5: Abstrakte Darstellung eines ESA für die Nukleotidsequenz ACAACAC. Der Index im Array wird durch i repräsentiert. Mit Hilfe von suf[i] kann die Position des Suffix in der Nukleotidsequenz lokalisiert werden. Durch lgp[i] ist es möglich, das LGP von i und i-1 zu identifizieren. Die Vernüpfung zum nächsten Suffix, das nicht im selben Teilbaum wie i und i-1 ist, wird durch child[i] dargestellt.

2.2.2 Informationssysteme und Datenbanksysteme

Ein IS verfügt über verschiedene Funktionalitäten, die während der Anforderungsanalyse identifiziert werden. Prinzipiell werden aber die grundlegenden Operationen *Create*, *Read*, *Update* und *Delete* (CRUD) durch ein IS bereitgestellt. Aufgrund der unterschiedlichen Anforderungen an ein IS gibt es verschiedene Arten, die innerhalb der Wirtschaft, Forschung und Wissenschaft sowie staatlicher Institutionen wichtige Geschäftsprozesse realisieren. In der Forschung und Wissenschaft sind Krankenhausinformationssysteme (KIS) und Labor-Informations- und Management-Systeme (LIMS) populäre Beispiele dieser Systeme. Die wesentliche Grundlage eines IS ist das DBS, wodurch die effiziente Speicherung, Verwaltung und Bereitstellung von Daten möglich ist. Ein DBS verfügt über zwei wichtige Bestandteile, einem Datenbankmanagementsystem (DBMS) und einer oder mehrerer DB. Als DB wird der persistente und strukturierte Datenbestand bezeichnet. Das DBMS ist eine Software zur Verwaltung und Organisation dieses Datenbestands. Insbesondere die Integritätsbedingungen der Atomarität, Konsistenz, Isoliertheit und Dauerhaftigkeit (AKID) sind durch das DBMS zu gewährleisten. Die Strukturierung der Daten innerhalb der DB erfolgt durch ein Datenbankmodell. Dieses Datenbankmodell wird durch das DBMS zur Verfügung gestellt und ermöglicht auch deren Klassifikation. Ein Datenbankmodell verfügt über Konzepte zur Realisierung der Syntax und der Semantik eines Datenbankschemas. Prinzipiell basieren DBMS auf unterschiedlichen Datenbankmodellen. Im Folgenden werden vier verschiedene Arten von Datenbankmodellen dargestellt:

1. Hierarchisches Datenbankmodell
2. Netzwerkdatenbankmodell
3. Objektorientiertes Datenbankmodell

4. Relationales Datenbankmodell

Des Weiteren gibt es das objektrelationale Datenbankmodell, wobei ein relationales Datenbankmodell um Konzepte aus der Objektorientierung erweitert wird [TS06]. Das relationale Datenbankmodell, das auf der Grundlage der Codd'schen Regeln [Cod90] basiert, hat sich als Standard etabliert. Insbesondere die Flexibilität, die Verfügbarkeit einer standardisierten Abfragesprache und die einfache Modellierung komplexer Sachverhalte sind signifikante Vorteile. Die relationalen Datenbankmanagementsysteme (RDBMS) DB2[4], MySQL[5] und Oracle[6] sind populäre Beispiele für diesen Standard.

In der Abbildung 2.4 wird eine Architektur eines IS dargestellt. Die zentrale Kom-

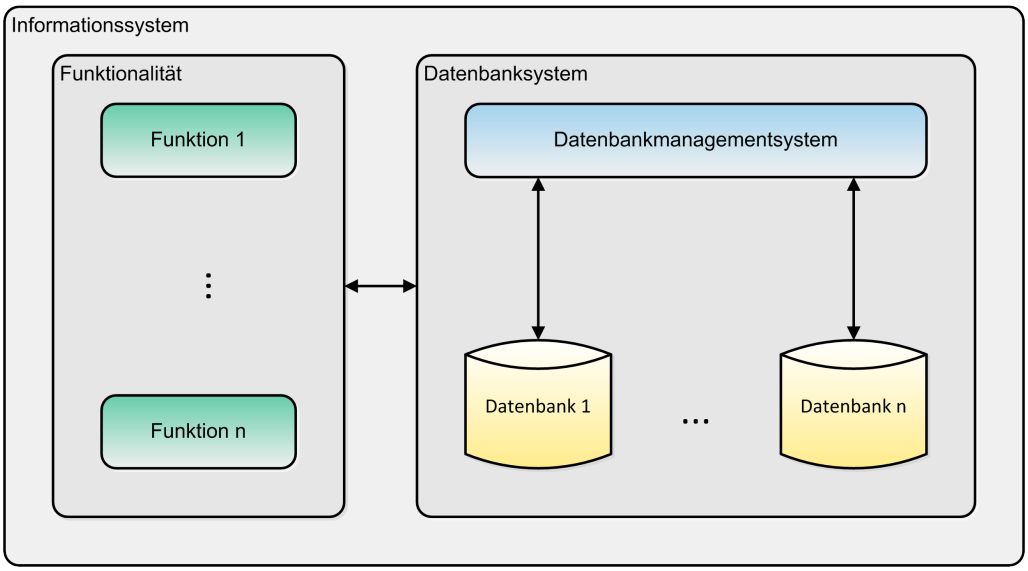


Abbildung 2.4: Architektur eines IS.

ponente bei dem IS in der Abbildung 2.4 ist das DBS, das die strukturierte, effiziente und widerspruchsfreie Speicherung, Manipulation und Verwaltung der Daten gewährleistet.

Durch das Internet sind inzwischen komplexe und interaktive Dienste und Anwendungen verfügbar, die im Kontext des *Web 2.0* realisiert wurden. Insbesondere das exponentielle Wachstum an Daten und deren Abhängigkeit, Komplexität und Vernetzung sind neue Herausforderungen für verteilte DBMS. Diese Problematik wird durch das CAP-Theorem (*Consistency*, *Availability* und *Partition Tolerance*) [GL02] repräsentiert. Infolgedessen ist eine neue Generation von DBMS entstanden, die durch den Begriff NoSQL repräsentiert werden. Prinzipiell verfügen derartige DBMS nach [EFH+11] über folgende Hauptmerkmale:

[4]http://www-01.ibm.com/software/data/db2/
[5]http://www.mysql.de/
[6]http://www.oracle.com/us/products/database/index.html

- Das zugrundeliegende Datenbankmodell ist nicht relational.
- Die Systeme sind auf eine verteilte und horizontale Skalierbarkeit ausgerichtet.
- Außerdem liegt dem System meistens auch ein anderes Konsistenzmodell zugrunde, weshalb die Integritätsbedingungen nicht auf AKID basieren, sondern auf *Basically Available*, *Soft State*, *Eventually Consistent* (BASE).

Als Beispiele für solche DBMS können CouchDB[7], MongoDB[8] und Neo4j[9] bezeichnet werden. Ob sich diese DBMS gegenüber den etablierten RDBMS durchsetzen werden, bleibt abzuwarten. Für eine detaillierte Einführung in NoSQL siehe [EFH+11].

2.2.2.1 Objektrelationale Abbildung

Die Implementierung einer Software erfolgt häufig durch Programmiersprachen, die auf der Grundlage objektorientierter Konzepte basieren. Allerdings ist der etablierte Standard von DBMS das relationale Datenbankmodell. Zwischen diesen beiden Paradigmen gibt es grundlegende Unterschiede. Dieser Konflikt wird auch als *Object-relational impedance mismatch* bezeichnet und kann durch verschiedene Lösungsansätze beseitigt werden. Die folgenden vier Lösungsansätze können diese Problematik beseitigen:

1. Objektrelationale Datenbankmanagementsysteme (ORDBMS)
2. Objektdatenbankmanagementsysteme (ODBMS)
3. Erweiterung der Programmiersprache um relationale Konzepte
4. Objektrelationale Abbildung

Aufgrund von unterschiedlichen Nachteilen in der Praxis haben sich die ersten drei Ansätze nicht durchgesetzt. Daher wird besonders die objektrelationale Abbildung zur Beseitigung dieser Problematik verwendet. Durch diese Variante ist es nicht notwendig, die jeweilige Programmiersprache um relationale Konzepte zu erweitern. Darüber hinaus kann die etablierte und ausgereifte Technik der RDBMS weiterhin eingesetzt werden. Die objektrelationale Abbildung ermöglicht die direkte Darstellung der Datenbanktabellen und deren Beziehungen untereinander als Objekt sowie eine direkte Speicherung der Objekte und deren Abhängigkeiten ohne Transformation. Außerdem können verschiedene DBMS eingesetzt werden, weil zwischen der

[7] http://couchdb.apache.org/
[8] http://www.mongodb.org/
[9] http://neo4j.org/

Anwendungsschicht und der Datenbankschicht eine zusätzliche Persistenzschicht integriert wird. Diese Funktionalitäten werden in der Regel durch ein *Framework* wie Hibernate[10] oder Apache Cayenne[11] bereitgestellt.

2.2.2.2 Klassifizierung molekularbiologischer Datenbanken

In den Lebenswissenschaften ist die Verwendung eines IS in Kombination mit einem DBS inzwischen ein wichtiger Faktor. Diese Systeme verfügen über effiziente und spezielle Funktionalitäten, sodass eine Bewerkstelligung der komplexen Fragestellungen der Lebenswissenschaften möglich ist. Allerdings sind die Aspekte der Benutzerfreundlichkeit, Aktualität und Flexibilität bei einigen IS kritisch zu bewerten, weswegen eine Reihe von diesen Systemen in der Praxis nicht eingesetzt werden. Außerdem sind durch Forschungsprojekte oder andere Initiativen inzwischen zahlreiche molekularbiologische DB und IS verfügbar.

Insbesondere die molekularbiologischen DB sind für die Realisierung neuer Softwarelösungen notwendig, weil diese Datenquellen unterschiedliche molekularbiologische Daten bereitstellen. Derzeit listet NAR etwa 1552 molekularbiologische DB, die in verschiedene Kategorien untergliedert werden können [FSRG14]. Im Folgenden werden die 15 Kategorien dargestellt, die eine abstrakte Klassifikation dieser DB ermöglichen:

1. *Nucleotide Sequence Databases*
2. *RNA sequence databases*
3. *Protein sequence databases*
4. *Structure Databases*
5. *Genomics Databases (non-vertebrate)*
6. *Metabolic and Signaling Pathways*
7. *Human and other Vertebrate Genomes*
8. *Human Genes and Diseases*
9. *Microarray Data and other Gene Expression Databases*
10. *Proteomics Resources*
11. *Other Molecular Biology Databases*
12. *Organelle databases*

[10] http://www.hibernate.org/
[11] http://cayenne.apache.org/

13. *Plant databases*

14. *Immunological databases*

15. *Cell biology*

In [FSRG14] wird eine ausführliche Klassifikation der molekularbiologischen DB dargestellt. Eine weitere Zusammenstellung molekularbiologischer DB wird durch das öffentliche Wiki *MetaBase* [BCP+11] zur Verfügung gestellt, wobei die Pflege und Erweiterung der Einträge von *MetaBase* durch die wissenschaftliche Gemeinschaft erfolgt. Darüber hinaus präsentiert das *Bioinformatics Links Directory* [BYYO11] eine aktuelle und strukturierte Übersicht von Softwarelösungen und Datenquellen aus der Bioinformatik.

Die molekularbiologischen DB verfügen über signifikante Unterschiede in der Verteilung, Autonomie und Heterogenität. Deswegen ist eine unkomplizierte Integration und Verwendung in externen Anwendungen nicht immer möglich. Mit Hilfe von Konzepten aus der Datenintegration kann diese Problematik beseitigt werden. Im Abschnitt 2.2.3 wird diese Thematik detaillierter behandelt.

2.2.3 Anforderungen und Methoden der Datenintegration

Dieser Abschnitt erläutert die Probleme der Datenintegration und die dafür verfügbaren Lösungsansätze. Die Integration von Datenbeständen aus Datenquellen mit unterschiedlichen Heterogenitäten ist nicht nur eine Herausforderung in der Wirtschaft, sondern auch in der Forschung und Wissenschaft. Insbesondere in den Lebenwissenschaften werden durch Experimente zahlreiche molekularbiologische Daten generiert, die über verschiedene Heterogenitäten verfügen. Die Speicherung, Bereitstellung und Verwaltung dieser Daten erfolgt in der Regel durch molekularbiologische DB. Normalerweise sind diese DB frei verfügbar, weltweit verteilt und durch explizite Querverweise miteinander verknüpft. Zudem gibt es signifikante Unterschiede in der Strukturierung der Daten, den Zugriffmöglichkeiten und dem Urheberrecht. Ein Aspekt der Bioinformatik ist die Implementierung von Anwendungen, mit deren Hilfe eine effektive Datenintegration von molekularbiologischen DB ermöglicht wird. Die Fragen der Datenintegration im Kontext molekularbiologischer Daten, die dafür verfügbaren Lösungansätze und entsprechende Anwendungen werden ausführlich in [KHH11] erörtert. Der Abschnitt 3.1 thematisiert ausgewählte Softwarelösungen, die auf der Data-Warehouse-Technik basieren.

Das Ziel der Datenintegration ist, eine Datenbasis zu realisieren, die über eine einheitliche Datenstruktur verfügt und alle notwendigen Daten aus den Datenquellen zur Verfügung stellt. Die Datenquellen verfügen in der Regel über verschiedene Schemata, weshalb als erstes eine Schematransformation und -integration notwendig ist. Danach erfolgt die eigentliche Integration der Datenbestände aus den jeweiligen

Datenquellen. Die Datenbestände werden dabei analysiert und validiert, sodass Inkonsistenzen und Duplikate identifiziert und beseitigt werden. Während dieser Datenbereinigung erfolgt ebenfalls die Zusammenführung und Vervollständigung von unvollständigen Datensätzen. In folge dieser Datenfusion wird ein vollständiger Datensatz realisiert, der mehr Informationen zur Verfügung stellt als die ursprünglichen Datensätze aus den Datenquellen. Die daraus resultierende konsistente und strukturierte Datenbasis ermöglicht eine effiziente und globale Sicht auf alle Datenbestände aus den Datenquellen. Allerdings ist die Zusammenführung von Datenbeständen aus unterschiedlichen Datenquellen mit drei Grundproblemen verknüpft, die in Abschnitt 2.2.3.1 thematisiert werden. In Abschnitt 2.2.3.2 wird die Thematik der Architekturen in der Datenintegration erläutert.

2.2.3.1 Verteilung, Autonomie und Heterogenität

Mit Hilfe von spezifischen Softwarelösungen wird die Integration von Datenbeständen aus Datenquellen realisiert. Solche Systeme verfügen normalerweise über verschiedene Integrationsarchitekturen, wodurch die drei Grundprobleme der Datenintegration erfolgreich beseitigt werden. Die Verteilung, Autonomie und Heterogenität einer Datenquelle repräsentieren diese Grundprobleme und werden auch als orthogonale Dimension der Datenintegration beschrieben [LN07]. In der Praxis verfügen die unterschiedlichen Projekte immer über alle drei Probleme. Daraus folgt, dass die notwendigen Datenquellen für ein Projekt sich in der Verteilung und der Heterogenität unterscheiden. Zudem gibt es Unterschiede in der Autonomie der einzelnen Organisationen, die für die jeweiligen Datenquellen verantwortlich sind. Im Folgenden werden die Verteilung, Autonomie und Heterogenität und deren unterschiedliche Arten thematisiert. Eine ausführliche Diskussion dieser Fragestellung mit Beispielen erfolgt in [LN07].

Ein Problem, das bei der Datenintegration beseitigt werden muss, ist die weltweite Verteilung der Datenquellen. Normalerweise werden die Datenbestände durch verschiedene Systeme zur Verfügung gestellt und sind geografisch verteilt. Aufgrund dieser unterschiedlichen Lokalisierung der Daten wird zwischen der physischen und der logischen Verteilung unterschieden. Mit Hilfe einer materialisierten Integrationsarchitektur, die in Abschnitt 2.2.3.4 beschrieben wird, kann die Problematik der physischen Verteilung entfernt werden. Die Bereitstellung von Metadaten und Methoden der Datenbereinigung durch das Integrationssystem ermöglicht die Beseitigung der logischen Verteilung.

Die Autonomie einer Datenquelle ist meist nicht zu verhindern, weil die verantwortliche Organisation einer Datenquelle in der Regel eigene Strategien und Technologien bei der Entwicklung einsetzt. Der Begriff Autonomie bedeutet im Zusammenhang mit der Datenintegration, das die Datenquelle selbstständig über die Bereitstellung, die Zugriffsmöglichkeiten und das Urheberrecht der Datenbestände entscheiden kann. Außerdem ist die Autonomie verantwortlich für unterschiedliche Probleme

der Heterogenität. Es werden grundsätzlich die folgenden Arten bei der Autonomie unterschieden:

- Design-, Schnittstellen- und Zugriffsautonomie
- Juristische Autonomie

In [Con97] werden als weitere Arten der Autonomie die Kommunikations- und Ausführungsautonomie diskutiert. Durch die Verwendung von etablierten und branchenspezifischen Standards bei der Entwicklung kann die Autonomie zu einem gewissen Grad eingeschränkt werden. Dadurch können auch einige Probleme bei der Heterogenität beseitigt werden.

Das Hauptproblem, das bei der Datenintegration beseitigt werden muss, ist die Heterogenität. Sofern zwei IS nicht identische Methoden, Modelle und Strukturen zum Zugriff auf den Datenbestand zur Verfügung stellen, werden diese als heterogen bezeichnet [LN07]. Die folgenden Arten der Heterogenität werden bei der Datenintegration unterschieden:

- Syntaktische, Strukturelle, Schematische, Semantische und Technische Heterogenität
- Datenmodellheterogenität

In erster Linie ist die Autonomie für die Heterogenität verantwortlich, aber auch die Verteilung kann Heterogenität erzeugen. Die Einschränkung der Autonomie durch standardisierte Austauschformate, Datenmodelle und Schnittstellen kann in einigen Aspekten Homogenität ermöglichen.

2.2.3.2 Architekturen der Datenintegration

Es gibt verschiedene Integrationsarchitekturen, die bei der Realisierung einer Softwarelösung zur Datenintegration eingesetzt werden können. In der Regel wird zwischen virtuellen und materialisierten Integrationsarchitekturen unterschieden. Der wesentliche Unterschied zwischen den beiden Integrationsarchitekturen ist der Speicherort der relevanten Datenbestände bei der Integration. Eine materialisierte Integrationsarchitektur ist eine zentrale und persistente Datenbasis und kopiert alle notwendigen Datenbestände aus den Datenquellen in die Datenbasis. Im Gegensatz dazu verfügt eine virtuelle Integrationsarchitektur über keine solche Datenbasis und kopiert somit auch keine Datenbestände. Daraus folgt, dass der integrierte und homogene Datenbestand bei einer virtuellen Integrationsarchitektur nur virtuell existiert und bei sämtlichen Anfragen erneut realisiert werden muss. Allerdings gibt es auch hybride Architekturen, die über materialisierte und virtuelle Datenbestände verfügen. In der Tabelle 2.6 werden anhand von ausgewählten Kriterien die Vor- und Nachteile dieser

Kriterien	**Materialisiert**	**Virtuell**
Aktualität	niedrig	hoch
Antwortzeit	niedrig	hoch
Komplexität	niedrig	hoch
Autonomie	Bereitstellung von Load-Dateien	Beantwortung von Anfragen
Anfrage-möglichkeiten	unbeschränkt	beschränkt
Read / Write	beides möglich	nur lesend
Speicherbedarf	hoch	niedrig
Belastung der Quellen	hoch, aber planbar	eher niedrig, schlecht planbar
Datenreinigung	möglich	nicht möglich

Tabelle 2.6: Vor- und Nachteile materialisierter und virtueller Integration nach [LN07].

beiden Integrationsarchitekturen dargestellt. In Abschnitt 2.2.3.3 wird die Thematik der virtuellen Integrationsarchitekturen diskutiert. Abschließend wird in Abschnitt 2.2.3.4 eine populäre materialisierte Integrationsarchitektur präsentiert.

2.2.3.3 Virtuelle Integrationsarchitekturen

Insbesondere föderierte Datenbanksysteme (FDBS), Peer-Daten-Management Systeme (PDMS) und Mediatorbasierte Integrationssysteme & Wrapper (MBS) sind charakteristische Beispiele für virtuelle Integrationsarchitekturen. Der wesentliche Vorteil dieser Architektur ist die Aktualität der Daten, weil für sämtliche Anfragen die derzeit aktuellen Daten aus den Datenquellen übertragen werden. Als Nachteile solcher Architekturen werden normalerweise die schlechte Performance bei Anfragen, die aufwendige Datenbereinigung und die beschränkten Anfragemöglichkeiten bezeichnet. Anhand der Tabelle 2.6 können weitere Vor- und Nachteile identifiziert werden.

Die Referenzarchitektur eines FDBS wird in der Abbildung 2.5 dargestellt. Die FDBS sind ein klassisches Beispiel für eine virtuelle Integrationsarchitektur. Eine detaillierte Einführung in die Thematik der FDBS erfolgt in [Con97, LN07]. In dieser Arbeit ist die Grundlage eine materialisierte Integrationsarchitektur, weshalb die virtuellen Integrationsarchitekturen nicht ausführlich thematisiert werden.

2.2.3.4 Materialisierte Integrationsarchitekturen

Insbesondere DWH repräsentieren die materialisierte Integrationsarchitektur, die als standardisierte Architektur für diese Integrationsarchitektur bezeichnet werden

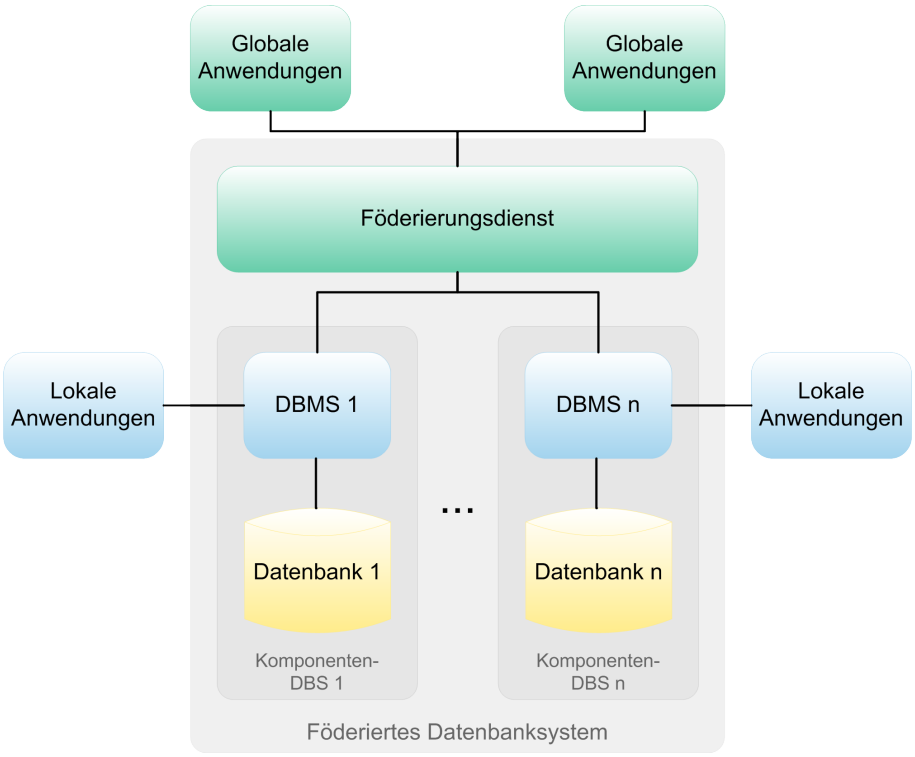


Abbildung 2.5: Referenzarchitektur eines FDBS nach [Con97].

kann. Ein DWH wird nach [GB09] wie folgt definiert:

Definition 2.1. „Ein Data Warehouse ist eine physische Datenbank, die eine integrierte Sicht auf beliebige Daten zu Analysezwecken ermöglicht.“[GB09]

Die traditionellen DBS werden dem klassischen Paradigma der Online-Transaction-Processing (OLTP) zugeordnet. Insbesondere die performante, sichere und korrekte Umsetzung aller Geschäftsprozesse sind die Hauptmerkmale von OLTP. Darüber hinaus gibt es ein Paradigma, das als Online Analytical Processing (OLAP) bezeichnet wird. Dadurch wird die interaktive und explorative Analyse der Daten gewährleistet. Die Funktionalität einer solchen Datenanalyse wird dem Anwender durch analytische IS zur Verfügung gestellt. In der Regel fungiert ein DWH dabei als Datenbasis, weshalb ein DWH durch OLAP klassifiziert werden kann. Ein DWH ist die zentrale Komponente in einem Data-Warehouse-System. Das Data-Warehouse-System ist ein spezielles IS und verfügt über einen Auswertebereich und einen Datenbeschaffungsbereich. In der Abbildung 2.6 wird die Referenzarchitektur eines Data-Warehouse-Systems dargestellt, woran deutlich wird, dass diese beiden Bereiche über verschiedene Komponenten verfügen. Die Funktionalität der einzelnen Komponenten wird ausführlich in [GB09] thematisiert.

Erst durch den Data-Warehouse-Prozess kann ein Data-Warehouse-System die unterschiedlichen Fragestellungen bewerkstelligen. Dieser dynamische Prozess ist für

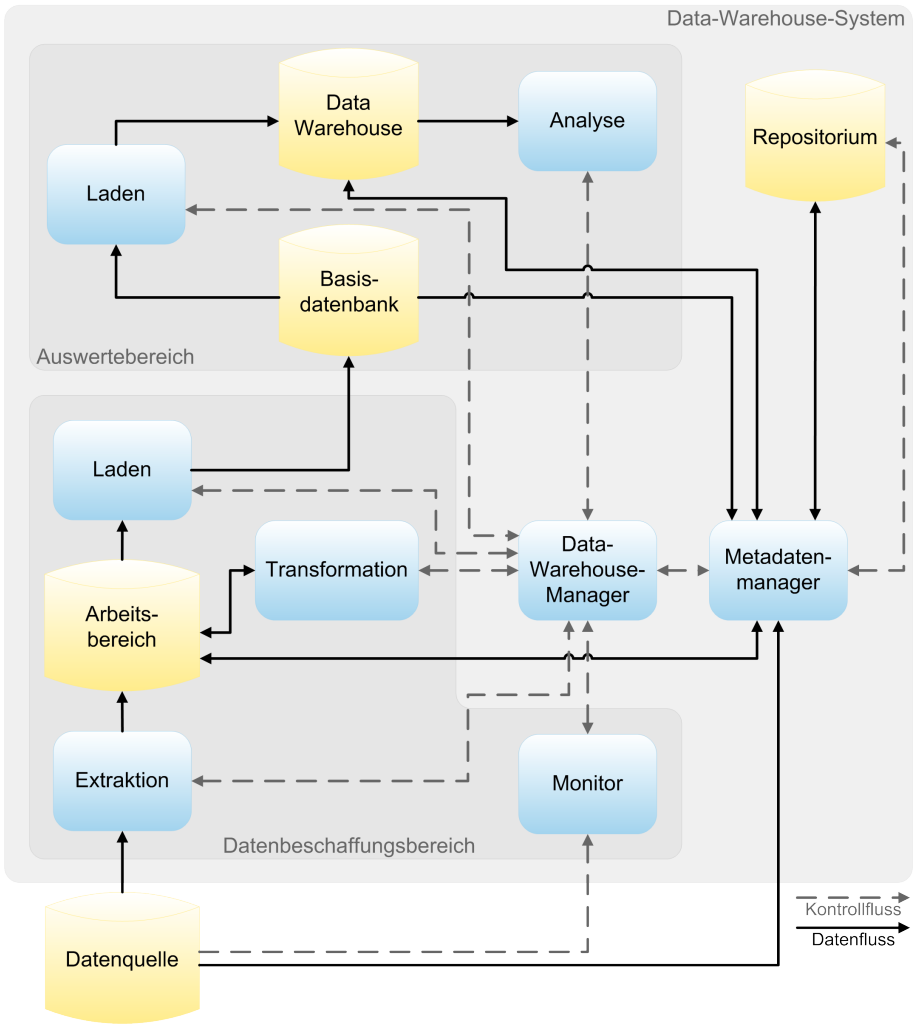


Abbildung 2.6: Referenzarchitektur von Data-Warehouse-Systemen nach [GB09].

die Beschaffung, Speicherung und Analyse der Daten verantwortlich. Der Data-Warehouse-Prozess kann in die folgenden vier Schritte eingeteilt werden:

1. Im ersten Schritt werden die Komponenten Extraktion, Transformation und Laden eingesetzt, die unter dem Begriff ETL-Komponenten zusammengefasst werden [GB09]. Dieser Schritt wird als ETL-Prozess bezeichnet und ist für die Extraktion der Datenbestände aus den Datenquellen und deren Transformation zuständig. Des Weiteren ist der ETL-Prozess für das Laden der strukturierten Datenbestände in das DWH verantwortlich.

2. Die persistente Speicherung der Datenbestände im DWH wird im nächsten Schritt realisiert. Allerdings benötigen einige Analysen oder spezialisierte Anwendungen nicht alle Datenbestände, weshalb sogenannte Data-Marts realisiert werden können.

3. Diese Data-Marts repräsentieren eine spezifische Sicht auf das DWH und werden im dritten Schritt erstellt.

4. Die eigentliche Analyse und Auswertung der Datenbestände erfolgt im letzten Schritt. Die Ergebnisse werden dann den unterschiedlichen Anwendungen bereitgestellt.

Des Weiteren sind Operational Data Stores (ODS) und Realtime Data-Warehouse-Systeme ebenfalls materialisierte Integrationsarchitekturen. Allerdings basieren diese beiden Konzepte auf der Referenzarchitektur eines Data-Warehouse-Systems und werden deshalb in der Regel als Erweiterung angesehen. Die wichtigsten Vorteile der materialisierten Integrationsarchitektur sind die effiziente Datenbereinigung, die unbeschränkten Anfragemöglichkeiten und die gute Performance bei Anfragen. Als Nachteil dieser Integrationsarchitektur kann unter gewissen Umständen die Aktualität des Datenbestands bezeichnet werden. Dieser Aspekt muss aber immer im Kontext der jeweiligen Analyse oder Fragestellung betrachtet werden, weil nicht jede Thematik aktuelle Datenbestände benötigt. Insbesondere bei komplexen Analysen der Finanzmärkte ist die Aktualität der Daten wichtig. Im Kontext der molekularbiologischen Forschung ist eine Aktualisierung des Datenbestands jedes Quartal ausreichend. Als Datenquellen fungieren normalerweise molekularbiologische DB und deren Aktualisierung erfolgt in der Regel jedes Quartal. Allerdings ist dieser mögliche Nachteil bei der Verwendung von ODS und Realtime Data-Warehouse-Systemen nicht existent, weil dort die Aktualität des Datenbestands gewährleistet wird. In der Tabelle 2.6 werden weitere Vor- und Nachteile dieser Integrationsarchitektur dargestellt.

2.3 Zusammenfassung

Die für diese Arbeit relevanten Grundlagen wurden in Kapitel 2 behandelt. Allerdings erfolgte keine detaillierte Einführung in die jeweiligen Grundlagen, weshalb auf Standardwerke aus der Literatur verwiesen wurde.

Als erstes wurden in Abschnitt 2.1 die Grundlagen der zellulären Molekularbiologie erläutert, wobei Abschnitt 2.1.1 die grundlegenden Aspekte von einem Genom und deren Bestandteile behandelt. Ein Gen verfügt über die notwendigen genetischen Informationen die zur Produktion eines Genprodukts benötigt werden. Der biologische Prozess, der diese Thematik repräsentiert, wird als Genexpression bezeichnet und wurde in Abschnitt 2.1.2 beschrieben. Die Proteine sind an zahlreichen Prozessen und Interaktionen beteiligt. Die grundlegenden Prozesse und Strukturen der Proteine wurden in Abschnitt 2.1.3 thematisiert.

Die Grundlagen der Informatik wurden in Abschnitt 2.2 behandelt. Insbesondere die Algorithmen und Datenstrukturen sind in der Informatik eine wichtige Thematik,

weshalb die für diese Arbeit notwendigen Algorithmen und Datenstrukturen in Abschnitt 2.2.1 erläutert wurden. Danach wurden in Abschnitt 2.2.2 unterschiedliche Aspekte von IS und DBS beschrieben sowie ein Lösungsansatz für einen Konflikt vorgestellt, der als *Object-relational Impedance Mismatch* bezeichnet wird. Aufgrund der zahlreichen molekularbiologischen DB wird eine eindeutige Klassifizierung für solche Datenquellen benötigt. Eine mögliche Struktur zur Klassifizierung der molekularbiologischen DB wurde in Abschnitt 2.2.2.2 vorgestellt. Der Abschnitt 2.2.3 behandelt den Themenbereich der Datenintegration. Dabei wurde die Problematik der Verteilung, Autonomie und Heterogenität in Abschnitt 2.2.3.1 behandelt, die auch als orthogonale Dimensionen der Datenintegration bezeichnet werden [LN07]. Die Grundlage der beiden Softwarelösungen, die in der vorliegenden Arbeit erläutert werden, ist eine materialisierte Integrationsarchitektur. Deswegen wurde in Abschnitt 2.2.3.4 dieses Themengebiet ausführlich erläutert.

Als nächstes werden in Kapitel 3 die relevanten und aktuellen verwandten Arbeiten für die Dissertation detailliert vorgestellt, die vergleichbare Softwarelösungen aus der Bioinformatik repräsentieren. Außerdem werden diese Softwarelösungen in Kapitel 3 evaluiert, sodass deren Vor- und Nachteile deutlich werden, die wiederum in Kapitel 4 und 5 berücksichtigt werden.

3 Verwandte Arbeiten

Kapitel 3 thematisiert den aktuellen Stand der Forschung, wobei ausschließlich für die Dissertation relevante verwandte Arbeiten aus der Bioinformatik vorgestellt werden. Es werden zum einen populäre Lösungsansätze der Datenintegration in der Bioinformatik behandelt, zum anderen verschiedene Softwarelösungen, welche die computergestützte Identifikation von regulatorischen Elementen in Nukleotidsequenzen ermöglichen. Die Vor- und Nachteile der Softwarelösungen und deren Lösungsansätze werden durch einen Vergleich und eine Bewertung deutlich, wofür ausgewählte Kriterien festgelegt werden. Allerdings sind nicht alle Leistungsmerkmale, Systemfunktionalitäten und -eigenschaften der verwandten Arbeiten in der Literatur beschrieben. Infolgedessen wird die Software der verwandten Arbeiten mittels Black-Box-Tests, die eine Testmethode der Softwaretests sind, zusätzlich evaluiert. Auf diese Weise können funktionale und nicht-funktionale Anforderungen bei der Software der verwandten Arbeiten überprüft werden.

Als erstes werden in Abschnitt 3.1 vier Softwarelösungen beschrieben, die populäre Lösungsansätze der Datenintegration in der Bioinformatik sind. Danach thematisiert der Abschnitt 3.2 vier Softwarelösungen, die zur Identifizierung von regulatorischen Elementen in Nukleotidsequenzen eingesetzt werden. Das Fazit in Abschnitt 3.3 erläutert die Erkenntnisse und Schlussfolgerungen der Software-Evaluierung, wobei die Vor- und Nachteile der verwandten Arbeiten ausführlich beschrieben werden. Abschließend erfolgt in Abschnitt 3.4 eine kurze Zusammenfassung des Kapitels.

3.1 Ansätze der Datenintegration

Die Realisierung und Bereitstellung einer strukturierten und umfangreichen Datenbasis ist in der Bioinformatik ein wichtiger Aspekt. Die unterschiedlichen Anwendungen und deren komplexe Analysen benötigen eine molekularbiologische Datenbasis als Grundlage. Mit Hilfe von spezifischen Softwarelösungen ist die Realisierung einer solchen Datenbasis möglich. Insbesondere die in Abschnitt 2.2.3.1 dargestellten Probleme müssen durch entsprechende Konzepte bewältigt werden. Die in der Bioinformatik verfügbaren Konzepte der Integration von molekularbiologischen Datenbeständen können in vier Kategorien eingeteilt werden [LR03]:

1. Ein populäres Beispiel für die Kategorie **Attributindexierungssysteme** ist das Sequence Retrieval System (SRS) [EA93]. Des Weiteren sind BioRS [KDH+06] und Entrez [WCE+04, MOPT07] ebenfalls Systeme dieser Kategorie.

2. Die Softwarelösungen BioKleisli [DOTW97], Discovery-Link [HSK+01] und das Object Protocol Model (OPM) [CM95] sind klassische Beispiele, die der Kategorie **Multidatenbanksprachen** zugeordnet werden.

3. Insbesondere das Projekt TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) [SBB+00] und die daraus resultierende Software repräsentiert die Kategorie der **ontologiebasierten Integration**. Die Anwendungen CoryneRegNet [PRT+11] und Ondex [KBT+06, KRV+05] können ebenfalls in diese Kategorie eingeteilt werden.

4. In der Bioinformatik wurden zahlreiche Softwarelösungen realisiert, die der Kategorie der **Data-Warehouse-Technik** zugeordnet werden. Dafür sind BioMart [HBS+09], BioWarehouse [LPW+06], Biozon [BY06], Booly [DEKB10], JBioWH [VPRP+13], PiPa [ASA+11], SYSTOMONAS [CML+07] und Unison [HM09] prominente Beispiele. Allerdings können unter gewissen Aspekten auch CoryneRegNet und Ondex dieser Kategorie zugeteilt werden.

Aufgrund der in Abschnitt 2.2.3.4 genannten Vorteile (z. B. Performance, Verfügbarkeit der Daten und einfache Konzeption) hat sich die Data-Warehouse-Technik in der Bioinformatik durchgesetzt. In den letzten Jahren wurden etliche Systeme realisiert und dem Anwender bereitgestellt. Diese Anwendungen wurden für spezifische molekularbiologische Fragestellungen entwickelt, wodurch eine Verwendung in anderen Projekten und deren Fragestellungen nicht oder nur durch umfangreiche Erweiterungen der jeweiligen Softwarelösung möglich ist. Außerdem sind einige Systeme wie Atlas [SHX+05], Columba [TRM+05], GIMS [CPH+03] und IGD [RKS+94] nicht mehr verfügbar oder werden nicht mehr weiterentwickelt. Darüber hinaus ist der Datenbestand von zahlreichen Systemen nicht mehr aktuell, sodass wichtige Informationen dem Anwender nicht zur Verfügung stehen. Insbesondere die Komplexität und Flexibilität der jeweiligen Software sowie die Einstellung der Projektfinanzierung sind dafür verantwortlich. Des Weiteren ist die Pflege/Weiterentwicklung durch externe Wissenschaftler/Softwareentwickler nicht möglich, weil die Systeme nicht als Open-Source-Software (OSS) realisiert und bereitgestellt wurden. Somit können aktuelle und komplexe molekularbiologische Fragestellungen mit den derzeit verfügbaren Anwendungen nicht erfolgreich bearbeitet werden.

In diesem Abschnitt 3.1 werden exemplarisch Softwarelösungen aus der Literatur wie BioWarehouse, PiPa, CoryneRegNet und Ondex vorgestellt. Diese Anwendungen verfügen über wichtige Erkenntnisse und reflektieren den aktuellen Stand der Forschung. Die Anwendungen BioWarehouse und PiPa repräsentieren klassische Projekte der Datenintegration in der Bioinformatik. Dabei sind besonders die Integration

von molekularbiologischen Daten und die Bereitstellung einer konsistenten Datenbasis der Schwerpunkt. Mit Hilfe einer Software-Infrastruktur kann der Anwender die Funktionalität von den Anwendungen BioWarehouse und PiPa verwenden. Zudem können diese Systeme lokal installiert und konfiguriert werden. Die Software BioWarehouse wird in Abschnitt 3.1.1 vorgestellt. Anschließend behandelt Abschnitt 3.1.2 die Anwendung PiPa. Darüber hinaus gibt es Systeme, die genau genommen zwei der oben genannten Kategorien der Datenintegration in der Bioinformatik repräsentieren. Exemplarisch dafür werden die Softwarelösungen CoryneRegNet und Ondex präsentiert. Aufgrund der Kombination der Data-Warehouse-Technik mit dem Konzept der Ontologie ist eine eindeutige Zuordnung nicht möglich. Darüber hinaus sind beide Anwendungen keine klassischen Projekte der Datenintegration in der Bioinformatik. Die Integration der molekularbiologischen Daten erfolgt bei CoryneRegNet und Ondex durch eine entsprechende Komponente innerhalb der Software. Mit Hilfe von CoryneRegNet und Ondex können die molekularbiologischen Daten visualisiert werden und durch zweckmäßige Methoden können neue molekularbiologische Daten identifiziert werden, sodass bisher unbekannte molekularbiologische Fragestellungen verständlich werden. In Abschnitt 3.1.3 wird CoryneRegNet diskutiert, das als Webanwendung implementiert wurde. Innerhalb der Software Ondex werden die Konzepte der semantischen Datenintegration, Text Mining und Methoden aus der Graphentheorie kombiniert und eingesetzt. Abschließend wird in Abschnitt 3.1.4 die Anwendung Ondex ausführlich erläutert.

3.1.1 BioWarehouse

BioWarehouse ist eine OSS und steht unter der Mozilla Public License (MPL). Derzeit ist BioWarehouse in der Version 4.6 verfügbar. Diese Software ist ein Teil des Bio-SPICE Projekts [GLP+03], das unterschiedliche Softwarelösungen für die Systembiologie bereitstellt.

Mit Hilfe von BioWarehouse können benutzerspezifische molekularbiologische DB erstellt werden, wobei die Realisierung als DWH erfolgt. Allerdings werden nur die RDBMS MySQL und Oracle unterstützt. Das Schema für das jeweilige DWH wird von BioWarehouse bereitgestellt. Diese Vorgehensweise entspricht somit dem Paradigma der engen Kopplung. Die Abbildung 3.1 zeigt ein abstraktes Schema von BioWarehouse, wodurch die einzelnen Datentypen und deren Beziehungen identifiziert werden können. Anhand der Abbildung 3.1 wird deutlich, dass die Datentypen in zwei Kategorien eingeteilt werden. Neben den spezifischen Datentypen für die molekularbiologischen Daten werden auch Datentypen benötigt, die Metainformationen repräsentieren. Damit externe Softwareentwickler und auch Anwender das Schema problemlos interpretieren können, wurde bei der Realisierung besonders die Verständlichkeit berücksichtigt. Dadurch kann das Schema schnell und problemlos um zusätzliche Datentypen erweitert werden. Zudem wurde das Schema vorzugsweise für Datenbestände von Prokaryoten optimiert. Allerdings soll in Zukunft auch

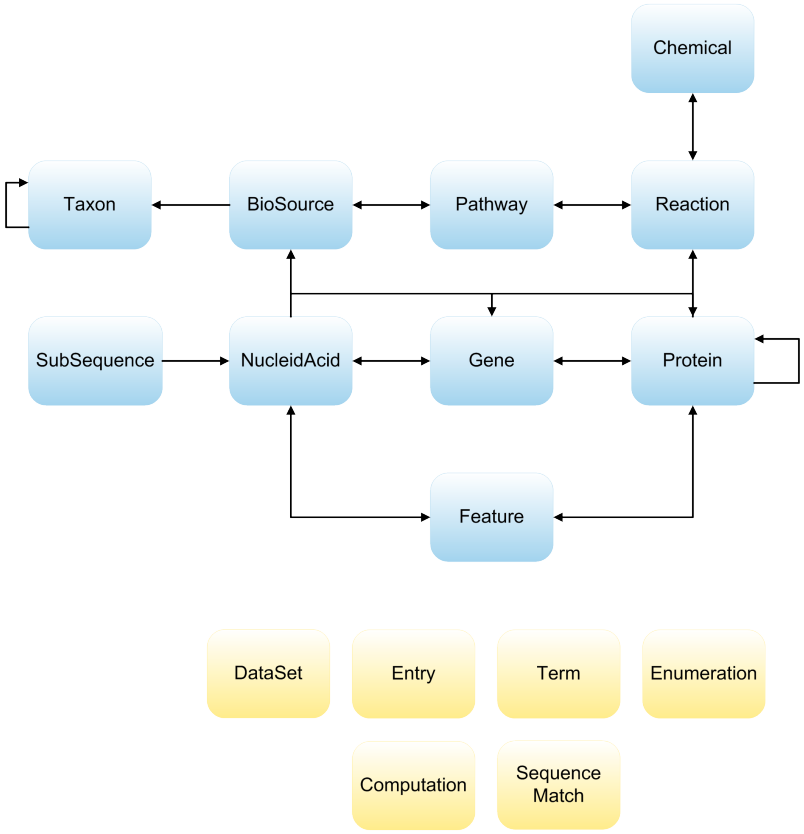


Abbildung 3.1: Schema von BioWarehouse nach [LPW+06].

die Unterstützung von eukaryotischen Datenbeständen verbessert und gewährleistet werden [LPW+06]. Des Weiteren ermöglichen entsprechende Metainformationen die Verwaltung von unterschiedlichen Versionen eines Datensatzes. Auf diese Weise ist die Identifizierung von Unterschieden zwischen neuen und alten Datensätzen möglich.

Durch verfügbare *Loader* ist die Integration der Daten aus den externen Datenquellen möglich. Bei der Implementierung wurden die Programmiersprachen Java und C verwendet. BioWarehouse verfügt über spezifische *Loader* für die molekularbiologischen DB BioCyc, Comprehensive Microbial Resource (CMR), ENZYME [Bai00], Eco2dbase, GenBank, Gene Ontology (GO) [The12a], KEGG, MetaCyc, NCBI Taxonomy und UniProt. Insbesondere die Datenbestände über metabolische Stoffwechselwege, Enzyme und vollständig sequenzierte Genome von Mikroorganismen werden dabei aus den Datenquellen berücksichtigt. Außerdem werden *Loader* für die standardisierten Datenaustauschformate BioPAX [DCP+10] und MAGE-ML zur Verfügung gestellt. Dadurch existiert die Möglichkeit, zusätzliche externe Daten zu integrieren. Die Konvertierung der Daten in das einheitliche Schema von BioWarehouse absolvieren ebenfalls die spezifischen *Loader*. Ein besonderes Merkmal der *Loader* ist deren Fehlertoleranz während der Integration. Sofern bei dem Prozess der Datenintegration ein Datensatz einen Fehler verursacht, wird dieser Prozess

dennoch ordnungsgemäß und vollständig beendet. Allerdings wird der fehlerhafte Datensatz explizit im System gekennzeichnet, woraus sich ein entscheidender Vorteil ergibt, weil die übrigen Datensätze konsistent verfügbar sind. Es ist somit nicht notwendig, den gesamten Prozess noch einmal durchzuführen, da stattdessen nur der „defekte Datensatz“ analysiert und der Fehler identifiziert werden muss.

Es werden spezielle Java *Utility*-Klassen von BioWarehouse bereitgestellt, wodurch Softwareentwickler/Wissenschaftler zusätzliche *Loader* implementieren können. Des Weiteren ist es damit möglich, Anwendungen zu entwickeln, die ein DWH als Datenbasis verwendet, das mit BioWarehouse erstellt wurde.

Die Software BioWarehouse kann auf zwei unterschiedliche Arten eingesetzt werden. Der Anwender kann einerseits die beiden öffentlichen Systeme PublicHouse und EcoliHouse benutzen (siehe Abschnitt 3.1.1.1), andererseits kann er eine Software-Infrastruktur auf der Kommandozeile verwenden, welche dann lokal installiert und konfiguriert werden kann. Allerdings wird dafür ein Linux-Derivat als Betriebssystem benötigt, infolgedessen die Plattformunabhängigkeit der Software nicht mehr gewährleistet ist. Darüber hinaus wird keine grafische Benutzeroberfläche (GBO) zur Verfügung gestellt. Mit Hilfe der Software-Infrastruktur können spezialisierte DWH konstruiert werden, worauf weitere Softwarelösungen operieren können. Die Aktualisierung der DWH erfolgt jedoch nicht automatisch, weil ein entsprechender Service nicht verfügbar ist. Aus diesem Grund muss der Anwender den Datenbestand des DWH manuell aktualisieren.

3.1.1.1 PublicHouse und EcoliHouse

Mit Hilfe von BioWarehouse wurden PublicHouse und EcoliHouse realisiert, die MySQL als RDBMS benutzen. Damit der Anwender diese beiden Systeme verwenden kann, ist eine Registrierung notwendig. Zudem wird der Zugriff über das Internet gewährleistet. Bei der Implementierung der GBO wurde jeweils phpMyAdmin [Rei09] eingesetzt. Diese Software bietet dem Anwender grundlegende Funktionen hinsichtlich der Formulierung von Abfragen an das DWH und der Administration. Darüber hinaus kann durch einen *Structured Query Language* (SQL) Client ebenfalls der Datenzugriff realisiert werden. Allerdings benötigt der Anwender bei beiden Zugriffsmöglichkeiten entsprechende Kenntnisse. Infolgedessen ist die Benutzerfreundlichkeit eher kritisch zu bewerten, zumal keine standardisierten Methoden zur Analyse von molekularbiologischen Fragestellungen vorhanden sind.

Des Weiteren gibt es noch zwei unterschiedliche Merkmale zwischen PublicHouse und EcoliHouse. Aufgrund von Restriktionen beim Urheberrecht ist der Datenbestand von KEGG in PublicHouse nicht verfügbar. Im Gegensatz zu PublicHouse ist EcoliHouse ein DWH speziell für das Bakterium Escherichia coli. Der Fokus der Datenbestände ist dabei der Sicherheitsstamm K12 von Escherichia coli.

3.1.2 PiPa

Mit Hilfe der Software PiPa können ausschließlich Datenbestände über Stoffwechselwege und Proteine integriert werden. Außerdem wird das Datenmodell für die Datenintegration von PiPa bereitgestellt. Diese Herangehensweise wird dem Paradigma der engen Kopplung zugeordnet. Die Abbildung 3.2 zeigt das Datenmodell von PiPa. Als RDBMS wird MySQL eingesetzt. Der Einsatz von zusätzlichen RDBMS erfor-

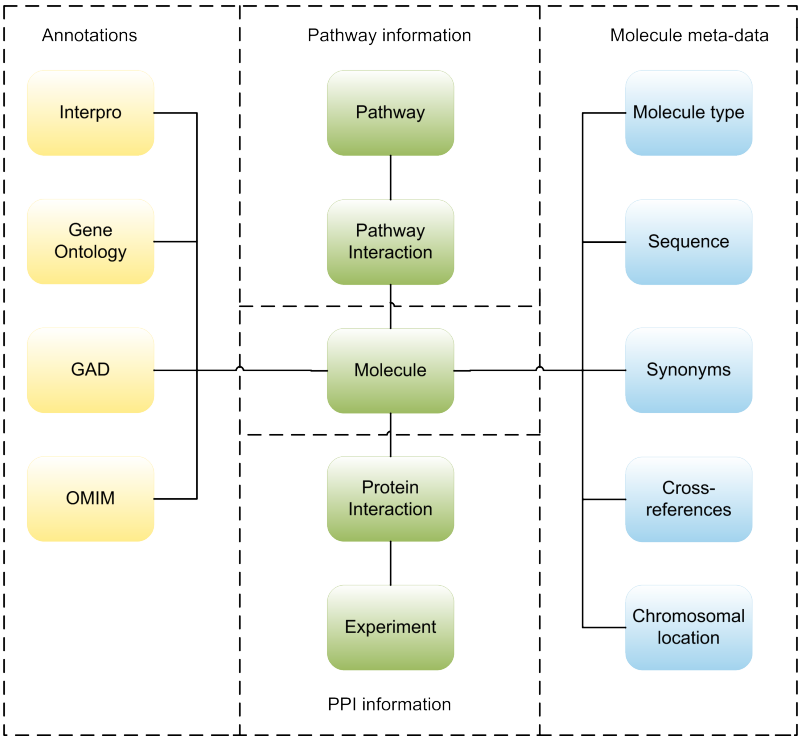


Abbildung 3.2: Datenmodell von PiPa nach [ASA+11].

dert Änderungen an PiPa und ist nicht beabsichtigt. Die Software und deren Quellcode ist nicht frei, sondern nur auf Anfrage bei den Softwareentwicklern verfügbar. Obwohl bei der Entwicklung primär die Programmiersprache Java eingesetzt wurde, ist PiPa nicht plattformunabhängig. Aufgrund von spezifischen UNIX-Parametern, die im Quellcode explizit angegeben werden, ist die Software derzeit nur unter einem UNIX-Derivat verwendbar.

Im Gegensatz zu BioWarehouse verfügt PiPa über eine GBO, wodurch der Anwender die Funktionalität der Datenintegration, Administration und Aktualisierung kontollieren kann. Zusätzlich bietet diese Oberfläche Statistiken zu den jeweiligen Datenquellen und dem integrierten Datenbestand.

Das Datenmodell in der Abbildung 3.2 ähnelt einem Schneeflockenschema, das häufig bei einem DWH eingesetzt wird. Außerdem zeigt die Abbildung 3.2 die Segmentierung der Datenbestände auf die jeweiligen Kategorien, die einzelnen Daten-

typen sowie deren Beziehungen. Der Fokus von PiPa liegt auf Datenbeständen über Stoffwechselwege, PPI und notwendigen Metadaten über Proteine. Daher wurden relevante Datenquellen selektiert, welche die entsprechenden Daten zur Verfügung stellen. Diese Datenquellen und deren Einteilung in PiPa werden in der Tabelle 3.1 dargestellt. Allerdings werden beim Prozess der Datenintegration nur Datenbestände

Stoffwechselwege	**PPI**	**Metainformationen über Proteine**
BioCyc	BioGrid	Gad
Inho	DIP	GO
KEGG	HPRD	Interpro
Pathway Commons	IntAct	KEGG
PID	MINT	OMIM
Reactome	MIPS	Reactome
Spike	-	UniProt

Tabelle 3.1: Verfügbare Datenquellen und deren Einteilung in PiPa nach [ASA+11].

der Organismen *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus*, *Nematoda*, *Rattus norvegicus* und *Saccharomyces cerevisiae* berücksichtigt.

Die Integration der Daten aus den einzelnen Datenquellen erfolgt durch spezielle *Wrapper*. Das Softwaredesign von PiPa verfügt über eine *Plug-in*-Struktur, wobei jeder *Wrapper* ein entsprechendes *Plug-in* symbolisiert. Durch diese Struktur können externe Wissenschaftler/Softwareentwickler eigene *Wrapper* implementieren und so die Software spezifisch weiterentwickeln. Die Informationen der Datenquellen sind normalerweise über standardisierte Austauschformate verfügbar. Dadurch kann ein *Wrapper* nicht nur explizit für eine Datenquelle eingesetzt werden, sondern auch für unterschiedliche Datenquellen aus derselben molekularbiologischen Kategorie. Infolgedessen ist es nicht notwendig, für jede Datenquelle einen spezifischen *Wrapper* bereitzustellen.

Allerdings kann durch diese Konzeption die Komplexität sowie die Wartbarkeit der *Wrapper* problematisch werden. Mit Hilfe der Tabelle 3.1 können die entsprechenden Datenquellen für die jeweilige Kategorie identifiziert werden. Die Datenbestände für die Kategorie Stoffwechselwege werden im Austauschformat BioPAX zur Verfügung gestellt. Infolgedessen wurde in PiPa ein spezieller *Wrapper* für diese Kategorie realisiert. Die Bezugsquellen für die PPI werden im Austauschformat PSI-MI XML [HMPB+04] bereitgestellt, sodass exklusiv dafür ein *Wrapper* implementiert wurde. Ein Sonderfall ist die Domäne der Metainformationen über Proteine. Dort war es nicht möglich, ein standardisiertes Austauschformat für alle Datenquellen zu lokalisieren, weil einige Bezugsquellen nur proprietäre Dateiformate anbieten. Aus diesem Grund verfügt PiPa für diese Kategorie über spezifische *Plug-ins* zum Analysieren und Konvertieren der Informationen. Außerdem erfolgt bei allen Datensätzen, die ein

Protein repräsentieren, eine Normalisierung des Identifikators, sodass jeder Datensatz über einen eindeutigen Identifikator von der molekularbiologischen DB UniProt verfügt. Aufgrund der Datenmenge ist die Datenintegration bei PiPa zeitintensiv, was durch den Prozess der Normalisierung nochmals forciert wird [ASA+11].

3.1.3 CoryneRegNet

Im Gegensatz zu BioWarehouse und PiPa wurde CoryneRegNet mittels der Skriptsprache PHP und der Programmiersprache Java als Webanwendung realisiert, wodurch die Plattformunabhängigkeit gewährleistet wird. Als RDBMS wird MySQL verwendet. CoryneRegNet ist ebenfalls eine OSS, die aber unter der *Academic Free License* (AFL) lizenziert ist. Mit Hilfe von CoryneRegNet ist die genomweite Rekonstruktion von transkriptionellen regulatorischen Netzwerken möglich. Allerdings wird derzeit auf 11 Corynebakterien sowie den Sicherheitsstamm K12 von Escherichia coli fokussiert. Das Ziel sind Hypothesen, die *in silico* selektiert und anschließend durch Laborexperimente verifiziert werden. Die Abbildung 3.3 zeigt die einzelnen Zyklen der Softwareentwicklung von CoryneRegNet als auch die Innovationen in der jeweiligen Version. CoryneRegNet ist derzeit in der Version 6.0 verfügbar.

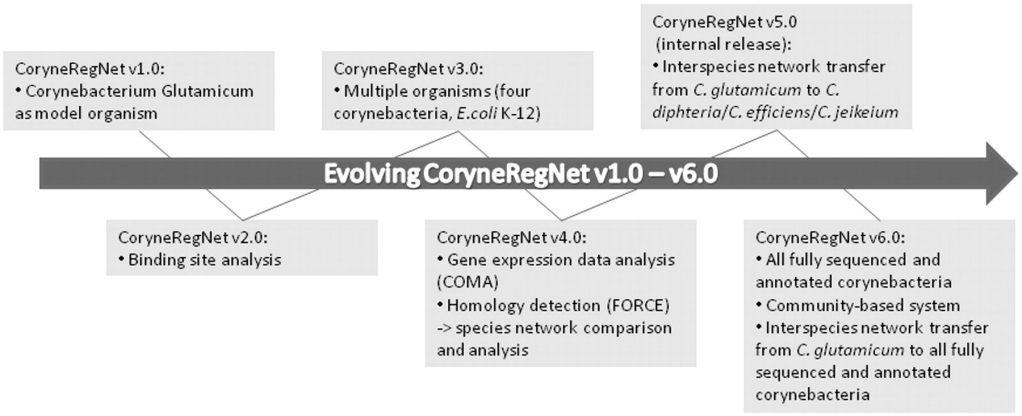


Abbildung 3.3: Die jeweiligen Entwicklungsstadien von CoryneRegNet [PRT+11].

Die Grundlage von CoryneRegNet ist eine Ontologie, die über ein definiertes und kontrolliertes Vokabular verfügt. Mit Hilfe dieser Ontologie werden die heterogenen Daten aus den externen Datenquellen auf die bereitgestellte Datenstruktur projiziert. Aufgrund der Festlegung der Strukturierung der Daten entspricht diese Konzeption der engen Kopplung. In der Abbildung 3.4 wird das Entity-Relationship-Modell (ERM) von CoryneRegNet dargestellt. Anhand der Abbildung 3.4 wird deutlich, dass es zwei Bereiche gibt. Die ontologiebasierte Datenstruktur repräsentiert die Informationen über Gene, Proteine sowie deren Interaktionen. Die Entitätstypen *Concept* und *Relation* sind dabei die zentralen Komponenten in dieser Datenstruktur. Durch die generalisierte Datenstruktur ist es möglich, spezifische Werte wie die

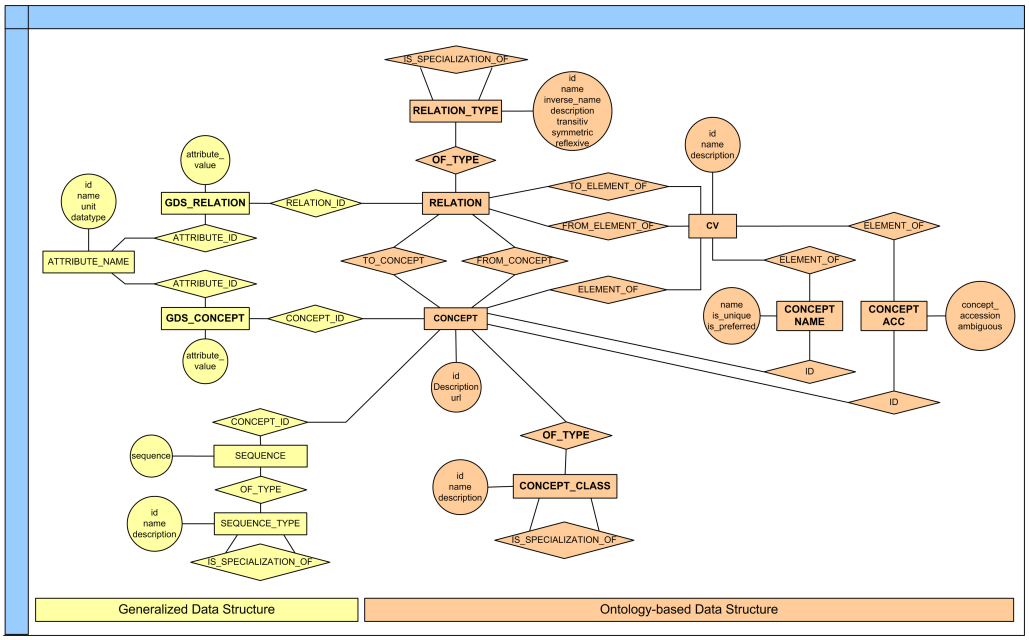


Abbildung 3.4: ERM von CoryneRegNet [BBC+06].

Start- und Stop-Position eines Gens darzustellen. Das Datenmodell von CoryneRegNet ist vergleichbar mit dem von Ondex, das im Abschnitt 3.1.4 vorgestellt wird. Ein Vorteil dieser Datenstruktur ist die Migration auf unterschiedliche Fragestellungen. Dieses Konzept wurde auch bei RhizoRegNet [KBW+11], MycoRegNet [KKG+09] und EhecRegNet[PRN+12] erfolgreich eingesetzt. Der Schwerpunkt dieser Softwarelösungen sind jedoch andere Organismen. Allerdings kann die Entwicklung einer Ontologie kompliziert und zeitintensiv sein. Darüber hinaus kann die Abbildung von speziellen Sachverhalten die Komplexität einer Ontologie forcieren, wodurch die Verständlichkeit reduziert wird.

Die Strukturierung der Systemarchitektur von CoryneRegNet erfolgt durch ein *Front-End* und ein *Back-End*. Die Abbildung 3.5 präsentiert die Systemarchitektur von CoryneRegNet, wobei deutlich wird, dass der Data-Warehouse-Prozess im *Back-End* erfolgt. Die Datenbeschaffung wird durch einen *Parser* realisiert, der einen ETL-Prozess zur Verfügung stellt. Insbesondere experimentelle Daten, Informationen über Genregulationen und Genomannotationen werden im DWH persistent gespeichert. Außerdem werden bereits im Data-Warehouse-Prozess notwendige und komplexe Kalkulationen realisiert, sodass diese nicht während der Laufzeit von CoryneRegNet durchgeführt werden müssen.

Der Anwender kann durch eine Webanwendung, die über zwei Perspektiven verfügt, auf die Informationen aus dem DWH zugreifen. Diese Perspektiven werden bei der Webanwendung als *Experimental* und *Predicted* bezeichnet. Zudem bietet die Webanwendung eine Funktionalität zur Visualisierung von Netzwerken und es werden unterschiedliche Suchformulare und -möglichkeiten bereitgestellt. Die Visua-

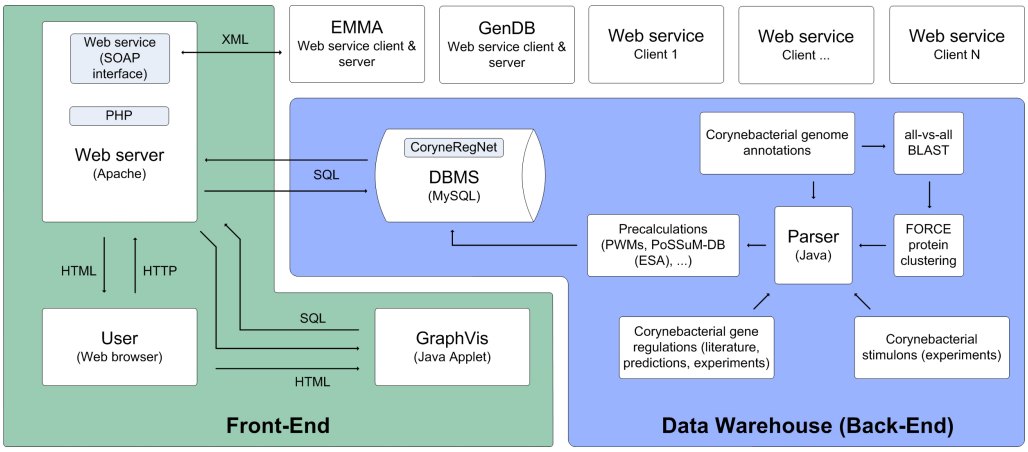


Abbildung 3.5: Systemarchitektur von CoryneRegNet [Bau07].

lisierung der Netzwerke erfolgt durch die Klassenbibliothek *yFiles for Java*[1] und ein Java-Applet, das in CoryneRegNet als GraphVis bezeichnet wird. Dabei erfolgt die Visualisierung der Daten durch Knoten und Kanten, wobei ein Knoten ein Gen darstellt und eine Kante eine regulative Interaktion repräsentiert. Des Weiteren kann der Anwender interaktiv eigene Datensätze über regulatorische Elemente hinzufügen und dadurch ein Netzwerk spezifisch erweitern. Diese Datensätze sind aber nur temporär während der gegenwärtigen *Session* verfügbar. Zudem verfügt das Java-Applet über standardisierte Algorithmen aus der Graphentheorie und grundlegende Techniken zur Visualisierung. Seit der Version 6.0 ist die Eingabe von eigenen Datenbeständen durch ein Webformular möglich. Diese Datensätze werden kontrolliert und sofern keine Unzulänglichkeiten identifiziert werden, erfolgt die persistente Speicherung im DWH. Danach haben alle Anwender Zugriff auf diesen neuen Datenbestand.

Die spezifischen Softwarekomponenten oder -lösungen wie FORCE [WBLR07], MoRAine [BWW+08], PoSSuMsearch [BSH+04] und TransClust [WEL+10] realisieren in CoryneRegNet zusätzliche Möglichkeiten bei der Datenanalyse. Durch PoSSuMsearch ist die Vorhersage von potenziellen TFBS in Nukleotidsequenzen möglich, wobei die dazu notwendigen ESA bereits im Data-Warehouse-Prozess angelegt wurden. Ein *Sequence clustering* kann in CoryneRegNet mittels FORCE durchgeführt werden. Dafür sind aber die Ergebnisse einer *all-vs-all BLAST* Analyse von allen Aminosäuresequenzen aller Organismen aus CoryneRegNet erforderlich. Diese zeitintensive Analyse erfolgte ebenfalls im Data-Warehouse-Prozess.

Darüber hinaus sind relevante Datenbestände von EMMA [DAG+09] und GenDB[2] über einen Webservice in CoryneRegNet verfügbar. Dadurch sind aktuelle und zusätzliche Datensätze verfügbar. Allerdings können Einschränkungen bei der Performance vorkommen, weil diese externen Datensätze nicht persistent über das DWH verfügbar sind. Damit externe Anwendungen den Datenbestand von CoryneRegNet

[1] http://www.yworks.com
[2] http://www.cebitec.uni-bielefeld.de/groups/brf/software/gendb_info/

verwenden können, wird dafür ein eigener Webservice bereitgestellt. Bei der Realisierung und Bereitstellung eines Webservice müssen Sicherheitsaspekte und der *Overhead* während der Datenübertragung berücksichtigt werden. Andernfalls ist die Verwendung durch externe Forschungseinrichtungen und deren Softwarelösungen nicht realisierbar und auch nicht praktikabel. Anhand Abbildung 3.5 wird die Interaktion dieser Softwarekomponenten im Kontext der Systemarchitektur deutlich.

Des Weiteren ist die Verifizierung von Ergebnissen aus DNA-Microarrays möglich. Anhand der verfügbaren regulativen Netzwerke aus CoryneRegNet werden die Ergebnisse auf ihre Konsistenz überprüft. Dadurch können entweder Unzulänglichkeiten oder unbekannte Interaktionen eines Laborexperiments identifiziert werden. Diese Funktion wird in der Webanwendung als COMA bezeichnet. Ein Anwendungsfall von CoryneRegNet ist in [BWKT09] dargestellt, wodurch der praktische Stellenwert deutlich wird.

3.1.4 Ondex

Anhand von Abbildung 3.6 wird deutlich, dass Ondex die Konzepte semantische Datenintegration, Text Mining und graphbasierte Analyse/Visualisierung auf drei Komponenten verteilt und diese kombiniert. Die Implementierung dieser unterschiedlichen Softwarekomponenten wurde durch die Programmiersprache Java sowie den Programmbibliotheken Apache CXF[3] und Apache Lucene[4] ermöglicht. Die Anwendung Ondex steht unter der *GNU General Public License* (GPL) und wurde als Software-Infrastruktur realisiert. Die aktuelle Version von Ondex ist vom Oktober 2012 und deren Verwendung erfordert eine einmalige Registrierung. Der Schwerpunkt von Ondex sind spezifische molekularbiologische Anwendungsfälle für Pflanzen. Durch Ondex ist es möglich, experimentelle Datenbestände zu analysieren und zu interpretieren. Dabei werden zwischen den experimentellen Daten und den Daten aus externen Datenquellen Beziehungen identifiziert, sodass Abhängigkeiten und Wechselwirkungen deutlich werden. Ein typischer Anwendungsfall von Ondex wird in [KBT+06] behandelt.

Durch generische und spezifische *Parser* wird die Integration der Daten aus externen Datenquellen und standardisierten Austauschformaten realisiert. Derzeit werden sieben generische *Parser* für standardisierte Austauschformate und 25 spezifische *Parser* für externe Datenquellen bereitgestellt. Außerdem können weitere Parser innerhalb von 1 - 10 Tagen implementiert werden [KBT+06]. Im Gegensatz zu BioWarehouse, CoryneRegNet und PiPa verfügt Ondex nicht über ein RDBMS. Mit Hilfe einer eingebetteten Berkeley-Datenbank (Berkeley DB) werden die Daten als Schlüssel-Wert-Beziehungen gespeichert.

Die externen Daten werden wie bei CoryneRegNet mit Hilfe einer Ontologie, welche

[3] http://cxf.apache.org/
[4] http://lucene.apache.org/

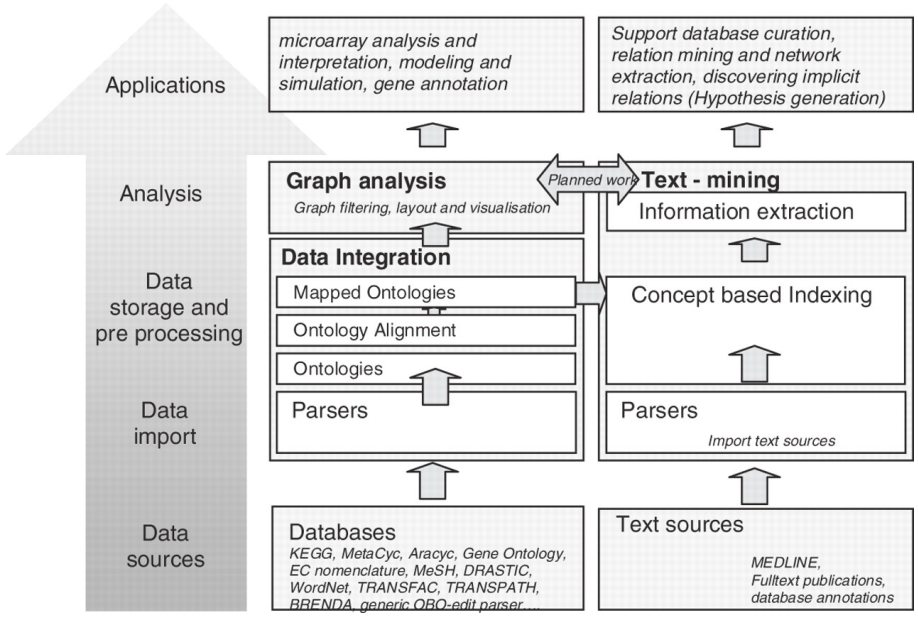


Abbildung 3.6: Systemarchitektur von Ondex [KBT+06].

ebenfalls über ein definiertes und kontrolliertes Vokabular verfügt, auf eine ontologiebasierte Datenstruktur projiziert. Damit eine graphbasierte Analyse und Visualisierung grundsätzlich möglich ist, verfügt die Datenstruktur über entsprechende Konzepte. In der Abbildung 3.7 wird das ERM von Ondex präsentiert. Dabei wird die Komplexität erkennbar, folglich ist die Verständlichkeit und Erweiterbarkeit des ERM problematisch. Darüber hinaus ist das ERM von Ondex und CoryneRegNet sehr ähnlich strukturiert. Bei einen Vergleich der Abbildungen 3.4 und 3.7 wird dieses deutlich. Allerdings ist das ERM von Ondex wegen der Konzepte Text Mining und graphbasierte Analyse und Visualisierung wesentlich komplexer.

Bei der Datenintegration von Ondex wird überwiegend die technische und semantische Heterogenität beseitigt. Anschließend werden zwischen den unterschiedlichen Daten, die als Entitäten repräsentiert werden, die Gemeinsamkeiten und Beziehungen identifiziert. Allerdings erfolgt keine Datenmigration zwischen gleichen oder ähnlichen Entitäten, sondern eine Verknüpfung. Dabei kann ein Vergleich der Entitäten über den Namen, die *Accession number* und die strukturellen Eigenschaften der Ontologie erfolgen. Sofern Entitäten Proteine oder Enyzme darstellen, ist auch ein Vergleich über die jeweilige Aminosäuresequenz möglich. Nach der Datenintegration verfügt jeder externer Datenbestand über die gleiche Datenstruktur und ist zudem noch mit gleichen oder ähnlichen Daten verknüpft. Dementsprechend wird dieser Ansatz dem Paradigma der losen Kopplung zugeordnet. Die Softwarekomponente, die das Text Mining durchführt, erweitert die Datenintegration. Infolgedessen ist die Identifikation und die Extraktion von Informationen aus Publikationen möglich. Dadurch können auch Informationen verwendet werden, die nicht über molekularbiologische DB verfügbar sind. Jedoch ist die korrekte Identifizierung und Interpre-

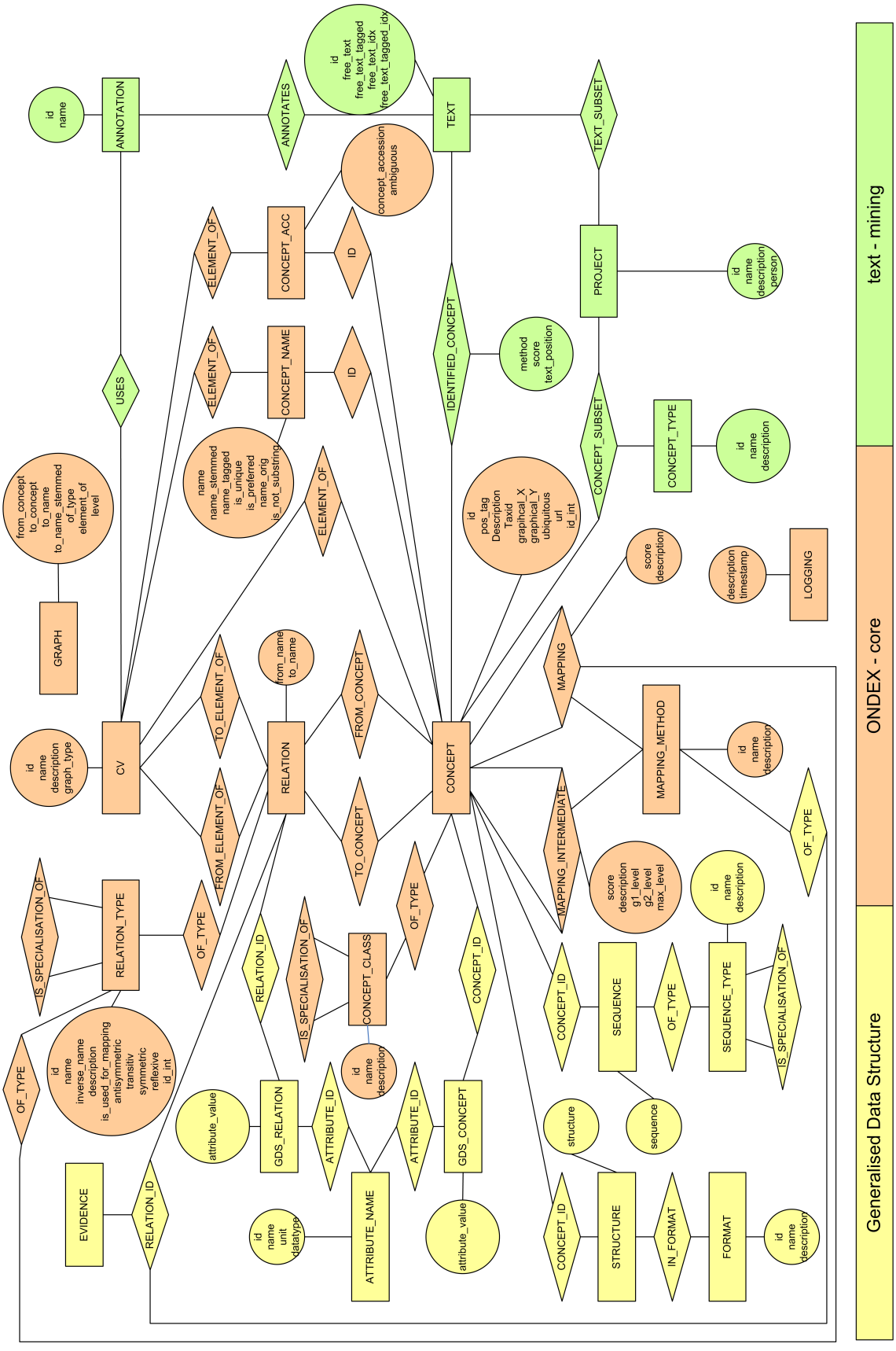


Abbildung 3.7: ERM von Ondex.

tation von Informationen aus einer Publikation durch Text Mining im Kontext der Molekularbiologie relativ aufwendig. Sofern keine Mechanismen zur Filterung oder entsprechende Regel verfügbar sind, sind diese Informationen nicht geeignet, um daraus neue molekularbiologische Daten zu generieren.

Die Darstellung der jeweiligen molekularbiologischen Anwendungsfälle und deren Daten erfolgt als Graph, sodass verschiedene Layouts, Filter und Analysen im Kontext der Graphentheorie angewendet werden können. Dabei ist das Ziel, unbekannte Interaktionen zu lokalisieren und komplexe Zusammenhänge aus der Molekularbiologie zu verstehen.

Durch einen Datenexport kann der Graph oder dessen Informationen in standardisierte Austauschformate exportiert werden. Auf diese Weise können die Daten auch in zusätzlichen Anwendungen analysiert werden oder externen Wissenschaftlern zur Verfügung gestellt werden.

3.2 Identifizierung regulatorischer Elemente

In den letzten Jahren wurden die Genome von zahlreichen Organismen erfolgreich sequenziert. Die daraus resultierenden Daten wurden der wissenschaftlichen Gemeinschaft für weitere Analysen zur Verfügung gestellt. Die Tabelle 2.1 in Abschnitt 2.1.1 zeigt ausgewählte Organismen, die bereits sequenziert wurden. Die Analyse der Daten konzentriert sich nicht nur auf die erfolgreiche Identifizierung der Gene und deren Funktionalität, sondern auch auf die Regulation der Gene durch TF. Die Notwendigkeit der TF und deren Eigenschaften wurde in Abschnitt 2.1.2 thematisiert. Insbesondere die Identifikation von potenziellen TFBS und deren Gene sowie die Interaktion zwischen TF und anderen DNA-bindenden Proteinen sind wichtige Aspekte bei der Grundlagenforschung. Außerdem ist die Identifizierung von charakteristischen Sequenzmotiven in der Wissenschaft eine interessante Thematik, wofür [SPdR+09, PIK+11] ein entsprechendes Beispiel ist.

Die TFBS ist eine spezifische Nukleotidsequenz auf der DNA und ermöglicht die Interaktion mit TF. Dadurch können TF die Effizienz der Transkription der Gene positiv oder negativ beeinträchtigen. Die TF können normalerweise mit unterschiedlichen TFBS interagieren und so die Regulation von zahlreichen Genen beeinflussen. Durch experimentelle Analysen wie *Chromatin-Immunopräzipitation* (ChIP) [OSP97], *DNase Footprinting Assay* [GS78] oder *Electrophoretic Mobility Shift Assay* (EMSA) können DNA-Protein-Interaktionen nachgewiesen werden. Es wurde in [SCB95] eine TFBS mittels EMSA identifiziert. Im Gegensatz dazu konnte in [LGM+08] eine TFBS durch ChIP identifiziert werden. Allerdings sind solche Analysen zeit- und kostenintensiv, weshalb spezielle Softwarelösungen aus der Bioinformatik benötigt werden. Diese Softwarelösungen können potenzielle TFBS oder andere charakteristische Sequenzmotive von regulatorischen Elementen identifizieren. Die Identifikation erfolgt entweder im vollständigen Genom oder in ausgewählten

Nukleotidsequenzen eines Organismus. Außerdem gibt es regulatorische Elemente, die evolutionär konserviert und an wichtigen Regulationen beteiligt sind. Die Organismen *Homo sapiens* und *Mus musculus* verfügen beide in der 5'-Upstream-Region des Gens Fhl2 an vergleichbarer Position über die TFBS 5'-CCTTATATGG-3' [PSD+04]. Durch diese TFBS ist der TF SRF (Serum-Response-Faktor) an der Regulation des Gens Fhl2 in beiden Organismen beteiligt [PSD+04]. Deswegen ist es sinnvoll, eine Vorhersage für verschiedene Organismen durchzuführen, weil ein Vergleich der Ergebnisse einen Hinweis auf die Signifikanz der regulatorischen Elemente liefern kann. Auf diese Weise kann auch die Anzahl von falsch positiven Ergebnissen in der Ergebnismenge reduziert werden [LSM+03]. Diese Herangehensweise wird als *phylogenetic footprinting* oder *cross-species* Vergleich bezeichnet. Durch spezialisierte Software der Bioinformatik können computergestützte Simulationen oder Analysen durchgeführt werden, die als *in silico* Experimente bezeichnet werden. Infolgedessen kann der Wissenschaftler effizient und kostengünstig Ergebnisse generieren, die auf einer Hypothese basieren. Danach müssen für vielversprechende Ergebnisse entsprechende *in vitro* Experimente durchgeführt werden, wodurch die Hypothese überprüft wird. Abschließend kann die Relevanz durch komplexe und langwierige Studien, die *in vivo* erfolgen, verifiziert werden.

Die verfügbaren Softwarelösungen der Bioinformatik verfügen über verschiedene Algorithmen und unterscheiden sich in den jeweiligen Lösungsansatz. Der Typ und die Komplexität des Organismus ist ein wichtiges Kriterium, das bei der Auswahl eines Algorithmus berücksichtigt werden sollte. Dadurch kann die Effizienz und die erfolgreiche Bewerkstelligung der Vorhersage von potentiellen TFBS beeinflusst werden. Es gibt spezifische Algorithmen, die für Prokaryoten und/oder Eukaryoten realisiert wurden und somit für die jeweiligen Organismen besser geeignet sind. Diese Algorithmen und deren Klassifikation werden in [DD07] beschrieben. Die verfügbaren Algorithmen basieren auf zwei unterschiedlichen Lösungsansätzen, wobei einige Algorithmen zur Vorhersage von potenziellen TFBS in Nukleotidsequenzen experimentell verifizierte TFBS aus der Literatur oder aus molekularbiologischen DB benötigen. Darüber hinaus gibt es Algorithmen, deren Strategie keine Informationen über TFBS erfordern. Diese beiden Lösungsansätze werden im Folgenden behandelt.

Eine Vorhersage, die auf *de novo* basiert, benötigt keine Informationen über experimentell verifizierte TFBS. Stattdessen erfolgt die Identifikation durch Promotorsequenzen co-regulierter Gene. Diese Strategie wird bei einigen Softwarelösungen eingesetzt, die in [TLB+05] behandelt werden. Dafür ist MEME (Multiple EM for Motif Elicitation) [BWML06] ein populäres Beispiel. Außerdem benötigen evolutionäre Algorithmen (EA) ebenfalls keine Informationen über TFBS und stellen somit eine weitere Möglichkeit dar. Ein Algorithmus, der auf EA basiert, wird in [WJ06] vorgestellt.

Allerdings benutzen zahlreiche Algorithmen experimentell verifizierte TFBS, um potenzielle TFBS in Nukleotidsequenzen vorherzusagen. Das Hidden Markov Model (HMM) wird bei einigen Algorithmen eingesetzt, wofür der Algorithmus in [EYSJ02]

ein entsprechendes Beispiel ist. Der Lösungsansatz, der im Folgenden vorgestellt wird, ist in der Praxis weit verbreitet. Damit die Variabilität der TFBS eines TF bei der Vorhersage von potenziellen TFBS berücksichtigt wird, ist eine entsprechende Repräsentation notwendig. Aufgrund von zahlreichen falsch positiven Ergebnisse in der Ergebnismenge ist die Konsensussequenz der TFBS dafür nicht geeignet [DM92]. Durch die Darstellung als *position frequency matrix* (PFM) werden alle Unterschiede zwischen den einzelnen TFBS von einem TF deutlich. Des Weiteren ist eine Konvertierung der PFM zu einer *position-specific scoring matrix* (PSSM) möglich, wodurch jede Position über spezifische Wahrscheinlichkeiten verfügt [Sto00, WS04]. Diese beiden Arten von Matrizen ermöglichen die Berücksichtigung aller Variationen einer TFBS bei der Vorhersage von potenziellen TFBS in Nukleotidsequenzen. Außerdem ist eine Visualisierung der PFM bzw. PSSM als Sequenzlogo [SS90] möglich. Der Algorithmus *ESAsearch* [BHGK06] basiert auf ESA und benutzt PSSM bei der Identifizierung. Die Softwarelösung, die während der Disseration realisiert wurde, verwendet den Algorithmus *ESAsearch* zur Identifikation von potenziellen TFBS in Nukleotidsequenzen und wird ausführlich in Kapitel 4 und 5 erläutert.

Die grundlegenden Informationen über TF und deren TFBS werden durch spezielle molekularbiologische DB wie JASPAR und TRANSFAC® [Win08] zur Verfügung gestellt. Diese beiden DB verfügen auch über Matrizen, die bei der Vorhersage von potenziellen TFBS in Nukleotidsequenzen eingesetzt werden können. Außerdem werden in einigen Publikationen die notwendigen Informationen zur Konstruktion solcher Matrizen dargestellt. Es gibt zahlreiche Softwarelösungen wie PoSSuMsearch [BSH+04], SMART [VRH10], AliBaba2 [Gra00], SIGNAL SCAN 3.0 [PS93], TESS, MatInspector, Match™ und NestedMICA [DH05], deren Vorhersage auf PFM bzw. PSSM basieren. Es gibt in der Literatur weitere Softwarelösungen, die speziell für diese Thematik realisiert wurden. Das Ziel der Software ist die Identifizierung von zahlreichen potenziellen TFBS mit möglichst wenigen falsch positiven Ergebnissen in der Ergebnismenge. Allerdings verfügen einige der Softwarelösungen über signifikante Defizite bei der Benutzerfreundlichkeit, Informationsqualität und -aktualität. Darüber hinaus ist die Bereitstellung der Funktionalität durch eine GBO nicht bei jeder Software gewährleistet. Stattdessen ist eine Bedienung nur über die Kommandozeile möglich. Mit Hilfe der gerade genannten Merkmale werden in den folgenden Abschnitten ausgewählte Softwarelösungen behandelt und deren Vor- und Nachteile beschrieben. Die ersten beiden Abschnitte 3.2.1 und 3.2.2 behandeln Match™ und MatInspector, die Beispiele für kommerzielle Software sind. Danach werden in Abschnitt 3.2.3 und 3.2.4 zwei Beispiele für OSS vorgestellt, wobei es sich um SiTaR und TESS handelt. Abschließend werden die Vor- und Nachteile der Softwarelösungen im Abschnitt 3.4 thematisiert.

Der spezifische TF *Nuclear factor kappa-light-chain-enhancer of activated B-cells* (NF-κB) ist an der Regulierung von zahlreichen Genen beteiligt. Durch die Abbildung 3.8 und die Tabelle 3.2, die das Sequenzlogo bzw. die PFM von NF-κB zeigt, wird die oben genannte Thematik der PFM bzw. PSSM beispielhaft veranschaulicht. Anhand der Abbildung 3.8 und der Tabelle 3.2 wird deutlich, dass

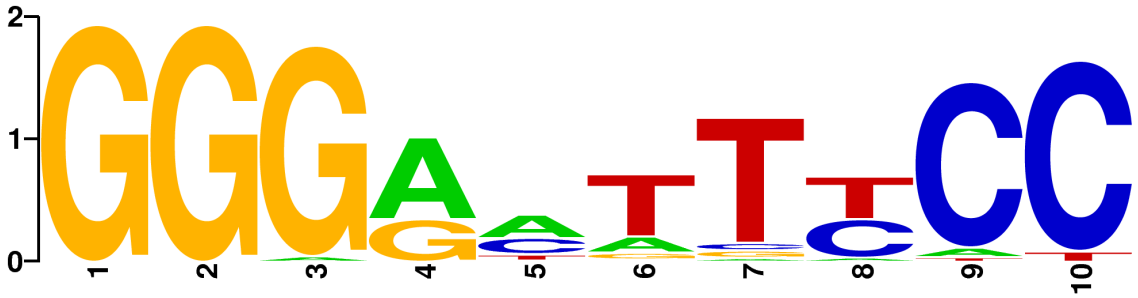


Abbildung 3.8: Darstellung der TFBS von NF-κB als Sequenzlogo. Die x-Achse repräsentiert die jeweiligen Positionen der TFBS. Auf der y-Achse wird die Höhe der Nukleotide in Bits dargestellt. Die Proportion der Nukleotide an den einzelnen Positionen ist abhängig von der Gewichtung in der jeweiligen Spalte der PFM.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	0	0	1	25	19	7	1	2	2	0
C	0	0	0	0	13	1	2	17	35	36
G	38	38	37	13	1	3	2	0	0	0
T	0	0	0	0	5	27	33	19	1	2

Tabelle 3.2: Darstellung der TFBS von NF-κB als PFM. Die PFM ermöglicht die exakte Darstellung der Gewichtung der Nukleotide an den einzelnen Positionen. Dadurch kann die Konsensussequenz 5'-GGGRNNYYCC-3' von NF-κB identifiziert werden. Diese Konsensussequenz wird als κB-Motiv bezeichnet.

für NF-κB verschiedene TFBS existieren. Ein Beispiel für die Interaktion mit unterschiedlichen TFBS durch NF-κB sind die experimentell verifizierten TFBS der Gene APOC3 und TNFRSF6. Die TFBS von APOC3 wird durch die Nukleotidsequenz 5'-GGGATTTCCC-3' repräsentiert und ist -159 bp relativ zum Transkriptionsstartpunkt (TSP) lokalisiert [GTRWL94]. Im Gegensatz dazu wird die TFBS bei TNFRSF6 durch 5'-GGGCGTTCCC-3' repräsentiert und ist -286 bp entfernt [CBOS99].

3.2.1 Match™

Das Unternehmen BioBase[5] bietet kommerzielle Softwarelösungen und verschiedene Dienstleistungen für die Lebenswissenschaften an. Dafür sind ExPlain™, PROTEOME™ und TRANSFAC® populäre Beispiele. Die Datenquelle TRANSFAC® beinhaltet hauptsächlich Informationen über eukaryotische TF, experimentell verifizierte TFBS und die entsprechenden Gene. Diese Informationen basieren auf öffentlich verfügbaren wissenschaftlichen Publikationen, sodass eine Authentizität gewährleistet ist. Die Software ExPlain™ verfügt über unterschiedli-

[5] http://www.biobase-international.com/

che Funktionalitäten zur Analyse von molekularbiologischen Daten. Dabei fungiert hauptsächlich TRANSFAC® und deren Datenbestände als Datenquelle.

Auf der Webseite von *Gene Regulation*[6] können potenzielle Kunden einer akademischen oder gemeinnützigen Organisation ausgewählte Softwarelösungen von BioBase testen. In erster Linie werden dort ältere Versionen von Anwendungen und Datenquellen bereitgestellt, die über eingeschränkte Funktionalitäten und limitierte Datenbestände verfügen und somit nicht für aktuelle molekularbiologische Fragestellungen geeignet sind. Die Tabelle 3.3 zeigt am Beispiel der Datenquelle TRANSFAC® die Unterschiede zwischen der kommerziellen und der öffentlichen Version, wobei deutlich wird, dass die kommerzielle Version von TRANSFAC® über einen wesentlich größeren Datenbestand verfügt als die öffentliche Version.

Datenbestand	**Kommerzielle Version (TRANSFAC® 2012.1)**	**Öffentliche Version (TRANSFAC® 7.0 Public 2005)**
Factors (including miRNSs)	18,211	6,133
Sites	34,742	7,915
Factor-Site Links	46,739	-
Genes	70,869	2,397
ChIP-chip Fragments	2,332,432	-
Matrices	1,665	398
References	26,676	-
Promotor Sequences	277,337	-

Tabelle 3.3: Statistik und Vergleich von TRANSFAC® 2012.1 und TRANSFAC® 7.0 Public 2005.

Ein Bestandteil bei ExPlain™ und TRANSFAC® ist Match™ [KGR+03], das eine Vorhersage von potentiellen TFBS in Nukleotidsequenzen ermöglicht. Die dafür notwendigen Datenbestände über TF, TFBS und PSSM werden durch die Datenquelle TRANSFAC® zur Verfügung gestellt. Mit Hilfe der Programmiersprache C und der Skriptsprache Perl wurde die ursprüngliche Version von Match™ implementiert. In erster Linie wurde Match™ und deren Funktionalität für die Verwendung innerhalb einer Webanwendung wie ExPlain™ konzipiert. Des Weiteren wird jeweils für Unix- und Linux-Derivate sowie Microsoft Windows eine entsprechende Version bereitgestellt, die ausschließlich über die Kommandozeile verwendet werden kann. Obwohl für verschiedene Betriebssysteme eine spezifische Version verfügbar ist, wird die Plattformunabhängigkeit nicht explizit gewährleistet. Hinsichtlich der Benutzerfreundlichkeit und Aktualität sind diese einzelnen Versionen eher kritisch

[6] http://www.gene-regulation.com/index2.html

zu bewerten, weil keine GBO verfügbar ist und nur ein beschränkter Funktionsumfang zur Verfügung gestellt wird. Es werden auf der Webseite von *Gene Regulation* neben Match^{TM} noch weitere Softwarelösungen bereitgestellt, die ebenfalls regulatorische Elemente identifizieren können. Dafür ist P-Match [CHK05] ein populäres Beispiel, welches Match^{TM} um ein *pattern matching* erweitert, das auf multiplen Sequenzalignments basiert. Auf diese Weise sollen falsch positive Ergebnisse reduziert werden. Im Gegensatz dazu verwendet Match^{TM} bei der Berechnung der Ähnlichkeit zwischen einer Nukleotidsequenz und einer PSSM nur einen *matrix similarity score* (MSS) und einen *core similarity score* (CSS). Außerdem wird für jede PSSM der Grad der Konserviertheit berechnet, der als Grundlage für den CSS dient. Die spezifische Region einer PSSM mit den fünf am stärksten konservierten und benachbarten Nukleotiden wird auch als *core region* bezeichnet. In erster Linie optimiert der CSS die Berechnung des Algorithmus, wodurch die Effizienz einer Vorhersage von potenziellen TFBS in Nukleotidsequenzen deutlich verbessert wird. Des Weiteren gibt es spezielle Strategien, die entweder falsch positive oder falsch negative Ergebnisse minimieren. Die genaue Funktionsweise dieser Strategien wird in der Literatur nicht detailliert erläutert. Im Folgenden werden ausschließlich Eigenschaften und Funktionen von Match^{TM} behandelt.

Das *Matrix Generation Tool* und das *Profile Generation Tool* sind zwei zusätzliche Softwarekomponenten von Match^{TM}, deren Funktionalität nur über eine Webanwendung wie ExPlain^{TM} verfügbar ist. Daraus ergibt sich, dass die Nutzung der Funktionen dieser beiden Softwarekomponenten über die Kommandozeile nicht möglich ist. Durch das *Matrix Generation Tool* kann der Benutzer eigene PSSM erstellen, löschen und editieren. Es werden bei der Konstruktion einer PSSM neben allgemeinen Informationen auch die Nukleotidsequenzen der TFBS benötigt. Diese Nukleotidsequenzen sind die Grundlage der PSSM und somit notwendig für die erfolgreiche Konstruktion. Die Eingabe dieser Nukleotidsequenzen erfolgt durch ein zusätzliches Eingabefeld, welches nur die Dateiformate von ClustalW und Gibbs sowie das FASTA-Format akzeptiert. Ein *profile* ist ein grundlegendes Konzept in Match^{TM} und symbolisiert eine definierte Menge von ausgewählten PSSM mit speziellen Eigenschaften. Außerdem bietet ein *profile* die Möglichkeit, für jede PSSM einen spezifischen Wert für den CSS und den MSS festzulegen. Infolgedessen kann die Vorhersage von potenziellen TFBS in Nukleotidsequenzen entweder durch ein spezifisches *profile* oder standesgemäß durch PSSM erfolgen. Es werden durch Match^{TM} einige *profiles* zur Verfügung gestellt, die auf einen individuellen molekularbiologischen Aspekt spezialisiert sind und in die Kategorien *tissue-specific*, *biological process-specific* oder *vertebrate non-redundant* aufgeteilt werden. Allerdings kann der Benutzer mit Hilfe des *Profile Generation Tool* auch eigene *profiles* erstellen, löschen und editieren. Darüber hinaus können die von Match^{TM} bereitgestellten *profiles* aus den bereits genannten Kategorien ebenfalls editiert werden. Die Konstruktion eines *profiles* erfordert neben grundlegenden Informationen das Auswählen und Hinzufügen von PSSM aus der Datenquelle TRANSFAC® und die Spezifikation der individuellen Kriterien. Sofern die Konstruktion durchgeführt wurde und die abschließende Va-

lidierung erfolgreich ist, kann der Benutzer die PSSM oder das *profile* in Match™ benutzen. Des Weiteren gewährleisten beide Softwarekomponenten einen Export der jeweiligen Metadaten in eine strukturierte Textdatei, sodass die Nutzung in anderen Softwarelösungen prinzipiell möglich ist.

Die Identifikation von potenziellen TFBS in Nukleotidsequenzen kann der Benutzer in Match™ in drei Schritten realisieren. Im ersten Schritt erfolgt die Eingabe der notwendigen Nukleotidsequenzen entweder über ein separates Eingabefeld oder durch den *Upload* einer Textdatei im FASTA-Format. Das Eingabefeld für die Nukleotidsequenzen akzeptiert nur die Dateiformate von TRANSFAC®, GenBank, IntelliGenetics (IG) sowie das FASTA-Format. Obwohl TRANSFAC® zahlreiche Datenbestände über Gene enthält, werden keine Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region von unterschiedlichen Genen und Organismen zur Verfügung gestellt. Diese Nukleotidsequenzen könnten vor allem die Konfiguration vereinfachen und so die Benutzerfreundlichkeit der Software verbessern. Zudem sind die 5'-Upstream-Region und die 3'-Downstream-Region eines Gens der Ausgangspunkt für verschiedene Interaktionen und Regulationsmechanismen und somit essentiell für die Molekularbiologie. Es können über 100000 bp analysiert werden, die entweder auf eine Nukleotidsequenz oder auf über 100 Nukleotidsequenzen verteilt sind. Sofern eine komplexe Vorhersage durchgeführt werden soll, empfiehlt BioBase Match™ die Komandozeile zu verwenden und nicht eine Webanwendung. Im darauf folgenden Schritt muss der Benutzer entweder ein spezielles *profile* oder die relevanten PSSM einzeln auswählen. Die Herangehensweise bei der Vorhersage durch den Algorithmus muss ebenfalls in diesem Schritt festgelegt werden. Abschließend kann der Benutzer die Vorhersage initiieren und sofern kein Konflikt bei den Parametern der Konfiguration besteht, wird die Vorhersage durchgeführt. Sobald die Ergebnisse verfügbar sind, kann der Benutzer zwischen zwei unterschiedlichen Darstellungen zur Visualisierung der Ergebnisse wählen. Die Ergebnisse können als strukturierte Tabelle oder in einer grafischen Darstellung repräsentiert werden. Sofern eine neue Vorhersage durchgeführt wird, werden alle temporären Ergebnisse überschrieben und sind anschließend nicht mehr verfügbar. Deswegen besteht die Möglichkeit, die Ergebnisse in eine strukturierte Textdatei zu exportieren oder persistent im Benutzerkonto zu speichern. Insbesondere bei komplexen Vorhersagen mit zahlreichen Ergebnissen werden nicht alle Ergebnisse durch die Webanwendung dargestellt. Diese Einschränkung besteht nicht, wenn Match™ über die Kommandozeile verwendet wird.

3.2.2 MatInspector

Ein weiteres Unternehmen, das Dienstleistungen, spezialisierte Hardware und kommerzielle Software für die Lebenswissenschaften zur Verfügung stellt, ist Genoma-

tix[7]. Dafür sind besonders *mygenomatix*[8], *Genomatix Mining Station* (GMS) und *Genomatix Software Suite* populäre Beispiele. Die *Genomatix Software Suite* ist zur Zeit in der Version 2.5 verfügbar und setzt sich aus unterschiedlichen Softwarekomponenten und Datenquellen zusammen. Es gibt vier Softwarekomponenten in der *Genomatix Software Suite*, welche die Identifizierung von potenziellen TFBS in Nukleotidsequenzen ermöglichen. Diese Softwarekomponenten sind Common TFs, DiAlign TF, IUPAC search und MatInspector [QFK$^+$95, CFG$^+$05]. Im Gegensatz zu den anderen Softwarekomponenten verfügt MatInspector über eine größere Anzahl von Funktionen und ist in der Literatur ein häufig aufgeführtes Beispiel für solche Softwarelösungen.

Im September 2011 wurde das derzeit aktuelle *Release* 8.0.5 von MatInspector veröffentlicht. Ursprünglich wurde MatInspector mit Hilfe der Programmiersprache C als eigenständige Software entwickelt, die Datenbestände aus den externen Datenquellen TRANSFAC® und EMBL-Bank benötigte. Inzwischen werden die *Genomatix Software Suite* und deren Softwarekomponenten als Webanwendung zur Verfügung gestellt, wodurch eine Plattformunabhängigkeit gewährleistet wird. Die notwendigen Datenbestände für die einzelnen Softwarekomponenten werden durch die zentralen Datenquellen ElDorado/Gene2Promoter und MatBase zur Verfügung gestellt, die ebenfalls Bestandteil der *Genomatix Software Suite* sind. Daraus ergibt sich, dass die direkte Abhängigkeit zwischen MatInspector und den oben genannten externen Datenquellen und deren Datenbestände nicht mehr existiert. Das aktuelle Release 5.3 von der Datenquelle ElDorado/Gene2Promoter wurde im April 2012 veröffentlicht und beinhaltet überwiegend genetische Informationen von 31 unterschiedlichen Organismen. Diese Datenquelle stützt sich hauptsächlich auf Datenbestände aus verschiedenen, öffentlich frei verfügbaren DB wie Ensembl, RefSeq und miRBase sowie Daten, die durch proprietäre Algorithmen generiert wurden. Die Datenquelle MatBase ist seit dem Juni 2011 in der Version 8.4 verfügbar und enthält ausschließlich Informationen über TF, experimentell verifizierte TFBS und PSSM. Zudem beinhaltet diese Datenquelle grundlegende Informationen über regulatorische Interaktionen zwischen TF und den entsprechenden Genen. In der Regel basieren alle Informationen aus MatBase auf wissenschaftlichen Publikationen, wodurch deren Qualität gewährleistet wird. Ein abstraktes Schema der Datenquelle MatBase wird in Abbildung 3.9 dargestellt. Im Gegensatz dazu zeigt die Tabelle 3.4 eine aktuelle Statistik der Datenquelle MatBase.

Die Softwarekomponente MatInspector verfügt über zahlreiche Parameter und Möglichkeiten bei der Konfiguration, wodurch die Verständlichkeit und vor allem die Benutzerfreundlichkeit erheblich beeinträchtigt wird. Obwohl Genomatix für die einzelnen Softwarekomponenten eine Dokumentation und einen *Support* zur Verfügung stellt, sind entsprechende Kenntnisse über die Thematik vorteilhaft. Aufgrund der umfangreichen Funktionalität werden im Folgenden ausschließlich grundlegende

[7] http://www.genomatix.de/
[8] http://www.mygenomatix.de/

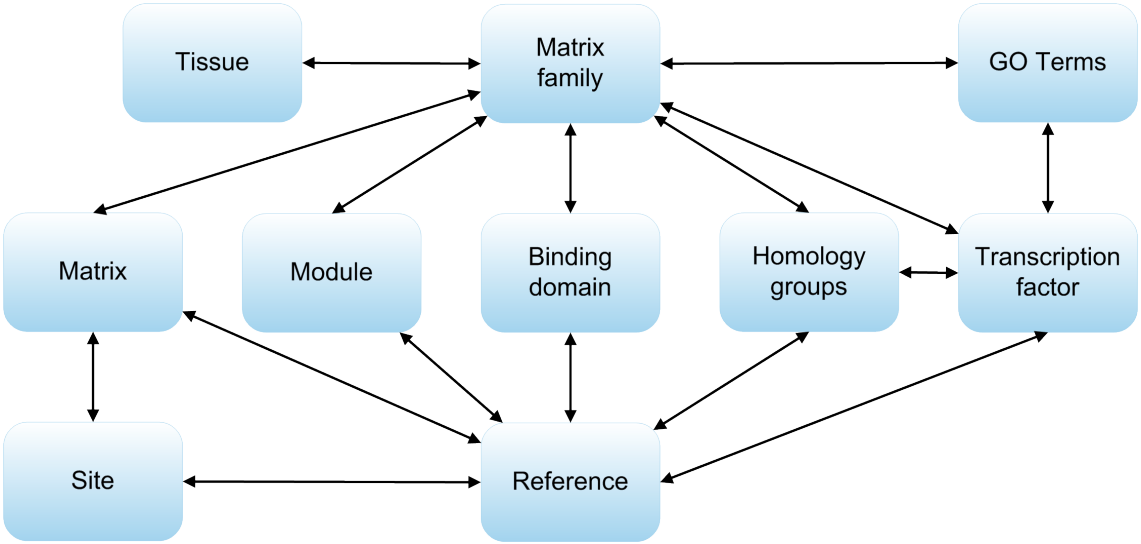


Abbildung 3.9: Schema von MatBase.

Funktionen dieser Softwarekomponente behandelt.

Datenbestand	**MatBase**	**Matrix Family Library**
Matrices	1,326	✓
Families	407	✓
Transcription factors described	21,203	-
Transcription factors associated with a matrix	8,495	✓
Functional modules	824	-
Transcription factor / Gene interactions	100,841	-
Sites	61,329	-
References	20,891	-

Tabelle 3.4: Statistik und Vergleich von MatBase 8.4 und der Matrix Family Library 8.4.

Eine Identifikation potenzieller TFBS in Nukleotidsequenzen kann durch MatInspector entweder unter Zuhilfenahme von PSSM erfolgen oder durch spezifische Sequenzmotive, die der Nukleinsäure-Nomenklatur der *International Union of Pure and Applied Chemistry* (IUPAC) entsprechen und als gewöhnliche Zeichenkette bereitgestellt werden. Daraus folgt, dass es zwei verschiedene Herangehensweisen für diese Identifikation in MatInspector gibt, die unterschiedliche Datenbestände voraussetzen. Die erste Herangehensweise basiert auf PSSM und benötigt nur bestimmte

Datenbestände aus MatBase, weshalb MatInspector die Matrix Family Library als Datenbasis verwendet. Anhand der Tabelle 3.4 wird deutlich, welche Datenbestände die Matrix Family Library aus MatBase beinhaltet und für MatInspector zur Verfügung stellt. Die Konstruktion der Matrix Family Library erfolgt mit Hilfe der Software MatInd und stützt sich auf PSSM aus wissenschaftlichen Publikationen, die überwiegend auf experimentell verifizierte TFBS basieren. Durch MatInd wird für jede PSSM die *core region* definiert, welche durch die vier am stärksten konservierten und benachbarten Nukleotide charakterisiert wird. Außerdem wird für sämtliche Positionen der PSSM ein *Ci-vector* konstruiert, der die Konservierung der Nukleotide in den einzelnen Positionen der PSSM repräsentiert und bei MatInspector berücksichtigt wird. Durch die Softwarekomponente MatDefine, die eine Erweiterung der Software MatInd und ebenfalls ein Bestandteil der *Genomatix Software Suite* ist, kann der Benutzer eigene PSSM konstruieren. Die Personal Matrix Library beinhaltet ausschließlich solche PSSM und ist auch für deren Bereitstellung innerhalb der *Genomatix Software Suite* verantwortlich. Mit Hilfe der Parameter MSS, CSS, Matrix Gruppen und Matrix-Familien kann bei MatInspector die Sensitivität der Algorithmik beeinflusst werden. Dabei ist vor allem das Konzept der Matrix-Familien hervorzuheben, weil dadurch jede Matrix über eine Matrix-Familie verfügt und so funktional ähnliche Matrizen gruppiert werden können. Im Gegensatz dazu erfolgt die Identifikation bei der zweiten Herangehensweise durch eine herkömmliche Zeichenkette, die normalerweise ein charakteristisches Sequenzmotiv repräsentiert. Die dafür bereitgestellte Datenbasis beinhaltet spezielle Sequenzmotive, die in der Regel einer IUPAC-Familie zugeordnet sind und retrovirale Primerbindungsstellen (PBS), Restriktionsschnittstellen oder TFBS von Pflanzen charakterisieren. Die Nutzung der Matrix-Familie und der IUPAC-Familie erlauben die signifikante Verdichtung und Vereinfachung der Ergebnisse.

Die Konfiguration und Bereitstellung der notwendigen Nukleotidsequenzen kann bei MatInspector durch drei verschiedene Möglichkeiten erfolgen. Diese drei Optionen und deren Charakteristika werden im Folgenden thematisiert:

1. Der Name des Gens wird als Eingabe bei der ersten Option benötigt und ermöglicht die Identifikation der entsprechenden Promotorsequenzen aus der Datenquelle ElDorado/Gene2Promoter. Das bedeutet, dass diese Variante letztendlich die verfügbaren Promotorsequenzen eines Gens zur Verfügung stellt. Eine Auflistung und Zusammenstellung durch den Benutzer ist im nächsten Schritt der Konfiguration möglich.

2. Des Weiteren ist die Eingabe der Nukleotidsequenzen durch ein zusätzliches Eingabefeld oder durch den *Upload* einer Textdatei möglich. Dabei ist zu beachten, dass die Größe der Textdatei maximal 100 Megabyte (MB) sein darf und dass beide Möglichkeiten lediglich die Dateiformate von EMBL-Bank, GCG, GCG-RSF, GenBank und IG sowie das FASTA-Format akzeptieren. Eine Alternative dazu ist die Eingabe einer oder mehrerer *accession number*, mit deren Hilfe die entsprechende Nukleotidsequenz aus den verfügbaren Da-

tenquellen identifiziert wird.

3. Eine Zusammenstellung von unterschiedlichen molekularbiologischen DB und Organismen wird durch die dritte Möglichkeit gewährleistet. Allerdings erfolgt bei dieser Variante die Vorhersage von potenziellen TFBS entweder in Promotorsequenzen oder im gesamten Genom eines Organismus. Die dafür erforderlichen Nukleotidsequenzen werden durch die Datenquellen RefSeq, EPD, GenBank und ElDorado/Gene2Promoter bereitgestellt.

Sobald die Konfiguration beendet und erfolgreich validiert wurde, erfolgt im nächsten Schritt die Initiierung und Durchführung der eigentlichen Vorhersage. Eine zusätzliche Funktion bietet die Möglichkeit einer Benachrichtung per E-Mail, die erfolgt, wenn die Vorhersage komplett durchgeführt wurde. Allerdings kann eine Vorhersage lediglich bis zu 5000 Ergebnisse bereitstellen, weil MatInspector im Hinblick auf die Anzahl der Ergebnisse einer Restriktion unterliegt und somit auch eine Vorhersage vorzeitig beendet. Das Benutzerkonto beinhaltet ein Projektmanagement, das unter anderem die Verwaltung der Prozesse und der Ergebnisse von den jeweiligen Softwarekomponenten gewährleistet. Die Ergebnisse werden normalerweise für einen Zeitraum von maximal 30 Tagen gespeichert und anschließend automatisch gelöscht. Deswegen ist der Export der Ergebnisse in die Dateiformate Microsoft Excel und Tab-Separated Values (TSV) möglich. Die Darstellung der Ergebnisse auf der Webseite erfolgt textbasiert durch eine strukturierte und dynamische Tabelle sowie interaktiv durch ein Java-Applet. Außerdem ist eine Statistik verfügbar, welche die Verteilung der Ergebnisse auf Matrix-Familien oder IUPAC-Familien veranschaulicht.

3.2.3 SiTaR

Die Software SiTaR (Site Tracking and Recognition) [FSS11] wurde für die Identifikation von potenziellen TFBS in Nukleotidsequenzen entwickelt. Allerdings können auch charakteristische Sequenzmotive in Nukleotidsequenzen identifiziert werden, die andere regulatorische Elemente repräsentieren. Im Folgenden wird die Funktionalität von SiTaR im Kontext der TFBS behandelt. Der Algorithmus von SiTaR basiert im Gegensatz zu anderen Algorithmen weder auf PSSM noch auf HMM. In einer linearen Zeitkomplexität kann der Algorithmus potenzielle TFBS in Nukleotidsequenzen identifizieren. Dabei ist die Zeitkomplexität des Algorithmus abhängig von der Anzahl der Nukleotidsequenzen und der TFBS sowie deren Länge.

Mit Hilfe der Skriptsprache PHP wurde SiTaR als Webanwendung realisiert und ist für Benutzer über das Internet frei verfügbar. Außerdem kann die Webanwendung ohne Registrierung sofort eingesetzt werden.

Eine Datenquelle, die als FunTF bezeichnet wird, fungiert bei SiTaR als Datenbasis. Diese Datenbasis beinhaltet ausschließlich Datensätze über TF und TFBS, die

in der Literatur einen Zusammenhang mit pathogenen Pilzen aufweisen. Allerdings werden durch FunTF derzeit lediglich 14 Datensätze bereitgestellt. Darüber hinaus werden keine Informationen über Gene oder relevante Nukleotidsequenzen zur Verfügung gestellt. Insbesondere die Bereitstellung der 5'-Upstream-Region und der 3'-Downstream-Region von unterschiedlichen Genen und Organismen könnte dabei von Nutzen sein. Diese limitierte und spezifische Datenquelle bietet dem Anwender geringe Möglichkeiten bei der Zusammenstellung der Elemente, die für eine Vorhersage notwendig sind.

Allerdings kann der Benutzer eigene TFBS definieren und bei der Identifizierung von potentiellen TFBS einsetzen. Dafür wurde in SiTaR ein spezielles Eingabefeld konzipiert, das eine spezifische Struktur der TFBS erfordert. Diese Struktur basiert auf dem FASTA-Format und ermöglicht so eine einfache Verwendung. Das FASTA-Format ist das standardisierte Dateiformat zur Darstellung und Speicherung von Nukleotid- und Aminosäuresequenzen in einer Textdatei. Die Länge dieser TFBS unterliegt keiner Beschränkung und kann theoretisch beliebig lang sein. Im molekularbiologischen Kontext sollte eine TFBS aber eine Länge von 6 - 12 bp aufweisen [FSS11]. Eine persistente Speicherung dieser Informationen in eine DB erfolgt nicht. Darüber hinaus benötigt die Vorhersage eine oder mehrere Nukleotidsequenzen, die der Benutzer durch ein zusätzliches Eingabefeld einfügen kann. Die Eingabe der Nukleotidsequenzen muss im FASTA-Format erfolgen. Außerdem kann der Anwender die Anzahl der erlaubten *mismatches* zwischen den TFBS und den Nukleotidsequenzen definieren. Wenn die jeweiligen Eingaben erfolgreich validiert wurden, wird die Vorhersage initiiert. Mit Hilfe der TFBS, die durch den Benutzer definiert werden müssen, und einer *scoring rule* versucht der Algorithmus von SiTaR potenzielle TFBS in den Nukleotidsequenzen zu identifizieren. Dabei berücksichtigt der Algorithmus die vom Anwender festgelegte Anzahl der *mismatches*. Sobald die Ergebnisse verfügbar sind, werden diese in einer strukturierten Tabelle dargestellt und können in die Dateiformate Comma-Separated Values (CSV), Portable Document Format (PDF) und Microsoft Excel exportiert werden. Die persistente Speicherung einer durchgeführten Vorhersage und deren notwendige Parameter in ein Profil oder in einer DB ist nicht möglich. Sofern der Benutzer die gleiche Vorhersage wiederholen oder Parameter ändern möchte, müssen die jeweiligen TFBS und Nukleotidsequenzen erneut eingegeben werden. Insbesondere das Ändern und Hinzufügen von TFBS und Nukleotidsequenzen ist bei einer umfangreichen Vorhersage im Kontext der Benutzerfreundlichkeit kritisch zu bewerten.

Prinzipiell ist mit SiTaR die Identifizierung von potenziellen TFBS im vollständigen Genom eines Organismus möglich. Diese Vorhersage sollte aber nicht mit der Webanwendung durchgeführt werden, weil diese sehr zeitintensiv und der Server in seinen Ressourcen beschränkt ist. Dafür bietet das Hans-Knöll-Institut (HKI), das SiTaR entwickelt hat, zwei Möglichkeiten. Solche Vorhersagen können direkt am HKI durchgeführt werden oder dem Benutzer wird der notwendige Quellcode bereitgestellt. Dadurch könnte der Anwender komplexe Vorhersagen auf einen eigenen Server mit ausreichenden Kapazitäten durchführen.

3.2.4 TESS

Das Transcription Element Search System (TESS) [SO97b, Sch08] wurde als Webanwendung implementiert und ermöglicht eine Vorhersage von potenziellen TFBS in Nukleotidsequenzen. Mit Hilfe der Skriptsprache Perl wurde die Webanwendung entwickelt. Diese Webanwendung ist über das Internet frei verfügbar und erfordert keine Registrierung. Darüber hinaus bietet TESS durch AnGEL (Annotation Grammar and Extraction Tool) [SMJ07] die Möglichkeit, cis-regulatorische Module in einem Genom zu identifizieren. Die letzten Erweiterungen und Aktualisierungen an TESS wurden im ersten Quartal 2007 durchgeführt. Eine Vorhersage von potenziellen TFBS wird in TESS automatisch für beide DNA-Stränge durchgeführt. Anhand der Nukleotidsequenz kann der Matrizenstrang und der kodierende Strang repliziert werden. Anschließend ist eine Identifikation von potenziellen TFBS für beide DNA-Stränge möglich. Durch die Verwendung von multiplen Sequenzalignments und PSSM realisiert der Algorithmus von TESS die Vorhersage von potenziellen TFBS in Nukleotidsequenzen.

Mit Hilfe der Datenquellen TRANSFAC®, JASPAR, Information Matrix Database (IMD) [CHS97] und CBIL-GibbsMat [SO97a] sowie deren Datenbeständen wurde eine Datenbasis für TESS realisiert. Allerdings wurde dabei eine ältere öffentlich frei verfügbare Version von TRANSFAC® eingesetzt, die im Gegensatz zu der kommerziellen Version von BioBase deutlich weniger Informationen bereitstellt. Anhand der Tabelle 3.3 in Abschnitt 3.2.1 wird dieser Unterschied zwischen der öffentlichen und der kommerziellen Version von TRANSFAC® deutlich. Eine zusätzliche Funktionalität der Webanwendung ist die Bereitstellung einer Suchfunktion unter Zuhilfenahme von ausgewählten Kriterien. Mit Hilfe der Suchfunktion kann der Anwender auf der Grundlage der Datenbasis von TESS eine wissenschaftliche Recherche durchführen. Die Ergebnisse der wissenschaftlichen Recherche werden mit detaillierten Informationen der Datensätze und den Beziehungen zu anderen Datensätzen dargestellt. Obwohl die Datenbasis einige Informationen über Gene beinhaltet, werden keine Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region von unterschiedlichen Genen und Organismen zur Verfügung gestellt. Insbesondere diese Nukleotidsequenzen könnten bei der Vorhersage von potenziellen TFBS hilfreich sein und die Konfiguration vereinfachen. Ein weiterer nicht zu unterschätzender Nachteil von TESS ist die fehlende Aktualität der Datenbestände in der Datenbasis. Infolgedessen kann eine erfolgreiche Bewerkstelligung komplexer und aktueller Fragestellungen der Molekularbiologie nicht gewährleistet werden.

Aufgrund der unterschiedlichen Datenbestände gibt es zwei verschiedene Herangehensweisen, eine Vorhersage durchzuführen. Die erste Herangehensweise, die als *string searching* bezeichnet wird, verwendet zur Identifikation keine PSSM, sondern einzelne TFBS, die als Zeichenkette bereitgestellt werden. Im Gegensatz dazu benutzt die zweite Herangehensweise das sogenannte *weight matrix searching* ausschließlich PSSM zur Identifizierung von potenziellen TFBS. Die dafür notwendigen TFBS bzw. PSSM kann der Anwender mit Hilfe der Datenbasis auswählen oder

durch ein zusätzliches Eingabefeld selbst definieren. Sofern die Voreinstellungen bei der Konfiguration einer Vorhersage benutzt werden, ist nur die Eingabe eines Titels, einer E-Mail-Adresse und einer oder mehrerer Nukleotidsequenzen notwendig. Das Eingabefeld für die Nukleotidsequenzen akzeptiert nur die Dateiformate von GenBank, EMBL-Bank und IG sowie das FASTA-Format. Außerdem gibt es eine Restriktion in der Länge der Nukleotidsequenzen. Eine Nukleotidsequenz darf nicht größer als 2000 bp sein und alle Nukleotidsequenzen zusammen müssen kleiner als 10000 bp sein. Zudem bietet die Konfiguration spezielle Einstellungen wie *factor filters*, *score filters* und *output control*. Obwohl eine Hilfe vorhanden ist, sind diese Einstellungen ohne Hintergrundwissen für die meisten Benutzer nicht praktikabel. Die Zusammenstellung der notwendigen Datensätze aus der Datenbasis ist ebenfalls nicht intuitiv bedienbar. Daraus ergibt sich, dass die Konfiguration im Kontext der Benutzerfreundlichkeit kritisch zu bewerten ist.

Sobald die Konfiguration beendet ist und alle erforderlichen Parameter der Konfiguration erfolgreich validiert wurden, ist eine Initiierung der Vorhersage möglich. Allerdings kann der verantwortliche Server nur eine definierte Anzahl von Vorhersagen gleichzeitig ausführen. Aufgrund der Limitierung kann es vorkommen, dass die Initiierung nicht möglich ist und zu einem späteren Zeitpunkt wiederholt werden muss. Wenn die Vorhersage erfolgreich initiiert oder durchgeführt wurde, erhält der Benutzer eine Benachrichtigung per E-Mail. Die Ergebnisse können dann als strukturierte Tabelle oder in einem speziellen Java-Applet dargestellt werden. Ein Export der Ergebnisse in die Dateiformate Microsoft Excel und Browser Extensible Data (BED) ist ebenso möglich wie der Versand der Ergebnisse via E-Mail. Allerdings werden durch TESS nur die 500 besten Ergebnisse zu jeder Nukleotidsequenz bereitgestellt. Des Weiteren werden die Ergebnisse sowie die notwendigen Parameter der Konfiguration von der Vorhersage für maximal einen Monat in einem Profil gespeichert. Dadurch kann der Anwender die jeweilige Vorhersage problemlos und effizient wiederholen oder einzelne Parameter der Konfiguration modifizieren.

3.3 Fazit

Die Softwarelösungen der verwandten Arbeiten aus Abschnitt 3.1 und 3.2 und deren Vor- und Nachteile werden in den folgenden zwei Abschnitten zusammengefasst und erläutert. Dabei erfolgt durch ausgewählte Kriterien auch ein Vergleich und eine abschließende Bewertung der Software (siehe Tabelle 3.5 und 3.6). Die daraus resultierenden Erkenntnisse und Schlussfolgerungen werden bei der Anforderungsanalyse und der Implementierung, die in Kapitel 4 und 5 thematisiert wird, berücksichtigt.

Als erstes erfolgt in Abschnitt 3.3.1 ein Vergleich und eine Bewertung für BioWarehouse, PiPa, CoryneRegNet und Ondex, die populäre Beispiele für die Ansätze der Datenintegration in der Bioinformatik sind. Die Vor- und Nachteile von Match™, MatInspector, SiTaR und TESS werden detailliert in Abschnitt 3.3.2 beschrieben.

Diese Anwendungen repräsentieren die zahlreichen Softwarelösungen, die potenzielle regulatorische Elemente in Nukleotidsequenzen identifizieren können.

3.3.1 Vergleich der Systeme mit Data-Warehouse-Technik

Die Systeme PiPa, CoryneRegNet und Ondex fokussieren auf ein festgelegtes DBMS, weshalb eine Migration auf andere DBMS nur durch aufwendige Änderungen der Anwendungslogik möglich ist. Im Gegensatz dazu kann bei BioWarehouse entweder MySQL oder Oracle als RDBMS verwendet werden. Damit zwischen der Anwendungsschicht und der Datenbankschicht keine Abhängigkeit besteht, kann die in Abschnitt 2.2.2.1 thematisierte Technik der objektrelationalen Abbildung eingesetzt werden. Infolgedessen könnten unterschiedliche DBMS eingesetzt werden, wodurch die Flexibilität und der Anwendungsbereich deutlich vergrößert wird.

Zwei weitere wichtige Aspekte sind die Benutzerfreundlichkeit und die Plattformunabhängigkeit einer Anwendung. Sofern keine GBO verfügbar und die Software auf ein Betriebssystem spezialisiert ist, werden diese beiden Aspekte verstärkt beeinträchtigt. Dafür ist BioWarehouse ein Beispiel, weil die Benutzung nur über die Kommandozeile möglich ist. Zudem wird als Betriebssystem ein Linux-Derivat benötigt. Im Gegensatz dazu verfügen die Systeme CoryneRegNet und Ondex diesbezüglich über keine Einschränkungen. Allerdings wird die Benutzerfreundlichkeit durch die umfangreichen Funktionen und deren Realisierung beeinträchtigt. Mit Hilfe einer Webanwendung in Kombination mit *Web 2.0* Technologien können benutzerfreundliche und plattformunabhängige Systeme implementiert werden. Darüber hinaus sind diese Systeme durch das Internet frei verfügbar und potenzielle Anwender können diese einsetzen. Aufgrund dieser Vorteile sollte die Realisierung als Webanwendung präferiert werden. Von den in Abschnitt 3.1 vorgestellten Anwendungen wurde lediglich CoryneRegNet als Webanwendung umgesetzt.

Eine zentrale Anforderung ist die Aktualität der Daten und welche molekularbiologischen Fragestellungen damit analysiert werden können. Die Aktualität ist nur bei Ondex hervorzuheben. Sofern die Aktualität einer Datenbasis nicht gewährleistet ist, können aktuelle Fragestellungen der Forschung nicht zufriedenstellend behandelt werden. Auch die Fokussierung der Datenbasis auf bestimmte molekularbiologische Themengebiete oder Organismen wie bei PiPa, CoryneRegNet und Ondex ist für eine umfangreiche Analyse nicht hilfreich. Einzig BioWarehouse ist im Hinblick auf dieses Kriterium eine Ausnahme. Deshalb sollte ein Data-Warehouse-System mit einer vielseitigen molekularbiologischen Datenbasis realisiert werden, sodass der Einsatz in unterschiedlichen Projekten möglich ist.

Die regelmäßige Pflege/Weiterentwicklung einer Anwendung ist ebenfalls ein wichtiges Kriterium. Sofern dieses Kriterium bei der Software gewährleistet ist, werden ggf. neue Funktionalitäten implementiert, Programmfehler entfernt und die Datenbasis aktualisiert, wodurch eine langfristige Verwendung der Software sichergestellt ist. In

Bezug auf dieses Kriterium ist auch die Lizenzierung der Software relevant und ob der Quellcode als *Open Source* zur Verfügung gestellt wird. In Abschnitt 3.1 wurde bereits auf diese Thematik hingewiesen. Die Systeme BioWarehouse, CoryneRegNet und Ondex sind im Hinblick auf diese Merkmale hervorzuheben. Demzufolge ist die Umsetzung unter dem Aspekt der OSS durchaus sinnvoll. Infolgedessen besteht für andere Forschungseinrichtungen die Möglichkeit, bei der Pflege/Weiterentwicklung mitzuwirken. In der Tabelle 3.5 werden ausgewählte Eigenschaften der Anwendungen BioWarehouse, PiPa, CoryneRegNet und Ondex sowie die Vor- und Nachteile dieser Systeme dargestellt.

3.3.2 Vergleich der Systeme zur Identifizierung von regulatorischen Elementen

Die regelmäßige Pflege/Weiterentwicklung der kommerziellen Softwarelösungen Match™ und MatInspector werden durch die entsprechenden Softwareunternehmen gewährleistet. Des Weiteren verfügen beide Softwarelösungen über zahlreiche Funktionalitäten, sodass verschiedene molekularbiologische Fragestellungen bewerkstelligt werden können. Allerdings ist die Benutzung von kommerziellen Softwarelösungen bei akademischen und gemeinnützigen Organisationen nicht immer die ideale Lösung und kann unter Umständen zu erheblichen Belastungen führen. Insbesondere der *Support* durch das Softwareunternehmen, die Komplexität der Software und die langfristige Finanzierung der Lizenzgebühr ist zu berücksichtigen. Deswegen werden häufig eigene Softwarelösungen entwickelt oder qualitativ hochwertige OSS eingesetzt.

Die Evaluation von Match™ und MatInspector erfolgte auf der Grundlage einer kostenlosen Testversion für akademische Zwecke, die von BioBase bzw. Genomatix bereitgestellt wurde. Dabei wurde deutlich, dass die Konfiguration der Software MatInspector äußerst kompliziert ist und das entsprechende Kenntnisse über die Thematik vorteilhaft sind. Daraus folgt, dass die Benutzerfreundlichkeit von MatInspector eher kritisch zu bewerten ist. Ein weiterer Nachteil ist die Restriktion im Hinblick auf die Anzahl der Ergebnisse, weil lediglich 5000 Ergebnisse bei MatInspector zulässig sind. Die Konfiguration bei Match™ ist strukturiert und erfordert keine besonderen Kenntnisse über das Themengebiet, weshalb die *Usability* positiv hervorzuheben ist. Außerdem werden durch das *Matrix Generation Tool* und das *Profile Generation Tool* zusätzliche Funktionalitäten zur Verfügung gestellt. Die Softwarekomponente MatInspector ist ein Bestandteil der *Genomatix Software Suite*, die als unabhängige und dynamische Webanwendung realisiert wurde und somit plattformunabhängig ist. In der Regel ist Match™ eine Softwarekomponente innerhalb einer Webanwendung wie ExPlain™ und gewährleistet auf diese Weise die Plattformunabhängigkeit. Darüber hinaus ist für verschiedene Betriebssysteme eine spezifische Software-Infrastruktur von Match™ verfügbar, deren Benutzung ausschließlich über die Kommandozeile erfolgt, weil keine GBO bereitgestellt wird. Sofern Match™ mit-

	BioWarehouse	**PiPa**	**CoryneRegNet**	**Ondex**
Integration	enge Kopplung	enge Kopplung	enge Kopplung	lose Kopplung
DBMS	MySQL, Oracle	MySQL	MySQL	Berkeley DB
Programmier-sprache(n)	Java, C	Java	PHP, Java	Java
Software-architektur	Software-Infrastruktur	Software-Infrastruktur	Webanwendung	Software-Infrastruktur
Plattform-unabhängigkeit	Nein, nur unter Linux-Derivaten.	Nein, nur unter Unix-Derivaten.	Ja	Ja
Aktualität	Manuell	Manuell	Unbekannt	Hoch
Pflege und Entwicklung	Regelmäßige Pflege und Weiterentwicklung der Software.	Unbekannt	Regelmäßige Pflege und Weiterentwicklung der Software.	Regelmäßige Pflege und Weiterentwicklung der Software.
Lizenz	MPL	Quellcode ist auf Anfrage verfügbar.	AFL	GPL
Open Source	Ja	Unbekannt	Ja	Ja

Tabelle 3.5: Vergleich zwischen BioWarehouse, PiPa, CoryneRegNet und Ondex, die auf der Data-Warehouse-Technik basieren. Anhand der für diese Arbeit aufgestellten Anforderungen werden die Vorteile in grün und die Nachteile in rot dargestellt.

tels einer Webanwendung wie ExPlain™ benutzt wird, können Probleme auftreten, weil diese Variante hinsichtlich der Anzahl der Nukleotidsequenzen und der Nukleotide einer Restriktion unterliegt. Deswegen sollen diese Vorhersagen gemäß BioBase mittels der Software-Infrastruktur über die Kommandozeile erfolgen. Allerdings ist diese Alternative im Kontext der *Usability* problematisch, weil nicht alle Funktionalitäten verfügbar sind und die Plattformunabhängigkeit ebenfalls nicht explizit gewährleistet wird. Insbesondere der umfassende Funktionsumfang, der aktuelle Datenbestand und die regelmäßige Pflege und Weiterentwicklung ist bei Match™ und MatInspector positiv hervorzuheben. Die Laufzeit der Algorithmik von Match™ und MatInspector wird in [BHGK06] bei $\mathcal{O}(mn)$ taxiert. Obwohl Match™ und MatInspector beide kommerzielle Softwarelösungen sind, verfügen beide über durchaus kritische Nachteile, die in Bezug auf die unterschiedlichen Fragestellungen aus den Lebenswissenschaften zu berücksichtigen sind.

Die Webanwendungen SiTaR und TESS sind keine kommerziellen Softwarelösungen und werden über das Internet kostenlos bereitgestellt. Durch die Realisierung als Webanwendung wird bei beiden Softwarelösungen die Plattformunabhängigkeit gewährleistet. Insbesondere die regelmäßige Pflege/Weiterentwicklung sowie die Aktualität ist bei TESS negativ zu bewerten, weil die letzte Aktualisierung im ersten Quartal 2007 durchgeführt wurde. Des Weiteren ist die Konfiguration bei einigen Parametern umständlich und erfordert entsprechende Kenntnisse über die Thematik. Infolgedessen wird die *Usability* von TESS kritisch bewertet. Die folgenden zwei Merkmale der Software TESS müssen unter gewissen Umständen ebenfalls kritisch bewertet werden. Die Anzahl der Ergebnisse ist der erste Kritikpunkt, weil zur Zeit lediglich die 500 besten Ergebnisse zu jeder Nukleotidsequenz verfügbar sind. Ein weiteres Problem ergibt sich aus der Restriktion des Servers, weil nur eine definierte Anzahl von Vorhersagen parallel durchgeführt werden kann. Darüber hinaus besteht bei TESS eine Restriktion in Bezug auf die Länge der Nukleotidsequenz, die nicht größer als 2000 bp sein darf und alle Nukleotidsequenzen zusammen müssen kleiner als 10000 bp sein. Obwohl TESS über einige interessante und notwendige Funktionalitäten verfügt, sind die bereits genannten Nachteile für ein positives Fazit in Bezug auf die Softwarequalität zu gravierend.

Die Software SiTaR ermöglicht die Identifizierung potenzieller TFBS in Nukleotidsequenzen nicht auf der Grundlage von PSSM wie bei Match™, MatInspector und TESS, sondern durch eine Strategie, die als *non-randomness* bezeichnet wird. Der Funktionsumfang und die Konfiguration von SiTaR bieten grundlegende Möglichkeiten und sind im Vergleich mit Match™, MatInspector und TESS kritisch zu bewerten. Eine detaillierte und umfassende Datenbasis wie bei den anderen Softwarelösungen aus Abschnitt 3.3.2 ist bei SiTaR ebenfalls nicht verfügbar. Die Eingabe der notwendigen Datenbestände muss überwiegend manuell durch den Benutzer erfolgen. Ein weiterer Kritikpunkt ergibt sich bei komplexen Vorhersagen, weil der Server in seinen Ressourcen eingeschränkt ist und SiTaR für solche Vorhersagen nicht konzipiert wurde. Allerdings ist die *Usability* positiv zu bewerten, weil die Webanwendung übersichtlich strukturiert ist und keine außergewöhnlichen Kenntnisse

über die Thematik erfordert. In welchem Umfang der Benutzer durch die erwähnten Nachteile von SiTaR beeinträchtigt wird, ist abhängig von der jeweiligen molekularbiologischen Fragestellung und deren Komplexität. In der Tabelle 3.6 werden ausgewählte Eigenschaften der Softwarelösungen Match™, MatInspector, SiTaR und TESS präsentiert. Die Vor- und Nachteile der jeweiligen Software werden ebenfalls in der Tabelle 3.6 dargestellt.

Es wurde deutlich, dass eine umfassende und aktuelle Datenbasis für beide Softwarelösungen zwingend erforderlich ist. Diese Datenbasis sollte auch Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region von relevanten Genen und Organismen beinhalten, weil diese der Ausgangspunkt für verschiedene Interaktionen und Regulationsmechanismen und somit bedeutend für die Molekularbiologie sind. In erster Linie ist eine solche Datenbasis die Grundlage für eine erfolgreiche Bewerkstelligung unterschiedlicher und komplexer Fragestellungen aus den Lebenswissenschaften. Der Abschnitt 3.2 zeigt ebenfalls, dass zahlreiche Softwarelösungen die Identifizierung von potenziellen TFBS in Nukleotidsequenzen auf der Grundlage von PSSM realisieren. Die Datenquellen JASPAR und TRANSFAC® beinhalten PSSM und sind somit notwendig für diese Herangehensweise. Außerdem ist die Performance der Algorithmik zu berücksichtigen, weshalb die Software eine geeignete Datenstruktur und einen effizienten Algorithmus einsetzen sollte. Eine geeignete Datenstruktur sind die ESA, die in Abschnitt 2.2.1.1 thematisiert wurden. Die performante Identifikation von Sequenzmotiven in Nukleotid- oder Aminosäuresequenzen kann mittels PSSM und ESA durch den nicht-heuristischen Algorithmus *ESAsearch* realisiert werden.

3.4 Zusammenfassung

Die relevanten verwandten Arbeiten für die Disseration, die zum einen Software aus der Bioinformatik repräsentieren, zum anderen einen spezifischen Funktionsumfang bereitstellten und unterschiedlichen Kategorien zugeordnet werden, wurden in Kapitel 3 ausführlich vorgestellt.

Als erstes wurde in Abschnitt 3.1 exemplarisch für die zahlreichen Ansätze der Datenintegration in der Bioinformatik die Funktionalität von BioWarehouse, PiPa, CoryneRegNet und Ondex erläutert. Diese Softwarelösungen sind klassische Lösungsmöglichkeiten für die facettenreiche Problematik der Datenintegration in der Bioinformatik. Der darauffolgende Abschnitt 3.2 behandelte den speziellen Funktionsumfang von Match™, MatInspector, SiTaR und TESS, die potenzielle regulatorische Elemente in Nukleotidsequenzen identifizieren können und populäre Beispiele für solche Softwarelösungen sind. Abschließend wurde in Abschnitt 3.3 eine Evaluierung der verwandten Arbeiten durchgeführt, sodass die Vor- und Nachteile der jeweiligen Software deutlich wurde. Die daraus resultierenden Erkenntnisse und Erfahrungen werden in Kapitel 4 und 5 berücksichtigt.

	Match™	**MatInspector**	**SiTaR**	**TESS**
Grundlage des Algorithmus	Paarweises Sequenzalignment, PSSM	PSSM	Non-Randomness	Multiples Sequenzalignment, PSSM
Programmier-sprache(n)	C, Perl	C	PHP	Perl
Software-architektur(en)	Software-Infrastruktur, Webanwendung	Webanwendung	Webanwendung	Webanwendung
Plattform-unabhängigkeit	Ist nur bei der Webanwendung gewährleistet.	Ja	Ja	Ja
Aktualität	Hoch	Hoch	Ist bei der DB FunTF unbekannt. Ansonsten ist eine manuelle Eingabe der Datenbestände erforderlich.	Der Datenbestand wurde seit dem ersten Quartal 2007 nicht mehr aktualisiert.
Pflege und Entwicklung	Regelmäßige Pflege und Weiterentwicklung der Software.	Regelmäßige Pflege und Weiterentwicklung der Software.	Unbekannt	Seit dem ersten Quartal 2007 keine Pflege und Weiterentwicklung der Software.
Lizenz	Kommerzielle Software	Kommerzielle Software	Quellcode ist auf Anfrage verfügbar.	Unbekannt
Open Source	Nein	Nein	Unbekannt	Unbekannt

Tabelle 3.6: Vergleich zwischen Match™, MatInspector, SiTaR und TESS, die eine Identifizierung von regulatorischen Elementen ermöglichen. Anhand der für diese Arbeit aufgestellten Anforderungen werden die Vorteile in grün und die Nachteile in rot dargestellt.

Das nächste Kapitel behandelt die Anforderungsanalyse und die Systemarchitektur der beiden Softwarelösungen, die während der Dissertation konzipiert und implementiert wurden. Dabei werden funktionale und nicht-funktionale Anforderungen der Software festgelegt als auch spezifische Anwendungsfalldiagramme konstruiert, die einen abstrakten Überblick über die Funktionalität einer Software grafisch veranschaulichen.

4 Anforderungsanalyse und Systemarchitektur

Im Rahmen der vorliegenden Arbeit wurden zwei Anzätze konzipiert und implementiert, deren Schwerpunkt verschiedene Fragestellungen aus den Lebenswissenschaften sind. Die daraus resultierenden Softwarelösungen sind zum einen ein webbasiertes Data-Warehouse-System für molekularbiologische Daten, welches durch das Akronym DAWIS-M.D. (*Data Warehouse Information System for Metabolic Data*) [Hip09, HKT+10] repräsentiert wird, zum anderen ein webbasiertes IS, das als TraBi (*Transcription Factor Binding Site Prediction*) bezeichnet wird. Durch TraBi ist die computergestützte Identifizierung von potenziellen TFBS in Nukleotidsequenzen möglich. Die erforderlichen Anforderungen und die Systemarchitekturen von DAWIS-M.D. und TraBi werden in den folgenden Abschnitten thematisiert. Insbesondere bei der Anforderungsanalyse und dem Softwareentwurf wurden die Erkenntnisse der Evaluierung der verwandten Arbeiten, die in Kapitel 3 durchgeführte wurde, als auch deren Vor- und Nachteile berücksichtigt. Die grundlegenden Kenntnisse für dieses Kapitel im Hinblick auf die Informatik wurden in Abschnitt 2.2 erläutert.

Die unterschiedlichen Anforderungen an eine Software werden hinsichtlich der Softwaretechnik in funktionale und nicht-funktionale Anforderungen klassifiziert. Eine Software sollte im Bezug auf die Thematik und die Fragestellung bestimmte Anforderungen gewährleisten. Deswegen ist der erste Schritt bei der Softwareentwicklung die Analyse und die Spezifikation der jeweiligen Anforderungen. Als erstes werden in Abschnitt 4.1 die nicht-funktionalen Anforderungen der Software behandelt. Anschließend thematisiert der Abschnitt 4.2 die funktionalen Anforderungen, welche die signifikante Funktionalität einer Software repräsentiert. Dabei werden mit Hilfe der Modellierungssprache *Unified Modeling Language* (UML) Anwendungsfalldiagramme konstruiert. Diese Diagramme können die funktionalen Anforderungen einer Software abstrakt darstellen (siehe Anhang F.1). Darüber hinaus ist eine spezifische Anforderungsanalyse notwendig, die in Abschnitt 4.3 erfolgt und notwendige molekularbiologische DB, Algorithmen und Datenstrukturen sowie die Herausforderungen bei der Datenintegration und die Konzeption der Datenbankschemata thematisiert. Der Abschnitt 4.4 behandelt die Systemarchitekturen der jeweiligen Software, sodass deren Charakteristika und die Interaktionen zwischen den einzelnen Schichten deutlich werden. Abschließend erfolgt in Abschnitt 4.5 eine Zusammenfassung.

4.1 Nicht-funktionale Anforderungen

Eine Software verfügt im Kontext der Fragestellung und der Thematik über verschiedene Funktionalitäten, die durch die funktionalen Anforderungen repräsentiert werden. Im Gegesatz dazu behandeln die nicht-funktionalen Anforderungen mit Hilfe von relevanten Kriterien ausschließlich die Qualität der Software. Anhand von etablierten Standards wie DIN 66272 oder ISO/IEC 9126 soll die Softwarequalität in der Softwaretechnik gewährleistet werden. Dabei sind insbesondere die nicht-funktionalen Anforderungen *Functionality*, *Usability*, *Reliability*, *Performance* und *Supportability* wichtige Kriterien, die durch das Akronym FURPS gekennzeichnet werden.

Durch die Evaluierung in Kapitel 3 wurde deutlich, dass die nicht-funktionalen Anforderungen wie Benutzerfreundlichkeit, Pflege/Weiterentwicklung als auch Plattformunabhängigkeit wichtige Eigenschaften einer Software sind. Darüber hinaus gibt es weitere nicht-funktionale Anforderungen, die eine Software im Hinblick auf die Qualität sicherstellen sollte. Ein wichtiges Merkmal hinsichtlich DAWIS-M.D. und TraBi ist die Softwarequalität, die im Folgenden mittels der nicht-funktionalen Anforderungen erläutert wird.

- Die **Benutzerfreundlichkeit** ist eines der wichtigsten Kriterien einer Software, weil es häufig ein erstes Indiz für die Akzeptanz und den Erfolg der Software ist. Deswegen ist es notwendig, dass DAWIS-M.D. und TraBi über eine intuitive Navigation sowie ein strukturiertes und übersichtliches Design verfügen. Außerdem sollte die Benutzung der Software keine speziellen Qualifikationen oder detaillierte Kenntnisse über eine besondere Thematik voraussetzen. Die zusätzliche Installation von Software oder eines *Add-on* als Voraussetzung zur Nutzung der beiden Softwarelösungen sollte ebenfalls nicht notwendig sein.

- Insbesondere die Abfragen an die DB und die Analyse der molekularbiologischen Daten müssen eine entsprechende **Effizienz** gewährleisten, sodass kurze Antwortzeiten möglich sind und relativ geringe Performance benötigt wird. Deshalb sind Indizes in der DB, geeignete Datenstrukturen und effiziente Algorithmen bei der Software notwendig, die in den Abschnitten 4.3.4 und 4.3.1 ausführlich behandelt werden.

- Die **Erweiterbarkeit/Wiederverwertbarkeit** einer Software ist ebenfalls ein wichtiges Merkmal. Die Implementierung von neuen Funktionalitäten oder die Integration von weiteren molekularbiologischen DB und deren Datenbestände sollte problemlos möglich sein, weshalb beide Softwarelösungen über eine modularisierte Struktur und entsprechende Schnittstellen verfügen müssen. Damit externe Anwendungssoftware problemlos bestimmte Datenbestände oder Funktionalitäten von DAWIS-M.D. oder TraBi benutzen können, ist die Realisierung einer Programmbibliothek oder einer standardisierten Programmierschnittstelle zu berücksichtigen.

- In Kapitel 3 wurde deutlich, dass die regelmäßige **Pflege/Weiterentwicklung** von Software eine existenzielle Schwachstelle werden kann, wenn die Projektfinanzierung eingestellt wird oder die verantwortlichen Softwareentwickler/Wissenschaftler anderweitig tätig sind. Infolgedessen ist es sinnvoll, DAWIS-M.D. und TraBi als frei verfügbare OSS bereitzustellen und den Quelltext über SourceForge[1] oder Google Code[2] zur Verfügung zu stellen.

- Es ist ebenfalls erforderlich, dass die **Plattformunabhängigkeit** bei beiden Softwarelösungen gewährleistet wird. Deswegen sollte die Realisierung als dynamische und interaktive Webanwendung erfolgen. Dabei ist zu beachten, dass etablierte Webbrowser wie Google Chrome, Opera, der Internet Explorer oder der Mozilla Firefox einwandfrei unterstützt werden. Darüber hinaus sollte keine Abhängigkeit zwischen der Anwendungsschicht und der Datenbankschicht bestehen, sodass eine Migration auf andere DBMS problemlos möglich ist. Durch die Objektrelationale Abbildung, die in Abschnitt 2.2.2.1 erläutert wurde, kann diese Abhängigkeit zwischen den beiden Schichten beseitigt werden. In Abschnitt 4.4 wird diese Abstraktion anhand der einzelnen Systemarchitekturen deutlich.

- Insbesondere wenn zahlreiche Benutzer eine Software gleichzeitig benutzen, müssen akzeptable Antwortzeiten und die Systemfunktionalität garantiert werden. Deswegen sollte schon während der Anforderungsanalyse die **Skalierbarkeit** der Software berücksichtigt werden, sodass eine vertikale und horizontale Skalierung problemlos möglich ist. Eine weitere Möglichkeit zur Optimierung der Effektivität der Software ist die Lastverteilung. Auf die Weise können komplexe Analysen, die auf molekularbiologischen Daten basieren, auf mehreren Prozessoren oder auf mehreren parallel arbeitenden Systemen verteilt werden.

- Ein Kernpunkt bei der Datenintegration ist die vollständige **Transparenz**, sodass ein integriertes IS die Eigenschaften eines lokalen, homogenen und konsistenten IS besitzt [LN07]. Das bedeutet, dass der Benutzer keine Kenntnisse über die physische Speicherung der Daten, die Struktur der DB und die Abfragesprache benötigt, um die Software zu benutzen. Allerdings müssen die Datenquellen von den jeweiligen Datenbeständen erkennbar sein, weshalb die Transparenz bei diesen Aspekt etwas relativiert werden sollte.

- In erster Linie sind Wissenschaftler aus den Lebenswissenschaften die **Zielgruppe** von DAWIS-M.D. und TraBi. Deshalb ist es wichtig, beide Softwarelösungen auf deren Fähigkeiten und spezifischen Forschungsinteressen hin zu optimieren. Außerdem sollte diese Klientel frühzeitig bei der Konzeption und der Implementierung der Software involviert werden.

Die bereits thematisierten nicht-funktionalen Anforderungen sind grundlegende

[1] http://sourceforge.net/
[2] http://code.google.com/

Aspekte, welche bei der Implementierung von DAWIS-M.D. und TraBi unbedingt berücksichtigt werden müssen.

4.2 Funktionale Anforderungen

Im Gegensatz zu den nicht-funktionalen Anforderungen definieren die funktionalen Anforderungen die erforderliche Funktionalität einer Software im Bezug auf die Fachdomäne und die spezifische Problemstellung. Als erstes wird die Benutzerverwaltung thematisiert, die eine wichtige Funktionalität bereitstellt. Diese funktionale Anforderung wird sowohl bei DAWIS-M.D. als auch bei TraBi benötigt. Danach werden spezielle funktionale Anforderungen der beiden Softwarelösungen erläutert, die im Hinblick auf das jeweilige Themengebiet notwendig oder wünschenswert sind. Mit Hilfe der Erkenntnisse aus Kapitel 3 wird die folgende Anforderungsanalyse durchgeführt. Dabei müssen vor allem die Vor- und Nachteile der verwandten Arbeiten berücksichtigt werden. Das Ziel sind zwei Ansätze aus den Lebenswissenschaften, die über verbesserte und innovative Konzepte verfügen sollen. Die daraus resultierenden Softwarelösungen sind DAWIS-M.D. und TraBi, deren Funktionsumfang durch Anwendungsfalldiagramme abstrakt dargestellt wird. Diese Diagramme bieten eine gute Übersicht über Abhängigkeiten und Beziehungen, die zwischen den externen Akteuren und den Anwendungsfällen existieren. Die folgenden zwei Definitionen sollen beim Verständnis der Anwendungsfalldiagramme behilflich sein:

Definition 4.1. „Ein **Akteur** modelliert einen Typ oder eine Rolle, die ein externer Benutzer oder ein externes System während der Interaktion mit einem System einnimmt.“[Kec11]

Definition 4.2. „Ein **Anwendungsfall** spezifiziert eine abgeschlossene Menge von Aktionen, die von einem System bereitgestellt werden und einen erkennbaren Nutzen für einen oder mehrere Akteure erbringen.“[Kec11]

Benutzerverwaltung

Es gibt kommerzielle molekularbiologische DB wie TRANSFAC® und TRANSPATH® [KPV+06], deren Daten normalerweise nicht frei zugänglich sind und eine gültige Lizenz erfordern. Darüber hinaus sind experimentelle Daten und deren Erkenntnisse erst dann in öffentlichen molekularbiologischen DB frei verfügbar, wenn diese erfolgreich publiziert wurden. Damit solche Datenbestände bei DAWIS-M.D. und TraBi verwendet werden können, sollte eine zentrale Benutzerverwaltung entwickelt werden, sodass ausschließlich registrierte Benutzer mit entsprechenden Zugriffsrechten diese Datenbestände abfragen können. Insbesondere bei TraBi wird eine Benutzerverwaltung benötigt, weil der Funktionsumfang und bestimmte Sektionen der Software sowie benutzerspezifische Metadaten erst zur

Verfügung stehen sollen, wenn der Anwender registriert und angemeldet ist. Die zentrale Funktionalität bei TraBi ist die Identifikation von potenziellen TFBS in Nukleotidsequenzen. Durch die Benutzerverwaltung könnten die aus der Vorhersage resultierenden Ergebnisse und deren Metadaten jeweils einem registrierten Benutzer zugeordnet werden. Dadurch wird sichergestellt, dass ausschließlich registrierte und autorisierte Benutzer einen Zugriff auf die entsprechenden Datenbestände erhalten. Eine automatische Registrierung per E-Mail und eine unkomplizierte Benutzeranmeldung und -abmeldung sollte ebenfalls durch die Benutzerverwaltung bereitgestellt werden. Sofern der Anwender die notwendigen Benutzerdaten zur Benutzeranmeldung (Benutzername und/oder Passwort) vergessen hat, sollte eine Anforderung via E-Mail möglich sein. Zudem müssen spezifische Datenbanktabellen für die Benutzerverwaltung erstellt werden, sodass relevante Metadaten der Benutzer in einer zentralen DB gespeichert werden können. Allerdings sollten sensible Metadaten wie das notwendige Passwort zur Benutzeranmeldung mittels *Data Encryption Standard* (DES) verschlüsselt werden.

Es sind für beide Softwarelösungen die drei externen Akteure Administrator, Benutzer und Anonymous erforderlich, deren Zugriffsrechte und Merkmale unterschiedlich sind. Im Gegensatz zum Akteur Anonymous besitzen die Akteure Administrator und Benutzer ein Benutzerkonto, das ein Benutzerprofil und eine bestimmte Benutzerrolle beinhaltet. Dabei ist zu beachten, dass Administrator eine Spezialisierung von Benutzer ist und über zusätzliche Möglichkeiten verfügt. Der wichtigste Akteur bei der Benutzerverwaltung ist der Administrator, der für das Erstellen und Löschen der Benutzerkonten sowie das Bearbeiten der Benutzerprofile und deren Benutzerrolle zuständig ist. Der Akteur Benutzer sollte beim Benutzerprofil lediglich seine Stammdaten (Vor- und Nachname, E-Mail-Adresse, Benutzername und Passwort) ändern können, nicht aber seine Benutzerrolle.

Das Anwendungsfalldiagramm für die Benutzerverwaltung wird in der Abbildung 4.1 dargestellt. Dieses Diagramm und dessen externe Akteure, Anwendungsfälle und Beziehungen werden im weiteren Verlauf beschrieben. Die Benutzerverwaltung kann als eigenständige Softwarekomponente bezeichnet werden und repräsentiert bei DAWIS-M.D. und TraBi einen eigenen Anwendungsfall. Deswegen wurde das Anwendungsfalldiagramm der Benutzerverwaltung in Abbildung 4.1 als Subsystem modelliert. Dieses fungiert für den Anwendungsfall `Benutzerverwaltung`, der ein Bestandteil der Anwendungsfalldiagramme der Abbildungen 4.2 und 4.3 ist, als Verfeinerung. Die Anwendungsfälle `Benutzeranmeldung`, `Benutzerdaten anfordern` und `Registrierung` sind ein Bestandteil des Anwendungsfalldiagramms in Abbildung 4.1 und inkludieren den Anwendungsfall `Benutzerkonten überprüfen`, wodurch die Validierung der Benutzerdaten sichergestellt ist. Das bedeutet, dass die beiden Anwendungsfälle `Benutzeranmeldung` und `Benutzerdaten anfordern` erst ausgeführt werden, wenn der Akteur Anonymous über ein gültiges Benutzerkonto verfügt. Außerdem inkludiert der Anwendungsfall `Registrierung` den Anwendungsfall `Benutzerkonto anlegen`, sodass der Akteur Anonymous ein Benutzerkonto er-

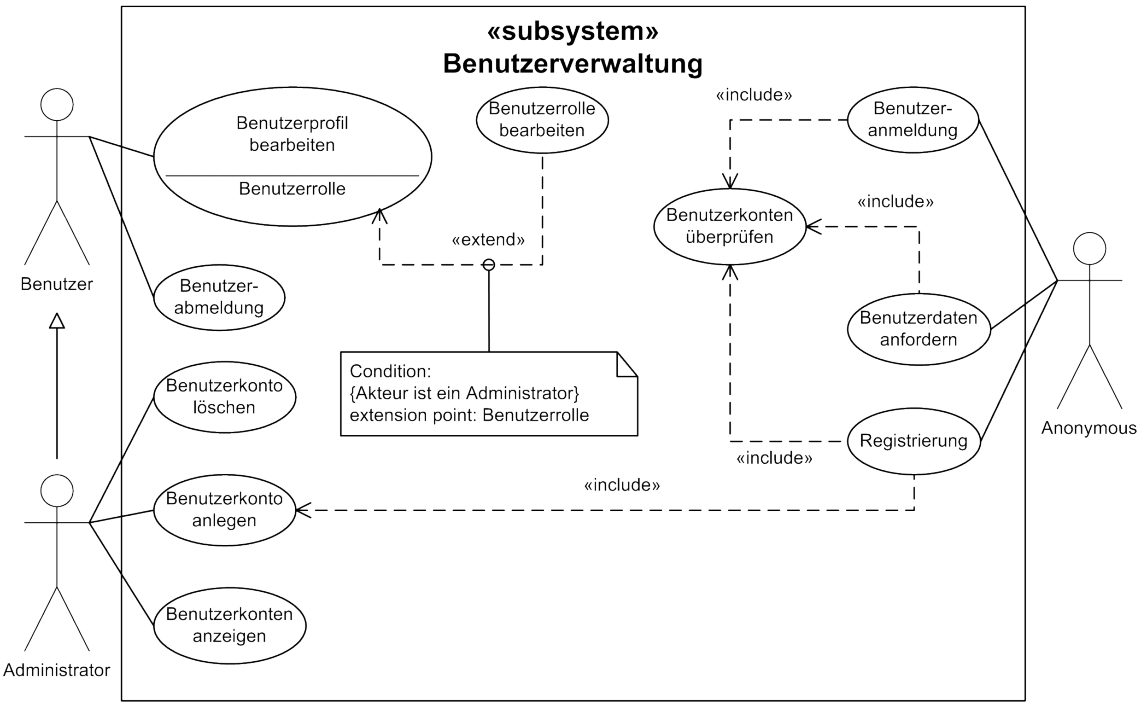


Abbildung 4.1: Anwendungsfalldiagramm für die Benutzerverwaltung.

stellen kann. Allerdings sollten nicht mehrere Benutzerkonten für ein und denselben Anwender existieren, weshalb der Anwendungsfall `Registrierung` einer Restriktion unterliegt, die der Anwendungsfall `Benutzerkonten überprüfen` sicherstellt. Durch diese Restriktion wird gewährleistet, dass jeweils ein Benutzerkonto pro Anwender existiert. Der Anwendungsfall `Benutzerrolle bearbeiten` kann in den Anwendungsfall `Benutzerprofil bearbeiten` am Erweiterungspunkt `Benutzerrolle` eingebunden werden. Diese Erweiterung ist mit einer Bedingung assoziiert, die erfüllt ist, wenn der Akteur einen Administrator repräsentiert. Daraus ergibt sich, dass ausschließlich ein Administrator die Benutzerrolle der Benutzer ändern kann. Die Benutzerverwaltung ist die einzige funktionale Anforderung, deren Funktionsumfang bei beiden Softwarelösungen benötigt wird. Infolgedessen müssen nicht-funktionale Anforderungen wie Erweiterbarkeit/Wiederverwertbarkeit bei der Realisierung der Benutzerverwaltung berücksichtigt werden, sodass eine zentrale Benutzerverwaltung bei DAWIS-M.D. und TraBi verwendet werden kann.

DAWIS-M.D. - Data Warehouse Information System for Metabolic Data

Die Zielsetzung bei DAWIS-M.D. ist die Implementierung einer webbasierten Software, die eine integrierte Sicht auf eine zugrundeliegende molekularbiologische Datenbasis bereitstellt. Die Thematik der Datenbasis wird in den Abschnitten 4.3.2, 4.3.3 und 4.3.4 thematisiert. Als nächstes werden besondere Merkmale und funktionale Anforderungen von DAWIS-M.D. definiert, wobei funktionale Anforderungen

die eigentliche Funktionalität einer Software beschreiben. In der Regel fungiert ein zentrales DBS als Datenbasis, das ein DBMS sowie eine oder mehrere DB repräsentiert. Dadurch wird eine strukturierte, effiziente und widerspruchsfreie Speicherung, Manipulation und Verwaltung der Daten gewährleistet. Allerdings sind zahlreiche DBMS verfügbar, die entweder kommerziell oder frei verfügbar sind und unterschiedliche Vor- und Nachteile aufweisen können. Deswegen sollte die Anwendungslogik der Software unabhängig vom DBMS sein, sodass verschiedene DBMS benutzt werden können. Eine mögliche Technologie, die diese Möglichkeit bietet, wurde bereits in Abschnitt 2.2.2.1 vorgestellt.

Damit die Datenbasis für aktuelle und komplexe molekularbiologische Forschungsprojekte und deren Fragestellungen geeignet ist, müssen verschiedene und zahlreiche molekularbiologischen Daten verfügbar sein. Dabei könnten mehrere kommerzielle und öffentlich frei verfügbare molekularbiologische DB als Datenquelle fungieren. Es ist aber zwingend notwendig, dass die Informationsqualität der Datenquellen berücksichtigt wird, weshalb ausschließlich Datenquellen von international etablierten Organisationen bzw. Unternehmen zu bevorzugen sind. Der eigentliche Ursprung der Datenbestände und die Zusammensetzung der Datenbasis sollten für den Anwender problemlos nachvollziehbar sein. Eine Statistik und zusätzliche Informationen über die jeweiligen molekularbiologischen DB, die als Datenquellen fungieren, könnten diesbezüglich Abhilfe schaffen. Zudem müssen benutzerspezifische, sensible und lizenzierte Datenbestände sowie Sektionen wie die System- und Benutzerverwaltung vor unbefugten Benutzern geschützt werden.

Eine umfangreiche und komplexe Datenbasis mit zahlreichen molekularbiologischen Informationen verfügt zwangsläufig über unterschiedliche Domänen zwischen denen Beziehungen und/oder Abhängigkeiten existieren. Die Domänen, Beziehungen und/oder Abhängigkeiten der molekularbiologischen Daten müssen identifiziert und als abstraktes Datenmodell abgebildet werden, das die Grundlage von DAWIS-M.D. sein sollte. Die Beziehungen und/oder Abhängigkeiten zwischen den jeweiligen Domänen können zum Beispiel PPI oder andere regulatorische Wechselwirkungen symbolisieren. Aufgrund der Relevanz müssen diese Informationen explizit hervorgehoben werden und auch als dynamisches und interaktives Netzwerk visualisiert werden, wodurch besonders (in)direkte Zusammenhänge dargestellt werden können. Außerdem sollte ein Export der Daten in standardisierte Austauschformate wie CSV, der *Extensible Markup Language* (XML), Microsoft Excel und der *Systems Biology Markup Language* (SBML) möglich sein. Insbesondere Dateien, die auf SBML basieren und biochemische Modelle repräsentieren, müssen validiert werden, weil auf diese Weise deren Konsistenz und Syntax sichergestellt wird. Es gibt in der Bioinformatik zahlreiche Softwarelösungen wie zum Beispiel spezialisierte Software zur Rekonstruktion und Visualisierung von biologischen Netzwerken, die einen Import der oben genannten Austauschformate gewährleisten. Dadurch ist es möglich, dass diese Softwarelösungen die exportierten Datenbestände ebenfalls verwenden können.

Darüber hinaus sollte eine effektive und unkomplizierte Suchfunktion zur Verfü-

gung stehen, sodass eine wissenschaftliche Recherche durchgeführt werden kann. Es müssen für die einzelnen Domänen spezifische Suchformulare bereitgestellt werden, die über eine intelligente Autovervollständigung verfügen und verschiedene Suchkriterien berücksichtigen. Das Design der Oberfläche sollte intuitiv, strukturiert und interaktiv sein, sodass die Funktionalität der Software und die Darstellung der Informationen verständlich sind. Infolgedessen müssen Nukleotid- und Aminosäuresequenzen als Textdatei zur Verfügung gestellt werden, wobei als Dateiformat das FASTA-Format zu bevorzugen ist. Die Visualisierung der PSSM als Sequenzlogo und das Bestimmen der *core region* wäre ebenfalls hilfreich, weil diese Art der Darstellung eine bessere grafische Übersicht über die Verteilung der Nukleotide bietet. Sofern Programmfehler oder fehlerhafte Datensätze existieren, sollte der Anwender den *Support* durch ein extra Formular über diese Probleme informieren können. Dadurch könnte die Qualität der Software verbessert werden.

Das Anwendungsfalldiagramm für DAWIS-M.D. wird in Abbildung 4.2 dargestellt. Dieses Diagramm und dessen externe Akteure, Anwendungsfälle und Beziehungen werden im Folgenden erläutert. In der Regel verfügen bei DAWIS-M.D. die Akteure Administrator, Benutzer und Anonymous über die selben Möglichkeiten. Dieses wird durch die Beziehungen zwischen den Akteuren und den Anwendungsfällen in der Abbildung 4.2 deutlich. Allerdings gibt es verschiedene Sonderfälle, für die bestimmte Akteure eine entsprechende Berechtigung besitzen müssen. Deswegen inkludieren die beiden Anwendungsfälle `Suche durchführen` und `Eintrag anzeigen` den Anwendungsfall `Berechtigung überprüfen`, wodurch die Berechtigung der Akteure überprüft wird. Das bedeutet, dass Akteure die nicht über eine entsprechende Berechtigung verfügen, diese beiden Anwendungsfälle nicht vollständig ausführen können und keinen Zugriff auf benutzerspezifische, sensible und lizenzierte Datenbestände erhalten. Auf diese Weise kann auch unbefugten Akteuren der Zugriff auf administrative Sektionen der Software untersagt werden.

Ein Eintrag repräsentiert einen Datensatz einer Domäne, der über Querverweise verfügen kann, die wiederum Beziehungen und/oder Abhängigkeiten zwischen den einzelnen Domänen und deren Datensätze symbolisieren. Die Metadaten der Querverweise sind die Grundlage für die Anwendungsfälle `Netzwerk anzeigen` und `Daten exportieren`. Daraus folgt, dass der Anwendungsfall `Eintrag anzeigen` am Erweiterungspunkt `Netzwerkvisualisierung` durch den Anwendungsfall `Netzwerk anzeigen` und am Erweiterungspunkt `Datenexport` durch den Anwendungsfall `Daten exportieren` erweitert werden kann. Die Erweiterungen sind jeweils mit einer Bedingung assoziiert, die am zugehörigen Erweiterungspunkt überprüft werden. Anhand der Abbildung 4.2 wird die jeweilige Erweiterungsbedingung für die Anwendungsfälle `Netzwerk anzeigen` und `Daten exportieren` deutlich.

Darüber hinaus sind beim Anwendungsfalldiagramm in Abbildung 4.2 die Anwendungsfälle `Benutzerverwaltung`, `Rezension abgeben` und `Statistik anzeigen` grundlegende Bestandteile. Dabei ist zu beachten, dass die detaillierte

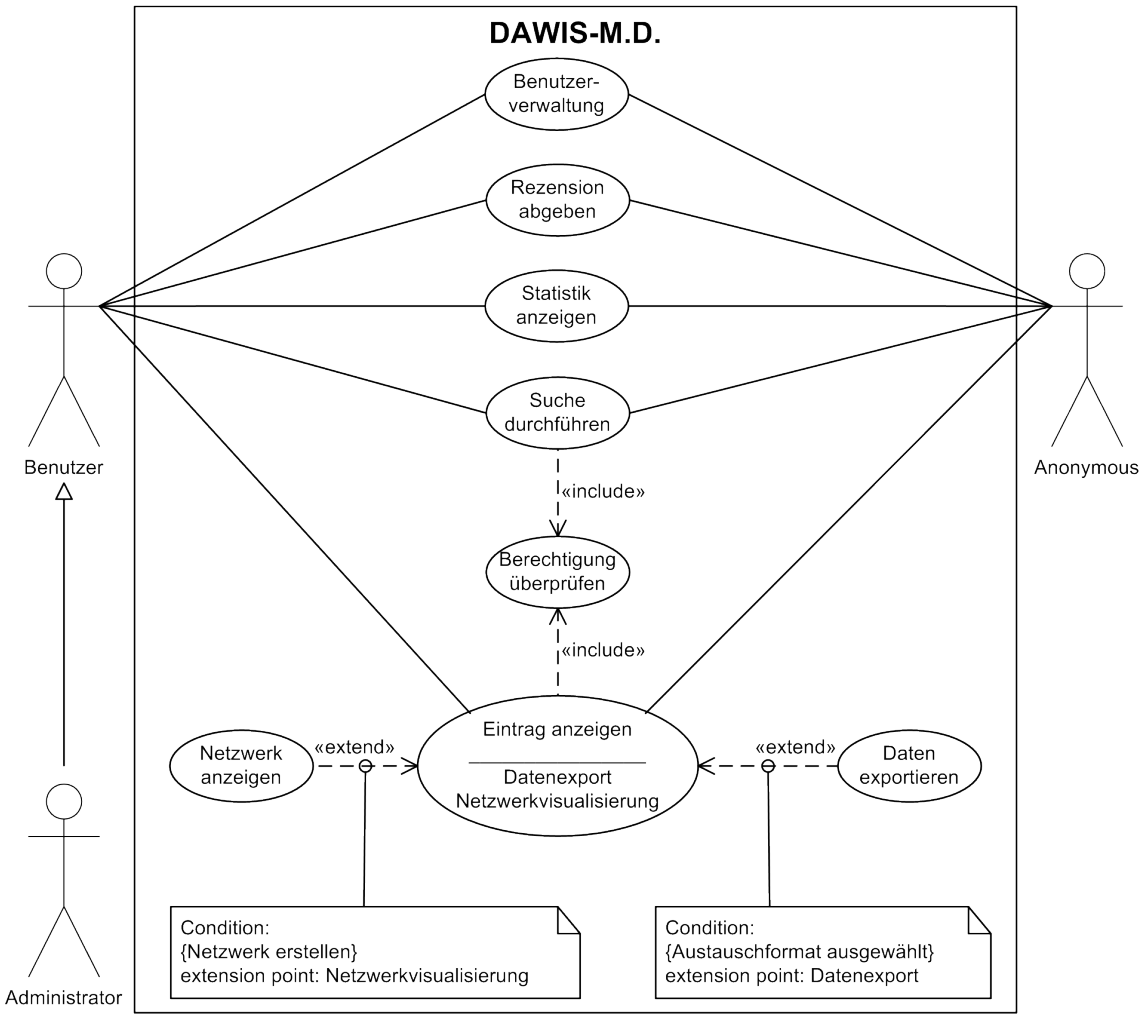


Abbildung 4.2: Anwendungsfalldiagramm für DAWIS-M.D.

Verfeinerung von Anwendungsfall `Benutzerverwaltung` in der Abbildung 4.1 dargestellt wird.

TraBi - Transcription Factor Binding Site Prediction

Das Ziel bei TraBi ist der Entwurf und die Implementierung eines webbasierten IS, das einen flexiblen Funktionsumfang zur computergestützten Vorhersage von potentiellen TFBS in Nukleotidsequenzen zur Verfügung stellt. Dafür wird eine effiziente Algorithmik und eine geeignete Datenstruktur benötigt (siehe Abschnitt 4.3.1). Außerdem sind zahlreiche Datenbestände über TF, TFBS, PSSM, Gene und Organismen erforderlich, welche die umfangreiche molekularbiologische Datenbasis von DAWIS-M.D. bereitstellen könnte. Diese Datenbasis ist das Kernstück bei DAWIS-M.D. und sollte auch für spezialisierte Softwarelösungen und deren Themengebiete wie TraBi verwendet werden. Insbesondere die Aktualisierung, Erweiterung und

Wartung einer zentralen Datenbasis ist ohne großen Aufwand möglich, sodass eine einheitliche Datenqualität und -aktualität für alle Softwarelösungen gewährleistet werden kann. Anhand der Anforderungsanalyse für DAWIS-M.D. wurde deutlich, dass ein zentrales DBS als Datenbasis fungiert und dass die Anwendungslogik der Software unabhängig vom DBMS sein sollte. Aufgrund der nicht-funktionalen Anforderungen wie Erweiterbarkeit/Wiederverwertbarkeit und Pflege/Weiterentwicklung müssen diese Kriterien auch bei TraBi berücksichtigt werden. Die Abschnitte 4.3.2, 4.3.3 und 4.3.4 beschreiben die Thematik der molekularbiologischen Datenbasis.

Die grundlegenden Eigenschaften und funktionalen Anforderungen, die den Funktionsumfang einer Software charakterisieren, werden im Folgenden für TraBi festgelegt. Die 5'-Upstream-Region und die 3'-Downstream-Region der Gene sind von besonderen Forschungsinteresse, weil die TFBS häufig in unmittelbarer Nähe der Gene vorhanden sind. Es gibt aber auch TFBS, die etliche Basenpaare von einem Gen entfernt sind. Ein entsprechendes Beispiel für beide Möglichkeiten ist in [PSD+04] dargestellt. Deswegen müssen potenzielle TFBS in Nukleotidsequenzen identifiziert werden, die jeweils eine Länge von 2500 bp, 5000 bp oder 10000 bp repräsentieren und auf der 5'-Upstream-Region oder der 3'-Downstream-Region der jeweiligen Gene eines Organismus basieren. Es sollten ausschließlich die Gene der eukaryotischen Organismen *Homo sapiens*, *Mus musculus* und *Rattus norvegicus* berücksichtigt werden. Diese festgelegten Kriterien sind die Grundlage für die Konfiguration einer Vorhersage, die durch einen dynamischen Konfigurationsassistent erfolgen könnte. Der Anwender sollte durch den Konfigurationsassistent eine Vorhersage in maximal fünf Schritten konfigurieren können. Die folgende Auflistung behandelt eine mögliche Struktur für den Konfigurationsassistent und dessen fünf Schritte:

1. Als erstes sollten grundlegende Einstellungen wie Organismus (*Homo sapiens*, *Mus musculus* oder *Rattus norvegicus*), Region (5'-Upstream-Region oder 3'-Downstream-Region) und deren Länge der Nukleotidsequenzen (2500 bp, 5000 bp oder 10000 bp) ausgewählt werden.

2. Der folgende Schritt sollte für die Zusammenstellung der relevanten Gene des zuvor ausgewählten Organismus zuständig sein. Die Übersicht der verfügbaren Gene des entsprechenden Organismus sollte strukturiert und interaktiv sein. Außerdem sollte diese Übersicht dynamische Sortier-, Filter- und Suchfunktionen bereitstellen, wobei die Suchfunktionen über eine Autovervollständigung verfügen sollten.

3. Danach sollten die relevanten TF und deren TFBS ausgewählt werden, wobei die TFBS durch eine PSSM repräsentiert werden. Die Übersicht der verfügbaren TF sollte ebenfalls strukturiert und interaktiv sein als auch eine dynamische Suchfunktion mit Autovervollständigung umfassen. Damit der Anwender einen besseren Überblick über die Verteilung der Nukleotide und die Struktur bei der PSSM erhält, ist deren Darstellung als Tabellenform und als Sequenzlogo notwendig.

4. Die Konfiguration der Algorithmik und deren Parameter sollte im vierten Schritt erfolgen. Diese Parameter werden in Abschnitt 4.3.1 erläutert. Zudem sollte der Anwender einen Namen für die Vorhersage eingeben können.

5. Abschließend sollte eine Übersicht der wichtigsten Einstellungen dargestellt werden. Dadurch kann der Anwender seine Einstellungen überprüfen und kurzfristige Änderungen durchführen.

Diese einzelnen Schritte und deren Eingaben und Zusammenstellungen müssen automatisch validiert werden. Auf diese Weise können potenzielle Fehler oder Unvollständigkeiten frühzeitig identifiziert und durch den Anwender beseitigt werden. Des Weiteren sollte der Konfigurationsassistent eine bidirektionale Navigation unterstützen und wichtige Einstellungen müssen über einen Standardwert als auch einen *Tooltip* verfügen.

Sofern die Konfiguration einer Vorhersage erfolgreich beendet wurde, sollte die Vorhersage initialisiert und durch entsprechende Schnittstellen in die Warteschlange einer eigenständigen und unabhängigen Softwarekomponente eingefügt werden. Diese Warteschlange sollte durch die Softwarekomponente kontinuierlich und automatisch abgearbeitet werden. Außerdem könnte die Softwarekomponente auf einen separaten Rechnerverbund ausgeführt werden, wodurch die Skalierbarkeit und Lastverteilung verbessert wird. Damit der jeweilige Status einer Vorhersage wie *Pending*, *Processing*, *Done* oder *Error* nachzuvollziehen ist, sollte der Anwender in regelmäßigen Zeitintervallen durch eine automatische Benachrichtigung per E-Mail informiert werden.

Ein DBS gewährleistet die strukturierte, effiziente und widerspruchsfreie Speicherung, Manipulation und Verwaltung von zahlreichen Datenbeständen. Deswegen sollte für die Ergebnisse, die aus einer Vorhersage resultieren, ein zentrales DBS eingesetzt werden. Zudem müssen die Ergebnisse in standardisierte Austauschformate wie CSV, Microsoft Excel, PDF und XML exportiert werden können. Die Ergebnisse sollten ebenfalls als strukturierte und interaktive Tabelle, die über eine Suche mit Autovervollständigung und eine Sortierung verfügt, dargestellt werden. Der Anwender sollte eine bereits durchgeführte Vorhersage problemlos wiederholen und/oder deren Einstellungen nachträglich modifizieren können. Infolgedessen ist es notwendig, dass ein Suchprofil mit einer Vorhersage assoziiert wird und deren Metadaten repräsentiert. Sobald die Vorhersage erfolgreich konfiguriert und in die Warteschlange eingefügt wurde, sollte das entsprechende Suchprofil der Vorhersage automatisch angelegt werden.

Aufgrund der in Abschnitt 3.2 genannten Vorteile sollten potenzielle TFBS in Nukleotidsequenzen mittels PSSM identifiziert werden. Es gibt verschiedene molekularbiologische DB wie JASPAR und TRANSFAC®, deren Datenbestände PSSM umfassen. Allerdings sollte ein Anwender auch benutzerspezifische PSSM anlegen können, die entweder auf wissenschaftliche Publikationen und/oder auf Laborexperimente basieren. Diese PSSM müssen persistent in das zugrundeliegende DBS ge-

speichert werden und für alle registrierte Anwender frei zugänglich sein. Außerdem sollten benutzerspezifische PSSM explizit kennzeichnet und hervorgehoben werden, sodass ein Unterscheidungsmerkmal zwischen den übrigen PSSM besteht. Das Design der Oberfläche sollte intuitiv und strukturiert sein und über interaktive und kollaborative Bestandteile verfügen. Insbesondere die spezifischen Oberflächen zur Verwaltung der Vorhersagen, der Suchprofile und der benutzerspezifischen PSSM müssen diese Charakteristika aufweisen.

Das Anwendungsfalldiagramm für TraBi wird in der Abbildung 4.3 dargestellt. Dieses Diagramm und dessen externe Akteure, Anwendungsfälle und Beziehungen werden im weiteren Verlauf erläutert. Anhand der Abbildung 4.3

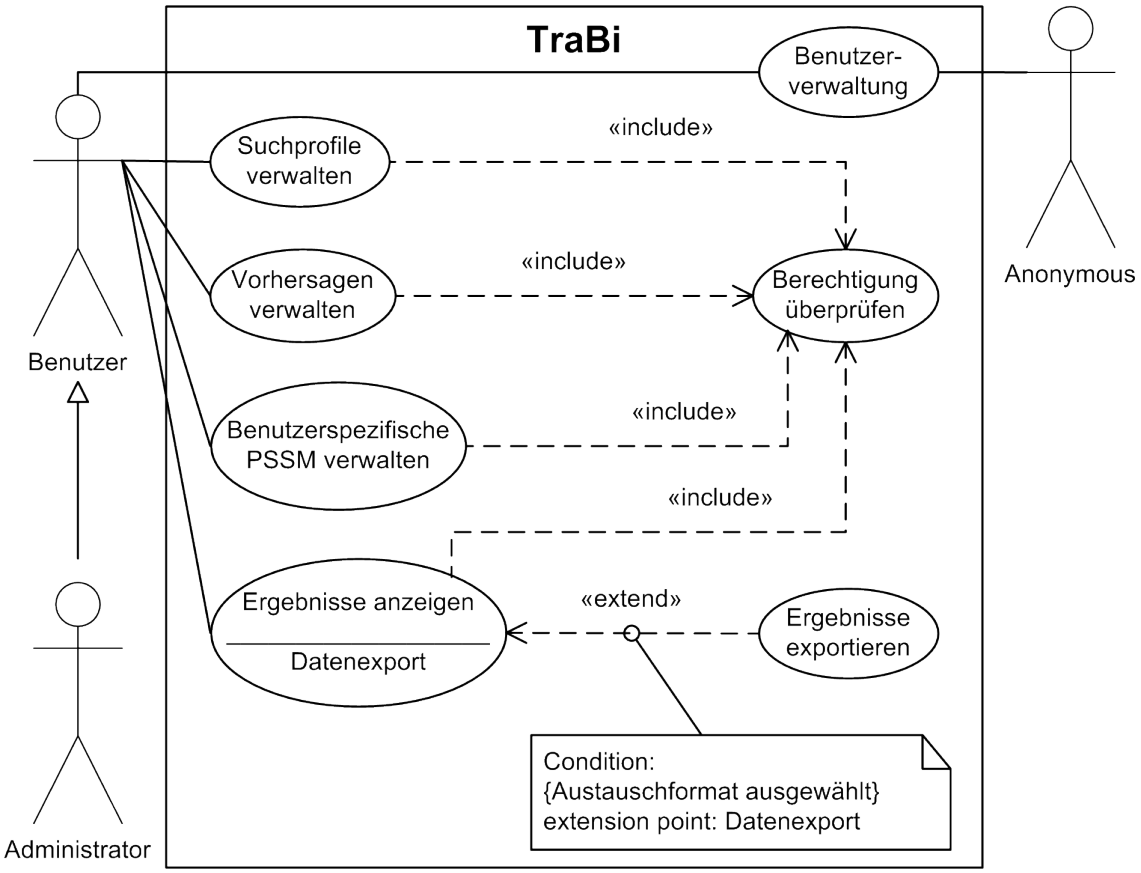


Abbildung 4.3: Anwendungsfalldiagramm für TraBi.

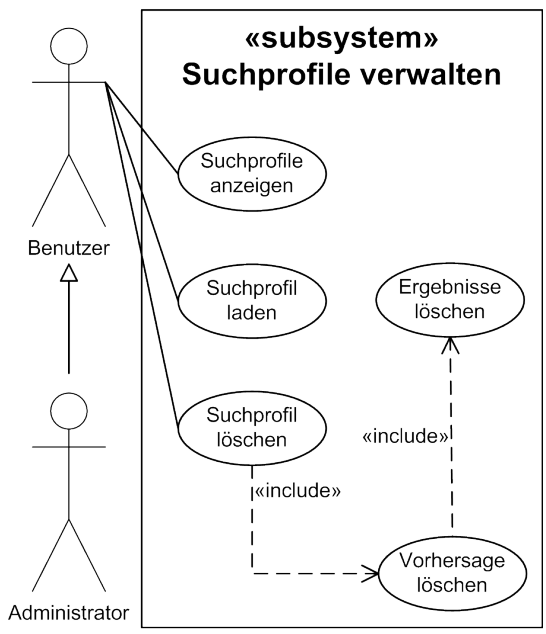
wird deutlich, dass ausschließlich die beiden externen Akteure Administrator und Benutzer den gesamten Funktionsumfang von TraBi benutzen können. Die Anwendungsfälle `Suchprofile verwalten`, `Vorhersagen verwalten`, `Benutzerspezifische PSSM verwalten` und `Ergebnisse anzeigen` inkludieren den Anwendungsfall `Berechtigung überprüfen`, wodurch die Berechtigung der Akteure überprüft wird. Auf diese Weise wird unbefugten Anwendern, die über kein gültiges Benutzerkonto verfügen oder nicht registriert/angemeldet sind, der Zugriff auf die jeweiligen Sektionen der Software und deren Funktionalität und

Datenbestände verweigert. Der Zugriff auf benutzerspezifische, sensible oder lizenzierte Datenbestände wird ebenfalls durch den Anwendungsfall `Berechtigung überprüfen` reglementiert. Daraus ergibt sich, dass Akteure entsprechende Zugriffsrechte für spezifische Datenbestände benötigen.

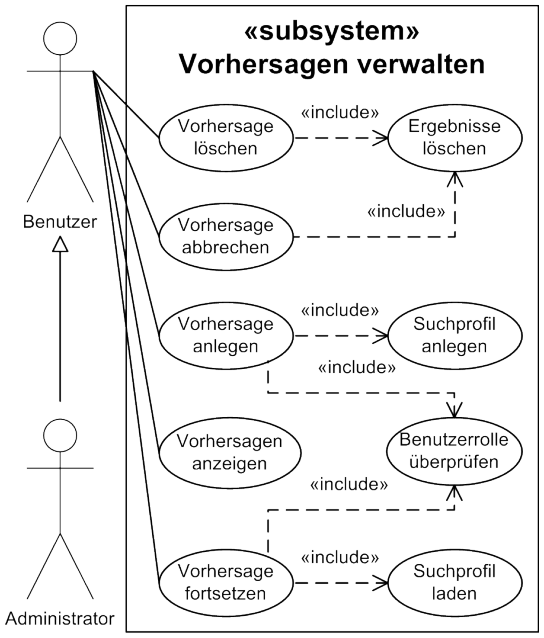
Der Anwendungsfall `Ergebnisse anzeigen` kann am Erweiterungspunkt `Datenexport` durch den Anwendungsfall `Ergebnisse exportieren` erweitert werden. Diese Erweiterung ist mit einer Bedingung assoziiert, die am zugehörigen Erweiterungspunkt überprüft wird. Die Erweiterungsbedingung ist erfüllt, wenn ein Austauschformat durch den Akteur ausgewählt wurde.

Die Abbildung 4.4 zeigt drei Anwendungsfalldiagramme, die jeweils eine Verfeinerung der Anwendungsfälle `Suchprofile verwalten`, `Vorhersagen verwalten` und `Benutzerspezifische PSSM verwalten` als Subsystem repräsentieren. Der Funktionsumfang zur Verwaltung der Suchprofile wird durch das Subsystem `Suchprofile verwalten` deutlich, das in der Abbildung 4.4(a) dargestellt wird. Der Anwendungsfall `Suchprofil löschen` inkludiert den Anwendungsfall `Vorhersage löschen`, der wiederum den Anwendungsfall `Ergebnisse löschen` inkludiert. Das bedeutet, dass ein Suchprofil und die mit dem Suchprofil assoziierte Vorhersage sowie die jeweiligen Ergebnisse der Vorhersage gelöscht werden. Die beiden Anwendungsfälle `Suchprofil laden` und `Suchprofil löschen` unterliegen einer Vorbedingung, die eine Voraussetzung für eine erfolgreiche Ausführung definiert. Damit der Akteur den Anwendungsfall `Suchprofil laden` korrekt ausführen kann, ist es notwendig, dass ein Suchprofil existiert. Im Gegensatz dazu wird der Anwendungsfall `Suchprofil löschen` vollständig ausgeführt, wenn die mit dem Suchprofil assoziierte Vorhersage nicht mehr aktiv ist und beendet wurde.

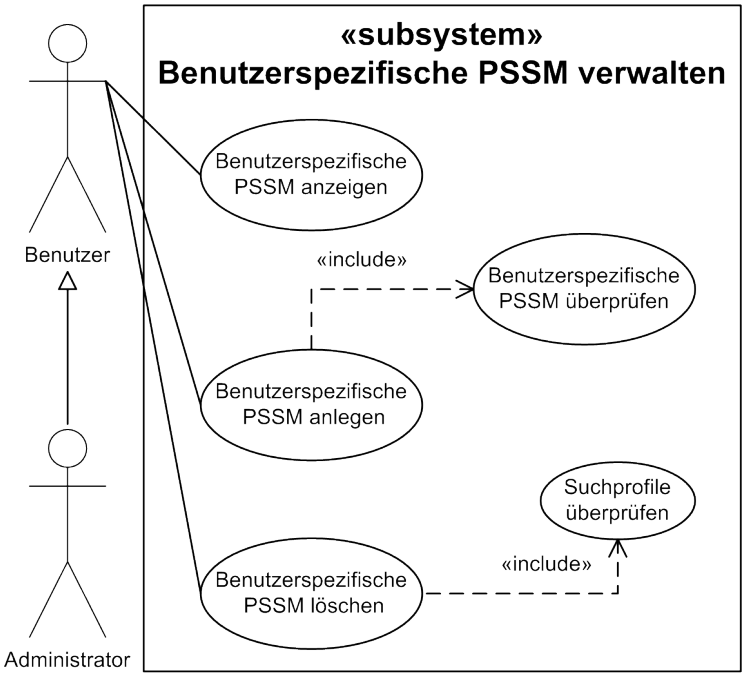
Die Abbildung 4.4(b) zeigt das Subsystem `Vorhersage verwalten`, das die wichtigste Funktionalität von TraBi veranschaulicht. Die zwei Anwendungsfälle `Vorhersage löschen` und `Vorhersage abbrechen` inkludieren den Anwendungsfall `Ergebnisse löschen`, der für beide Anwendungsfälle das Löschen der Ergebnisse durchführt. Eine Vorhersage, die nicht mehr aktiv ist und beendet wurde, kann durch den Anwendungsfall `Vorhersage löschen` gelöscht werden. Sofern eine Vorhersage noch nicht beendet wurde, kann der Anwendungsfall `Vorhersage abbrechen` eine Vorhersage vorzeitig abbrechen, die wiederum durch den Anwendungsfall `Vorhersage fortsetzen` zu einem späteren Zeitpunkt fortgesetzt werden kann. Damit der Anwendungsfall `Vorhersage fortsetzen` eine bestimmte Vorhersage fortsetzen kann, wird das entsprechende Suchprofil benötigt, weshalb der Anwendungsfall `Suchprofil laden` inkludiert wird. Ein spezifisches Suchprofil für eine Vorhersage wird automatisch angelegt, wenn die gesamte Konfiguration der Vorhersage und deren Initialisierung erfolgreich durchgeführt wurde. Diese Anforderung wird gewährleistet indem der Anwendungsfall `Vorhersage anlegen` den Anwendungsfall `Suchprofil anlegen` inkludiert. Die beiden Akteure Administrator und Benutzer können alle Funktionalitäten von TraBi benut-



(a) Anwendungsfall `Suchprofile verwalten` als Subsystem.



(b) Anwendungsfall `Vorhersagen verwalten` als Subsystem.



(c) Anwendungsfall `Benutzerspezifische PSSM verwalten` als Subsystem.

Abbildung 4.4: Verfeinerung der Anwendungsfälle `Suchprofile verwalten`, `Vorhersagen verwalten` und `Benutzerspezifische PSSM verwalten` als Subsysteme.

zen, das wird durch die beiden Anwendungsfalldiagramme in den Abbildungen 4.3 und 4.4 deutlich. Allerdings unterliegt der Akteur Benutzer im Bezug auf die Anzahl der Vorhersagen einer Restriktion, weshalb die Benutzerrolle bei den Anwendungsfällen `Vorhersage anlegen` und `Vorhersage fortsetzen` überprüft wird. Diese Restriktion sollte die Anzahl der Vorhersagen auf maximal drei pro Benutzerkonto limitieren. Aufgrund der Restriktion müssen die Anwendungsfälle `Vorhersage anlegen` und `Vorhersage fortsetzen` den Anwendungsfall `Benutzerrolle überprüfen` inkludieren.

Das Subsystem `Benutzerspezifische PSSM verwalten` in Abbildung 4.4(c) zeigt abstrakt die grundlegende Funktionalität zur Verwaltung von benutzerspezifischen PSSM bei TraBi. Der Anwendungsfall `Benutzerspezifische PSSM anlegen` inkludiert den Anwendungsfall `Benutzerspezifische PSSM überprüfen`, wodurch eine sofortige Validierung der benutzerspezifischen PSSM erfolgt. In der Regel sollten PSSM, die ein DNA-Sequenzmotiv repräsentieren, mindestens vier Positionen umfassen, weil kleinere DNA-Sequenzmotive keine besondere Relevanz für die molekularbiologische Grundlagenforschung aufweisen. Außerdem verursacht eine PSSM mit sehr wenigen Positionen eine umfangreiche Ergebnismenge, die zahlreiche falsch positive Ergebnisse beinhalten kann. Infolgedessen sollte dieses wichtige Kriterium bei der Validierung der benutzerspezifischen PSSM unbedingt berücksichtigt werden. Sofern die Validierung nicht erfolgreich ist, wird eine benutzerspezifische PSSM nicht angelegt, sodass eine verifizierte Datenqualität gewährleistet werden kann. Der Anwendungsfall `Benutzerspezifische PSSM löschen` inkludiert den Anwendungsfall `Suchprofile überprüfen` und unterliegt den folgenden Vorbedingungen, wodurch das Löschen einer benutzerspezifischen PSSM eingeschränkt wird. Das Löschen einer benutzerspezifischen PSSM kann durchgeführt werden, wenn die bestehenden Suchprofile, Vorhersagen und Ergebnismengen dadurch nicht in der Funktionsweise beeinträchtigt werden. Der registrierte Anwender kann ausschließlich die benutzerspezifischen PSSM löschen, die durch sein Benutzerkonto angelegt wurden.

4.3 Spezifische Anforderungsanalyse

Im Folgenden werden spezifische Sachverhalte der beiden Softwarelösungen im Rahmen einer speziellen Anforderungsanalyse behandelt, die im Kontext der eigentlichen Problemstellung erforderlich sind. Ein Kernpunkt bei TraBi und deren Funktionalität ist die Algorithmik und die Datenstruktur, welche in erster Linie eine performante Vorhersage von potenziellen TFBS in Nukleotidsequenzen gewährleisten soll. Der Abschnitt 4.3.1 thematisiert diese Aspekte und deren Charakteristika und erläutert weitere Möglichkeiten, um eine effiziente Vorhersage zu realisieren. Außerdem benötigten TraBi und DAWIS-M.D. eine umfangreiche Datenbasis, die verschiedene molekularbiologischen Daten beinhaltet und so die Bewerkstelligung unterschiedlicher Fragestellungen aus der molekularbiologischen Grundlagenforschung

ermöglicht. Diese Thematik und deren Herausforderungen werden in den Abschnitten 4.3.2, 4.3.3 und 4.3.4 erörtert.

4.3.1 Algorithmen und Datenstrukturen

Ein wichtiges Leistungsmerkmal einer Software, die potenzielle TFBS in Nukleotidsequenzen identifiziert, ist die Platz- und Zeitkomplexität der Algorithmik. Infolgedessen sollte die Algorithmik von TraBi eine lineare Zeitkomplexität sicherstellen, weshalb eine geeignete Datenstruktur und ein Algorithmus notwendig ist, der diese Kriterien gewährleistet. Aufgrund der genannten Vorteile in Abschnitt 2.2.1.1, sind die ESA eine geeignete Datenstruktur. Des Weiteren sollte diese Identifizierung wegen der Variabilität der TFBS auf der Grundlage von PSSM erfolgen. Allerdings sollten keine proprietären Algorithmen eingesetzt werden, weil deren Funktionsweise normalerweise nicht ausführlich dokumentiert ist. Darüber hinaus ist der Quelltext von proprietären Algorithmen nicht frei verfügbar.

Ein Algorithmus, der die erwähnten Kriterien gewährleistet, ist der nicht-heuristische Algorithmus *ESAsearch*. Sofern die Aminosäure- oder Nukleotidsequenzen nicht kürzer als $|A|^m + m - 1$ sind, ist die Zeitkomplexität des Algorithmus im *worst-case* linear [BHGK06]. Dabei repräsentiert m die Länge der PSSM und A eine endliche Menge von unterschiedlichen Symbolen. Es werden lediglich 9n Bytes benötigt, weil der Algorithmus auf ESA basiert [BHGK06]. Eine grundlegende These, die *ESAsearch* berücksichtigt, ist die Tatsache, dass Zeichenketten, die auf kleinen Alphabeten basieren, über mehrere repetitive Teilworte verfügen. Dafür sind das Alphabet $A = \{A, C\}$, die Zeichenkette $ACAACAC$ und das Teilwort ACA ein Beispiel. Das bedeutet, dass ein Sequenzmotiv, wenn es nicht innerhalb der ersten Repetition existiert, auch in keinen weiteren vorkommen wird. Ein weiteres systematisches Konzept in [BHGK06] ermöglicht die Reduzierung des Alphabets bei Aminosäuresequenzen und PSSM, wodurch eine zusätzliche Beschleunigung realisiert werden kann. Allerdings wird dieses Konzept bei der vorliegenden Arbeit nicht berücksichtigt, weil TraBi auf einen anderen Schwerpunkt fokussiert. Der Pseudocode 4.1 zeigt den Algorithmus *ESAsearch* und in Pseudocode 4.2 wird die Methode *skipchain* dargestellt, die ein Bestandteil des Algorithmus ist. Die Methode *skipchain*, die in Pseudocode 4.2 dargestellt ist, ermittelt durch die Tabelle *skp* welche Suffixe in der Tabelle suf ignoriert werden können.

```
1  depth ← 0;
2  i ← 0;
3  while i < n do
4    if n - m < suf[i] then
5      while(n - m < suf[i]) ∧ (i < n) do
6        i ← i + 1;
7        depth ← min{depth, lcp[i]};
8      end
9      if i ≥ n then return;
10   end
11   if depth = 0 then score ← 0 else score ← C[depth - 1];
12   d ← depth - 1;
13   do
14     d ← d + 1;
15     score ← score + M(d, S_suf[i] + d);
16     C[d] ← score;
17   while(d < m - 1) ∧ (score ≥ th_d);
18   if(d = m - 1) ∧ (score ≥ th) then
19     print "match at position suf[i] with score: score";
20     while i < n do
21       i ← i + 1;
22       if lcp[i] ≥ m then print "match at position suf[i] with score: score" else break;
23     end
24   else
25     i ← skipchain(lcp, skp, n, i, d);
26   end
27   depth ← lcp[i];
28 end
```

Pseudocode 4.1: Algorithmus *ESAsearch* als Pseudocode nach [BHGK06].

```
1  begin
2    if i < n then
3      j ← i + 1;
4      while(j ≤ n) ∧ (lcp[j] > d) do
5        j ← skp[i] + 1;
6      end
7    else
8      j ← n;
9    end
10   return j;
11 end
```

Pseudocode 4.2: Methode *skipchain* als Pseudocode nach [BHGK06].

Als Eingabe benötigt *ESAsearch* ein ESA für die Zeichenkette $S\$$, das die Tabellen suf, lcp und skp beinhaltet, eine PSSM M der Länge m, einen absoluten *Threshold* th und einen temporären *Threshold* th_d, wobei $0 \leq d < m$ gilt. Durch das *Lookahead scoring* wird sichergestellt, dass eine Berechnung, die den absoluten *Threshold* nicht überschreitet, frühzeitig beendet wird [WNMB00]. Dafür berechnet der Algorithmus im voraus einen temporären *Threshold*, der prophezeit, ob der absolute *Threshold* zu erreichen ist. Sofern der absolute *Threshold* nicht zu erreichen ist, wird durch die Tabelle *skp* ermittelt, wie viele Suffixe ignoriert werden können.

Dieses Problem wird nach [BHGK06, MW11] in der folgenden Definition formal dargestellt:

Definition 4.3. Wenn $max_d := max\{M(d,a)|a \in A\}$ der größte *Score* an der Position d ist und $\sigma_d := \sum_{h=d+1}^{n} max_{m-1}$ der maximal mögliche *Score* ab der Position d, dann ist $th_d = th - \sigma_d$ der temporäre *Threshold*, den eine Zeichenkette bis zur Position d erreichen muss, um den absoluten *Threshold* noch zu erreichen. Des Weiteren ist $pfxsc_d(w,M) := \sum_{h=0}^{d} M(h, w[h])$ der *Score* der Präfixe von den ersten d Zeichen einer Zeichenkette w mit einer Matrix M.

Daraus ergibt sich, dass eine Berechnung beendet werden kann, wenn $pfxcs_d(w,M) \leq th_d$ gilt. Eine ausführliche Erläuterung und Analyse des Algorithmus *ESAsearch* erfolgt in [BHGK06].

Die ESA sollten nicht zur Laufzeit der Software erstellt werden, weil auf diese Weise ein „Flaschenhals“ erzeugt wird, der die Algorithmik negativ beeinträchtigen kann. Außerdem basieren die ESA auf Datenbeständen, die aus einer molekularbiologischen DB resultieren, die im Idealfall jedes Quartal aktualisiert wird. Aufgrund der Zeitspanne der Aktualisierung der Datenquelle ist die Variabilität der Datenbestände und deren Aktualität zweitrangig, weshalb das Erstellen der ESA zur Laufzeit der Software nicht zwingend erforderlich ist. Durch eine zusätzliche Vorverarbeitung der relevanten Datenbestände während der Datenakquisition könnten die ESA angelegt und eine ergänzende Datenbereinigung und -fusion durchgeführt werden. Allerdings sollten die ESA nicht in eine DB gespeichert werden, sondern als Datei im Dateisystem. Davon würde besonders die Algorithmik der Software und deren Effizienz profitieren.

Eine weitere Möglichkeit, die Effizienz des Algorithmus *ESAsearch* zu optimieren und gleichzeitig falsch positive Ergebnisse zu reduzieren, kann durch die Berechnung der *core region* für jede PSSM realisiert werden. Diese spezifische Region repräsentiert in der PSSM die am stärksten konservierten und benachbarten Positionen, die für die molekularbiologische Grundlagenforschung von besonderen Forschungsinteresse sind. Die Tabelle 3.2 und die Abbildung 3.8 zeigen die TFBS von NF-κB als PFM bzw. Sequenzlogo. Dabei werden die Positionen 1 - 3 als *core region* bezeichnet, weil bei der jeweiligen Position die Variation zwischen den einzelnen Nukleotiden minimal ist. Das bedeutet, dass eine Position konserviert ist, wenn der größte Anteil der Wahrscheinlichkeiten auf ein Nukleotid konzentriert ist. Je höher der Grad der Konserviertheit an einer Position ist, desto größer ist die Wahrscheinlichkeit einen niedrigen *Score* zu erzielen. Damit eine *core region* der Länge eins nicht möglich ist, sollte eine minimale Länge für die *core region* definiert werden. Die *core region* einer PSSM sollte bei TraBi mindestens drei Positionen umfassen. Sofern eine *core region* über einen schlechten *score* verfügt, ist die Interaktion mit TF äußerst unwahrscheinlich. Infolgedessen werden diese Ergebnisse nicht berücksichtigt, weil deren Signifikanz für die molekularbiologische Grundlagenforschung mit großer

Wahrscheinlichkeit sehr gering ist. Eine ähnliche Strategie wird auch bei Match™ und MatInspector eingesetzt (siehe Abschnitt 3.2.1 bzw. 3.2.2). Der absolute *Score* und die Konserviertheit sind in Kombination die beiden Kriterien, welche die Relevanz einer *core region* bestimmen. Die formale Notation der Relevanz zeigt die folgende Definition:

Definition 4.4. Die Relevanz einer Position $R[i]$ ist durch $\frac{\sum_{c,d \in A} |M(i,c) - M(i,d)|}{|A| - 1}$ definiert. Dieses Kriterium ermöglicht die Herleitung der Formel zur Berechnung der *core region*. Die *core region* $[a, b]$ einer PSSM wird durch $\arg\max_{a,b} \frac{\sum_{i=a}^{b} R[i]}{b-a} \forall b > a; a, b \in [0, m-1]$ berechnet. Der akkumulative Charakter der Berechnung würde eine längere *core region* präferieren, weshalb die Division durch die Länge der möglichen *core region* $(b - a)$ notwendig ist.

Die Erweiterung des Algorithmus *ESAsearch* berechnet im voraus für jede PSSM eine *core region*, die in der weiteren Durchführung als erstes analysiert wird. Auf diese Weise wird die Performance des Algorithmus verbessert. Dafür ist besonders die Konserviertheit der *core region* verantwortlich, weil dadurch der Abbruch durch das *Lookahead scoring* in puncto Effizienz optimiert wird. Infolgedessen können durch die Methode *skipchain*, die ein Bestandteil von *ESAsearch* ist, wesentlich mehr Suffixe bei der Analyse ignoriert werden. Sofern zuerst eine niedrig konservierte Region analysiert wird, ist der Zeitraum unter bestimmten Umständen sehr viel länger bis der temporäre *Threshold* unterschritten wird. Allerdings benötigt diese Erweiterung einen zusätzlichen *Threshold*, der explizit für die *core region* definiert wird.

Darüber hinaus können Datensätze im voraus durch eine definierte Vorbedingung reduziert werden. Davon könnte auch die Algorithmik profitieren, weil die Gesamtmenge der Datensätze minimiert wird, die *ESAsearch* analysieren müsste. Sofern der Benutzer bei der Konfiguration einer Vorhersage eine minimale und maximale Länge für TFBS definiert, werden ausschließlich die Datensätze berücksichtigt, die diese Vorbedingung gewährleisten. Die Datenquellen JASPAR und TRANSFAC® verfügen über verschiedene Datensätze, die primär TF, TFBS und PSSM repräsentieren. Dabei kann die Länge der TFBS von Datensatz zu Datensatz zwischen 4 und 33 bp variieren. Allerdings wird in [FSS11] die charakteristische Länge einer TFBS mit 6 - 12 bp und in seltenen Kombinationen bis zu 18 bp beschrieben. Deswegen ist die eben erwähnte Vorbedingung zur Reduzierung der Datensätze auch hinsichtlich der Molekularbiologie sinnvoll.

4.3.2 Molekularbiologische Datenbanken

Damit verschiedene molekularbiologische Fragestellungen durch die beiden Softwarelösungen erfolgreich bewerkstelligt werden können, ist eine aktuelle und umfang-

reiche Datenbasis notwendig. Außerdem sollte diese Datenbasis nicht auf einen bestimmten Organismus fokussieren, sondern Datenbestände über prokaryotische Organismen (*Escherichia coli*) und eukaryotische Organismen (*Homo sapiens* und *Mus musculus*) bereitstellen. Es gibt zur Zeit etwa 1552 molekularbiologische DB, deren Datenbestände normalerweise für akademische Zwecke frei verfügbar sind und für eine solche Datenbasis verwendet werden können [FSRG14]. Aufgrund der zahlreichen Datenquellen ist eine Zusammenstellung von geeigneten molekularbiologischen DB erforderlich. Diese Datenquellen müssen bestimmte Kriterien gewährleisten und verschiedene Datenbestände zur Verfügung stellen. Dabei sind besonders die Aktualität, Qualität und Konsistenz der Daten, die regelmäßige Pflege/Weiterentwicklung durch etablierte internationale Organisationen bzw. Unternehmen und die Verfügbarkeit bei den einzelnen Datenquellen zu beachten. Die Beziehungen zu anderen molekularbiologischen DB oder zwischen den unterschiedlichen Datenbeständen innerhalb der molekularbiologischen DB sind ebenfalls zu berücksichtigen, weil diese für molekularbiologische Interaktionen relevant sind. Des Weiteren ist die Bereitstellung der Datenbestände durch standardisierte Austauschformate zu bevorzugen, weil proprietäre Dateiformate eine höhere Abhängigkeit verursachen und die Datenintegration unnötig erschweren. Mit Hilfe der oben genannten Kriterien und der Berücksichtigung der Anforderungen von DAWIS-M.D. und TraBi wurden 14 unterschiedliche Datenquellen identifiziert, deren Charakteristika im Folgenden vorgestellt werden:

1. Die ***Braunschweig Enzyme Database*** (BRENDA) [CSG+09] beinhaltet zahlreiche biochemische und molekularbiologische Datenbestände über Enzyme und Stoffwechselwege.

2. Die **EMBL-Bank** ist die zentrale DB für Nukleotidsequenzen in Europa. Das *European Bioinformatics Institute* (EBI) ist für die Pflege/Weiterentwicklung und deren Bereitstellung verantwortlich. Durch die *International Nucleotide Sequence Database Collaboration*[3] erfolgt täglich ein Datenabgleich zwischen GenBank, *DNA Data Bank of Japan* (DDBJ) und EMBL-Bank, wodurch die identische Datenqualität und -aktualität in allen drei DB gewährleistet wird.

3. Das EBI und das *Wellcome Trust Sanger Institute* sind beide für das Forschungsprojekt **Ensembl** [FAB+12] verantwortlich, das vor allem Software für die Bioinformatik entwickelt und Informationen über verschiedene Genome bereitstellt. Die Softwarelösungen und die Datenbestände, die durch dieses Forschungsprojekt erzeugt werden, sind für die akademische Forschung frei verfügbar. In erster Linie sind Datenbestände über unterschiedliche *Vertebraten* und Modellorganismen (*Saccharomyces cerevisiae*) verfügbar.

4. Die Datenquelle **ENZYME** enthält zahlreiche Daten über Enzyme wie den offiziellen Namen und die katalytische Reaktion. Darüber hinaus sind Querverweise zu Datenquellen wie UniProt und PROSITE [SCdC+10] verfügbar.

[3] http://www.insdc.org/

5. Durch die molekularbiologische Datenquelle **EPD** werden ausschließlich experimentell verifizierte Informationen über eukaryotische Promotoren bereitgestellt.

6. Ein internationales Konsortium ist für **GO** verantwortlich, das ursprünglich zur Vereinheitlichung des Vokabulars der Lebenswissenschaften entwickelt wurde. Die zelluläre Komponente, biologischer Prozess und molekulare Funktion sind drei kontrollierte Vokabularien, die durch GO bereitgestellt werden und die Klassifikation von Genprodukten ermöglichen.

7. Eine DB, die ausschließlich humane Datenbestände über PPI, PTM, Enzym-Substrat-Beziehungen und Querverweise zu Krankheiten zur Verfügung stellt, ist die ***Human Protein Reference Database*** (HPRD) [KPGK+09].

8. Der Schwerpunkt der Datenquelle **JASPAR** sind Informationen über TF und deren TFBS. Des Weiteren sind PSSM verfügbar, die eine Identifikation von potenziellen TFBS in Nukleotidsequenzen ermöglichen.

9. Die Datenquelle **KEGG** verfügt über strukturierte Informationen, welche Biomoleküle, Medikamente, Reaktionsgleichungen, Stoffwechselwege und Gene repräsentieren. Die Pflege/Weiterentwicklung wird durch die *Kanehisa Laboratories* an der Universität Kyoto und dem *Human Genome Center* der Universität Tokio gewährleistet. Allerdings benötigen die strukturierten *flat files* und die Textdateien im Dateiformat der XML inzwischen eine gültige und kostenpflichtige Lizenz und sind somit nicht mehr frei verfügbar.

10. Die Datenquelle ***Online Mendelian Inheritance in Man*** (OMIM) [ABSH09] enthält Datenbestände über genetische Mutationen und deren Erkrankungen beim Menschen. Dabei ist der Schwerpunkt der Zusammenhang zwischen Phenotyp und Genotyp.

11. Eine populäre DB, die Informationen hinsichtlich der hierarchischen Klassifizierung von Proteinen beinhaltet, ist die ***Structural Classification of Proteins*** (SCOP) [AHC+08].

12. Die kommerzielle Datenquelle **TRANSFAC®** wird durch BioBase zur Verfügung gestellt und beinhaltet Datenbestände über eukaryotische TF, TFBS und die entsprechenden Gene. Außerdem verfügt diese DB über PSSM, die häufig bei der Identifizierung von potenziellen TFBS in Nukleotidsequenzen verwendet werden.

13. Eine weitere kommerzielle DB, welche ebenfalls durch BioBase bereitgestellt wird, ist **TRANSPATH®**. Es sind hauptsächlich Informationen über die Signaltransduktion und über Stoffwechselwege im Hinblick auf Säugetiere verfügbar.

14. Als wichtigste DB für Aminosäuresequenzen, Proteinfunktionen und -strukturen kann **UniProt** bezeichnet werden. Die Pflege/Weiterentwicklung erfolgt durch ein internationales Konsortium.

In der Tabelle 4.1 werden die erforderlichen molekularbiologischen DB, deren Klassifikation und sofern bekannt die Informationen im Bezug auf das zur Zeit aktuelle *Release* dargestellt. Dabei erfolgte die Klassifizierung der Datenquellen auf der Grundlage der Klassifikation in Abschnitt 2.2.2.2. Aufgrund der unterschiedlichen Datenbestände sind einige molekularbiologische DB nicht eindeutig zu klassifizieren. Anhand der Tabelle 4.1 wird deutlich, dass die Datenquellen KEGG, TRANSFAC® und TRANSPATH® jeweils zwei Klassifikationen entsprechen können. Außerdem verfügen einige der Datenquellen über Restriktionen, welches ebenfalls durch Tabelle 4.1 dargestellt wird.

Eine Registrierung und eine Zustimmung beim Endbenutzer-Lizenzvertrag ist bei BRENDA, HPRD und OMIM erforderlich. Danach werden die jeweiligen Datenbestände von der entsprechenden Organisation bereitgestellt und sind für den akademischen Zweck frei verfügbar. Die kommerziellen Datenquellen TRANSFAC® und TRANSPATH® und deren Datenbestände erfordern eine gültige und kostenpflichtige Lizenz, die innerhalb des Forschungsprojekts CardioWorkBench[4] erworben wurde. Allerdings erlaubt diese Lizenz keine Verteilung und Veröffentlichung der Daten, weshalb ausschließlich Benutzer mit einer geeigneten Autorisierung einen Datenzugriff erhalten sollten. Es gibt zwar von beiden Datenquellen eine freie Version für die akademische Forschung, aber deren Datenbestände sind nicht vollständig und mittlerweile veraltet. Die Tabelle 3.3 zeigt einen Vergleich von TRANSFAC® 2012.1 und TRANSFAC® 7.0 *Public* 2005, wodurch diese Problematik deutlich wird. Infolgedessen sind diese Versionen von TRANSFAC® und TRANSPATH® keine geeigneten Alternativen. Das letzte freie *Release* 58.1 von KEGG wird in der vorliegenden Arbeit benutzt, weil die Datenbestände als XML und als strukturierte *flat file* inzwischen eine gültige und kostenpflichtige Lizenz benötigen und somit nicht mehr frei verfügbar sind. Im Gegensatz dazu werden die Datenbestände von EMBL-Bank, Ensembl, ENZYME, EPD, GO, JASPAR, SCOP und UniProt ohne Restriktionen zur Verfügung gestellt und sind ebenfalls ausschließlich für den akademischen Zweck frei verfügbar. Die Datenbestände der 14 Datenquellen werden in unterschiedlichen standardisierten Austauschformaten bereitgestellt, sodass eine unkomplizierte Weiterverarbeitung der Daten möglich ist. Die unterschiedlichen Klassifikationen von den einzelnen Datenquellen und die Querverweise zwischen den molekularbiologischen DB und den Datenbeständen sind bei der Datenintegration, die im nächsten Abschnitt behandelt wird, zu berücksichtigen.

Die beiden Softwarelösungen und die Datenbasis, die auf der Basis der Konzepte der vorliegenden Arbeit entwickelt wurden, benutzen ausschließlich molekularbiologische Daten aus den Datenquellen mit dem entsprechenden *Release* in Tabelle 4.1.

[4] http://www.cardioworkbench.eu/

Datenquelle	**Klassifikation**	***Release***	**Datum**	**Nutzungs-Restriktion**
BRENDA	*Metabolic and Signaling Pathways*	-	08.12.2011	✓
EMBL-Bank	*Nucleotide Sequence Databases*	112	Juni 2012	-
Ensembl	*Human and other Vertebrate Genomes*	67	10.05.2012	-
ENZYME	*Metabolic and Signaling Pathways*	-	13.06.2012	-
EPD	*Nucleotide Sequence Databases*	112	-	-
GO	*Genomics Databases (non-vertebrate)*	-	Juli 2012	-
HPRD	*Human and other Vertebrate Genomes*	9	-	✓
JASPAR	*Nucleotide Sequence Databases*	2009	12.10.2009	-
KEGG	*Genomics Databases (non-vertebrate)*, *Metabolic and Signaling Pathways*	58.1	01.06.2011	✓
OMIM	*Human Genes and Diseases*	-	-	✓
SCOP	*Structure Databases*	1.75	Juni 2009	-
TRANSFAC®	*Nucleotide Sequence Databases*, *Microarray Data and other Gene Expression Databases*	2009.1	27.03.2009	✓
TRANSPATH®	*Nucleotide Sequence Databases*, *Metabolic and Signaling Pathways*	-	-	✓
UniProt	*Protein sequence databases*	2012_06	13.06.2012	-

Tabelle 4.1: Klassifikation und *Release* der molekularbiologischen DB.

4.3.3 Herausforderungen bei der Datenintegration

Die Datenbestände der ausgewählten molekularbiologischen DB sind die Grundlage der Datenbasis, die für DAWIS-M.D. und TraBi erforderlich ist. Außerdem kann diese Datenbasis auch für andere Softwarelösungen eine potenzielle Datenbasis sein, weshalb die Bereitstellung von standardisierten Schnittstellen sinnvoll ist. Insbesondere die unterschiedlichen Heterogenitäten, die Autonomie und die Verteilung der einzelnen Datenquellen sind elementare Herausforderungen bei der Integration von molekularbiologischen Daten. Die Identifizierung der Beziehungen zwischen den 14 molekularbiologischen DB und deren Datenbeständen ist ebenfalls notwendig. Dadurch werden unvollständige Datensätze vervollständigt, wodurch zusätzliche Informationen verfügbar sind und Zusammenhänge deutlich werden. Des Weiteren werden diese Beziehungen bei der automatischen Rekonstruktion von PPI, metabolischen und genregulatorischen Netzwerken benötigt, die durch spezielle Anwendungssoftware wie VANESA (*Visualization and Analysis of Networks in System Biology*) [JKT+10], VANTED (*Visualization and Analysis of Networks containing Experimental Data*) [JKS06, RJH+12] oder Cytoscape [SMO+03] ermöglicht werden. Wie bereits erwähnt, verfügen einige Datenquellen über Restriktionen, die aber nicht schwerwiegend sind und den Prozess der Datenintegration nur indirekt beeinträchtigen. Dadurch ist lediglich die automatische Extraktion und Aktualisierung der Daten bei den Datenquellen BRENDA, HPRD, KEGG, OMIM, TRANSFAC® und TRANSPATH® eingeschränkt. Im Hinblick auf diese Herausforderungen bei der Datenintegration ist eine geeignete Integrationsarchitektur erforderlich. Aufgrund der Argumente in Abschnitt 2.2.3.4 und der genannten Vorteile in [GB09, LN07, LR03] wird die notwendige Datenbasis als DWH realisiert. Darüber hinaus können auf der Grundlage des DWH sogenannte Data-Marts erstellt werden, die einen spezifischen Datenbestand des DWH repräsentieren und für spezielle Anwendungssoftware sinnvoll sind. In der Regel wird eine materialisierte Integration in der Bioinformatik bevorzugt, weil eine solche Integrationsarchitektur äußerst effizient ist und für die Datenintegration häufig nur ein *Parser* benötigt wird. Zudem bietet diese Integrationsarchitektur einen vollständigen Zugriff auf die Daten und eine direkte Abhängigkeit von den einzelnen Datenquellen besteht ebenfalls nicht mehr.

Die unterschiedlichen molekularbiologischen DB sind durch explizite Querverweise und/oder Fremdschlüssel miteinander verknüpft, wofür DBGET/LinkDB [Kan97, FGM+98] ein charakterisches Beispiel ist. Die zahlreichen Beziehungen zwischen den verschiedenen molekularbiologischen DB innerhalb DBGET/LinkDB sind in der Abbildung 4.5 dargestellt. Durch die folgenden vier Maßnahmen können solche Beziehungen zwischen molekularbiologischen DB identifiziert werden:

1. In der Bioinformatik gibt es zahlreiche webbasierte IS, deren Datenbasis häufig ein zentrales DBS mit molekularbiologischen Daten ist. Durch diese IS können Beziehungen zu anderen molekularbiologischen DB ermittelt werden, indem verschiedene Einträge analysiert und ausgewertet werden. Allerdings ist diese

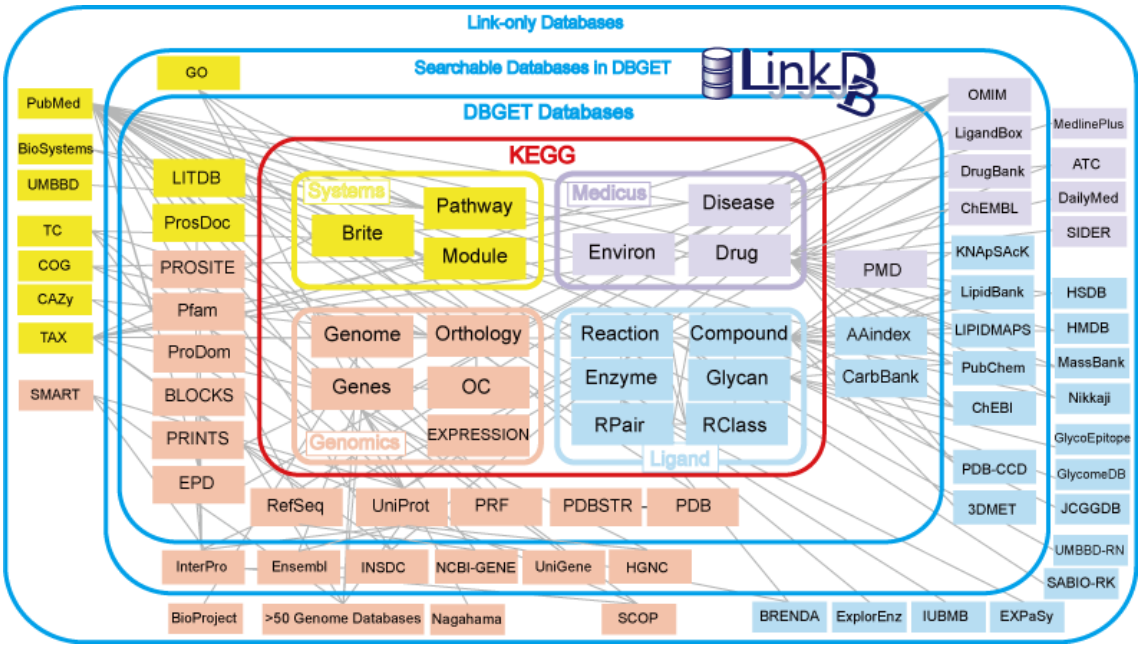


Abbildung 4.5: Beziehungen zwischen den unterschiedlichen molekularbiologischen DB innerhalb DBGET/LinkDB.

Herangehensweise sehr aufwendig und ineffizient.

2. Der Datenbestand der molekularbiologischen DB wird durch standardisierte Austauschformate wie XML oder durch strukturierte *flat files* zur Verfügung gestellt. Durch die Analyse der Textdateien ist eine Identifizierung von Querverweisen möglich. Die Struktur von diesen Textdateien wird in der Dokumentation der einzelnen Datenquellen erläutert.

3. Durch die Analyse der Schemata der jeweiligen molekularbiologischen DB können ebenfalls Querverweise identifiziert werden. Allerdings kann diese Vorgehensweise äußerst aufwendig sein, weil einige Schemata sehr komplex sind und etliche Datenbanktabellen umfassen. Durch eine eindeutige Strukturierung der Schemata kann diese Problematik reduziert werden.

4. Darüber hinaus werden die Querverweise bei einigen molekularbiologischen DB in der Dokumentation durch eine explizite Auflistung dargestellt. Sofern eine geeignete Dokumentation der Datenquelle verfügbar ist, sollte diese Strategie bevorzugt werden.

Insbesondere zusätzliche Informationen in Bezug auf die molekularbiologischen Mechanismen der Regulation und Interaktion können so identifiziert werden. Die Identifikation von Zusammenhänge zwischen den unterschiedlichen Domänen wie beispielsweise einer Krankheit und einem Enzym ist ebenfalls möglich. Ein klassisches

Beispiel dafür ist der Zusammenhang zwischen der Stoffwechselkrankheit Phenylketonurie (PKU) und dem Enzym Phenylalaninhydroxylase, das hauptsächlich für PKU verantwortlich ist. Mit Hilfe der oben genannten Maßnahmen wurden zahlreiche Beziehungen zwischen den 14 molekularbiologischen DB identifiziert (siehe Abbildung 4.6). Anhand der Abbildung 4.6 wird deutlich, dass zwischen SCOP und

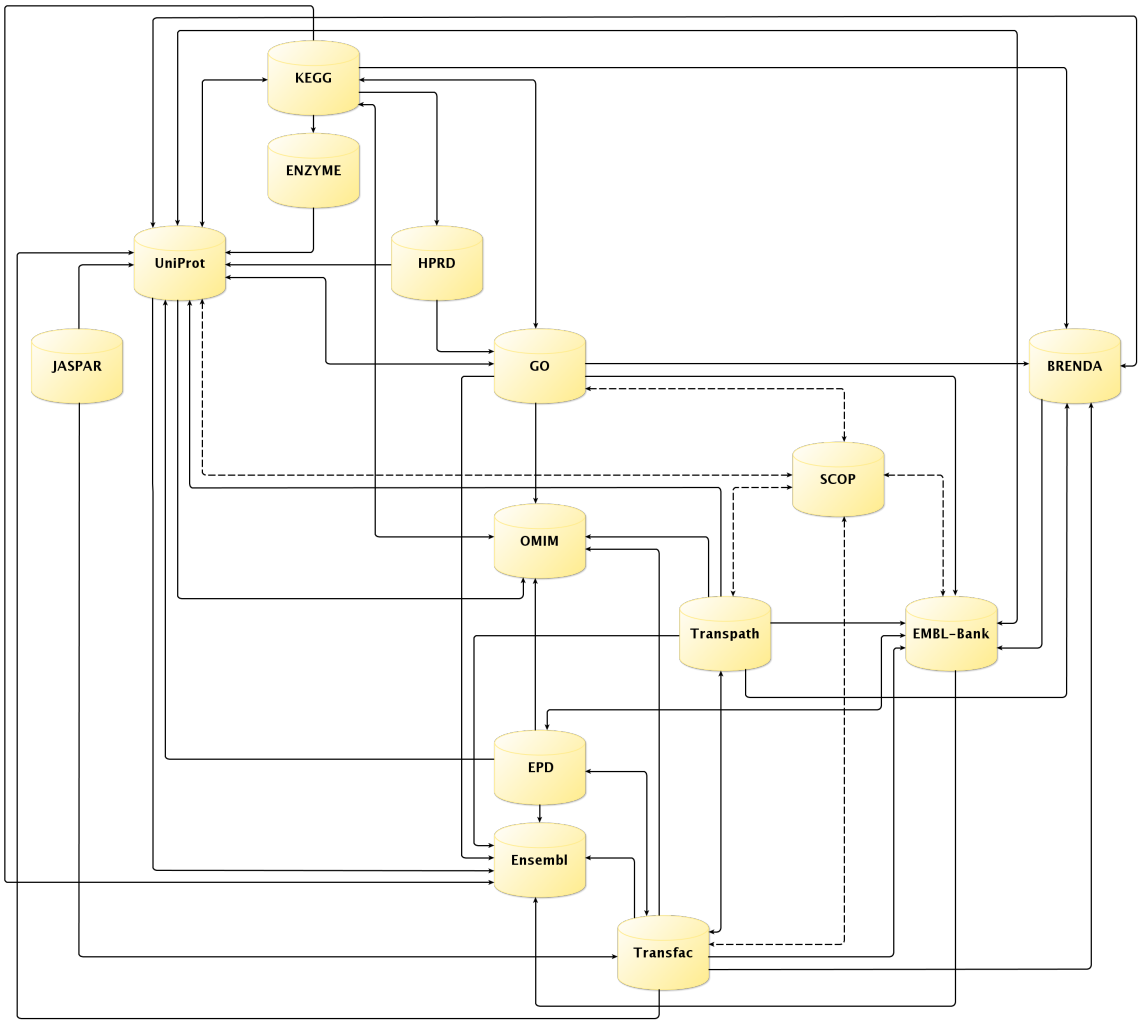


Abbildung 4.6: Direkte und indirekte Beziehungen zwischen den molekularbiologischen DB.

den anderen Datenquellen keine direkten Querverweise existieren. Allerdings basieren die Datenbestände der Datenquelle SCOP auf den Daten der *Protein Data Bank* (PDB) und verfügen auch über deren eindeutigen Identifikationsnummer. Durch diese Identifikationsnummer ist eine Identifikation von indirekten Querverweisen zwischen SCOP und anderen molekularbiologischen DB wie UniProt und EMBL-Bank möglich. In der Abbildung 4.6 werden diese Beziehungen gestrichelt dargestellt.

Die 14 molekularbiologischen DB verfügen über verschiedene Klassifikationen und vereinzelt ist eine eindeutige Klassifizierung nicht möglich. Insbesondere die unter-

schiedlichen Datenbestände sind für mehrere Klassifikationen bei einer Datenquelle verantwortlich. Dafür sind KEGG, TRANSFAC® und TRANSPATH® entsprechende Beispiele. Allerdings können durch diese Klassifikationen spezifische Domänen ermittelt werden, sodass eine Segmentierung der Datenbestände auf unterschiedliche Domänen möglich ist. Darüber hinaus können diese Domänen die Grundlage für ein abstraktes Datenmodell sein, das für DAWIS-M.D. erforderlich ist und in Abschnitt 5.2.2 behandelt wird. Insgesamt wurden 13 spezielle Domänen identifiziert, die in

Domäne	**Beschreibung**	**Datenquelle**
Compound	Chemische Verbindungen und Strukturen	KEGG, TRANSPATH®
Disease	Stoffwechselkrankheiten beim Menschen	KEGG, OMIM
Drug	Wirkstoffe	KEGG
Enzyme	Nomenklatur der Enzyme, Hierarchische Klassifizierung der strukturellen Domäne	BRENDA, ENZYME, KEGG, SCOP
Gene	Gene	EMBL-Bank, Ensembl, KEGG, TRANSFAC®, TRANSPATH®
Gene Ontology	Strukturiertes und standardisiertes Vokabular	GO
Genome	Genome	KEGG
Glycan	Strukturen der Polysaccharide	KEGG
Pathway	Stoffwechselwege	KEGG, TRANSPATH®
Protein	Proteine, Hierarchische Klassifizierung der strukturellen Domäne	HPRD, SCOP, UniProt
Reaction	Biochemische Reaktionen	KEGG, TRANSPATH®
Reactant Pair	Substrate und Produkte bei biochemischen Reaktionen	KEGG
Transcription Factor	TF, TFBS, PSSM, Promotor	EPD, JASPAR, TRANSFAC®

Tabelle 4.2: Segmentierung der Datenbestände auf die einzelnen Domänen.

Tabelle 4.2 dargestellt und erläutert werden. Mit Hilfe der Tabelle 4.2 wird deutlich, dass Datenquellen wie KEGG, TRANSFAC®, TRANSPATH® und SCOP mehreren Domänen zugeordnet wurden. Das ist erforderlich, weil KEGG, TRANSFAC® und TRANSPATH® verschiedene Datenbestände aus unterschiedlichen Kategorien der Molekularbiologie zur Verfügung stellen. Im Gegensatz dazu wird der Datenbestand der Datenquelle SCOP zwei Domänen zugeordnet, weil die Domäne *Enzyme* eine Spezialisierung der Domäne *Protein* ist und somit auch Informationen über die Klassifikation eines Proteins bereitstellen kann. Das abstrakte Datenmodell wird auf

der Grundlage der Domänen erstellt und ist in der Abbildung 4.7 dargestellt. Die Beziehungen zwischen den einzelnen Domänen ergeben sich aus der Segmentierung der Datenbestände. Anhand der Abbildung 4.7 wird deutlich, dass die Domänen

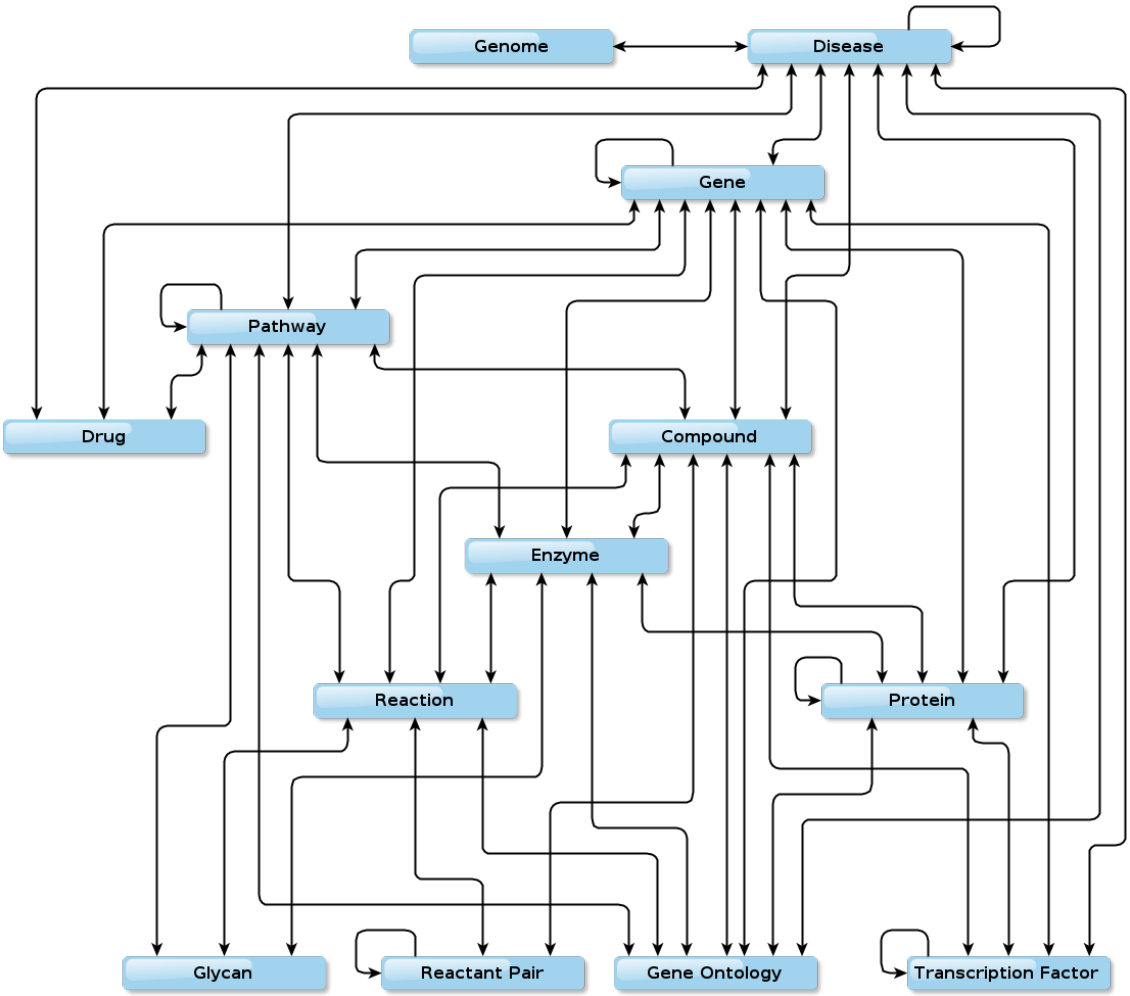


Abbildung 4.7: Beziehungen zwischen den einzelnen Domänen.

Disease, *Gene*, *Pathway*, *Protein*, *Reactant Pair* und *Transcription Factor* über eine rekursive Beziehung verfügen. Das bedeutet, dass einige Datensätze wieder auf andere Datensätze der identischen Domäne referenzieren können. Die Beziehungen zwischen zwei Domänen sind ungerichtet.

Durch einen spezifischen *Parser*, der Bestandteil eines ETL-Prozesses ist, wird die eigentliche Integration der Datenquellen initiiert und durchgeführt. Die relevanten Informationen zur Identifikation der Beziehungen zwischen den molekularbiologischen DB werden ebenfalls durch den *Parser* ermittelt und in spezielle Datenbanktabellen gespeichert. Ein Sonderfall ist die Datenquelle Ensembl, deren Datenbestände ausschließlich für TraBi und deren Funktionsumfang benötigt werden. Mit Hilfe von BioMart [KKH+11] ist es möglich, ausgewählte Daten der Datenquelle Ensembl in

eine Textdatei im FASTA-Format zu exportieren. Damit eine Identifizierung von potenziellen TFBS in Nukleotidsequenzen erfolgen kann, sind Nukleotidsequenzen der 5'-Upstream-Region und/oder der 3'-Downstream-Region notwendig sowie grundlegende Informationen über die entsprechenden Gene. Durch geeignete Einstellungen bei BioMart kann der Export der eben erwähnten Informationen realisiert werden. Dabei müssen hinsichtlich der Anforderungen von TraBi die folgenden Kriterien beachtet werden.

- Es werden ausschließlich Datensätze aus der Datenquelle Ensembl verwendet, die über eine eindeutige Ensembl-Identifikationsnummer verfügen. Dafür ist die Ensembl-Identifikationsnummer ENSG00000137573 ein Beispiel, die bei Ensembl das humane Gen SULF1 eindeutig identifiziert.
- Sofern der Status eines Gens nicht bekannt ist, wird der entsprechende Datensatz nicht berücksichtigt.
- Die einzelnen Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region der jeweiligen Gene sollten jeweils eine Länge von maximal 2500 bp, 5000 bp und 10000 bp aufweisen.
- Es sind lediglich die Gene der eukaryotischen Organismen *Homo sapiens*, *Mus musculus* und *Rattus norvegicus* zu berücksichtigen.

Die daraus resultierende Textdatei kann problemlos weiterverarbeitet werden und beinhaltet im Bezug auf das Gen die eindeutige Ensembl-Identifikationsnummer, den Namen, die Beschreibung, die exakte Position im Genom und die jeweilige Nukleotidsequenz der 5'-Upstream-Region oder der 3'-Downstream-Region. Anhand der Tabelle 4.3 wird deutlich, wie viele Gene und welche Chromosomen der drei Organismen durch die oben genannten Kriterien exportiert wurden und bei TraBi verfügbar sind. Durch die Datenquellen TRANSFAC® und JASPAR werden

Organismus	**Anzahl der Gene**	**Chromosom**	**5'-Upstream-Region**	**3'-Downstream-Region**
Homo sapiens	19288	1 - 22, X, Y	✓	
Mus musculus	23180	1 - 19, X, Y	✓	
Rattus norvegicus	15461	1 - 20, X	✓	

Tabelle 4.3: Übersicht der exportierten Daten aus Ensembl mittels BioMart.

die erforderlichen Datenbestände über TF, TFBS und PSSM bereitgestellt. Allerdings müssen die Daten der beiden Datenquellen nachträglich überarbeitet und mit weiteren Informationen ergänzt werden, weil nicht alle Datensätze über eine Konsensussequenz und eine eindeutige Identifikationsnummer hinsichtlich der Taxonomie verfügen. Darüber hinaus ist der Name der Organismen und deren Syntax nicht einheitlich, sodass eine Organismus spezifische Klassifikation der Datensätze

nicht möglich ist. Der erweiterte Algorithmus *ESAsearch* benötigt noch ESA, die auf der Grundlage der Nukleotidsequenzen erstellt werden. Aufgrund der Spezifität der oben genannten Merkmale ist es empfehlenswert, eine eigenständige Software-Infrastruktur zu entwickeln, die ausschließlich essentielle Datenbestände für TraBi erstellt. Außerdem wäre eine nachträgliche Aktualisierung und Erweiterung der Datenbestände durch diese Software-Infrastruktur ohne großen Aufwand möglich. Durch einen spezifischen *Wrapper* könnte die automatische Extraktion der Daten aus den unterschiedlichen Datenquellen wie einen Webservice oder eine molekularbiologische DB realisiert werden. Danach erfolgt eine einheitliche Strukturierung der Daten. Sofern Inkonsistenzen oder Heterogenitäten existieren, werden diese ebenfalls beseitigt. Abschließend werden die entsprechenden Daten in die dafür vorgesehenen Datenbanktabellen des DWH persistent gespeichert. Im Gegensatz dazu kann die Textdatei im FASTA-Format durch einen speziellen *Parser* analysiert werden, sodass die relevanten Daten identifiziert und danach in die jeweiligen Datenbanktabellen des DWH gespeichert werden. Das Erstellen der ESA kann ebenfalls durch den *Parser* erfolgen, aber diese Datenstruktur sollte nicht im DWH gespeichert werden, weil diese Herangehensweise ineffizient ist und den Grad der Komplexität unnötig erhöht. Stattdessen ist es sinnvoll solche Datenstrukturen als Datei im Dateisystem zu speichern. Es ist erforderlich für jedes Gen mehrere ESA zu erstellen, die entweder die 5'-Upstream-Region oder die 3'-Downstream-Region repräsentieren und jeweils auf Nukleotidsequenzen der Länge 2500 bp, 5000 bp oder 10000 bp basieren. Die exakte Funktionsweise der Software-Infrastruktur und deren Struktur und Design wird in Abschnitt 5.2.3.1 behandelt.

Der nächste Abschnitt erläutert die dafür notwendigen Datenbankschemata und deren Eigenschaften, die zentrale Bestandteile des DWH sind. Die eigentliche Konzeption des DWH wird ebenfalls im Abschnitt 4.3.4 behandelt.

4.3.4 Konzeption der Datenbankschemata

Das DWH besteht aus zwei DB und einem DBMS, das für die Verwaltung der beiden DB zuständig ist. Aufgrund der unterschiedlichen Datenbestände sind zwei DB sinnvoll, weil dadurch eine exakte Abgrenzung der Daten realisiert werden kann. Die DB *metadata* ist ausschließlich für die Speicherung der erforderlichen Metadaten im Hinblick auf die Analyse, Benutzerverwaltung, Registrierung und Statistik sowie das *Logging* zuständig. Im Gegensatz dazu werden die Datenbestände der relevanten molekularbiologischen DB in die DB *dawismd* gespeichert. Dabei sollte die Transaktionssicherheit und referentielle Integrität (RI) über Fremdschlüssel gewährleistet werden. Diese Anforderung kann durch die Storage-Engine InnoDB realisiert werden, welche inzwischen die Standard-Speicherengine bei MySQL ist. Das performante Suchen und Sortieren bei bestimmten Attributen kann durch eine Indexstruktur ermöglicht werden. Eine geeignete Indexstruktur bei DB ist der B-Baum, weil das Einfügen, Suchen und Löschen der Daten in amortisiert loga-

rithmischer Zeit erfolgt. Darüber hinaus ist diese Indexstruktur in der Praxis weit verbreitet und etabliert.

In der Abbildung 4.8 wird das Datenbankschema von *metadata* dargestellt, das aus 17 Datenbanktabellen besteht. Die Beziehungen und deren Kardinalitäten, die zwischen den jeweiligen Datenbanktabellen bei der DB *metadata* existieren werden ebenfalls durch die Abbildung 4.8 dargestellt. Außerdem zeigt Abbildung 4.8 die Indizes der jeweiligen Datenbanktabellen, sodass indexierte Attribute deutlich werden. Die Datenbanktabellen *db_data* und *db_statistics* sind für die Metadaten der Datenquellen und der Statistik zuständig. Das *Logging* kann mit Hilfe der DB erfolgen, das unter anderem durch die Datenbanktabelle *logging* ermöglicht wird. Insbesondere die Datenbanktabellen *trabi_jobs*, *trabi_result*, *trabi_gene*, *trabi_pssm* und *trabi_pssm_data* werden für TraBi und deren Funktionalität benötigt. Dabei ist *trabi_jobs* für die Metadaten von laufenden, beendeten und fehlerhaften Vorhersagen verantwortlich und *trabi_result* für die Ergebnisse einer Vorhersage. Die notwendigen Datenbestände über TF, TFBS und PSSM sowie die grundlegenden Informationen der Gene (Name, Beschreibung und die exakte Position im Genom) werden durch *trabi_pssm* und *trabi_pssm_data* bzw. *trabi_gene* bereitgestellt. Der Anwender kann bei TraBi benutzerspezifische PSSM erstellen und diese persistent in die DB *metadata* speichern. Dafür sind die beiden Datenbanktabellen *user_pssm* und *user_pssm_data* notwendig. Ein Suchprofil repräsentiert bei TraBi die relevanten Metadaten einer konfigurierten Vorhersage von potenziellen TFBS in Nukleotidsequenzen, die persistent in die DB gespeichert werden. Dadurch kann eine Vorhersage problemlos wiederholt und/oder deren Einstellungen nachträglich modifiziert werden, das mittels der Datenbanktabellen *user_search_profile*, *user_search_profile_gene* und *user_search_profile_pssm* sichergestellt wird. Die Benutzerverwaltung, die Registrierung und die An- und Abmeldung bei DAWIS-M.D. und TraBi wird mit Hilfe der Datenbanktabellen *session*, *user*, *user_details* und *user_account* realisiert.

Das Datenbankschema von *dawismd* kann nicht grafisch dargestellt werden, weil die Struktur äußerst komplex ist und gegenwärtig 526 Datenbanktabellen umfasst. Deswegen wird im Folgenden dieses Datenbankschema anhand von zwei exemplarischen Beispielen behandelt. Die DB *dawismd* verfügt über einen Datenbestand aus 13 molekularbiologischen DB, wobei jede Datenquelle durch ein eigenes Datenbankschema repräsentiert wird. Diese Herangehensweise wird als lose Kopplung bezeichnet und ermöglicht eine unkomplizierte Aktualisierung der Datenbestände der einzelnen Datenquellen. Außerdem ist durch die eindeutige Bezeichnung der Datenbanktabellen eine problemlose Lokalisierung und Identifizierung der Datenquelle und deren Datenbestände möglich. Diese Bezeichnung setzt sich aus dem Namen der Datenquelle und dem Datenbestand, den die Datenbanktabelle beinhaltet zusammen. Dafür ist die Datenbanktabelle *embl_sequence* ein Beispiel, die Nukleotidsequenzen der Gene beinhaltet, welche ursprünglich aus der Datenquelle EMBL-Bank resultieren. Die unterschiedlichen Datenbankschemata der jeweiligen Datenquellen in der DB *dawismd* basieren auf dem Sternschema oder dem Schneeflockenschema, die etablierte

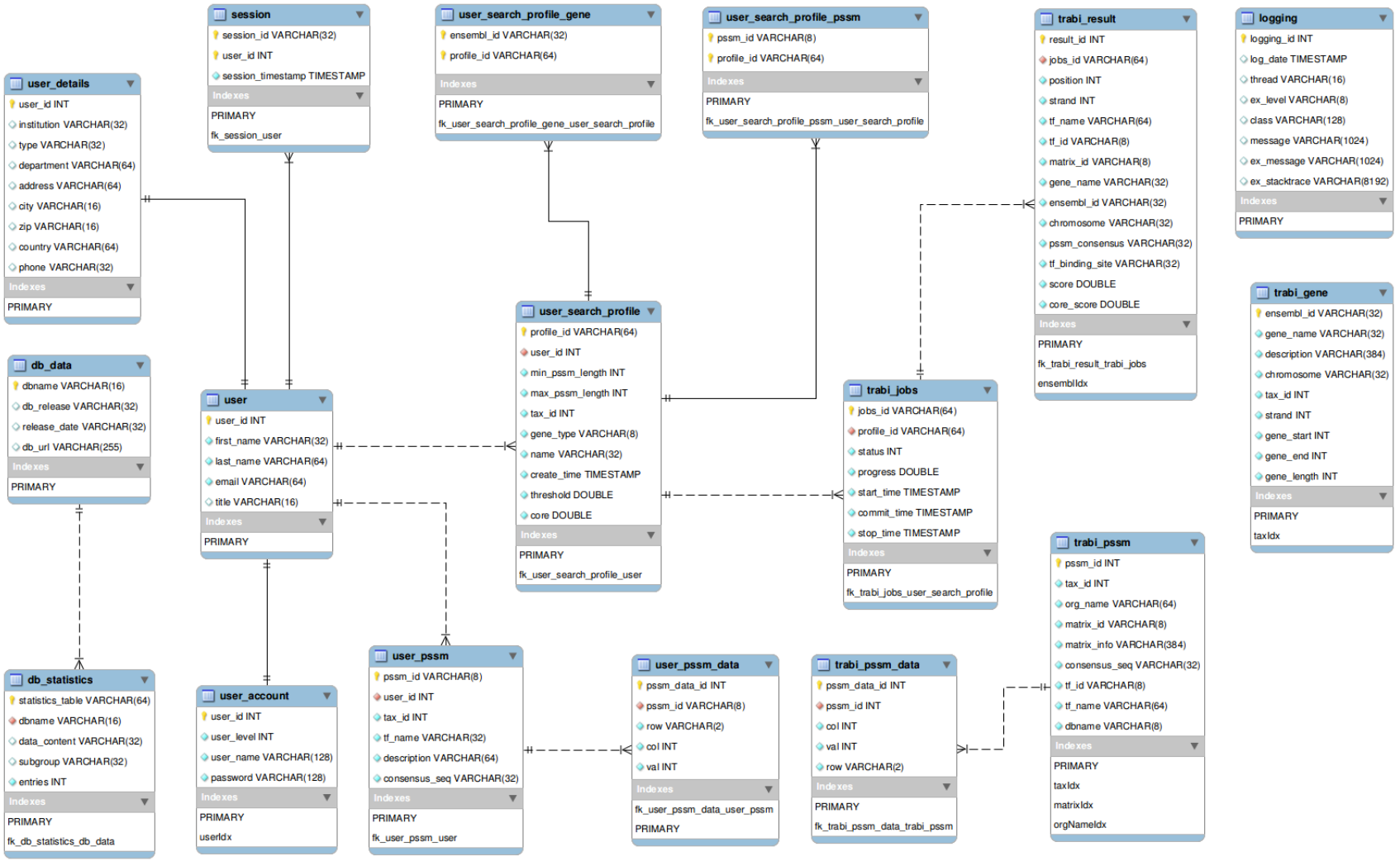


Abbildung 4.8: Datenbankschema der DB *metadata*.

Schemata bei einem DWH sind. Allerdings kann keine allgemeine These darüber aufgestellt werden, welches Schema besser geeignet ist, weil das von den konkreten Charakteristika der Daten und der Anfragen abhängig ist. Deswegen werden in der Praxis häufig Mischformen benutzt. Insbesondere die einfache Struktur, die problemlose und flexible Darstellung sowie die performante Anfrageverarbeitung sind Vorteile des Sternschemas. Das Schneeflockenschema ist eine Erweiterung des Sternschemas und benötigt eine geringere Speicherkapazität, weil die Dimensionstabellen aufgrund der Normalisierung keine redundanten Daten beinhalten. Im Gegensatz zum Schneeflockenschema verfügt das Sternschema über eine bessere Effizienz bei der Anfrageverarbeitung, das durch eine bewusste Denormalisierung bei den Dimensionstabellen ermöglicht wird. Die Normalisierung minimiert die Redundanz der Daten, sodass mögliche Anomalien und Inkonsistenzen verhindert werden.

Es gibt fünf Normalformen und die spezielle Boyce-Codd-Normalform (BCNF), wobei in der Praxis nur die ersten drei Normalformen angewendet werden. In der Regel gewährleisten alle Datenbanktabellen im DWH die erste und zweite Normalform. Damit eine effiziente Anfrageverarbeitung möglich ist, wurde eine Normalisierung auf Grundlage der dritten Normalform bei einigen Datenbanktabellen nicht durchgeführt. Das bedeutet, dass durch kontrollierte Redundanz eine verbesserte Performance realisiert wird. Im Folgenden werden zwei Schemata dargestellt, die Bestandteil der DB *dawismd* sind und auf einem Sternschema bzw. Schneeflockenschema basieren und verschiedene Datenquellen repräsentieren. Anhand der entsprechenden Abbildungen werden auch die Beziehungen und deren Kardinalitäten bei beiden Schemata deutlich. Außerdem werden die Indizes der einzelnen Datenbanktabellen in beiden Abbildungen explizit dargestellt. Die Abbildung 4.9 zeigt das Datenbankschema der Datenquelle ENZYME, das auf einem Sternschema basiert. Dabei kann die Datenbanktabelle *enzyme_ enyzme* als Faktentabelle und die übrigen Datenbanktabellen als Dimensionstabellen bezeichnet werden. Das Datenbankschema der Datenquelle EMBL-Bank basiert auf dem Schneeflockenschema und wird in Abbildung 4.10 dargestellt. Die Datenbanktabelle *embl_ identification* kann als Faktentabelle und die anderen Datenbanktabellen können als Dimensionstabellen bezeichnet werden. Anhand der Abbildung 4.10 wird deutlich, dass die Dimensionstabellen *embl_ dblinks* und *embl_ ref* durch strikte Normalisierung weiter verfeinert wurden. Dadurch wird primär die Redundanz der Daten minimiert und so eine bessere Strukturierung der Daten ermöglicht, die aber zusätzliche Verbundoperationen benötigt.

Die Querverweise zu anderen molekularbiologischen DB oder zwischen Datenbeständen innerhalb der molekularbiologischen DB werden in separaten Datenbanktabellen gespeichert, wodurch eine einfache und effiziente Identifizierung möglich ist. Diese Datenbanktabellen werden als Mapping-Tabellen bezeichnet und beinhalten mindestens die Identifikationsnummern von denjenigen Datensätzen, zwischen denen eine Beziehung existiert. Des Weiteren sind häufig der Name der molekularbiologischen DB, auf die referenziert wird, und eine zusätzliche Identifikationsnummer oder Anmerkungen verfügbar. Dafür sind *embl_ dblinks* in Abbildung 4.10 und *enzyme_* -

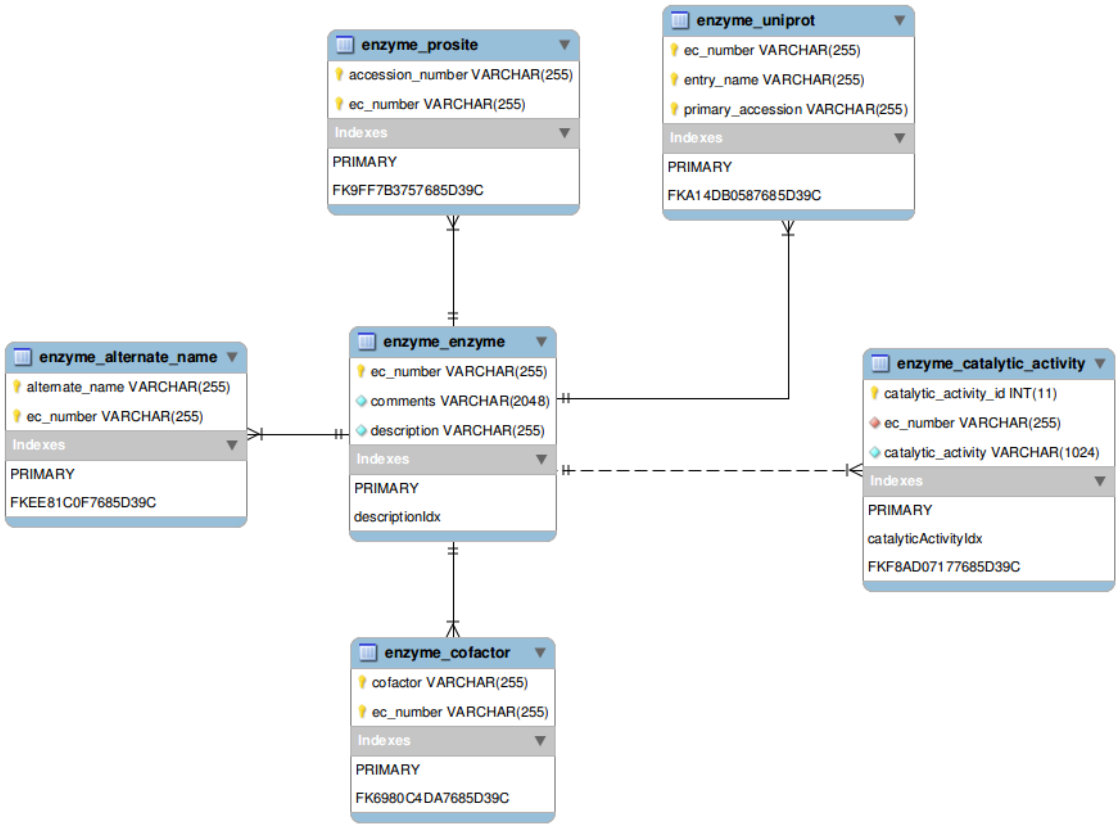


Abbildung 4.9: Datenbankschema der Datenquelle ENZYME.

prosite sowie *enzyme_ uniprot* in Abbildung 4.9 typische Beispiele.

4.4 Systemarchitektur

Ausgehend von der Anforderungsanalyse wird die Systemarchitektur von DAWIS-M.D. und TraBi abgeleitet, sodass deren Eigenschaften und die Interaktionen und Abhängigkeiten zwischen den einzelnen Schichten deutlich werden. Die Datenintegration ist ein Hauptbestandteil in der vorliegenden Arbeit und wurde durch BioDWH ermöglicht. Deswegen wird zunächst in Abschnitt 4.4.1 die Systemarchitektur von BioDWH beschrieben. Eine ausführliche Erläuterung dieser Systemarchitektur erfolgt in [TKKH08, Kor10]. Die entsprechenden Systemarchitekturen von DAWIS-M.D. und TraBi werden in Abschnitt 4.4.2 bzw. Abschnitt 4.4.3 vorgestellt.

Die Softwarequalität einer Webanwendung ist in erster Linie von der Systemarchitektur abhängig. Darüber hinaus erfordern Webanwendungen höhere Anforderungen als klassische Anwendungssoftware. Eine unzureichende Architektur ist somit primär verantwortlich für die negative Beeinträchtigung der Software im Hinblick auf die Performance, Verfügbarkeit, Wartbarkeit und Erweiterbarkeit. Außerdem wird

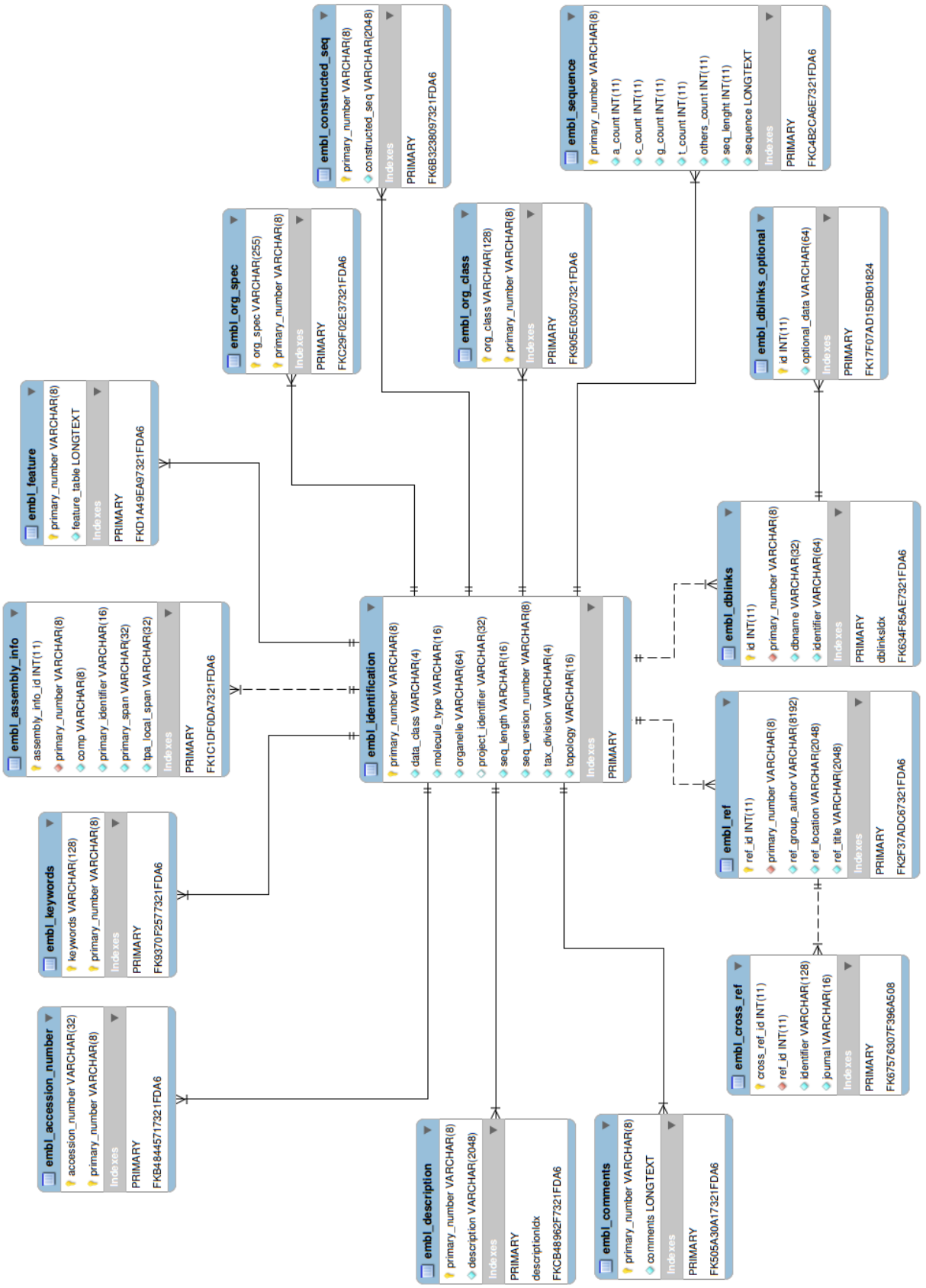


Abbildung 4.10: Datenbankschema der Datenquelle EMBL-Bank.

die Systemarchitektur durch die funktionalen und nicht-funktionalen Anforderungen einer Software beeinflusst.

Die 2-Schichten-Architektur, die auch als Client-Server-Architektur bezeichnet wird, und die 3-Schichten-Architektur ermöglichen eine Software entsprechend der Anforderungen zu strukturieren. Eine Erweiterung der 3-Schichten-Architektur ist die N-Schichten-Architektur, die über zusätzliche Schichten verfügt und so eine höhere Flexibilität gewährleistet. Allerdings ist eine 2-Schichten-Architektur ausschließlich für statische Webanwendungen geeignet. Das bedeutet, dass diese Architektur für multimediale Webanwendungen mit komplexen Funktionalitäten und zahlreichen gleichzeitigen Zugriffen nicht geeignet ist. Deswegen ist es sinnvoll, dass die Systemarchitekturen von DAWIS-M.D. und TraBi auf mehreren Schichten basieren. Die grundlegenden Prinzipien bei der Konzeption einer Architektur sind die Reduzierung der Komplexität, eine lose Kopplung zwischen den einzelnen Schichten und eine starke Kohäsion der Schichten. Durch eine geeignete Systemarchitektur, die diese Prinzipien berücksichtigt, ist eine Gewährleistung der in Abschnitt 4.1 thematisierten nicht-funktionalen Anforderungen wie Wartbarkeit, Skalierbarkeit, Erweiterbarkeit und Wiederverwertbarkeit bei DAWIS-M.D. und TraBi möglich. Eine Schicht symbolisiert eine eigenständige Komponente und realisiert somit die Aufteilung der Funktionalität oder der Datenbestände, wodurch letztendlich eine Trennung der Zuständigkeiten erfolgt. Infolgedessen wird das Design, die Implementierung und der Test der einzelnen Komponenten stark vereinfacht. Des Weiteren wird auf diese Weise eine modularisierte Struktur erzeugt und es werden entsprechende Schnittstellen definiert, welches ebenfalls eine notwendige Voraussetzung aus Abschnitt 4.1 ist. Eine weitere nicht-funktionale Anforderung die Abschnitt 4.1 verlangt, ist die Unabhängigkeit zwischen der Anwendungsschicht und der Datenbankschicht. Damit diese Unabhängigkeit bei DAWIS-M.D. und TraBi existiert, werden die entsprechenden Architekturen um eine zusätzliche Schicht erweitert, die als Persistenzschicht bezeichnet wird. Dadurch ist eine unkomplizierte und schnelle Migration auf andere DBMS möglich, ohne die Anwendungslogik der Software zu modifizieren.

4.4.1 Software-Infrastruktur zur Integration von molekularbiologischen Datenbanken

Die Abbildung 2.6 zeigt die Referenzarchitektur eines Data-Warehouse-Systems. Diese Architektur ist die Grundlage der Systemarchitektur von BioDWH (siehe Abbildung 4.11). Das Repositorium und der Auswertebereich sind kein Bestandteil der Systemarchitektur, das wird durch einen Vergleich der Abbildungen 2.6 und 4.11 deutlich. Darüber hinaus ist in BioDWH keine eigenständige Komponente für die Administration der Metadaten verfügbar, die als Metadatenmanager bezeichnet wird. Der Data-Warehouse-Manager ist die zentrale Komponente bei der Systemarchitektur, weil diese für die Initiierung, Steuerung und Überwachung der jeweiligen Prozesse verantwortlich ist. Außerdem ist diese Komponente für das *Logging* und

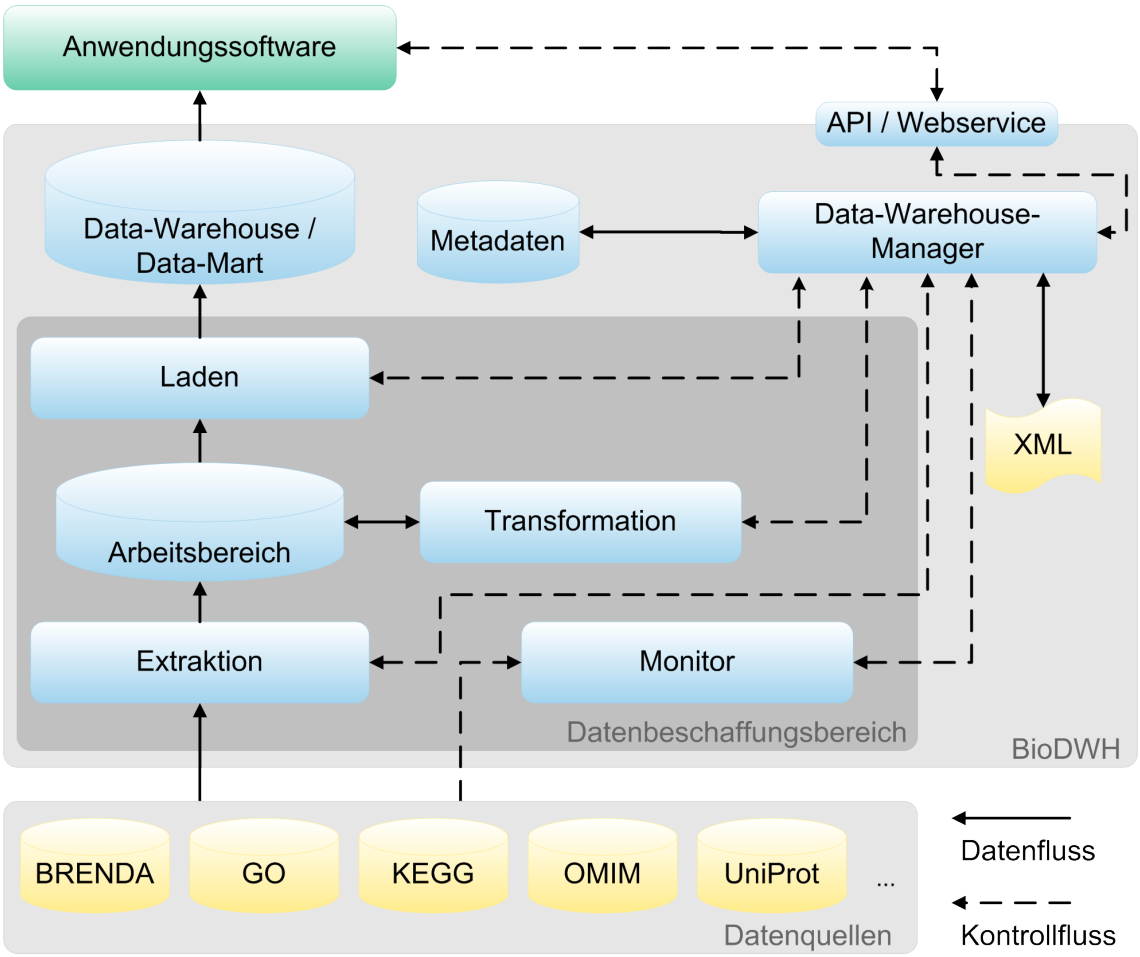


Abbildung 4.11: Systemarchitektur von BioDWH nach [Kor10].

die Konfiguration bei BioDWH zuständig. Damit eine einfache Wiederherstellung von fehlerhaften Datensätzen möglich ist, werden die Metadaten im Hinblick auf das *Logging* in eine zusätzliche DB gespeichert. Im Gegensatz dazu, erfolgt die Speicherung der relevanten Metadaten bei der Konfiguration in eine Textdatei, die auf XML basiert.

Eine Analyse der Daten ist durch diese Software-Infrastruktur nicht beabsichtigt. Infolgedessen ist der Auswertebereich und die hierfür spezialisierten Komponenten nicht erforderlich. Die eigentliche Analyse wird durch spezifische Anwendungssoftware wie CELLmicrocosmos [STK+10], TraBi oder VANESA durchgeführt. Die dafür notwendigen molekularbiologischen Datenbestände werden durch ein DWH oder einen spezifischen Data-Mart bereitgestellt, deren Realisierung mittels BioDWH möglich ist. Der ETL-Prozess und dessen Komponenten ermöglichen die Datenintegration bei BioDWH und sind wie der Monitor und der Arbeitsbereich essentielle Bestandteile des Datenbeschaffungsbereichs.

Die Schicht der Datenquellen ist zwar kein Bestandteil der Software-Infrastruktur,

kann aber als Ausgangspunkt des Datenflusses beim Data-Warehouse-Prozess bezeichnet werden. Außerdem repräsentiert diese Schicht mehrere Datenquellen deren Integration durch BioDWH gewährleistet wird (siehe Tabelle 5.2). Das DBMS und die zugrundeliegende DB wird in der Abbildung 4.11 durch das Data-Warehouse/Data-Mart repräsentiert.

4.4.2 Data-Warehouse-System für molekularbiologische Daten

Die Systemarchitektur von DAWIS-M.D. basiert auf der N-Schichten-Architektur, die gegenüber der 3-Schichten-Architektur eine höhere Flexibilität bietet. Anhand der Abbildung 4.12 wird deutlich, dass diese Systemarchitektur aus fünf unterschiedlichen Schichten besteht, die im Folgenden erläutert werden:

Anwendungsschicht: Die Anwendungsschicht repräsentiert eine dynamische und interaktive Webanwendung und weitere spezielle Anwendungssoftware (CELLmicrocosmos und VANESA) sowie deren erforderlichen Komponenten. Durch diese Schicht und die unterschiedlichen Softwarelösungen kann die Interaktion mit dem Benutzer erfolgen. Außerdem wird dem Anwender eine homogene, konsistente und integrierte Sicht auf den Datenbestand aus dem DWH oder dem jeweiligen Data-Mart zur Verfügung gestellt. Das standardisierte *Hypertext Transfer Protocol* (HTTP) realisiert den Datenzugriff und -austausch zwischen der Webanwendung und dem Webserver. Die übrige Software benutzt ausschließlich einen Webservice und das *Simple Object Access Protocol* (SOAP) zur Kommunikation. Die Anwendungsschicht wiederum kommuniziert lediglich mit der Persistenzschicht, das durch die Systemarchitektur in der Abbildung 4.12 veranschaulicht wird.

Persistenzschicht: Damit keine direkte Abhängigkeit zwischen der Anwendungsschicht und der Datenbankschicht existiert, verfügt die Systemarchitektur über eine zusätzliche Schicht, die als Persistenzschicht bezeichnet wird. Dadurch können verschiedene DBMS wie MySQL, PostgreSQL oder Oracle verwendet werden, ohne die Anwendungslogik der Webanwendung und der jeweiligen Anwendungssoftware zu modifizieren. Diese Schicht beinhaltet eine spezielle Komponente, welche die Technik der objektrelationalen Abbildung bereitstellt und so die eigentliche Abstraktion zwischen der Anwendungsschicht und der Datenbankschicht realisiert.

Datenbankschicht: Die zentrale Komponente der Datenbankschicht ist das DWH, das zwei verschiedene DB beinhaltet. Der gesamte molekularbiologische Datenbestand der Datenquellen wird in der DB *dawismd* gespeichert. Im Gegensatz dazu, speichert die DB *metadata* ausschließlich Metadaten, die hinsichtlich einer Analyse oder einer Softwarelösung benötigt werden. Das Erstellen von

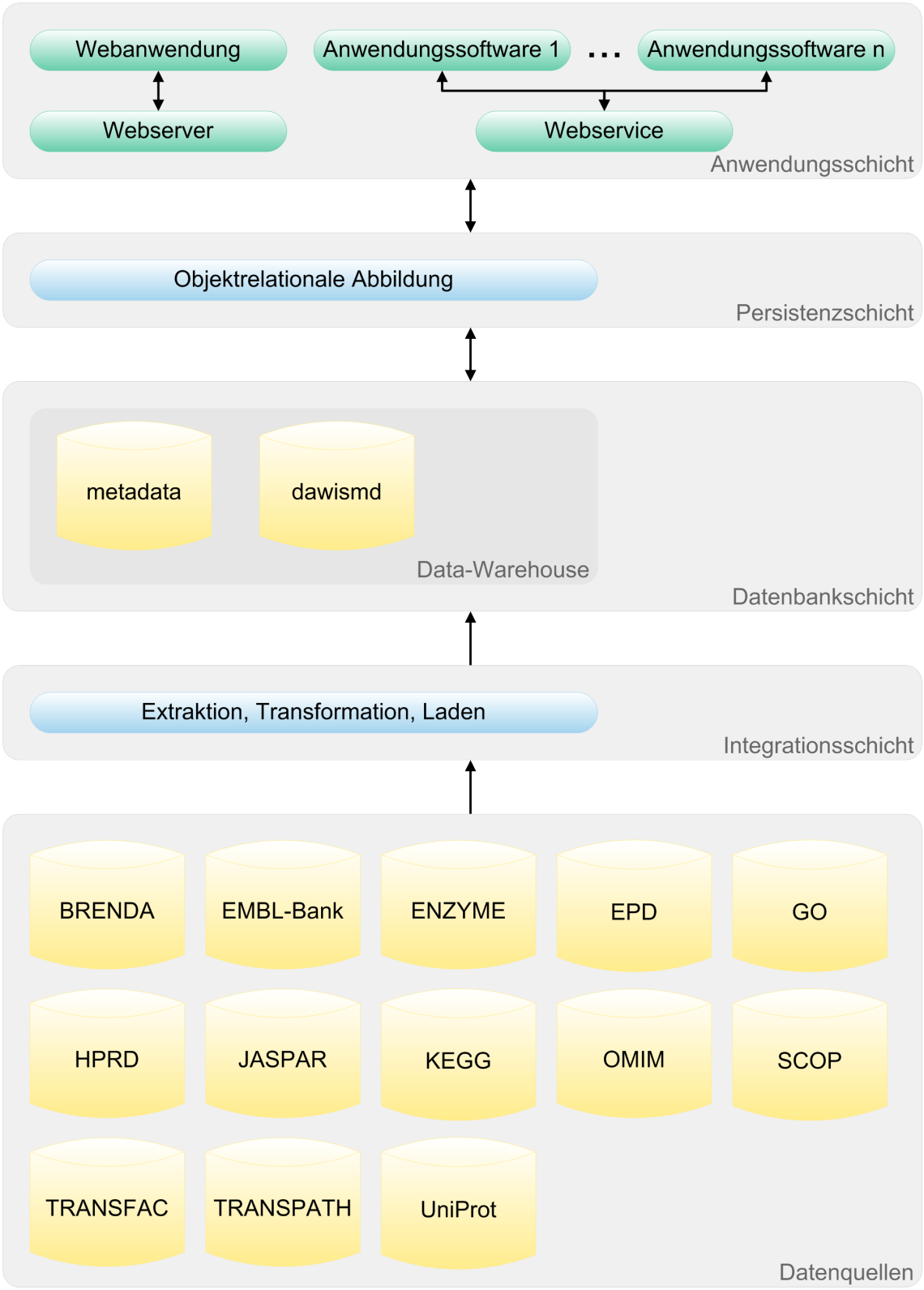


Abbildung 4.12: Systemarchitektur von DAWIS-M.D. nach [HKT+10].

spezifischen Data-Marts wird ebenfalls durch diese Schicht gewährleistet und ist für bestimmte Softwarelösungen wie CELLmicrocosmos und VANESA sinnvoll, weil diese nur einen speziellen molekularbiologischen Datenbestand aus dem DWH benötigen. Die Datenbankschnittstelle *Java Database Connectivity* (JDBC) ist für die Kommunikation zwischen der Persistenz-, Datenbank- und Integrationsschicht zuständig.

Integrationsschicht: Die Extraktion der Datenbestände aus den molekularbiologischen DB und die nachfolgende Transformation der Daten in das jeweilige Schema und Format wird durch die Integrationsschicht ermöglicht. Diese Schicht ist ebenfalls für das effiziente Laden der relevanten Datenbestände in die jeweilige DB des DWH verantwortlich. Außerdem werden die einzelnen Datenquellen in regelmäßigen Zeitabständen durch einen Monitor kontrolliert, wodurch Änderungen automatisch identifiziert werden können. Es wird für zahlreiche molekularbiologische DB ein *Parser* bereitgestellt, der einen ETL-Prozess initiiert und die eigentliche Datenintegration durchführt. Die notwendige Identifizierung der Beziehungen zwischen den einzelnen Datenquellen und den Domänen erfolgt während der Datenintegration. Diese Daten werden in speziellen Datenbanktabellen gespeichert, die als Mapping-Tabellen bezeichnet werden.

Datenquellen: Diese Schicht repräsentiert 13 verschiedene molekularbiologischen DB und deren Datenbestände, die hinsichtlich der Anforderungen erforderlich sind. In der Tabelle 4.1 werden diese molekularbiologischen DB und deren Klassifikation dargestellt. Der Datenbestand der einzelnen molekularbiologischen DB wird als strukturierte *flat file* oder in XML zur Verfügung gestellt.

4.4.3 Informationssystem zur Identifikation von potenziellen Transkriptionsfaktorbindestellen in Nukleotidsequenzen

Die Abbildung 4.13 zeigt die Systemarchitektur von TraBi, die ebenfalls auf der N-Schichten-Architektur basiert. Diese Architektur verfügt über fünf verschiedene Schichten (siehe Abbildung 4.13). Die einzelnen Schichten der Systemarchitektur werden im Folgenden behandelt:

Anwendungsschicht: Im Gegensatz zu der Anwendungsschicht in der Abbildung 4.12 repräsentiert die Anwendungsschicht in Abbildung 4.13 eine dynamische und interaktive Webanwendung sowie einen Webserver, der hinsichtlich der Webanwendung benötigt wird. Dabei wird die Kommunikation zwischen diesen beiden Komponenten durch das standardisierte Netzwerkprotokoll HTTP realisiert. Die Webanwendung verfügt über zahlreiche Funktionalitäten und eine intuitive GBO, die interaktive und kollaborative Elemente im Kontext

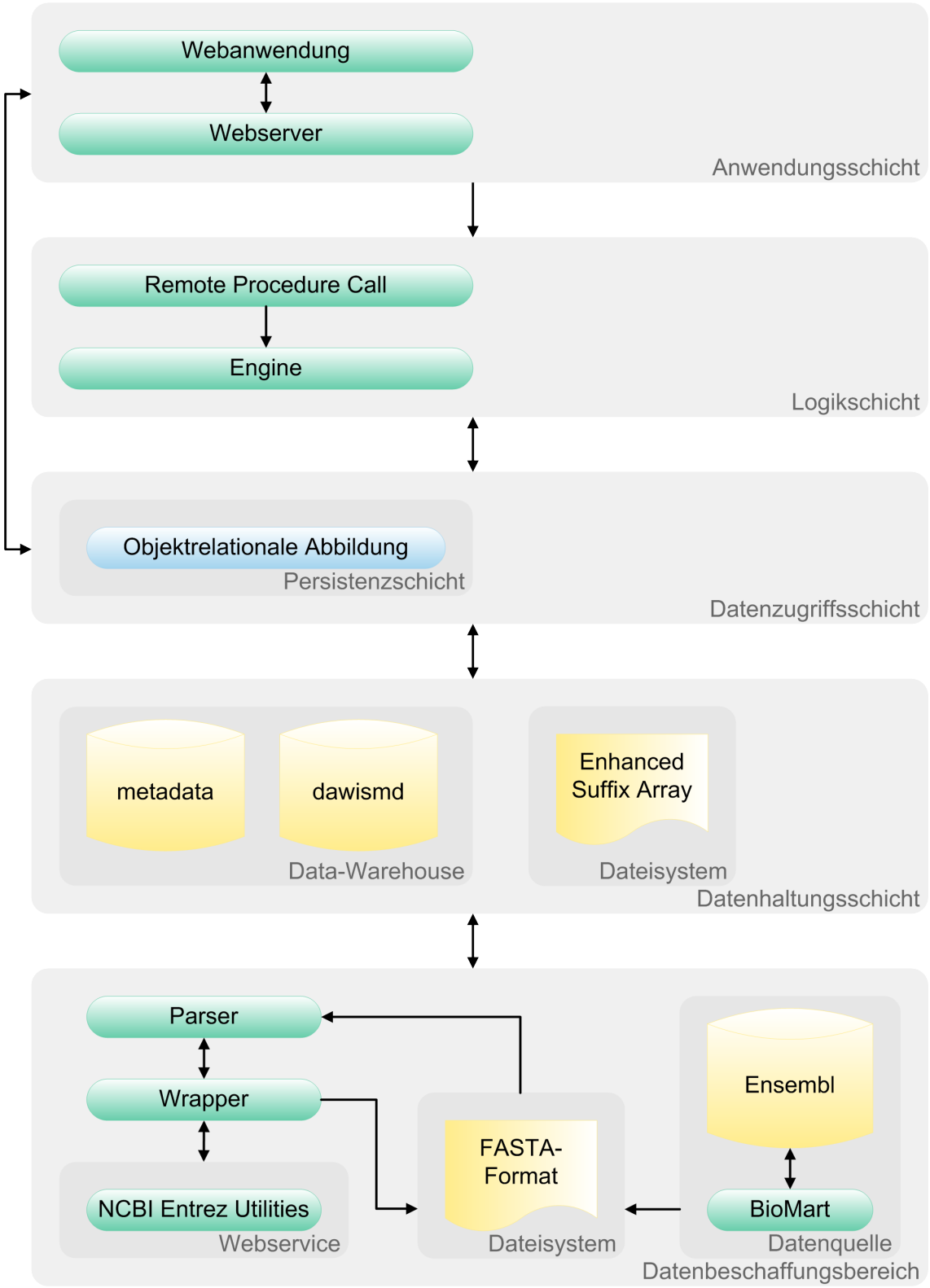


Abbildung 4.13: Systemarchitektur von TraBi.

des *Web 2.0* bereitstellt. Daraus ergibt sich, dass die Interaktion zwischen Benutzer und Anwendungsschicht ausschließlich durch die Webanwendung erfolgt. Die Anwendungsschicht kommuniziert mit der Datenzugriffsschicht und der Logikschicht, welche über die essentielle Anwendungslogik für die Identifikation von potenziellen TFBS in Nukleotidsequenzen verfügt. Aufgrund der Platz- und Zeitkomplexität sowie der Flexibilität erfolgt diese Identifikation nicht innerhalb der Anwendungsschicht, sondern durch die Logikschicht.

Logikschicht: Aufgrund der Platz- und Zeitkomplexität der Datenverarbeitung wird die dafür zuständige Anwendungslogik als eigenständige und unabhängige Logikschicht realisiert. Durch diese Autonomie ist eine unkomplizierte Lastverteilung auf einen separaten Rechnerverbund möglich, wodurch die Skalierbarkeit der Software profitiert. Der erweiterte Algorithmus *ESAsearch* ist ein Bestandteil der Engine, die eine zentrale Softwarekomponente ist und die einzelnen Identifikation von potenziellen TFBS in Nukleotidsequenzen sequentiell durchführt. Dabei wird *First-come, first-served* (FCFS) als Strategie eingesetzt. Die Technik *Remote Procedure Call* (RPC) ermöglicht eine Interprozesskommunikation zwischen der Anwendungsschicht und der Logikschicht, sodass eine Methode oder ein Prozess in der Logikschicht initiiert werden kann. Eine Möglichkeit, diese Art der Kommunikation zu implementieren, ist *Remote Method Invocation* (RMI), das auf der Programmiersprache Java basiert.

Datenzugriffsschicht: Die Datenzugriffsschicht ermöglicht der Anwendungsschicht und der Logikschicht einen direkten Zugriff auf die jeweiligen Daten in der Datenhaltungsschicht. Das persistente Speichern von Datenbeständen in das DWH wird ebenfalls durch diese Schicht realisiert. Dabei ist die Persistenzschicht, die eine zusätzliche Schicht innerhalb der Datenzugriffsschicht ist, ausschließlich für den unabhängigen Datenzugriff auf das DWH zuständig. Die objektrelationale Abbildung ist ein Bestandteil der Persistenzschicht, wodurch unter anderem der konzeptionelle Widerspruch *object-relational impedance mismatch* beseitigt wird. Außerdem bietet die Datenzugriffsschicht einen Zugriff auf die ESA, die als Datei im Dateisystem zur Verfügung gestellt werden. Aufgrund der Informationssicherheit verfügt diese Schicht über einen eingeschränkten Zugriff auf das Dateisystem.

Datenhaltungsschicht: Die zentrale Komponente der Datenhaltungsschicht ist das DWH, das die beiden DB *metadata* und *dawismd* sowie ein DBMS repräsentiert. Das DWH beinhaltet in den entsprechenden DB molekularbiologische Daten und erforderliche Metadaten hinsichtlich der Webanwendung und der Anwendungssoftware. Des Weiteren verfügt diese Schicht über die notwendigen ESA, die ebenfalls ein Bestandteil der Datenbasis sind, aber nicht im DWH gespeichert werden, sondern als Datei im Dateisystem. Die spezifische Kommunikation mit der DB wird durch die standardisierte Datenbankschnittstelle JDBC realisiert.

Datenbeschaffungsbereich: Die unterste Schicht repräsentiert verschiedene Schritte und Datenquellen und wird als Datenbeschaffungsbereich bezeichnet. Diese Schicht sollte als eigenständige Software-Infrastruktur realisiert werden, welche die notwendigen Softwarekomponenten für den Prozess zur Datenakquisition, -bereinigung und -fusion bei TraBi bereitstellt. Dabei sind hauptsächlich der *Parser* und der *Wrapper* elementare Bestandteile, weil diese Softwarekomponenten grundlegende Funktionalitäten zur Verfügung stellen.

Als erstes müssen die Metadaten der Gene und deren Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region mittels BioMart aus der Datenquelle Ensembl manuell exportiert werden. Der Export erfolgt als strukturierte Textdatei deren Dateiformat das FASTA-Format ist, wodurch eine problemlose und automatische Weiterverarbeitung durch den spezifischen *Parser* möglich ist. Der *Parser* analysiert die Textdatei und identifiziert die relevanten Metadaten der Gene und erstellt für jedes Gen mehrere ESA, die jeweils auf eine Nukleotidsequenz der Länge 2500 bp, 5000 bp oder 10000 bp basieren und die 5'-Upstream-Region oder die 3'-Downstream-Region repräsentieren. Diese ESA sind die Grundlage für eine Identifikation von potenziellen TFBS in der 5'-Upstream-Region oder der 3'-Downstream-Region, wobei die Länge der Nukleotidsequenz entweder 2500 bp, 5000 bp oder 10000 bp ist. Außerdem werden Datenbestände über TF, TFBS und PSSM benötigt, welche die DB *dawismd* bereitstellt, die ein Bestandteil des DWH ist. Der *Wrapper* ist unter anderem für die automatische Extraktion der Datenbestände aus der DB *dawismd* verantwortlich, wobei die Kommunikation mittels der etablierten Datenbankschnittstelle JDBC erfolgt.

Sobald diese Extraktion beendet ist, wird im nächsten Schritt die Weiterverarbeitung der Datensätze durchgeführt, die ebenfalls durch den *Parser* erfolgt. Dabei werden die Datensätze einheitlich strukturiert, validiert und mit zusätzlichen Informationen ergänzt, sodass alle Datensätze über eine Konsenssequenz, eine eindeutige Identifikationsnummer hinsichtlich der Taxonomie und eine identische Bezeichnung für die Organismen verfügen. Infolgedessen können die Datensätze Organismus spezifisch sortiert werden. Allerdings wird die computergestützte Vervollständigung der Datensätze und die Datenbereinigung durch den externen *NCBI Entrez Utilities* Webservice unterstützt. Der Datenaustausch zwischen *Parser* und Webservice wird durch den *Wrapper* ermöglicht, der mittels SOAP die Kommunikation zum Webservice realisiert. Außerdem bietet der Webservice einen Datenzugriff auf die unterschiedlichen Datenquellen des NCBI (*Gene*, *Sequences* und *Taxonomy*). Dadurch kann eine automatisierte Alternative zu dem oben genannten manuellen Export mittels BioMart zur Verfügung gestellt werden, wobei der *Wrapper* die Extraktion und den Export der Daten durchführt.

Sofern die einzelnen Schritte erfolgreich durchgeführt wurden, werden die relevanten Metadaten der Gene, TF, TFBS und PSSM persistent in die jeweiligen Datenbanktabellen der DB *metadata* gespeichert, die ein Bestandteil des DWH

ist. Im Gegensatz dazu werden die essentiellen ESA nicht innerhalb des DWH gespeichert, sondern als Datei im Dateisystem.

4.5 Zusammenfassung

Dieses Kapitel thematisierte von DAWIS-M.D. und TraBi die Anforderungsanalyse und die Systemarchitektur. Des Weiteren wurde die Systemarchitektur der Software-Infrastruktur BioDWH erörtert, die hinsichtlich der Integration von molekularbiologischen DB eine wichtige Funktionalität für DAWIS-M.D. und TraBi zur Verfügung stellt.

Als erstes wurden in den beiden Abschnitten 4.1 und 4.2 die nicht-funktionalen und funktionalen Anforderungen der beiden Softwarelösungen festgelegt und behandelt. Außerdem zeigte der Abschnitt 4.2 die Systemfunktionalität als Anwendungsfalldiagramm, wodurch die Spezifikation der funktionalen Anforderungen der entsprechenden Software abstrakt dargestellt wurde. Danach thematisierte der Abschnitt 4.3 eine spezifische Anforderungsanalyse, die ausschließlich spezielle Aspekte der Problemstellung bzw. der Softwaretechnik berücksichtigte. Dabei wurde zuerst der Algorithmus *ESAsearch*, der im Rahmen der vorliegenden Arbeit erweitert wurde, erläutert. Darüber hinaus wurden in Abschnitt 4.3 die erforderlichen molekularbiologischen DB und deren Einschränkungen und Herausforderungen bei der Datenintegration sowie die Konzeption der Datenbankschemata thematisiert. Abschließend wurden in Abschnitt 4.4 die einzelnen Systemarchitekturen von BioDWH, DAWIS-M.D. und TraBi als Abbildungen dargestellt und erläutert, wobei eine Fokussierung auf DAWIS-M.D. und TraBi erfolgte. Insbesondere die unterschiedlichen Schichten und deren Beziehungen untereinander sowie die Bestandteile innerhalb einer Schicht wurden erklärt, sodass die Hauptaufgabe der jeweiligen Schicht deutlich wurde.

Die Einzelheiten der Implementierung und des Designs sowie die Funktionalität von DAWIS-M.D. und TraBi werden im nächsten Kapitel beschrieben. Aufgrund der Signifikanz für die diese Arbeit, wird auch der grundlegende Funktionsumfang von BioDWH erläutert, sodass die Funktionsweise der Software-Infrastruktur nachzuvollziehen ist.

5 Design und Implementierung

Das Design und die Implementierung von DAWIS-M.D. und TraBi wird als nächstes behandelt, sodass die Bestandteile und die Funktionalität der jeweiligen Software deutlich wird. Dabei werden einige dynamische Aspekte der beiden Softwarelösungen durch Aktivitätsdiagramme erläutert, die als Bestandteil der Modellierungssprache UML dafür besonders geeignet sind (siehe Anhang F.2). Die Aktivitätsdiagramme charakterisieren häufig die Systematik eines Anwendungsfalls, der im Kontext der UML zur Modellierung von Anforderungen bei Anwendungsfalldiagrammen eingesetzt wird. Allerdings basieren die Aktivitätsdiagramme in der vorliegenden Arbeit auf der UML 1.x, die einem älteren Standard der UML entspricht. Anhand von Abbildungen wird das Design der Prototypen veranschaulicht. Die grundlegenden Merkmale der Software-Infrastruktur BioDWH werden ebenfalls vorgestellt, weil BioDWH für die eigentliche Datenintegration verantwortlich ist und die dazu notwendigen Funktionalitäten bereitstellt. Außerdem zeigt dieses Kapitel, welche zusätzliche Software als Laufzeitumgebung benötigt wird und welche Technologien und Programmbibliotheken/Programmierschnittstellen bei der Entwicklung von DAWIS-M.D. und TraBi verwendet wurden.

Die Zielgruppe sollte bei der Implementierung der Software frühzeitig involviert werden, weshalb das aus der Softwaretechnik populäre Prototyping als Vorgehensweise eingesetzt wurde. Durch dieses Paradigma können Anforderungen jederzeit präzisiert und überprüft werden, weil die Zielgruppe die laufende Entwicklung aktiv begleitet. Dadurch wird auch das Risiko einer unzureichenden Software, welche die Anforderungen nicht gewährleistet, stark reduziert, weil die Qualitätssicherung zeitnah erfolgen kann. Darüber hinaus sind kurzfristig Prototypen der Software verfügbar und kritische Abhängigkeiten der Softwarekomponenten können ebenfalls rechtzeitig identifiziert werden.

Eine Webanwendung verfügt unter anderem über einen hohen Grad an Plattformunabhängigkeit und Flexibilität, weswegen DAWIS-M.D. und TraBi jeweils als eigenständige, dynamische und interaktive Webanwendung realisiert wurden. Die Wartung und Pflege/Weiterentwicklung einer Webanwendung ist ebenfalls problemlos möglich und es wird lediglich ein Webbrowser vorausgesetzt, der durch das Betriebssystem bereitgestellt wird. Insbesondere die exzellente Perspektive und das immense Potential sind entscheidende Argumente für eine Webanwendung und werden besonders durch das Schlagwort *Web 2.0* repräsentiert. Allerdings ist die Sicherheit einer

Webanwendung während der Konzeption und der Implementierung nicht zu unterschätzen, weil unterschiedliche Szenarien für einen Angriff auf eine Webanwendung wie *Cross-Site-Scripting* und *SQL-Injection* existieren. Deswegen ist es notwendig, dass entsprechende Konzepte die Sicherheit gewährleisten. Durch das Internet sind Webanwendungen weltweit verfügbar, sodass die Nutzung der jeweiligen Software durch zahlreiche potentielle Anwender möglich ist. Die beiden Softwarelösungen wurden im Kontext der OSS realisiert und sind der GPL unterstellt. Der Quelltext wird mittels SourceForge für interessierte Wissenschaftler/Softwareentwickler frei zur Verfügung gestellt. Dadurch soll die kontinuierliche Pflege/Weiterentwicklung und Verfügbarkeit der Software sichergestellt werden.

Als erstes wird in Abschnitt 5.1 die Realisierung der jeweiligen Systemarchitekturen erörtert, die in Abschnitt 4.4 bereits abstrakt und theoretisch thematisiert wurden. Danach gibt Abschnitt 5.2 einen exakten Überblick über die Struktur und den Funktionsumfang der einzelnen Softwarelösungen. Abschließend erfolgt in Abschnitt 5.3 eine Zusammenfassung.

5.1 Realisierung der Systemarchitekturen

In Abschnitt 4.4 wurde die Systemarchitektur von BioDWH, DAWIS-M.D. und TraBi dargestellt und die Konzeption der einzelnen Schichten vorgestellt. Der folgende Abschnitt gibt einen Überblick über die zusätzliche Software und die erforderlichen Technologien, die zur Realisierung der Systemarchitektur von DAWIS-M.D. und TraBi benötigt werden. Dabei wird deutlich, wie die jeweiligen Schichten bei der entsprechenden Systemarchitektur realisiert wurden. Außerdem wurden durch die Anforderungsanalyse in Kapitel 4 verschiedene Funktionalitäten und Anforderungen definiert, welche die beiden Softwarelösungen gewährleisten müssen. Dafür sind teilweise externe Programmbibliotheken/Programmierschnittstellen notwendig, weil dadurch die Implementierung erheblich vereinfacht wird. Die Tabelle 5.1 zeigt wichtige Programmbibliotheken/Programmierschnittstellen sowie deren Version, die bei TraBi, DAWIS-M.D. und/oder BioDWH eingesetzt wurden, aber im Folgenden nicht explizit erwähnt werden.

Data-Warehouse

Die zentrale Komponente bei beiden Systemarchitekturen ist das DWH, welches die DB *dawismd* und die DB *metadata* sowie ein DBMS repräsentiert. Die Software-Infrastruktur BioDWH verfügt über zahlreiche spezifische *Parser* für populäre molekularbiologische DB, sodass die Integration der erforderlichen molekularbiologischen DB aus Abschnitt 4.3.2 sichergestellt ist. Das bedeutet, dass BioDWH das DWH erstellt und ebenfalls für die Aktualisierung der molekularbiologischen Datenbestände im DWH und die Verwaltung der molekularbiologischen DB verantwortlich ist. Auf-

Programmbibliothek/ Programmierschnittstelle	**Version**	**DAWIS-M.D.**	**TraBi**	**BioDWH**
Apache Commons Collections	3.2	✓		
Apache Commons Email	1.2	✓		-
Apache Commons FileUpload	1.2.1	-	✓	-
Apache Commons IO	2.4	✓	-	✓
	1.4	-	✓	-
Apache MyFaces	2.1.9	✓		-
Apache POI	3.2-FINAL	-	✓	-
atmosphere-compat	0.5.1	-	✓	-
atmosphere-runtime				
iText	2.1.7	-	✓	-
JavaMail	1.4.5	✓		-
JExcelApi	2.6.12	✓	-	
JSON.simple	1.1.1	✓	-	
Jxlayer	136	-	✓	
log4j	1.2.17	✓		
MigLayout	4.0	-	✓	
opencsv	2.3	✓		-
Prototype	1.7.1	✓	-	
script.aculo.us	1.9.0	✓	-	
SLF4J	1.7.2	✓		

Tabelle 5.1: Übersicht der benutzten Programmbibliotheken/Programmierschnittstellen.

grund der Vorteile in Abschnitt 2.2.2 wird als DBMS ein RDBMS verwendet. Dafür ist die freie Software MySQL prädestiniert, die in der Praxis ein populäres RDBMS ist. Die Version 5.5.25a von MySQL wird derzeit eingesetzt. Der Datenzugriff und -austausch wird durch die standardisierte Datenbankschnittstelle JDBC ermöglicht, die besonders für RDBMS optimiert wurde. Der dazu benötigte MySQL Connector/J in der Version 5.1.22 implementiert JDBC 3.0 und repräsentiert einen Typ-4-Treiber, der einen äußerst schellen Zugriff auf die DB bietet. Damit es grundsätzlich möglich ist, verschiedene RDBMS wie MySQL, Oracle oder PostgreSQL einzusetzen ohne die Anwendungslogik der Software zu modifizieren, ist eine zusätzliche Schicht erforderlich, die als Persistenzschicht bezeichnet wird. Die Persistenzschicht wurde durch das *Framework* Hibernate in der Version 3.6.10 realisiert. Insbesondere die objektrelationale Abbildung und die eigene objektorientierte Abfragesprache *Hibernate Query Language* (HQL) sind elementare Funktionalitäten von Hibernate. Allerdings sollten alle Abfragen in HQL formuliert werden, welche Hibernate in SQL der entsprechenden DB transformiert. Dieses *Framework* wurde auch bei BioDWH eingesetzt, deren Charakteristika und Funktionalitäten in Abschnitt 5.2.1 behandelt werden.

Webanwendung

Die Implementierung der Webanwendungen erfolgte durch die serverseitige Technologie *JavaServer Pages* (JSP) und *JavaServer Faces* (JSF), die ein Bestandteil der Spezifikation *Java Platform, Enterprise Edition* (Java EE) sind. Außerdem wurde die grundlegende *Hypertext Markup Language* (HTML), die Skriptsprache JavaScript und zur einheitlichen Strukturierung und Gestaltung *Cascading Style Sheets* (CSS) verwendet.

Als zentrale Technik des *Web 2.0* kann Ajax bezeichnet werden, das ein Apronym für *Asynchronous JavaScript and XML* ist und die asynchrone Datenübertragung mittels HTTP zwischen Webbrowser und Webserver ermöglicht. Durch diese Technik konnten bei Webanwendungen interaktive und dynamische Komponenten entwickelt werden, wodurch die Effektivität und die Benutzerfreundlichkeit profitieren. Ein Webserver, der für JSP und JSF geeignet ist und in der Praxis über eine gute Stabilität und Performance verfügt, ist der Apache Tomcat[1]. Die kontinuierliche Pflege/Weiterentwicklung wird durch die *Apache Software Foundation* (ASF) gewährleistet. Die Version 7.0.29 fungiert als Webserver für beide Webanwendungen. Anhand der Abbildung 4.12 wird deutlich, dass ein Webservice verfügbar ist, der einen direkten Datenzugriff auf das DWH zur Verfügung stellt. Dadurch ist es möglich, dass spezialisierte Anwendungssoftware (CELLmicrocosmos und VANESA) die Daten des DWH problemlos verwenden kann. Der Webservice wurde mittels Apache Axis2[2] implementiert, das ebenfalls ein Projekt der ASF ist. Die Kommunikation erfolgt durch das etablierte Netzwerkprotokoll SOAP und kann asynchron oder synchron erfolgen. Sowohl das persistente Speichern von Datensätzen als auch der Zugriff auf die Datenbestände der kommerziellen Datenquellen TRANSFAC® und TRANSPATH® ist durch den Webservice nicht möglich.

Anwendungslogik

Aufgrund der Platz- und Zeitkomplexität wurde die Anwendungslogik, welche die eigentliche Identifizierung von potenziellen TFBS in Nukleotidsequenzen durchführt als eigenständige und unabhängige Softwarekomponente umgesetzt. Diese Softwarekomponente wurde bei der Systemarchitektur als Logikschicht dargestellt (siehe Abbildung 4.13). Dadurch profitiert die Skalierbarkeit, weil eine problemlose Lastverteilung auf einen separaten Rechnerverbund möglich ist. Der erweiterte Algorithmus *ESAsearch* ist ein Bestandteil der Softwarekomponente, die auf der Grundlage von RMI entwickelt wurde, das eine spezifische Technik der Programmiersprache Java ist und eine Interprozesskommunikation hinsichtlich RPC ermöglicht. Die Abbildung 5.1 zeigt eine schematische Übersicht von RMI und deren Komponenten. Die Softwarekomponente registriert als erstes eine Methode bei der *RMI-Registry*,

[1] http://tomcat.apache.org/
[2] http://axis.apache.org/

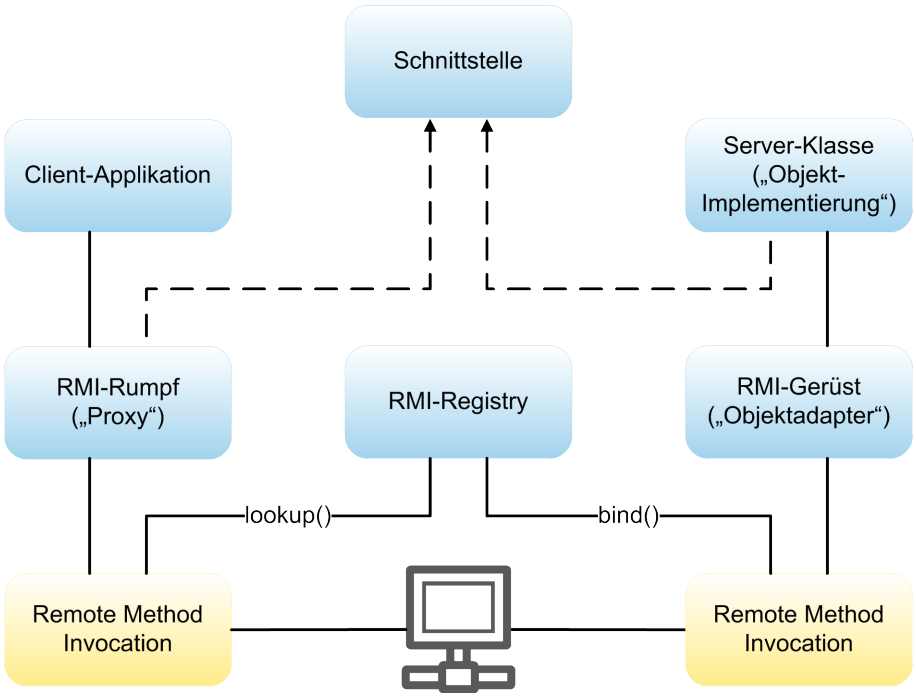


Abbildung 5.1: Überblick über Komponenten und Kommunikation mit RMI nach [Dar05].

deren abstrakte Struktur durch die Schnittstelle definiert und innerhalb der Softwarekomponente implementiert wurde. Die *Client-Applikation* in der Abbildung 5.1 repräsentiert stark vereinfacht die TraBi - Webanwendung, die mit Hilfe der *RMI-Registry* die entfernte Methode lokalisiert und anschließend mit Parametern ausführen kann. Das bedeutet, dass die Identifikation von potenziellen TFBS in Nukleotidsequenzen zwar durch die *Client-Applikation* initiiert wird, aber auf einen anderen Server mittels der Softwarekomponente durchgeführt wird. Dabei ist die Engine der Softwarekomponente ausschließlich für die Durchführung der Identifikation zuständig. Außerdem wurde bei der Implementierung der Engine das Konzept des *Multithreading* berücksichtigt, wodurch die Performanz verbessert wird. Die übrigen zur Kommunikation essentiellen Klassen oder Komponenten werden entweder automatisch durch die *Java Virtual Machine* (JVM) erzeugt oder direkt durch RMI bereitgestellt.

Datenakquisition

Die unterste Schicht bei der Systemarchitektur ist der Datenbeschaffungsbereich (siehe Abbildung 4.13), der verschiedene Daten aus unterschiedlichen molekularbiologischen Datenquellen akquiriert und daraus die notwendigen Daten für TraBi erstellt. Aufgrund der Spezifität der Problemstellung wurde diese Schicht als eigenständige Software-Infrastruktur realisiert, deren Merkmale und Funkti-

onsumfang in Abschnitt 5.2.3.1 beschrieben werden. Die Implementierung der Software-Infrastruktur erfolgte ausschließlich durch die Programmiersprache Java. Die Software-Infrastruktur verfügt über eine verständliche und strukturierte GBO, wobei Synthetica in der Version 2.15.0 als *Look and Feel* (LAF) eingesetzt wurde. Dadurch ist eine unkomplizierte Benutzung der Software-Infrastruktur möglich und die Benutzerfreundlichkeit profitiert ebenfalls davon. Insbesondere der *Parser* und der *Wrapper* sind grundlegende Bestandteile der Software-Infrastruktur, weil diese Softwarekomponenten die wichtigste Funktionalität bereitstellen. Einen direkten Datenzugriff auf die einzelnen molekularbiologischen DB des NCBI (*Gene*, *Sequences* oder *Taxonomy*) bietet der *NCBI Entrez Utilities* Webservice, der in der Version 2.0 verfügbar ist und als Kommunikationsprotokoll ausschließlich SOAP unterstützt. Durch das *Framework* Apache Axis2 wurde der synchrone Datenaustausch zwischen Software-Infrastruktur und Webservice implementiert, der eine Notwendigkeit bei der Datenfusion und -bereinigung durch die Software-Infrastruktur darstellt. In der Dokumentation des NCBI wird eine Kompatibilität bis zur Version 1.4.1 von Apache Axis2 gewährleistet. Allerdings zeigte sich in der Praxis durch ausführliche Testverfahren, dass auch die Version 1.5.6 einwandfrei funktioniert, weshalb diese Version bei der Entwicklung verwendet wurde. Es wurde ebenfalls deutlich, dass der *NCBI Entrez Utilities* Webservice und die Version 1.6.2 von Apache Axis2 inkompatibel sind. Der Datenaustausch und -zugriff zwischen Software-Infrastruktur und DWH wurde durch die spezifische Datenbankschnittstelle MySQL Connector/J ermöglicht, die JDBC 3.0 implementiert und einen Typ-4-Treiber repräsentiert. Der *Wrapper* ist die Softwarekomponente innerhalb der Software-Infrastruktur, welche die entsprechende Funktionalität zur Kommunikation mit einer molekularbiologischen DB oder einem Webservice zur Verfügung stellt. Im Gegensatz dazu ist der *Parser* für das automatische Importieren, Identifizieren und Strukturieren von molekularbiologischen Datenbeständen verantwortlich und erstellt die notwendigen ESA, die existenziell für TraBi sind. Aufgrund der Anforderungen sind für ein Gen mehrere ESA erforderlich, die entweder die 5'-Upstream-Region oder die 3'-Downstream-Region repräsentieren und jeweils auf einer Nukleotidsequenz der Länge 2500 bp, 5000 bp oder 10000 bp basieren. Infolgedessen wurde das Konzept des *Multithreading* beim *Parser* berücksichtigt, weil dieses Konzept vor allem einen positiven Effekt beim Erstellen der ESA bzgl. der Effizienz erzeugt.

5.2 Struktur und Funktionsumfang

Die Struktur und die Funktionalität von DAWIS-M.D. und TraBi wird im Folgenden behandelt. Des Weiteren werden grundlegende Funktionen der Software-Infrastruktur BioDWH vorgestellt, welche für die Verwaltung der molekularbiologischen DB und die Aktualisierung der molekularbiologischen Datenbestände im DWH erforderlich sind. Das DWH wurde mittels BioDWH erstellt und ist für DAWIS-M.D. und TraBi die zentrale Datenbasis. Das Design der jeweiligen Softwarelösung wird

durch Abbildungen veranschaulicht. Im Gegensatz dazu werden signifikante dynamische Aspekte einer Software als Aktivitätsdiagramm dargestellt.

Die Integration von molekularbiologischen DB durch die Software-Infrastruktur BioDWH wird in Abschnitt 5.2.1 erläutert. Das webbasierte Data-Warehouse-System für molekularbiologische Daten, das als DAWIS-M.D. bezeichnet wird, thematisiert der Abschnitt 5.2.2 ausführlich. Abschließend wird in Abschnitt 5.2.3 die spezifische TraBi - Software-Infrastruktur und die TraBi - Webanwendung beschrieben, wobei deren Schwerpunkte und Eigenschaften unterschiedlich sind.

5.2.1 BioDWH

BioDWH ist eine flexible und plattformunabhängige Software-Infrastruktur zur Integration von unterschiedlichen molekularbiologischen DB, die speziell für verschiedene RDBMS entwickelt wurde und deren Anwendungslogik durch die Programmiersprache Java realisiert wurde. Damit BioDWH unterschiedliche RDBMS wie MySQL, PostgreSQL oder Oracle benutzen kann, wurde die Technik der objektrelationalen Abbildung verwendet, die das *Framework* Hibernate zur Verfügung stellt. Dieses *Framework* wurde ebenfalls bei DAWIS-M.D. und TraBi eingesetzt. Die Konfiguration und Administration bei BioDWH basiert auf XML, wodurch eine unkomplizierte Modifikation der einzelnen Parameter möglich ist. Außerdem unterliegt BioDWH der GPL und der Quelltext wird für interessierte Softwareentwickler/Wissenschaftler mittels SourceForge frei zur Verfügung gestellt, sodass eine problemlose und kontinuierliche Pfege/Weiterentwicklung realistisch ist. Mit Hilfe der Software-Infrastruktur wurde das DWH erstellt, das die zentrale Datenbasis für DAWIS-M.D. und TraBi darstellt. Der Anwendungsbereich der Software-Infrastruktur ist die Bioinformatik, aber auch andere Bereiche der Lebenswissenschaften können durch den bereitgestellten Funktionsumfang profitieren. Durch BioDWH ist die Integration von zahlreichen populären molekularbiologischen DB möglich (siehe Tabelle 5.2). Dafür werden spezifische *Parser* hinsichtlich der molekularbiologischen DB bereitgestellt, die durch einen ETL-Prozess die eigentliche Datenintegration ermöglichen. Anhand der Tabelle 5.2 wird deutlich, dass die Integration von derzeit 18 unterschiedlichen molekularbiologischen DB durch BioDWH möglich ist.

Die Software-Infrastruktur bietet eine übersichtliche und intuiive GBO, wobei Synthetica als LAF verwendet wurde. Des Weiteren wird der Benutzer bei der Handhabung der Software durch einen Konfigurationsassistenten unterstützt, dessen Vorgehensweise abstrakt als Aktivitätsdiagramm in der Abbildung 5.2 dargestellt wird. Durch den Konfigurationsassistenten ist die Benutzerfreundlichkeit bei BioDWH besonders hervorzuheben, weil die Konfiguration stark vereinfacht wird. Die Abbildung 5.2 zeigt die einzelnen Schritte der Konfiguration, die für den Prozess der Datenintegration mittels BioDWH notwendig sind. Der erste Schritt beim Konfigurationsassistenten ist die Eingabe der Projektbeschreibung (siehe Abbildung C.11(a)). Danach ist die Eingabe der Datenbankeinstellungen erforderlich, die auf Korrektheit

Datenquelle	Klassifikation	Release	Datum
BRENDA	*Metabolic and Signaling Pathways*	-	08.12.2011
EMBL-Bank	*Nucleotide Sequence Databases*	112	Juni 2012
Ensembl	*Human and other Vertebrate Genomes*	67	10.05.2012
ENZYME	*Metabolic and Signaling Pathways*	-	13.06.2012
EPD	*Nucleotide Sequence Databases*	112	-
GO	*Genomics Databases (non-vertebrate)*	-	Juli 2012
HPRD	*Human and other Vertebrate Genomes*	9	-
IntAct	*Metabolic and Signaling Pathways*	-	17.08.2012
iProClass	*Protein sequence databases*	4.7	05.09.2012
JASPAR	*Nucleotide Sequence Databases*	2009	12.10.2009
KEGG	*Genomics Databases (non-vertebrate), Metabolic and Signaling Pathways*	58.1	01.06.2011
MINT	*Metabolic and Signaling Pathways*	-	05.12.2011
PROSITE	*Protein sequence databases*	20.85	30.08.2012
OMIM	*Human Genes and Diseases*	-	-
Reactome	*Metabolic and Metabolic and*	42	-
SCOP	*Structure Databases*	1.75	Juni 2009
TRANSFAC®	*Nucleotide Sequence Databases, Microarray Data and other Gene Expression Databases*	2009.1	27.03.2009
TRANSPATH®	*Nucleotide Sequence Databases, Metabolic and Signaling Pathways*	-	-
UniProt	*Protein sequence databases*	2012_06	13.06.2012

Tabelle 5.2: Molekularbiologische DB, deren Integration durch BioDWH gewährleistet wird.

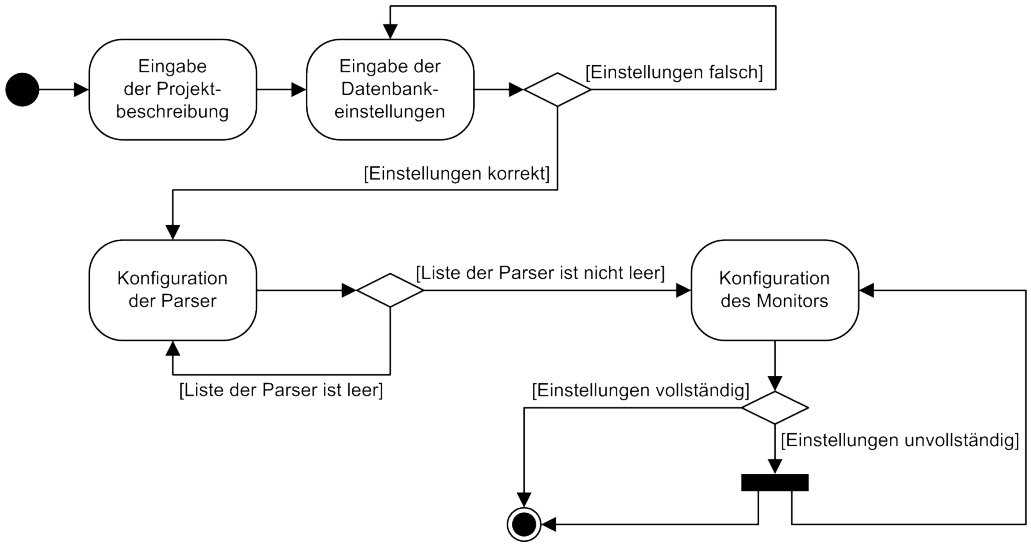


Abbildung 5.2: Konfigurationsassistent von BioDWH als Aktivitätsdiagramm nach [Kor10].

überprüft werden. Der dritte Schritt ist für die Konfiguration der *Parser* zuständig, wobei die Einstellungen validiert werden, sodass mögliche Fehler sofort deutlich werden. Abschließend erfolgt die Konfiguration des Monitors (siehe Abbildung C.11(b)). Sobald diese Konfiguration vollständig und korrekt durchgeführt wurde, kann eine Integration für eine und/oder mehrere molekularbiologische DB initiiert werden. Der Konfigurationsassistent von BioDWH wird detailliert in [Kor10] erläutert.

Der Monitor ist eine spezielle Softwarekomponente innerhalb der Software-Infrastruktur, die den Prozess der Datenintegration kontrolliert und auch neue Versionen der einzelnen Datenquellen identifizieren kann. Sofern eine neue Version verfügbar ist, werden die entsprechenden Textdateien automatisch heruntergeladen und extrahiert. Danach erfolgt die Transformation der Daten und zum Schluss werden die Daten persistent in der DB gespeichert. Dadurch wird die Aktualität der Datenbestände einer Datenbasis, die durch BioDWH erstellt wurde und in der Regel auf unterschiedlichen molekularbiologischen DB basiert, gewährleistet. Diese drei Schritte sind essentielle Bestandteile eines *Parsers* und werden durch den ETL-Prozess repräsentiert, der genau genommen die vollständige Datenintegration durchführt. Allerdings können verschiedene Fehler oder andere Probleme während der Datenintegration auftreten, sodass bei einer fehlerhaften Datenintegration die Konsistenz der Datensätze nicht sichergestellt ist. Deswegen bietet BioDWH zwei optionale Mechanismen für die Datenintegration, die eine Datensicherung und ein *Logging* realisieren. Die Datensicherung erstellt einen Datenbankdump, wodurch die ursprüngliche konsistente Datenbasis problemlos wiederhergestellt werden kann. Durch das *Logging* können fehlerhafte Datensätze oder andere Programmfehler identifiziert und beseitigt werden. Die Konzeption und die Entwicklung sowie die jeweiligen Funktionalitäten von BioDWH werden ausführlich in [TKKH08, Kor10] beschrieben.

5.2.2 DAWIS-M.D. - Data Warehouse Information System for Metabolic Data

Das webbasierte Data-Warehouse-System für molekularbiologische Daten, das als DAWIS-M.D. bezeichnet wird, bietet auf das zugrundeliegende molekularbiologische DWH eine homogene, konsistente und integrierte Sicht. Dieses DWH wurde mittels BioDWH erstellt und beinhaltet Datenbestände aus 13 molekularbiologischen DB (siehe Tabelle 4.1). Aufgrund der Benutzerfreundlichkeit, Flexibilität und Plattformunabhängigkeit wurde DAWIS-M.D. als strukturierte und interaktive Webanwendung realisiert, deren Funktionsumfang und Merkmale im weiteren Verlauf beschrieben werden. Dabei wurde als Entwurfsmuster das *Model 2* als Variante von *Model View Controller* (MVC) für Java-basiertes *Web Engineering* verwendet (siehe Abbildung 5.3). Auf diese Weise wird bei einer Webanwendung eine strikte Tren-

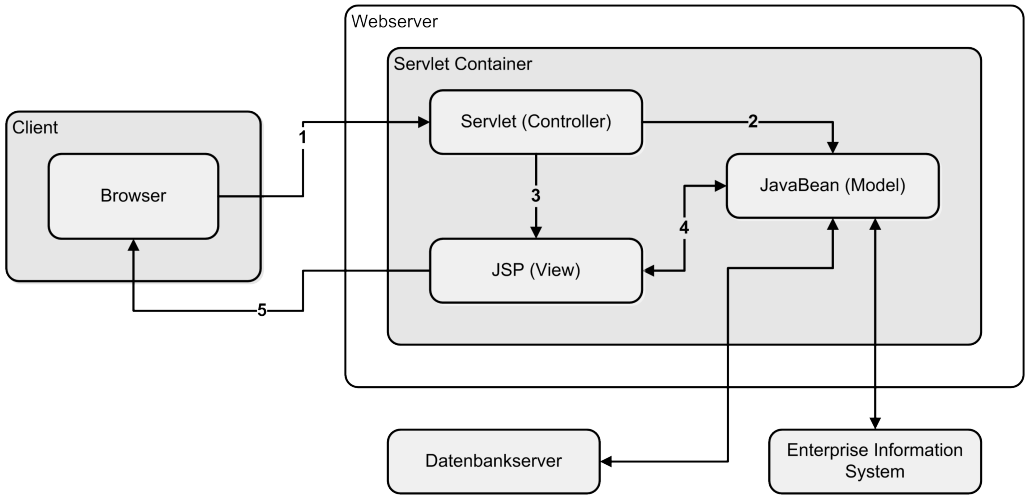


Abbildung 5.3: JSP-Model-2-Architektur nach [KPRR04].

nung der Präsentationsschicht und der eigentlichen Anwendungslogik gewährleistet. Infolgedessen werden bei DAWIS-M.D. spezifische Webseiten basierend auf JSP als auch *JavaBeans* und *Servlets* eingesetzt.

Die Webanwendung stützt sich auf ein abstraktes molekularbiologisches Datenmodell (siehe Abbildung 4.7). Dieses Datenmodell reflektiert explizit die Datenbestände der 13 molekularbiologischen DB und deren Beziehungen/Abhängigkeiten untereinander. Das Datenmodell verfügt über 13 verschiedene Domänen, die spezifische Wissensbereiche der Molekularbiologie wie *Compound*, *Glycan* oder *Transcription Factor* repräsentieren können (siehe Tabelle 4.2).

Konfiguration der Webanwendung

Durch eine Konfigurationsdatei, die auf XML basiert, ist eine systemspezifische Konfiguration der Webanwendung möglich. Die nachträgliche Modifikation der Parameter ist problemlos möglich, weil die Struktur der Konfigurationsdatei nachvollziehbar ist. Ein exemplarisches Beispiel für diese Konfigurationsdatei wird in Quelltext B.3 dargestellt.

Struktur und Design der Webseiten

Die Webseiten der Webanwendung sind ähnlich strukturiert und in drei Bereiche untergliedert, die im Folgenden beschrieben werden:

- Der erste Bereich ist eine einheitliche Kopfzeile, welche bei der Webanwendung eine globale Navigation realisiert. Diese Kopfzeile wird bei allen Webseiten der Webanwendung inkludiert, sodass eine benutzerfreundliche und dynamische Navigation möglich ist. Auf diese Weise ist innerhalb der Webanwendung eine unkomplizierte Navigation zwischen den unterschiedlichen Webseiten sichergestellt. Diese Webseiten sind entweder domänenspezifisch oder für eine andere Systemfunktionalität wie die System- und Benutzerverwaltung optimiert.
- Der mittlere Bereich ist ausschließlich für den eigentlichen Hauptinhalt und die dazu notwendige Funktionalität vorgesehen. Durch einen Vergleich der Abbildungen C.3, 5.5 und 5.6 wird ersichtlich, dass der mittlere Bereich nicht statisch ist, sondern einer besonderen Variabilität unterliegt. Insbesondere die unterschiedlichen Anforderungen, Aufgabenbereiche und Datenbestände sind dafür verantwortlich.
- Abschließend wird durch eine einheitliche Fußzeile, die lediglich das *Copyright* darstellt, das Ende der jeweiligen Webseite festgelegt.

Mit Hilfe der Abbildung C.3, welche die Startseite der Webanwendung zeigt, ist die generelle Struktur der Webanwendung und deren Webseiten nachvollziehbar.

Suchverfahren

Aufgrund der Spezifität der jeweiligen Domänen und deren Datenbestände wird für jede Domäne ein eigenständiges und spezielles Suchformular mit Autovervollständigung und unterschiedlichen Filter- und Suchmöglichkeiten bereitgestellt. Eine detaillierte Übersicht der verfügbaren domänenspezifischen Suchformulare und deren Filter- und Suchmöglichkeiten wird in der Tabelle 5.3 dargestellt. Eine Suche für Attribute wie Identifikationsnummer und Name wird bei allen Suchformularen gewährleistet (siehe Tabelle 5.3). Außerdem verfügen die Suchformulare der beiden

Domäne	Attribute	Filter
Compound	Identifikationsnummer, Name	-
Disease	Identifikationsnummer, Name, Symptome	-
Drug	Identifikationsnummer, Name, Aktivität	-
Enzyme	Identifikationsnummer, Name, Klasse	-
Gene	Identifikationsnummer, Name, Definition, Schlüsselwörter	Organismus
Gene Ontology	Identifikationsnummer, Name, Synonyme	-
Genome	Identifikationsnummer, Name, Schlüsselwörter	-
Glycan	Identifikationsnummer, Name	-
Pathway	Identifikationsnummer, Name, Klasse, Organismus	-
Protein	Identifikationsnummer, *Accession number*, Name, Schlüsselwörter	Organismus
Reaction	Identifikationsnummer, Name	-
Reactant Pair	Identifikationsnummer, Name	-
Transcription Factor	Identifikationsnummer, Name	-

Tabelle 5.3: Filter- und Suchmöglichkeiten der einzelnen Domänen bei DAWIS-M.D.

Domänen *Gene* und *Protein* über eine Organismus spezifische Filterung und Reduzierung der Ergebnismenge. Dadurch werden ausschließlich solche Datensätze bei der Ergebnismenge berücksichtigt, die das zuvor definierte Kriterium erfüllen. Das Design, die Struktur und der Funktionsumfang der domänenspezifischen Suchformulare ist identisch, weshalb in Abbildung 5.4 exemplarisch das Suchformular der Domäne *Protein* dargestellt wird. Anhand der Abbildung 5.4 wird deutlich, dass für jedes Attribut ein eigenes Suchfeld, das mit einer Autovervollständigung assoziiert ist, zur Verfügung gestellt wird. Durch diese Autovervollständigung werden für die Benutzereingaben potenzielle Vervollständigungen angezeigt, sodass eine automatische und sinnvolle Ergänzung erfolgen kann. In der Abbildung 5.4 wird die Funktionalität der Autovervollständigung für ein Suchfeld veranschaulicht. Darüber hinaus werden für alle Suchfelder mindestens drei charakteristische Beispiele bereitgestellt, die als Hilfestellung bei einer Suche fungieren können. Die Datenbestände, die aus unterschiedlichen molekularbiologischen DB resultieren, wurden auf die entsprechenden Domänen aufgeteilt (siehe Tabelle 4.2). Daraus ergibt sich, dass ein domänenspezifisches Suchformular für Datenbestände einer Domäne zuständig ist, deren Ursprung verschiedene molekularbiologische DB sein können. Deswegen verfügen die einzelnen

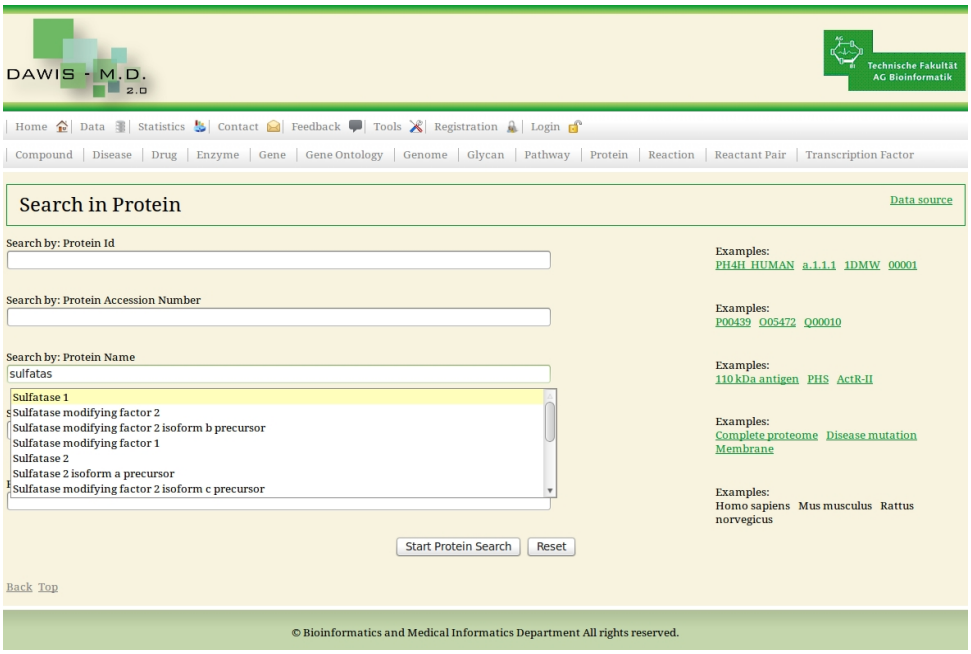


Abbildung 5.4: Suchformular für die Domäne *Protein* bei DAWIS-M.D.

Suchformulare der Domänen über eine Eigenschaft, die als *Data source* bezeichnet wird, wodurch der Anwender die ursprüngliche Datenquelle ermitteln kann. Auf diese Weise wird die Transparenz im Bezug auf die Datenintegration etwas aufgelockert. Die grundsätzliche Durchführung einer Suche wird als nächstes durch das Aktivitätsdiagramm in der Abbildung 5.5 erläutert. Das Anwendungsfalldiagramm

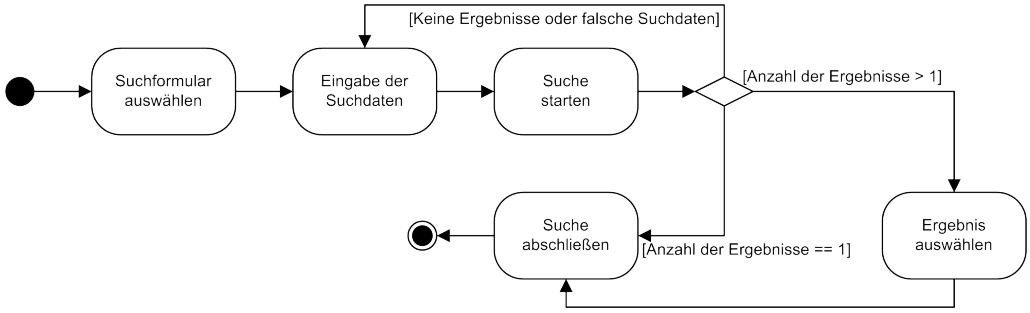


Abbildung 5.5: Suche bei DAWIS-M.D. als Aktivitätsdiagramm.

in der Abbildung 4.2 beinhaltet einen Anwendungsfall `Suche durchführen`, deren einzelne Verarbeitungsschritte das Aktivitätsdiagramm in der Abbildung 5.5 zeigt. Dieses Diagramm besteht aus fünf Aktivitäten, einen Start- und Endpunkt sowie einer Entscheidungsaktivität, die durch eine Raute symbolisiert wird. Als erstes selektiert der Anwender ein domänenspezifisches Suchformular, das für seine Suche geeignet ist. Danach kann der Anwender die eigentlichen Suchdaten in das dafür vorgesehene Suchfeld eingeben und die Suche starten. Die Entscheidungsaktivität ist mit drei Bedingungen assoziiert, wobei die erste Bedingung erfüllt ist, wenn

keine Suchergebnisse existieren oder die Suchdaten nicht mit dem Suchfeld übereinstimmen. Infolgedessen müssen die Suchdaten korrigiert und die Suche wiederholt werden. Eine Ergebnismenge, die ursprünglich aus einer Suchanfrage resultiert, ist Voraussetzung für die anderen beiden Bedingungen. Mit Hilfe der Abbildung 5.5 wird deutlich, wann jeweils die zweite und die dritte Bedingung erfüllt ist. Sofern die zweite Bedingung vorliegt, ist es notwendig, dass der Anwender ein Ergebnis aus der Ergebnismenge selektiert, das der Suchanfrage entspricht. Im Gegensatz dazu wird das Ergebnis bei der dritten Bedingung automatisch ausgewählt, weil die Ergebnismenge lediglich ein Ergebnis beinhaltet. Die Suche wird durch die letzte Aktivität beendet und der Kontroll- und Objektfluss erreicht den Endpunkt, weshalb die gesamte Aktivität als abgeschlossen gilt.

Struktur und Darstellung der Informationen

Sobald die Suche erfolgreich durchgeführt und beendet wurde, erfolgt eine automatische Weiterleitung auf eine spezialisierte Webseite, die detailliert den Datensatz und deren Informationen zeigt. Diese Webseiten sind zum einen domänenspezifisch, zum anderen spezialisiert auf die molekularbiologische DB, die als Datenquelle fungierte. Allerdings verfügen diese Webseiten über eine einheitliche Struktur, dasselbe Design und einen standardisierten Funktionsumfang. Ein Beispiel für eine solche Webseite wird in der Abbildung 5.6 dargestellt. Es sind zwei Ursachen für diese Vorgehenweise bei DAWIS-M.D. ausschlaggebend, wobei die erste Ursache auf das verwendete Paradigma bei der Datenintegration zurückzuführen ist. Durch die lose Kopplung verfügt jede Datenquelle innerhalb des DWH über ein eigenes Datenbankschema, sodass die Schematransformation und -integration auf der Ebene der Anwendungslogik erfolgt. Die vollständige Transparenz ist häufig das wichtigste Ziel bei der Datenintegration [LN07]. Die zweite Ursache ist der Grad der Transparenz, wodurch Aspekte wie Informationsverlust und -qualität stark beeinflusst werden. Insbesondere die ursprüngliche Datenquelle der Datenbestände ist das wichtigste Kennzeichen für die Einschätzung der Informationsqualität. Deswegen wurde bei DAWIS-M.D. keine maximale Transparenz angestrebt, sondern der Grad der Transparenz verringert. Das Risiko eines Informationsverlustes wird auf diese Weise minimiert und die ursprüngliche molekularbiologische DB der Datenbestände kann problemlos nachvollzogen werden. Anhand der Abbildung 5.6 wird deutlich, dass der Datensatz und dessen Informationen durch eine zweispaltige Tabellenform dargestellt wird. Dabei werden in der ersten Spalte die Schlüsselwörter (*Gene Symbol*, *Alternate Names* und *Expression*) und in der zweiten Spalte die dazugehörigen Informationen abgebildet. Außerdem werden die Informationen der Schlüsselwörter wie *Disease*, *Gene Ontology*, *Gene* und *Protein*, die genau genommen Querverweise zu anderen Datensätzen symbolisieren, extra hervorgehoben. Das zusätzliche Hervorheben der Querverweise wird ebenfalls durch die Abbildung 5.6 veranschaulicht.

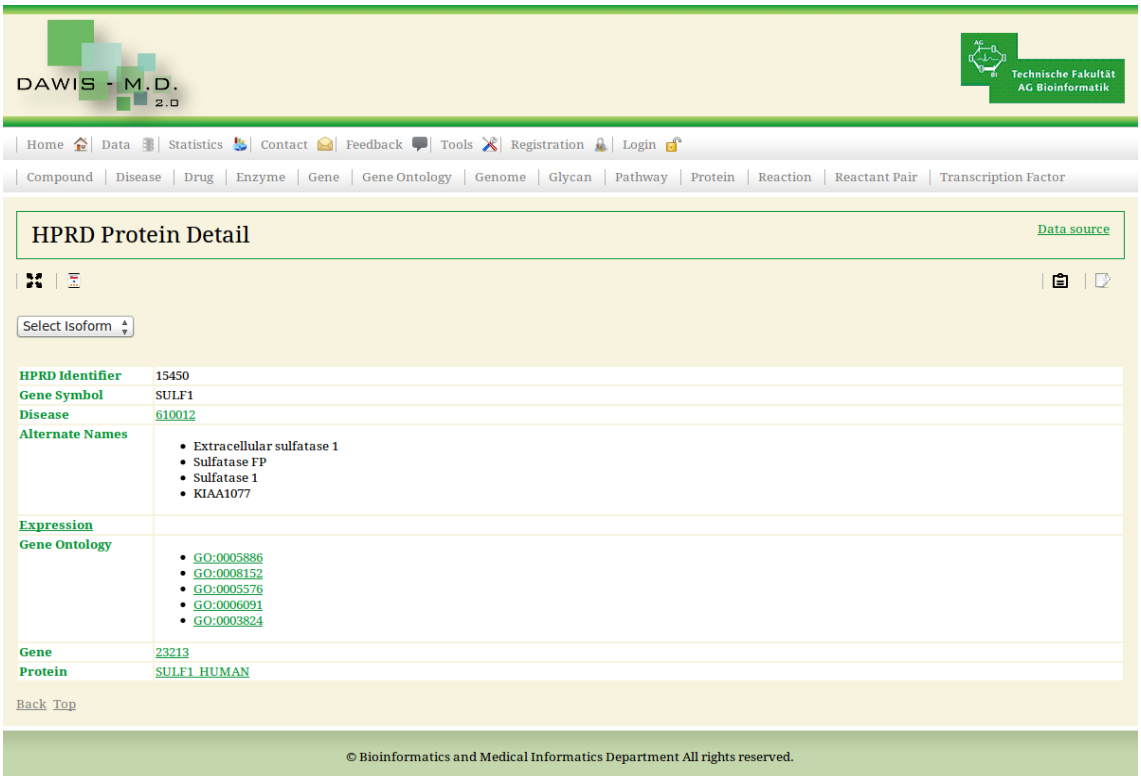


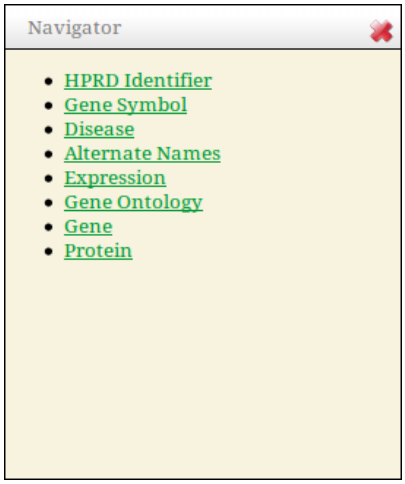
Abbildung 5.6: Beispiel für eine Webseite bei DAWIS-M.D., die einen Datensatz und deren Informationen detailliert darstellt.

Navigationsleiste

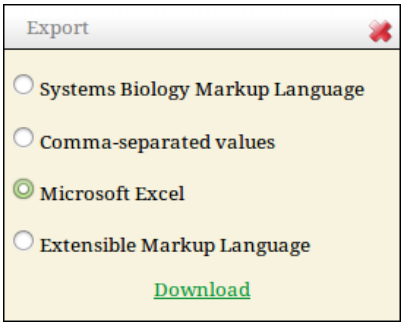
Es gibt einige Datensätze bei DAWIS-M.D., die sehr umfangreich und präzise sind, weswegen jede Webseite, die einen Datensatz detailliert zeigt, über eine lokale Navigation verfügt und das Ein- und Ausblenden von Informationen ermöglicht. Die lokale Navigation für die Abbildung 5.6 wird in der Abbildung 5.7(a) dargestellt. Die Schlüsselwörter eines Datensatzes wie *Gene Symbol* oder *Alternate Names* fungieren bei der lokalen Navigation als Sprungmarke/Anker, wodurch eine unkomplizierte und schnelle Navigation innerhalb der Webseite ermöglicht wird.

Querverweise zwischen Domänen

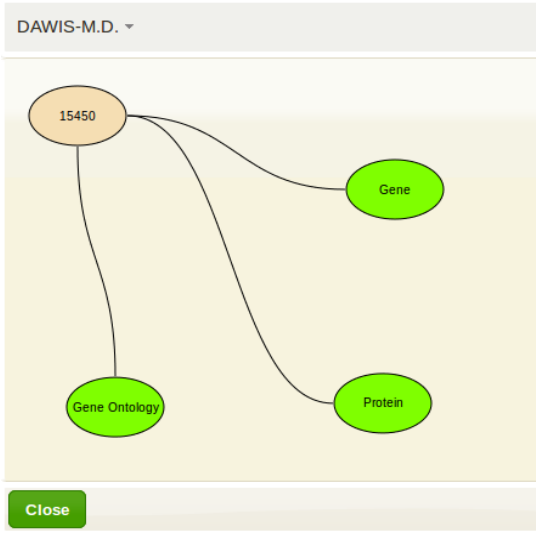
Die Querverweise symbolisieren Beziehungen und/oder Abhängigkeiten zwischen den einzelnen Domänen und deren Datensätzen, die wiederum PPI oder andere regulatorische Wechselwirkungen repräsentieren können. Diese Informationen sind die Grundlage zur automatischen Rekonstruktion von molekularbiologischen Netzwerken durch spezialisierte Anwendungssoftware wie VANESA, VANTED oder Cytoscape. Durch eine zusätzliche Oberfläche, die Abbildung 5.7(b) zeigt, können diese Informationen eines Datensatzes in standardisierte Austauschformate wie CSV, Mi-



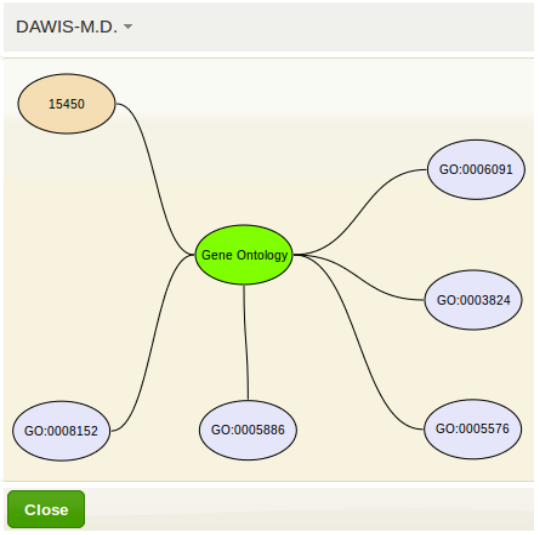
(a) Beispiel für eine lokale Navigation.



(b) Oberfläche für den Export der Daten in standardisierte Austauschformate.



(c) Wurzelknoten (*HPRD Identifier* 15450) und deren Kindknoten (*Gene*, *Gene Ontology* und *Protein*), die wiederum Knoten beinhalten.



(d) Wurzelknoten (*HPRD Identifier* 15450) und einer seiner Kindknoten (*Gene Ontology*), wobei dessen Knoten expandiert sind.

Abbildung 5.7: Spezialisierte Oberflächen bei DAWIS-M.D. für die lokale Navigation, die Netzwerkvisualisierung und den Datenexport.

crosoft Excel, SBML und XML exportiert werden. Die Konsistenz und die Syntax der SBML-Dateien wird durch die Programmbibliothek/Programmierschnittstelle JSBML [DRD+11] in der Version 0.8 gewährleistet.

Netzwerkvisualisierung

Des Weiteren kann jeder Datensatz als dynamisches und interaktives Netzwerk visualisiert werden, wobei ausschließlich die Informationen der Querverweise zwischen den einzelnen Domänen und deren Datensätze berücksichtigt werden. Dafür wird ein Beispiel in den beiden Abbildungen 5.7(c) und 5.7(d) dargestellt. Dieses Beispiel basiert auf dem Datensatz, den Abbildung 5.6 zeigt. Die Komponentenbibliothek PrimeFaces[3] in der Version 3.4.1 ermöglicht die Netzwerkvisualisierung. Das Netzwerk in der Abbildung 5.7(c) besteht aus vier Knoten und drei Kanten. Dabei fungiert der *HPRD Identifier* 15450 als Wurzelknoten und die übrigen Knoten (*Gene*, *Gene Ontology* und *Protein*) sind deren Kindknoten, die wiederum Knoten beinhalten. Sobald einer der Knoten (*Gene*, *Gene Ontology* und *Protein*) expandiert wird, werden deren Kindknoten dargestellt. Dies zeigt Abbildung 5.7(d) exemplarisch für den Knoten *Gene Ontology*. Sofern die durch DAWIS-M.D. bereitgestellte Netzwerkvisualisierung und deren Funktionalität für den Anwender ausreichend sind, müssen keine der oben genannten externen Softwarelösungen verwendet werden.

Systemfunktionalitäten und -eigenschaften

Darüber hinaus gibt es weitere charakteristische Systemfunktionalitäten und -eigenschaften, wodurch ausgewählte Informationen der unterschiedlichen Datensätze übersichtlich und verständlich dargestellt und explizit hervorgehoben werden. Diese zusätzlichen Merkmale für die Gestaltung der Informationen und/oder andere Funktionen der Webseiten werden in der folgenden Auflistung erläutert:

- Durch eine Sprungmarke, die auch als Anker bezeichnet wird, kann der Anwender zwischen zwei Sektionen auf der Webseite, die im voraus extra festgelegt wurden, schnell und unkompliziert navigieren. Des Weiteren verfügen einige Webseiten über einen internen Verlauf, sodass der Anwender seine zuletzt besuchten Webseiten ohne großen Aufwand aufrufen kann. Diese Funktion wird durch die Skriptsprache JavaScript realisiert.

- Das FASTA-Format ist das standardisierte Dateiformat zur strukturierten Darstellung und Speicherung von Aminosäure- und Nukleotidsequenzen in einer Textdatei. Insbesondere Aminosäure- und Nukleotidsequenzen, die eine bestimmte Länge überschreiten werden, als Textdatei im FASTA-Format

[3] http://primefaces.org/

gespeichert und als *Download* bereitgestellt. Diese Textdateien kann der Anwender entweder in einen separaten Tab des Webbrowsers betrachten oder herunterladen und mit einer externen Anwendungssoftware öffnen. Der *Download* wird auf den Webseiten durch ein spezielles Symbol und eine farbliche Kennzeichnung hervorgehoben.

- Eine PSSM wird als strukturierte Textdatei gespeichert und der Anwender kann mittels *Download* diese Textdatei von der Webseite herunterladen. Außerdem werden die PSSM auf der Webseite als Sequenzlogo dargestellt, wodurch die Verteilung der Nukleotide auf die jeweiligen Positionen und die Konsenussequenz durch eine Grafik veranschaulicht werden. Dabei wird die *core region* der PSSM ebenfalls identifiziert und beim Sequenzlogo durch eine farbliche Markierung extra hervorgehoben.

- Die Strukturformel einer chemischen Substanz und einige Stoffwechselwege werden zusätzlich durch Grafiken veranschaulicht, wodurch der Anwender deren Zusammensetzung besser nachvollziehen kann.

- Durch einen entsprechenden *Hyperlink* auf der Webseite, der unter *Data source* verfügbar ist, kann der Anwender das Original des Datensatzes bei der ursprünglichen molekularbiologischen DB direkt aufrufen und betrachten. Auf diese Weise wird für jeden Datensatz die ursprüngliche Datenquelle deutlich. Darüber hinaus kann der Anwender den Datensatz, der bei DAWIS-M.D. dargestellt wird, mit dem Original vergleichen und potenzielle Fehler und/oder Unstimmigkeiten identifizieren.

System- und Benutzerverwaltung

Die System- und Benutzerverwaltung, Registrierung, Benutzeranmeldung und -abmeldung sind weitere funktionale Anforderungen für DAWIS-M.D., die in Abschnitt 4.2 definiert wurden.

Durch die System- und Benutzerverwaltung kann ausschließlich der Administrator die Software administrieren und konfigurieren sowie Benutzer anlegen, löschen oder das Benutzerkonto und deren Benutzerrolle editieren. Die Registrierung und die Benutzeranmeldung und -abmeldung sind elementare Bestandteile der Benutzerverwaltung und realisieren eine wichtige Funktionalität. Infolgedessen kann der Anwender die eigentliche Registrierung und/oder Benutzeranmeldung und -abmeldung bei DAWIS-M.D. selbstständig durchführen. Die Webseite für die System- und Benutzerverwaltung wird in der Abbildung C.1 dargestellt.

Eine Möglichkeit zur direkten Kommunikation zwischen Anwender und *Support* ist bei DAWIS-M.D. ebenfalls gegeben und wird als *Feedback* bezeichnet. Auf diese Weise bekommt der *Support* schnell und unkompliziert Informationen über fehlerhafte Datensätze oder mögliche Programmfehler sowie Verbesserungsvorschläge und

Rezensionen der Anwender.

Statistik und Informationen der integrierten molekularbiologischen DB

Damit die Aktualität, Verteilung und Zusammensetzung der Datenbestände sowie das *Release* der Datenquelle nachvollzogen werden können, sind eine Statistik und zusätzliche Informationen der molekularbiologischen DB, die als Datenquellen fungierten, verfügbar. Mit Hilfe der Programmbibliothek/Programmierschnittstelle JFreeChart in der Version 1.0.14 werden verschiedene Kreis- und Balkendiagramme bei der Statistik bereitgestellt. Die beiden Abbildungen C.2 und C.4 zeigen die entsprechenden Webseiten.

5.2.3 TraBi - Transcription Factor Binding Site Prediction

Die Software, die als TraBi bezeichnet wird, repräsentiert eigentlich zwei Softwarelösungen, die jeweils in den folgenden beiden Abschitten thematisiert werden. Als erstes wird in Abschnitt 5.2.3.1 die TraBi - Software-Infrastruktur erläutert. Diese Softwarelösung fungiert ausschließlich als Hilfsmittel für die erforderlichen Datenbestände, die für eine korrekte Funktionsweise der TraBi - Webanwendung und deren Funktionalität benötigt werden. Abschließend behandelt der Abschnitt 5.2.3.2 die interaktive und dynamische TraBi - Webanwendung, die als zentrale Software angesehen werden kann und einen benutzerfreundlichen Funktionsumfang zur computergestützten Identifikation von potenziellen TFBS in Nukleotidsequenzen bereitstellt.

5.2.3.1 TraBi - Software-Infrastruktur

Die TraBi - Software-Infrastruktur wurde mittels der Programmiersprache Java realisiert, wodurch die Plattformunabhängigkeit gewährleistet wird. Diese Software-Infrastruktur unterstützt *Multithreading* und wurde explizit für Mehrkernprozessoren optimiert. Dadurch wird die Zeitkomplexität bei der Datenverarbeitung reduziert. Ein weiteres Merkmal ist die benutzerfreundliche GBO, die Synthetica als LAF verwendet. Die Technik der objektrelationalen Abbildung wurde ebenfalls berücksichtigt und mittels Hibernate realisiert. Infolgedessen können verschiedene RDBMS wie MySQL, PostgreSQL oder Oracle verwendet werden, ohne die Anwendungslogik der TraBi - Software-Infrastruktur zu modifizieren.

Funktionsumfang zur Datenakquisition, -bereinigung und -fusion

Der spezielle Funktionsumfang zur Datenakquisition, -bereinigung und -fusion, der durch die TraBi - Software-Infrastruktur bereitgestellt wird, wurde explizit für die Datenbestände konzipiert, welche für die Funktionalität der TraBi - Webanwendung

existenziell sind. Das bedeutet, dass die spezifischen Datenbanktabellen und deren unterschiedlichen Datensätze, die bei der TraBi - Webanwendung notwendig sind, durch die TraBi - Software-Infrastruktur angelegt werden. Dabei werden hauptsächlich Datenbestände über Gene, Organismen, TF, TFBS und PSSM in die dafür vorgesehenen Datenbanktabellen der DB *metadata*, die ein Bestandteil des DWH ist, gespeichert. Außerdem kann der Anwender durch diese Softwarelösung die erforderlichen ESA erstellen, die eine Grundvoraussetzung für den Algorithmus *ESAsearch* sind. Der Algorithmus *ESAsearch* wurde in der vorliegenden Arbeit erweitert und ermöglicht die Identifizierung von potenziellen TFBS in Nukleotidsequenzen, welche durch die TraBi - Webanwendung initiiert wird. Die ESA werden nicht in einer der beiden DB des DWH gespeichert, weil eine solche Vorgehensweise die Effizienz und die Komplexität einer Software und deren Algorithmik negativ beeinflussen würde. Stattdessen speichert die TraBi - Software-Infrastruktur die ESA als Datei im Dateisystem. Demzufolge fungieren die DB *metadata* oder eine andere DB und die ESA, die jeweils durch die TraBi - Software-Infrastruktur angelegt werden, als Datenbasis für die TraBi - Webanwendung. Eine nachträgliche Aktualisierung und Erweiterung der Datenbasis ist ebenfalls durch die TraBi - Software-Infrastruktur problemlos möglich. Auf diese Weise wird die Informationsqualität und -aktualität sichergestellt.

Konfiguration der Datenbankeinstellungen

Damit der vollständige Funktionsumfang der TraBi - Software-Infrastruktur einwandfrei funktioniert, müssen als erstes die Datenbankeinstellungen für das zugrundeliegende molekularbiologische DWH und der Proxy konfiguriert werden. Dafür wird eine spezifische Oberfläche zur Verfügung gestellt (siehe Abbildung 5.8). Die

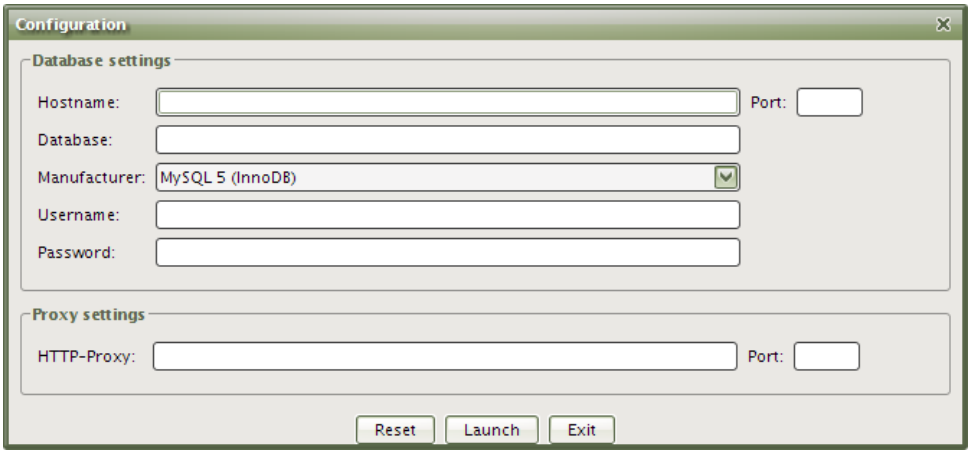


Abbildung 5.8: Oberfläche zur Konfiguration bei der TraBi - Software-Infrastruktur.

jeweiligen Einstellungen für die einzelnen Parameter wie Benutzername, Passwort und *Host* werden in einer Konfigurationsdatei gespeichert. Diese Konfigurationsdatei

basiert auf XML und wird in Quelltext B.4 dargestellt. Aufgrund der verständlichen Syntax der Konfigurationsdatei ist eine nachträgliche Modifikation der Parameter möglich.

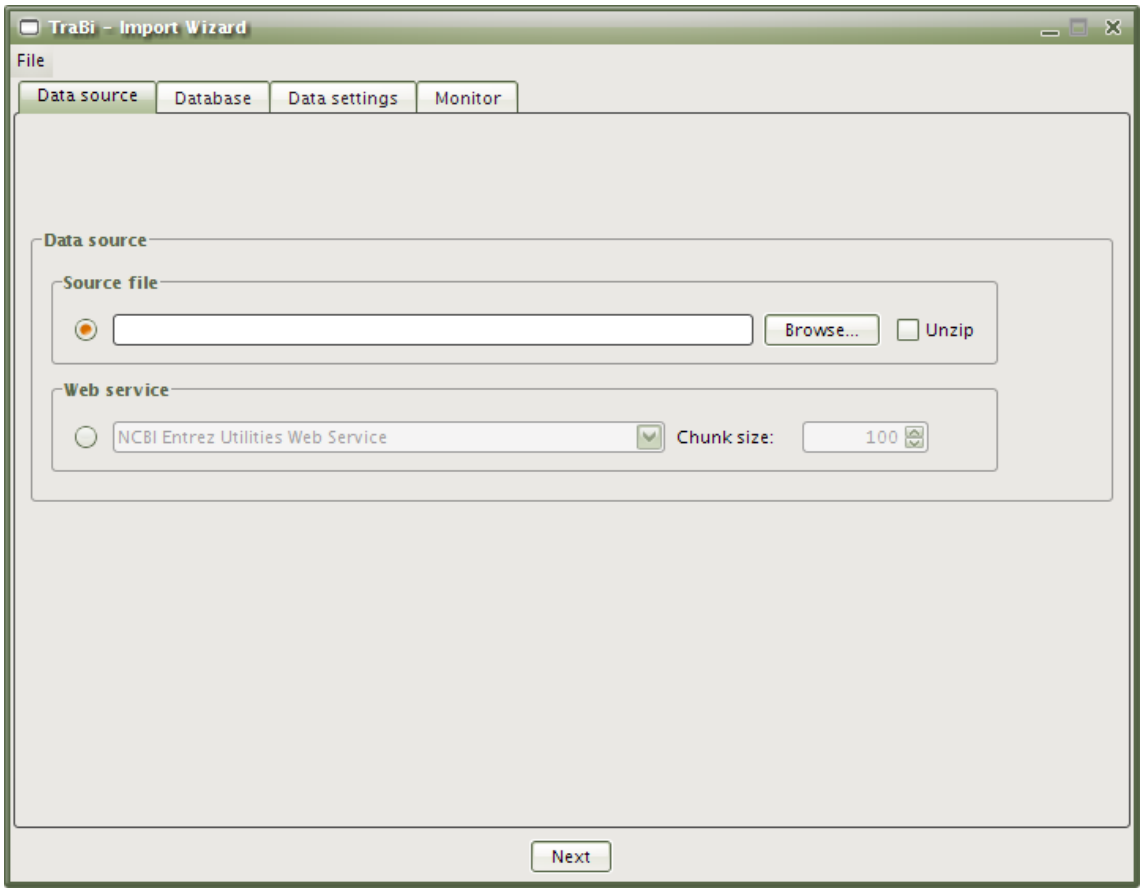
Die Initialisierung der TraBi - Software-Infrastruktur erfolgt, wenn der Anwender die Konfiguration vollständig und fehlerfrei durchführt. Während der Initialisierung wird automatisch eine Textdatei im Benutzerverzeichnis angelegt, die für eine korrekte Funktionsweise der TraBi - Software-Infrastruktur notwendig ist. Diese Textdatei wird exemplarisch in Quelltext B.1 dargestellt und beinhaltet für zahlreiche Organismen die eindeutige Identifikationsnummer der Taxonomie und die etablierte Schreibweise eines Organismus. Danach kann die TraBi - Software-Infrastruktur und deren Funktionalität ohne Beschränkung benutzt werden.

Struktur und Design des Konfigurationsassistenten

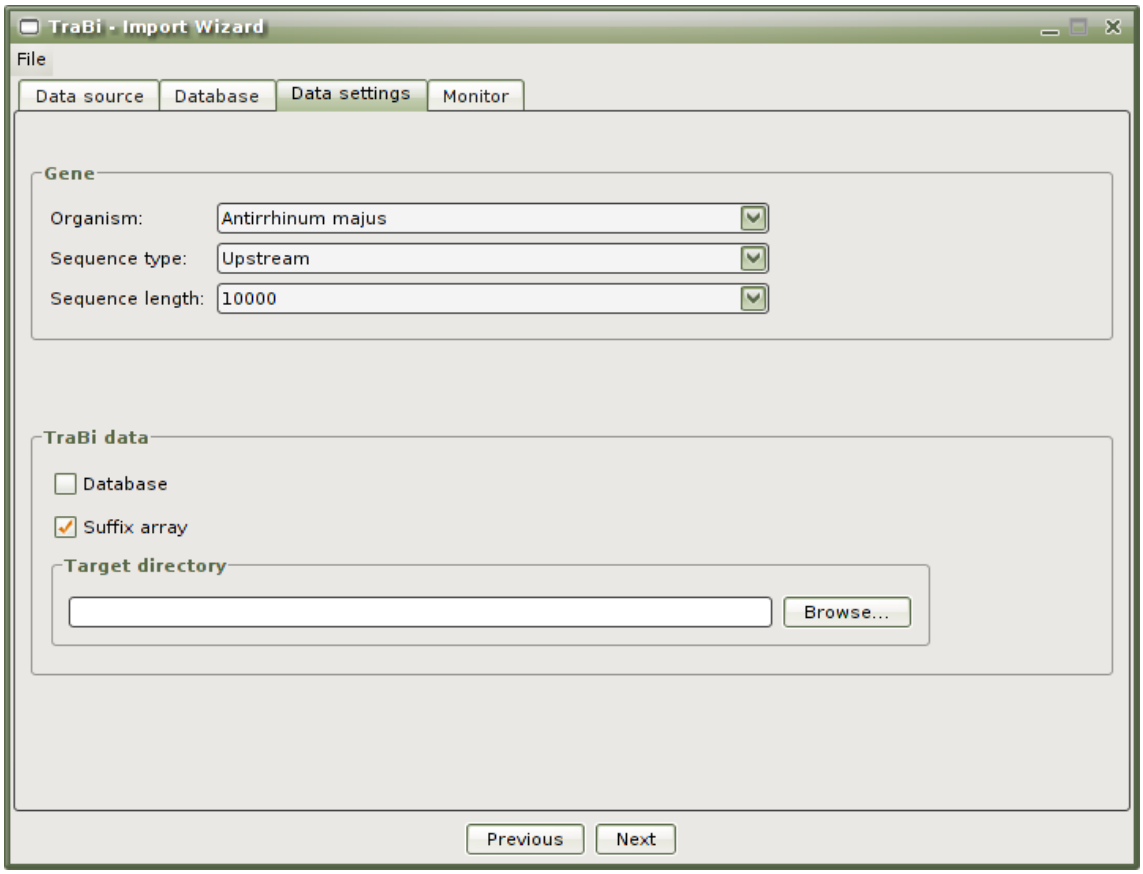
Die Struktur und das Design der GBO werden durch die beiden Abbildungen 5.9(a) und 5.9(b) veranschaulicht. Anhand der beiden Abbildungen wird deutlich, dass die GBO aus vier Registerkarten besteht. Dabei zeigt Abbildung 5.9(a) die erste Registerkarte, die verschiedene Einstellungen zur Konfiguration der jeweiligen Datenquelle zur Verfügung stellt. Die Einstellungen der Daten können durch die dritte Registerkarte konfiguriert werden, die in der Abbildung 5.9(b) dargestellt wird. Die vier Registerkarten sind Bestandteile eines Konfigurationsassistenten, wodurch der Anwender den Prozess zur Datenakquisition, -bereinigung und -fusion konfigurieren kann. Davon profitiert vor allem die Benutzerfreundlichkeit der TraBi - Software-Infrastruktur, weil auf diese Weise die Konfiguration interaktiv erfolgt und stark vereinfacht wird.

Durch das Aktivitätsdiagramm in der Abbildung 5.10 werden im weiteren Verlauf der Konfigurationsassistent und deren einzelnen Schritte erläutert. Dieses Diagramm besteht aus fünf Aktivitäten, einen Start- und Endpunkt sowie einer Entscheidungsaktivität.

Als erstes ist es notwendig, dass eine Datenquelle ausgewählt und konfiguriert wird, wobei entweder eine Textdatei im FASTA-Format oder der *NCBI Entrez Utilities Web Service* als Datenquelle fungieren kann. Allerdings muss diese Textdatei eine bestimmte Struktur aufweisen, die exemplarisch für die beiden Gene SULF1 und SULF2 in Quelltext B.2 dargestellt wird. Eine solche Textdatei kann mittels BioMart für verschiedene Organismen und deren Gene erstellt werden. Die Extraktion der erforderlichen genetischen Informationen eines Organismus kann jeweils durch eine der beiden Datenquellen erfolgen. Dabei werden in Bezug auf ein Gen die eindeutige Ensembl Identifikationsnummer, der Name, die Beschreibung, die exakte Position im Genom und die jeweilige Nukleotidsequenz der 5'-Upstream-Region oder der 3'-Downstream-Region berücksichtigt. Im Gegensatz dazu wird das zugrundeliegende molekularbiologische DWH für die Extraktion der Datenbestände über TF, TFBS und PSSM benötigt. Diese Datenbestände resultieren ursprünglich aus den beiden



(a) Konfiguration der Datenquelle.



(b) Konfiguration der Dateneinstellungen.

Abbildung 5.9: Konfigurationsassistent bei der TraBi - Software-Infrastruktur.

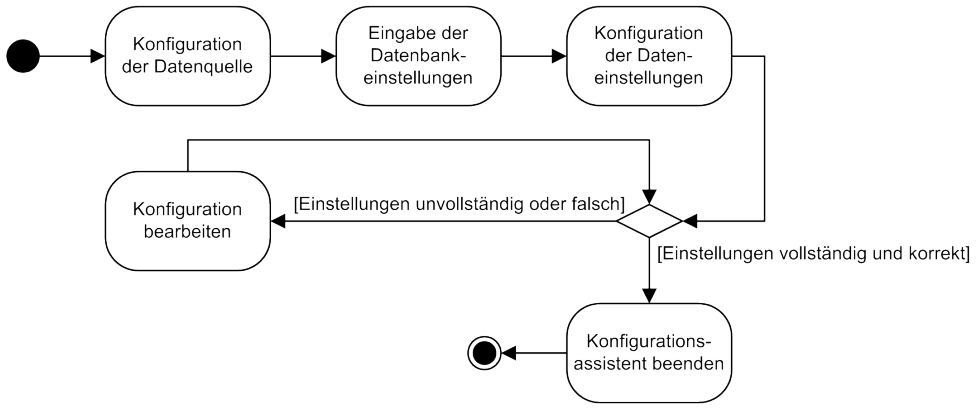


Abbildung 5.10: Konfigurationsassistent bei der TraBi - Software-Infrastruktur als Aktivitätsdiagramm.

Datenquellen TRANSFAC® und JASPAR. Sofern der Webservice als Datenquelle fungiert, werden die genetischen Informationen in eine temporäre Datei zwischengespeichert, deren Syntax äquivalent mit Quelltext B.2 ist.

Der zweite Schritt des Konfigurationsassistenten ist für die Eingabe der Datenbankeinstellungen (Benutzername, Passwort und Port) zuständig. Die entsprechenden Parameter sind standardmäßig für die DB *metadata* konfiguriert. Allerdings können diese Datenbankeinstellungen problemlos geändert werden. Außerdem können durch das *Framework* Hibernate verschiedene DBS, die auf ein RDBMS basieren, eingesetzt werden.

Danach müssen die Einstellungen der Daten konfiguriert werden (siehe Abbildung 5.9(b)). Anhand Abbildung 5.9(b) ist zu erkennen, dass ein Organimus (*Homo sapiens*, *Mus musculus*, *Rattus norvegicus* oder andere Organismen), der Typ der Nukleotidsequenz (5'-Upstream-Region oder 3'-Downstream-Region) und deren Länge (10000 bp) im voraus durch den Anwender festgelegt werden müssen. Diese Metadaten werden für die weitere Verarbeitung der genetischen Datenbestände als auch für deren automatische Extraktion durch den *NCBI Entrez Utilities Web Service* benötigt. Ein geeignetes Verzeichnis zur Speicherung der ESA sollte ebenfalls ausgewählt werden, weil diese Datenstruktur nicht in einer DB, sondern als Datei im Dateisystem gespeichert wird.

Abschließend werden alle benutzerspezifischen Einstellungen durch den Konfigurationsassistenten validiert, sodass fehlerhafte und/oder unzureichende Einstellungen sofort nachzuvollziehen sind. Die Entscheidungsaktivität ist mit zwei Bedingungen assoziiert, wobei die erste Bedingung erfüllt ist, wenn die Einstellungen der Konfiguration unvollständig oder falsch sind. Anschließend ist eine Korrektur der Konfiguration durch den Anwender notwendig, die abermals validiert wird. Die zweite Bedingung ist erfüllt, wenn alle Einstellungen der Konfiguration vollständig und korrekt sind. Dann wird der Konfigurationsassistent erfolgreich beendet und der Kontroll- und Objektfluss erreicht den Endpunkt, weshalb die gesamte Aktivität als

abgeschlossen gilt.

Prozess zur Datenakquisition, -bereinigung und -fusion

Der Anwender kann als nächstes den eigentlichen Prozess zur Datenakquisition, -bereinigung und -fusion initiieren, der unter gewissen Aspekten einen ETL-Prozess ähnelt. Als erstes erfolgt die Extraktion der relevanten Daten aus den unterschiedlichen Datenquellen (Textdatei oder Webservice und DWH), die im weiteren Verlauf in ein einheitliches Datenformat und -schema transformiert werden. Mit Hilfe der Datenfusion und -bereinigung werden lückenhafte Datensätze zusammengeführt und vervollständigt als auch Duplikate und Inkonsistenzen identifiziert und beseitigt. Auf diese Weise werden einige Datensätze, die ursprünglich aus der Datenquelle TRANSFAC® oder JASPAR resultieren, um eine Konsensussequenz und/oder eine eindeutige Identifikationsnummer für die Taxonomie ergänzt. Der Name der Organismen und deren Syntax wird ebenfalls einheitlich strukturiert, sodass für diese Datensätze eine Organismus spezifische Klassifikation durchgeführt werden kann. Das NCBI ist verantwortlich für verschiedene molekularbiologischen DB (*Gene*, *Taxonomy* und *Sequences*). Diese Datenquellen und deren Informationen werden bei der Datenfusion und -bereinigung der Datensätze involviert, weshalb der *NCBI Entrez Utilities Web Service* konsultiert wird. Dafür bietet der Webservice einen direkten und unkomplizierten Datenzugriff. Danach werden die entsprechenden Datenbestände in die zuvor festgelegte DB persistent gespeichert. Der letzte Schritt ist das Erstellen der ESA, wobei für jedes Gen insgesamt sechs ESA angelegt werden. Die ESA basieren entweder auf der Nukleotidsequenz der 5'-Upstream-Region oder der 3'-Downstream-Region und repräsentieren jeweils eine Länge von 2500 bp, 5000 bp oder 10000 bp. Diese Datenstruktur wird als Datei in das zuvor festgelegte Verzeichnis gespeichert. Der initiierte Prozess zur Datenakquisition, -bereinigung und -fusion wird automatisch beendet, wenn die gesamte Systematik erfolgreich durchgeführt wurde oder Laufzeitfehler und/oder andere Probleme aufgetreten sind. Der gegenwärtige Status als auch die entsprechenden Mitteilungen über Erfolg oder Misserfolg werden durch die GBO übersichtlich dargestellt.

5.2.3.2 TraBi - Webanwendung

Die TraBi - Webanwendung bietet einen benutzerfreundlichen Funktionsumfang zur computergestützten Identifikation von potentiellen TFBS in Nukleotidsequenzen. Die entsprechende Funktionalität der interaktiven und dynamischen TraBi - Webanwendung und deren Merkmale wird im Folgenden erläutert. Als Entwurfsmuster wurde wie bei der Webanwendung in Abschnitt 5.2.2 ebenfalls *Model 2* eingesetzt, das als Variante von MVC für Java-basiertes *Web Engineering* bezeichnet werden kann. Dadurch wird die strikte Trennung zwischen der Präsentationsschicht und der Anwendungslogik sichergestellt. Die *Extensible Hypertext Markup Language*

(XHTML) ist eine strengere Formulierung von HTML auf der Grundlage von XML und wurde als Grundgerüst für die spezifischen Webseiten verwendet. Im Gegensatz zu der Webanwendung in Abschnitt 5.2.2 erfolgte die Implementierung der TraBi - Webanwendung mittels JSF, deren Anfragebearbeitung in der Abbildung 5.11 dargestellt wird. Diese Abbildung und deren sechs Phasen werden in [MÏ0] ausführlich

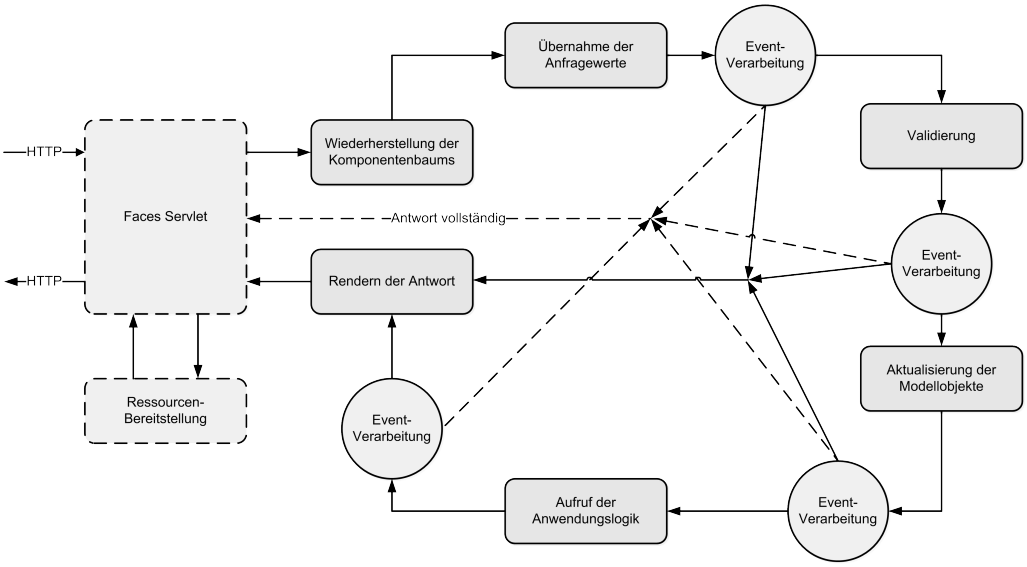


Abbildung 5.11: Bearbeitungsmodell einer JSF-Anfrage nach [MÏ0].

erläutert. Insbesondere *JavaBeans*, *Servlets* und die Komponentenbibliothek PrimeFaces in der Version 2.2.1 wurden bei der Implementierung der TraBi - Webanwendung eingesetzt. Durch diese Komponentenbibliothek kann auch das LAF einer Webanwendung problemlos geändert werden. Die TraBi - Webanwendung verwendet *south-street* als LAF, das ein zusätzliches *Add-on* der Komponentenbibliothek PrimeFaces ist.

Konfiguration der Webanwendung

Eine Konfigurationsdatei, die auf XML basiert, ermöglicht eine unkomplizierte und systemspezifische Konfiguration der TraBi - Webanwendung. Durch die nachvollziehbare Struktur der Konfigurationsdatei ist eine nachträgliche Modifikation der Parameter problemlos möglich. Der Quelltext B.5 zeigt ein exemplarisches Beispiel für diese Konfigurationsdatei.

Struktur und Design der Webseiten

Die generelle Struktur und das einheitliche Design der einzelnen Webseiten wird exemplarisch durch Abbildung 5.12 veranschaulicht. Diese Abbildung zeigt die Startseite der TraBi - Webanwendung. Anhand der Abbildung 5.12 wird deutlich, dass

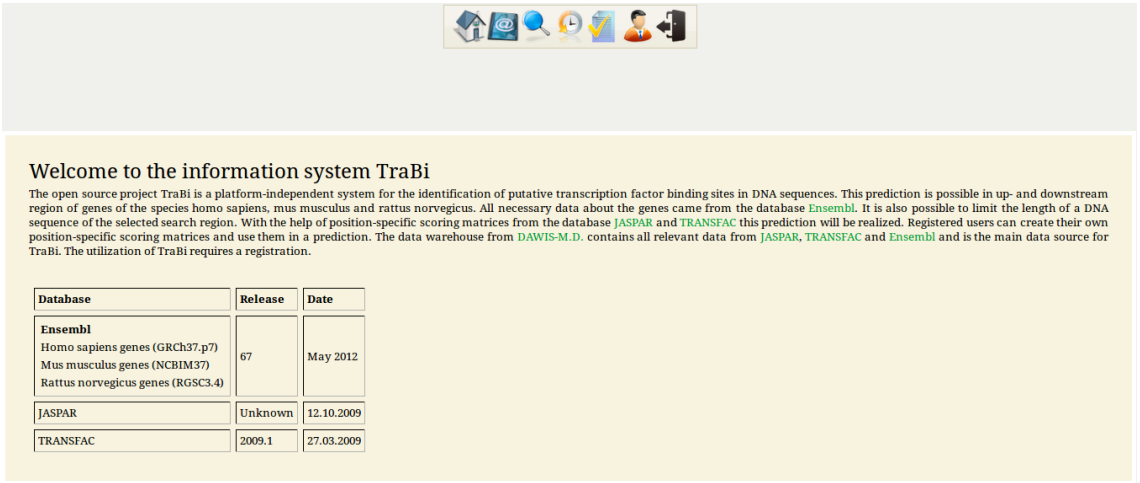


Abbildung 5.12: Startseite bei der TraBi - Webanwendung.

die globale Navigation durch eine eigenständige Navigationsleiste erfolgt, die in der Kopfzeile lokalisiert ist. Auf diese Weise ist eine schnelle und einfache Navigation zwischen den jeweiligen Bereichen und deren spezifischen Webseiten möglich. Die unterschiedlichen Symbole in der Navigationsleiste charakterisieren einen bestimmten Bereich und sind mit einem *Hyperlink* assoziiert, wodurch die globale Navigation ermöglicht wird.

System- und Benutzerverwaltung

Die gesamte Funktionalität und der Datenzugriff wird durch eine System- und Benutzerverwaltung reglementiert, weshalb eine Registrierung und eine Benutzeranmeldung erforderlich ist. Mit Hilfe einer speziellen Webseite, die Abbildung C.5 zeigt, kann der Anwender entweder die Registrierung oder die Benutzeranmeldung selbständig und ohne großen Aufwand durchführen. Als Eingabe sind bei der Registrierung der Vor- und Nachname sowie eine gültige E-Mail-Adresse zwingend erforderlich. Im Gegensatz dazu erfolgt die Benutzeranmeldung durch die Eingabe der Benutzerdaten (Benutzername und Passwort), die der Anwender nach der erfolgreichen Registrierung automatisch per E-Mail erhält. Sofern der Anwender seinen Benutzernamen und/oder sein Passwort vergessen hat, können die Benutzerdaten ebenfalls durch die Webseite in der Abbildung C.5 angefordert werden.

Konfigurationsassistent zur computergestützten Vorhersage von potentiellen TFBS in Nukleotidsequenzen

Eine Vorhersage ist eine computergestützte Identifizierung von potentiellen TFBS in Nukleotidsequenzen und wird durch einen intuitiven und dynamischen Konfigurationsassistenten angelegt. Der Konfigurationsassistent umfasst insgesamt fünf Schrit-

te und gewährleistet eine bidirektionale Navigation. Darüber hinaus wird für jede Einstellung und deren Parameter eine entsprechende Hilfestellung und ein *Tooltip* bereitgestellt. Dadurch wird die Konfiguration einer Vorhersage erheblich vereinfacht und kann problemlos durchgeführt werden. Durch diese Vorgehensweise wird besonders die Benutzerfreundlichkeit und die Software-Ergonomie (SE) der TraBi - Webanwendung verbessert. Die Abbildung 5.13 zeigt den Konfigurationsassistenten der TraBi - Webanwendung als Aktivitätsdiagramm. Das Anwendungsfalldiagramm

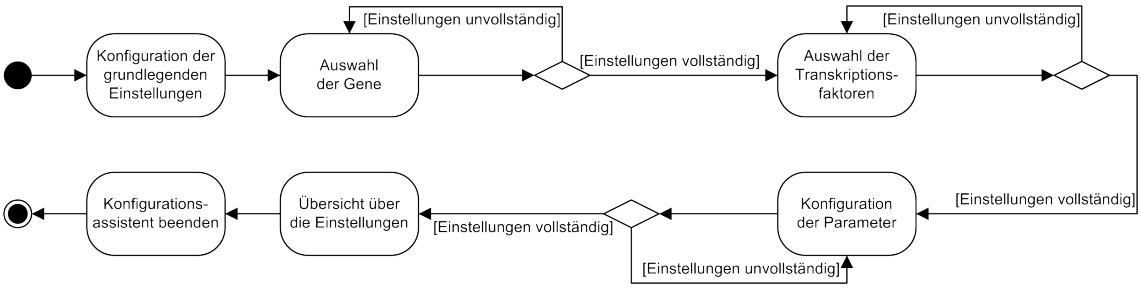


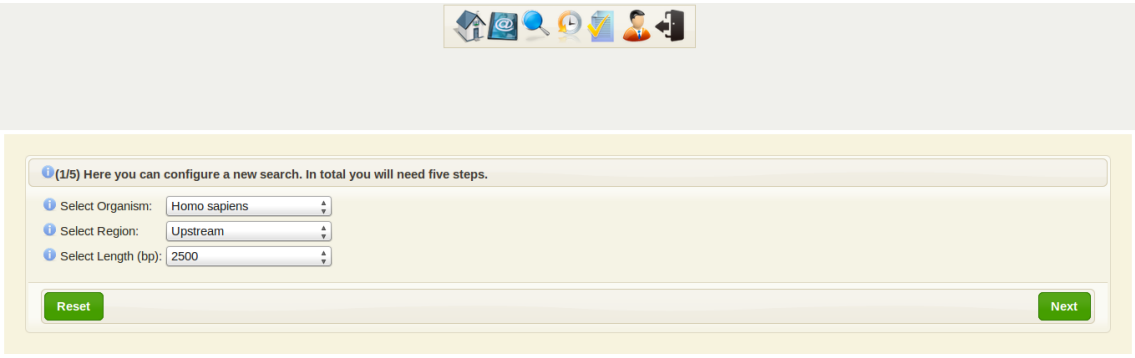
Abbildung 5.13: Konfigurationsassistent bei der TraBi - Webanwendung als Aktivitätsdiagramm.

in der Abbildung 4.4(b) beinhaltet einen Anwendungsfall `Vorhersage anlegen`, deren einzelne Verarbeitungsschritte das Aktivitätsdiagramm in der Abbildung 5.13 darstellt. Dieses Diagramm besteht aus sechs Aktivitäten, einen Start- und Endpunkt sowie drei rautenförmige Entscheidungsaktivitäten. Als nächstes werden die generelle Systematik des Konfigurationsassistenten und die Eigenschaften der einzelnen Schritte erläutert. Dabei fungieren verschiedene Abbildungen und das Anwendungsfalldiagramm in der Abbildung 4.4(b) als Hilfsmittel.

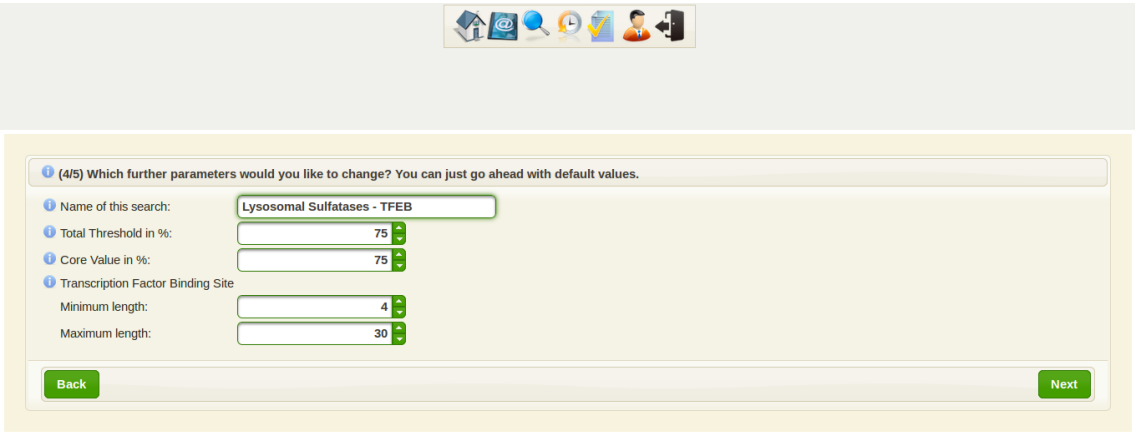
Die grundlegenden Einstellungen wie Organismus (*Homo sapiens*, *Mus musculus* oder *Rattus norvegicus*), Region (5'-Upstream-Region oder 3'-Downstream-Region) und deren Länge der Nukleotidsequenzen (2500 bp, 5000 bp oder 10000 bp) müssen als erstes ausgewählt werden. Die entsprechende Oberfläche des Konfigurationsassistenten wird in der Abbildung 5.14(a) dargestellt.

Der zweite Schritt ist für die Zusammenstellung der relevanten Gene des zuvor ausgewählten Organismus zuständig. Die TraBi - Webanwendung verfügt derzeit über genetische Informationen von drei eukaryotischen Organismen. Die exakte Anzahl der Gene eines Organismus ist durch Tabelle 4.3 nachzuvollziehen. Die verfügbaren Gene des entsprechenden Organismus werden als strukturierte Tabelle dargestellt. Diese interaktive Tabelle verfügt über dynamische Sortier-, Filter- und Suchfunktionen, wodurch die Zusammenstellung der Gene stark vereinfacht wird. Die Suchfunktionen sind jeweils mit einer Autovervollständigung assoziiert, sodass für Benutzereingaben potenzielle Vervollständigungen angezeigt werden. Die Abbildung C.6 zeigt die Oberfläche des Konfigurationsassistenten, welche die Zusammenstellung der Gene ermöglicht.

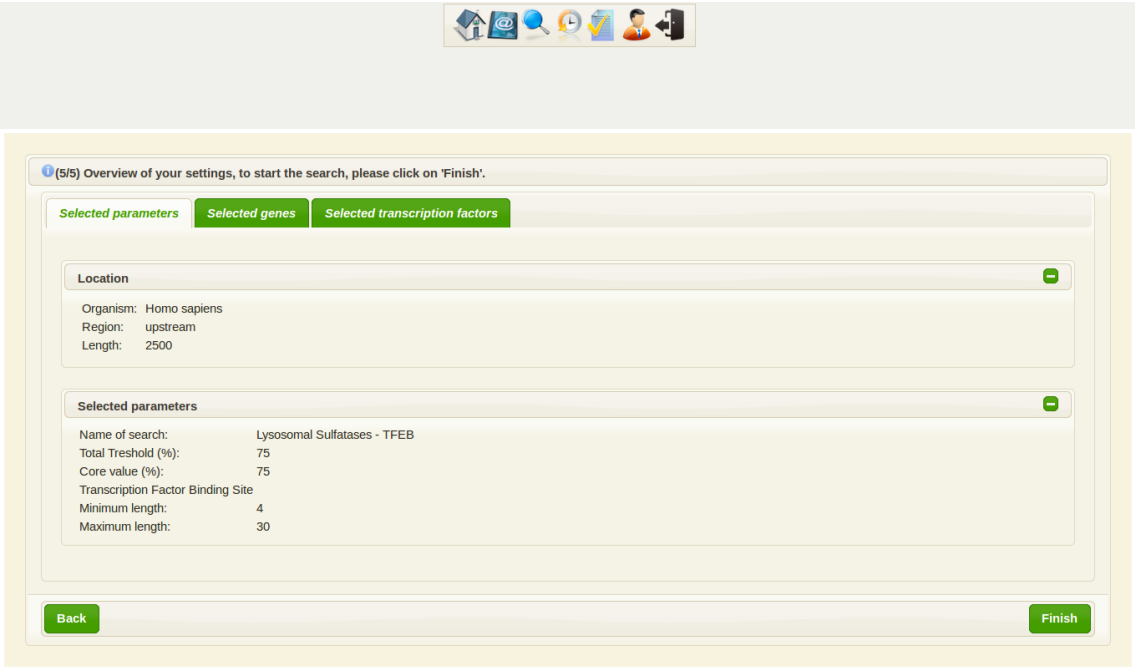
Anschließend müssen die relevanten TF und deren TFBS ausgewählt werden, wobei



(a) Oberfläche zur Konfiguration des Organismus, der Region und deren Länge der Nukleotidsequenzen.



(b) Oberfläche zur Konfiguration der Algorithmik und deren Parameter.



(c) Übersicht der zuvor durchgeführten Einstellungen und der ausgewählten Gene und TF.

Abbildung 5.14: Konfigurationsassistent bei der TraBi - Webanwendung.

die TFBS durch eine PSSM repräsentiert werden. Des Weiteren sind die TF mit eukaryotischen und/oder prokaryotischen Organismen assoziiert. Die TF werden durch eine strukturierte Tabelle dargestellt, die über dynamische Sortier-, Filter- und Suchfunktionen verfügt. Die Suchfunktionen sind jeweils mit einer Autovervollständigung assoziiert, sodass für Benutzereingaben potenzielle Vervollständigungen angezeigt werden. Darüber hinaus werden die PSSM als Tabellenform und Sequenzlogo veranschaulicht, wodurch die Verteilung der Nukleotide und die Struktur deutlich wird. Aufgrund der molekularbiologischen Signifikanz wird die *core region* der PSSM beim Sequenzlogo extra hervorgehoben. Insbesondere durch die textuellen und grafischen Komponenten und deren Funktionalitäten kann das Zusammenstellen der relevanten TF ohne großen Aufwand erfolgen. Die Oberfläche des Konfigurationsassistenten, durch die der Benutzer die relevanten TF und deren TFBS auswählen kann, ist in Abbildung C.7 dargestellt.

Der vierte Schritt ist für die Parametrisierung und die Sensitivität der Algorithmik verantwortlich. Durch entsprechende Parameter kann der MSS und der CSS sowie die minimale und maximale Länge einer TFBS eingestellt werden. Diese Parameter können die Effizienz des erweiterten Algorithmus *ESAsearch* und die Ergebnismenge einer Vorhersage positiv oder negativ beeinflussen. Zudem ist die Eingabe einer eindeutigen Bezeichnung für die Vorhersage notwendig, die maximal 32 Zeichen umfassen darf. Die Oberfläche des Konfigurationsassistenten, welche für die Konfiguration der Algorithmik und deren Parameter zuständig ist, wird in der Abbildung 5.14(b) dargestellt.

Abschließend wird eine zusammenfassende Übersicht der wichtigsten Einstellungen und der ausgewählten Gene und TF dargestellt (siehe Abbildung 5.14(c)). Dadurch kann der Anwender seine Einstellungen und Zusammenstellungen überprüfen und kurzfristige Änderungen durchführen. Anhand der Abbildung 5.13 wird deutlich, dass nach den Aktivitäten `Auswahl der Gene`, `Auswahl der Transkriptionsfaktoren` und `Konfiguration der Parameter` eine Entscheidungsaktivität folgt. Diese Entscheidungsaktivitäten sind mit zwei Bedingungen assoziiert, wobei die erste Bedingung erfüllt ist, wenn alle Einstellungen vollständig und korrekt sind. Sofern die zweite Bedingung erfüllt ist, wird die nachfolgende Aktivität nicht ausgeführt und die Einstellungen müssen durch den Anwender überprüft und korrigiert werden. Die fehlerhaften und/oder unvollständigen Einstellungen werden explizit gekennzeichnet, wodurch die Korrektur der Einstellungen problemlos durchgeführt werden kann. Das bedeutet, dass diese drei Schritte des Konfigurationsassistenten und deren Benutzereingaben und Zusammenstellungen automatisch validiert werden und der darauf folgende Schritt erst nach einer positiven Validierung verfügbar ist. Auf diese Weise werden potenzielle Fehler und/oder Unvollständigkeiten frühzeitig identifiziert und müssen durch den Anwender beseitigt werden. Der Konfigurationsassistent wird durch die letzte Aktivität beendet und der Kontroll- und Objektfluss erreicht den Endpunkt, weshalb die gesamte Aktivität als abgeschlossen gilt.

Prozess zur computergestützten Vorhersage von potentiellen TFBS in Nukleotidsequenzen

Aufgrund der Platz- und Zeitkomplexität werden die eigentlichen Vorhersagen nicht durch die TraBi - Webanwendung durchgeführt, sondern durch eine eigenständige und unabhängige Softwarekomponente. Diese Softwarekomponente wurde auf der Grundlage von RMI entwickelt und kann auf einem separaten Rechnerverbund ausgeführt werden. Außerdem wurde die Softwarekomponente für Mehrkernprozessoren optimiert und unterstützt *Multithreading*, wodurch die Performance verbessert wird. Die zentrale Anwendungslogik der Softwarekomponente ist der erweiterte Algorithmus *ESASearch*, der die effiziente Identifikation von potenziellen TFBS in Nukleotidsequenzen ermöglicht. Daraus ergibt sich, dass die TraBi - Webanwendung die Präsentationsschicht und die Softwarekomponente die Logikschicht repräsentiert. Durch die strikte Trennung der Schichten ist der Grad der Abhängigkeiten äußerst gering, sodass eine optimale Lastverteilung und eine unkomplizierte Skalierbarkeit gewährleistet werden kann. Sobald die Konfiguration einer Vorhersage erfolgreich beendet wurde, wird die Vorhersage initialisiert und in die Warteschlange der Softwarekomponente eingefügt. Diese Warteschlange wird sequentiell und kontinuierlich durch die Softwarekomponente abgearbeitet, wobei FCFS als Strategie eingesetzt wird. Darüber hinaus wird während der Initialisierung ein spezifisches Suchprofil für die Vorhersage angelegt. Dieses Suchprofil ist mit der entsprechenden Vorhersage assoziiert und beinhaltet deren relevanten Metadaten. Die Suchprofile werden in die Datenbanktabellen *user_search_profile*, *user_search_profile_gene* und *user_search_profile_pssm* gespeichert, die Bestandteile der DB *metadata* sind. Dadurch können bereits durchgeführte oder fehlerhafte Vorhersagen problemlos wiederholt und/oder deren Einstellungen nachträglich modifiziert werden. Des Weiteren erhält der Anwender automatische Benachrichtigungen per E-Mail, die über den jeweiligen Status einer Vorhersage (*Pending*, *Processing*, *Done* oder *Error*) informieren. Der Status *Processing* repräsentiert eine aktive Vorhersage, die zur Zeit von der Softwarekomponente abgearbeitet wird, wobei die Ergebnisse in regelmäßigen Zeitintervallen persistent in die DB *metadata* gespeichert werden. Auf diese Weise können die bereits verfügbaren Ergebnisse zeitnah durch den Anwender abgerufen und analysiert werden.

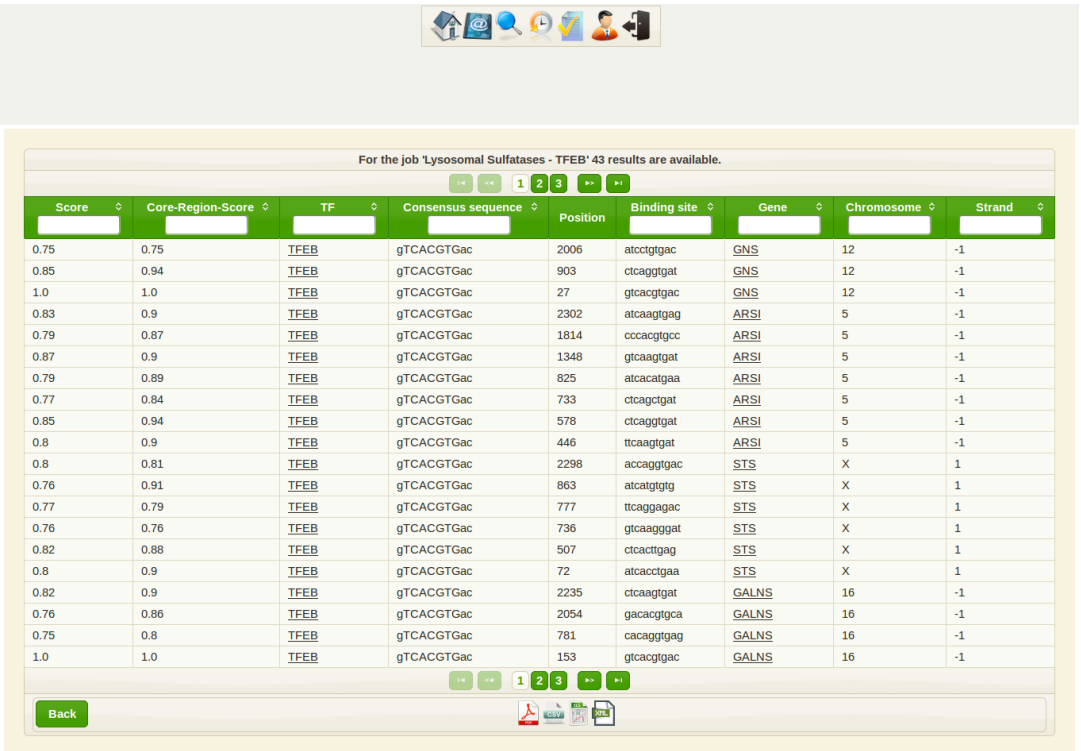
Verwaltung der Vorhersagen

Die Verwaltung der Vorhersagen erfolgt über eine spezialisierte Webseite der TraBi - Webanwendung (siehe Abbildung C.8). Die Informationen (Name, *Committed*, *Started*, *Estimated duration*, Status und *Progress*) werden als Tabellenform dargestellt (siehe Abbildung C.8). Diese interaktive Tabelle wird mittels Ajax alle 15 Sekunden automatisch aktualisiert und verfügt über dynamische Sortier-, Filter- und Suchfunktionen. Die Suchfunktionen sind jeweils mit einer Autovervollständigung assoziiert, sodass für Benutzereingaben potenzielle Vervollständigungen angezeigt

werden. Der Fortschritt der Vorhersagen wird in der Tabelle als Fortschrittsbalken dargestellt. Dadurch kann der Fortschritt einer aktiven Vorhersage problemlos nachvollzogen werden. Durch vier Schaltflächen (*Result*, *Remove*, *Stop* und *Resume*) in der Fußzeile der Tabelle können weitere Funktionen ausgeführt werden. Diese Funktionen wurden durch die Anwendungsfalldiagramme 4.3 und 4.4(b) sowie deren Anwendungsfälle beschrieben.

Struktur und Design der Ergebnisse

Die Ergebnisse, die aus einer Vorhersage resultieren, werden persistent in die Datenbanktabelle *trabi_result* gespeichert. Diese Datenbanktabelle ist ein Bestandteil der DB *metadata*, wobei die DB *metadata* und die DB *dawismd* sowie ein RDBMS das DWH repräsentieren. Das Datenbankschema der DB *metadata* wurde in der Abbildung 4.8 dargestellt. Die Abbildung 5.15 zeigt eine spezifische Webseite der TraBi - Webanwendung, die eine vollständige Ergebnismenge einer Vorhersage als strukturierte Tabelle darstellt. Diese interaktive Tabelle basiert auf *Pagination* und


For the job 'Lysosomal Sulfatases - TFEB' 43 results are available.

1 2 3

Score	Core-Region-Score	TF	Consensus sequence	Position	Binding site	Gene	Chromosome	Strand
0.75	0.75	TFEB	gTCACGTGac	2006	atcctgtgac	GNS	12	-1
0.85	0.94	TFEB	gTCACGTGac	903	ctcaggtgat	GNS	12	-1
1.0	1.0	TFEB	gTCACGTGac	27	gtcacgtgac	GNS	12	-1
0.83	0.9	TFEB	gTCACGTGac	2302	atcaagtgag	ARSI	5	-1
0.79	0.87	TFEB	gTCACGTGac	1814	cccacgtgcc	ARSI	5	-1
0.87	0.9	TFEB	gTCACGTGac	1348	gtcaagtgat	ARSI	5	-1
0.79	0.89	TFEB	gTCACGTGac	825	atcacatgaa	ARSI	5	-1
0.77	0.84	TFEB	gTCACGTGac	733	ctcagctgat	ARSI	5	-1
0.85	0.94	TFEB	gTCACGTGac	578	ctcaggtgat	ARSI	5	-1
0.8	0.9	TFEB	gTCACGTGac	446	ttcaagtgat	ARSI	5	-1
0.8	0.81	TFEB	gTCACGTGac	2298	accaggtgac	STS	X	1
0.76	0.91	TFEB	gTCACGTGac	863	atcatgtgtg	STS	X	1
0.77	0.79	TFEB	gTCACGTGac	777	ttcaggagac	STS	X	1
0.76	0.76	TFEB	gTCACGTGac	736	gtcaagggat	STS	X	1
0.82	0.88	TFEB	gTCACGTGac	507	ctcacttgag	STS	X	1
0.8	0.9	TFEB	gTCACGTGac	72	atcacctgaa	STS	X	1
0.82	0.9	TFEB	gTCACGTGac	2235	ctcaagtgat	GALNS	16	-1
0.76	0.86	TFEB	gTCACGTGac	2054	gacacgtgca	GALNS	16	-1
0.75	0.8	TFEB	gTCACGTGac	781	cacaggtgag	GALNS	16	-1
1.0	1.0	TFEB	gTCACGTGac	153	gtcacgtgac	GALNS	16	-1

1 2 3

Back

Abbildung 5.15: Webseite zur Darstellung der Ergebnisse bei der TraBi - Webanwendung.

verfügt über dynamische Sortier-, Filter- und Suchfunktionen. Durch das *Pagination* segmentiert die Tabelle die Ergebnismenge auf mehrere Bereiche, die maximal 20 Ergebnisse beinhalten können. Die Suchfunktionen sind jeweils mit einer Autover-

vollständigung assoziiert, sodass für Benutzereingaben potenzielle Vervollständigungen angezeigt werden. Insbesondere umfangreiche Ergebnismengen profitieren von der Funktionalität der Tabelle, weil die Navigation und somit die Analyse der Daten als auch die wissenschaftliche Recherche stark vereinfacht wird. Die Anzahl der verfügbaren Ergebnisse der jeweiligen Vorhersage wird in der Kopfzeile der Tabelle dargestellt. Der Export der gesamten Ergebnismenge in standardisierte Austauschformate wie CSV, Microsoft Excel, PDF und XML ist ebenfalls möglich. Durch vier Schaltflächen in der Fußzeile der Tabelle, die jeweils ein Austauschformat repräsentieren und über ein charakteristisches Symbol verfügen, können alle Ergebnisse einer Vorhersage in eine Datei exportiert werden. Auf diese Weise kann die Auswertung der Ergebnismenge auch durch spezialisierte Softwarelösungen erfolgen.

Verwaltung der Suchprofile

Die Verwaltung der Suchprofile erfolgt durch eine spezielle Webseite der TraBi - Webanwendung (siehe Abbildung C.9). Die Funktionalität der Webseite wurde abstrakt durch das Anwendungsfalldiagramm 4.4(a) und deren Anwendungsfälle beschrieben. Anhand der Abbildung C.9 wird deutlich, dass die Informationen (Name, *Threshold* und *Appointed*) als strukturierte Tabelle dargestellt werden. Diese interaktive Tabelle verfügt über dynamische Sortier-, Filter- und Suchfunktionen. Die Suchfunktionen sind jeweils mit einer Autovervollständigung assoziiert, sodass für Benutzereingaben potenzielle Vervollständigungen angezeigt werden. Durch zwei Schaltflächen (*Remove* und *Load*) in der Fußzeile der Tabelle ist entweder das Löschen oder Laden eines Suchprofils möglich.

Erstellen einer benutzerspezifischen PSSM

Durch eine spezifische Webseite erfolgt die Verwaltung der benutzerspezifischen PSSM (siehe Abbildung C.10), die auf wissenschaftlichen Publikationen und/oder auf experimentelle Studien basieren sollten. Diese Webseite zeigt auch einige Stammdaten (*First Name*, *Last Name* und E-Mail) des entsprechenden Benutzerkontos.

In der Regel verfügt ein TF über mehrere TFBS, deren Zusammensetzung hinsichtlich der Nukleotide unterschiedlich sein kann und gewissen Punktmutationen unterliegen kann. Eine Möglichkeit, diese Variationen zwischen den einzelnen TFBS eines TF übersichtlich darzustellen, sind PSSM, die häufig zur Identifikation von potenziellen TFBS eingesetzt werden. Die TraBi - Webanwendung bietet einen intuitiven Funktionsumfang zur Erstellung einer benutzerspezifischen PSSM, die einem TF zugeordnet werden sollte. Die Informationen (Name, *Description* und *Organism*) werden auf der Webseite zur Verwaltung der benutzerspezifischen PSSM als Tabellenform dargestellt (siehe Abbildung C.10). Die Tabelle verfügt in der Fußzeile über zwei Schaltflächen (*Remove* und *Create*), wodurch entweder das Löschen oder Erstellen einer benutzerspezifischen PSSM durchgeführt werden kann. Allerdings ist das

Löschen einer benutzerspezifischen PSSM nicht möglich, wenn dadurch bestehende Suchprofile, Vorhersagen und Ergebnismengen in der Funktionsweise beeinträchtigt werden. Außerdem werden die benutzerspezifischen PSSM einem Benutzerkonto zugeordnet, weshalb ausschließlich der entsprechende Anwender das Löschen durchführen kann.

Im Gegensatz dazu erfolgt das Erstellen einer benutzerspezifischen PSSM durch eine spezialisierte Webseite (siehe Abbildung 5.16). Durch ein benutzerfreundliches und

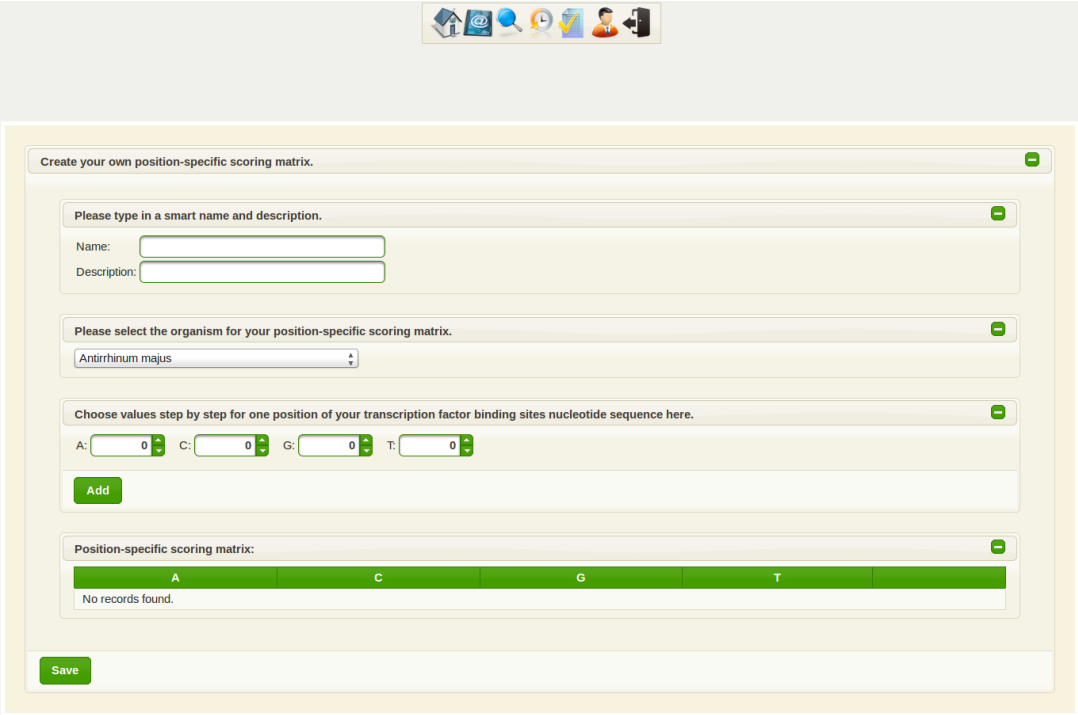


Abbildung 5.16: Webseite zur Erstellung einer benutzerspezifischen PSSM bei der TraBi - Webanwendung.

interaktives Eingabeformular auf der Webseite, das im Folgenden erläutert wird, kann eine benutzerspezifische PSSM angelegt werden.

Als erstes müssen die beiden Eingabefelder (Name und *Description*) ausgefüllt werden, die maximal 32 bzw. 64 Zeichen beinhalten dürfen.

Danach ist es notwendig, dass der entsprechende Organimus aus der Auswahlliste ausgewählt wird, die prokaryotische und eukaryotische Organismen bereitstellt. Die Eingabe der Werte für die jeweiligen Nukleotide (Adenin, Cytosin, Guanin und Thymin) an den einzelnen Positionen der benutzerspezifischen PSSM erfolgt zeilenweise. Allerdings kann eine benutzerspezifische PSSM, deren Länge weniger als vier Positionen umfasst, nicht angelegt werden.

Abschließend werden alle Benutzereingaben validiert, sodass fehlerhafte Eingaben frühzeitig identifiziert werden und eine ausreichende Informationsqualität gewährlei-

stet werden kann. Sofern die Validierung erfolgreich ist, wird die benutzerspezifische PSSM persistent in die DB *metadata* gespeichert.

Die benutzerspezifischen PSSM werden explizit gekennzeichnet und hervorgehoben, sodass ein Unterscheidungsmerkmal zu den übrigen PSSM besteht, die ursprünglich aus den Datenquellen JASPAR und TRANSFAC® resultieren. Darüber hinaus sind die benutzerspezifischen PSSM und deren Informationen für alle registrierten Benutzer frei zugänglich. Auf diese Weise soll der freie Informationsaustausch in der Wissenschaftsgemeinschaft unterstützt werden, wodurch besonders die Grundlagenforschung profitieren würde. Allerdings ist das „Wiki-Prinzip“ in der Wissenschaft noch nicht stark verbreitet und erst teilweise etabliert. Durch [SPdR+09, PIK+11] und [BKLP06] konnten drei benutzerspezifische PSSM angelegt werden, die für alle registrierten Benutzer frei verfügbar sind und auf wissenschaftlichen Publikationen basieren. Der *Transcription factor EB* (TFEB) und zwei weitere TF (Ftz und Ftz-F1) sind jeweils mit einer der drei benutzerspezifischen PSSM assoziiert.

5.3 Zusammenfassung

Dieses Kapitel thematisierte das Design und die Implementierung der Software, wobei DAWIS-M.D. und TraBi detailliert behandelt wurden. Im Gegensatz dazu wurde BioDWH nicht ausführlich erläutert, weswegen auf die entsprechenden Publikationen der Software-Infrastruktur verwiesen wurde.

Die Realisierung der Systemarchitekturen und die dazu notwendigen Technologien sowie die zusätzliche Software als Laufzeitumgebung wurde in Abschnitt 5.1 beschrieben. Darüber hinaus wurde in der Tabelle 5.1 eine Zusammenstellung der erforderlichen Programmbibliotheken/Programmierschnittstellen dargestellt. Diese Programmbibliotheken/Programmierschnittstellen wurden bei der Implementierung der Software und deren Funktionalität benötigt. Anschließend wurde im Abschnitt 5.2 die Struktur und der Funktionsumfang der Softwarelösungen beschrieben. Dabei konnten wichtige dynamische Aspekte der jeweiligen Software abstrakt als Aktivitätsdiagramm veranschaulicht werden, sodass deren Vorgehensweise und Merkmale deutlich wurde. Der Abschnitt 5.2 zeigte auch einige Abbildungen der Prototypen, wodurch das Design und die Struktur der entsprechenden Software problemlos nachvollzogen werden konnte.

Das nächste Kapitel thematisiert einen Anwendungsfall aus der molekularbiologischen Grundlagenforschung, wobei als erstes die notwendige Theorie und der aktuelle Stand der Forschung erläutert wird. Auf diese Weise werden die Funktionsvielfalt und der Anwendungsbereich der beiden Softwarelösungen in der Praxis veranschaulicht. Insbesondere die Familie der Sulfatasen, die eine spezielle Klasse der Enzyme darstellt, sind Gegenstand des Anwendungsfalls. Ein weiteres Szenario in Kapitel 6 behandelt einen TF, der die Transkription von zahlreichen lysosomalen Genen positiv reguliert, indem eine direkte Interaktion mit einer speziellen *Enhancer Box*

(E-box) im Promotor erfolgt [SPdR+09].

6 Anwendungsfall

Der hier beschriebene Anwendungsfall behandelt eine spezifische Thematik aus der molekularbiologischen Grundlagenforschung. Auf diese Weise soll der Funktionsumfang, der Anwendungsbereich und der praktische Stellenwert der beiden Softwarelösungen verdeutlicht werden. Allerdings müssen zum besseren Verständnis des Anwendungsfalls zuerst der aktuelle Stand der Forschung und das theoretische Hintergrundwissen der molekularbiologischen Thematik erläutert werden.

Die klassischen Experimente der Lebenswissenschaften werden entweder *in vitro* oder *in vivo* durchgeführt und sind zeit- und kostenaufwendig, verbrauchen Ressourcen und liefern im schlechtesten Fall keine eindeutigen Ergebnisse. Durch Softwarelösungen aus der Bioinformatik können computergestützte Analysen und Simulationen durchgeführt werden, sogenannte *in silico* Experimente, die eine kostengünstigere Alternative darstellen können. Die aus den *in silico* Experimenten resultierenden Ergebnisse können neue Anregungen für zukünftige Laborexperimente liefern sowie Hypothesen bestätigen oder widerlegen, sodass keine aufwendigen experimentellen Studien durchgeführt werden müssen.

Eine computergestützte Identifikation von potenziellen TFBS in Nukleotidsequenzen kann mittels TraBi durchgeführt werden und ist ein typisches Beispiel für ein solches Experiment. Zudem ist eine nachträgliche Analyse der Ergebnisse durch spezialisierte Softwarelösungen (CELLmicrocosmos und VANESA) möglich, da TraBi die Daten auch in standardisierte Austauschformate exportieren kann. Als Datenbasis für diese Software (CELLmicrocosmos, TraBi und VANESA) fungiert jeweils ein Data-Mart, der auf dem zugrundeliegenden DWH von DAWIS-M.D. basiert. Dadurch können verschiedene Fragestellungen der Molekularbiologie bewerkstelligt werden, da unterschiedliche molekularbiologische Datenbestände verfügbar sind.

Die molekularbiologische Thematik des Anwendungsfalls behandelt zum einen die Familie der Sulfatasen, zum anderen einen spezifischen TF, der die Genexpression lysosomaler Hydrolasen, lysosomaler Membranproteine aber auch nicht-lysosomaler Proteine koordiniert. Die Zielsetzung des Anwendungsfalls besteht darin, *in silico* Experimente für diese Thematik durchzuführen, wobei (un-)bekannte Interaktionen und Regulationen durch TF identifiziert werden sollen. Mittels TraBi werden verschiedene Vorhersagen durchgeführt, die auf der molekularbiologischen Thematik in den Abschnitten 6.1 und 6.2 basieren. Die daraus resultierenden Ergebnisse können

für zukünftige experimentelle Studien hilfreich sein.

Als erstes werden in Abschnitt 6.1 die Sulfatasen und deren Klassifikation behandelt. Die Sulfatasen sind eine spezielle Klasse von Enzymen und werden in drei Typen unterteilt. Es werden vor allem die Sulfatase 1 (Sulf1) und die Sulfatase 2 (Sulf2) beschrieben, weil diese beiden Sulfatasen bei der Regulation der Embryogenese als auch der Homöostase im adulten Organismus involviert sind [Mil08]. Danach wird in Abschnitt 6.2 ein TF thematisiert, der als TFEB bezeichnet wird und die Transkription von zahlreichen lysosomalen Genen positiv reguliert. Diese positive Regulation lysosomaler Gene erfolgt durch eine direkte Interaktion mit einer speziellen E-box im Promotor [SPdR$^+$09]. Die *in silico* Experimente und deren Ergebnisse sowie die daraus resultierenden Laborexperimente werden in Abschnitt 6.3 diskutiert. Abschließend erfolgt in Abschnitt 6.4 eine Zusammenfassung.

6.1 Familie der Sulfatasen

Die Sulfatasen sind Enzyme, die die hydrolytische Spaltung von Sulfatestern und Sulfamaten katalysieren. Diese Enzymklasse wird durch die *Enzyme Commission number* (EC-Nummer) 3.1.6.- repräsentiert. Aufgrund der unterschiedlichen aktiven Zentren und Reaktionsmechanismen werden die Sulfatasen mechanistisch in die folgenden drei Typen unterteilt:

1. Die Typ I-Sulfatasen sind in Eukaryoten und Prokaryoten vorhanden und besitzen eine starke Sequenzhomologie. Insbesondere das katalytisch aktive Formylglycin im aktiven Zentrum charakterisiert die Typ I-Sulfatasen, das aus einer PTM eines Cystein- oder Serinrest resultiert und essentiell für die Spaltung von Sulfatestern ist.

2. Die Sulfatasen der Typ II-Sulfatasen werden der Superfamilie der Dioxygenasen zugeordnet. Diese Sulfatasen fungieren als Cofaktor für Fe(II) und α-Ketoglutarat abhängige Enzyme.

3. Das aktive Zentrum der Typ III-Sulfatasen verfügt über ein spezifisches Bindemotiv für Zn^{2+}, wodurch die hydrolytische Spaltung realisiert wird.

Das humane Genom kodiert 17 verschiedene Sulfatasen, die in der Tabelle 6.1 dargestellt sind und den Typ I-Sulfatasen zugeordnet werden.

Die humanen Sulfatasen umfassen durchschnittlich 500-600 AS und erhalten während der Passage des sekretorischen Weges N-Glykosylierungen. Bezüglich des Molekulargewichts stellen Sulf1 und Sulf2 dahingehend Ausnahmefälle dar, da sich diese beiden Sulfatasen aus etwa 870 AS zusammensetzen. Die räumliche Proteinstruktur mit allen Untereinheiten wurde erst für wenige Sulfatasen experimentell verifiziert.

Durch Kristallstrukturanalysen konnten die Strukturen für vier menschliche Typ I-Sulfatasen erfolgreich bestimmt werden, wobei es sich um die humane Galaktosamin-6-Sulfatase, Arylsulfatase A, B und C handelt [RCSKG12, LKT+98, BCA+97, HGHP+03]. Ein Beispiel einer Typ II-Sulfatase bzw. einer Typ III-Sulfatase, deren Struktur ebenfalls durch die Kristallstrukturanalyse bestimmt wurde, wird ausführlich in [MKP+04] bzw. [HAW+06] erläutert. Durch Kristallstrukturanalyse konnte gezeigt werden, dass Typ I-Sulfatasen aus einer globulären Struktur bestehen, wobei eine N- und C-terminale Domäne unterschieden wird. Ein Sonderfall ist die Domänenstruktur der Arylsulfatase C, die zusätzlich zu der N- und C-terminalen Domäne über eine Transmembrandomäne verfügt. Durch diese Transmembrandomäne ist die Arylsulfatase C in der Membran des ER verankert. Das aktive Zentrum der N-terminalen Domäne ist am stärksten konserviert. Dabei wird die Formation und Struktur des aktiven Zentrums durch ein charakterisches Sequenzmotiv festgelegt, dessen Konsensussequenz CXPSR ist und als Erkennungssequenz für den *Sulfatase-modifying factor 1* (FGE) fungiert. Dieses Sequenzmotiv ist essentiell für die Konversion des Cystein in das katalytisch aktive Formylglycin [DSvF97, DLS+99, DSB+03].

Aufgrund der Unterschiede in ihrer subzellulären Lokalisierung, ihrer pH-Optima und der biologischen Funktion können die humanen Sulfatasen in die folgenden drei Gruppen eingeteilt werden:

1. Die erste Gruppe ist im Lysosom lokalisiert und verfügt über ein saures pH-Optimum, wofür die Arylsulfatasen A, B und G entsprechende Beispiele sind [HBW04, FSD08]. Die Hauptfunktion der lysosomalen Sulfatasen besteht in der enzymatischen Degradation von Glykosaminoglykanen (GAG) und Sulfolipiden in katabolen Stoffwechselwegen.

2. Im Gegensatz dazu besitzen die humanen Sulfatasen der zweiten Gruppe ein neutrales pH-Optimum und sind im ER oder im Golgi-Apparat lokalisiert. Die humanen Sulfatasen der zweiten Gruppe wie die Arylsulfatase C [RPW+05, SDP+07] zeichnen sich besonders durch den Abbau von Steroidhormonen aus.

3. Die dritte Gruppe der humanen Sulfatasen verfügen ebenfalls über ein neutrales pH-Optimum, sind aber an der Zelloberfläche lokalisiert. Dafür sind Sulf1 und Sulf2 entsprechende Beispiele, die in Abschnitt 6.1.1 ausführlich beschrieben werden. Die Schlüsselfunktion von Sulf1 und Sulf2 ist die Regulation der Signaltransduktion mittels Heparansulfat-abhängiger Wachstumsfaktoren und Morphogene.

Anhand der Tabelle 6.1 wird deutlich, dass einige Enzyme wie die Arylsulfatase A und die Arylsulfatase B für lysosomale Speicherkrankheiten (LSK) und die Arylsulfatase C und die Arylsulfatase E mit anderen monogenetischen Erbkrankheiten assoziiert sind. Die LSK sind eine Gruppe monogenetischer Erkrankungen, die durch Fehlfunktionen im Lysosom verursacht werden. Die Krankheiten in der Tabelle 6.1 sind meist auf Genmutationen oder -defekte innerhalb der kodierenden Gene der

Enzym	Genlocus	Substrat	Lokalisierung	Krankheit
Arylsulfatase A	22q13	Cerebrosid-3-sulfat	Lysosom	Metachromatische Leukodystrophie
Arylsulfatase B	5q13	Dermatansulfat / Chondroitinsulfat (4S)	Lysosom	Mukopolysaccharidose VI (Maroteaux-Lamy)
Arylsulfatase C (Steroidsulfatase)	Xq22.3	Steroidsulfat	Endoplasmatisches Reticulum	X-chromosomal-rezessive Ichthyosis vulgaris
Arylsulfatase D	Xq22.3	-	Endoplasmatisches Reticulum	-
Arylsulfatase E	Xq22.3	-	Golgi-Apparat	Chondrodysplasia Punctata
Arylsulfatase F	Xq22.3	-	Endoplasmatisches Reticulum	-
Arylsulfatase G	17q23-24	Heparansulfat (3S)	Lysosom	Mukopolysaccharidose IIIe*
Arylsulfatase H	Xq22.3	-	-	-
Arylsulfatase I	5q32	-	*Endoplasmatisches Reticulum*‡	-
Arylsulfatase J	4q26	-	*Zelloberfläche*‡	-
Arylsulfatase K	5q15	-	*Lysosom*‡	-
Galaktosamin-6-Sulfatase	16q24	Chondroitinsulfat / Keratansulfat (6S)	Lysosom	Mukopolysaccharidose IVa (Morquio A)
Glukosamin-6-Sulfatase	12q14	Heparansulfat / Keratansulfat (6S)	Lysosom	Mukopolysaccharidose IIId (Sanfilippo D)
Iduronat-2-Sulfatase	Xq27-28	Dermatansulfat / Heparansulfat (2S)	Lysosom	Mukopolysaccharidose II (Hunter)
Sulfatase 1	8q13.2-13.3	Heparansulfat (6S)	Zelloberfläche	-
Sulfatase 2	20q13.12	Heparansulfat (6S)	Zelloberfläche	-
Sulfamidase	17q25.3	Heparansulfat (NS)	Lysosom	Mukopolysaccharidose IIIa (Sanfilippo A)

* Ein Gen-Knockout der Arylsulfatase G bei *Mus musculus* führt zur Mukopolysaccharidose IIIe [KLL+12].

‡ Die Lokalisierung der Arylsulfatase I, J und K wurde bis zum jetzigen Zeitpunkt nicht eindeutig experimentell verifiziert. Allerdings liefern einige experimentelle Studien gewisse Hinweise, sodass eine hypothetische Lokalisierung der drei Arylsulfatasen möglich ist.

Tabelle 6.1: Übersicht über die humanen Sulfatasen nach [Mil12].

Sulfatasen zurückzuführen. Infolgedessen ist häufig die Enzymaktivität der Sulfatase stark eingeschränkt oder fehlt gänzlich, sodass eine Speicherung von nichtdegradierten Makromolekülen als Speichermaterial erfolgt. Das Krankheitsbild, das daraus resultiert, ist entweder eine Erbkrankheit wie die Chondrodysplasia punctata oder eine LSK wie die Metachromatische Leukodystrophie. Durch Enzymersatztherapien (EET) oder Knochenmarktransplantationen (KMT), die als kausale Therapien etabliert sind, können einige LSK wie die Mukopolysaccharidose (MPS) erfolgreich therapiert werden. Die synthetischen Enzyme der EET können die Blut-Hirn-Schranke nicht überwinden, weshalb die KMT für einige LSK eine mögliche Alternative ist. Sofern geeignetes Knochenmark verfügbar und der Patient für eine KMT geeignet ist, kann eine rechtzeitige KMT im frühen Krankheitsstadium die Progression der LSK verhindern. Allerdings sind beide Therapieansätze mit gewissen Einschränkungen assoziiert. Deswegen versucht die Grundlagenforschung für MPS entsprechende Konzepte zur Überwindung der Blut-Hirn-Schranke zu entwickeln, wofür eine geeignete Gentherapie oder die Stop-Codon-Read-Through-Therapie entsprechende Beispiele sind. Im Gegensatz dazu kann die Metachromatische Leukodystrophie gegenwärtig nicht durch eine Kausaltherapie behandelt werden, weshalb eine symptomatische Therapie erfolgt. Die meisten der in Tabelle 6.1 genannten Krankheiten sind für einen schweren Phänotyp verantwortlich und können bereits im Kindesalter tödlich verlaufen.

6.1.1 Sulfatase 1 und Sulfatase 2

Durch eine experimentelle Studie *Sonic hedgehog* (Shh) responsiver Gene in Embryonen der Wachtel konnte QSulf1 und deren positive Regulation auf die Signaltransduktion von *Wingless Int-1* (Wnt) nachgewiesen werden [DGA$^+$01]. Ferner konnten Orthologe in weiteren Organismen wie *Homo sapiens*, *Mus musculus* und *Rattus norvegicus* identifiziert werden. Darüber hinaus wurde eine zweite Isoform für *Homo sapiens* und *Mus musculus* identifiziert, die als Sulf2 bezeichnet wird. Eine ausführliche Übersicht der Orthologen von Sulf1 und Sulf2 in *Metazoa* ist in [Mil12] dargestellt.

Im Gegensatz zu den anderen Sulfatasen sind Sulf1 und Sulf2 nicht intrazellulär lokalisiert, sondern befinden sich an der Zelloberfläche. Ein weiteres Merkmal der beiden Sulfatasen ist die Anzahl der AS, die sich aus etwa 870 AS zusammensetzen, wohingegen die übrigen humanen Sulfatasen durchschnittlich 500-600 AS umfassen. Hinsichtlich der Domänenstruktur setzen sich Sulf1 und Sulf2 aus einer katalytisch aktiven N-terminalen Domäne, einer zentralen hydrophilen Domäne (HD) und einer C-terminalen Domäne zusammen. Insbesondere die HD, die Organismus und Isoform übergreifend stark konserviert ist, stellt ein wichtiges und charakteristisches Strukturmerkmal der beiden Sulfatasen dar. Die HD ist für die enzymatische Aktivität sowie die Lokalisierung auf der Zelloberfläche erforderlich und bei der Bindung von Heparansulfat (HS) involviert [ADKG$^+$06, FMD$^+$09, TR09].

Das natürliche Substrat von Sulf1 und Sulf2 auf der Zelloberfläche ist HS, das wie Keratan-, Chondroitin- und Dermatansulfat den GAG zugeordnet wird. Die GAG sind linare Polymere, deren Struktur aus repetitiven Disaccharideinheiten besteht [GA13]. Anhand der experimentellen Studien in [LWS$^+$00, TPG01] wurde deutlich, dass HS essentiell für die Embryogenese und für die Homöostase im adulten Organismus ist. Zudem ist HS ein strukturelles Merkmal der Heparansulfat-Proteoglykane (HSPG) auf der Zelloberfläche, sodass Sulf1 und Sulf2 zelluläre Prozesse beeinflussen, woran HSPG beteiligt sind. Es werden in [GA13] fünf verschiedene Familien der HSPG beschrieben, die im Folgenden aufgelistet sind:

- Agrin
- Betaglycan
- Glypicane
- Perlecan
- Syndecane

Die Biosynthese der HSPG erfolgt im ER und Golgi-Apparat. Durch die HSPG werden zahlreiche PPI vermittelt. Die Regulation der Bindung von unterschiedlichen Wachstumsfaktoren an deren entsprechenden Rezeptoren auf der Zelloberfläche ist ein charakterisches Beispiel für eine PPI, wobei HS als Corezeptor fungieren kann. Dafür sind die positive oder negative Regulation der Signaltransduktion von unterschiedlichen Wachstumsfaktoren und Signalmolekülen wie Wnt und *Fibroblast growth factor-2* (FGF-2) populäre Beispiele [ADL$^+$03, WAF$^+$04]. Auf diese Weise werden zelluläre Prozesse wie die Zelldifferenzierung und das Zellwachstum gesteuert. Die Abbildung 6.1 gibt einen Überblick über die zellulären Prozesse, die durch HSPG beeinflusst werden. Anhand Abbildung 6.1 wird deutlich, dass HS auch bei der Zelladhäsion und -migration involviert ist. Die enzymatische Aktivität von Sulf1 und Sulf2 besteht in der 6-O-Desulfatierung von HS, weshalb diese beiden Sulfatasen auch als 6-O-Endosulfatasen bezeichnet werden. Sulf1 und Sulf2 können das komplexe Sulfatierungsmuster von HS, das während der Biosynthese im Golgi-Apparat eingeführt wurde, postsynthetisch modifizieren. Die zahlreichen durch HS vermittelten PPI werden durch dessen 6-O-Sulfatierung gesteuert, wodurch Sulf1 und/oder Sulf2 entweder einen aktivierenden oder einen inhibierenden Effekt besitzen. Die Tabelle 6.2 zeigt eine Übersicht über die Wachstumsfaktoren, die positiv oder negativ durch Sulf1 und/oder Sulf2 reguliert werden. Demnach üben die durch Sulf1 und/oder Sulf2 katalysierte Desulfatierung einen regulatorischen Effekt auf zelluläre Prozesse wie Apoptose, Proliferation und Zelldifferenzierung aus. Die systemischen Auswirkungen einer fehlerhaften Regulation von Sulf1 und/oder Sulf2 auf zelluläre Prozesse und der daraus resultierende Phänotyp wird durch Knockout-Mäuse deutlich, die einen Gen-Knockout für Sulf1 und/oder Sulf2 aufweisen. Dafür sind die experimentellen Studien in [LBP$^+$06, HBRG$^+$07] entsprechende Beispiele.

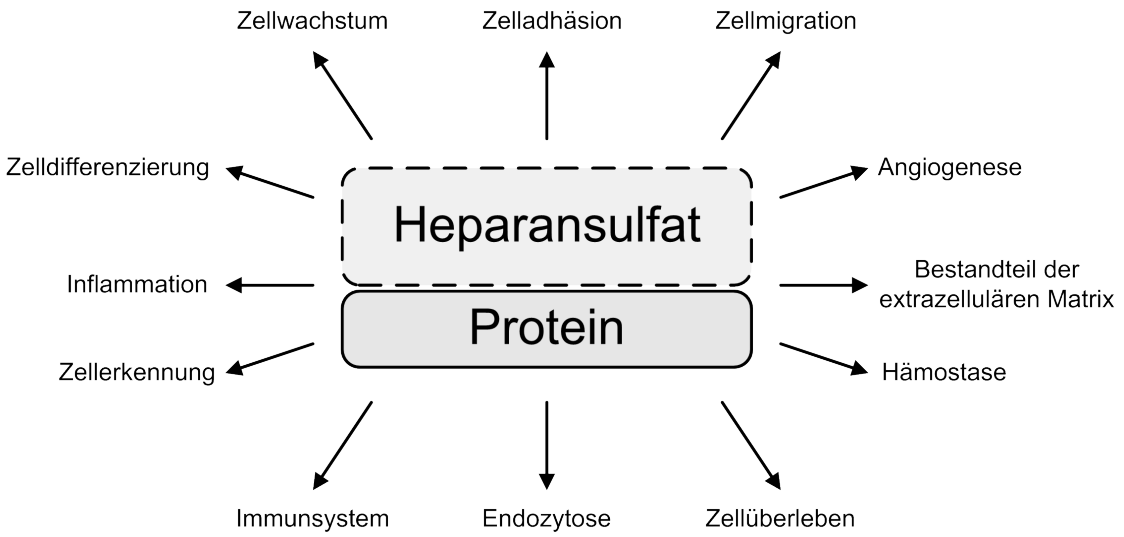


Abbildung 6.1: Überblick der beeinflussten zellulären Prozessen durch HSPG nach [DRJ+09].

Signalmolekül	**Regulation**
Bone morphogenetic protein	Positiv
Epidermal growth factor-1, Amphiregulin (Mitglied der EGF-Superfamilie)	Negativ
Fibroblast growth factor-1	Negativ
Fibroblast growth factor-2	Negativ
Glial cell line-derived neurotrophic factor	Positiv
Hepatocyte growth factor	Negativ
Platelet-derived growth factor	Negativ
Stromal cell-derived factor-1	Positiv
Sonic hedgehog	Positiv
Vascular endothelial growth factor	Negativ
Wingless Int-1	Positiv

Tabelle 6.2: Übersicht über Sulf1 und/oder Sulf2 regulierte Wachstumsfaktoren nach [Mil12].

Es wurden bis zum jetzigen Zeitpunkt mehrere TF experimentell nachgewiesen, die bei der Genexpression von Sulf1 und Sulf2 involviert sind. Diese TF können sowohl einen induzierenden als auch einen inhibierenden Effekt auf die Transkription der beiden Sulfatasen ausüben. Daraus ergibt sich, dass die Regulation der Sulf1 und/oder Sulf2 abhängigen Signaltransduktionen wie der Wnt-Signalweg[1] oder der Hedgehog-Signalweg[2] nicht statisch ist, sondern flexibel. Die Tabelle 6.3 zeigt eine

Transkriptionsfaktor	**Enzym**	**Effekt auf Transkription**
Segmentation protein fushi tarazu/ Nuclear hormone receptor FTZ-F1	Sulfated*	Induzierend
Homeobox protein Nkx-2.2	Sulfatase 1	Inhibierend
Paired box protein Pax-6	Sulfatase 1	Induzierend
Wilms tumor protein	Sulfatase 1, Sulfatase 2	Induzierend
Cellular tumor antigen p53	Sulfatase 2	Induzierend
Variant hepatic nuclear factor 1	Sulfatase 1	Inhibierend
Hypoxia-inducible factor 1-α, *Hypoxia-inducible factor 2-α*	Sulfatase 1	Inhibierend

* aus *Drosophila melanogaster* [BKLP06]

Tabelle 6.3: Übersicht über die TF, die einen direkten Effekt auf die Transkription bei Sulf1 und/oder Sulf2 aufweisen nach [Mil12].

Übersicht über die TF, die einen direkten Effekt auf die Transkription bei Sulf1 und/oder Sulf2 aufweisen. Die TF in der Tabelle 6.3 sind entweder gewebespezifisch als auch funktionsrelevant oder wichtige Faktoren bei der Embryogenese.

Die TF, die eine Homöodomäne in der Domänenstruktur aufweisen, werden durch Hox-Gene kodiert, die über ein charakteristisches Sequenzmotiv verfügen, das als Homöobox bezeichnet wird. Sofern eine Mutation in den Hox-Genen vorliegt, wird die Embryogenese schwerwiegend beeinträchtigt, weshalb die betroffenen Embryonen einer deutlich höheren Sterblichkeit unterliegen. Diese besondere Gruppe von TF können die Transkription einer ganzen Kaskade von funktionell zusammenhängenden Genen regulieren, die wiederum maßgeblich bei der Embryogenese beteiligt sind. Insbesondere TF, deren Domänenstruktur eine Homöodomäne beinhalten, können die Transkription von Sulf1 und/oder Sulf2 aktivieren oder reprimieren. Dafür sind das *Paired box protein Pax-6* (PAX6) und das *Homeobox protein Nkx-2.2* (NKX2-2) entsprechende Beispiele [GPP+09].

Im Gegensatz dazu wird der *Hypoxia-inducible factor* (HIF), der aus einer regulatorischen α-Protein-Untereinheit und einer konstitutiv exprimierten β-Protein-Untereinheit besteht, den Helix-Loop-Helix-Proteinen zugeordnet. Es existieren für

[1] siehe Abbildung D.1
[2] siehe Abbildung D.2

die α-Protein-Untereinheit drei Isoformen (HIF-1, -2, und -3), wobei der *Hypoxia-inducible factor 1-*α (HIF-1-α) ubiquitär und der *Hypoxia-inducible factor 2-*α (HIF-2-α) zellspezifisch exprimiert wird [FPK$^+$12]. Der Sauerstoffbedarf und die Sauerstoffversorgung der Zelle wird überwiegend durch HIF reguliert [SRR08]. Inwieweit der *Hypoxia-inducible factor 3-*α (HIF-3-α) daran beteiligt ist, wurde noch nicht zweifelsfrei nachgewiesen [CK10, HPKM11]. Es wurde durch [KLM$^+$11] deutlich, dass HIF-1-α und HIF-2-α die Transkription von Sulf1 inhibiert. Darüber hinaus deuten die experimentiellen Studien in [SISK92, MLD$^+$04] auf eine Korrelation zwischen HIF-1-α und den *Vascular Endothelial Growth Factor* (VEGF), der durch Sulf1 und Sulf2 negativ reguliert wird [NSC$^+$06, UMTB$^+$06].

Das *Cellular tumor antigen p53* (p53) und das *Wilms tumor protein* (WT1) sind TF bei *Vertebrata*, für die eine Schlüsselfunktion bei der Tumorgenese und Entwicklungsstörungen nachgewiesen wurde. Diese Krankheiten können auf Genmutationen (Punktmutation, Deletion, Insertion und Genduplikation) zurückgeführt werden, woraus ein fehlerhaftes Genprodukt resultiert. Die beiden TF sind Tumorsuppressoren, wobei p53 auch als „Wächter des Genoms“ bezeichnet wird [Lan92]. Anhand der Tabelle 6.3 wird deutlich, dass p53 und WT1 einen induzierenden Effekt auf die Transkription von Sulf1 und/oder Sulf2 ausüben [CDS$^+$09, HHP$^+$10].

Die TF in der Tabelle 6.3 wurden überwiegend mittels ChIP, *in-situ*-Hybridisierung, Polymerase-Kettenreaktion (PCR) oder quantitative Real-Time-PCR (qRT-PCR) identifiziert und verifiziert. Außerdem wurden für p53 und den *Variant hepatic nuclear factor 1* (vHNF1) sogenannte *in silico* Experimente durchgeführt, deren Ergebnisse durch eine der oben genannten experimentellen Methoden verifiziert wurden [CDS$^+$09, LKR$^+$09]. Obwohl der regulatorische Effekt von einigen TF auf Sulf1 und/oder Sulf2 nachgewiesen wurde, ist die Regulation der Genexpression der beiden Sulfatasen bis zum jetzigen Zeitpunkt nicht vollständig verstanden.

Die beiden Sulfatasen sind (in)direkt in physiologische Prozesse involviert, sodass Sulf1 und Sulf2 wichtige Faktoren bei der Embryogenese und der Homöostase im adulten Organismus sind. Durch experimentelle Studien wurde deutlich, dass eine fehlerhafte Regulation von Sulf1 und/oder Sulf2 bei zahlreichen Tumoren wie Brust-, Magen- und Blasenkrebs vorliegt. Dabei bestimmt der regulierte Wachstumsfaktor, ob Sulf1 oder Sulf2 als Tumorsuppressoren oder Onkogen fungieren. Eine ausführliche Übersicht über die Tumore, die eine fehlerhafte Regulation von Sulf1 und/oder Sulf2 aufweisen, ist in [Mil12] dargestellt. Das langfristige Ziel der Grundlagenforschung sind Sulf1 und Sulf2 abhängige Tumortherapien, die gegenwärtig als Modell existieren, aber noch einige Forschungsarbeit beanspruchen werden. Des Weiteren wurde in [WKB$^+$06, CLS$^+$10] nachgewiesen, dass Sulf2 an Typ-2-Diabetes beteiligt ist.

6.2 Lysosomale Gene und der *Transcription factor EB*

Durch die experimentellen Studien in [SPdR+09, PIK+11] wurde TFEB als TF identifiziert, der die Transkription von zahlreichen lysosomalen Genen positiv reguliert. Dabei wird die Transkription durch die direkte Bindung an eine spezifische E-box innerhalb des Promotors induziert, die durch ein palindromisches Sequenzmotiv charakterisiert wird. Dieses Sequenzmotiv wird durch die Konsensussequenz 5'-GTCACGTGAC-3' repräsentiert, das als CLEAR-Element (*Coordinated Lysosomal Expression and Regulation*) bezeichnet wird. Mit Hilfe der experimentellen Daten aus [SPdR+09] konnte für TFEB ein Sequenzlogo erstellt werden, das in der Abbildung 6.2 dargestellt wird. Das Sequenzlogo in der Abbildung 6.2 basiert

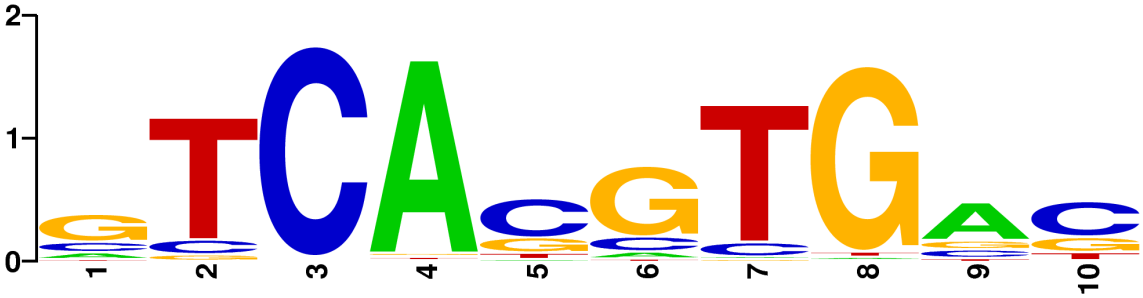


Abbildung 6.2: Darstellung der TFBS von TFEB als Sequenzlogo nach [SPdR+09].

	1	2	3	4	5	6	7	8	9	10
A	15	1	1	104	5	8	2	2	69	3
C	24	11	106	1	67	17	10	1	16	64
G	63	5	1	2	26	80	2	103	17	27
T	7	92	1	2	11	4	95	3	7	15

Tabelle 6.4: Darstellung der TFBS von TFEB als PFM nach [SPdR+09].

auf einer PFM, die Tabelle 6.4 zeigt. Anhand der Tabelle 6.4 wird die palindromische Konsensussequenz 5'-GTCACGTGAC-3' für das CLEAR-Element deutlich. Das CLEAR-Element wurde bis zum jetzigen Zeitpunkt in 68 lysosomalen Genen identifiziert, wobei das Sequenzmotiv zwischen der Position -650 bp und +288 bp relativ zum TSP lokalisiert ist. Außerdem ist TFEB bei der lysosomalen Biogenese involviert und an weiteren zellulären Prozessen wie der Autophagozytose beteiligt. Anhand der Strukturbereiche können TF in fünf verschiedene Gruppen unterteilt werden, wobei TFEB den Helix-Loop-Helix-Proteinen zugeordnet wird [CS90].

Die Tabelle 6.1 umfasst 17 humane Sulfatasen, wovon sieben eindeutig im Lysosom lokalisiert sind. Die Lysosomen sind wichtige membranumschlossene Organellen eukaryotischer Zellen und sind durch einen konstant sauren pH-Wert im Lumen

charakterisiert. Insbesondere die enzymatische Degradation von körperfremden und -eigenen Makromolekülen erfolgt durch die Lysosomen, was als primäre biologische Funktion der Lysosomen bezeichnet werden kann. Dabei sind zahlreiche verschiedene Hydrolasen beteiligt, die hydrolytische Reaktionen katalysieren und ein pH-Optimum im sauren Bereich besitzen. Die Hydrolasen sind die dritte Enzymklasse der Enzymnomenklatur und werden durch die EC-Nummer 3.-.-.- repräsentiert. Darüber hinaus sind die Lysosomen bei zellulären Prozessen wie der Autophago-, Phago-, Exo- und Endozytose involviert.

Es sind gegenwärtig etwa 60 lösliche Hydrolasen in der lysosomalen Matrix bekannt [LLS09], die in folgende Gruppen unterteilt werden können:

- Glykosidasen
- Lipasen
- Nukleasen
- Phosphatasen
- Phospholipasen
- Proteasen
- Sulfatasen

Außerdem sind zahlreiche lysosomale Transmembranproteine bekannt, wobei die Fachliteratur keine exakte Anzahl bereitstellt, weil stetig weitere solcher Proteine identifiziert werden. Sofern lysosomale Enzyme eine eingeschränkte Enzymaktivität aufweisen, können daraus LSK resultieren, die der monogenetischen Stoffwechselerkrankung zugeordnet werden. Die eingeschränkte Enzymaktivität ist häufig auf Genmutationen oder -defekte innerhalb des kodierenden Gens zurückzuführen. Dies führt dazu, dass die enzymatische Degradation von körperfremden und -eigenen Makromolekülen nicht mehr möglich oder unzureichend ist. Infolgedessen erfolgt eine stetige Zunahme von nicht degradierten Makromolekülen in den betroffenen Zellen und Organen, woraus eine LSK resultieren kann. Derzeit sind über 50 solcher LSK bekannt [LLS09, PIK$^+$11]. Ein detaillierter Überblick der LSK ist in [PBvdS12] zu finden.

Die Tabelle 6.5 zeigt sechs humane Gene, die lysosomale Sulfatasen kodieren (siehe Tabelle 6.1). Diese Gene verfügen bis auf das humane Gen der Iduronat-2-Sulfatase über bis zu drei CLEAR-Elemente innerhalb des Promotors. Die Tabelle 6.5 basiert auf den experimentellen Studien aus [SPdR$^+$09], wobei das kodierende Gen der Sulfamidase nicht berücksichtigt wurde, das ebenfalls ein lysosomales Enzym ist. Außerdem wurden die Arylsulfatasen I, J und K bei den experimentellen Studien in [SPdR$^+$09] nicht berücksichtigt, was auf die nicht experimentell verifizierte Lokalisierung der drei Arylsulfatasen zurückgeführt werden kann.

Gen		CLEAR-Element	Position*
Symbol	**Name**		
ARSA	Arylsulfatase A	GCCAAGTGAC	80
ARSB	Arylsulfatase B	GTCAGCTGAG	288
ARSG	Arylsulfatase G	GCCACGTGTG	183
GALNS	Galaktosamin-6-Sulfatase	GTCACGTGAC GTCACGCGGC GTCACGTGGC	-147 -128 -5
GNS	Glukosamin-6-Sulfatase	GTCACGTGAC CTCACGTGAT	-42 -2
IDS	Iduronat-2-Sulfatase	-	-

* Die Postion bezieht sich auf den Startpunkt der Transkription.

Tabelle 6.5: Verteilung der CLEAR-Elemente auf humane Gene innerhalb des Promotors nach [SPdR+09].

Durch TraBi können weitere Gene identifiziert werden, die mindestens ein CLEAR-Element aufweisen. Dafür sind beispielsweise die Gene für die Arylsulfatasen I, J und K als auch der Sulfamidase besonderes geeignet. Insbesondere die Ergebnisse für die Arylsulfatasen I, J und K sind für die Hypothese der Lokalisierung der drei Arylsulfatasen im Lysosom hilfreich. Des Weiteren kann TraBi die experimentellen Daten aus [SPdR+09] nachträglich überprüfen, sodass fehlerhafte oder widersprüchliche Daten deutlich werden.

6.3 Ergebnisse und Diskussion

Die *in silico* Experimente wurden mittels TraBi durchgeführt, während die erforderliche wissenschaftliche Recherche durch DAWIS-M.D. erfolgte, das eine umfangreiche molekularbiologische Datenbasis bereitstellt (siehe Kapitel 4 und 5). Dabei sind die molekularbiologischen Fragestellungen, die in den vorherigen beiden Abschnitten behandelt wurden, die grundlegende Thematik. Es wurden *in silico* Experimente sowohl für TFEB (siehe Abschnitt 6.2) durchgeführt als auch für ausgewählte humane Gene, deren Enzyme an der Biosynthese von HS beteiligt sind. Dafür sind besonders Sulf1 und Sulf2 populäre Beispiele, weil HS das natürliche Substrat der beiden Sulfatasen auf der Zelloberfläche ist (siehe Abschnitt 6.1.1).

Als erstes werden in Abschnitt 6.3.1 die besonderen Eigenschaften der *in silico* Experimente erläutert, wobei hauptsächlich die Konfiguration bei TraBi beschrieben wird. Die aus den *in silico* Experimenten resultierende Ergebnismenge[3] wurde durch den Experimentator analysiert. Dabei wurden geeignete Ergebnisse für experimentelle Studien ausgewählt, die in den folgenden Abschnitten als Tabellenform dargestellt

[3]Die beiliegende CD-ROM beinhaltet die vollständige Ergebnismenge der *in silico* Experimente.

werden. Die beiden Tabellen 6.6 und 6.7 zeigen vielversprechende Ergebnisse für TFEB (siehe Abschnitt 6.3.2), für die auch Laborexperimente erfolgt sind. Diese Experimente sind auf die *in silico* Experimente zurückzuführen. Im Gegensatz dazu werden die Ergebnisse für die humanen Gene der beiden Sulfatasen in den Tabellen 6.8 und 6.9 dargestellt (siehe Abschnitt 6.3.3). Diese Ergebnisse können für gegenwärtige experimentelle Studien hilfreich sein und neue Anregungen für zukünftige Laborexperimente liefern. Die Ergebnisse der übrigen humanen Gene, deren Enzyme wie Sulf1 und Sulf2 ebenfalls an der Biosynthese von HS beteiligt sind, werden in den Tabellen 6.11 und 6.12 dargestellt (siehe Abschnitt 6.3.3).

6.3.1 Merkmale und Konfiguration der *in silico* Experimente

Der Schwerpunkt der molekularbiologischen Thematik und deren Fragestellungen, die in Abschnitt 6.1 und 6.2 erläutert wurden, ist überwiegend der menschliche Organismus. Deswegen wurden die *in silico* Experimente nur für das humane Genom durchgeführt. Diese Experimente können aber mittels Trabi problemlos für *Mus musculus* und *Rattus norvegicus* wiederholt werden, sodass zukünftig ein *cross-species* Vergleich möglich ist.

Es wurde nicht nur die 5'-Upstream-Region der Gene berücksichtigt, sondern auch die 3'-Downstream-Region, deren Nukleotidsequenz ebenfalls potenzielle TFBS beinhalten kann. Die Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region können bei TraBi bis zu 10000 bp umfassen, die sich relativ zum Start bzw. Ende eines Gens befinden. Auf diese Weise kann TraBi auch potenzielle TFBS identifizieren, die nicht in unmittelbarer Nähe eines Gens lokalisiert sind. Die TFBS für SRF kann etliche Basenpaare entfernt sein, wobei sich die Position relativ zum Startpunkt der Translation befindet. Dafür sind ACTC, Fhl2, Keratin-17 und weitere Gene bei *Homo sapiens* und *Mus musculus* entsprechende Beispiele [PSD+04]. Allerdings wurde für die *in silico* Experimente die Länge der Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region auf maximal 2500 bp eingeschränkt, weil potenzielle TFBS in unmittelbarer Nähe der Gene vermutet wurden.

Die genetische Variabilität der TFBS bei TF kann durch die PSSM repräsentiert werden. Damit die genetische Variabilität der TFBS bei der Identifikation von potenziellen TFBS in Nukleotidsequenzen berücksichtigt wird, stützt sich TraBi auf einen Lösungsansatz der auf PSSM basiert. Dafür ist die Konsensussequenz nicht geeignet, weil die Ergebnismenge dann deutlich mehr falsch positive Ergebnisse beinhalten würde [DM92]. Die *core region* der PSSM sind die am stärksten konservierten und benachbarten Positionen, wofür die Positionen 2 - 8 bei Abbildung 6.2 und Tabelle 6.4 entsprechende Beispiele sind. Insbesondere stark konservierte Sequenzabschnitte sind für die molekularbiologische Grundlagenforschung von besonderem Forschungsinteresse. Sofern die *core region* über einen schlechten *score* verfügt, ist die Interaktion mit TF äußerst unwahrscheinlich, weshalb für den CSS ein Schwellen-

wert festgelegt wurde. Der definierte Schwellenwert für den CSS gewährleistet einen überdurchschnittlichen *score* für die *core region*, wodurch falsch positive Ergebnisse in der Ergebnissemenge beseitigt wurden.

Die potenziellen TFBS, die TraBi in Nukleotidsequenzen identifiziert, können über eine Länge zwischen 6 und 33 bp verfügen. Allerdings wird in [FSS11] die charakteristische Länge einer TFBS mit 6 - 12 bp und in seltenen Kombinationen bis zu 18 bp beschrieben. Infolgedessen wurde für die *in silico* Experimente die minimale Länge einer TFBS auf 6 bp und die maximale Länge auf 12 bp festgelegt. Dadurch wurde die molekularbiologische Signifikanz der Ergebnismenge verbessert, weil ausschließlich Ergebnisse berücksichtigt wurden, die das gerade genannte Kriterium erfüllen.

6.3.2 Ergebnisse für *Transcription factor EB*

Das CLEAR-Element ist ein Sequenzmotiv, das durch die palindromische Konsensussequenz 5'-GTCACGTGAC-3' repräsentiert wird und sich häufig innerhalb von Promotoren lysosomaler Gene befindet (siehe Abschnitt 6.2). Dieses Sequenzmotiv wurde in 68 von 96 lysosomalen Genen identifiziert und fungiert für TFEB als TFBS, wobei das CLEAR-Element zwischen der Position -650 bp und +288 bp relativ zum TSP lokalisiert ist [SPdR$^+$09]. Daraus ergibt sich, dass TFEB die Transkription von zahlreichen lysosomalen Genen positiv reguliert [SPdR$^+$09, PIK$^+$11]. Anhand der Tabelle 6.1 wird deutlich, dass sieben der 17 humanen Sulfatasen eindeutig im Lysosom lokalisiert sind, während die Lokalisation der Arylsulfatase H, I, J und K bis zum jetzigen Zeitpunkt nicht zweifelsfrei nachgewiesen wurde. Allerdings liefern einige experimentelle Studien für die Arylsulfatase I, J und K gewisse Hinweise, worauf eine Lokalisation der drei Arylsulfatase möglich ist (siehe Tabelle 6.1). Dabei gilt besonders die Lokalisation der Arylsulfatase K im Lysosom als sehr wahrscheinlich [WWK$^+$13]. Im Gegensatz dazu gilt die bisher angenommene Lokalisation der Arylsulfatase I und J als wage, da biochemische Befunde eine lysosomale Lokalisierung unterstützen.

Damit eine eindeutige Lokalisation der Arylsulfatase I, J und K möglich ist, sind zusätzliche Hinweise notwendig, die neue Anregungen für zukünftige Laborexperimente liefern und Hypothesen bestätigen oder widerlegen können. Deswegen wurden *in silico* Experimente für die humanen Gene der drei Arylsulfatasen (ARSI, ARSJ und ARSK) und der sieben lysosomalen Sulfatasen (ARSA, ARSB, ARSG, GALNS, GNS, IDS und SGSH) durchgeführt. Auf diese Weise wurden potenzielle TFBS für TFEB in der 5'-Upstream-Region und der 3'-Downstream-Region der humanen Gene identifiziert, die experimentell verifiziert werden müssen. Die nächsten beiden Tabellen zeigen Ergebnisse, die aus der Ergebnismenge der *in silico* Experimente ausgewählt wurden, wobei der *Core-Region-Score* und der allgemeine *Score* als Auswahlkriterien fungierten. Dabei werden in der Tabelle 6.6 die Ergebnisse der 5'-Upstream-Region dargestellt, wohingegen Tabelle 6.7 die Ergebnisse der 3'-Downstream-Region zeigt.

Score	*Core-Region-Score*	Gen	Position*	CLEAR-Element
1.00	1.00	GNS	27	gtcacgtgac‡
1.00	1.00	GALNS	153	gtcacgtgac‡
0.94	1.00		11	gtcacgtggc‡
0.93	1.00	ARSK	67	ttcacgtgac
0.89	0.94	ARSA	1347	atcaggtgac
0.85	0.94	ARSI	578	ctcaggtgat
0.80	0.90		446	ttcaagtgat
0.74	0.91	ARSB	507	ttcatgtgtt
0.78	0.91	ARSJ	2456	ctcatgtggg
0.79	0.90	IDS	11	ttcacctgtc
0.77	0.90	SGSH	876	ctcacctggg
0.74	0.90	ARSG	1740	ctcaagtgca

* Die Position der potenziellen CLEAR-Elemente befindet sich relativ zum Start des Gens.

‡ Das CLEAR-Element wurde in [SPdR+09] experimentell verifiziert (siehe Tabelle 6.5).

Tabelle 6.6: CLEAR-Elemente für TFEB, die stromaufwärts bei humanen lysosomalen Genen lokalisiert sind.

Der Promotor der humanen Gene der Glukosamin-6-Sulfatase und Galaktosamin-6-Sulfatase (GNS und GALNS) beinhaltet zwei bzw. drei CLEAR-Elemente. Im Gegensatz dazu verfügen die humanen Gene der Arylsulfatase A, B und G (ARSA, ARSB und ARSG) über ein CLEAR-Element innerhalb des Promotors. Die CLEAR-Elemente der fünf Gene (GNS, GALNS, ARSA, ARSB und ARSG) wurden in [SPdR+09] experimentell nachgewiesen, während für das humane Gen der Iduronat-2-Sulfatase (IDS) keine CLEAR-Elemente innerhalb des Promotors identifiziert wurden. Die Tabelle 6.5 gibt einen Überblick über die Verteilung der CLEAR-Elemente auf die humanen Gene, die lysosomale Sulfatasen kodieren und in [SPdR+09] berücksichtigt wurden.

Durch Tabelle 6.6 wird deutlich, dass TraBi in der 5'-Upstream-Region der humanen Gene der Glukosamin-6-Sulfatase und Galaktosamin-6-Sulfatase einige CLEAR-Elemente identifizieren konnte, die in [SPdR+09] durch experimentelle Studien nachgewiesen wurden (siehe Tabelle 6.5). Außerdem wurde in der 5'-Upstream-Region des humanen Gens der Arylsulfatase K (ARSK) eine potenzielle TFBS für TFEB identifiziert, die nach [SPdR+09] auch als potenzielles CLEAR-Element bezeichnet werden kann (siehe Tabelle 6.6). Sofern die Transkription des humanen Gens der Arylsulfatase K durch TFEB induziert wird, ist das ein Hinweis für die lysosomale Lokalisierung der Arylsulfatase K, weil TFEB zahlreiche lysosomale Gene positiv reguliert. Diese Gene können mehrere CLEAR-Elemente innerhalb des Promotors beinhalten. Der Promotor des humanen Gens der Iduronat-2-Sulfatase beinhaltet nach [SPdR+09] keine CLEAR-Elemente (siehe Tabelle 6.5). Allerdings konnte Tra-

Score	*Core-Region-Score*	Gen	Position*	CLEAR-Element
0.87	1.00	ARSI	2464	ttcacgtgcc
0.79	0.90		827	gtcacctgtt
0.77	0.94	ARSB	1573	atcaggtgct
0.85	0.94	GNS	1234	ctcaggtgat
0.84	0.94	IDS	2306	atcaggtgat
0.77	0.94	ARSJ	762	ctcaggtgtt
			1956	ttcaggtggt
0.85	0.94	SGSH	163	ctcaggtgat
0.83	0.94	GALNS	74	atcaggtgcc
0.83	0.91		396	ctcatgtggc
0.80	0.90	ARSK	131	ttcaagtgat
0.74	0.90	ARSA	547	ttcacctgtg

* Die Position der potenziellen CLEAR-Elemente befindet sich relativ zum Ende des Gens.

Tabelle 6.7: CLEAR-Elemente für TFEB, die stromabwärts bei humanen lysosomalen Genen lokalisiert sind.

Bi in der 5'-Upstream-Region des humanen Gens der Iduronat-2-Sulfatase eine potenzielle TFBS für TFEB identifizieren, die sich in unmittelbarer Nähe zum Start des Gens befindet (siehe Tabelle 6.6). Aufgrund der nicht übereinstimmenden Ergebnisse ist eine aussagekräftige Schlussfolgerung nicht möglich. Daher sollten zusätzliche Laborexperimente und/oder *in silico* Experimente durchgeführt werden. Darüber hinaus wurde in der 5'-Upstream-Region der humanen Gene der Sulfamidase (SGSH) und der Arylsulfatase A, B, G, I und J (ARSA, ARSB, ARSG, ARSI und ARSJ) potenzielle TFBS für TFEB identifiziert, die aber, obwohl der *Core-Region-Score* $\geq$90% ist, keine CLEAR-Elemente repräsentieren, weil deren Position nicht innerhalb des Promotors lokalisiert sind (siehe Tabelle 6.6).

Die experimentellen Studien, die in zwei Forschungspraktika [Ort12, Gar13] durchgeführt wurden, sind auf die *in silico* Experimente zurückzuführen. Insbesondere die Ergebnisse der 5'-Upstream-Region, die in der Tabelle 6.6 dargestellt sind, fungierten als Ausgangspunkt für die Laborexperimente. Im Gegensatz dazu können die Ergebnisse der 3'-Downstream-Region, die Tabelle 6.7 zeigt, für zukünftige experimentelle Studien hilfreich sein. Die Arylsulfatase G, Cathepsin F (CATF), Hexosaminidase A (HEXA) und Mucolipin-1 (MCOLN1) sind humane lysosomale Enzyme, deren Gene innerhalb des Promotors bis zu drei CLEAR-Elemente beinhalten und durch TFEB positiv reguliert werden [SPdR$^+$09]. Die zelluläre Lokalisierung der humanen Arylsulfatase I, J und K wurde bis zum jetzigen Zeitpunkt nicht eindeutig nachgewiesen (siehe Tabelle 6.1), obwohl einige Hinweise die Hypothese der lysosomalen Lokalisierung unterstützen. Diese sieben Enzyme wurden unter Zuhilfenahme der *in silico* Experimente in beiden Forschungspraktika untersucht, wobei auch die experimentellen Studien in [SPdR$^+$09] überprüft wurden. Dabei wurde die Auswirkung der

Überexpression von TFEB auf die Expressionsstärke der vier lysosomalen Enzyme und der Arylsulfatase I, J und K in zwei eukaryotischen Zelllinien (HeLa-Zellen[4] und HT1080) untersucht [Gar13]. Durch die transiente und stabile Transfektion mittels Lipofektion wurde TFEB in beide Zelllinien eingebracht (siehe Tabelle E.1). Danach wurde die RNA sowohl aus den transfizierten Zellen als auch aus dessen Wildtyp (WT) extrahiert und in cDNA (*complementary* DNA) umgeschrieben. Abschließend wurde die Expressionsstärke mittels relative Quantifizierung der qRT-PCR nachgewiesen. Insbesondere für ARSK konnte in HT1080 und HT1080-2 eine erhöhte Expression nachgewiesen werden, wohingegen die übrigen Zellenlinien keine signifikante Veränderung zum WT aufweisen (siehe Abbildung E.6(c)). Anhand der Tabelle E.4 wird deutlich, dass sich der C_T-Wert (*Cycle Threshold*) für ARSK zwischen 22 und 27 Zyklen befindet. Die Gelelektrophorese ist eine Methode der Elektrophorese, wobei ein Gel als Trägermedium fungiert. Auf diese Weise werden die Moleküle (DNA, RNA und Proteine) getrennt und als Bandenmuster im Gel sichtbar. Die entsprechenden Banden für ARSK sind im Gel bei etwa 100 bp sichtbar (siehe Abbildungen E.2, E.3 und E.4). Die Hypothese, dass TFEB die Transkription des humanen Gens der Arylsulfatase K positiv reguliert, konnte durch die Überexpression von TFEB in HT1080 bestätigt werden, weil die Expressionsstärke der ARSK auf das 1,5 bis 2-fache erhöht wurde (siehe Abbildung E.6(c)). Diese Hinweise unterstützen auch die Hypothese, dass die Arylsulfatase K im Lysosom lokalisiert ist bzw. an katabolen Prozesse in der Zelle beteiligt ist.

Außerdem wurde durch die Induktion von Saccharose in HeLa-Zellen eine natürliche lysosomale Stressreaktion simuliert, wodurch Saccharose im Lysosom angereichert wird [KIB+97]. Infolgedessen wird das Lysosom vergrößert, die Biogenese der Lysosomen erhöht und die Aktivität der Genexpression der lysosomalen Hydrolasen verstärkt. Dabei wurde die Auswirkung der Saccharose-Induktion auf die Expressionsstärke und das Expressionsmuster der humanen Gene der sieben Enzyme untersucht.

Die Auswirkung der Überexpression von TFEB in HeLa-Zellen und HT1080 auf die Expressionsstärke der oben genannten Enzyme und die Ergebnisse der Saccharose-Induktion werden ausführlich in [Gar13] beschrieben. Darüber hinaus sind wichtige Ergebnisse der Laborexperimente im Anhang E dargestellt.

6.3.3 Ergebnisse für Sulfatase 1 und Sulfatase 2

Durch experimentelle Studien wurden sowohl für Sulf1 als auch Sulf2 mehrere TF nachgewiesen, die einen induzierenden oder inhibierenden Effekt auf die Transkription der beiden Sulfatasen ausüben können (siehe Tabelle 6.3). Dabei wurde zwar der regulatorische Effekt der TF auf Sulf1 und/oder Sulf2 nachgewiesen, aber die Genregulation der Genexpression der beiden Sulfatasen ist bis zum jetzigen Zeit-

[4] **He**nrietta **La**cks

Score	*Core-Region-Score*	TF	Gen	Position*	TFBS
0.97	1.00	STAT5A	SULF1	44	cacttctc
0.93	1.00			20	cctttctc
0.97	1.00	TBP, TFIID		31	tttatag
0.91	1.00	CSX		47	tcccacttct
0.90	0.92	FOXC1		60	tctctgta
≥0.90	1.00	STAT5A, STAT4	SULF2	18 25	cctttcta ggtttctc
0.96	1.00	ZEB (1124 AA)		26	cggtttctc
0.97	1.00	SPI1		33	aggaag
0.92	1.00	PU.1		35	ggaggaag
0.92	1.00	ETS1		49	attcct

* Die Position der potenziellen TFBS befindet sich relativ zum Start des Gens.

Tabelle 6.8: TF und deren DNA-Bindestellen, die stromaufwärts bei den humanen Genen für Sulf1 und/oder Sulf2 lokalisiert sind.

punkt nicht vollständig verstanden. Damit komplexe und zusammenhängende zelluläre Prozesse, woran Sulf1 und/oder Sulf2 beteiligt sind, besser nachvollzogen werden können, sind neue Erkenntnisse über genetische Interaktionen und Regulationen zwingend erforderlich. Die Forschungsergebnisse können besonders für das bessere Verständnis der Biosynthese von HS hilfreich sein, weil HS auf der Zelloberfläche das natürliche Substrat von Sulf1 und Sulf2 ist. Durch die neuen Erkenntnisse in der Grundlagenforschung sollen gegenwärtige Krebs-, Gen- oder Stammzelltherapien verbessert werden und es wird der Ausgangspunkt für zukunftsweisende Therapien in der Medizin geschaffen.

Die TF in der Tabelle 6.3, sind die einzigen experimentell nachgewiesen TF, die einen regulatorischen Effekt auf Sulf1 und/oder Sulf2 ausüben können. Darüber hinaus sind keine konkreten Hinweise oder Hypothesen für DNA-Protein-Interaktionen verfügbar, weshalb es sinnvoll ist, als erstes *in silico* Experimente durchzuführen. Es können auch experimentelle Analyse wie ChIP, *DNase Footprinting Assay* oder EMSA durchgeführt werden, aber solche Experimente sind häufig sehr zeit- und kostenaufwendig, verbrauchen Ressourcen und liefern im schlechtesten Fall keine eindeutigen Ergebnisse. Deswegen wurden *in silico* Experimente für die humanen Gene (SULF1 und SULF2) der beiden Sulfatasen durchgeführt, wobei ausschließlich TF berücksichtigt wurden, die für das humane Genom relevant sind. Auf diese Weise wurden potenzielle TFBS in der 5'-Upstream-Region und der 3'-Downstream-Region der beiden Gene identifiziert, die durch experimentelle Studien verifiziert werden müssen. Die Tabelle 6.8 und die Tabelle 6.9 zeigen ausgewählte Ergebnisse der *in silico* Experimente, die neue Anregungen für zukünftige Laborexperimente liefern können. Allerdings sind für Genexpressionsanalysen neben der experimentellen Expertise auch unterschiedliche Methoden und Materialien aus der Molekular-, Zellbiologie und Biochemie zwingend erforderlich. Die Ergebnisse der *in silico* Expe-

Score	*Core-Region-Score*	TF	Gen	Position*	TFBS
1.00	1.00	ZEB (1124 AA)	SULF1	0	ctgtttcag
0.96	0.95	TEF-1		10	gaaatg
0.90	1.00	RUNX1		29	ttttgtggtac
1.00	1.00	AML1a, AML1		32	tgtggt
1.00	1.00	STAT5A		43	catttctt
0.94	1.00	Nkx6-2	SULF2	1	atattaaatact
≥0.92	1.00	GATA-1, GATA-6, GATA-1 isoform 1		10 12	actgataact tgataac
0.93	1,00	GATA-2, GATA-4, GATA-3 isoform-1, GATA-5		12	tgataac
0.97	1.00	FOXC1		17	actaagta
0.90	1.00	c-Myb		57	agatgcagttt

* Die Position der potenziellen TFBS befindet sich relativ zum Ende des Gens.

Tabelle 6.9: TF und deren DNA-Bindestellen, die stromabwärts bei den humanen Genen für Sulf1 und/oder Sulf2 lokalisiert sind.

rimente können durch eine transiente und stabile Transfektion mittels Lipofektion [FGH+87] überprüft werden. Dafür sind aber geeignete eukaryotische Zelllinien wie *Chinese Hamster Ovary* (CHO), HEK-293 (*Human Embryonic Kidney*), HeLa oder HT1080 und effiziente Primer notwendig, wobei die Spezifität und Effizienz der Primer-Bindung mittels qRT-PCR überprüft werden kann. Mit Hilfe der qRT-PCR können auch die Expressionsstärke und das Expressionsmuster der Gene nachgewiesen werden. Damit die Kultivierung der Zellen erfolgreich verläuft, wird für die eukaryotischen Zelllinien ein passendes Nährmedium benötigt.

Aufgrund der zahlreichen Ergebnisse wurden jeweils für Sulf1 und Sulf2 die fünf besten Ergebnisse ausgewählt. Dabei fungierten als Auswahlkriterien der *Core-Region-Score* und der allgemeine *Score*, während der Schwellenwert für den CSS und den MSS jeweils bei 90% festgelegt wurde. Außerdem wurde die Position der potenziellen TFBS berücksichtigt, die sich in unmittelbarer Nähe zum Start oder Ende des Gens befindet. Die Tabelle 6.11 zeigt TF und deren TFBS, die in der 5'-Upstream-Region der humanen Gene der beiden Sulfatasen identifiziert wurden. Die TF und deren TFBS, die in der 3'-Downstream-Region der beiden Genen identifiziert wurden, werden in der Tabelle 6.9 dargestellt.

Die Tabelle 6.10 gibt einen Überblick über weitere humane Enzyme, die wie Sulf1 und Sulf2 an der Biosynthese von HS beteiligt sind. Die Biosynthese der drei GAG (Heparan-, Chondroitin- und Dermatansulfat) erfolgt im Golgi-Apparat und kann in die folgenden drei Schritte unterteilt werden [Mil12]:

1. Generierung eines Tetrasaccharid-Linkers

Enzym	EC-Nummer	Gen
Beta-1,4-galactosyltransferase 7	2.4.1.-	B4GALT7
Exostosin-like 3	2.4.1.223	EXTL3
Bifunctional heparan sulfate N-deacetylase/ N-sulfotransferase 1	2.8.2.8	NDST1
Bifunctional heparan sulfate N-deacetylase/ N-sulfotransferase 2		NDST2
Bifunctional heparan sulfate N-deacetylase/ N-sulfotransferase 3		NDST3
Bifunctional heparan sulfate N-deacetylase/ N-sulfotransferase 4		NDST4
D-glucuronyl C5-epimerase	5.1.3.17	GLCE
Heparan sulfate 2-O-sulfotransferase	2.8.2.-	HS2ST1
Heparan sulfate glucosamine 3-O-sulfotransferase 1	2.8.2.23	HS3ST1
Heparan sulfate glucosamine 3-O-sulfotransferase 4		HS3ST4
Heparan sulfate glucosamine 3-O-sulfotransferase 5		HS3ST5
Heparan sulfate glucosamine 3-O-sulfotransferase 6		HS3ST6
Heparan sulfate glucosamine 3-O-sulfotransferase 2	2.8.2.29	HS3ST3
Heparan sulfate glucosamine 3-O-sulfotransferase 3A1	2.8.2.30	HS3ST3A1
Heparan sulfate glucosamine 3-O-sulfotransferase 3B1		HS3ST3B1
Heparan-sulfate 6-O-sulfotransferase 1	2.8.2.-	HS6ST1
Heparan-sulfate 6-O-sulfotransferase 2		HS6ST2
Heparan-sulfate 6-O-sulfotransferase 3		HS6ST3

Tabelle 6.10: Enzyme, die wie Sulf1 und Sulf2 an der Biosynthese von HS beteiligt sind.

2. Ketten-Elongation

3. Ketten-Modifizierung

Die Generierung eines Tetrasaccharid-Linkers erfolgt durch die *Beta-1,4-galactosyltransferase* und *Exostosin-like 3*, die den Glycosyltransferasen zugeordnet werden. Im Gegensatz dazu erfolgt die Ketten-Modifizierung durch Sulfotransferasen wie die *Heparan sulfate 2-O-sulfotransferase* und die *D-glucuronyl C5-epimerase*, die wiederum den Isomerasen zugeteilt wird. Die Tabelle 6.10 zeigt weitere Sulfotransferasen, die als modifizierende Enzyme bei der Biosynthese von HS fungieren. Es wurden für die humanen Gene der in Tabelle 6.10 aufgeführten Enzyme zusätzliche *in silico* Experimente durchgeführt, wobei die oben genannten Versuchsbedingungen (Konfiguration der Software und Zusammenstellung der TF nach Signifikanz für *Homo sapiens*) berücksichtigt wurden. Die Ergebnismenge der *in silico* Experimente beinhaltet zahlreiche Ergebnisse, weshalb eine nachträgliche Datenanalyse durch den Experimentator zwingend notwendig ist. Dabei kann der Experimentator durch spezialisierte Anwendungssoftware (CELLmicrocosmos und VANESA) unterstützt

werden, sodass eine Datenanalyse ohne großen Aufwand möglich ist. Die nächsten beiden Tabellen zeigen für einige Gene (NDST1, GLCE, HS2ST1, HS3ST1 und HS6ST1) die fünf besten Ergebnisse der 5'-Upstream-Region (siehe Tabelle 6.11) und der 3'-Downstream-Region (siehe Tabelle 6.12). Diese Ergebnisse wurden unter Zuhilfenahme der bereits erwähnten Auswahlkriterien (*Core-Region-Score*, *Score* und Position der potenziellen TFBS) selektiert.

Anhand der Tabellen 6.8, 6.9, 6.11 und 6.12 wird deutlich, dass beispielsweise die STAT-Proteine (*Signal Transducers and Activators of Transcription*), die SMAD-Proteine und weitere TF wie WT1 und der *Hepatocyte nuclear factor* 4 (HNF4) verstärkt in der Ergebnismenge der *in silico* Experimente vorkommen. Diese Proteine werden in den vier Tabellen durch eine farbliche Kennzeichnung explizit hervorgehoben. Es werden als nächstes die charakteristischen Eigenschaften der STAT- und SMAD-Proteine exemplarisch beschrieben.

Die STAT-Proteine können über den intrazellulären JAK-STAT-Signalweg[5] (Januskinase) das Immunsystem, das Zellwachstum und die Zellproliferation beeinflussen [PDF97, KBBS02]. Es wurden bis zum jetzigen Zeitpunkt sieben STAT-Proteine (STAT1 - STAT4, STAT5A, STAT5B und STAT6) identifiziert, die als TF fungieren. Außerdem sind einige Krankheitsbilder wie das Laron-Syndrom auf Genmutationen oder -defekte innerhalb der kodierenden Gene der STAT-Proteine zurückzuführen [KHL+03].

Der extrazelluläre Wachstumsfaktor *Transforming Growth Factor* (TGF) ist bei der Embryogenese und der Zelldifferenzierung im adulten Organismus beteiligt und wird in zwei Typen (TGF-α und TGF-β) unterschieden. Die zellulären Prozesse wie Zellproliferation und -differenzierung können durch TGF-β beeinflusst werden, das ein stark konserviertes Protein ist und für *Mammalia* in drei Isoformen (TGF-β1, -β2, und -β3) existiert [GJFS00]. Durch TGF-β werden auch die intrazellulären SMAD-Proteine aktiviert. Diese Proteine werden in die folgenden drei Gruppen unterteilt:

1. SMAD1, SMAD2, SMAD3, SMAD5 und SMAD8/9 werden durch den Rezeptor reguliert und als R-SMAD bezeichnet [WHC+01].

2. Die zweite Gruppe umfasst ausschließlich SMAD4, der als *common-mediator* SMAD (co-SMAD) fungiert und mit R-SMAD interagiert [SHL+97].

3. SMAD6 und SMAD7 sind antagonistische oder inhibitorische SMAD (I-SMAD) und blockieren die Aktivierung der R-SMAD und der co-SMAD [IAS+01].

Die SMAD-Proteine können zusammen mit Coaktivatoren oder -repressoren die Effizienz der Genexpression positiv oder negativ beeinträchtigen. Der

[5] siehe Abbildung D.3

Score	*Core-Region-Score*	TF	Gen	Position*	TFBS
0.91	1.00	MAZ	NDST1	12	cgggaggg
0.94	1,00	TFAP2A, AP-2alphaA		14	gccgggagg
1.00	1.00	c-Myc, USF1, Max-isoform2		42	cacgtgg
1.00	1.00	LEF-1, TCF-1		59	tcaaag
0.99	0.98	HNF-4		62	gggtca
0.99	1.00	SPI1	GLCE	16	gggaag
0.97	1.00	Sp1, Sp2, Sp3, Sp4		46	ccccgcccct
0.97	0.99	WT1, WT1-del2 WT1 -KTS		59	ccccccgcc
0.92	1.00	Smad1, Smad2-L Smad3, Smad4		33	agactccga
0.94	1.0	HOXA4		148	acaattgg
0.96	0.95	TEF-1	HS2ST1	27	gggatg
0.97	1.00	MZF1_1-4		29	cgggga
0.95	1.00	E2F-1, E2F-4, DP-1, E2F:DP		31	ggcggg
≥0.93	1.00	Sp1, Sp3, Sp4, Sp3-isoform1		33	ggggcgggga
0.99	1.00	HIF-1		65	tgcgtgcgc
0.93	1.00	STAT5A	HS3ST1	12	cctttctc
0.90	1.00	HOXA5		38	cttaaatt
0.95	1.00	BRCA1		67	ccaaccc
≥0.89	1.00	STAT3, STAT5A		72	cgattcca
0.92	1.00	pax2		93	ggcaaaccc
0.94	1.00	TFAP2A, AP-2alphaA	HS6ST1	18	gccccgcgc
0.93	1.00	ZBTB7B		35	gccctccca
0.90	1.00	T3R-alpha1, RAR-beta2		38	cctgccctc
0.95	1.00	WT1, WT1-del2, WT1 -KTS		42	gcctcctgc
0.91	1.00	c-Ets-2		41	cctcctg

* Die Position der potenziellen TFBS befindet sich relativ zum Start des Gens.

Tabelle 6.11: TF und deren DNA-Bindestellen, die stromaufwärts bei humanen Genen lokalisiert sind, deren Enzyme an der Biosynthese von HS beteiligt sind.

Score	*Core-Region-Score*	TF	Gen	Position*	TFBS
0.93	1.00	RXR-alpha, RXR-beta	NDST1	13	tctgaccttt
0.99	1.00	Nur77		14	ctgacctttt
0.99	1.00	SPI1		37	gggaag
0.92	0.97	HNF-4, HNF-4alpha		68	cggcca
0.90	1.00	pax2, pax2-isoform1		70	gccaaagcc
≥0.90	1.00	STAT4, STAT5A	GLCE	12 63	tatttctc gatttctg
0.92	1.00	IRF-1		15	ttctctt
0.93	1.00	HOXA5		26	ctcaaatt
0.93	1.00	YY1		71	accatt
0.92	1.00	ESE-1		73	catttcctgt
0.98	1.00	Nkx6-2	HS2ST1	4	gaattaattatt
0.95	1.00	LHX3a, LHX3b		5	aattaattat
0.97	1.00	TFIID, TBP		14	tttatat
0.90	1.00	PARP		20	tgagaaaaat
≥0.96	1.00	STAT4, STAT5A		46	gaattcta
0.92	1.00	STAT5A	HS3ST1	12	tctttctt
0.91	1.00	GATA3		21	tgattg
0.95	1.00	BRCA1		26	gcaaccc
0.92	1.00	NFIC		34	ctggca
1.00	1.00	FOXC1		62	ggtaagta
0.92	1.00	NFIC	HS6ST1	0	ctggca
0.93	1.00	TFAP2A, AP-2alphaA		11	gccctcagc
0.99	1.00	HNF-4, HNF-4alpha		23	agggca
0.89	1.00	Smad1, Smad2-L, Smad3, Smad4 Smad6, Smad7		47	tccccagacac
0.95	1.00	Sp1, Sp2, Sp3, Sp4		115	ccccgcccca

* Die Position der potenziellen TFBS befindet sich relativ zum Ende des Gens.

Tabelle 6.12: TF und deren DNA-Bindestellen, die stromabwärts bei humanen Genen lokalisiert sind, deren Enzyme an der Biosynthese von HS beteiligt sind.

TGF-β-Signalweg[6] reguliert zahlreiche zelluläre Prozesse wie Zellproliferation, -differenzierung, -adhäsion und Apoptose, wobei die SMAD-Proteine und das Signalmolekül *bone morphogenetic protein* beteiligt sind. Dieses Signalmolekül wird durch Sulf1 und/oder Sulf2 positiv reguliert (siehe Tabelle 6.2). Die fehlerhafte Regulation des TGF-β-Signalweges ist für das Marfan-Syndrom, das Loeys-Dietz-Syndrom und einige Krebserkrankungen verantwortlich [BSL00, EB05, MSH$^+$09]. Das Grundprinzip der Signalübertragung des TGF-β-Signalweges ähnelt der Signaltransduktion des JAK-STAT-Signalweges [WM09].

6.4 Zusammenfassung

Der Anwendungsfall, der in Kapitel 6 behandelt wurde, basiert auf einer spezifischen Thematik aus der molekularbiologischen Grundlagenforschung. Durch den Anwendungsfall sollte der Funktionsumfang, der Anwendungsbereich und der praktische Stellenwert von TraBi und DAWIS-M.D. verdeutlicht werden. Insbesondere fachspezifische *in silico* Experimente, die mittels TraBi durchgeführt wurden, charakterisieren den Anwendungsfall. Die daraus resultierenden Ergebnisse können der Ausgangspunkt für zukünftige experimentelle Studien oder Hypothesen sein.

Als erstes wurde in Abschnitt 6.1 die Klassifikation und die Struktur der Sulfatasen beschrieben, die in drei Typen unterteilt werden. Diese Enzyme katalysieren die hydrolytische Spaltung von Sulfatestern und Sulfamaten. Danach thematisierte der Abschnitt 6.2 einen spezifischen TF, der als TFEB bezeichnet wird und zahlreiche lysosomale Gene positiv reguliert. Diese molekularbiologische Thematik fungierte als Grundlage für die experimentellen Studien, deren Ergebnisse und Schlußfolgerungen in Abschnitt 6.3 erläutert wurden.

Das nächste Kapitel fasst zentrale Themenbereiche zusammen und erläutert die Schlussfolgerungen und Thesen der vorliegenden Arbeit. Darüber hinaus werden zukünftige Implementierungen und Optimierungen von DAWIS-M.D. und TraBi behandelt, die das wissenschaftliche Spektrum, die Systemfunktionalität und die Benutzerfreundlichkeit der beiden Softwarelösungen verbessern.

[6] siehe Abbildung D.4

7 Zusammenfassung und Ausblick

Das abschließende Kapitel fasst die vorliegende Arbeit zusammen und gibt einen Ausblick, der mögliche Verbesserungen und Weiterentwicklungen von DAWIS-M.D. und TraBi erläutert.

Als erstes werden in Abschnitt 7.1 die wesentlichen Themenbereiche der Dissertation zusammengefasst. Dabei werden auch die zentralen Schlussfolgerungen und Thesen der vorliegenden Arbeit vorgestellt. Der Ausblick, der zukünftige Implementierungen und Optimierungen von DAWIS-M.D. und TraBi thematisiert, wird in Abschnitt 7.2 behandelt. Die möglichen Erweiterungen und Verbesserung der beiden Softwarelösungen, die in Abschnitt 7.2 genannt werden, sind besonders für das wissenschaftliche Spektrum als auch für die Systemfunktionalität und Benutzerfreundlichkeit der Software sinnvoll. Diese zusätzlichen Optimierungen können die Signifikanz und die Akzeptanz von DAWIS-M.D. und TraBi in der Wissenschaftsgemeinde verbessern. Durch zukünftige Softwareprojekte, die Erweiterungen und Verbesserungen bei DAWIS-M.D. und TraBi realisieren, wird die langfristige Softwarequalität der beiden Softwarelösungen gewährleistet. Auf diese Weise ist DAWIS-M.D. und TraBi auch für zukünftige Forschungsprojekte aus den Lebenswissenschaften interessant.

7.1 Zusammenfassung

Die unterschiedlichen Forschungsgebiete der Lebenswissenschaften erzeugen durch verschiedene experimentelle Methoden eine immense und vielschichtige Datenmenge. In der Regel werden solche Datenbestände in DBS gespeichert. Auf diese Weise wird eine effiziente, persistente und widerspruchsfreie Speicherung, Manipulation und Verwaltung der Daten gewährleistet. Durch IS, deren zentrale Komponente häufig ein DBS ist, sind die molekularbiologischen Daten für die akademische und industrielle Grundlagenforschung frei zugänglich sowie global verfügbar. Das jährliche *Database Issue* der Zeitschrift NAR listet derzeit etwa 1552 molekularbiologische DB, die Informationen aus unterschiedlichen Kategorien der Molekularbiologie bereitstellen [FSRG14]. Die Anzahl der molekularbiologischen DB als auch deren Dankbankeinträge ist in den letzten Jahren exponentiell angestiegen (siehe Abbildungen 1.1 und 1.2). Dafür sind vor allem der technologische Fortschritt und die compu-

tergestützte Laborautomatisierung verantwortlich. Die Merkmale eines IS und eines DBS als auch die Thematik der molekularbiologischen DB und deren Klassifizierung wurden in Abschnitt 2.2.2 erläutert.

Aufgrund der Datenmenge und deren Komplexität müssen Wissenschaftler aus den Lebenswissenschaften bei der Forschungstätigkeit unterstützt werden. Außerdem sind für ganzheitliche Fragestellungen der Lebenswissenschaften, die beispielsweise (in)direkte Wechselwirkungen und Abhängigkeiten zwischen Gen, Protein und Krankheit sowie genetische Regulationsmechanismen behandeln, verschiedene Informationen notwendig. Infolgedessen sind zum einen spezialisierte Softwarelösungen erforderlich, zum anderen eine konsistente und homogene Datenbasis, die umfangreiche molekularbiologische Datenbestände beinhaltet. Als Datenquelle für eine solche Datenbasis müssen mehrere kommerzielle und öffentlich frei verfügbare molekularbiologische DB fungieren, deren Informationsqualität und -aktualität durch international etablierte Organisationen bzw. Unternehmen gewährleistet wird. Durch die Bioinformatik werden entsprechende Lösungsansätze bereitgestellt, wobei die Datenintegration, Sequenzanalyse und computergestützte Visualisierung der Daten wesentliche Aspekte sind.

Die zentralen Anforderungen und Methoden der Datenintegration wurden in Abschnitt 2.2.3 beschrieben. Dabei wurde deutlich, dass eine konsistente und homogene Datenbasis das Ziel der Datenintegration ist. Die drei Hauptprobleme der Datenintegration sind die Verteilung, Autonomie und Heterogenität, die auch als die orthogonalen Dimensionen der Datenintegration bezeichnet werden [LN07]. Diese Probleme können durch verschiedene Integrationsarchitekturen beseitigt werden, wobei zwischen materialisierten und virtuellen Integrationsarchitekturen unterschieden wird. Das DWH ist eine materialisierte Integrationsarchitektur und die vorherrschende Integrationsarchitektur in der Bioinformatik. Aufgrund der in Abschnitt 2.2.3.4 genannten Vorteile basieren zahlreiche Softwarelösungen auf der Data-Warehouse-Technik. Das DWH ist die zentrale Komponente eines Data-Warehouse-Systems, das ein spezielles IS ist und über einen Datenbeschaffungsbereich und einen Auswertebereich verfügt (siehe Abbildung 2.6).

Es sind zahlreiche Softwarelösungen aus der (Bio-)Informatik verfügbar, die kommerziell oder frei verfügbar sind und verschiedene Vor- und Nachteile aufweisen. Die Software aus der Bioinformatik ist überwiegend auf bestimmte Anwendungsbereiche wie Datenintegration, -analyse, oder -visualisierung spezialisiert. Außerdem sind durch einige Softwarelösungen computergestützte Simulationen und Vorhersagen möglich, die als *in silico* Experimente bezeichnet werden. Diese Experimente sind eine Alternative zu den klassischen Experimenten der Lebenswissenschaften, die entweder *in vitro* oder *in vivo* durchgeführt werden. Die *in vitro* und *in vivo* Experimente sind zeit- und kostenaufwendig, verbrauchen Ressourcen und liefern im schlechtesten Fall keine eindeutigen Ergebnisse. Deswegen ist es sinnvoll, als erstes *in silico* Experimente durchzuführen, weil die daraus resultierenden Ergebnisse neue Anregungen für zukünftige Laborexperimente liefern können sowie Hypothe-

sen bestätigen oder widerlegen, sodass keine aufwendigen experimentellen Studien notwendig sind. Die Analyse und die Interpretation der Genregulation, welche für die Steuerung der Genexpression zuständig ist, ist ein traditionelles Forschungsgebiet der Bioinformatik. Dabei sind spezielle Proteine involviert, sogenannte TF, die mit regulatorischen Nukleotidsequenzen, den TFBS, oder DNA-bindenden Proteinen interagieren können und so die Transkription aktivieren oder reprimieren (siehe Abschnitt 2.1.2). Die computergestützte Identifikation von regulatorischen Elementen in Nukleotidsequenzen ist durch spezifische Anwendungssoftware möglich.

Der aktuelle Stand der Forschung wurde in Kapitel 3 beschrieben, wobei dieses Kapitel ausschließlich verwandte Arbeiten aus der Bioinformatik behandelt. Die Softwarelösungen der verwandten Arbeiten sind entweder populäre Ansätze der Datenintegration in der Bioinformatik (siehe Abschnitt 3.1) oder ermöglichen die computergestützte Identifikation von regulatorischen Elementen in Nukleotidsequenzen (siehe Abschnitt 3.2). Die Vor- und Nachteile der verwandten Arbeiten wurden durch einen Vergleich und eine Bewertung deutlich, wofür ausgewählte Kriterien festgelegt wurden (siehe Abschnitt 3.3). Allerdings sind nicht alle Leistungsmerkmale, Systemfunktionalitäten und -eigenschaften der verwandten Arbeiten in der Literatur beschrieben. Deswegen wurde die Software der verwandten Arbeiten mittels Black-Box-Tests, die eine Testmethode der Softwaretests sind, zusätzlich evaluiert. Dadurch konnten funktionale und nicht-funktionale Anforderungen bei den Softwarelösungen der verwandten Arbeiten überprüft werden. Die Nachteile der verwandten Arbeiten lassen sich auf folgende Punkte zusammenfassen:

- Die Software wurde ausschließlich für ein spezielles Forschungsprojekt entwickelt, das eine bestimmte molekularbiologische Fragestellung behandelt. Das bedeutet, dass die Software für andere Forschungsprojekte und deren Fragestellungen nicht geeinigt ist, weil die Weiterentwicklung der ursprünglichen Systemfunktionalität bzw. Datenbasis äußerst aufwendig ist.

- Der Funktionsumfang der Software oder der Datenbestand der zugrundeliegenden Datenbasis ist explizit auf einen bestimmtem Organismus ausgerichtet. Außerdem sind einige Softwarelösungen entweder auf prokaryotische oder eukaryotische Organismen spezialisiert.

- Die zugrundeliegende Datenbasis der Software ist für komplexe Fragestellungen aus der Molekularbiologie nicht geeignet oder beinhaltet keine vollständigen Datenbestände. Darüber hinaus ist die Informationsqualität und -aktualität der Datenbasis bei einigen Softwarelösungen nicht mehr gewährleistet.

- Außerdem unterliegen die Softwarelösungen unterschiedlichen informationstechnischen Nachteilen. Dafür sind die fehlende Funktionalität einer Netzwerkvisualisierung und der Export der Datenbestände in standardisierte Austauschformate, die aufwendige Software-Migration auf andere RDBMS und die schlechte Zeit- und Platzkomplexität der Algorithmen entsprechende Beispiele.

Darüber hinaus kann kommerzielle Software unter gewissen Umständen kritisch betrachtet werden, weil besonders finanzschwache Forschungseinrichtungen keine finanziellen Mittel für Softwarelizenzen bereitstellen können. Deswegen werden häufig eigene Softwarelösungen implementiert oder geeignete OSS eingesetzt. Allerdings werden die Vor- und Nachteile von *Open Source* vs. kommerzielle Software durchaus kontrovers diskutiert, weshalb kommerzielle Software als auch OSS nicht per se als Vor- oder Nachteil bezeichnet werden kann.

Der praktische Teil der Dissertation wurde in Kapitel 4, 5 und 6 erläutert, wobei Kapitel 4 und 5 zwei Ansätze aus der Bioinformatik und deren Implementierung beschreiben, die Wissenschaftler aus den Lebenswissenschaften bei unterschiedlichen Forschungsprojekten unterstützen. Die daraus resultierenden Softwarelösungen sind einerseits ein webbasiertes Data-Warehouse-System für molekularbiologische Daten, welches durch das Akronym DAWIS-M.D. repräsentiert wird, andererseits ein webbasiertes IS, das als TraBi bezeichnet wird und die computergestützte Vorhersage von potenziellen TFBS in Nukleotidsequenzen ermöglicht. Durch die Schlussfolgerungen aus Kapitel 3 und die Anforderungsanalyse in Kapitel 4 wurde deutlich, dass für aktuelle und zukünftige Forschungsprojekte aus den Lebenswissenschaften eine umfangreiche Datenbasis notwendig ist, die Datenbestände über verschiedene Domänen (z. B. *Gene*, *Protein* und *Disease*) und Organismen (Eukaryoten und Prokaryoten) beinhaltet. Dabei ist vor allem die Informationsqualität und -aktualität der Datenbasis sicherzustellen, weshalb geeignete molekularbiologische DB als Datenquelle fungieren müssen, deren Pflege/Weiterentwicklung, Datenintegrität und -konsistenz durch international etablierte Organisationen bzw. Unternehmen gewährleistet wird. Mit Hilfe der gerade genannten und weiteren Kriterien in Abschnitt 4.3.2 konnten 14 molekularbiologische DB identifiziert werden (siehe Tabelle 4.1), die als Datenquelle für die zugrundeliegende Datenbasis von DAWIS-M.D. und TraBi fungieren.

Aufgrund der Argumente in Abschnitt 2.2.3.4 und der genannten Vorteile in [LR03, LN07, GB09] wurde die Datenbasis als DWH realisiert. Außerdem ist die Data-Warehouse-Technik die vorherrschende Integrationsarchitektur in der Bioinformatik (siehe Abschnitt 3.1). Anhand der Abbildung 4.5 wurde deutlich, dass zwischen den molekularbiologischen DB explizite Querverweise und/oder Fremdschlüssel existieren. Diese Beziehungen symbolisieren häufig molekularbiologische Mechanismen der Regulation und Interaktion oder Zusammenhänge zwischen unterschiedlichen Domänen. Infolgedessen wurden in Abschnitt 4.3.3 vier Maßnahmen vorgestellt, mittels denen Beziehungen zwischen molekularbiologischen DB identifiziert werden können. Unter Verwendung der vier Maßnahmen konnten zahlreiche (in)direkte Beziehungen zwischen den 14 molekularbiologischen DB (siehe Tabelle 4.1) identifiziert werden, die schematisch in der Abbildung 4.6 dargestellt wurden. Die molekularbiologischen DB werden unterschiedlichen Klassifikation zugeordnet (siehe Abschnitt 2.2.2.2), sodass eine Segmentierung der Datenbestände auf verschiedene Domänen erforderlich ist. Die Datenbestände der 14 molekularbiologischen DB, die Tabelle 4.1 zeigt, konnten auf 13 Domänen aufgeteilt werden (siehe Tabelle 4.2). Das abstrakte Datenmodell in der Abbildung 4.7 ist auf diese Domänen zu-

rückzuführen, wobei die Beziehungen zwischen den einzelnen Domänen sich aus der Segmentierung der Datenbestände ergeben. Dieses Datenmodell wurde bei DAWIS-M.D. umgesetzt und reflektiert die Datenbestände der 14 molekularbiologischen DB (siehe Tabelle 4.1) als auch deren Beziehungen. Das Datenmodell umfasst somit 13 Domänen, die spezifische Wissensbereiche der Molekularbiologie wie *Gene*, *Protein* oder *Disease* repräsentieren.

Der Prozess der Datenintegration wurde mittels BioDWH durchgeführt, das eine flexible und plattformunabhängige Software-Infrastruktur ist, die spezifische *Parser* für die Integration von molekularbiologischen DB zur Verfügung stellt. Der Hauptbestandteil der *Parser* ist ein ETL-Prozess, der für die Extraktion und Transformation der Daten sowie für das Laden der Daten ins DWH zuständig ist. Die Querverweise und/oder Fremdschlüssel zwischen den molekularbiologischen DB werden ebenfalls durch den *Parser* identifiziert und in spezielle Datenbanktabellen gespeichert, die als Mapping-Tabellen bezeichnet werden. Das DWH besteht aus zwei DB und einem DBMS, wobei das DBMS für die Datenverwaltung verantwortlich ist und die beiden DB verschiedene Datenbestände beinhalten. Die DB *metadata* verfügt ausschließlich über Metadaten, welche für die Datenanalyse, System- und Benutzerverwaltung notwendig sind. Im Gegensatz dazu werden die Datenbestände der 14 molekularbiologischen DB (siehe Tabelle 4.1) in der DB *dawismd* gespeichert. Das Datenbankschema der DB *metadata* wurde in der Abbildung 4.8 dargestellt. Die Struktur des Datenbankschemas der DB *dawismd* ist äußerst komplex und umfasst gegenwärtig 526 Datenbanktabellen. Deswegen wurde das Datenbankschema der DB *dawismd* durch zwei exemplarische Beispiele erläutert (siehe Abbildungen 4.9 und 4.10). Die 14 molekularbiologischen DB (siehe Tabelle 4.1), die als Datenquellen für das DWH fungieren, werden im DWH jeweils durch ein eigenes Datenbankschema repräsentiert, das entweder auf dem Sternschema (siehe Abbildung 4.9) oder dem Schneeflockenschema (siehe Abbildung 4.10) basiert. Die gerade beschriebene Vorgehensweise wird als lose Kopplung bezeichnet und zeichnet sich durch folgende Vorteile aus:

- Die Datenbestände einer oder mehrerer Datenquellen können problemlos im DWH aktualisiert, modifiziert oder gelöscht werden.
- Durch eine eindeutige Bezeichnung der Datenbanktabellen im DWH ist eine unkomplizierte Lokalisierung und Identifizierung der jeweiligen Datenquellen und deren Datenbestände möglich, sodass keine zusätzlichen Metadaten notwendig sind.
- Sofern das Datenformat oder -schema einer Datenquelle modifiziert oder erweitert wurde, kann das entsprechende Datenbankschema im DWH ohne großen Aufwand angepasst werden.

Aufgrund der in Abschnitt 2.2.2 genannten Vorteile wurde als DBMS ein RDBMS eingesetzt. Das DWH, welches zwei DB (*dawismd* und *metadata*) und ein RDBMS

repräsentiert, ist die zugrundeliegende Datenbasis für DAWIS-M.D. und TraBi, wobei MySQL als RDBMS eingesetzt wurde. Das webbasierte Data-Warehouse-System für molekularbiologische Daten, das als DAWIS-M.D. bezeichnet wird, bietet eine homogene, konsistente und integrierte Sicht auf das DWH. Dabei erfolgt die Schematransformation und -integration auf der Ebene der Anwendungslogik, weil die lose Kopplung als Paradigma eingesetzt wurde. Es sind bei DAWIS-M.D. insgesamt 13 Domänen verfügbar, die verschiedene Wissensbereiche der Molekularbiologie repräsentieren. Aufgrund der Spezifität der jeweiligen Domänen und deren Datenbestände wird für jede Domäne ein spezialisiertes Suchformular mit unterschiedlichen Filter- und Suchmöglichkeiten zur Verfügung gestellt (siehe Tabelle 5.3). Die Querverweise zwischen den einzelnen Domänen und deren Datensätzen symbolisieren Beziehungen und/oder Abhängigkeiten, die wiederum PPI oder andere regulatorische Wechselwirkungen repräsentieren können. Diese Informationen eines Datensatzes können bei DAWIS-M.D. in standardisierte Austauschformate exportiert oder durch eine zusätzliche Softwarekomponente, die eine Netzwerkvisualisierung ermöglicht, als dynamisches und interaktives Netzwerk visualisiert werden (siehe Abbildung 5.7).

Die computergestützte Identifikation von regulatorischen Elementen in Nukleotidsequenzen ist für die Bioinformatik eine Herausforderung, weil einerseits die Zeit- und Platzkomplexität der Algorithmik wichtig ist, andererseits möglichst viele falsch positive Ergebnisse beseitigt werden müssen. Dafür bietet die Bioinformatik zahlreiche Softwarelösungen, die aber verschiedene informationstechnische Vor- und Nachteile aufweisen und für aktuelle molekularbiologische Forschungsprojekte nicht immer geeignet sind (siehe Abschnitt 3.2). Infolgedessen wurde in der vorliegenden Arbeit ein verbessertes Konzept für die computergestützte Vorhersage von potenziellen TFBS in Nukleotidsequenzen vorgestellt. Die daraus resultierende Softwarelösung ist ein webbasiertes IS, das als TraBi bezeichnet wird. Es gibt unterschiedliche Lösungsansätze um potentielle TFBS in Nukleotidsequenzen vorherzusagen. Diese Ansätze basieren häufig auf PSSM, HMM oder *de novo*, wobei die PSSM in der Praxis weit verbreitet sind. Außerdem sind etliche Softwarelösungen verfügbar, deren Lösungsansätze auf der PSSM basieren. Die PSSM beinhaltet auch die Variabilität der TFBS bei TF, sodass die Gewichtung der Nukleotide an den einzelnen Positionen berücksichtigt wird. Im Gegensatz dazu ist die Konsensussequenz nicht geeignet für die Identifikation von regulatorischen Elementen in Nukleotidsequenzen, weil die Ergebnismenge zahlreiche falsch positive Ergebnisse beinhaltet [DM92]. Diese positiven Merkmale sind ausschlaggebend für einen Lösungsansatz, der auf PSSM basiert. Als wichtige Leistungsmerkmale für TraBi wurden während der Anforderungsanalyse eine lineare Zeitkomplexität und eine minimale Platzkomplexität der Algorithmik festgelegt (siehe Abschnitt 4.3.1). Die gerade genannten Kriterien gewährleistet der nicht-heuristische Algorithmus *ESAsearch*, der eine lineare Zeitkomplexität und eine geringe Platzkomplexität besitzt [BHGK06]. Dabei wird als Datenstruktur das ESA eingesetzt, das die Vorteile der Suffix-Bäume und -Arrays kombiniert (siehe Abschnitt 2.2.1.1). Der Algorithmus *ESAsearch* wurde im Rahmen der Dissertation um zusätzliche Bestandteile ergänzt, wodurch zum einen die Anzahl der falsch positiven

Ergebnisse reduziert wurde, zum anderen die Effizienz des Algorithmus nochmals optimiert wurde. Dieses wurde durch die folgenden drei Ansätze ermöglicht:

1. Die ESA, die auf Nukleotidsequenzen der 5'-Upstream-Region oder der 3'-Downstream-Region basieren und jeweils eine Länge bis zu 2500 bp, 5000 bp oder 10000 bp repräsentieren, werden nicht zur Laufzeit angelegt, weil das einen „Flaschenhals“ erzeugen würde und die Algorithmik negativ beeinträchtigt. Deswegen wurde der Prozess der Datenakquisition um eine zusätzliche Vorverarbeitung ergänzt. Das bedeutet, dass die ESA während der Datenakquisition angelegt werden und als Datei im Dateisystem persistent gespeichert werden. Die persistente Speicherung im DWH erfolgt nicht, weil auf diese Weise die Komplexität unnötig erhöht wird.

2. Das DWH umfasst Datenbestände über TF, TFBS und PSSM, die ursprünglich aus den Datenquellen JASPAR und TRANSFAC® resultieren. Dabei variiert die Länge der TFBS von Datensatz zu Datensatz zwischen 4 und 33 bp. Allerdings wird in [FSS11] die charakteristische Länge einer TFBS mit 6 - 12 bp und in seltenen Kombinationen bis zu 18 bp beschrieben. Infolgedessen wurde eine definierbare Vorbedingung hinzugefügt, welche die minimale und maximale Länge der TFBS berücksichtigt. Dadurch profitiert die Effizienz der Algorithmik, weil die Gesamtmenge der Datensätze, die der Algorithmus analysieren müsste, bereits im voraus reduziert wird. Außerdem wird die Ergebnismenge verfeinert, weil ausschließlich Datensätze berücksichtigt werden, welche die definierte Vorbedingung gewährleisten.

3. Es wird für jede ausgewählte PSSM zuerst eine *core region* berechnet, welche die am stärksten konservierten und benachbarten Positionen repräsentiert. Die *core region* wird als erstes analysiert, wodurch die Performance des Algorithmus deutlich verbessert wird, weil die Konserviertheit der *core region* das *Lookahead scoring* in puncto Effizienz optimiert. Deswegen kann die Methode *skipchain*, die ein Bestandteil von *ESAsearch* ist, wesentlich mehr Suffixe bei der Analyse ignorieren. Damit eine *core region* der Länge eins nicht möglich ist, wurde eine minimale Länge für die *core region* definiert. Sofern eine *core region* über einen schlechten *score* verfügt, ist die Interaktion mit TF äußerst unwahrscheinlich. Deswegen werden diese Ergebnisse nicht berücksichtigt, weil deren Signifikanz für die molekularbiologische Grundlagenforschung mit großer Wahrscheinlichkeit sehr gering ist.

Die erweiterte Variante des Algorithmus *ESAsearch* wurde bei TraBi umgesetzt. Auf diese Weise ermöglicht TraBi eine leistungsfähige Vorhersage von potenziellen TFBS in Nukleotidsequenzen. Im Gegensatz zu Match™ und MatInspector, deren Zeitkomplexität laut [BHGK06] bei $\mathcal{O}(mn)$ taxiert wird, ist das eine erkennbare Verbesserung. Durch TraBi ist es möglich potentielle TFBS sowohl in der 5'-Upstream-Region als auch in der 3'-Downstream-Region vorherzusagen, wobei die

entsprechenden Nukleotidsequenzen eine Länge bis zu 2500 bp, 5000 bp oder 10000 bp aufweisen können. Allerdings verfügt TraBi derzeit über genetische Informationen von drei eukaryotischen Organismen (*Homo sapiens*, *Mus musculus* und *Rattus norvegicus*). Die Ergebnismenge, die aus einer Vorhersage resultiert, kann bei TraBi in standardisierte Austauschformate exportiert werden, sodass eine Datenauswertung in spezialisierte Softwarelösungen möglich ist. Als Datenbasis fungiert bei TraBi das DWH, welches Datenbestände über Gene, TF, TFBS und PSSM beinhaltet, die ursprünglich aus den Datenquellen Ensembl, JASPAR und TRANSFAC® resultieren. Es ist aber auch möglich benutzerspezifische PSSM bei TraBi anzulegen, die für alle registrierten Benutzer frei zugänglich sind und auf wissenschaftlichen Publikationen und/oder auf experimentelle Studien basieren können.

Die Systemarchitekturen der beiden Softwarelösungen (siehe Abbildungen 4.12 und 4.13), die im Rahmen der Dissertation implementiert wurden, basieren jeweils auf der N-Schichten-Architektur. Diese Architektur ermöglicht gegenüber der 2- und 3-Schichten-Architektur eine Reduzierung der Komplexität, die lose Kopplung zwischen den einzelnen Schichten und eine starke Kohäsion der Schichten. Damit zwischen der Anwendungsschicht und der Datenbankschicht keine direkte Abhängigkeit existiert, wurden beide Systemarchitekturen um eine zusätzliche Schicht erweitert, die als Persistenzschicht bezeichnet wird. Auf diese Weise wurde der konzeptionelle Widerspruch *object-relational impedance mismatch* beseitigt (siehe Abschnitt 2.2.2.1). Außerdem können durch die Persistenzschicht verschiedene RDBMS eingesetzt werden ohne die Anwendungslogik anzupassen. Die beiden Softwarelösungen wurden jeweils als benutzerfreundliche und plattformunabhängige Webanwendung implementiert, wobei etablierte Programmier- und Skriptsprachen sowie standardisierte Entwurfsmuster eingesetzt wurden.

Der Anwendungsfall, der in Kapitel 6 beschrieben wurde, erläutert eine spezifische Thematik aus der molekularbiologischen Grundlagenforschung. Dabei wurde zum einen die Familie der Sulfatasen (siehe Abschnitt 6.1) behandelt, zum anderen der spezifische TF TFEB (siehe Abschnitt 6.2). Die *in silico* Experimente, die mittels Trabi durchgeführt wurden, sind auf die jeweiligen Fragestellungen der molekularbiologischen Thematik zurückzuführen. Die daraus resultierenden Ergebnisse wurden in Abschnitt 6.3 erläutert und liefern neue Anregungen für zukünftige Laborexperimente. Infolgedessen konnten Hypothesen bestätigt oder widerlegt werden, sodass keine aufwendigen experimentellen Studien durchgeführt werden müssen. Es konnten einerseits bereits bekannte genregulatorische Mechanismen der Sulfatasen nachgewiesen werden, andererseits wurden auch unbekannte bzw. experimentell noch nicht verifizierte TF der Sulfatasen identifiziert. Insbesondere für das humane Gen der Arylsulfatase K konnte eine potentielle TFBS für TFEB in der 5'-Upstream-Region vorhergesagt werden, die durch experimentelle Studien bestätigt wurde. Dabei wurde durch Überexpression von TFEB in HT1080 eine erhöhte Expression für ARSK nachgewiesen, weshalb es als sehr wahrscheinlich gilt, dass TFEB einen positiven regulatorischen Effekt auf ARSK ausübt. Diese Hinweise unterstützen auch die Hypothese, dass die Arylsulfatase K im Lysosom lokalisiert ist, weil TFEB die Tran-

skription von zahlreichen lysosomalen Genen positiv reguliert [SPdR+09, PIK+11].

7.2 Ausblick

Damit DAWIS-M.D. und TraBi auch bei zukünftigen Forschungsprojekten aus den Lebenswissenschaften berücksichtigt werden, sollten zusätzliche Softwareprojekte durchgeführt werden. Auf diese Weise wird die Softwarequalität sichergestellt, weil die beiden Softwarelösungen kontinuierlich weiterentwickelt und verbessert werden. Durch die Optimierung der Software sind neue Systemfunktionalitäten und -eigenschaften verfügbar, wovon unter anderem die Benutzerfreundlichkeit und das wissenschaftliche Spektrum profitiert. Außerdem könnte die Informationsqualität und -aktualität der Datenbasis gewährleistet werden, weil das DWH regelmäßig aktualisiert wird. Die folgenden drei Punkte beschreiben potenzielle Erweiterungen und Verbesserungen für TraBi, DAWIS-M.D. oder das DWH, welches die zugrundeliegende Datenbasis der beiden Softwarelösungen ist:

1. Das DWH beinhaltet derzeit Datenbestände, die aus 14 molekularbiologischen DB (siehe Tabelle 4.1) resultieren. Allerdings sollte das DWH auch den Datenbestand der Datenquellen IntAct [KAB+12], iProClass [HBCW03], MINT [LBP+12], PROSITE und Reactome [MGG+09] umfassen, sodass zusätzliche Informationen und Domänen verfügbar sind. Die dazu notwendigen *Parser* werden bereits durch BioDWH bereitgestellt, sodass die Integration der Datenquellen problemlos möglich ist.

   Darüber hinaus könnte für das DWH ein globales und einheitliches Datenbankschema konzipiert werden, welches auf dem Datenmodell in der Abbildung 4.7 basiert. Diese Vorgehensweise wird als enge Kopplung bezeichnet. Infolgedessen ist eine Schematransformation und -integration auf Ebene der Anwendungslogik nicht mehr notwendig.

2. Das Suchverfahren bei DAWIS-M.D. könnte um eine Facetten- und Volltextsuche ergänzt werden. Auf diese Weise werden die Performance, die Filter- und Suchmöglichkeiten beim Suchverfahren deutlich verbessert. Davon würde der Wissenschaftler und seine Recherche profitierten, weil die Suche dann ergebnisorientiert ist. Dafür wurde bereits ein Prototyp implementiert, deren Grundidee und Softwareentwicklung in [FÏ12] vorgestellt wird.

   Damit die Netzwerkvisualisierung bei DAWIS-M.D. über mehr Funktionalitäten verfügt, sollten zusätzliche Layout-Algorithmen (z. B. Hierarchisches Layout und Force-Directed Layout) und eine Tiefen- und Breitensuche hinzugefügt werden. Dabei sollte auch die *Systems Biology Graphical Notation* (SBGN) [NHM+09, CKS10, vIVC+12] beachtet werden, weil diese grafische Notation eine immer größere Akzeptanz in der Systembiologie erhält.

3. Es sind bei TraBi derzeit genetische Informationen von drei eukaryotischen Organismen verfügbar. Damit eine Vorhersage von potentiellen TFBS in Nukleotidsequenzen auch für andere Organismen möglich ist, sollten zusätzliche genetische Informationen hinzugefügt werden, die wiederum zahlreiche eukaryotische und prokaryotische Organismen repräsentieren. Dieses kann durch die TraBi - Software-Infrastruktur ohne großen Aufwand erfolgen. Damit TraBi potenzielle TFBS nicht nur in den Nukleotidsequenzen der 5'-Upstream-Region und der 3'-Downstream-Region identifiziert, sollte eine genomweite Identifikation ebenfalls möglich sein.

   Die Gruppierung der TF sollte ebenfalls in TraBi möglich sein, wobei ein *Clustering* nach ähnlichen Sequenzmotiven oder Strukturbereichen (Helix-Turn-Helix-Motiv, Zinkfinger, Homöodomäne, Leucin-Zipper-Proteine oder Helix-Loop-Helix-Proteine) sinnvoll ist. Der Lösungsansatz, der in [SKWB13] thematisiert wird, könnte die Grundlage für eine solche Clusteranalyse sein.

   Die Ergebnismenge, die aus einer Vorhersage resultiert, wird bei TraBi in Tabellenform dargestellt. Diese Ergebnisse sollten grafisch in einem interaktiven *Genome browser* veranschaulicht werden. Dafür sind JBrowse [SUS+09, SH10] oder die *Genome browser* bei Ensembl und der University of California, Santa Cruz (UCSC) [KHK13] entsprechende Beispiele. Dadurch werden die einzelnen Ergebnisse auf die exakte Position im Genom aufgeteilt und Zusammenhänge besser deutlich.

   Die Variante des Algorithmus *ESAsearch*, die bei TraBi umgesetzt wurde, könnte um einen weiteren Ansatz ergänzt werden, sodass ein paarweises oder multiples Sequenzalignment der TFBS erfolgt. Dadurch könnten weitere falschpositive Ergebnisse in der Ergebnismenge beseitigt werden.

Das „Wiki-Prinzip“ ist in der Wissenschaftsgemeinde noch nicht weit verbreitet, weil Patentschutz und kommerzielle Aspekte häufig überwiegen. Dennoch sollten beide Softwarelösungen das „Wiki-Prinzip“ zukünftig berücksichtigen. Auf diese Weise kann der gegenseitige Informationsaustausch und der Dialog der unterschiedlichen Fachbereiche aus den Lebenswissenschaften problemlos erfolgen. Infolgedessen sind schnellere Fortschritte bzgl. einer Problemstellung möglich, weil alle Forschungseinrichtungen denselben Wissensstand besitzen und gemeinsam an einem potentiellen Lösungsansatz arbeiten können. Die Vor- und Nachteile des „Wiki-Prinzips“ und mögliche Lösungsansätze für die Lebenswissenschaften werden in [DTW+13, FGB12] erläutert.

Es sind zahlreiche wissenschaftliche Publikationen verfügbar, deren Informationen zeitverzögert oder gar nicht durch molekularbiologischen DB bereitgestellt werden. Daher ist eine Analyse der Publikationen mittels Text Mining durchaus sinnvoll, weil zusätzliche Informationen zur Verfügung stehen. Dadurch könnten bestehende Datenbestände vervollständigt und validiert werden. Allerdings ist die Algorithmik bei Softwarelösungen, die Text Mining einsetzten, sehr komplex und fehleranfällig. Deswegen liefert Text Mining im Kontext der Molekularbiologie häufig keine

eindeutigen oder nutzbaren Datenbestände. Dennoch wurde für DAWIS-M.D. ein Prototyp einer Softwarekomponente implementiert, deren Lösungsansatz auf Text Mining basiert. Diese Komponente und sein Ansatz werden in [Wit13] vorgestellt.

Danksagung

Als erstes möchte ich mich ganz herzlich bei Prof. Dr. Ralf Hofestädt und bei Prof. Dr. Thomas Dierks bedanken, die meine Dissertation unterstützt und betreut haben. Insbesondere für die Mitarbeit an einem interdisziplinären und vielseitigen Forschungsprojekt möchte ich mich herzlich bedanken. Ein besonderer Dank geht an Dr.-Ing. Benjamin Kormeier und Dr.-Ing. Thoralf Töpel, die jederzeit wichtige Ratschläge, konstruktive Kritik und motivierende Worte zum Gelingen der Dissertation beigetragen haben. Die biochemische Expertise von apl. Prof. Dr. Torben Lübke und seine hilfreichen und kritischen Korrekturen waren sehr wichtig für meine Dissertation, wofür ich mich ganz herzlich bedanken möchte. Meinen ehemaligen Studenten Dominic Gardner, Jan Füßmann, Marvin Meinold, Mirco Westermeyer und Pascal Witthus möchte ich für ihre Projekt- und Abschlussarbeiten danken, die ebenfalls einen gewissen Anteil zur Dissertation beigesteuert haben. Außerdem möchte ich allen Mitarbeiter/-innen der Arbeitsgruppe Bioinformatik und Medizinische Informatik als auch der Arbeitsgruppe Biochemie I für das motivierende und sehr angenehme Arbeitsumfeld danken.

Insbesondere meiner Mutter Ingeborg Hippe möchte ich für die Unterstützung und den Rückhalt während der Promotion und des Studiums danken sowie für den Zuspruch und Beistand in den Jahren davor. Meinem Bruder Thorsten Hippe möchte ich ebenfalls für die entscheidenden Ratschläge in den letzten Jahren und für das Korrekturlesen meiner Dissertation danken. Darüber hinaus danke ich meinem Vater Franz Hippe für seine Rücksicht und seinen Verzicht während der Promotion.

Mein letzter Dank richtet sich an Katrin Jergus, Vedrana Käfér, Andre Käfér, Carsten Hensiek, Christoph Benjamin, Dennis Nissen, Jameel Jaddou, Kamil Ertürk, Malte Timm und Mats Sören van Meerbeck sowie an alle Mannschaftskollegen der Basketballmannschaft „zwOTB“.

A WWW-Adressen

A.1 Molekularbiologische Datenbanken

BRENDA	http://www.brenda-enzymes.info/
EMBL-Bank	http://www.ebi.ac.uk/embl/
Ensembl	http://www.ensembl.org/
ENZYME	http://www.expasy.ch/enzyme/
EPD	http://epd.vital-it.ch/
GO	http://www.geneontology.org/
HPRD	http://www.hprd.org/
JASPAR	http://jaspar.genereg.net/
KEGG	http://www.genome.jp/kegg/
OMIM	http://www.ncbi.nlm.nih.gov/omim/
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop/
TRANSFAC®	http://www.biobase-international.com/ http://www.gene-regulation.com/index2.html
TRANSPATH®	http://www.biobase-international.com/ http://www.gene-regulation.com/index2.html
UniProt	http://www.uniprot.org/

A.2 Verwandte Arbeiten

BioWarehouse	http://biowarehouse.ai.sri.com/
CoryneRegNet	http://www.coryneregnet.de/
Match™	http://www.biobase-international.com/
MatInspector	http://www.genomatix.de/
ONDEX	http://www.ondex.org/
PiPa	http://www.informatik.hu-berlin.de/forschung/gebiete/wbi/resources/pipa/
SiTaR	https://sbi.hki-jena.de/sitar/
TESS	http://www.cbil.upenn.edu/cgi-bin/tess/

A.3 Softwarelösungen der Arbeitsgruppe Bioinformatik und Medizinische Informatik

BioDWH	http://agbi.techfak.uni-bielefeld.de/biodwh/ http://sourceforge.net/projects/biodwh/
CELLmicrocosmos	http://www.cellmicrocosmos.org/
DAWIS-M.D.	http://agbi.techfak.uni-bielefeld.de/DAWISMD/
RAMEDIS	https://www.imbio.de/stable/php/ramedis/htdocs/eng/index.php
TraBi	http://sourceforge.net/projects/auto-matrix/ http://auto-matrix.sourceforge.net/ http://agbi.techfak.uni-bielefeld.de/TraBi/
VANESA	http://sourceforge.net/projects/vanesa/ http://vanesa.sourceforge.net/

B Quelltext

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>TraBi - Software-Infrastruktur - Organism-File</comment>

  <entry key="3702">Arabidopsis thaliana</entry>
  <entry key="4530">Oryza sativa</entry>
  <entry key="4577">Zea mays</entry>
  <entry key="4932">Saccharomyces cerevisiae</entry>
  <entry key="7227">Drosophila melanogaster</entry>
  <entry key="7955">Danio rerio</entry>
  <entry key="9031">Gallus gallus</entry>
  <entry key="9606">Homo sapiens</entry>
  <entry key="9913">Bos taurus</entry>
  <entry key="9940">Ovis aries</entry>
  <entry key="10090">Mus musculus</entry>
  <entry key="10116">Rattus norvegicus</entry>
  <entry key="10117">Rattus rattus</entry>
</properties>
```

Quelltext B.1: Beispiel für eine Textdatei zur Identifikation der Organismen bei der TraBi - Software-Infrastruktur.

```
>ENSG00000137573|SULF1|sulfatase 1 [Source:HGNC Symbol;Acc:20391]|70378859|70573150|8|1
TCAAGATGAACCTTTTATCTCCTATGGAAGGGAAAACGTCTTGCTGAGATGGCCCTTCCA
TAGACGGGCAGTAAAAATGTTTAGGGTTTCTGTGCATCCCACTGAGGGTCCCCATCATCT
CAGGGGGAGCACGAGGAGTTGTGAAAAGTAACGCACAACTGGTCATCAGAATAATTCATT
CCCTCCACCTTCTCTGTAGCACCTCCCACTTCTCCTCTTTTTATAGCAGTCCTTTCTCTC
TGAAAATCTC

>ENSG00000196562|SULF2|sulfatase 2 [Source:HGNC Symbol;Acc:20392]|46285092|46415360|20|-1
AATAAGTTACTGCTGTTTACAAGTGCTTTCCCACAGCGGCGTCTGTTTCCCCTTAGCTAG
CAACTCGGCTGTGTTTTCTGCAGCTGCTGGTGAGTTCTCTGCCCGCTCTTTGCCCACCCG
CGTCAGGCCGGTCCCCCTCCGGCCTCTTCTGTGGCGCGAGGGACAGCGGAAACCACGGTA
GACAGCACCCCCTTGAGTCCAATTCCTCCCCTTTCGGAGGAAGCCGGTTTCTCCTTTCTA
TGCTACTCCC
```

Quelltext B.2: Beispiel für eine Textdatei im FASTA-Format.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Configuration file for the DAWIS-M.D. web application.</comment>

  <entry key="logfiles_dir">logfiles/</entry>
  <entry key="tmp_dir">tmp/</entry>
  <entry key="webserver">agbi</entry>
  <entry key="http_port">8080</entry>
  <entry key="dawismd_datasource">java:/comp/env/jdbc/dawismd</entry>
  <entry key="metadata_datasource">java:/comp/env/jdbc/metadata</entry>
  <entry key="project">DAWISMD</entry>
  <entry key="email_host">smarthost.TechFak.Uni-Bielefeld.DE</entry>
  <entry key="email_address">juser@TechFak.Uni-Bielefeld.DE</entry>
</properties>
```

Quelltext B.3: Beispiel für eine Konfigurationsdatei bei DAWIS-M.D.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Configuration file for the TraBi - Software-Infrastructure.</comment>

  <entry key="database_host">agbi</entry>
  <entry key="database_port">3306</entry>
  <entry key="database">dawismd</entry>
  <entry key="database_manufacturer">org.hibernate.dialect.MySQL5InnoDBDialect</entry>
  <entry key="database_username">anonymous</entry>
  <entry key="database_password">secret</entry>
  <entry key="proxy_http">www-proxy.TechFak.Uni-Bielefeld.DE</entry>
  <entry key="proxy_port">80</entry>
</properties>
```

Quelltext B.4: Beispiel für eine Konfigurationsdatei bei der TraBi - Software-Infrastruktur.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Configuration file for the TraBi - web application.</comment>

  <entry key="dbUser">anonymous</entry>
  <entry key="dbPassword">secret</entry>
  <entry key="dbServer">agbi</entry>
  <entry key="dbPort">3306</entry>
  <entry key="database">metadata</entry>
  <entry key="datasource">java:/comp/env/jdbc/metadata</entry>
  <entry key="rmiLookupName">RMIManager</entry>
  <entry key="rmiHost">127.0.1.1</entry>
  <entry key="rmiPort">1099</entry>
  <entry key="userSuffixArrayDirectory">user/</entry>
  <entry key="suffixArrayDirectory">organism/</entry>
  <entry key="arraysLocation">/data/webapps/EnhancedSuffixArray/</entry>
  <entry key="emailHost">smarthost.TechFak.Uni-Bielefeld.DE</entry>
  <entry key="emailAddress">juser@TechFak.Uni-Bielefeld.DE</entry>
  <entry key="threadpoolSize">8</entry>
  <entry key="minCoreLength">3</entry>
</properties>
```

Quelltext B.5: Beispiel für eine Konfigurationsdatei bei der TraBi - Webanwendung.

C Zusätzliche Abbildungen der Softwarelösungen

C.1 DAWIS-M.D.

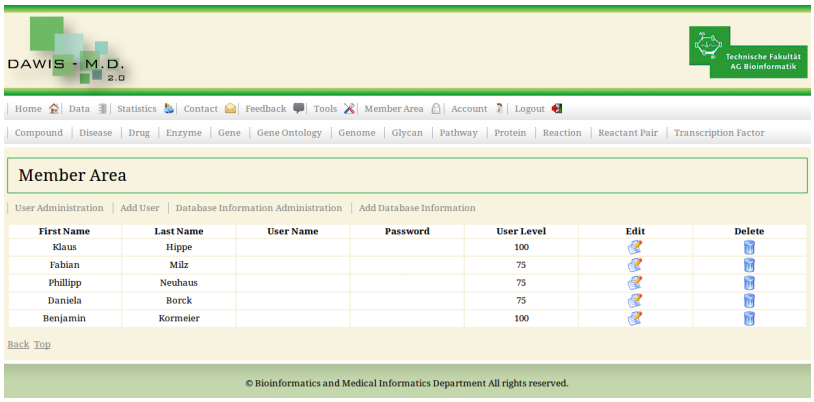


Abbildung C.1: Webseite für die System- und Benutzerverwaltung bei DAWIS-M.D.

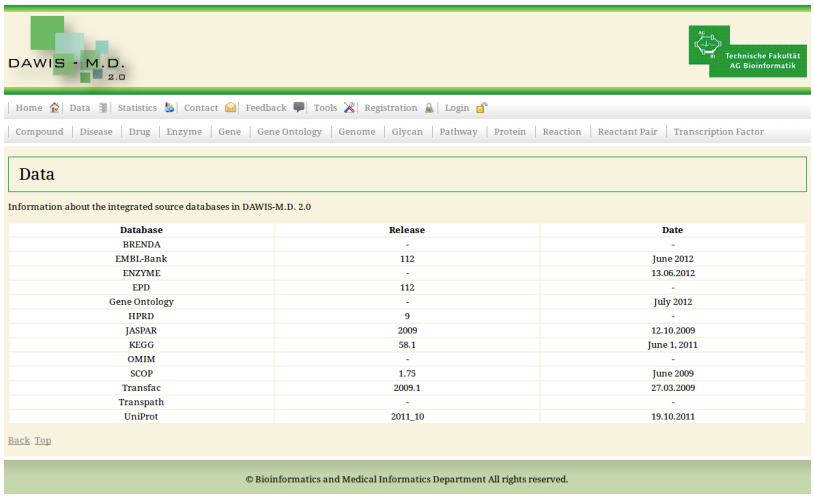


Abbildung C.2: Webseite bei DAWIS-M.D., die Informationen der molekularbiologischen DB darstellt.

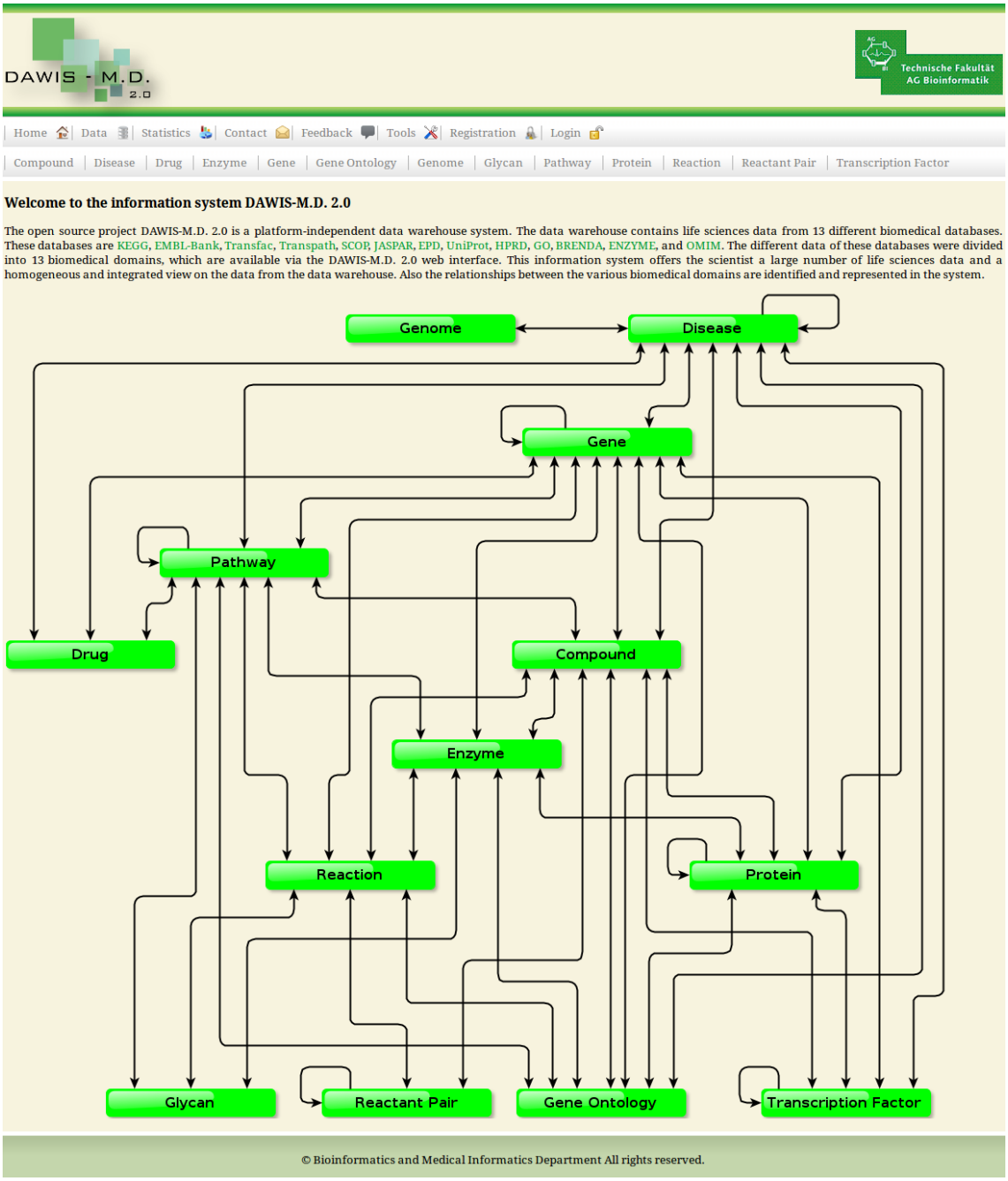
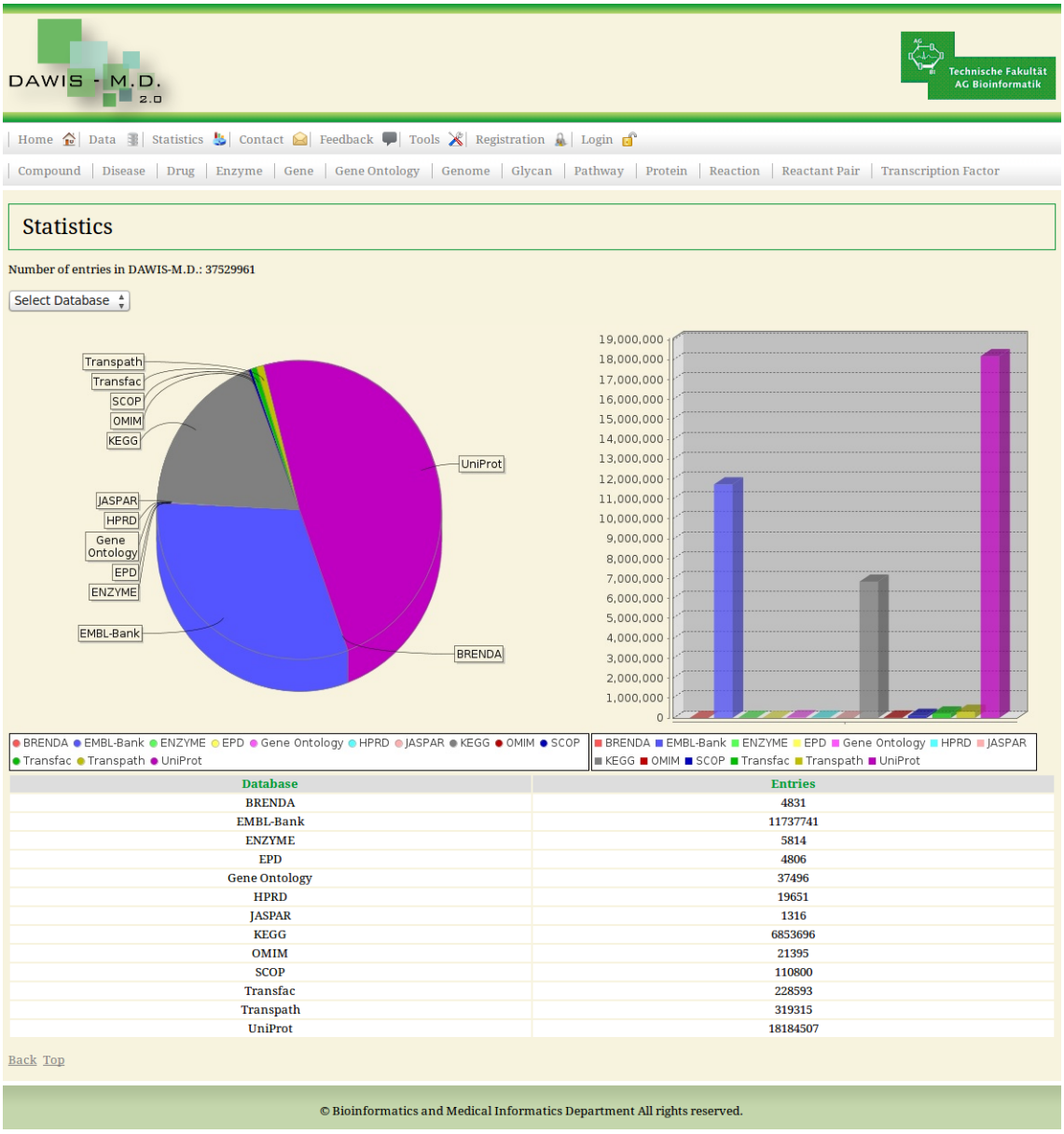


Abbildung C.3: Startseite bei DAWIS-M.D.



Database	Entries
BRENDA	4831
EMBL-Bank	11737741
ENZYME	5814
EPD	4806
Gene Ontology	37496
HPRD	19651
JASPAR	1316
KEGG	6853696
OMIM	21395
SCOP	110800
Transfac	228593
Transpath	319315
UniProt	18184507

Abbildung C.4: Webseite für die Statistik bei DAWIS-M.D.

C.2 TraBi - Webanwendung

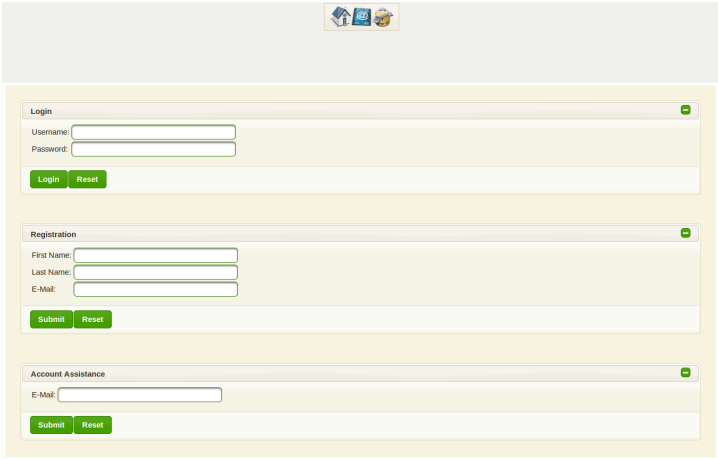


Abbildung C.5: Webseite zur Benutzeranmeldung, Registrierung und Benutzerdaten anfordern bei der TraBi - Webanwendung.

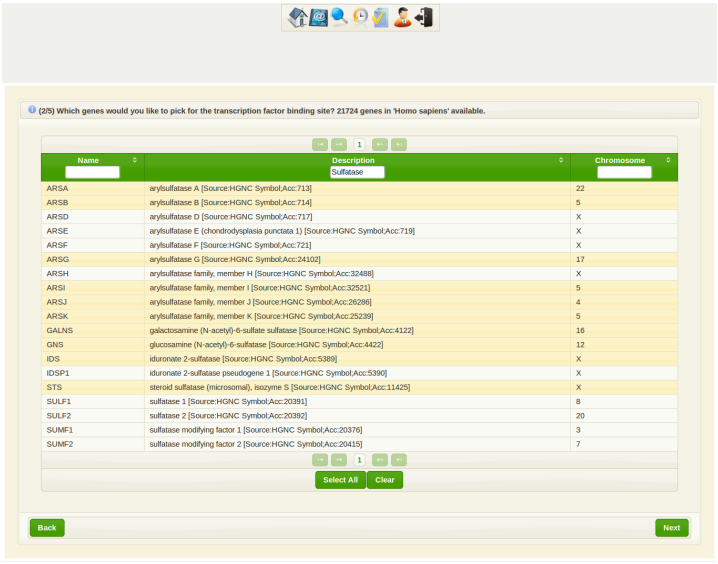


Abbildung C.6: Oberfläche zur Zusammenstellung der Gene beim Konfigurationsassistent der TraBi - Webanwendung.

Abbildung C.7: Oberfläche zur Zusammenstellung der TF beim Konfigurationsassistent der TraBi - Webanwendung.

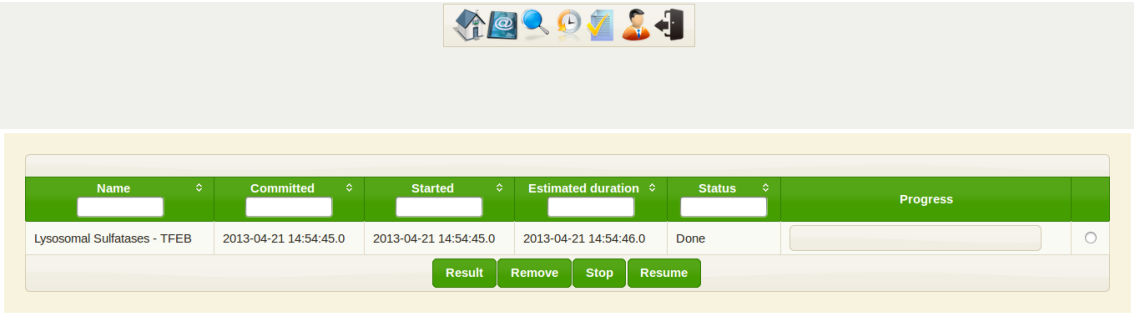


Abbildung C.8: Webseite zur Verwaltung der Vorhersagen bei der TraBi - Webanwendung.

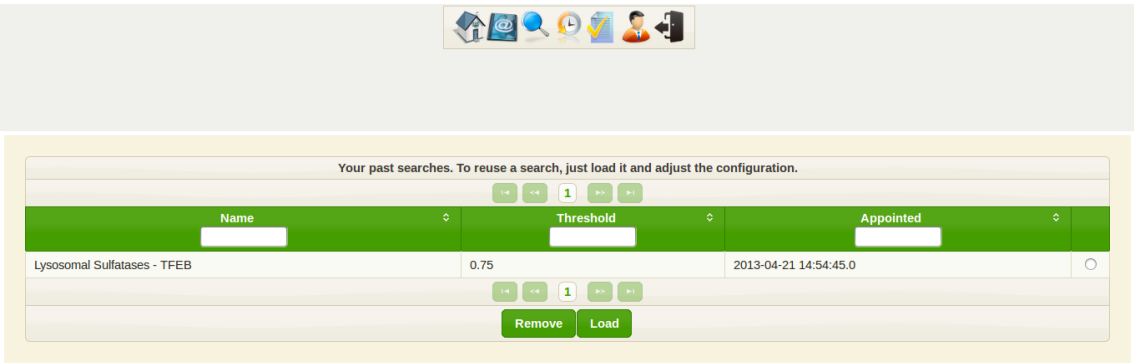


Abbildung C.9: Webseite zur Verwaltung der Suchprofile bei der TraBi - Webanwendung.

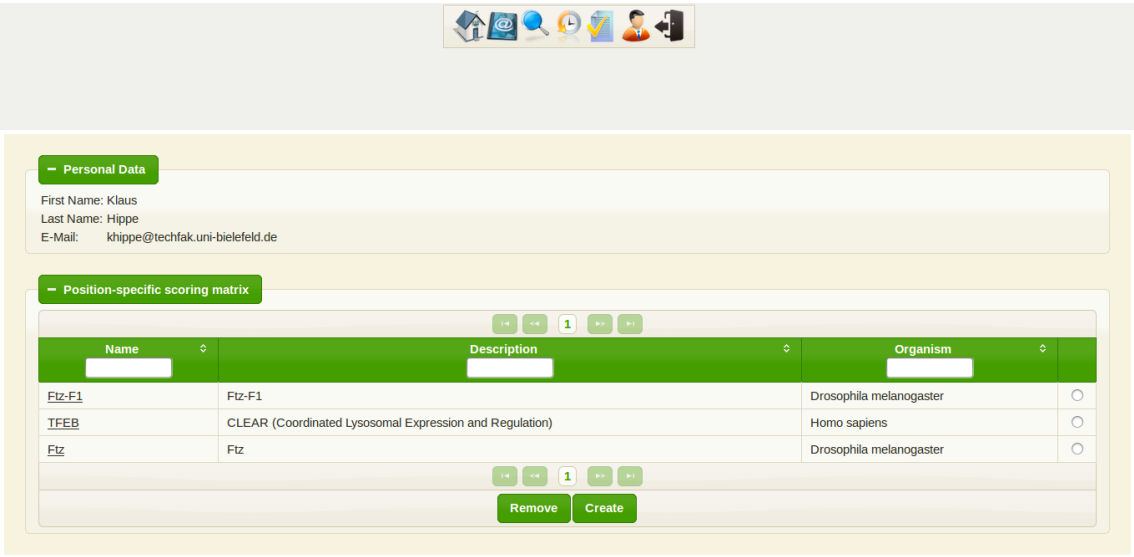
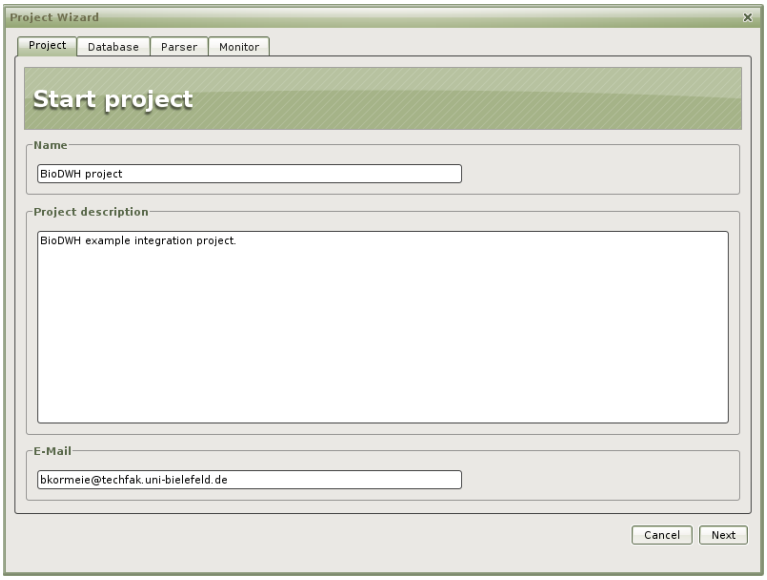
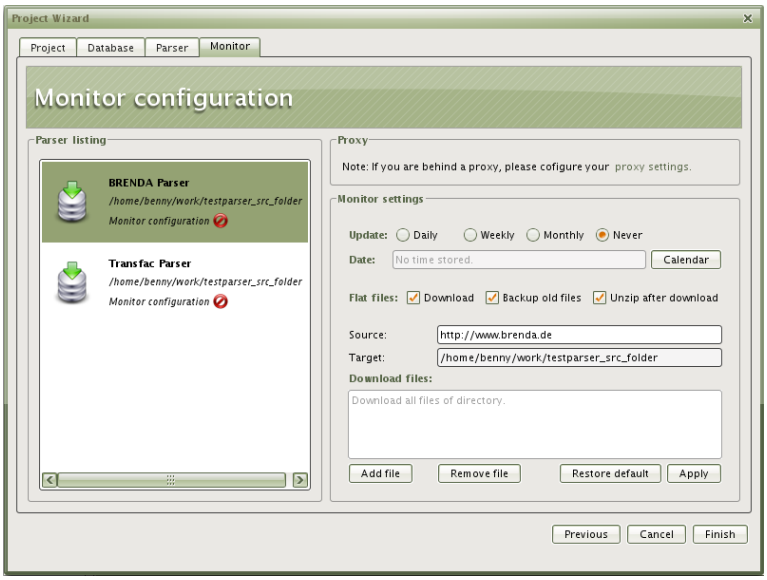


Abbildung C.10: Webseite zur Verwaltung der benutzerspezifischen PSSM bei der TraBi - Webanwendung.

C.3 BioDWH



(a) Eingabe der Projektbeschreibung.



(b) Konfiguration des Monitors.

Abbildung C.11: Konfigurationsassistent bei BioDWH.

D Signalwege

D.1 Wnt-Signalweg

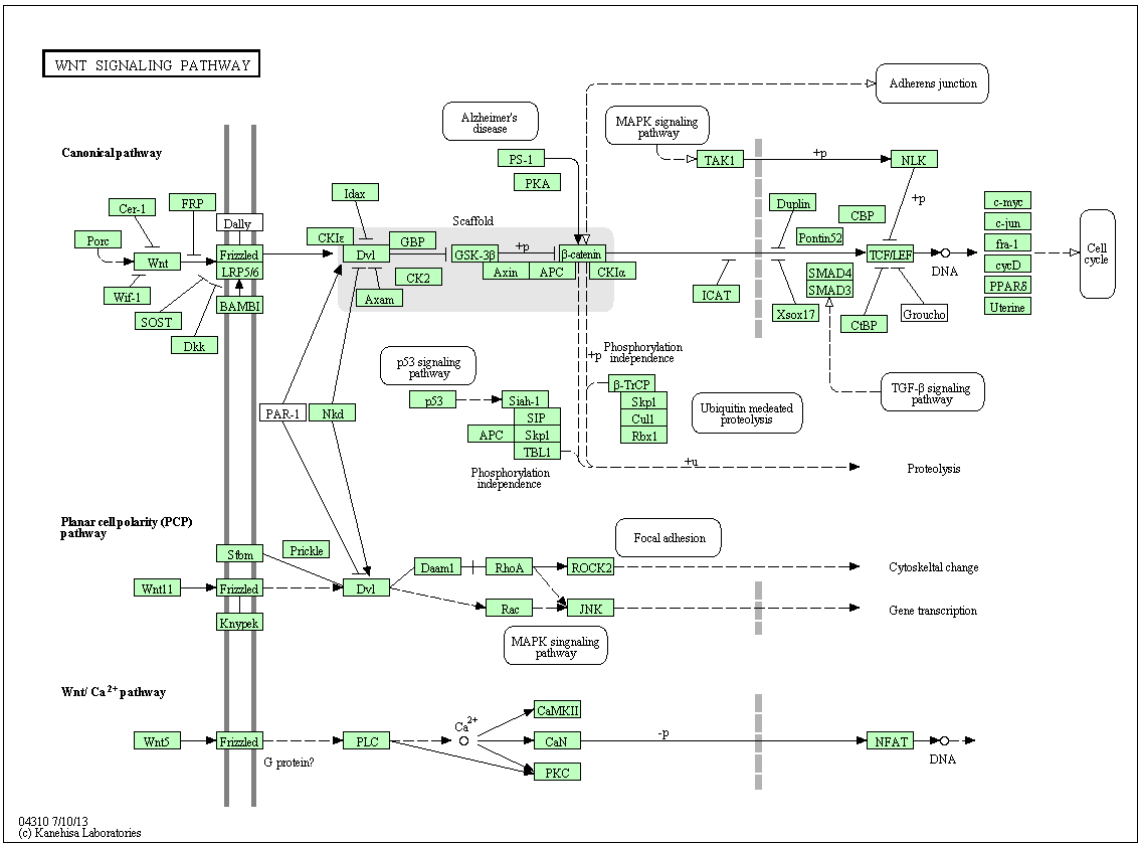


Abbildung D.1: Wnt-Signalweg für *Homo sapiens* [KGS+12].

D.2 Hedgehog-Signalweg

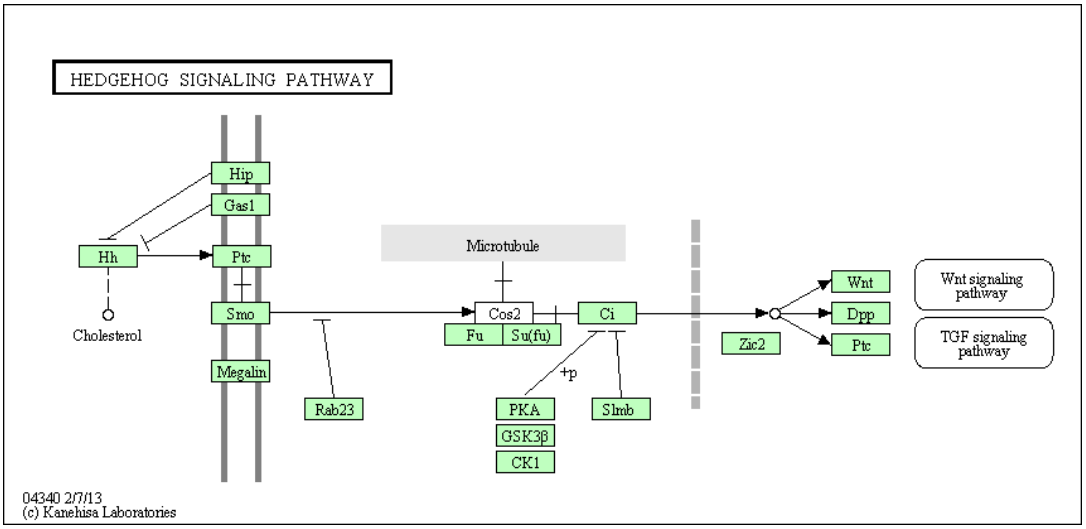


Abbildung D.2: Hedgehog-Signalweg für *Homo sapiens* [KGS+12].

D.3 JAK-STAT-Signalweg

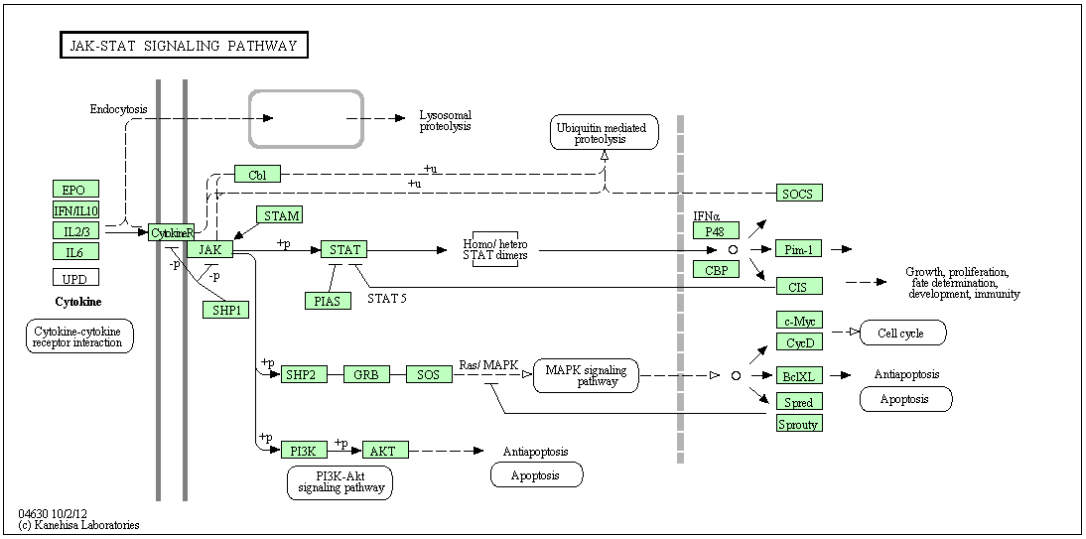


Abbildung D.3: JAK-STAT-Signalweg für *Homo sapiens* [KGS+12].

D.4 TGF-β-Signalweg

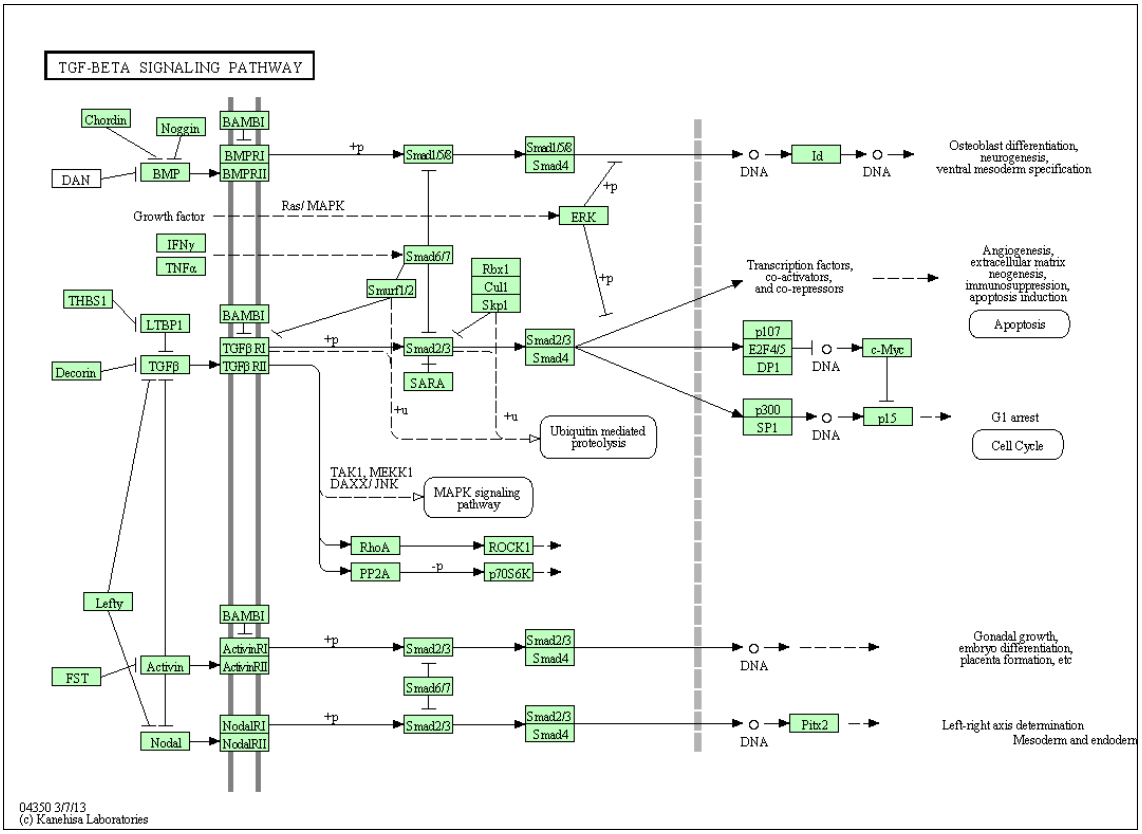


Abbildung D.4: TGF-β-Signalweg für *Homo sapiens* [KGS+12].

E Ergebnisse der Laborexperimente

Zelllinie	Transfektion	Gelelektrophorese
HT1080	Transient	siehe Abbildung E.2
HT1080-2		siehe Abbildung E.3
HeLa		
HeLa-hygro	Stabil	siehe Abbildung E.4
HeLa-hygro-2		-
HeLa-neo		siehe Abbildung E.3
HeLa-neo-2		-

Tabelle E.1: Bezeichnung der eukaryotischen Zelllinien (HeLa und HT1080), Methoden der Transfektion (Stabil oder Transient) und der eingeführte Vektor bzw. Plasmid nach [Gar13].

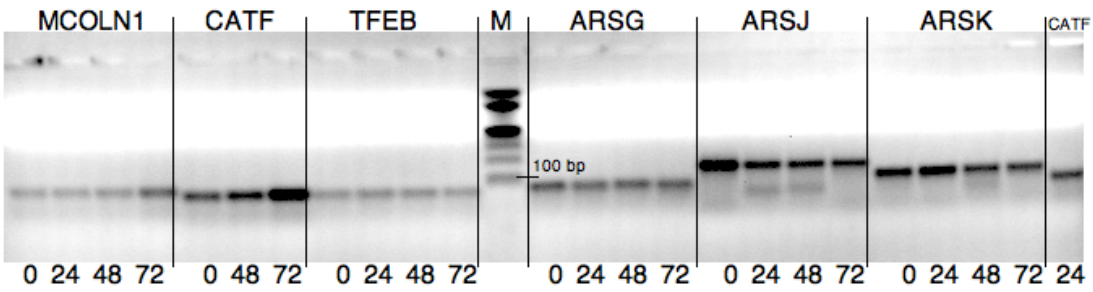


Abbildung E.1: Ergebnisse der qRT-PCR der Induktion von Saccharose nach 0 - 72 Stunden für den WT der HeLa-Zellen nach [Gar13].

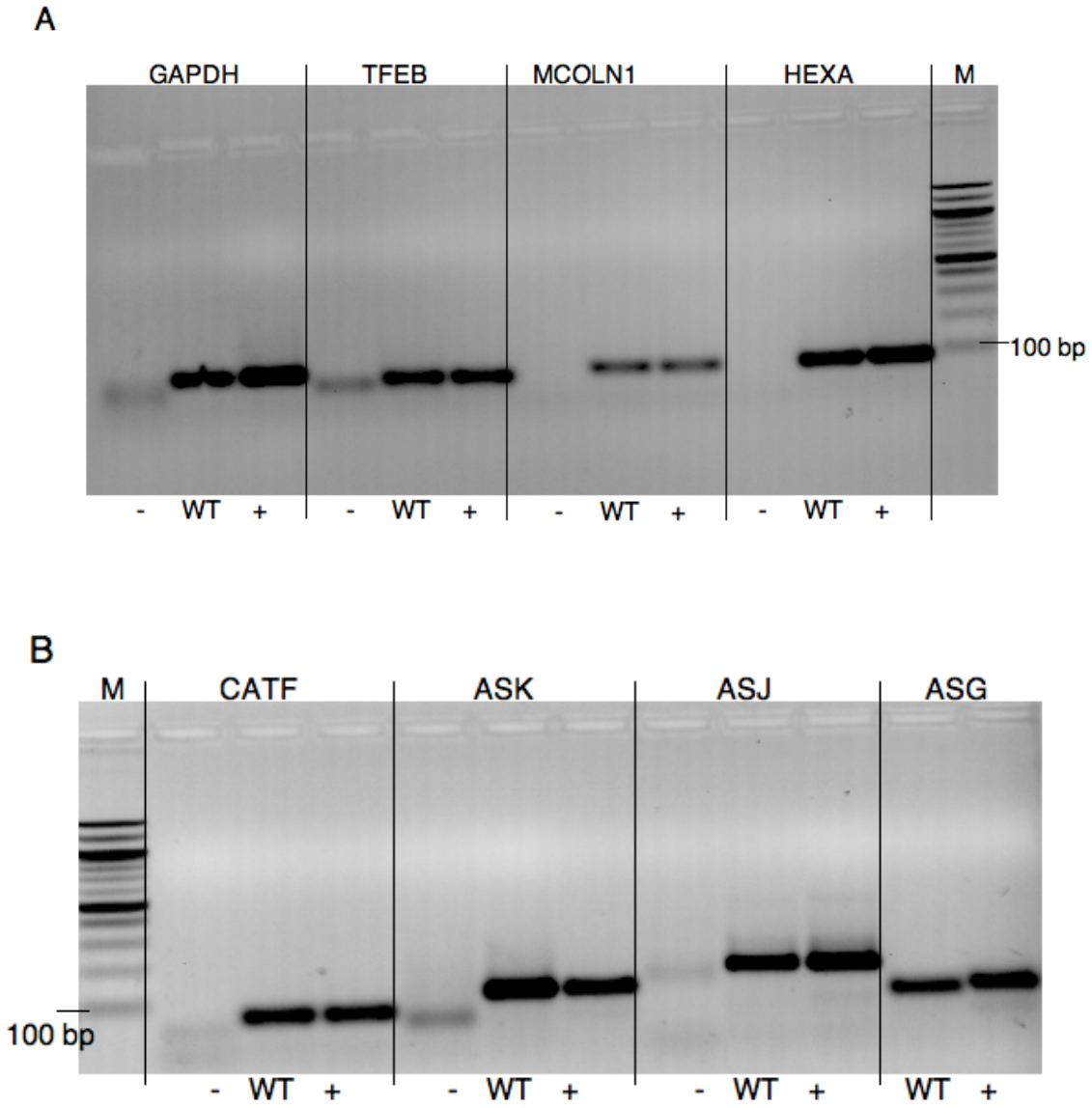


Abbildung E.2: Ergebnisse der qRT-PCR der Negativkontrolle des WT und der mit TFEB transfizierten Zellen von HT1080 nach [Gar13]. Dabei symbolisiert das Minuszeichen (-) die Negativkontrolle des WT und das Pluszeichen (+) die mit TFEB transfizierten HT1080-Zellen.

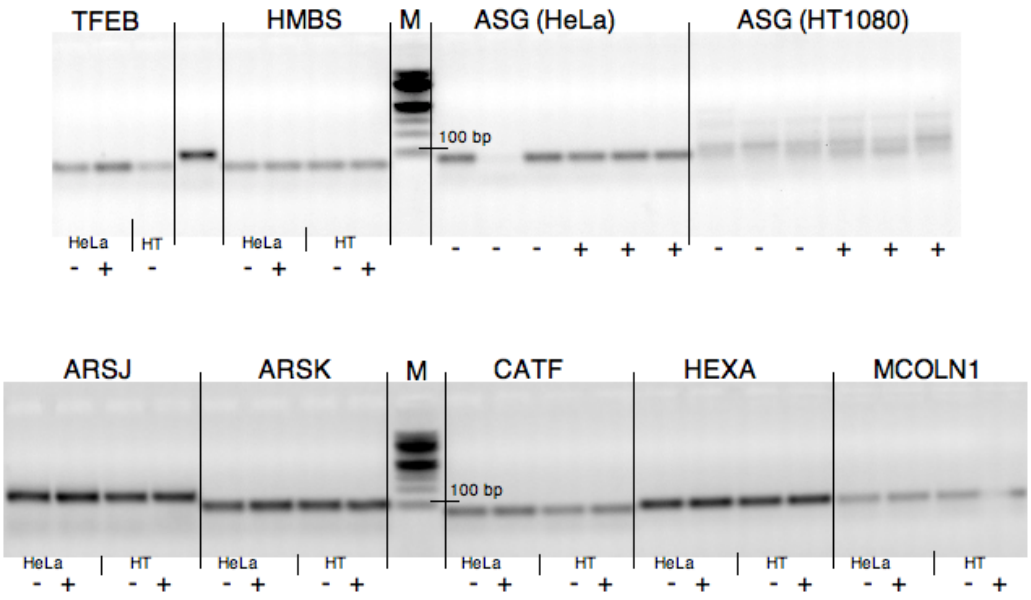


Abbildung E.3: Ergebnisse der qRT-PCR des WT und der mit TFEB transfizierten eukaryotischen Zelllinien, wobei HeLa und HT1080 eingesetzt wurden. Dabei symbolisiert das Minuszeichen (-) den WT und das Pluszeichen (+) die mit TFEB transfizierten Zellen von HeLa und HT1080-2 nach [Gar13].

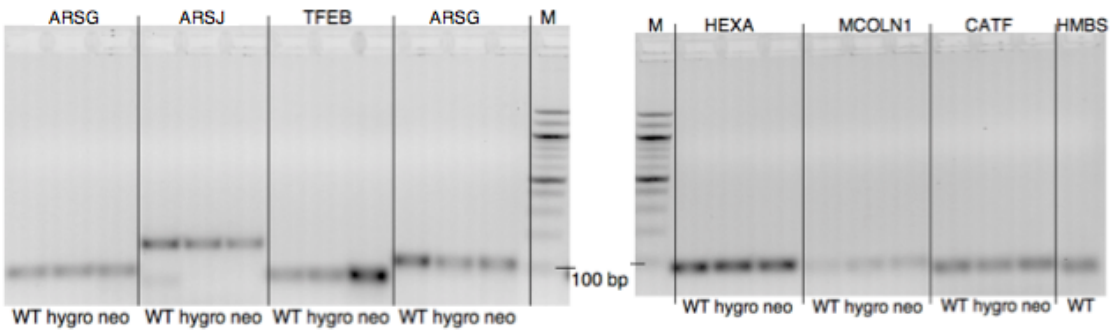
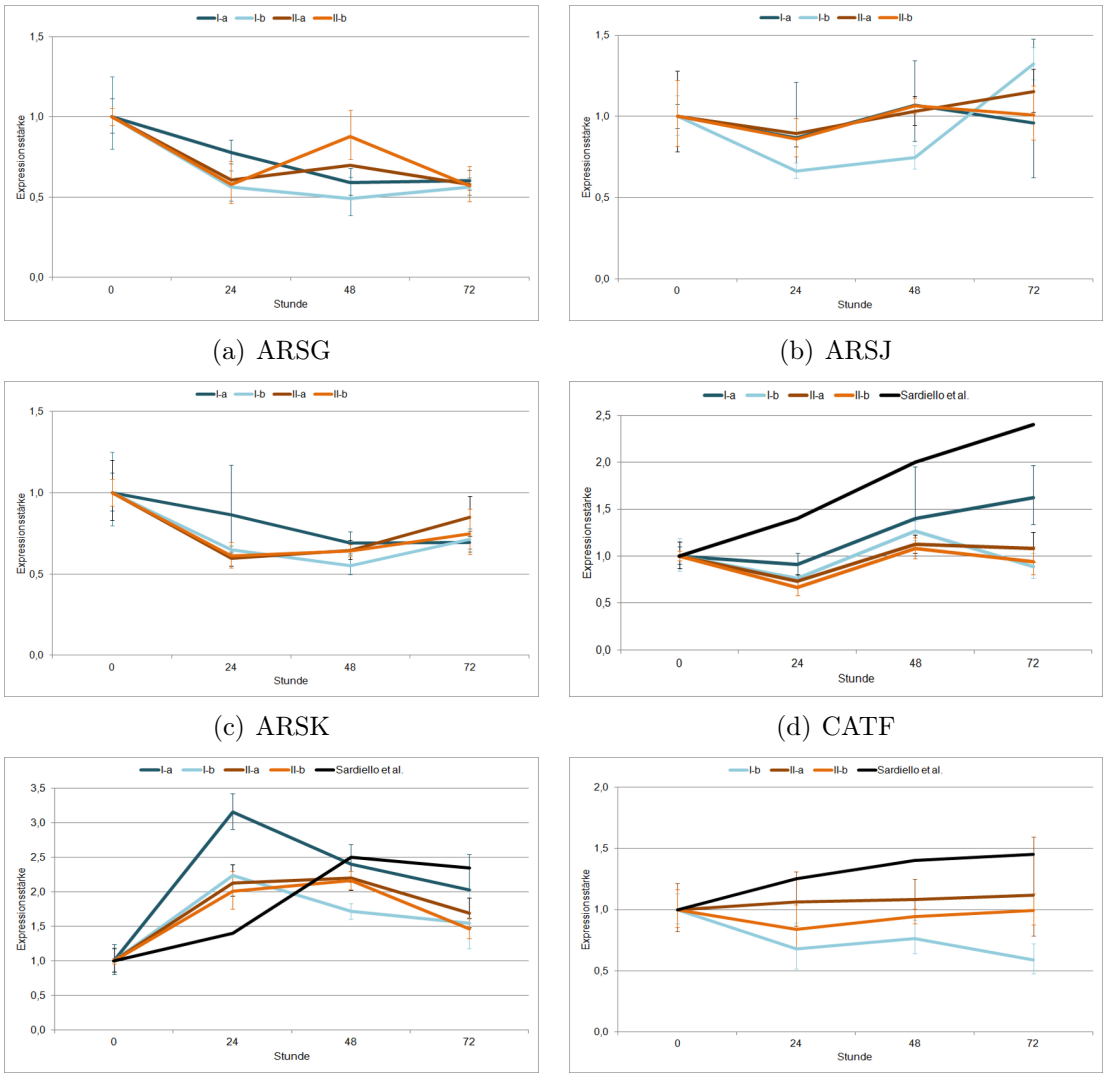


Abbildung E.4: Ergebnisse der qRT-PCR des WT und der mit TFEB transfizierten HeLa-Zellen (neo und hygro) nach [Gar13].

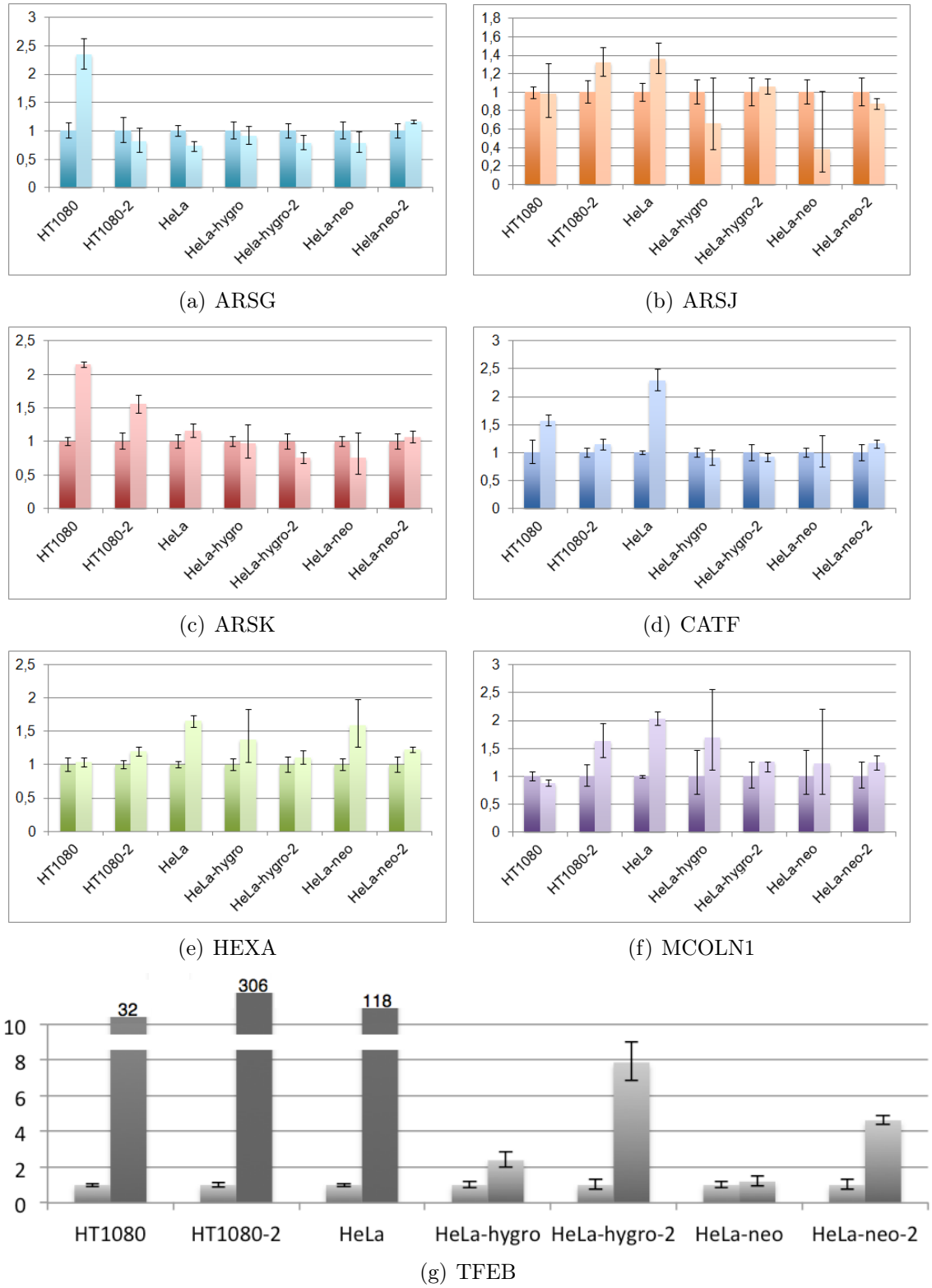


(a) ARSG (b) ARSJ

(c) ARSK (d) CATF

(e) MCOLN1 (f) TFEB

Abbildung E.5: Auswirkung der Induktion von Saccharose auf die Expressionsstärke der humanen Gene, wobei Cathepsin F, Mucolipin-1, Transcription factor EB und die Arylsulfatase G, I und K untersucht wurden nach [Gar13]. Dabei sind die Zeitreihen I und II, deren Replikate a und b als auch die Ergebnisse aus [SPdR+09] dargestellt.



(a) ARSG (b) ARSJ

(c) ARSK (d) CATF

(e) HEXA (f) MCOLN1

(g) TFEB

Abbildung E.6: Expressionsstärke der humanen Gene der transfizierten Zelllinie im Vergleich zum WT der Zelllinie, wobei Cathepsin F, Mucolipin-1, Transcription factor EB, Hexosaminidase A und die Arylsulfatase G, I und K untersucht wurden nach [Gar13]. Die Ergebnisse der transfizierten Zelllinie sind rechts dargestellt, wohingegen die Ergebnisse des WT der Zelllinie links abgebildet sind.

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ia								
0	28,9978	0,0864	2,1777	0,0556	0.0000	1,0000	0,1015	0,1129
24	28,9574	0,0473	2,5394	0,0496	0,3618	0,7782	0,0709	0,0780
48	28,8320	0,1161	2,9390	0,0742	0,5899	0,5899	0,0785	0,0906
72	29,2646	0,0432	2,9059	0,0523	0,7282	0,6036	0,0577	0,0640
Zeitintervall Ib								
0	29,3147	0,1755	5,0043	0,1161	0,0000	1,0000	0,2003	0,2505
24	29,3589	0,1433	5,8323	0,0847	0,8280	0,5633	0,0847	0,0997
48	29,3536	0,2108	6,0322	0,1262	1,0279	0,4904	0,1058	0,1349
72	29,4500	0,0733	5,8305	0,0499	0,8262	0,5640	0,0516	0,0569
Zeitintervall IIa								
0	29,1245	0,1690	0,7523	0,1165	0,0000	1,0000	0,2008	0,2513
24	29,4162	0,1193	1,4720	0,0798	0,7198	0,6072	0,0864	0,1007
48	28,6027	0,0348	1,2726	0,0390	0,5204	0,6972	0,0505	0,0544
72	30,8406	0,3905	1,5419	0,2324	0,7897	0,5785	0,2087	0,3263
Zeitintervall IIb								
0	30,3821	0,0194	2,4447	0,0246	0,0000	1,0000	0,0527	0,0557
24	30,5945	0,1669	3,2343	0,1165	0,7896	0,5785	0,1162	0,1454
48	29,4706	0,1072	2,6336	0,0788	0,1889	0,8773	0,1400	0,1665
72	31,8730	0,1542	3,2498	0,1001	0,8051	0,5723	0,1002	0,1216

Tabelle E.2: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von ARSG nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ia								
0	25,7444	0,0510	-1,0758	0,0383	0,0000	1,0000	0,0711	0,0765
24	25,5469	0,2928	-0,8711	0,1741	0,2047	0,8677	0,2470	0,3453
48	24,7219	0,2004	-1,7110	0,1200	-0,0953	1,0683	0,2203	0,2775
72	26,4374	-	-	-	-	0,9602	0,3357	0,5160
Zeitintervall Ib								
0	27,4128	0,0484	3,1025	0,0632	0,0000	1,0000	0,1146	0,1294
24	27,2155	0,0580	3,6689	0,0381	0,5864	0,6660	0,0471	0,0507
48	2608448	0,0635	3,5235	0,0497	0,4209	0,7469	0,0681	0,0750
72	26,3178	0,0484	2,6984	0,0384	-0,4041	1,3230	0,0940	0,1018
Zeitintervall IIa								
0	26,0070	0,1919	-2,3653	0,1278	0,0000	1,0000	0,2180	0,2787
24	25,7360	0,0514	-2,2082	0,0500	0,1571	0,8968	0,0823	0,0907
48	24,9177	0,0522	-2,4124	0,0450	-0,0471	1,0332	0,0858	0,0936
72	26,7284	0,0325	-2,5702	0,0596	-0,2049	1,1526	0,1249	0,1401
Zeitintervall IIb								
0	26,6600	0,1796	-1,2774	0,1057	0,0000	1,0000	0,1840	0,2255
24	26,2960	0,0474	-1,0641	0,0710	0,2133	0,8626	0,1102	0,1263
48	25,4684	0,0129	-1,3686	0,0228	-0,0912	1,0653	0,0458	0,0478
72	27,3341	0,1265	-1,2891	0,0862	-0,0117	1,0081	0,1540	0,1819

Tabelle E.3: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von ARSJ nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ia								
0	26,8630	0,0953	0,0429	0,0603	0,0000	1,0000	0,1095	0,1229
24	26,6722	0,2646	0,2542	0,1583	0,2113	0,8637	0,2268	0,3076
48	26,4701	0,0686	0,5771	0,0509	0,5432	0,6905	0,0643	0,0711
72	26,9232	0,0262	0,5644	0,0484	0,5216	0,6966	0,0619	0,0680
Zeitintervall Ib								
0	27,1199	0,1770	2,8096	0,1169	0,0000	1,0000	0,2014	0,2522
24	26,9555	0,0012	3,4289	0,0181	0,6193	0,6510	0,0223	0,0231
48	26,9889	0,0774	3,6675	0,0558	0,8579	0,5518	0,0563	0,0626
72	26,9138	0,0619	3,2943	0,0444	0,4847	0,7146	0,0585	0,0638
Zeitintervall IIa								
0	27,2710	0,1222	-1,1013	0,0950	0,0000	1,0000	0,1672	0,2007
24	27,5848	0,0329	-0,3594	0,0445	0,7419	0,5980	0,0491	0,0535
48	26,8566	0,0592	-0,4735	0,0478	0,6278	0,6472	0,0570	0,0597
72	28,4354	0,0827	-0,8632	0,0740	0,2380	0,8479	0,1126	0,1298
Zeitintervall IIb								
0	26,7687	0,0447	-1,1687	0,0376	0,0000	1,0000	0,0796	0,0865
24	26,8961	0,0264	-0,4641	0,0673	0,7046	0,6136	0,0745	0,0848
48	26,3043	0,0380	-0,5327	0,0307	0,6359	0,6435	0,0369	0,0392
72	27,8741	0,1463	-0,7492	0,0961	0,4195	0,7477	0,1262	0,1518

Tabelle E.4: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von ARSK nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ia								
0	26,0096	0,0701	-0,8106	0,0479	0,0000	1,0000	0,0880	0,0965
24	25,7413	0,0866	-0,6767	0,0649	0,1339	0,9114	0,1071	0,1213
48	24,5983	0,2951	-1,2947	0,1733	-0,4841	1,3987	0,3967	0,5538
72	24,8514	0,1541	-1,5074	0,1002	-0,6968	1,6209	0,2842	0,3446
Zeitintervall Ib								
0	28,2212	0,1188	3,9108	0,0890	0,0000	1,0000	0,1575	0,1869
24	27,8241	0,0147	4,2975	0,0200	0,3867	0,7649	0,0289	0,0300
48	26,8919	0,0449	3,5705	0,0423	-0,3403	1,2660	0,0990	0,1075
72	27,6964	0,1270	4,0770	0,0779	0,1661	0,8912	0,1241	0,1433
Zeitintervall IIa								
0	26,5298	0,0671	-1,8425	0,0745	0,0000	1,0000	0,1336	0,1542
24	26,5444	0,0423	-1,3999	0,0471	0,4427	0,7358	0,0638	0,0698
48	25,3201	0,0498	-2,0100	0,0441	-0,1675	1,1231	0,0915	0,0996
72	27,3395	0,0494	-1,9591	0,0665	-0,1166	1,0842	0,1478	0,1712
Zeitintervall IIb								
0	26,9387	0,0293	-0,9987	0,0265	0,0000	1,0000	0,0496	0,0522
24	26,9475	0,0496	-0,4127	0,0715	0,5860	0,6662	0,0857	0,0983
48	25,7257	0,0856	-1,1113	0,0539	-0,1126	1,0812	0,1066	0,1182
72	27,1350	0,1137	-0,9097	0,0800	0,0890	0,9402	0,1341	0,1564

Tabelle E.5: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von CATF nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ia								
0	25,1072	0,1908	-1,7130	0,1129	0,0000	1,0000	0,1953	0,2426
24	23,0481	0,0151	-3,3699	0,0424	-1,6569	3,1533	0,2469	0,2678
48	22,9163	0,0852	-2,9767	0,0586	-1,2637	2,4011	0,2562	0,2867
72	23,6272	0,1877	-2,7315	0,1777	-1,0185	2,0259	0,4108	0,5152
Zeitintervall Ib								
0	27,7344	0,1031	3,4241	0,0822	0,0000	1,0000	0,1464	0,1715
24	25,7868	0,0457	2,2602	0,0320	-1,1639	2,2406	0,1338	0,1423
48	25,9658	0,0121	2,6445	0,0342	-0,7796	1,7167	0,1094	0,1168
72	26,4146	0,2429	2,7952	0,1427	-0,6289	1,5464	0,3714	0,4888
Zeitintervall IIa								
0	28,4468	0,1056	0,0745	0,0881	0,0000	1,0000	0,1560	0,1849
24	26,9283	0,0474	-1,0160	0,0448	-1,1380	2,1295	0,1904	0,2693
48	26,2666	0,0515	-1,0635	0,0448	-1,1380	2,2008	0,1817	0,1980
72	28,6177	0,0591	-0,6810	0,0660	-0,7555	1,6882	0,2015	0,2288
Zeitintervall IIb								
0	28,4822	0,0027	0,5448	0,0205	0,0000	1,0000	0,0441	0,0462
24	26,9014	0,0416	-0,4587	0,0698	-1,0036	2,0050	0,2520	0,2882
48	26,2705	0,0412	-0,5664	0,0321	-1,1113	2,1604	0,1294	0,1376
72	28,6157	0,0470	-0,0075	0,0531	-0,5523	1,4664	0,1425	0,1579

Tabelle E.6: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von MCOLN1 nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

	C_T-Mean	C_T-Dev	ΔC_T-Mean	ΔC_T-Err	$\Delta\Delta C_T$	Expressionsstärke	d-	d+
Zeitintervall Ib								
0	29,9346	0,0530	5,6242	0,0645	0,0000	1,0000	0,1167	0,1321
24	29,7089	0,2431	6,1823	0,1415	0,5581	0,6792	0,1619	0,2126
48	29,3318	0,1493	6,0104	0,0925	0,3862	0,7651	0,1247	0,1491
72	30,0133	0,1789	6,3938	0,1066	0,7695	0,5866	0,1088	0,1335
Zeitintervall IIa								
0	29,8844	0,1370	1,5121	0,1015	0,0000	1,0000	0,1775	0,2158
24	29,3673	0,1721	1,4231	0,1072	-0,0891	1,0637	0,1984	0,2438
48	28,7248	0,1139	1,3947	0,0738	-0,1174	1,0848	0,1436	0,1656
72	30,6499	0,3039	1,3513	0,1844	-0,1609	1,1179	0,3339	0,4762
Zeitintervall IIb								
0	28,8973	0,0947	0,9599	0,0700	0,0000	1,0000	0,1431	0,1669
24	28,5756	0,1017	1,2155	0,0973	0,2556	0,8376	0,1618	0,2006
48	27,8837	0,0432	1,0467	0,0330	0,0868	0,9416	0,0579	0,0616
72	29,5936	0,0834	0,9704	0,0664	0,0105	0,9927	0,1190	0,1353

Tabelle E.7: C_T-Werte (Standardabweichung und Mittelwert), ΔC_T-Werte (Mittelwert und Fehler), $\Delta\Delta C_T$-Werte und Expressionsstärke (negative und positive Fehler) der Ergebnisse der qRT-PCR von TFEB nach der Induktion von Saccharose in den WT der HeLa-Zellen nach [Gar13].

F Notationselemente der UML

F.1 Anwendungsfalldiagramm

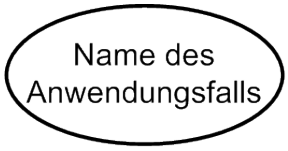
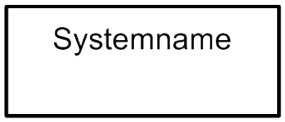
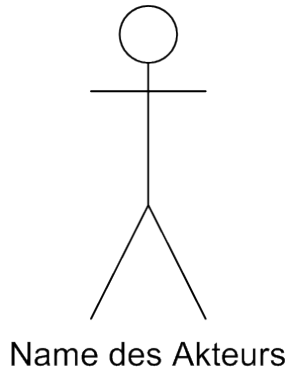
Name	Notation	Erklärung
Anwendungsfall		Beschreibt ein Verhalten des Systems, das einem Benutzer bereitgestellt wird.
System		Ist die Einheit, die das Verhalten, das durch Anwendungsfälle beschrieben wird, realisiert und anbietet.
Akteur		Interagiert mit den Anwendungsfällen des Systems und steht immer außerhalb davon.

Tabelle F.1: Notationselemente für ein Anwendungsfalldiagramm nach [RQdS12].

F.2 Aktivitätsdiagramm

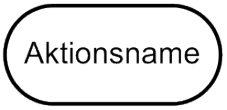
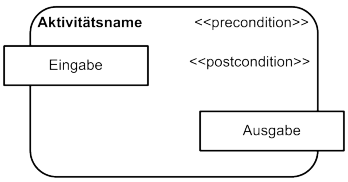
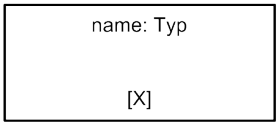
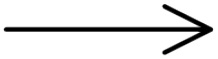
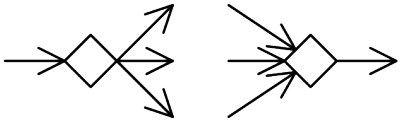
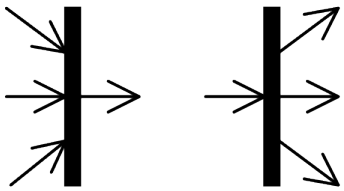
Name	Notation	Erklärung
Aktion		Steht für den Aufruf eines Verhaltens oder die Abarbeitung einer Funktion, die nicht weiter zerlegt wird.
Aktivität		Bezeichnet die gesamte Einheit, die in einem Aktivitätsdiagramm modelliert wird, und kann Ein- und Ausgabeparameter besitzen.
Objektknoten		Repräsentiert eine Ausprägung eines bestimmten Typs in einem bestimmten Zustand.
Kante		Ist ein Übergang zwischen zwei Knoten.
Startknoten		Markiert den Startpunkt und Endpunkt eines Ablaufs bei Aktivierung einer Aktivität.
Endknoten		
Verzweigungs- und Verbindungs-knoten		Ein Verzweigungsknoten spaltet eine Kante in mehrere alternative Abläufe auf, die in einem Verbindungsknoten zusammengeführt werden.
Synchroni-sations- und Parallelisie-rungsknoten		Ein Parallelisierungsknoten spaltet eine Kante in mehrere parallele Abläufe auf, die in einem Synchronisationsknoten zusammengeführt werden.

Tabelle F.2: Notationselemente für ein Aktivitätsdiagramm nach [RQdS12].

Literaturverzeichnis

[ABSH09] Amberger, J.; Bocchini, C. A.; Scott, A. F.; Hamosh, A.: McKusick's Online Mendelian Inheritance in Man (OMIM®). *Nucleic Acids Research*, Band 37, Nr. suppl 1, S. D793–D796, 2009.

[ADKG+06] Ai, X.; Do, A.-T.; Kusche-Gullberg, M.; Lindahl, U.; Lu, K.; Emerson, C. P.: Substrate Specificity and Domain Functions of Extracellular Heparan Sulfate 6-O-Endosulfatases, QSulf1 and QSulf2. *Journal of Biological Chemistry*, Band 281, Nr. 8, S. 4969–4976, 2006.

[ADL+03] Ai, X.; Do, A.-T.; Lozynska, O.; Kusche-Gullberg, M.; Lindahl, U.; Emerson, C. P.: QSulf1 remodels the 6-O sulfation states of cell surface heparan sulfate proteoglycans to promote Wnt signaling. *The Journal of Cell Biology*, Band 162, Nr. 2, S. 341–351, 2003.

[AHC+08] Andreeva, A.; Howorth, D.; Chandonia, J.-M.; Brenner, S. E.; Hubbard, T. J. P.; Chothia, C.; Murzin, A. G.: Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Research*, Band 36, Nr. suppl 1, S. D419–D425, 2008.

[AKO04] Abouelhoda, M. I.; Kurtz, S.; Ohlebusch, E.: Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, Band 2, Nr. 1, S. 53–86, 2004.

[AMM44] Avery, O. T.; MacLeod, C. M.; McCarty, M.: Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types: Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type III. *The Journal of Experimental Medicine*, Band 79, Nr. 2, S. 137–158, 1944.

[ASA+11] Arzt, S.; Starlinger, J.; Arnold, O.; Kröger, S.; Jaeger, S.; Leser, U.: PiPa: Custom Integration of Protein Interactions and Pathways. In *GI-Jahrestagung*, 2011.

[Bai00] Bairoch, A.: The ENZYME database in 2000. *Nucleic Acids Research*, Band 28, Nr. 1, S. 304–305, 2000.

[Bau07] Baumbach, J.: Coryneregnet 4.0 - a reference database for corynebacterial gene regulatory networks. *BMC Bioinformatics*, Band 8, Nr. 1, S. 429, 2007.

[BBC+06] Baumbach, J.; Brinkrolf, K.; Czaja, L.; Rahmann, S.; Tauch, A.: Coryneregnet: An ontology-based data warehouse of corynebacterial transcription factors and regulatory networks. *BMC Genomics*, Band 7, Nr. 1, S. 24, 2006.

[BCA+97] Bond, C. S.; Clements, P. R.; Ashby, S. J.; Collyer, C. A.; Harrop, S. J.; Hopwood, J. J.; Guss, J. M.: Structure of a human lysosomal sulfatase. *Structure*, Band 5, Nr. 2, S. 277–289, 1997.

[BCP+11] Bolser, D. M.; Chibon, P.-Y.; Palopoli, N.; Gong, S.; Jacob, D.; Angel, V. D. D.; Swan, D.; Bassi, S. et al.: MetaBase-the wiki-database of biological databases. *Nucleic Acids Research*, S. 1–5, 2011.

[BHGK06] Beckstette, M.; Homann, R.; Giegerich, R.; Kurtz, S.: Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, Band 7, S. 389, 2006.

[BKLP06] Bowler, T.; Kosman, D.; Licht, J. D.; Pick, L.: Computational Identification of Ftz/Ftz-F1 downstream target genes. *Developmental Biology*, Band 299, Nr. 1, S. 78–90, 2006.

[BSH+04] Beckstette, M.; Strothmann, D.; Homann, R.; Giegerich, R.; Kurtz, S.: PoSSuMsearch: Fast and Sensitive Matching of Position Specific Scoring Matrices using Enhanced Suffix Arrays. In *German Conference on Bioinformatics*, S. 53–64, 2004.

[BSL00] Blobe, G. C.; Schiemann, W. P.; Lodish, H. F.: Role of Transforming Growth Factor β in Human Disease. *The New England Journal of Medicine*, Band 342, Nr. 18, S. 1350–1358, 2000.

[BWKT09] Baumbach, J.; Wittkop, T.; Kleindt, C. K.; Tauch, A.: Integrated analysis and reconstruction of microbial transcriptional gene regulatory networks using coryneregnet. *Nature Protocols*, Band 4, Nr. 6, S. 992–1005, 2009.

[BWML06] Bailey, T. L.; Williams, N.; Misleh, C.; Li, W. W.: MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research*, Band 34, S. W369–W373, 2006.

[BWW+08] Baumbach, J.; Wittkop, T.; Weile, J.; Kohl, T.; Rahmann, S.: MoRAine - A web server for fast computational transcription factor binding motif re-annotation. *Journal of Integrative Bioinformatics*, Band 5, Nr. 2, 2008.

[BY06] Birkland, A.; Yona, G.: BIOZON: a system for unification, management and analysis of heterogeneous biological data. *BMC Bioinformatics*, Band 7, Nr. 1, S. 70, 2006.

[BYYO11] Brazas, M. D.; Yim, D. S.; Yamada, J. T.; Ouellette, B. F. F.: The 2011 bioinformatics links directory update: more resources, tools and databases and features to empower the bioinformatics community. *Nucleic Acids Research*, Band 39, Nr. Web Server issue, S. W3–W7, 2011.

[CAB+09] Cochrane, G.; Akhtar, R.; Bonfield, J.; Bower, L.; Demiralp, F.; Faruque, N.; Gibson, R.; Hoad, G. et al.: Petabyte-scale innovations at the European Nucleotide Archive. *Nucleic Acids Research*, Band 37, Nr. suppl 1, S. D19–D25, 2009.

[CBOS99] Chan, H.; Bartos, D. P.; Owen-Schaub, L. B.: Activation-Dependent Transcriptional Regulation of the Human fas Promoter Requires NF-κB p50-p65 Recruitment. *Molecular and Cellular Biology*, Band 19, Nr. 3, S. 2098–2108, 1999.

[CDS+09] Chau, B. N.; Diaz, R. L.; Saunders, M. A.; Cheng, C.; Chang, A. N.; Warrener, P.; Bradshaw, J.; Linsley, P. S. et al.: Identification of SULF2 as a Novel Transcriptional Target of p53 by Use of Integrated Genomic Analyses. *Cancer Research*, Band 69, Nr. 4, S. 1368–1374, 2009.

[CFG+05] Cartharius, K.; Frech, K.; Grote, K.; Klocke, B.; Haltmeier, M.; Klingenhoff, A.; Frisch, M.; Bayerlein, M. et al.: Matinspector and beyond: promoter analysis based on transcription factor binding sites. *Bioinformatics*, Band 21, Nr. 13, S. 2933–2942, 2005.

[CHK05] Chekmenev, D. S.; Haid, C.; Kel, A. E.: P-Match: transcription factor binding site search by combining patterns and weight matrices. *Nucleic Acids Research*, Band 33, Nr. suppl 2, S. W432–W437, 2005.

[CHS97] Chen, Q. K.; Hertz, G. Z.; Stormo, G. D.: PromFD 1.0: a computer program that predicts eukaryotic pol II promoters using strings and IMD matrices. *Computer Applications in the Biosciences*, Band 13, Nr. 1, S. 29–35, 1997.

[CK10] Castrop, H.; Kurtz, A.: Functional evidence confirmed by histological localization: overlapping expression of erythropoietin and HIF-2α in interstitial fibroblasts of the renal cortex. *Kidney International*, Band 77, Nr. 4, S. 269–271, 2010.

[CKS10] Czauderna, T.; Klukas, C.; Schreiber, F.: Editing, validating and translating of SBGN maps. *Bioinformatics*, Band 26, Nr. 18, S. 2340–2341, 2010.

[CLS+10] Chen, K.; Liu, M.-L.; Schaffer, L.; Li, M.; Boden, G.; Wu, X.; Williams, K. J.: Type 2 diabetes in mice induces hepatic overexpression of sulfatase 2, a novel factor that suppresses uptake of remnant lipoproteins. *Hepatology*, Band 52, Nr. 6, S. 1957–1967, 2010.

[CM95] Chen, I.-M. A.; Markowitz, V. M.: An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools. *Information Systems*, Band 20, Nr. 5, S. 393–418, 1995.

[CML+07] Choi, C.; Münch, R.; Leupold, S.; Klein, J.; Siegel, I.; Thielen, B.; Benkert, B.; Kucklick, M. et al.: SYSTOMONAS - an integrated database for systems biology analysis of Pseudomonas. *Nucleic Acids Research*, Band 35, Nr. suppl 1, S. D533–D537, 2007.

[Cod90] Codd, E. F.: *The Relational Model for Database Management, Version 2.* Addison-Wesley, 1990.

[Con97] Conrad, S.: *Föderierte Datenbanksysteme.* Springer-Verlag, Berlin, 1997.

[CPH+03] Cornell, M.; Paton, N. W.; Hedeler, C.; Kirby, P.; Delneri, D.; Hayes, A.; Oliver, S. G.: GIMS: an integrated data storage and analysis environment for genomic and functional data. *Yeast*, Band 20, Nr. 15, S. 1291–1306, 2003.

[CS90] Carr, C. S.; Sharp, P. A.: A helix-loop-helix protein related to the immunoglobulin e box-binding proteins. *Molecular and Cellular Biology*, Band 10, Nr. 8, S. 4384–4388, 1990.

[CSG+09] Chang, A.; Scheer, M.; Grote, A.; Schomburg, I.; Schomburg, D.: BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009. *Nucleic Acids Research*, Band 37, Nr. suppl 1, S. D588–D592, 2009.

[DAG+09] Dondrup, M.; Albaum, S. P.; Griebel, T.; Henckel, K.; Jünemann, S.; Kahlke, T.; Kleindt, C. K.; Küster, H. et al.: EMMA 2 - A MAGE-compliant system for the collaborative analysis and integration of microarray data. *BMC Bioinformatics*, Band 10, 2009.

[Dar05] Darwin, I. F.: *Java Kochbuch.* O'Reilly Verlag, Köln, 2. Auflage, 2005.

[DCP+10] Demir, E.; Cary, M. P.; Paley, S.; Fukuda, K.; Lemer, C.; Vastrik, I.; Wu, G.; D'Eustachio, P. et al.: The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, Band 28, Nr. 12, S. 1308, 2010.

[DD07] Das, M. K.; Dai, H.-K.: A survey of DNA motif finding algorithms. *BMC Bioinformatics*, Band 8, 2007.

[DEKB10] Do, L.; Esteves, F.; Karten, H.; Bier, E.: Booly: a new data integration platform. *BMC Bioinformatics*, Band 11, Nr. 1, S. 513, 2010.

[DGA+01] Dhoot, G. K.; Gustafsson, M. K.; Ai, X.; Sun, W.; Standiford, D. M.; Emerson Jr., C. P.: Regulation of Wnt Signaling and Embryo Patterning by an Extracellular Sulfatase. *Science*, Band 293, Nr. 5535, S. 1663–1666, 2001.

[DH05] Down, T. A.; Hubbard, T. J. P.: NestedMICA: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Research*, Band 33, Nr. 5, S. 1445–1453, 2005.

[DLS+99] Dierks, T.; Lecca, M. R.; Schlotterhose, P.; Schmidt, B.; Figura, K. v.: Sequence determinants directing conversion of cysteine to formylglycine in eukaryotic sulfatases. *The EMBO Journal*, Band 18, Nr. 8, S. 2084–2091, 1999.

[DM92] Day, W. H.; McMorris, F.: Critical comparison of consensus methods for molecular sequences. *Nucleic Acids Research*, Band 20, Nr. 5, S. 1093–1099, 1992.

[DOTW97] Davidson, S. B.; Overton, G. C.; Tannen, V.; Wong, L.: BioKleisli: A Digital Library for Biomedical Researchers. *International Journal on Digital Libraries*, Band 1, Nr. 1, S. 36–53, 1997.

[DQR+07] Das, R.; Qian, B.; Raman, S.; Vernon, R.; Thompson, J.; Bradley, P.; Khare, S.; Tyka, M. D. et al.: Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. *Proteins*, Band 69, Nr. S8, S. 118–128, 2007.

[DRD+11] Dräger, A.; Rodriguez, N.; Dumousseau, M.; Dörr, A.; Wrzodek, C.; Le Novère, N.; Zell, A.; Hucka, M.: JSBML: a flexible Java library for working with SBML. *Bioinformatics*, Band 27, Nr. 15, S. 2167–2168, 2011.

[DRJ+09] Dreyfuss, J. L.; Regatieri, C. V.; Jarrouge, T. R.; Cavalheiro, R. P.; Sampaio, L. O.; Nader, H. B.: Heparan sulfate proteoglycans: structure, protein interactions and cell signaling. *Anais da Academia Brasileira de Ciências*, Band 81, S. 409–429, 2009.

[DSB+03] Dierks, T.; Schmidt, B.; Borissenko, L. V.; Peng, J.; Preusser, A.; Mariappan, M.; Figura, K. v.: Multiple sulfatase deficiency is caused by mutations in the gene encoding the human c-alpha-formylglycine generating enzyme. *CELL*, Band 113, Nr. 4, S. 435–444, 2003.

[DSvF97] Dierks, T.; Schmidt, B.; Figura, K. v.: Conversion of cysteine to formylglycine: A protein modification in the endoplasmic reticulum. *Proceedings of the National Academy of Sciences of the United States of America*, Band 94, Nr. 22, S. 11963–11968, 1997.

[DTW+13] Dai, L.; Tian, M.; Wu, J.; Xiao, J.; Wang, X.; Townsend, J. P.; Zhang, Z.: AuthorReward: increasing community curation in biological knowledge wikis through automated authorship quantification. *Bioinformatics*, Band 29, Nr. 14, S. 1837–1839, 2013.

[EA93] Etzold, T.; Argos, P.: SRS - an indexing and retrieval tool for flat file data libraries. *Computer Applications in the Biosciences*, Band 9, Nr. 1, S. 49–57, 1993.

[EB05] Elliott, R. L.; Blobe, G. C.: Role of Transforming Growth Factor Beta in Human Cancer. *Journal of Clinical Oncology*, Band 23, Nr. 9, S. 2078–2093, 2005.

[EFH+11] Edlich, S.; Friedland, A.; Hampe, J.; Brauer, B.; Brückner, M.: *NoSQL*. Hanser Verlag, München, 2. Auflage, 2011.

[EYSJ02] Ellrott, K.; Yang, C.; Sladek, F. M.; Jiang, T.: Identifying transcription factor binding sites through markov chain optimization. *Bioinformatics*, Band 18, Nr. suppl 2, S. S100–S109, 2002.

[FÏ2] Füßmann, J.: Implementierung einer Suchmaschine und dynamischer Indexierung für biomedizinische Daten. Master's thesis, Universität Bielefeld, 2012.

[FAB+12] Flicek, P.; Amode, M. R.; Barrell, D.; Beal, K.; Brent, S.; Carvalho-Silva, D.; Clapham, P.; Coates, G. et al.: Ensembl 2012. *Nucleic Acids Research*, Band 40, Nr. D1, S. D84–D90, 2012.

[FGB12] Finn, R. D.; Gardner, P. P.; Bateman, A.: Making your database available through Wikipedia: the pros and cons. *Nucleic Acids Research*, Band 40, Nr. D1, S. D9–D12, 2012.

[FGH+87] Felgner, P. L.; Gadek, T. R.; Holm, M.; Roman, R.; Chan, H. W.; Wenz, M.; Northrop, J. P.; Ringold, G. M. et al.: Lipofection: a highly efficient, lipid-mediated DNA-transfection procedure. *Proceedings of the National Academy of Sciences*, Band 84, Nr. 21, S. 7413–7417, 1987.

[FGM+98] Fujibuchi, W.; Goto, S.; Migimatsu, H.; Uchiyama, I.; Ogiwara, A.; Akiyama, Y.; Kanehisa, M.: DBGET/LinkDB: an integrated database retrieval system. *Pacific Symposium on Biocomputing*, S. 683–694, 1998.

[FMD+09] Frese, M.-A.; Milz, F.; Dick, M.; Lamanna, W. C.; Dierks, T.: Characterization of the Human Sulfatase Sulf1 and Its High Affinity Heparin/Heparan Sulfate Interaction Domain. *Journal of Biological Chemistry*, Band 284, Nr. 41, S. 28033–28044, 2009.

[FPK+12] Fähling, M.; Persson, A. B.; Klinger, B.; Benko, E.; Steege, A.; Kasim, M.; Patzak, A.; Persson, P. B. et al.: Multilevel regulation of HIF-1 signaling by TTP. *Molecular Biology of the Cell*, Band 23, Nr. 20, S. 4129–4141, 2012.

[FSD08] Frese, M.-A.; Schulz, S.; Dierks, T.: Arylsulfatase G, a Novel Lysosomal Sulfatase. *Journal of Biological Chemistry*, Band 283, Nr. 17, S. 11388–11395, 2008.

[FSRG14] Fernández-Suárez, X. M.; Rigden, D. J.; Galperin, M. Y.: The 2014 Nucleic Acids Research Database Issue and an updated NAR online Molecular Biology Database Collection. *Nucleic Acids Research*, Band 42, Nr. D1, S. D1–D6, 2014.

[FSS11] Fazius, E.; Shelest, V.; Shelest, E.: SiTaR: a novel tool for transcription factor binding site prediction. *Bioinformatics*, 2011.

[GA13] Gressner, A. M.; Arndt, T.: *Lexikon der Medizinischen Laboratoriumsdiagnostik*. Springer-Verlag, Berlin, 2. Auflage, 2013.

[Gar13] Gardner, D.: Regulation neuer lysosomaler Hydrolasen durch TFEB. Universität Bielefeld, 2013.

[GB09] Günzel, H.; Bauer, A.: *Data-Warehouse-Systeme*. dpunkt.verlag, Heidelberg, 3. Auflage, 2009.

[Gil78] Gilbert, W.: Why genes in pieces? *Nature*, Band 271, Nr. 5645, S. 501, 1978.

[GJFS00] Gold, L. I.; Jussila, T.; Fusenig, N. E.; Stenbäck, F.: TGF-β isoforms are differentially expressed in increasing malignant grades of HaCaT keratinocytes, suggesting separate roles in skin carcinogenesis. *The Journal of Pathology*, Band 190, Nr. 5, S. 579–588, 2000.

[GK97] Giegerich, R.; Kurtz, S.: From Ukkonen to McCreight and Weiner: A Unifying View of Linear-Time Suffix Tree Construction. *Algorithmica*, Band 19, Nr. 3, 1997.

[GKS03] Giegerich, R.; Kurtz, S.; Stoye, J.: Efficient implementation of lazy suffix trees. *Software: Practice and Experience*, Band 33, Nr. 11, S. 1035–1049, 2003.

[GL02] Gilbert, S.; Lynch, N.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, Band 33, Nr. 2, S. 51–59, 2002.

[GLP+03] Garvey, T. D.; Lincoln, P.; Pedersen, C. J.; Martin, D.; Johnson, M.: BioSPICE: Access to the Most Current Computational Tools for Biologists. *OMICS: A Journal of Integrative Biology*, Band 7, Nr. 4, S. 411–420, 2003.

[GPP+09] Genethliou, N.; Panayiotou, E.; Panayi, H.; Orford, M.; Mean, R.; Lapathitis, G.; Gill, H.; Raoof, S. et al.: SOX1 links the function of neural patterning and Notch signalling in the ventral spinal cord during the neuron-glial fate switch. *Biochemical and Biophysical Research Communications*, Band 390, Nr. 4, S. 1114–1120, 2009.

[Gra00] Grabe, N.: AliBaba2: Context specific identification of transcription factor binding sites. *In Silico Biology*, Band 1, S. 19, 2000.

[Gra10] Graw, J.: *Genetik*. Springer-Verlag, Berlin, 5. Auflage, 2010.

[GS78] Galas, D. J.; Schmitz, A.: DNAase footprinting a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Research*, Band 5, Nr. 9, S. 3157–3170, 1978.

[GTRWL94] Gruber, P. J.; Torres-Rosado, A.; Wolak, M. L.; Left, T.: Apo CIII gene transcription is regulated by a cytokine inducible NF-κB element. *Nucleic Acids Research*, Band 22, Nr. 12, S. 2417–2422, 1994.

[HAW+06] Hagelueken, G.; Adams, T. M.; Wiehlmann, L.; Widow, U.; Kolmar, H.; Tümmler, B.; Heinz, D. W.; Schubert, W.-D.: The crystal structure of sdsa1, an alkylsulfatase from pseudomonas aeruginosa, defines a third class of sulfatases. *Proceedings of the National Academy of Sciences*, Band 103, Nr. 20, S. 7631–7636, 2006.

[HBCW03] Huang, H.; Barker, W. C.; Chen, Y.; Wu, C. H.: iProClass: an integrated database of protein family, function and structure information. *Nucleic Acids Research*, Band 31, Nr. 1, S. 390–392, 2003.

[HBRG+07] Holst, C. R.; Bou-Reslan, H.; Gore, B. B.; Wong, K.; Grant, D.; Chalasani, S.; Carano, R. A.; Frantz, G. D. et al.: Secreted Sulfatases Sulf1 and Sulf2 Have Overlapping yet Essential Roles in Mouse Neonatal Survival. *PLOS ONE*, Band 2, Nr. 6, S. e575, 06 2007.

[HBS+09] Haider, S.; Ballester, B.; Smedley, D.; Zhang, J.; Rice, P.; Kasprzyk, A.: BioMart Central Portal-unified access to biological data. *Nucleic Acids Research*, Band 37, Nr. suppl 2, S. W23–W27, 2009.

[HBW04] Hanson, S. R.; Best, M. D.; Wong, C.-H.: Sulfatases: Structure, Mechanism, Biological Activity, Inhibition, and Synthetic Utility. *Angewandte Chemie International Edition*, Band 43, Nr. 43, S. 5736–5763, 2004.

[HGHP+03] Hernandez-Guzman, F. G.; Higashiyama, T.; Pangborn, W.; Osawa, Y.; Ghosh, D.: Structure of human estrone sulfatase suggests functional roles of membrane association. *Journal of Biological Chemistry*, Band 278, Nr. 25, S. 22989–22997, 2003.

[HHP+10] Hartwig, S.; Ho, J.; Pandey, P.; MacIsaac, K.; Taglienti, M.; Xiang, M.; Alterovitz, G.; Ramoni, M. et al.: Genomic characterization of Wilms' tumor suppressor 1 targets in nephron progenitor cells during kidney development. *Development*, Band 137, Nr. 7, S. 1189–1203, 2010.

[Hip09] Hippe, K.: DAWIS-M.D. - Ein Data-Warehouse-System für metabolische Daten. Master's thesis, Universität Bielefeld, 2009.

[HKT+10] Hippe, K.; Kormeier, B.; Töpel, T.; Janowski, S.; Hofestädt, R.: DAWIS-M.D. - A Data Warehouse System for Metabolic Data. In *GI Jahrestagung (2)*, S. 720–725, 2010.

[HM09] Hart, R. K.; Mukhyala, K.: Unison: An integrated platform for computational biology discovery. In *Pacific Symposium on Biocomputing*, S. 403–414, 2009.

[HMPB+04] Hermjakob, H.; Montecchi-Palazzi, L.; Bader, G.; Wojcik, J.; Salwinski, L.; Ceol, A.; Moore, S.; Orchard, S. et al.: The HUPO PSI's molecular interaction format - a community standard for the representation of protein interaction data. *Nature Biotechnology*, Band 22, Nr. 2, S. 177–183, 2004.

[HPKM11] Heikkilä, M.; Pasanen, A.; Kivirikko, K. I.; Myllyharju, J.: Roles of the human hypoxia-inducible factor (HIF)-3α variants in the hypoxia response. *Cellular and Molecular Life Sciences*, Band 68, Nr. 23, S. 3885–3901, 2011.

[HSK+01] Haas, L. M.; Schwarz, P. M.; Kodali, P.; Kotlar, E.; Rice, J. E.; Swope, W. C.: DiscoveryLink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, Band 40, Nr. 2, 2001.

[IAS+01] Itoh, F.; Asao, H.; Sugamura, K.; Heldin, C.-H.; Dijke, P. t.; Itoh, S.: Promoting bone morphogenetic protein signaling through negative regulation of inhibitory smads. *The EMBO Journal*, Band 20, Nr. 15, S. 4132–4142, 2001.

[Int01] International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome. *Nature*, Band 409, S. 860–921, 2001.

[JF53] J.D., W.; F.H.C., C.: A Structure for Deoxyribose Nucleic Acid. *Nature*, Band 171, S. 737–738, 1953.

[JKS06] Junker, B. H.; Klukas, C.; Schreiber, F.: VANTED: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, Band 7, S. 109, 2006.

[JKT+10] Janowski, S.; Kormeier, B.; Töpel, T.; Hippe, K.; Hofestädt, R.; Willassen, N.; Friesen, R.; Rubert, S. et al.: Modeling of cell-cell communication processes with Petri nets using the example of quorum sensing. *In Silico Biology*, Band 10, Nr. 0003, 2010.

[KA03] Ko, P.; Aluru, S.: Space Efficient Linear Time Construction of Suffix Arrays. In *CPM*, S. 200–210, 2003.

[KAB+12] Kerrien, S.; Aranda, B.; Breuza, L.; Bridge, A.; Broackes-Carter, F.; Chen, C.; Duesbury, M.; Dumousseau, M. et al.: The IntAct molecular interaction database in 2012. *Nucleic Acids Research*, Band 40, Nr. D1, S. D841–D846, 2012.

[Kan97] Kanehisa, M.: Linking databases and organisms: GenomeNet resources in Japan. *Trends in Biochemical Sciences*, Band 22, Nr. 11, S. 442–444, 1997.

[KBBS02] Kisseleva, T.; Bhattacharya, S.; Braunstein, J.; Schindler, C. W.: Signaling through the JAK/STAT pathway, recent advances and future challenges. *Gene*, Band 285, Nr. 1-2, S. 1–24, 2002.

[KBT+06] Köhler, J.; Baumbach, J.; Taubert, J.; Specht, M.; Skusa, A.; Rüegg, A.; Rawlings, C.; Verrier, P. et al.: Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics*, Band 22, Nr. 11, S. 1383–1390, 2006.

[KBW+11] Krol, E.; Blom, J.; Winnebald, J.; Berhörster, A.; Barnett, M. J.; Goesmann, A.; Baumbach, J.; Becker, A.: RhizoRegNet - A database of rhizobial transcription factors and regulatory networks. *Journal of Biotechnology*, Band 155, Nr. 1, S. 127–134, 2011.

[KCS07] Klug, W. S.; Cummings, M. R.; Spencer, C. A.: *Genetik*. Pearson Studium, München, 2007.

[KDH+06] Kaps, A.; Dyshlevoi, K.; Heumann, K.; Jost, R.; Kontodinas, I.; Wolff, M.; Hani, J.: The BioRS(TM) Integration and Retrieval System: An open system for distributed data integration. *Journal of Integrative Bioinformatics*, Band 3, Nr. 2, 2006.

[Kec11] Kecher, C.: *UML 2*. Galileo Press, Bonn, 4. Auflage, 2011.

[KGR+03] Kel, A.; Gößling, E.; Reuter, I.; Cheremushkin, E.; Kel-Margoulis, O.; Wingender, E.: MATCH™: a tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Research*, Band 31, Nr. 13, S. 3576–3579, 2003.

[KGS+12] Kanehisa, M.; Goto, S.; Sato, Y.; Furumichi, M.; Tanabe, M.: KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, Band 40, Nr. D1, S. D109–D114, 2012.

[KHH11] Kormeier, B.; Hippe, K.; Hofestädt, R.: Data Warehouses in Bioinformatics: Integration of Molecular Biological Data. *it - Information Technology*, Band 53, Nr. 5, S. 241–249, 2011.

[KHK13] Kuhn, R. M.; Haussler, D.; Kent, W. J.: The UCSC genome browser and associated tools. *Briefings in Bioinformatics*, Band 14, Nr. 2, S. 144–161, 2013.

[KHL+03] Kofoed, E. M.; Hwa, V.; Little, B.; Woods, K. A.; Buckway, C. K.; Tsubaki, J.; Pratt, K. L.; Bezrodnik, L. et al.: Growth Hormone Insensitivity Associated with a STAT5b Mutation. *The New England Journal of Medicine*, Band 349, Nr. 12, S. 1139–1147, 2003.

[KIB+97] Karageorgos, L. E.; Isaac, E. L.; Brooks, D. A.; Ravenscroft, E. M.; Davey, R.; Hopwood, J. J.; Meikle, P. J.: Lysosomal Biogenesis in Lysosomal Storage Disorders. *Experimental Cell Research*, Band 234, Nr. 1, S. 85–97, 1997.

[KKG+09] Krawczyk, J.; Kohl, T. A.; Goesmann, A.; Kalinowski, J.; Baumbach, J.: From corynebacterium glutamicum to mycobacterium tuberculosis-towards transfers of gene regulatory networks and integrated data analyses with mycoregnet. *Nucleic Acids Research*, Band 37, Nr. 14, S. e97, 2009.

[KKH+11] Kinsella, R. J.; Kähäri, A.; Haider, S.; Zamora, J.; Proctor, G.; Spudich, G.; Almeida-King, J.; Staines, D. et al.: Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database*, Band 2011, 2011.

[KLL+12] Kowalewski, B.; Lamanna, W. C.; Lawrence, R.; Damme, M.; Stroobants, S.; Padva, M.; Kalus, I.; Frese, M.-A. et al.: Arylsulfatase G inactivation causes loss of heparan sulfate 3-O-sulfatase activity and mucopolysaccharidosis in mice. *Proceedings of the National Academy of Sciences*, 2012.

[KLM+11] Khurana, A.; Liu, P.; Mellone, P.; Lorenzon, L.; Vincenzi, B.; Datta, K.; Yang, B.; Linhardt, R. J. et al.: HSulf-1 Modulates FGF2- and Hypoxia-Mediated Migration and Invasion of Breast Cancer Cells. *Cancer Research*, Band 71, Nr. 6, S. 2152–2161, 2011.

[Kni06] Knippers, R.: *Molekulare Genetik*. Georg Thieme Verlag, Stuttgart, 9. Auflage, 2006.

[Kor10] Kormeier, B.: *Semi-automated reconstruction of biological networks based on a life science data warehouse*. Dissertation, Universität Bielefeld, 2010.

[KPGK+09] Keshava Prasad, T. S.; Goel, R.; Kandasamy, K.; Keerthikumar, S.; Kumar, S.; Mathivanan, S.; Telikicherla, D.; Raju, R. et al.: Human protein reference database-2009 update. *Nucleic Acids Research*, Band 37, Nr. suppl 1, S. D767–D772, 2009.

[KPRR04] Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W.: *Web Engineering*. dpunkt.verlag, Heidelberg, 1. Auflage, 2004.

[KPV+06] Krull, M.; Pistor, S.; Voss, N.; Kel, A.; Reuter, I.; Kronenberg, D.; Michael, H.; Schwarzer, K. et al.: TRANSPATH®: an information resource for storing and visualizing signaling pathways and their pathological aberrations. *Nucleic Acids Research*, Band 34, Nr. suppl 1, S. D546–D551, 2006.

[KRV+05] Köhler, J.; Rawlings, C.; Verrier, P.; Mitchell, R.; Skusa, A.; Rüegg, A.; Philippi, S.: Linking experimental results, biological networks and sequence analysis methods using Ontologies and Generalised Data Structures. *In Silico Biology*, Band 5, Nr. 1, S. 33–44, 2005.

[KS03] Kärkkäinen, J.; Sanders, P.: Simple Linear Work Suffix Array Construction. In *ICALP*, S. 943–955, 2003.

[Lan92] Lane, D. P.: Cancer. p53, guardian of the genome. *Nature*, Band 358, Nr. 6381, S. 15–16, 1992.

[LBP+06] Lamanna, W. C.; Baldwin, R. J.; Padva, M.; Kalus, I.; Dam, G. t.; Kuppevelt, T. H. v.; Gallagher, J. T.; Figura, K. v. et al.: Heparan sulfate 6-O-endosulfatases: discrete in vivo activities and functional co-operativity. *Biochemical Journal*, Band 400, Nr. 1, S. 63–73, 2006.

[LBP+12] Licata, L.; Briganti, L.; Peluso, D.; Perfetto, L.; Iannuccelli, M.; Galeota, E.; Sacco, F.; Palma, A. et al.: MINT, the molecular interaction database: 2012 update. *Nucleic Acids Research*, Band 40, Nr. D1, S. D857–D861, 2012.

[LGM+08] Liu, Y.; Gao, H.; Marstrand, T. T.; Ström, A.; Valen, E.; Sandelin, A.; Gustafsson, J.-A.; Dahlman-Wright, K.: The genome landscape of ERα- and ERβ-binding DNA regions. *Proceedings of the National Academy of Sciences*, Band 105, Nr. 7, S. 2604–2609, 2008.

[LKR+09] Liu, P.; Khurana, A.; Rattan, R.; He, X.; Kalloger, S.; Dowdy, S.; Gilks, B.; Shridhar, V.: Regulation of HSulf-1 Expression by Variant Hepatic Nuclear Factor 1 in Ovarian Cancer. *Cancer Research*, Band 69, Nr. 11, S. 4843–4850, 2009.

[LKT+98] Lukatela, G.; Krauss, N.; Theis, K.; Selmer, T.; Gieselmann, V.; Figura, K. v.; Saenger, W.: Crystal Structure of Human Arylsulfatase A: The Aldehyde Function and the Metal Ion at the Active Site Suggest a Novel Mechanism for Sulfate Ester Hydrolysis. *Biochemistry*, Band 37, Nr. 11, S. 3654–3664, 1998.

[LL52] Linderstrøm-Lang, K. U.: *Lane Medical Lectures: Proteins and Enzymes*. University series: Medical sciences. Stanford University Press, 1952.

[LLS09] Lübke, T.; Lobel, P.; Sleat, D. E.: Proteomics of the Lysosome. *Biochimica et Biophysica Acta*, Band 1793, Nr. 4, S. 625–635, 2009.

[LN07] Leser, U.; Naumann, F.: *Informationsintegration*. dpunkt.verlag, Heidelberg, 1. Auflage, 2007.

[LP98] Löffler, G.; Petrides, P. E.: *Biochemie und Pathobiochemie*. Springer-Verlag, Berlin, 6. Auflage, 1998.

[LPW+06] Lee, T.; Pouliot, Y.; Wagner, V.; Gupta, P.; Stringer-Calvert, D.; Tenenbaum, J.; Karp, P.: BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics*, Band 7, Nr. 1, S. 170, 2006.

[LR03] Leser, U.; Rieger, P.: Integration molekularbiologischer Daten. *Datenbank-Spektrum*, Band 6, S. 56–66, 2003.

[LSM+03] Lenhard, B.; Sandelin, A.; Mendoza, L.; Engstrom, P.; Jareborg, N.; Wasserman, W.: Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology*, Band 2, Nr. 2, S. 13, 2003.

[LWS+00] Lin, X.; Wei, G.; Shi, Z.; Dryer, L.; Esko, J. D.; Wells, D. E.; Matzuk, M. M.: Disruption of Gastrulation and Heparan Sulfate Biosynthesis in EXT1-Deficient Mice. *Developmental Biology*, Band 224, Nr. 2, S. 299–311, 2000.

[MÏ0] Müller, B.: *JavaServer Faces 2.0*. Hanser Verlag, München, 2. Auflage, 2010.

[MGG+09] Matthews, L.; Gopinath, G.; Gillespie, M.; Caudy, M.; Croft, D.; Bono, B. d.; Garapati, P.; Hemish, J. et al.: Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Research*, Band 37, Nr. suppl 1, S. D619–D622, 2009.

[Mil08] Milz, F.: Biochemische und zellbiologische Charakterisierung der Hydrophilen Domäne der humanen Sulfatase Sulf1. Master's thesis, Universität Bielefeld, 2008.

[Mil12] Milz, F.: *Die humane Sulfatase Sulf1: funktionale, biochemische und biophysikalische Studien zur Interaktion mit dem Substrat Heparansulfat.* Dissertation, Universität Bielefeld, 2012.

[MKP+04] Müller, I.; Kahnert, A.; Pape, T.; Sheldrick, G. M.; Meyer-Klaucke, W.; Dierks, T.; Kertesz, M.; Usón, I.: Crystal Structure of the Alkylsulfatase AtsK: Insights into the Catalytic Mechanism of the Fe(II) α-Ketoglutarate-Dependent Dioxygenase Superfamily. *Biochemistry*, Band 43, Nr. 11, S. 3075–3088, 2004.

[MLD+04] Mohamed, K. M.; Le, A.; Duong, H.; Wu, Y.; Zhang, Q.; Messadi, D. V.: Correlation between VEGF and HIF-1α expression in human oral squamous cell carcinoma. *Experimental and Molecular Pathology*, Band 76, Nr. 2, S. 143–152, 2004.

[MOPT07] Maglott, D.; Ostell, J.; Pruitt, K. D.; Tatusova, T.: Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, Band 35, Nr. suppl 1, S. D26–D31, 2007.

[MSH+09] Matt, P.; Schoenhoff, F.; Habashi, J.; Holm, T.; Van Erp, C.; Loch, D.; Carlson, O. D.; Griswold, B. F. et al.: Circulating transforming growth factor-β in Marfan syndrome. *Circulation*, Band 120, Nr. 6, S. 526–532, 2009.

[MW03] Merkl, R.; Waack, S.: *Bioinformatik Interaktiv.* WILEY-VCH Verlag, Weinheim, 2003.

[MW11] Meinold, M.; Westermeyer, M.: Nutzerorientierte Entwicklung eines modularen Software-Systems zur Suche regulatorischer Motive in großen Nukleotidsequenzen. Master's thesis, Universität Bielefeld, 2011.

[NHM+09] Novere, N. L.; Hucka, M.; Mi, H.; Moodie, S.; Schreiber, F.; Sorokin, A.; Demir, E.; Wegner, K. et al.: The Systems Biology Graphical Notation. *Nature Biotechnology*, Band 27, Nr. 8, S. 735–741, 2009.

[NSC+06] Narita, K.; Staub, J.; Chien, J.; Meyer, K.; Bauer, M.; Friedl, A.; Ramakrishnan, S.; Shridhar, V.: HSulf-1 Inhibits Angiogenesis and Tumorigenesis In vivo. *Cancer Research*, Band 66, Nr. 12, S. 6025–6032, 2006.

[Ort12] Ortkras, T.: Expressionsanalysen der humanen Arylsulfatase I. Universität Bielefeld, 2012.

[OSP97] Orlando, V.; Strutt, H.; Paro, R.: Analysis of chromatin structure by in vivo formaldehyde cross-linking. *Methods*, Band 11, Nr. 2, S. 205–214, 1997.

[PBvdS12] Platt, F. M.; Boland, B.; Spoel, A. C. v. d.: Lysosomal storage disorders: The cellular impact of lysosomal dysfunction. *The Journal of Cell Biology*, Band 199, Nr. 5, S. 723–734, 2012.

[PCTK+09] Portales-Casamar, E.; Thongjuea, S.; Kwon, A. T.; Arenillas, D.; Zhao, X.; Valen, E.; Yusuf, D.; Lenhard, B. et al.: JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. *Nucleic Acids Research*, Band 38, S. D105–D110, 2009.

[PDF97] Pellegrini, S.; Dusanter-Fourt, I.: The Structure, Regulation and Function of the Janus Kinases (JAKs) and the Signal Transducers and Activators of Transcription (STATs). *European Journal of Biochemistry*, Band 248, Nr. 3, S. 615–633, 1997.

[PIK+11] Palmieri, M.; Impey, S.; Kang, H.; Ronza, A. d.; Pelz, C.; Sardiello, M.; Ballabio, A.: Characterization of the CLEAR network reveals an integrated control of cellular clearance pathways. *Human Molecular Genetics*, 2011.

[PRN+12] Pauling, J.; Röttger, R.; Neuner, A.; Salgado, H.; Collado-Vides, J.; Kalaghatgi, P.; Azevedo, V.; Tauch, A. et al.: On the trail of EHEC/EAEC-unraveling the gene regulatory networks of human pathogenic Escherichia coli bacteria. *Integrative Biology*, Band 4, Nr. 7, S. 728–733, 2012.

[PRT+11] Pauling, J.; Röttger, R.; Tauch, A.; Azevedo, V.; Baumbach, J.: Coryneregnet 6.0 - updated database content, new analysis methods and novel features focusing on community demands. *Nucleic Acids Research*, 2011.

[PS93] Prestridge, D. S.; Stormo, G.: SIGNAL SCAN 3.0: new database and program features. *Computer applications in the biosciences*, Band 9, Nr. 1, S. 113–115, 1993.

[PSD+04] Philippar, U.; Schratt, G.; Dieterich, C.; Müller, J. M.; Galgóczy, P.; Engel, F. B.; Keating, M. T.; Gertler, F. et al.: The SRF Target Gene Fhl2 Antagonizes RhoA/MAL-Dependent Activation of SRF. *Molecular Cell*, Band 16, Nr. 6, S. 867–880, 2004.

[QFK+95] Quandt, K.; Frech, K.; Karas, H.; Wingender, E.; Werner, T.: MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Research*, Band 23, Nr. 23, S. 4878–4884, 1995.

[RCSKG12] Rivera-Colón, Y.; Schutsky, E. K.; Kita, A. Z.; Garman, S. C.: The structure of human GALNS reveals the molecular basis for mucopolysaccharidosis IV A. *Journal of Molecular Biology*, Band 423, Nr. 5, S. 736–751, 2012.

[Rei09] Reibold, H.: *phpMyAdmin kompakt: Alles, was Sie für den erfolgreichen Einstieg in den MySQL-Datenbankmanager wissen müssen.* Brain-Media.de, Saarbrücken, 2. Auflage, 2009.

[RJH+12] Rohn, H.; Junker, A.; Hartmann, A.; Grafahrend-Belau, E.; Treutler, H.; Klapperstück, M.; Czauderna, T.; Klukas, C. et al.: VANTED v2: a framework for systems biology applications. *BMC Systems Biology*, Band 6, S. 139, 2012.

[RKS+94] Ritter, O.; Kocab, P.; Senger, M.; Wolf, D.; Suhai, S.: Prototype Implementation of the Integrated Genomic Database. *Computers and Biomedical Research*, Band 27, Nr. 2, S. 97–115, 1994.

[RPW+05] Reed, M. J.; Purohit, A.; Woo, L. W. L.; Newman, S. P.; Potter, B. V. L.: Steroid Sulfatase: Molecular Biology, Regulation, and Inhibition. *Endocrine Reviews*, Band 26, Nr. 2, S. 171–202, 2005.

[RQdS12] Rupp, C.; Queins, S.; SOPHISTen d.: *UML 2 glasklar: Praxiswissen für die UML-Modellierung.* Hanser Verlag, München, 4. Auflage, 2012.

[SBB+00] Stevens, R.; Baker, P.; Bechhofer, S.; Ng, G.; Jacoby, A.; Paton, N. W.; Goble, C. A.; Brass, A.: TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. *Bioinformatics*, Band 16, Nr. 2, S. 184–186, 2000.

[SCB95] Swanson, H. I.; Chan, W. K.; Bradfield, C. A.: DNA Binding Specificities and Pairing Rules of the Ah Receptor, ARNT, and SIM Proteins. *Journal of Biological Chemistry*, Band 270, Nr. 44, 1995.

[SCdC+10] Sigrist, C. J. A.; Cerutti, L.; Castro, E. d.; Langendijk-Genevaux, P. S.; Bulliard, V.; Bairoch, A.; Hulo, N.: PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, Band 38, Nr. suppl 1, S. D161–D166, 2010.

[Sch08] Schug, J.: Using TESS to Predict Transcription Factor Binding Sites in DNA Sequence. In *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc., 2008.

[SD05] Sadée, W.; Dai, Z.: Pharmacogenetics/genomics and personalized medicine. *Human Molecular Genetics*, Band 14, Nr. suppl 2, S. R207–R214, 2005.

[SDP+07] Stanway, S. J.; Delavault, P.; Purohit, A.; Woo, L. W. L.; Thurieau, C.; Potter, B. V. L.; Reed, M. J.: Steroid Sulfatase: A New Target for the Endocrine Therapy of Breast Cancer. *The Oncologist*, Band 12, Nr. 4, S. 370–374, 2007.

[SH10] Skinner, M. E.; Holmes, I. H.: Setting Up the JBrowse Genome Browser. In *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc., 2010.

[SHL+97] Shi, Y.; Hata, A.; Lo, R. S.; Massagué, J.; Pavletich, N. P.: A structural basis for mutational inactivation of the tumour suppressor Smad4. *Nature*, Band 388, Nr. 6637, S. 87–93, 1997.

[SHX+05] Shah, S.; Huang, Y.; Xu, T.; Yuen, M.; Ling, J.; Ouellette, B. F.: Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, Band 6, Nr. 1, S. 34, 2005.

[SISK92] Shweiki, D.; Itin, A.; Soffer, D.; Keshet, E.: Vascular endothelial growth factor induced by hypoxia may mediate hypoxia-initiated angiogenesis. *Nature*, Band 359, Nr. 6398, S. 843–845, 1992.

[SKWB13] Stegmaier, P.; Kel, A.; Wingender, E.; Borlak, J.: A Discriminative Approach for Unsupervised Clustering of DNA Sequence Motifs. *PLOS Computational Biology*, Band 9, Nr. 3, S. e1002958, 03 2013.

[SMJ07] Schug, J.; Mintz, M.; Jr., C. J. S.: Data Integration and Pattern-Finding in Biological Sequence with TESS's Annotation Grammar and Extraction Language (AnGEL). In *DILS*, S. 188–203, 2007.

[SMO+03] Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N. S.; Wang, J. T.; Ramage, D.; Amin, N.; Schwikowski, B. et al.: Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, Band 13, Nr. 11, S. 2498–2504, 2003.

[SO97a] Schug, J.; Overton, G. C.: Modeling Transcription Factor Binding Sites with Gibbs Sampling and Minimum Description Length Encoding. In *ISMB*, S. 268–271, 1997.

[SO97b] Schug, J.; Overton, G. C.: TESS: Transcription Element Search Software on the WWW. Technischer Bericht, University of Pennsylvania, 1997.

[SPdR+09] Sardiello, M.; Palmieri, M.; Ronza, A. d.; Medina, D. L.; Valenza, M.; Gennarino, V. A.; Di Malta, C.; Donaudy, F. et al.: A Gene Network Regulating Lysosomal Biogenesis and Function. *Science*, Band 325, Nr. 5939, S. 473–477, 2009.

[SPPB06] Schmid, C. D.; Perier, R.; Praz, V.; Bucher, P.: EPD in its twentieth year: towards complete promoter coverage of selected model organisms. *Nucleic Acids Research*, Band 34, Nr. suppl 1, S. D82–D85, 2006.

[SRR08] Smith, T. G.; Robbins, P. A.; Ratcliffe, P. J.: The human side of hypoxia-inducible factor. *British Journal of Haematology*, Band 141, Nr. 3, S. 325–334, 2008.

[SS90] Schneider, T. D.; Stephens, R. M.: Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, Band 18, Nr. 20, S. 6097–6100, 1990.

[SS10] Saake, G.; Sattler, K.-U.: *Algorithmen und Datenstrukturen*. dpunkt.verlag, Heidelberg, 4. Auflage, 2010.

[STK+10] Sommer, B.; Tiys, E. S.; Kormeier, B.; Hippe, K.; Janowski, S. J.; Ivanisenko, T. V.; Bragin, A. O.; Arrigo, P. et al.: Visualization and Analysis of a Cardio Vascular Disease- and MUPP1-related Biological Network combining Text Mining and Data Warehouse Approaches. *Journal of Integrative Bioinformatics*, Band 7, Nr. 1, 2010.

[Sto00] Stormo, G. D.: DNA binding sites: representation and discovery. *Bioinformatics*, Band 16, Nr. 1, S. 16–23, 2000.

[SUS+09] Skinner, M. E.; Uzilov, A. V.; Stein, L. D.; Mungall, C. J.; Holmes, I. H.: JBrowse: A next-generation genome browser. *Genome Research*, Band 19, Nr. 9, S. 1630–1638, 2009.

[The10] The 1000 Genomes Project Consortium: A map of human genome variation from population-scale sequencing. *Nature*, Band 467, S. 1061–1073, 2010.

[The12a] The Gene Ontology Consortium: The Gene Ontology: enhancements for 2011. *Nucleic Acids Research*, Band 40, Nr. D1, S. D559–D564, 2012.

[The12b] The UniProt Consortium: Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Research*, Band 40, Nr. D1, S. D71–D75, 2012.

[TKKH08] Töpel, T.; Kormeier, B.; Klassen, A.; Hofestädt, R.: BioDWH: A Data Warehouse Kit for Life Science Data Integration. *Journal of Integrative Bioinformatics*, Band 5, Nr. 2, 2008.

[TLB+05] Tompa, M.; Li, N.; Bailey, T. L.; Church, G. M.; Moor, B. D.; Eskin, E.; Favorov, A. V.; Frith, M. C. et al.: Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, Band 23, S. 137–144, 2005.

[TPG01] Turnbull, J.; Powell, A.; Guimond, S.: Heparan sulfate: decoding a dynamic multifunctional cell regulator. *Trends in Cell Biology*, Band 11, Nr. 2, S. 75–82, 2001.

[TR09] Tang, R.; Rosen, S. D.: Functional Consequences of the Subdomain Organization of the Sulfs. *Journal of Biological Chemistry*, Band 284, Nr. 32, S. 21505–21514, 2009.

[TRM+05] Trißl, S.; Rother, K.; Müller, H.; Steinke, T.; Koch, I.; Preissner, R.; Frömmel, C.; Leser, U.: Columba: an integrated database of proteins, structures, and annotations. *BMC Bioinformatics*, Band 6, Nr. 1, S. 81, 2005.

[TS06] Türker, C.; Saake, G.: *Objektrelationale Datenbanken.* dpunkt.verlag, Heidelberg, 2006.

[UMTB+06] Uchimura, K.; Morimoto-Tomita, M.; Bistrup, A.; Li, J.; Lyon, M.; Gallagher, J.; Werb, Z.; Rosen, S.: HSulf-2, an extracellular endoglucosamine-6-sulfatase, selectively mobilizes heparin-bound growth factors and chemokines: effects on VEGF, FGF-1, and SDF-1. *BMC Biochemistry*, Band 7, Nr. 1, S. 2, 2006.

[vIVC+12] Iersel, M. P. v.; Villéger, A.; Czauderna, T.; Boyd, S. E.; Bergmann, F. T.; Luna, A.; Demir, E.; Sorokin, A. A. et al.: Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics*, Band 28, Nr. 15, S. 2016–2021, 2012.

[VPRP+13] Vera, R.; Perez-Riverol, Y.; Perez, S.; Ligeti, B.; Kertész-Farkas, A.; Pongor, S.: JBioWH: an open-source Java framework for bioinformatics data integration. *Database*, Band 2013, 2013.

[VRH10] Veerla, S.; Ringner, M.; Hoglund, M.: Genome-wide transcription factor binding site/promoter databases for the analysis of gene sets and co-occurrence of transcription factor binding motifs. *BMC Genomics*, Band 11, 2010.

[WAF+04] Wang, S.; Ai, X.; Freeman, S. D.; Pownall, M. E.; Lu, Q.; Kessler, D. S.; Emerson, C. P.: QSulf1, a heparan sulfate 6-O-endosulfatase, inhibits

fibroblast growth factor signaling in mesoderm induction and angiogenesis. *Proceedings of the National Academy of Sciences of the United States of America*, Band 101, Nr. 14, S. 4833–4838, 2004.

[WBLR07] Wittkop, T.; Baumbach, J.; Lobo, F.; Rahmann, S.: Large scale clustering of protein sequences with FORCE - A layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, Band 8, Nr. 1, S. 396, 2007.

[WCE+04] Wheeler, D. L.; Church, D. M.; Edgar, R.; Federhen, S.; Helmberg, W.; Madden, T. L.; Pontius, J. U.; Schuler, G. D. et al.: Database resources of the National Center for Biotechnology Information: update. *Nucleic Acids Research*, Band 32, Nr. suppl 1, S. D35–D40, 2004.

[WEL+10] Wittkop, T.; Emig, D.; Lange, S.; Rahmann, S.; Albrecht, M.; Morris, J. H.; Böcker, S.; Stoye, J. et al.: Partitioning biological data with transitivity clustering. *Nature methods*, Band 7, Nr. 6, S. 419–420, 2010.

[WHC+01] Wu, J. W.; Hu, M.; Chai, J.; Seoane, J.; Huse, M.; Li, C.; Rigotti, D. J.; Kyin, S. et al.: Crystal structure of a phosphorylated Smad2. Recognition of phosphoserine by the MH2 domain and insights on Smad function in TGF-beta signaling. *Molecular Cell*, Band 8, Nr. 6, S. 1277–1289, 2001.

[Win08] Wingender, E.: The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Briefings in Bioinformatics*, Band 9, Nr. 4, S. 326–332, 2008.

[Wit13] Witthus, P.: Erstellung einer Webanwendung für die Analyse von biomedizinischen Artikeln basierend auf Text Mining. Master's thesis, Universität Bielefeld, 2013.

[WJ06] Wei, Z.; Jensen, S. T.: GAME: detecting *cis*-regulatory elements using a genetic algorithm. *Bioinformatics*, Band 22, Nr. 13, S. 1577–1584, 2006.

[WKB+06] Wang, P.; Keijer, J.; Bunschoten, A.; Bouwman, F.; Renes, J.; Mariman, E.: Insulin modulates the secretion of proteins from mature 3T3-L1 adipocytes: a role for transcriptional regulation of processing. *Diabetologia*, Band 49, Nr. 10, S. 2453–2462, 2006.

[WM09] Wagener, C.; Müller, O.: *Molekulare Onkologie*. Georg Thieme Verlag, Stuttgart, 3. Auflage, 2009.

[WNMB00] Wu, T. D.; Nevill-Manning, C. G.; Brutlag, D. L.: Fast probabilistic analysis of sequence function using scoring matrices. *Bioinformatics*, Band 16, Nr. 3, S. 233–244, 2000.

[WS04] Wasserman, W. W.; Sandelin, A.: Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, Band 5, Nr. 4, S. 276–287, 2004.

[WWK+13] Wiegmann, E. M.; Westendorf, E.; Kalus, I.; Pringle, T. H.; Lübke, T.; Dierks, T.: Arylsulfatase K, a Novel Lysosomal Sulfatase. *Journal of Biological Chemistry*, Band 288, Nr. 42, S. 30019–30028, 2013.

[YBS+12] Yusuf, D.; Butland, S.; Swanson, M.; Bolotin, E.; Ticoll, A.; Cheung, W.; Zhang, X.; Dickman, C. et al.: The Transcription Factor Encyclopedia. *Genome Biology*, Band 13, Nr. 3, S. R24, 2012.