D I S S E R T A T I O N

Deduktiv unterstützte Rekonstruktion biologischer Netzwerke aus flexibel analysierten Textdaten

An der Technischen Fakultät
der Universität Bielefeld

vorgelegt von
Thorben Wallmeyer

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

Dipl.-Inform. Thorben Wallmeyer:
Deduktiv unterstützte Rekonstruktion biologischer Netzwerke aus flexibel analysierten Textdaten

Der Technischen Fakultät der Universität Bielefeld
am 04. Juli 2016 vorgelegt,
am 21. September 2016 verteidigt und genehmigt.

Gutachter:
Prof. Dr. Ralf Hofestädt, Universität Bielefeld
Prof. Dr. Jan Baumbach, Süddänische Universität

Prüfungsausschuss:
Prof. Dr. Philipp Cimiano, Universität Bielefeld
Prof. Dr. Ralf Hofestädt, Universität Bielefeld
Prof. Dr. Jan Baumbach, Süddänische Universität
Dr.-Ing. Sebastian Wrede, Universität Bielefeld

182 Seiten
64 Abbildungen
15 Tabellen

Gedruckt auf alterungsbeständigem Papier (ISO 9706)

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Harsewinkel, Juni 2016

Abstract

Für ein tiefgreifendes Verständnis lebender Organismen ist eine Betrachtung ihrer intra- sowie extrazellulären Prozesse unerlässlich. Sie sind daher Gegenstand der aktuellen Forschung in der Biologie und Medizin. Die Ergebnisse werden in Datenbanken oder semi-strukturierten Textdaten erfasst. Es ist eine Aufgabe der Bioinformatik, aus diesen Daten biologische Netzwerke zu rekonstruieren. Sie stellen die zellulären Prozesse eines Organismus formal dar und können entscheidende Impulse für die weitere Forschung geben. Eine computergestützte Rekonstruktion (*Pathway Prediction*) wird aktuell von zwei verschiedenen Ansätzen verfolgt. State of the Art ist der Einsatz komplexer *Textmining (TM)*-Algorithmen sowie etablierter *Datenintegrations (DIs)*-Verfahren. Sie fassen die erforderlichen Informationen aus verschiedenen Datenbanken zusammen (DI) oder extrahieren sie aus Textdaten (TM).

Das Ziel dieser Arbeit ist es, die etablierten *Pathway Prediction*-Verfahren um eine Logikkomponente zu erweitern. Sie verfügt über zusätzliches Regelwissen, unterstützt Schlussfolgerungen und weist damit den Charakter einer deduktiven Datenbank auf. Sobald rekonstruierte Netzwerke in ihr gespeichert werden, wird mittels Deduktion zusätzlicher Einfluss auf sie ausgeübt. Dadurch sollen neue Zusammenhänge erkannt und die Aussagekraft der rekonstruierten Netzwerke erhöht werden. Der Fokus dieser Arbeit liegt dabei auf der Netzwerkvorhersage mittels TM. Der entwickelte Prototyp *Framework for Medical Textmining (FraMeTex)*[1] greift hierfür auf bereits existierende Komponenten zurück. Es wird daher weder ein neuer TM-Algorithmus noch ein neues Datenbanksystem entwickelt. Stattdessen bietet der Prototyp eine weitreichende Flexibilität, um bereits existierende Ressourcen gewinnbringend einzusetzen. Es können beliebige TM-Algorithmen genutzt und zur Vorhersage möglichst präziser Netzwerk kombiniert werden. Gleichzeitig wird dadurch die Möglichkeit eröffnet, verschiedene biologische Netzwerke zu rekonstruieren. Gegenüber existierenden Systemen dieser Art stellt dies eine weitere Neuerung dar.

Neben der konzeptionellen Erweiterung unterscheidet sich die in dieser Arbeit präsentierte Netzwerkrekonstruktion insbesondere durch ihren ganzheitlichen Ansatz. Die Rekonstruktion biologischer Netzwerke wird als komplexer Prozess aufgefasst, der auch unterstützende Komponenten umfasst. Sie bieten Zugriff auf die zu analysierenden Textdaten und können diese effizient filtern. Beides ist eine essentielle Grundvoraussetzung für die Netzwerkrekonstruktion aus Textdaten und wurde bisher stets vernachlässigt. Dies führte zu einem modular konzipierten System. Seine Funktionalitäten können zur Rekonstruktion biologischer Netzwerke zusammengestellt oder unabhängig und einzeln genutzt werden. Die Realisierung des Prototyps

[1] http://agbi.techfak.uni-bielefeld.de/frametex

verfolgt dabei das langfristige Ziel, eine Integration der Funktionalitäten in existierende Workflow-Systeme der Bioinformatik zu ermöglichen. Da Workflow-Systeme von der technischen Ebene der zugrundeliegenden Funktionalitäten abstrahieren, sind Biologen und Mediziner mit ihnen vielfach vertraut. Der hier präsentierte, neuartige Ansatz der Netzwerkrekonstruktion kann ihre Forschung damit unmittelbar unterstützen.

Die Leistungsfähigkeit des in dieser Arbeit konzipierten Systems zeigen abschließend verschiedene Anwendungsfälle. Der Schwerpunkt liegt auf der Vorhersage eines MPDZ/MUPP1-Proteinnetzwerks aus Textdaten sowie der anschließenden Identifikation potentieller Protein-Komplexe mittels Deduktion. Da das MPDZ/MUPP1-Protein mit Herz- und Gefäßerkrankungen in Verbindung gebracht wird, wurden in der Vergangenheit bereits mit TM-basierten sowie integrativen Systemen entsprechende Netzwerke rekonstruiert. Die unterschiedlich rekonstruierten Netzwerke werden nun durch die Deduktionskomponente weiterverarbeitet und die erzielten Ergebnisse verglichen. Dieser Vergleich zeigt auch verschiedene Entwicklungspotentiale auf, die zum Abschluss der Arbeit zusammengefasst werden.

Inhaltsverzeichnis

1 Einleitung

Im Abschnitt 1.1 wird zunächst die Motivation dieser Arbeit erörtert. Es werden die Notwendigkeit und bestehenden Möglichkeiten der automatisierten Rekonstruktion biologischer Netzwerke (*Pathway Prediction*) skizziert. Hieraus leiten sich im Abschnitt 1.2 die zentralen Ziele dieser Arbeit ab. Ein Überblick auf ihre Gliederung im Abschnitt 1.3 schließt das Kapitel ab.

1.1 Motivation

Für das Verständnis lebender Organismen galt lange Zeit die vollständige Entschlüsselung ihres Erbguts (Genoms) als unerlässlich. Mittlerweile wurde diese Sichtweise jedoch durch die intensive Erforschung ihrer intra- und extrazellulären Prozesse abgelöst. Die schematische Darstellung dieser Prozesse dient zur Repräsentation von Stoffwechselwegen (*Metabolic Pathways*), Signalwegen zwischen Zellkomponenten (*Signaling Pathways*) oder Protein-Interaktionen und Genregulationen. In diesem Kontext ist die frei verfügbare Darstellung der *Biochemical Pathways* von Gerhard Michal [Mic99] besonders bekannt, deren erste Version bereits im Jahr 1965 veröffentlicht wurde[1]. Michals Arbeit visualisiert die in Organismen ablaufenden, bekannten Reaktionen mit den jeweils daran beteiligten Enzymen in einem komplexen Netzwerk. Seine Arbeit kann als Anstoß zur Aufbereitung digitaler Pathway-Maps angesehen werden, die in Datenbanken wie der *Kyoto Encyclopedia of Genes and Genomes (KEGG)* [KG00] abgelegt sind. Wegweisend war außerdem die spätere Arbeit von Donald Nicholson. Sie fasste alle bis dahin bekannten Stoffwechselwege in einem *Metabolic Pathway Chart* zusammen [Nic01].

Eine stetige, manuelle Aktualisierung der Pathways ist jedoch undenkbar. Die ungeheure Menge der ihnen zugrundeliegenden Daten macht dies nahezu unmöglich. Zudem wächst die Datenmenge aufgrund der voranschreitenden Forschung kontinuierlich weiter an. Der Aufbau sowie die Aktualisierung von Pathway-Maps wird daher durch Methoden der Bioinformatik unterstützt. Sie sind in der Lage Pathways (semi-)automatisch auf Basis bereits verfügbarer Daten zu generieren (*Pathway Prediction*). Technisch lassen sich an dieser Stelle zwei State of the Art Ansätze der *Pathway Prediction* unterscheiden. Eine gängige Methode ist die Integration mehrerer bestehender Datenquellen, deren Gesamtheit schließlich die Vorhersage potentieller Pathways zulässt. Eine Alternative ist die Analyse biomedizinischer Veröffentlichungen, denen die hierfür notwendigen Daten mit Hilfe hochspezialisierter TM-Algorithmen entnommen werden. Die beiden unterschiedlichen Ansätze führen schließlich zu Netzwerkvorhersagen, die

[1] http://www.roche.com/sustainability/for_communities_and_environment/philanthropy/science_-education/pathways.htm

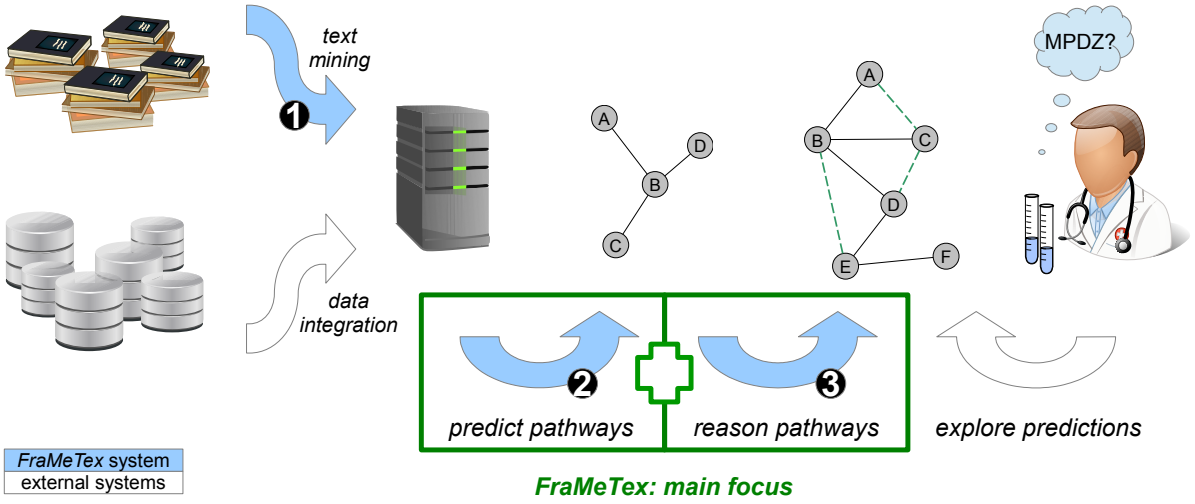


Abbildung 1.1: Einordnung des motivierten Ansatzes in etablierte Rekonstruktionsverfahren

Biologen und Medizinern wertvolle Impulse liefern können. Der Speicherung dieser Netzwerke wird bisher kaum Bedeutung geschenkt und erfolgt häufig in einer relationalen Datenbank. An dieser Stelle setzt diese Arbeit an. Sie stellt der genutzten Datenbank eine Logikkomponente zur Seite und versucht durch die Anwendung spezifischer Regeln zusätzliches Potential zu heben. Dies soll zur Ableitung weiterer Zusammenhänge führen und die Leistungsfähigkeit der *Pathway Prediction* insgesamt steigern. Eine umfangreiche Recherche und Analyse existierender Systeme zeigte, dass die skizzierte Funktionalität bisher nicht geboten wird. Die Entwicklung eines leistungsfähigen sowie flexiblen Frameworks, das alle Anforderungen an eine logikgestützte *Pathway Prediction* angemessen berücksichtigt, erscheint daher sinnvoll.

1.2 Zielsetzung

Im Zentrum dieser Arbeit steht die Erweiterung der textminingbasierten *Pathway Prediction* um eine logikunterstützte Speicherung der gewonnenen Netzwerke (Abbildung 1.1). Konzeptionell entspricht dies einer Verknüpfung auf *Pathway Prediction* spezialisierter TM-Algorithmen und Systeme mit einer deduktiven Datenbank-Komponente (i). Zur Vorhersage möglichst präziser Netzwerke sollen beliebige TM-Algorithmen unproblematisch kombiniert werden können (ii). Diese Flexibilität soll zugleich eine Rekonstruktion verschiedener, biologischer Netzwerke erlauben und eine Festlegung auf ein bestimmtes Netzwerk vermeiden (iii). Der motivierte Ansatz wird flexibel konzipiert, so dass in der motivierten Datenbank auch anderweitig (integrativ) vorhergesagte Netzwerke verarbeitet werden können (iv). Die Implementierung bedient sich existierender Komponenten und realisiert weder ein eigenständiges Datenbanksystem noch etwaige TM-Ressourcen (v). Mit dem verfolgten Ansatz soll vielmehr die bereits existierende Vielfalt entsprechender Algorithmen und Systeme unproblematisch genutzt werden können.

Mit einer Konzeptstudie (*FraMeTex*[2]) soll zunächst die technische Realisierbarkeit geprüft werden (*proof of concept*). Im Anschluss sollen mit dem entsprechenden Prototyp konkrete biologische Netzwerke rekonstruiert werden. Dies umfasst auch die experimentelle Rekonstruktion eines metabolischen Netzwerks. Den Schwerpunkt bildet die Vorhersage und Verarbeitung von MPDZ/MUPP1-Proteinnetzwerken. Das spezielle Protein wird mit Herz- und Gefäßerkrankungen in Verbindung gebracht und war in der Vergangenheit bereits Gegenstand verschiedener Netzwerkvorhersagen. Zum Einsatz kamen hierfür die beiden Systeme *Associative Network Discovery System (ANDSystem)* sowie *Visualization and Analysis of Networks in System Biology Applications (VANESA)*. Während VANESA auf einem integrierten Datenbestand setzt, verfolgt ANDSystem einen TM-basierten Ansatz. Die von ihnen rekonstruierten MPDZ/MUPP1-Netzwerke werden mit dem entsprechenden, von *FraMeTex* rekonstruierten Netzwerk verglichen. Damit kann das Deduktionspotential im Kontext der Netzwerkrekonstruktion erstmals erörtert werden. Anschließend werden die drei unterschiedlich rekonstruierten Netzwerke mit der geschaffenen Deduktionskomponente weiterverarbeitet. Dies zeigt ihr problemloses Zusammenspiel mit den beiden etablierten *Pathway Prediction*-Verfahren. Beabsichtigt ist die Identifikation von Protein-Komplexen, die als Auslöser von Krankheiten gelten und daher in der Biologie und Medizin von besonderem Interesse sind. Die mittels Deduktion in den Netzwerken identifizierten Protein-Komplexe werden abschließend ebenfalls einander gegenübergestellt. Anhand der in den Anwendungsfällen erzielten Ergebnisse soll schließlich eine qualifizierte Antwort auf eine bisher unbeantwortete Frage gefunden werden:

> *„Kann eine mit Logik unterstützte Datenbank die Pathway Prediction aus Textdaten gewinnbringend unterstützen?“*

Die mit dem Prototyp erzielbaren Ergebnisse hängen allerdings von Faktoren ab, die mit der biomedizinischen Problemstellung nur indirekt in Verbindung stehen. Einerseits müssen aus dem stetig wachsenden Datenvolumen zunächst aussagekräftige Eingabedaten selektiert werden, andererseits ist die Wahl der für ihre Analyse genutzten TM-Algorithmen entscheidend. Beides muss daher ebenfalls Berücksichtigung finden. Im Idealfall können unterschiedliche TM-Algorithmen zur Analyse herangezogen werden und es erfolgt keine konkrete Festlegung durch das System. In Abhängigkeit ihrer Spezialisierung sowie der selektierten Eingabedaten soll der resultierende Prototyp zur Rekonstruktion verschiedener Netzwerke genutzt werden können. Die Realisierung des Konzepts erfolgt in Form einzelner Module, die jeweils einzelne Funktionsbausteine zur Verfügung stellen. Sie sollen verkettet werden und in ihrer Gesamtheit alle erforderlichen Funktionalitäten bieten können. Damit wird nicht nur die motivierte *Pathway Prediction* abgedeckt, sondern auch die zwingend erforderliche Anbindung und Filterung der zu analysierenden Textdaten. Die prototypische Implementierung soll bereits alle in *Medical Literature Analysis and Retrieval System Online (Medline)* verfügbaren Abstracts in die *Pathway Prediction* einbeziehen können. Zukünftig sollen darüber hinaus weitere Datenquellen erschlossen werden können. Eine leistungsfähige Filterung der Eingabedaten ist daher eine unumgängliche Voraussetzung.

Besondere Berücksichtigung erfordert der Aufbau der Deduktionskomponente. Sie muss die rekonstruierten Netzwerke aufnehmen, strukturiert speichern und mit zusätzlichem Regelwis-

[2] http://agbi.techfak.uni-bielefeld.de/frametex

sen in Verbindung bringen. Außerdem soll sie auch Schlussfolgerungen in Netzwerken ermöglichen, die mit anderen Tools vorhergesagt wurden. Nach Möglichkeit soll sowohl der unkomplizierte Zugriff auf bereits existierendes Regelwissen als auch die Formulierung individueller Regeln möglich sein. Im Rahmen der prototypischen Implementierung erfolgt zunächst jedoch nur die Formulierung ausgewählter Regeln, deren Anwendung beispielsweise die motivierte Identifikation der Protein-Komplexe zum Ziel hat. Der Aufbau eines komplexeren Regelwerks kann dazu später analog erfolgen.

1.3 Gliederung der Arbeit

Im Kapitel 2 werden zunächst ausgewählte Grundlagen präsentiert, die mit dieser Arbeit in Verbindung stehen. Aufgrund des interdisziplinären Charakters der motivierten *Pathway Prediction* umfassen sie sowohl biologische als auch technische Grundlagen. Zu Beginn des Kapitel 3 werden verwandte Arbeiten aus dem Umfeld der *Pathway Prediction* präsentiert. Sie zeigen die existierenden Möglichkeiten auf, biomedizinische Textdaten zu analysieren und Netzwerke aus ihnen zu rekonstruieren. Außerdem werden Systeme präsentiert, mit denen der Aufbau einer Deduktionskomponente möglich ist. Im Kapitel 4 erfolgt schließlich der konkrete Einstieg in diese Arbeit. Die zentralen Anforderungen an eine deduktiv unterstützte *Pathway Prediction* werden formuliert und auf das zu ihrer Umsetzung verfolgte Systemkonzept eingegangen. Die prototypische Implementierung des konzipierten Systems wird im Kapitel 5 vorgestellt. Die erfolgreiche Rekonstruktion verschiedener, biologischer Netzwerke mit Unterstützung der Deduktionskomponente zeigen die im Kapitel 6 präsentierten Anwendungsfälle. Dies schließt auch eine deduktive Identifikation von Protein-Komplexen in unterschiedlich rekonstruierten MPDZ/MUPP1-Netzwerken ein. Zusätzlich wird anhand der erfolgreichen Rekonstruktion metabolischer Pathways aus Textdaten die Flexibilität des geschaffenen Frameworks unter Beweis gestellt. Im Rahmen einer abschließenden Diskussion im Kapitel 7 werden die Potentiale der deduktiv unterstützten *Pathway Prediction* noch einmal zusammengefasst. Dies umfasst auch eine selbstkritische Bewertung des Ansatzes, die sowohl seine Potentiale hervorhebt als auch mögliche Grenzen aufzeigt.

2 Grundlagen

Der interdisziplinäre Charakter dieser Arbeit legt die Präsentation elementarer biologischer und technischer Grundlagen nahe. Im Abschnitt 2.1 wird zunächst auf die biologischen Grundlagen eingegangen. Sie zielen darauf ab, ein Verständnis für die in biologischen Netzwerken dargestellten Zusammenhänge zu entwickeln. In diesem Kontext wird auch Medline präsentiert, dessen textbasierte Daten in dieser Arbeit zur Rekonstruktion biologischer Netzwerke dienen. Technologien, die einen Bezug zur im Abschnitt 1.2 motivierten *Pathway Prediction* haben, werden anschließend im Abschnitt 2.2 diskutiert. Sie bilden die Basis für die Implementierung eines entsprechenden Prototyps.

2.1 Biologische Grundlagen

Bereits die Motivation im Abschnitt 1.1 zeigte, dass intra- sowie extrazelluläre Prozesse in den Lebenswissenschaften von elementarer Bedeutung sind. Sie beruhen zumeist auf biochemischen Reaktionen einzelner Zellbestandteile, die in biologischen Netzwerken formal erfasst werden können. Die wesentlichen Zusammenhänge werden nachfolgend skizziert[1].

2.1.1 Zellen in Organismen

Die organische Zelle wurde bereits 1665 von Robert Hooke entdeckt. Erst deutlich später wurde jedoch der zelluläre Aufbau aller Organismen in der Zelltheorie formuliert. Sie sieht die Zelle als kleinste, lebende Einheit eines Organismus an. Die Zelle verfügt beispielsweise über einen eigenen Stoff- und Energiewechsel, ist in der Lage auf Reize zu reagieren und kann sich durch Zellteilung vermehren.

Anhand ihrer Zellanzahl werden die Organismen in Einzeller und Vielzeller unterschieden. Die Zellen selbst werden aufgrund ihrer Struktur noch einmal in eukaryotische (Eucyten) sowie prokaryotische Zellen (Procyten) unterteilt. Entscheidendes Merkmal für die Differenzierung der beiden Zelltypen ist der Zellkern (Nukleus), der nur in Eucyten vorhanden ist. Im Regelfall entsprechen Procyten einem einzelligen Organismus, während Eucyten auch Teil eines mehrzelligen Organismus sein können. In Abhängigkeit ihres Zelltyps werden Organismen in Prokaryoten und Eukaryoten unterschieden. Unabhängig vom Zelltyp weist jede Zelle eine Zellmembran auf, die die Zelle nach außen begrenzt und ihr Inneres (Zytoplasma) umschließt.

[1] Einen umfassenden Einblick in die Thematik bieten [PELS07] [CH04] [Alb08].

Base	Abkürzung
Adenin	A
Guanin	G
Thymin	T
Guanin	G
Uracil	U

Tabelle 2.1: Reihenfolge der Basen in der DNS mit üblichen Abkürzungen

Zugleich kontrolliert die Membran den Austausch von Stoffen mit der Zellumgebung. Die Zelle kann sich darüber mit Nährstoffen versorgen, um sie in Energie zu verwandeln und sich mit ihr selbst am Leben zu halten. Innerhalb der Zelle finden sich zahlreiche Komponenten. Besonders hervorzuheben sind neben Proteinen, Ribosomen und Enzymen die Erbinformationen des Organismus (Genom). Die genetischen Informationen werden von der *Desoxyribonukleinsäure (DNS)* getragen. Aus chemischer Sicht handelt es sich bei der DNS um Nukleinsäuren, die langen Molekülketten (Polymeren) entsprechen. Die Ketten entstehen durch eine bestimmte Anordnung von vier verschiedenen Basen, die von einem Rückgrat aus Zucker und Phosphaten zusammengehalten werden. Die Verbindung einer Base mit diesem Rückgrat wird als Nukleotid bezeichnet. Die Tabelle 2.1 zeigt die Anordnung der Basen für die DNS.

Strukturell entspricht die DNS einer Doppelhelix, die aus zwei entgegengesetzt angeordneten (antiparallelen) Strängen entsteht und eine rechtsdrehende Spirale darstellt. Dabei ist exakt festgelegt, welche der vier Basen des einen Strangs sich mit welcher Base des anderen Strangs verbindet. Die Zuordnung wird als komplementäre Basenpaarung bezeichnet. Dadurch liegen die in der DNS kodierten Informationen redundant vor und können nach dem Reißverschlussprinzip voneinander getrennt werden. Für die Replikation der DNS ist dies von elementarer Bedeutung. Im Gegensatz zur DNS entspricht die *Ribonukleinsäure (RNS)* einem einfachen Strang, in dem die Base Thymin allerdings durch Uracil ausgetauscht ist. Erstmal formuliert wurde das Modell der DNS von Watson und Crick im Jahr 1953 [WC53]. Die Abbildung 2.1 zeigt ein schematische Darstellung dieses Modells.

Bei Eukaryoten ist die DNS in regelmäßigen Abständen um große Proteinkomplexe aufgewickelt. Die Struktur wird als Chromatin bezeichnet und ist im Zellkern angesiedelt. In hoch kondensierter Form entspricht es Chromosomen, die jedoch nur in bestimmten Zellteilungsphasen existieren. Damit liegt jeder Zelle ein Bauplan vor, der den Aufbau aller anderen Zellbestandteile erlaubt. Dies umfasst die bereits erwähnten Proteine und Enzyme aber auch die RNS sowie weitere Moleküle. Umwelteinflüsse sowie instabile Basenpaare können jedoch Änderungen an der DNS hervorrufen (Mutationen) und zu einem fehlerhaften Aufbau der Zelle führen. Diese spontanen, auf einzelne Basenpaare oder Basenabschnitte begrenzten Schäden treten ständig auf. Bis auf wenige Ausnahmen werden sie durch einen komplexen Reparaturapparat immer wieder behoben. Größere Schäden können jedoch zu genetischen Veränderungen führen, die Krankheiten auslösen. Ein bekanntes Beispiel hierfür ist Trisomie 21 [WPU13].

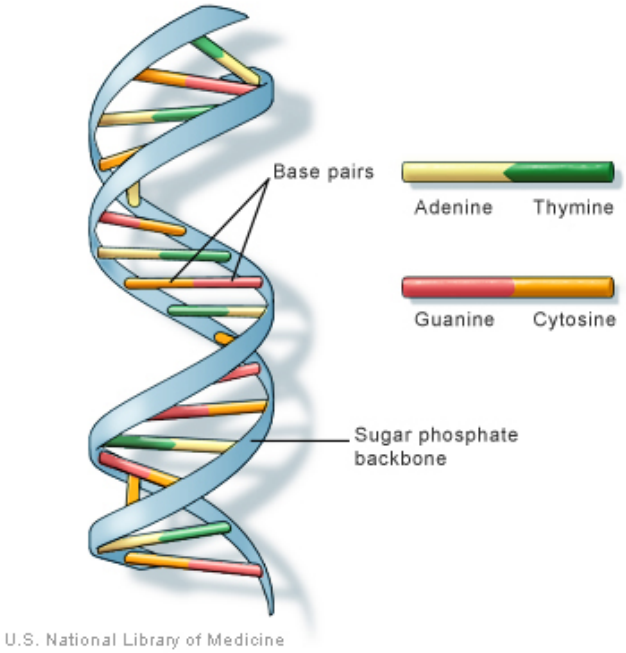


Abbildung 2.1: DNS-Modell mit komplementären Basen und Zucker-Phosphat-Rückgrat (Quelle: http://ghr.nlm.nih.gov/handbook/illustrations/dnsstructure.jpg)

2.1.2 Zellbestandteile: Ribosome, Proteine, Enzyme

Bereits im vorherigen Abschnitt wurde hervorgehoben, dass Proteine in allen Zellen enthalten sind. Ihnen kommen dabei zwei zentrale Aufgaben zu. Einerseits geben sie einer Zelle ihre spezifische Struktur, andererseits sind sie an fast allen biochemischen Reaktionen innerhalb der Zelle beteiligt. Chemisch entsprechen Proteine Molekülen aus Aminosäuren, die durch Peptidbindungen verkettet sind. Eine Kette kann dabei weniger als hundert Aminosäuren umfassen (Peptide), aber auch eine Länge von mehreren tausend Aminosäuren aufweisen (Polypeptide). Die Reihenfolge der Aminosäuren, die ein Protein bilden (Sequenz), ist in der DNS kodiert. Die in den Zellen enthaltenen Ribosomen werten diese Information aus und sind für den Aufbau der Proteine verantwortlich. Im menschlichen Körper werden die Proteine aus 20 verschiedenen Aminosäuren sowie Selenocystein gebildet. Acht dieser Aminosäuren können vom Körper allerdings nicht selbst produziert werden und müssen daher mit der Nahrung aufgenommen werden. Die konkrete Funktion bzw. Wirkung eines Proteins in der Zelle hängt letztlich von dessen aufgebauter, dreidimensionaler Struktur (Folding) ab. Die Struktur eines Proteins kann bis heute allerdings noch nicht anhand seiner DNS-Sequenz vorhergesagt werden. Proteine, deren Strukturen sich nur in Nuancen voneinander unterscheiden, werden als isoform bezeichnet. Im menschlichen Körper finden sich ungefähr 50.000 verschiedene Proteine, die für die Immunabwehr verantwortlich sein können oder chemische Reaktionen beeinflussen.

Proteine, die chemische Reaktionen auslösen können, werden als Enzyme bezeichnet. Enzy-

me sind damit zumeist Proteine, die eine bestimmte Wirkung haben. Aus chemischer Sicht entsprechen sie Katalysatoren, die chemische Reaktionen um einen Faktor von bis zu 10^{20} beschleunigen können, ohne selbst von der ausgelösten Reaktion verbraucht zu werden. Mittlerweile sind fast 4.000 verschiedene Enzyme bekannt, die biochemische Reaktionen auslösen können [Bai00]. Gegenüber anderen Katalysatoren sind Enzyme jedoch hochspezifisch und lösen daher nur ganz bestimmte Reaktionen aus. Aufgrund dieser Tatsache spiegeln die Namen der Enzyme häufig die Reaktion bzw. die daran beteiligten Elemente wieder. Sie werden hierfür lediglich um das Suffix *-ase* erweitert. Ein weithin bekanntes Beispiel ist die DNS Polymerase, der im Rahmen der Genexpression eine zentrale Rolle zukommt (Abschnitt 2.1.3). Mit einer von der *International Union of Biochemistry and Molecular Biology (IUBMB)* eingeführten Nomenklatur wird die Namensvergabe für Enzyme (*Enzyme Commission (EC)* numbers) weiter strukturiert [Bar97]. Die von ihr vergebenen, eindeutigen Nummern lassen sogar Rückschlüsse auf die vom jeweiligen Enzym katalysierten Reaktionen zu. Für das Verständnis einer Reaktion reicht die alleinige Betrachtung der Enzyme jedoch oft nicht aus, da Enzyme selbst wiederum von anderen Molekülen beeinflusst werden können. Bestimmte Moleküle können die Enzymaktivität hemmen (Inhibitoren) oder als Aktivator noch weiter verstärken. Viele Medikamente nutzen dies aus und wirken als Inhibitoren auf Zellprozesse ein. Zusätzlichen Einfluss auf die Enzymaktivität können zudem äußere Einflüsse, wie beispielsweise die Umgebungstemperatur, haben.

2.1.3 Genexpression: Transkription, Translation

Die Genexpression verantwortet den Aufbau (Synthese) von Proteinen. Da Proteine ihrerseits der Zelle ihre Struktur verleihen und an den chemischen Reaktionen innerhalb der Zelle beteiligt sind (Abschnitt 2.1.2), beeinflusst die Genexpression damit sowohl Aussehen als auch Funktion einer Zelle. Verantwortlich für die Proteinsynthese sind die Ribosomen der Zelle. Die hierfür notwendigen Informationen sind in der DNS kodiert, die entweder in dem Zellkern (Eukaryoten) oder in Zytoplasma (Prokaryoten) der Zelle enthalten ist. Die Genexpression ist damit für die Ausbildung des Phänotyps (Ausprägung eines Organismus) anhand des Genotyps (kodierte Erbinformationen des Genom) verantwortlich.

Die Genexpression wird von der Genregulation gesteuert und läuft in zwei elementaren Schritten ab. Der gesamte Vorgang ist in Abbildung 2.2 schematisch dargestellt. Zu Beginn erfolgt die Transkription, die sich für die Synthese von RNS anhand der gegebenen DNS einer Zelle verantwortlich zeichnet. Hierzu bindet sich das Enzym (RNS Polymerase II) an die DNS und öffnet den antiparallelen Strang der Doppelhelix. Von dem geöffnetem Strang werden nun die Gene abgelesen und als *messenger RNS (mRNS)* vervielfältigt. Bei Prokaryoten wird der Beginn eines kodierten Gens innerhalb der DNS Sequenz durch ein Promotor angezeigt, an das sich die RNS Polymerase bindet. Das Ende markiert ein expliziter Terminator. Ob ein Gen allerdings überhaupt kopiert werden soll, wird von Transkriptionsfaktoren entschieden. Diese speziellen Proteine können die Transkription eines Gens hemmen (Repressor) oder aktivieren (Aktivator). Dies geschieht, indem sich die Transkriptionsfaktoren an speziellen Stellen an die DNS binden. Eine negative Genregulierung (Repression) ist immer dann erforderlich, wenn bereits

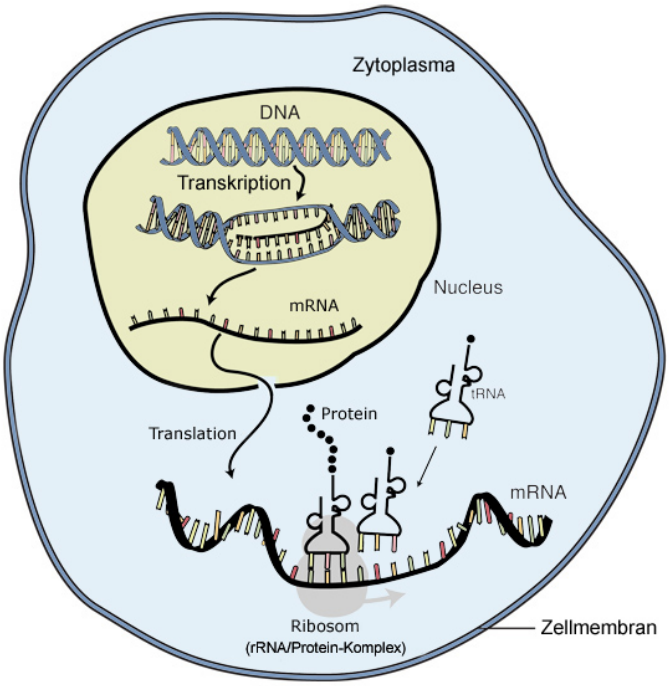


Abbildung 2.2: Vereinfachte Darstellung der Genexpression mit Transkription und Translation (Quelle: http://commons.wikimedia.org/wiki/File:Transkription_Translation_01.jpg

genug Substanz in der Zelle verfügbar ist. Im Regelfall erfolgt daher eine Aktivierung, um eine spezifische Substanz zu erzeugen, die momentan noch nicht in der Zelle zur Verfügung steht.

Nach Abschluss der Transkription entspricht die Kopie dem Komplementär des abgelesenen Strangs. In diesem ist jedoch die DNS spezifische Aminosäure Thymin durch Uracil ersetzt worden (Abschnitt 2.1.1). Nachdem die mRNS vorliegt, folgt mit der Translation der zweite Schritt der Genexpression. Sie dekodiert die zuvor während der Transkription erzeugte mRNS und produziert auf dessen Basis die Proteine. Diese Aufgabe übernehmen die Ribosomen im Zytoplasma der Zelle, die entsprechende (Poly-)Peptide erzeugen. Unterstützt wird der Vorgang von der *transfer RNS (tRNS)*. Anhand der in der mRNS kodierten Informationen vermittelt sie die jeweils korrekten, korrespondierenden Aminosäuren zum Aufbau der Polypeptide bzw. Proteine. Die beiden Teilschritte der Genexpression, Transkription und Translation, können sich durchaus überlagern. Die Ribosomen beginnen daher oftmals bereits mit der Proteinsynthese während die mRNS noch erzeugt bzw. kopiert wird. Die simultane Synthese verschiedener Proteine wird als Koexpression bezeichnet.

2.1.4 Biologische Netzwerke und Netzwerkvorhersage

In den vorherigen Abschnitten wurde der grundlegende Aufbau sowie die elementare Funktion einer Zelle erörtert. Es wurde deutlich, dass zelluläre Abläufe auf molekularen Prozessen

basieren. Sie entsprechen chemischen Reaktionen, die von Enzymen katalysiert werden können. In diesem Zusammenhang wurde die von der Genregulation gesteuerte Genexpression als ein zentraler, molekularer Prozess exemplarisch hervorgehoben. Die Funktionsweise sowie der Ablauf dieser Prozesse war in den vergangenen Jahren Gegenstand intensiver Forschung. Dennoch existiert bis heute von ihnen nur ein abstraktes Verständnis, dass auf ihre extreme Komplexität zurückzuführen ist. Zur Beschreibung der zellulären Prozesse dienen biologische Netzwerke, die sich in vier Kategorien unterscheiden lassen [JS08]:

- metabolische Netzwerke
- genregulatorische Netze
- Signalübertragungs-Netzwerke
- Protein-Interaktions-Netzwerke

In der Bioinformatik werden biologische Netzwerke formell als Graph aufgefasst. Ein Graph G ist ein 2-Tupel (V, E) bestehend aus einer endlichen Menge Knoten (Vertices) und Kanten (Edges). Eine Kante verbindet dabei jeweils zwei Knoten und wird durch eine Linie repräsentiert. Sind die Zusammenhänge zwischen den Knoten in einem Graphen jeweils nur in eine Richtung gültig, handelt es sich um einen gerichteten Graphen andernfalls um einen ungerichteten Graphen. In gerichteten Graphen werden die Kanten im Regelfall durch Pfeile dargestellt, denen die Richtung der Beziehung entnommen werden kann [JS08]. In biologischen Netzwerken wird ein konkreter Pfad, also eine Verbindung mehrerer Knoten über Kanten, auch als *Pathway* bezeichnet. Ein Pathway repräsentiert in diesem Kontext einen einzelnen zellulären Prozess. Mehrere Pathways bilden schließlich ein zusammenhängendes, biologisches Netzwerk.

2.1.4.1 Metabolische Netzwerke

Metabolische Netzwerke bilden den Stoffwechsel in Zellen ab. Dies umfasst die Aufnahme und den Transport von Stoffen sowie die Abgabe von Stoffwechselprodukten an die Umgebung. Die Prozesse dienen dazu, die Zellen eines Organismus mit Energie zu versorgen, die beispielsweise zur DNS-Replikation (Abschnitt 2.1.3) oder zur Umwandlung in Bewegung benötigt wird. Die diesen Stoffwechselwegen (*metabolic pathways*) zugrundeliegenden, enzymatischen Reaktionen verantworten letztlich die Transformation einzelner Moleküle und sind hochgradig organisiert [KHK+08]. Ein einzelner Pathway setzt sich dabei in der Regel aus mehreren, hintereinander folgenden enzymatischen Reaktionen zusammen. Die Zwischenprodukte dieser Reaktionen werden als Metabolite bezeichnet und die Menge aller enzymatischen Reaktionen innerhalb einer Zelle als Metabolismus. In der graphbasierten Darstellung biologischer Netzwerke entsprechen die Knoten Metaboliten und jede Kante einer einzelnen, enzymatischen Reaktion. Da Stoffwechsel im Regelfall in einer bestimmten Richtung ablaufen, erfolgt die Repräsentation mit einem gerichtetem Graphen. Aufgrund ihres Umfangs werden metabolische Netzwerke oftmals in Form großer Karten dargestellt. Besonders bekannt sind Donald Nicholsons *Pathway*

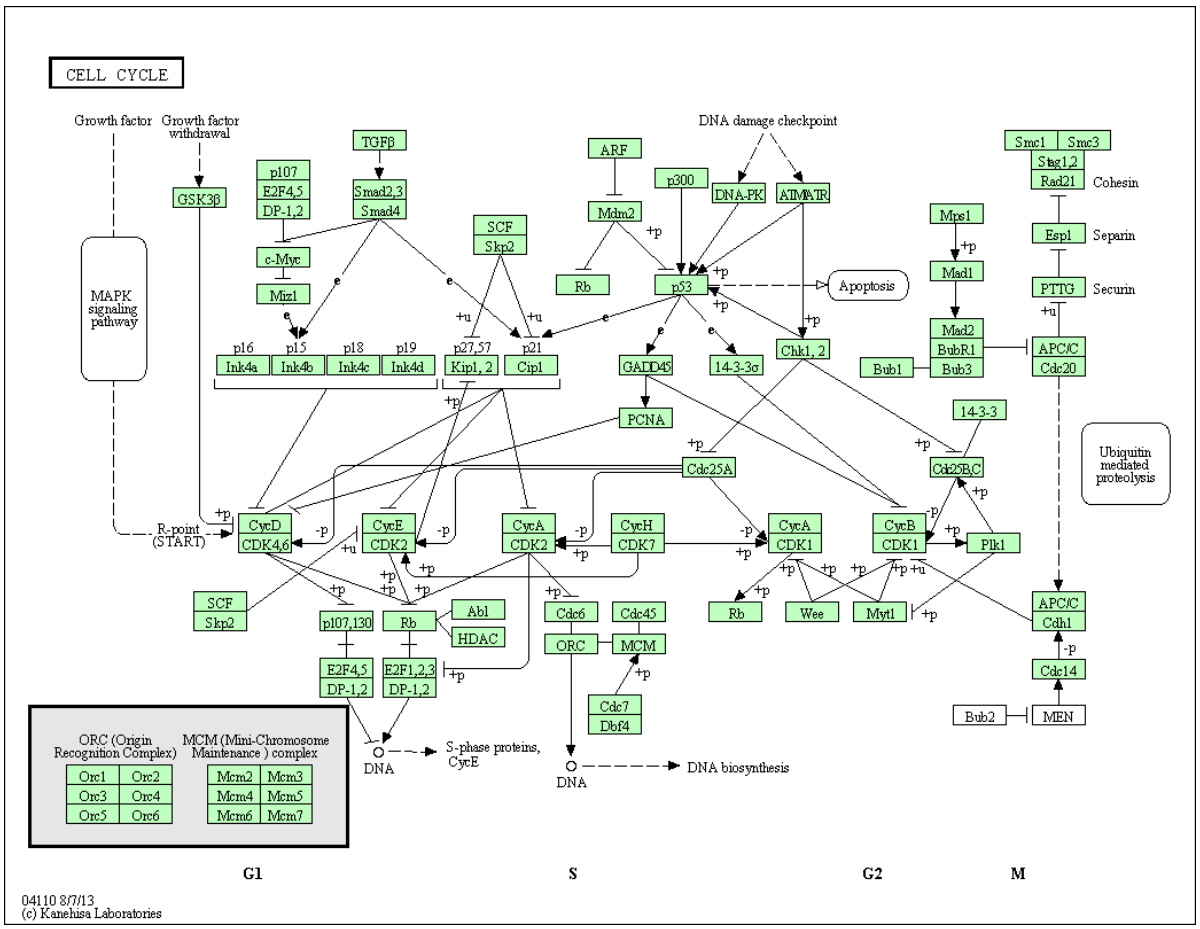


Abbildung 2.3: Darstellung des humanen Zellzyklus aus KEGG (Quelle: http://www.genome.jp/kegg-bin/show_pathway?hsa04110)

Chart [Nic01] sowie Gerhard Michals *Biochemical Pathways* [Mic99]. In den letzten Jahren wurden sie jedoch auch zunehmend digitalisiert und stehen in großen Datenbanken wie KEGG[2] [KG00], Reactome[3] [JGV+05] oder *BRaunschweig ENzyme DAtabase (BRENDA)*[4] [SCS02] öffentlich zur Verfügung. Metabolische Netzwerke stehen daher sowohl im Fokus der Biologie als auch der Bioinformatik und werden intensiv erforscht. Einen exemplarischen Pathway aus KEGG zeigt die Abbildung 2.3.

Eine enzymatische Reaktion beginnt mit der Bindung eines bestimmten Moleküls (Substrats) an das Enzym. Das Enzym verfügt hierfür über ein aktives Zentrum, das die Bindung erlaubt. Im Anschluss erfolgt die eigentliche Reaktion, bei der das Substrat verbraucht und das von der Reaktion erzeugte Produkt vom Enzym entlassen wird. Neben dem Substrat kann sich auch ein Inhibitor-Molekül an das Enzym binden und damit die zuvor beschriebene Bindung eines Substrats verhindern. Eine enzymatische Reaktion kann in diesem Fall nicht erfolgen. Abbildung

[2] http://www.genome.jp/kegg/
[3] http://www.reactome.org/
[4] http://www.brenda-enzymes.org/

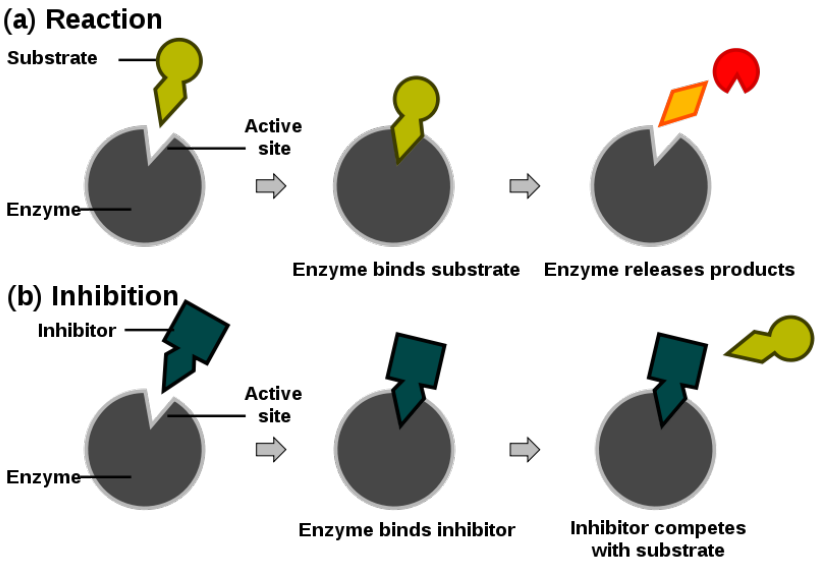


Abbildung 2.4: Inhibition und Reaktion von Enzymen (Quelle: http://de.wikipedia.org/wiki/Enzym)

2.4 stellt sowohl eine potentielle Reaktion als auch eine Inhibition schematisch dar. Formell lässt sich eine enzymatische Reaktion auffassen durch:

$$E + S \rightleftharpoons ES \rightleftharpoons EP \rightleftharpoons E + P \tag{2.1}$$

Zu Beginn formen das Enzym E sowie das Substrat S einen Enzym-Substrat-Komplex (ES), der in einen Enzym-Produkt-Komplex (EP) überführt wird. Eine chemische Reaktion splittet diesen Komplex anschließend auf und das Enzym kann das erzeugte Produkt entlassen. Innerhalb einer Sekunde können Enzyme mehrere Millionen dieser Reaktionen katalysieren, die durchaus bidirektional ablaufen können. Ein Produkt P kann sich wiederum an ein Enzym binden und eine weitere Reaktion katalysieren. Die Richtung der Reaktion hängt von verschiedenen Parametern des jeweiligen Enzyms ab, zu denen u.a. die Affinität (Bindungsstärke) zählt.

2.1.4.2 Genregulatorische Netzwerke

Genregulatorische Netzwerke haben einen hohen Stellenwert, da sie für das Verständnis der Genexpression elementar sind. Sie wurden daher in den vergangenen Jahren intensiv erforscht. Ein *genregulatorisches Netzwerk (GRN)* beschreibt die Interaktion von DNS-Segmenten innerhalb einer Zelle, die regulierend auf die Expression der Gene wirken. Die Interaktionen können sowohl direkt als auch indirekt über andere Zellbestandteile erfolgen. Damit sind vielfach auch Proteine oder Enzyme an den Interaktionen beteiligt. Von besonderem Interesse sind in diesem Kontext die Transkriptionsfaktoren und damit diejenigen Proteine, die einzelne Gene aktivieren oder hemmen können (Abschnitt 2.1.3). Dies geschieht durch ihre Bindung an einen Promotor und führt zur Produktion neuer Proteine, die ihrerseits wiederum als Transkriptionsfaktoren agieren können. Dies führt zu komplexen Kaskaden, die mit Hilfe genregulatori-

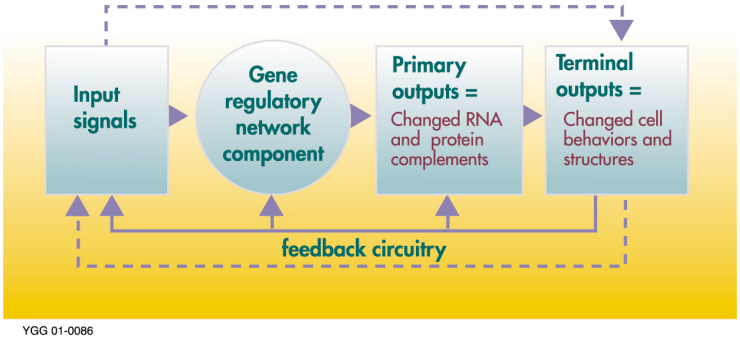


Abbildung 2.5: Signale beeinflussen genregulatorisches Netzwerk (Quelle: http://genomics.energy.gov)

scher Netzwerke beschrieben werden. Letztlich führt die darüber kontrollierte Genregulation zu Veränderungen an der Zelle, die sich beispielsweise in ihrer Struktur oder ihrem Verhalten bemerkbar machen können.

Aus abstrakter Sicht stellt ein GRN ein chemisches System dar. Es zeigt für die einzelnen Zellbestandteile die möglichen Wechselwirkungen auf. In der graphbasierten Darstellung entsprechen die Knoten des Netzwerks den interagierenden Zellbestandteilen und damit den DNS-Segmenten und Proteinen. Die Kanten zwischen den Knoten spiegeln molekulare Reaktionen wider. Sie repräsentieren die Interaktionen zwischen den beteiligten Komponenten. Dies kann auch zu Rückkopplungen führen, die über zyklische Pathways im GRN abgebildet werden. Aufgrund der Ähnlichkeit des GRN zu einem chemischen System [Gar69] kann das Ergebnis einer vom GRN kontrollierten Genexpression mit biochemischen Computermodellen berechnet werden. Voraussetzung hierfür ist jedoch die Identifikation bestimmter DNS-Sequenzen, an die sich die spezifischen Transkriptionsfaktoren binden können.

Entscheidend für die Aktivität eines GRN ist ein äußerer Einfluss (Signal) auf die Zelle. Dieser löst die Interaktionen bzw. Reaktionen innerhalb der Zelle aus, die zur Transkription von RNS führen und damit die Ribosomen zur Synthese von Proteinen veranlassen können. Der Ablauf ist in Abbildung 2.5 schematisch dargestellt. Er zeigt auch eine potentielle Rückkopplung. Sie tritt auf, wenn der vom GRN erzeugte Output das GRN selbst erneut beeinflusst und führt zu den bereits erwähnten Kaskaden der Genregulation.

2.1.4.3 Signalübertragungs-Netzwerke

Die Zellen eines Organismus müssen in der Lage sein Informationen aus der Zellumgebung zu empfangen. Die Übertragung der Informationen aus der Zellumgebung in die zellulären Systeme des Zytoplasma entspricht einem chemischen Prozess. Jede Zelle verfügt hierfür über Rezeptoren in der Zellmembran, die Reize (Stimuli) empfangen können. Rezeptoren sind spezielle Proteine, die sich in etwa 20 verschiedene Klassen unterscheiden lassen. Innerhalb jeder

Klasse gibt es eine Vielzahl Isoforme und damit eine große Bandbreite spezifischer Rezeptoren. Sie können jeweils auf einen bestimmten Stimulus reagieren [PELS07]. Potentielle Stimuli können chemische Substanzen, Licht oder auch Wachstumsfaktoren sein. Sobald ein Rezeptor stimuliert wurde, wird das aufgenommene Signal über Signalwege (signaling pathways) an Effektorproteine weitergeleitet. Dies erfolgt über Protein-Interaktionen und entspricht der eigentlichen Signaltransduktion. Damit können durchaus mehrere hundert Proteine innerhalb der Zelle an der Weiterleitung des Signals beteiligt sein. Die sich daraus ergebenen Netzwerke verantworten letztlich komplexe Zellfunktionen [JI98] [JLI00].

Eine Zelle reagiert damit auf einen Stimulus mit der Anpassung ihrer Zellaktivität. Dies wurde bereits anhand der GRNs im Abschnitt 2.1.4.2 deutlich, die durch Signale zur Regulation der Genexpression angeregt werden. Unabhängig vom konkreten Stimulus sind es drei Schritte bis zur Reaktion der Zelle:

1. Ein Rezeptor empfängt einen Stimulus indem sich eine Substanz an ihn bindet.
2. Der Stimulus wird vom Rezeptor in ein chemisches Signal umgewandelt (*Messenger*) und innerhalb der Zelle mittels Protein-Interaktionen weitergegeben. Die Umwandlung des Signals wird als *Transduktion* bezeichnet.
3. Effektoren reagieren auf die Messenger und führen zur Veränderung des Zellverhaltens.

Der skizzierte Ablauf wird von parallelen Signalwegen jedoch zusätzlich beeinflusst. Signalübertragungsnetzwerke sind daher für ihre vielfältigen Verknüpfungen bekannt, die sie insbesondere von metabolischen und genregulatorischen Netzwerken unterscheiden.

2.1.4.4 Protein-Interaktions-Netzwerke

Eine einfache Protein-Interaktion beschreibt die Wechselwirkung zwischen zwei Proteinen. Die Interaktion mehrerer Proteine werden in einem Protein-Interaktionsnetzwerk zusammengefasst. In einem derartigem Netzwerk repräsentiert ein Knoten ein bestimmtes Protein und die Kanten die bekannten Interaktionen. Die Erforschung von Protein-Interaktionen ist sowohl in der Biologie als auch in der Medizin von entscheidender Bedeutung, da sie an fast allen biologischen Prozessen beteiligt sind. Besonders hervorzuheben sind in diesem Zusammenhang einerseits metabolische Netzwerke, sowie andererseits Signalübertragungsnetzwerke. Protein-Interaktionen dienen beispielsweise zur intrazellulären Signalübertragung und ermöglichen es damit einer Zelle auf einen extrazellulären Reiz zu reagieren.

Insbesondere auch die Erforschung schwerer Krankheiten stützt sich auf Protein-Interaktions-Netzwerke. Da die Netzwerke die Grundlage vieler molekularer Prozesse bilden, werden Abweichungen von regulären Interaktionen als potentieller Ursprung einer Erkrankung angesehen. Im menschlichen Organismus sind aktuell über 650.000 Interaktionen bekannt, die in öffentlichen Datenbanken wie KEGG oder Reactome hinterlegt sind [STS+08]. Neben experimentellen Ansätzen werden häufig auch computerbasierte Ansätze zum Aufdecken von Protein-Interaktionen herangezogen. Sie analysieren große Datenmengen, versuchen potentielle Interaktionen zu extrahieren und fassen sie anschließend in einem Netzwerk zusammen.

2.1.4.5 Netzwerkvorhersagen

Die Informatik bietet aktuell zwei etablierte Verfahren, mit denen ein semi-automatischer Aufbau biologischer Netzwerke möglich ist. Einerseits besteht die Möglichkeit mit Textmining-Verfahren die erforderlichen Informationen aus biomedizinische Textdaten zu extrahieren, andererseits ist auch eine Integration verschiedener, bestehender Datenbestände möglich. Die Gesamtheit der integrierten Daten lässt dann die Konstruktion entsprechender Netzwerke zu. Oftmals wird der Vorgang auch als *Pathway Prediction* bezeichnet, da zunächst einzelne Pfade (re)konstruiert werden und anschließend in einem Netzwerk zusammengefasst werden. Beide Verfahren garantieren jedoch nicht die Korrektheit der resultierenden Netzwerke und erfordern daher stets eine manuelle Validierung durch Experten. Dennoch liefern sie Biologen wie Medizinern wertvolle Hinweise, die eine zielgerichtete Forschung erlauben.

2.1.5 Medline: textbasiertes Wissen für Netzwerkvorhersagen

Medline zählt zu den bekanntesten, textbasierten Datenquellen der Medizin und Biologie. Medline wurde vom *National Center for Biotechnology Information (NCBI)* entwickelt und hat aufgrund des umfangreichen Datenbestands auch für die Medizin- und Bioinformatik eine elementare Bedeutung. In Medline werden die Veröffentlichungen aus über 5.000, zumeist akademischen, Zeitschriften in fast vierzig Sprachen zusammengefasst. Die ältesten Einträge stammen aus dem Jahr 1946. Mehrere Updates spielen wöchentlich zwischen zwei- und viertausend neue Einträge ein. Mittlerweile umfasst Medline so über 24.000.000 Datensätze (März 2015), von denen jeder zumindest den Autor, den Titel und die Quelle der jeweiligen Veröffentlichung enthält. In den meisten Fällen ist zusätzlich eine kurze Zusammenfassung (Abstract) verfügbar, die fast immer in Englisch verfasst ist [med13]. Eine Besonderheit in Medline sind die *Medical Subject Headings (MeSH)*. Sie stellen ein kontrolliertes Vokabular dar, mit dessen Hilfe eine Kategorisierung und erste Indizierung der Einträge in Medline möglich ist [LB94]. Die Zuordnung erfolgt manuell und wird maßgeblich von spezialisierten Suchalgorithmen genutzt, um relevante Einträge leichter und zielgerichteter in dem riesigen Datenbestand identifizieren zu können. Dies nutzt auch das webbasierte *Public Medline (PubMed)*[5] aus, mit dem ein standardisierter und zugleich kostenloser Zugriff auf Medline möglich ist.

Genügt PubMed den Anforderungen hingegen nicht, weil beispielsweise individuellere Datenzugriffe gewünscht sind, kann gegen eine Lizenzgebühr auch Zugriff auf die physischen Datenstrukturen erlangt werden. In diesem Fall wird Medline dateibasiert distribuiert und der gesamte Datenbestand ist auf eine Vielzahl komprimierter Dateien verteilt. Jede dieser komprimierten Dateien enthält selbst wiederum mehrere Dateien mit *eXtended Markup Language (XML)*-Daten, die jeweils mehrere hundert Veröffentlichungen in ihrer XML-Struktur beschreiben. Die in Medline enthaltenen Informationen sind somit semi-strukturiert, da einzelne Felder innerhalb der XML-Struktur identifiziert werden können. Auf die Informationen bzw. das Wissen in den unstrukturierten Textdaten dieser Felder selbst kann jedoch nicht ohne weiteres zugegriffen werden. Hierfür sind spezialisierte Wissensextraktionsverfahren erforderlich, die sich

[5] http://www.ncbi.nlm.nih.gov/pubmed

insbesondere auf die Abstracts der einzelnen Datensätze konzentrieren. Die Wissensextraktion wird durch die ungeheure Datenmenge jedoch sehr komplex. Alle distribuierten Dateien ergeben zusammen ein Datenvolumen, dass mehrere *Giga Byte (GB)* umfasst. Dies stellt auch besondere Anforderungen an etwaige, individuell realisierte Datenzugriffe. Sie müssen nicht nur die Datenmenge verarbeiten, sondern auch die verschachtelte und komprimierte Dateistruktur traversieren sowie die einzelnen XML-Dateien effizient verarbeiten können.

Diese Herausforderung führte zur Entwicklung von *SemanticMedline* [KFR+08]. Das System leitet mit PubMed identifizierte Medline-Einträge automatisch an eine Wissensextraktion weiter. Mit *SemRep* [RF03] werden Aussagen extrahiert, die sich jeweils aus Subjekt, Prädikat und Objekt (Triplett) zusammensetzen. Die Aussagen werden anschließend anhand verschiedener Kriterien (z.B. Krankheitsbehandlung, Substanz-Interaktionen) gruppiert und in unterschiedlichen Graphstrukturen zusammengefasst [FRK04]. Sie können vom Anwender exploriert werden und verweisen zugleich auf die analysierten Datensätze.

2.2 Technische Grundlagen

Die motivierte Rekonstruktion biologischer Netzwerke stützt sich auf vielfältige Technologien. Sie sind teilweise selbst noch Gegenstand intensiver Forschung und damit hochaktuell. Dies gilt insbesondere für die Wissensextraktions-Algorithmen (Textmining) sowie den Aufbau einer logikunterstützten Wissensbasis. Die Grundzüge dieser sich noch stetig weiterentwickelnden Verfahren werden nachfolgend verständlich präsentiert. Ergänzend wird außerdem auf die seit längerem etablierten Konzepte zur Datenfilterung und -persistierung eingegangen.

2.2.1 Wissensextraktion aus Textdaten

Die große Herausforderung bei der Verarbeitung von natürlich-sprachlichen Textdaten ist ihre meist vollständig fehlende Datenstruktur. Nach allgemeinem Verständnis umfasst das Textmining daher „*eine Gruppe methodischer Ansätze, um Texte zu strukturieren und damit neue relevante Informationen zu extrahieren. Als Grundlage dienen statistische und musterbasierte Verfahren.*“ [HQW06][S. 4] Zwei grundlegende Ansätze sind hierbei zu unterscheiden:

1. *Information Retrieval (IR)* umfasst sämtliche Vorgänge, die zum erneuten Auffinden bereits zuvor bekannter Informationen eingesetzt werden [BYRN+99] [MRS08]. Hierzu gehören insbesondere Suchalgorithmen, mit denen Daten identifiziert und Zugriff auf sie erlangt werden kann.

2. *Information Extraction (IE)* beschreibt Analyse-Prozesse, die sich auf Data-Mining-Verfahren stützen. Ihre Intention ist es, unstrukturierten oder nur schwach strukturierten Daten relevante Informationen zu entnehmen. Je nach Spezialisierung können sie beispielsweise Textdaten oder auch digitale Bilder verarbeiten.

Die nachfolgenden Ausführungen konzentrieren sich auf Textdaten. Ihre Analyse wird insbesondere durch sprachliche Besonderheiten erschwert, die maschinell nicht einfach abzubilden sind. So kann beispielsweise ein Wort, je nach Kontext, eine andere Bedeutung haben (*Homonym*) oder unterschiedliche Wörter können semantisch äquivalent sein (*Synonym*). Eine weitere Besonderheit stellt die Aufteilung in Unter- (*Hyponym*) und Oberbegriffe (*Hyperonym*) dar, die bei einer qualifizierten Analyse von Textdaten ebenfalls berücksichtigt werden müssen. Auf den ersten Blick erscheint es daher so, als unterlägen Texte keinerlei Gesetzmäßigkeiten, da sie nahezu beliebig aufgebaut werden können. Dass dennoch, teils unscharfe, Strukturen erkannt werden können, zeigen zwei repräsentative Beispiele:

1. *Zipf'sches Gesetz*. Das Gesetz beschreibt eine inhaltliche Regularität in Textdaten und bildet damit ebenfalls eine Grundlage für Textmining-Verfahren. Der amerikanische Linguist Georg K. Zipf ordnete die Wörter eines natürlich-sprachlichen Textes absteigend nach der Häufigkeit ihres Auftretens (Rang: $1 \ldots n$) und konnte wiederkehrende Korrelationen identifizieren [Zip35] [Zip49]:
   - der Rang (r) eines Wortes, multipliziert mit dessen Häufigkeit (n) entspricht einer textabhängigen Konstante k mit $k \approx r * n$
   - die Häufigkeit (n) ist antiproportional zum Rang (r) und es gilt $n \sim \frac{1}{r}$.
2. *Formale Grammatiken* ermöglichen eine strukturelle Gliederung und beschreiben den Aufbau einer Sprache. Eine Grammatik ist als vier-Tupel (Φ, Σ, R, S) definiert und umfasst damit eine Menge von Variablen (Φ) sowie eine Menge von zu Φ disjunkten Terminalsymbolen (Σ). Ausgehend von einem Startsymbol (S) kann mit einer endlichen, nicht leeren Menge von Regeln (R) die Sprache $L(G)$ abgeleitet werden.

Es existieren zwei bekannte Grammatiktypen. Die *Konstituentengrammatiken* (= Phrasenstrukturgrammatik [PS94], [Mülo8]) versuchen Sätze in immer kleinere Einheiten (Konstituenten) zu zerlegen. Typisch für diesen Grammatik-Typ ist die anfängliche Ableitung des Startsymbols S in einen *Noun Phrase (NP)* sowie *Verb Phrase (VP)* (s. Abbildung 2.7). Ihr stehen die *Dependenzgrammatiken* gegenüber, bei denen stets das Verb das Wurzelelement eines Abbildungsbaumes bildet. Die Knoten des Baumes repräsentieren immer ein Terminalsymbol (Wort) und verweisen gleichzeitig auf höchstens eine Variable. Auf diese Weise wird eine strikte Mutter-Tochter-Relation (1:1) definiert, die immer die Abhängigkeit eines untergeordneten Wortes (*Dependens*) vom übergeordneten (*Regens*) beschreibt.

Die aufgeführten Gesetzmäßigkeiten reichen zum Aufbau einer leistungsfähigen Wissensextraktion jedoch nicht aus. Aktuelle Verfahren basieren daher fast immer auf probabilistischen Ansätzen, die mit Unschärfen arbeiten können. Als Grundlage dieser Verfahren kann die Automatentheorie angesehen werden, die einen engen Bezug zu formalen Grammatiken hat. Automaten unterschiedlicher Komplexität[6] können die von formalen Grammatiken beschriebenen Sprachen[7] ableiten. Ihre regelbasierten Abläufe (Übergangsfunktionen) werden zur Verarbeitung natürlicher Sprache mit Wahrscheinlichkeiten belegt, um den Unschärfen der natürlichen

[6] endlicher Automat, nicht deterministischer endlicher Automat, Kellerautomat, Turing-Maschine
[7] Chomsky-Hierarchie unterscheidet Regelsprachen in $Typ_{0\ldots3}$ [Cho56] [Cho59].

Sprache Rechnung zu tragen. Prominente Vertreter dieser Gattung sind *Conditional Random Fields (CRFs)* sowie *Hidden Markov Models (HMMs)*, auf die hier jedoch nicht im Detail eingegangen werden soll. Einen guten Überblick in das komplexe Themengebiet bieten [Rab89] und [LMP01] sowie [SP03]. Die wahrscheinlichkeitsbasierten Modelle unterstützen nahezu alle Verarbeitungsschritte eines Textmining-Prozesses:

- *Tokenisierung & Segmentierung* ermittelt Wortgrenzen (Tokenisierung) und gliedert den zu analysierenden Text in Sätze (Segmentierung).
- *Part-of-Speech-Tagging (POS)*. Es wird versucht, Wortformen im Text zu erkennen und durch aussagekräftige Kürzel zu markieren. Die Anzahl der verfügbaren Kürzel (Tagset) hängt von der zu analysierenden Sprache ab. Im Bereich der Bioinformatik ist *GENIA*[8] bekannt. Es ist auf die Analyse der englischsprachigen Texte in Medline (Abschnitt 2.1.5) spezialisiert.
- *Named-Entity-Recognition (NER)*. Der Analyseschritt zielt darauf ab, Terme (Named Entities) im Text zu kennzeichnen [STMA08], die bestimmte Entitäten kennzeichnen (z.B. Proteine, Enzyme oder Gene).
- *Parsing*. Mit Hilfe einer Grammatik wird ein Syntaxbaum aufgebaut und versucht die grundlegende Struktur des Textes zu erkennen.
- *Koreferenten-Resolution*. Koreferenten bezeichnen syntaktisch unterschiedliche Objekte im Text, die inhaltlich jedoch identisch sind. Das Ziel der Koreferenten-Auflösung im Text ist es, die Anzahl der Named Entities zu reduzieren.

Die Einführung zusätzlicher aber auch das Auslassen einzelner Phasen ist möglich. Häufig erfolgt außerdem ein *Stemming* sowie eine *Kookurrenz*-Analyse. Beim Stemming werden Wörter auf ihre Grundform reduziert, um Tempi und Casi vernachlässigen zu können. Die Kookurrenz-Analyse zielt darauf ab, Entitäten in Texten zu identifizieren, die häufig gemeinsam auftreten. Ihnen wird eine semantische Beziehung unterstellt. Aufgrund des probabilistischen Ansatzes erfordern die Ergebnisse einer Textmining-Analyse jedoch häufig eine nachträgliche Validierung. Zur qualifizierten Beurteilung verschiedener Textmining-Verfahren sind daher die Attribute *Precision* und *Recall* definiert worden. Während Recall die Wahrscheinlichkeit angibt, dass ein Algorithmus überhaupt ein Ergebnis liefert, kennzeichnet die Precision die Wahrscheinlichkeit für die Korrektheit eines gelieferten Ergebnisses. Ein theoretisch optimaler Algorithmus hätte demnach sowohl einen Recall als auch eine Precision von eins (1).

2.2.2 Methoden der Wissensrepräsentation

Elementare Voraussetzung für eine algorithmische Wissensverarbeitung ist eine Formalisierung (*Konzeptionalisierung*) des betrachteten Wissensbereichs (*Domäne* oder *Diskursuniversum*[9]). Die Konzeptionalisierung liefert eine abstrakte Sicht auf die zu modellierende Domäne,

[8] http://www.nactem.ac.uk/GENIA/tagger/

[9] *Universe of Discourse*. Begriff wurde von De Morgan und Bool geprägt [Boo54].

Concept	Slot	Filler
claudin-8	*is-a*	protein
	interacts	MUPP-1
	encoded-by-gene	CLDN8
	species	homo sapiens

Abbildung 2.6: Frames: konzeptionalisiertes Wissen zu Protein *claudin-8*

die alle relevanten Objekte (*Konzepte*) und deren Beziehungen (*Relationen*) untereinander beschreibt. Die Menge aller Konzepte wird zusätzlich noch einmal in *Individuen* und *Klassen* unterschieden [RN10]. Während Individuen ein konkretes Objekt beschreiben (z.B. „Protein claudin-8"), gruppiert eine Klasse zusammengehörige Individuen (z.B. „Proteine"). Darüber hinaus sind zwei grundsätzlich verschiedene Modellierungsansätze zu unterscheiden, die sich unmittelbar auf die Verarbeitung des modellierten Wissens auswirken. Während die *Closed World Assumption (CWA)* eine vollständige Modellierung des existierenden Wissens unterstellt, geht die *Open World Assumption (OWA)* nur von einer Teilmodellierung aus. Damit sind unter eine CWA alle nicht modellierten Zusammenhänge explizit falsch, während sie unter einer OWA durchaus wahr sein können.

Für die formale Repräsentation des konzeptionalisierten Wissens kann schließlich aus verschiedenen Wissensrepräsentations-Formalismen gewählt werden. Sie bieten unterschiedliche Ausdrucksstärken und offenbaren damit zugleich eine fundamentale Problematik der Wissensrepräsentation. Die Verfahren sollen einerseits möglichst ausdrucksstark, andererseits aber auch effizient berechenbar sein. Dies führt zwangsläufig zu einem Zielkonflikt, da eine gleichzeitige Optimierung beider Anforderungen nicht realisierbar ist [BL85]. Dies gilt damit auch für die drei bekanntesten Formalismen:

1. Frames (Schema)
2. Prädikatenlogik (Logik)
3. semantische Netze (Schema)

Frames sind ein Grundstein der formalen Wissensrepräsentation. Sie wurden bereits 1974 von Minsky entworfen, um konzeptionalisiertes Wissen darstellen zu können [Min74]. Ein Frame verfügt hierfür über eine beliebige Anzahl *Slots*, die eine Beschreibung der Eigenschaften des durch den Frame repräsentierten Konzepts ermöglichen (Abbildung 2.6). Eine Vererbungshierarchie erlaubt es dabei, die aktuellen Slotwerte (*Filler*) eines übergeordneten Frames an einen untergeordneten Frame weiterzureichen. Zusätzlich können die Slots Restriktionen (*Facetten*) unterliegen, die beispielsweise den Datentyp eines Slots einschränken. Möglich sind neben primitiven Datentypen auch Verweise auf andere Frames, um komplexere Sachverhalte abbilden zu können.

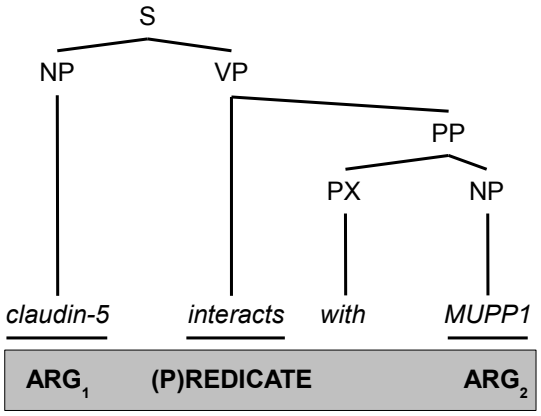


Abbildung 2.7: Mittels TM aus Medline Eintrag 12403818 extrahierte Prädikat-Argument-Struktur

2.2.2.1 Prädikatenlogik und Prädikat-Argument-Strukturen

Die Prädikatenlogik (first-order-logic) [Smu95] [Mat97] erweitert die Aussagenlogik um Quantoren ($\forall$, $\exists$). Das zentrale Element ist das *Prädikat*. Es repräsentiert die Eigenschaft oder die Beziehung (z.B. *interact*), die für ein bestimmtes Konzept gilt. Die Anzahl der Parameter (*Stelligkeit*), die einem Prädikat zugewiesen werden, um eine Aussage auf ihren Wahrheitsgehalt zu prüfen, ist dabei nicht begrenzt. Formell stellt sich ein Prädikat somit als n-stellige Funktion dar, die auf einen Wahrheitswert abbildet und für jede Belegung geprüft werden kann:

$$P(x_1, \ldots, x_n) \rightarrow \{\texttt{WAHR}, \texttt{FALSCH}\} \quad x_1, \ldots, x_n \in D \tag{2.2}$$

Die Parameter $x_1 \ldots x_n$ sind aus dem Diskursuniversum (D) wählbare Individuen. Das folgende, zweistellige Prädikat *interact* macht dies deutlich:

$$interact(claudin\text{–}8, MUPP\text{–}1) \tag{2.3}$$

Es ist mit den Individuen *claudin-8* und *MUPP-1* belegt und beschreibt eine Protein-Interaktion. Der Sachverhalt hätte auch mit dem spezifischeren, einstelligen Prädikat *interact_claudin-8* formuliert werden können. Die Flexibilität der Wissensdarstellung würde dies jedoch unnötig einschränken. Werden ausschließlich Individuen mit den Quantoren belegt, handelt es sich um die Prädikatenlogik erster Stufe. Noch mächtiger sind die Prädikatenlogiken höherer Stufen, die auch eine Qualifizierung der Prädikate erlauben. Die formelle Darstellung einer Aussage in der Prädikatenlogik hat eine große Ähnlichkeit zur linguistisch motivierten *Predicate-Argument-Structure (PAS)* [MKM+94] [SHWA03]. Häufig wird eine PAS von Textminingprozessen zur Darstellung der extrahierten Strukturen herangezogen (Abbildung 2.7)[10]. Im Gegensatz zur Prädikatenlogik dient die PAS allerdings maßgeblich zur Repräsentation der syntaktischen sowie semantischen Relationen eines Textes oder Satzes. Die Möglichkeit, die dargestellten Strukturen zusätzlich, wie in der Prädikatenlogik, zu quantifizieren, ist hier nicht gegeben.

[10] Genutzt wurde der TM-Algorithmus *Enju* (Abschnitt 3.1.3, http://www.nactem.ac.uk/enju/).

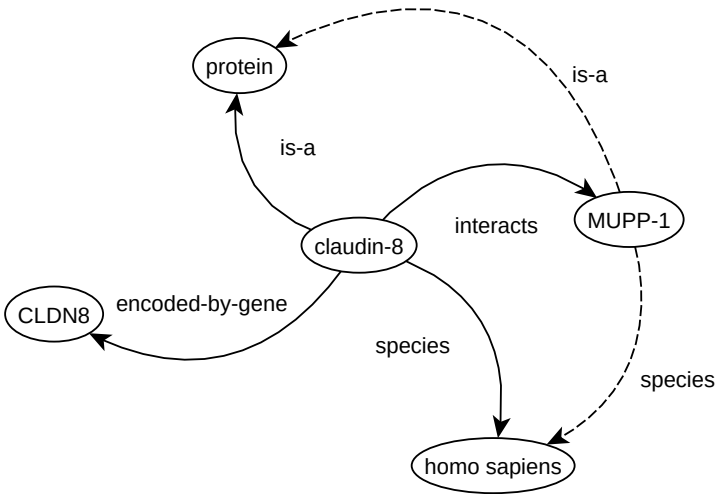


Abbildung 2.8: Protein-Wissen zu *claudin-8* im semantischen Netz mit Inferenz (gestrichelte Linie)

2.2.2.2 Semantische Netze und Semantic Web

Semantische Netze repräsentieren Wissen in Graphen (Abschnitt 2.1.4). Die Knoten repräsentieren Konzepte und die Kanten die zwischen ihnen existierenden Relationen. Abbildung 2.8 zeigt das zuvor in Frames repräsentierte Protein-Wissen zu *claudin-8* in einem semantischen Netz. Eine Möglichkeit, semantische Netze algorithmisch abzubilden, bieten die Konzepte des *Semantic Web*. Die Technologie geht auf einen Vorschlag des „Internet-Erfinders" Tim Berners-Lee zurück. Seine Intention war es, die im Internet rasant anwachsenden Informationen maschinenlesbar aufzubereiten [BL96] [BLHL+01]. Die wesentlichen Eigenschaften und Strukturen des Semantic Webs sind:

- *Ontologie*. Im Kontext des Semantic Web wird zusammenhängendes Wissen als Ontologie[11] bezeichnet. Es wird in einem *Repository* gespeichert, das konzeptionell einer graphbasierten Datenbank ähnelt[12].
- *Statements* sind Aussagen aus Subjekt, Prädikat und Objekt. Jedes Statement wird durch einen einfachen Graphen repräsentiert, dessen Knoten Subjekt und Objekt darstellen und durch das Prädikat (die Kante) verbunden sind.
- *Resource & Property* sind die formalen Bezeichnungen für die drei Komponenten eines Statements. Subjekt und Objekt entsprechen *Ressourcen*, das Prädikat dem *Property*. Jedes Property kann jedoch auch als Resource aufgefasst werden und damit wiederum von anderen Properties referenziert werden.
- *Uniform Resource Identifier (URI)*. Resources werden noch einmal in *Literale* und URIs unterschieden. Während Literale einen konkreten Wert repräsentieren, referenziert eine URI ein Konzept. URIs sind standardisiert und werden von einem *Namespace* (z.B.

[11] Begriff ist ursprünglich philosophisch geprägt („*Möglichkeiten und Bedingungen des Seienden*" [Heso2]).
[12] Der Formalismus graphbasierter Datenbanken ist gegenüber einem Repository restriktiver.

rdf, *rdfs*) eingeleitet. Für selbst definierte Resources ist dieser frei wählbar und reflektiert zumeist den Kontext der Applikation[13].

- *SPARQL Protocol And RDF Query Language (SPARQL)* ist eine Anfrage-Sprache, die konzeptionell der *Structured Query Language (SQL)* für relationale Datenbanken ähnelt [PS+08]. Gegenüber SQL arbeitet sie jedoch nicht auf Basis der relationalen Algebra, sondern stützt sich auf graphbasierte Anfrage-Muster.

- *Reasoner* ermöglichen Schlussfolgerungen (Inferenzen) im Repository. Die Reasoner nutzen hierfür die formalen Grundlagen der Sprachebenen des Semantic Webs aus, die sich auf Beschreibungslogiken abbilden lassen [HKRS08] [AH11] [Gei09].

Innerhalb des Semantic Web werden vier Sprachebenen mit wachsender Ausdrucksstärke unterschieden. Das *Resource Description Format (RDF)* ist die einfachste Sprachebene. *RDF-Schema (RDFS)* definiert die nächste Ausdrucksstufe und ermöglicht die Definition individueller Klassen mit denen Resources getypt werden können. Zusätzlich können Klassenhierarchien gebildet werden. Die *Web Ontology Language (OWL)* erweitert das Vokabular ein letztes Mal, so dass die Ausdrucksstärke an die Prädikatenlogik heranreicht [MVH+04]. Da diese selbst unentscheidbar ist, gibt es mit *OWL-Lite*, *OWL-DL* sowie *OWL-Full* Abstufungen. Von den verfügbaren Sprachmitteln in OWL ist das Property *sameAs* besonders hervorzuheben. Es drückt eine Äquivalenz zwischen zwei Resources aus und wird von Reasonern interpretiert. Derartige Eigenschaften können in OWL zudem jedem Property zugewiesen werden. Neben der Symmetrie sind auch funktionale oder transitive Relationen möglich. Die Leistungsfähigkeit der verfügbaren Reasoner unterscheidet sich jedoch zum Teil erheblich. Mit dem *Description Logic Implementation Group (DIG)*-Interface wurde daher ein Standard geschaffen, um verschiedene Reasoner in einem Repository nutzen zu können [BM03] [BLL+06].

2.2.3 Deduktive Datenbanken und Deduktion

Das Konzept deduktiver Datenbanken ist aus relationalen Datenbanken hervorgegangen. Sie erweitern diese um eine Logikkomponente. Im Gegensatz zu einer relationalen Datenbank werden in einer deduktiven Datenbank neben den eigentlichen Daten (Wissen) zusätzlich Regeln hinterlegt. Es erfolgt daher eine formelle Unterscheidung in *A-Box* (Wissen) und *T-Box* (Regeln). Die Regeln können mit den Daten in Verbindung gebracht werden und ermöglichen ein automatisiertes Schlussfolgern. Die Möglichkeit Wissen mit Regeln abzuleiten, stellt an ein deduktives Datenbanksystem jedoch besondere Anforderungen. Es muss sicherstellen, dass durch Inferenz gewonnenes Wissen zurückgezogen werden kann. Dies ist erforderlich, wenn Regeln nicht mehr gültig sind oder sich verändert haben. Nur so kann eine wahrheitsgetreue, konsistente Datenhaltung gewährleistet werden. Je nach Komplexität der Wissensbasis kann dies durchaus dazu führen, dass vollständige Baumstrukturen revidiert werden müssen. Verantwortlich für die komplexe Aufgabe ist ein *Truth Maintainance System (TMS)* [Doy79].

[13] Die von *FraMeTex* verarbeiteten Pathways nutzen den Namespace *http://knowledge/METEX (METEX)*.

2.2.3.1 Relationale Datenbanken

Relationale Datenbanken verfolgen eine mengenorientierte Datenverwaltung und definieren heute de facto den Standard. Aufgrund ihrer großen Akzeptanz existieren verschiedene, relationale Datenbanksysteme. Eine bekannte und frei verfügbare Datenbank ist MySQL[14]. Unter den kommerziellen Systemen zählen Oracle[15] und Sybase[16] momentan zu den Marktführern. Konzeptionell basieren alle Systeme auf dem von Codd eingeführten, relationalen Datenmodell [Cod70]. An das jeweils zugrundeliegende *Datenbankmanagement-System (DBMS)* werden zentrale Bedingungen geknüpft, die in den Codd'schen Regeln manifestiert sind [Cod82]. Die zu speichernden Daten sind in Tabellen (Relationen) organisiert, die jeweils mehrere Spalten (Attribute) umfassen. Die Zeile einer Tabelle entspricht einem Datensatz, der durch einen Primärschlüssel eindeutig identifiziert ist. Hierüber kann er von anderen Datensätzen, in anderen Tabellen referenziert werden (Fremdschlüssel). Der Grundgedanke hinter diesem Konzept ist es, eine möglichst redundanzfreie Speicherung der Daten zu erreichen. Der zugrundeliegende, konzeptionelle Vorgang wird als Normalisierung bezeichnet und folgt festen Regeln [SH00].

2.2.3.2 Deduktive Datenbanken

Deduktive Datenbanken verschmelzen das relationale Datenbankkonzept mit der Logikprogrammierung. Die Grundlage deduktiver Datenbanken wurde im Jahr 1968 durch eine Arbeit von Cordell Green und Betram Raphael [GR68] gelegt. Sie erkannten das Potential, dass sich durch die Anwendung von Logikverfahren auf die in Datenbanken gespeicherten Daten ergeben konnte. Publik wurde das deduktive Datenbankkonzept jedoch erst im Jahr 1977 durch einen gemeinsamen Workshop von Hervé Gallaire und Jack Mincker [Min14]. Die Ergebnisse wurden in [GM78] veröffentlicht und machten die Forschung auf das neue Konzept aufmerksam. In den folgenden Jahren wurde schließlich eine formale Grundlage für deduktive Datenbanken geschaffen [GMN84] [LT85] [LT86] [GM89]. Für die Formulierung der Logik wurde zunächst auf die spezielle Programmiersprache *Programmation en Logique (PROLOG)* zurückgegriffen [CM10]. Sie wurde bereits 1972 entwickelt [CR93].

Das Konzept deduktiver Datenbanken warf jedoch auch neue Probleme auf. Sie betrafen insbesondere die Anwendung logischer Regeln (Resolution) auf die großen Datenmengen in einer Datenbank. Einerseits verarbeitet PROLOG sämtliche Daten im Primärspeicher, andererseits kann die Terminierung eines Programms nicht garantiert werden [RU95]. Aus diesem Grund wurde mit DATALOG eine spezielle Variante von PROLOG entwickelt und an die Bedürfnisse deduktiver Datenbanken angepasst. Die Realisierung der bis dato theoretischen Konzepte wurde jedoch erst nach 1980 in den Fokus gerückt. Sie wurde durch das japanische *Fifth Generation Computer Systems project (FGCS)* zusätzlich motiviert [FM83]. Das 1982 gestartete Projekt

[14] http://www.mysql.de
[15] http://www.oracle.com/de
[16] http://www.sybase.de

fand weltweite Beachtung und zielte auch darauf ab, automatisiertes Schlussfolgern in der Wissensverarbeitung zu ermöglichen. Aufgrund der Anstrengungen wurden in der Vergangenheit vielfältige Prototypen eines deduktiven Datenbanksystems entwickelt:

- *LDL & LDL++* [CGK+90] [Zan96]
- *Glue-Nail* [PDR91]
- *LOLA* & *O!-LOLA* [ZF97] [Spe98]
- *Aditi* [VRK+94]
- *XSB*[17] [SSW94] [RSS+97] [SW12]
- *CORAL* [RSSS93]
- *BDDBDDB* [WACL05]
- *DO2* [LWL98]
- *SOLAR* [NIIR10]

Darüber hinaus existieren noch einige, weniger bekannte Prototypen [RH94]. Sie weisen wie fast alle deduktiven Datenbanken eine spezielle *Compiler*-Komponente auf. Sie transformiert ein logisches Programm in relationale Algebra, die vom zugrundeliegenden Datenbanksystem zur Laufzeit ausgeführt wird. Bemerkenswert ist, dass mit Hilfe von *CORAL* bereits im Jahr 1996 eine *Pathway Prediction* in KEGG konzipiert wurde [FGK96]. Die Grundzüge dieses Verfahrens finden noch heute in *PathPred* Anwendung und werden daher im Rahmen der verwandten Arbeiten noch ausführlicher präsentiert. Von den aufgeführten Systemen wird aktuell nur noch *XSB* weiterentwickelt. Es bietet umfangreiche Schnittstellen, von denen die *Open Database Connectivity (ODBC)* hervorzuheben ist. Sie agiert als Compiler und konvertiert die DATALOG Ausdrücke automatisch in SQL. Mit Ausnahme von *LDL++* und *O!-LOLA* kam es ansonsten zu keiner Weiterentwicklung der verschiedenen Prototypen. Während *LDL++* auf eine bessere Integration mit externen Datenbanken abzielte, bot *O!-LOLA* eine objektorientierte Erweiterung des Ursprungssystems. *DO2* zielte von Anfang an auf eine objektorientierte Speicherung der Daten ab.

Trotz ihrer unbestrittenen Vorteile führen deduktive Datenbanken bis heute ein Schattendasein. Neben *XSB* sind keine aktuellen oder gar kommerziellen Systeme verfügbar. Ihre grundlegende Idee, Wissen mit Regeln anzureichern und Schlussfolgerungen zu unterstützen, findet sich jedoch in den alternativen Konzepten des Semantic Web wieder (Abschnitt 2.2.2.2). Die Überschneidungen der beiden Konzepte wurden bereits diskutiert [MGFS11] [KRS15].

[17] http://xsb.sourceforge.net/

2.2.3.3 Logisches Schließen

Die Schlussfolgerungen (Inferenzen) in deduktiven Datenbanken basieren auf formaler Logik. Nachfolgend wird ein kurzer Einblick gegeben und elementare Konzepte vorgestellt. Einen detaillierten Einblick bieten [GN87] [RN10] [Gen10]. Ein Grundstein für automatisierte Schlussfolgerungen ist die Deduktion. Aus Voraussetzungen (Prämissen) kann ein logischer Schluss (Konklusion) gezogen werden. Die formale Grundlage bildet der *modus ponens*.

$$\frac{\begin{array}{l} A \\ A \rightarrow B \end{array}}{B} \tag{2.4}$$

Unter der Voraussetzung das A wahr ist, kann mit der Regel $A \rightarrow B$ auf B geschlossen werden. Die exemplarische Anwendung des *modus ponens* in einem Protein-Interaktionsnetzwerk (Abschnitt 2.1.4.4) zeigt Abbildung 2.8. In dem semantischen Netz können mit seiner Hilfe für den Knoten *MUPP-1* zwei zusätzliche Relationen abgeleitet werden. Sie sind gegenüber den anderen Relationen gestrichelt dargestellt. Unter der Annahme, dass eine Interaktion nur zwischen Proteinen erfolgen kann und *claudin-8* bereits als Protein bekannt ist (Prämissen), muss auch *MUPP-1* ein Protein sein (Konklusion). Zusätzlich ist ein weiterer Schluss auf die Spezies von *MUPP-1* möglich. Da *claudin-8* bereits der Spezies *homo sapiens* zugeordnet wurde und eine Interaktion nur innerhalb eines Organismus erfolgen kann (Prämissen), muss auch das Protein *MUPP-1* mit der Spezies *homo sapiens* in Verbindung stehen (Konklusion). Die konkrete Formulierung der Regeln ist technisch bedingt und hängt von den eingesetzten Wissensverarbeitungsmethoden bzw. Frameworks ab (z.B. Semantic Web).

Neben dem *modus ponens* existieren weitere deduktive Schlussformeln. Von besonderer Bedeutung ist der *modus barbara*, der einen Kettenschluss ermöglicht. Er kann damit eine Transitivität zum Ausdruck bringen, die insbesondere für die Vorhersage metabolischer Netzwerke (Abschnitt 2.1.4.1) interessant ist.

$$\frac{\begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array}}{A \rightarrow C} \tag{2.5}$$

Die Prämissen des *modus barbara* sind zwei Regeln, die in metabolischen Netzwerken als enzymatische Reaktionen (Definition 2.1) aufgefasst werden können. Die erste Reaktion ($A \rightarrow B$) überführt das Substrat A in das Produkt B. Die zweite Reaktion ($B \rightarrow C$) das Substrat B wiederum in das Produkt C (siehe 2.5). Die Anwendung des *modus barbara* erlaubt es schließlich, die einzelnen Reaktionen zu verketten und einen Pathway $A \rightarrow C$ abzuleiten. Inferenzen beschränken sich jedoch nicht nur auf die Deduktion. Ein besonderer Schluss ist die Abduktion, die im Gegensatz zur Deduktion von der Konklusion auf die Prämisse schließt. Da dies jedoch keineswegs zu wahrem Wissen führen muss, wird die Abduktion häufig zur Hypothesengenerierung genutzt. Anhand der abstrakten, enzymatischen Reaktionen aus 2.5 ist dies leicht ersichtlich. Liegt das Produkt B vor, *kann* es durch die enzymatische Reaktion $A \rightarrow B$ erzeugt worden sein.

Dies ist jedoch nur eine Möglichkeit. Es ist durchaus denkbar, dass weitere Pathways innerhalb eines metabolischen Netzwerks existieren, die ebenfalls zum Produkt *B* führen könnten.

Unabhängig von Deduktion und Abduktion sind maschinelle Inferenzen jedoch nur möglich, wenn der Reasoner die formulierten Regeln korrekt interpretieren kann. In welcher Reihenfolge die Regeln angewendet werden, hängt dabei von der verfolgten Inferenzstrategie des Reasoners ab. Ausgehend vom Startpunkt (Faktum) wählt die Vorwärtsverkettung (forward chaining) die Regeln anhand zutreffender Prämissen aus. In den folgenden Schritten werden dann nur diejenigen Regeln betrachtet, deren Prämissen der Konklusion der zuvor angewendeten Regel entsprechen. Aufgrund dieses *top-down* Ansatzes wird die Vorwärtsverkettung auch als datengetriebene Inferenz bezeichnet. Im Gegensatz dazu verfolgt die Rückwärtsverkettung (backward chaining) einen *bottom-up* Ansatz. Sie geht vom zu zeigenden Ziel aus und wählt die Regeln anhand ihrer Konklusionen aus. Die Strategie entspricht damit einer zielorientierten Inferenz.

2.2.4 Indexbasierte Filterung von Textdaten

Die Auswahl aussagekräftiger Eingabedaten ist für die Vorhersage biologischer Netzwerke aus Textdaten entscheidend. Sie ist daher auch für die eingangs motivierte *Pathway Prediction* auf Basis des stetig wachsenden Medline-Datenbestand (Abschnitt 2.1.5) unumgänglich. Anhand von Schlagwörtern (z.B. Proteine, Enzyme, Gene) sollten relevante Textdaten effizient ermittelt werden können. Aus diesem Grund werden Textdaten häufig indiziert, um auch in großen Datenmengen unkompliziert suchen zu können. In den letzten Jahren hat sich hierfür die *Lucene API*[18] von Apache immer wieder bewährt. Sie definiert daher mittlerweile faktisch den Standard für komplexe Indexstrukturen und Suchoperationen [OG05]. Die nachfolgenden Ausführungen beziehen sich auf die Lucene Version 2.2.0.

Lucene gliedert sich in zwei Hauptkomponenten. Der *IndexWriter* verantwortet den Aufbau sowie die Aktualisierung der Indexstrukturen, während der *IndexSearcher* für sämtliche Suchanfragen zuständig ist. Beim Aufbau einer Indexstruktur werden stets *key/value*-Paare indiziert, die jeweils als *Field* repräsentiert werden. Ein indizierter Datensatz wird damit durch mehrere Fields dargestellt, die in einem *Document* zusammengefasst werden. Der *key* gibt dem Field einen Namen, der *value* repräsentiert die zugehörigen, indizierten Textdaten. In einer Suchanfrage werden daher neben dem Suchbegriff immer auch die zu durchsuchenden Felder benannt. Auf das Verhalten der Indizierung jedes einzelnen Feldes kann individuell eingewirkt werden. Es kann festgelegt werden, ob:

- ein Feld überhaupt indiziert wird
- die Textdaten eines Feldes im Index hinterlegt werden
- eine Häufigkeitsverteilung einzelner Textterme verfügbar ist

[18] http://lucene.apache.org

Für eine erfolgreiche Suche ist zumindest das Ablegen der Schlüsselattribute eines indizierten Datensatzes im Index unerlässlich. Würde dies nicht erfolgen, könnte Lucene lediglich Treffer melden. Die damit verknüpften Datensätze könnten jedoch im Anschluss nicht identifiziert werden. Die optionale Häufigkeitsverteilung ermöglicht eine Ähnlichkeitssuche, die auf einfachen Verfahren der Musterklassifikation basiert. Zum Aufbau einer Indexstruktur können die verfügbaren Optionen beliebig kombiniert werden. Zusätzlich besteht die Möglichkeit mit *Analyzern* weiteren Einfluss auf den Indizierungs-Vorgang zu nehmen. Ein Analyzer kann die Textdaten vor ihrer Indizierung noch einmal explizit vorverarbeiten und beispielsweise in Segmente zerlegen. Dies kann zu einer weiteren Verbesserung der Suchergebnisse führen. Wichtig ist jedoch, dass auch die zur späteren Suche herangezogenen Schlagwörter zunächst mit dem identischen Analyzer vorverarbeitet werden. Nur so kann sichergestellt werden, dass korrekte Einträge zu den Suchbegriffen im Index gefunden werden.

Bevor mit dem Index jedoch gearbeitet werden kann, muss er zunächst in einer separaten Vorverarbeitung aufgebaut werden. Im Regelfall wird er von Lucene im Dateisystem abgelegt. Parallele, lesende Zugriffe werden unterstützt, so dass mehrere Suchanfragen unabhängig von einander ausgeführt werden können. Alternativ besteht aber auch die Möglichkeit, den Index in Datenbanken zu persistieren. Die zusätzliche Kommunikation mit der Datenbank bringt allerdings unnötige Verzögerungen mit sich. Die Suchergebnisse können dann nicht mehr binnen weniger Millisekunden präsentiert werden.

2.2.5 Service-Orientierte-Architektur

Das Ziel der in Abschnitt 1.2 motivierten *Pathway Prediction* ist es, die Forschung in den Lebenswissensschaften zu unterstützen. Sie nutzt die rekonstruierten, biologischen Netzwerke, um interessante Vorhersagen experimentell zu prüfen. Damit dies gelingen kann, muss das System für Biologen und Mediziner unkompliziert nutzbar sein. Ein tiefgreifendes technisches Verständnis für das zugrundeliegende System darf daher keine Voraussetzung für dessen Einsatz sein. Ein Konzept, dass auf eine Abstraktion von der technischen Realisierung abzielt, ist die *Service-Orientierte-Architekur (SOA)*. Sie gliedert *ein* Gesamtsystem in funktionale Komponenten (Services), die unabhängig von einander realisiert werden können. Auch wenn im Rahmen der vorliegenden Arbeit zunächst nur eine prototypische Implementierung realisiert wird, müssen die konzeptionellen Grundlagen der SOA bereits jetzt Berücksichtigung finden. Die SOA ist eng mit der *Enterprise Application Integration (EAI)* verbunden, die es sich zum Ziel setzt, *mehrere*, zumeist heterogene IT-Systeme zu einem logischen System zu verknüpfen.

Jeder Service einer SOA bildet eine Teil-Anforderung des Gesamtsystems ab und kann über einheitliche Schnittstellen angesprochen werden. Dem allgemeinen Verständnis entsprechend sollten die Schnittstellen als Webservice ausgelegt sein. Durch eine entsprechende Verkettung (*Orchestrierung*) unterschiedlicher Services wird dann der Funktionsumfang des konzipierten Gesamtsystems (*Workflow*) gebildet. Im Idealfall weisen die beteiligten Services einen derart hohen Abstraktionsgrad auf, dass sie in verschiedenen Workflows genutzt werden können. Die individuelle Orchestrierung der Services wird bereits von leistungsfähigen Werkzeugen unterstützt. Mit ihrer Hilfe kann eine visuelle Definition der Workflows erfolgen. Im Bereich der

Bioinformatik hat sich Taverna[19] hierfür etabliert [OAF+04]. Da Forschende in den Lebenswissensschaften mit Taverna bereits vertraut sind, haben sie hierüber Zugriff auf sämtliche Funktionalitäten, die dem SOA-Konzept unterliegen. Auch die in dieser Arbeit motivierte Netzwerkrekonstruktion stünde ihnen damit unmittelbar zur Verfügung, wenn sie als SOA konzipiert wird.

Die eingangs motivierte *Pathway Prediction* ist prädestiniert dafür, auf Basis einer SOA realisiert zu werden. Einerseits lässt der grundsätzliche Datenfluss (Abbildung 1.1) bereits funktionale Komponenten erahnen, andererseits erscheint der Einsatz einzelner Services auch außerhalb der Rekonstruktion biologischer Netzwerke sinnvoll. Dies gilt insbesondere für die erforderliche Filterung relevanter Textdaten. Bereits jetzt sind Projekte im Bereich der Lokalisation (sub)zellulärer Komponenten an dieser Funktionalität interessiert. Sie könnten die implementierten Services ebenfalls nutzen und damit wertvolle Entwicklungszeit sparen.

2.3 Zusammenfassung

Dieses Kapitel legte elementare Grundlagen, die ihr Verständnis erleichtern. Zunächst erfolgte ein Einstieg in die biologischen Grundlagen. Sie zielten darauf ab, die in biologischen Netzwerken repräsentierten Zusammenhänge besser verstehen zu können. Hierfür wurde zunächst die grundlegende Struktur einer Zelle umrissen und ihre wichtigsten Komponenten wurden hervorgehoben. Es zeigte sich, dass insbesondere Proteine entscheidenden Einfluss auf die Funktionen einer Zelle haben. Sie sind an fast allen intra- sowie extrazellulären Prozessen beteiligt. Mit vier verschiedenen, biologischen Netzwerken wird versucht, diese Zellprozesse zu formalisieren. Die besonderen Eigenschaften der vier Netzwerke wurden daher ebenfalls kurz hervorgehoben. In diesem Zusammenhang wurde auch auf die Möglichkeiten der Rekonstruktion biologischer Netze eingegangen. Mit Methoden der Informatik wird hierbei versucht, die Netzwerke auf Basis verfügbarer Daten vorherzusagen. Im Anschluss erfolgte mit der Vorstellung von Medline der Übergang zu den technischen Grundlagen. In Medline werden sämtliche Veröffentlichungen aus dem Bereich der Lebenswissenschaften zusammengefasst. Die umfangreichen, textbasierten Daten können zur Vorhersage biologischer Netzwerke aus Textdaten herangezogen werden.

In der Informatik wurden zur automatischen Wissensextraktion aus Textdaten spezielle Algorithmen entwickelt. Die Grundzüge dieser Textmining-Verfahren wurden zu Beginn der technischen Grundlagen vorgestellt. Die formale Repräsentation der extrahierten Information stand anschließend im Mittelpunkt. Es existieren verschiedene Repräsentationsformalismen, von denen die Prädikatenlogik sowie semantische Netze exemplarisch herausgegriffen wurden. Anschließend wurde mit der Semantic Web Technologie eine Möglichkeit vorgestellt, Daten in semantischen Netzen zu verwalten. Die Technologie bietet einen ähnlichen Leistungsumfang wie deduktive Datenbanken, auf deren Eigenschaften daher ebenfalls ausführlich eingegangen

[19] http://www.taverna.org.uk

wurde. Obwohl die Konzepte seit langem bekannt sind, existieren derartige Systeme mit wenigen Ausnahmen (XSB) nur als Prototypen. Aus konzeptioneller Sicht erweitern sie relationale Datenbanken um eine Logikkomponente, mit deren Hilfe sowie zusätzlicher Regeln neue Zusammenhänge abgeleitet werden können. Die dem Schlussfolgern zugrundeliegende Deduktion wurde in diesem Kapitel daher ebenso beleuchtet. Die Vorhersage biologischer Netzwerke aus Textdaten erfordert darüber hinaus eine leistungsfähige Filterung der zu analysierenden Textdaten. Dies geschieht mit Hilfe von Indexstrukturen, auf die im weiteren Verlauf ebenfalls eingegangen wurde. Zum Ende dieses Kapitels wurde mit der SOA schließlich eine spezielle Software-Architektur präsentiert, die von den technischen Details einer System-Implementierung abstrahiert. Dies soll es Biologen sowie Medizinern erleichtern, mit dem in dieser Arbeit konzipierten Prototypen effektiv arbeiten zu können.

3 Verwandte Arbeiten

In diesem Kapitel werden ausgewählte Arbeiten mit Bezug zur motivierten *Pathway Prediction* (Abschnitt 1.2) vorgestellt. Im Abschnitt 3.1 wird zunächst auf verschiedene TM-Ressourcen eingegangen. Ihr Einsatz zur angestrebten Rekonstruktion verschiedener sowie möglichst präziser Netzwerke aus Textdaten ist denkbar (Anforderung 2 & 3, Abschnitt 4.1). Im Anschluss werden Systeme präsentiert, die auf die Rekonstruktion komplexer Netzwerke spezialisiert sind. Zur Vorhersage weiterer Pathways könnten sie mit der motivierten Deduktionskomponente verknüpft werden (Anforderung 4). Etablierte Frameworks die den Aufbau einer logikunterstützten Datenbank ermöglichen stehen im Abschnitt 3.3 im Fokus. Dies schließt die Präsentation einiger, prominenter Wissensressourcen ein, die potentielles Regelwissen für ein Schlussfolgern in biologischen Netzwerken bieten.

3.1 Analyse biomedizinischer Textdaten

In der Bioinformatik existieren unzählige Textmining-Algorithmen [KEV+05] [KMS+08]. Sie können hilfreiche Informationen und damit einzelne Bausteine zur Vorhersage biologischer Netzwerke liefern [HKA+05]. Einige TM-Algorithmen wurden zu Expertensystemen ausgebaut und ihre Wissensextraktion zusätzlich mit biologischen Ontologien unterstützt[1]. Die verfügbaren Algorithmen versucht das *National Centre for Textmining (NaCTeM)*[2] zu bündeln. Es ist allerdings unmöglich, eine vollständige Auflistung zu bieten. Mit den in diesen Abschnitt präsentierten Arbeiten soll dennoch ein Eindruck vom verfügbaren Spektrum gewonnen werden. Die Algorithmen basieren zumeist auf probabilistischen Verfahren, die mit dem GENIA-Textkorpus[3] trainiert wurden. Der Korpus enthält ausgewählte Medline-Abstracts, die anhand der MeSH-Terme *human*, *blood cell* und *transcription factor* identifiziert wurden. Die selektierten Abstracts wurden segmentiert (Abschnitt 2.2.1) und von biomedizinischen Experten annotiert [OTK02] [KOTT03].

Auffallend ist die große Anzahl derjenigen Algorithmen, die auf die Protein-Interaktionen spezialisiert sind (Abschnitt 2.1.4.4). Von den existierenden Systemen heben sich daher die Wenigen mit einem Fokus auf Signalwege (*GeneWays*[4] [RIK+04]) oder enzymatische Reaktionen besonders ab. Für die motivierte *Pathway Prediction* kommen viele der verfügbaren Systeme

[1] *Biological Information Extraction and Query Answering (BIEQA)* [AD07]
[2] http://www.nactem.ac.uk/
[3] http://www.nactem.ac.uk/meta-knowledge/Meta-knowledge_GENIA_corpus.zip
[4] Tool ist aktuell nicht mehr verfügbar.

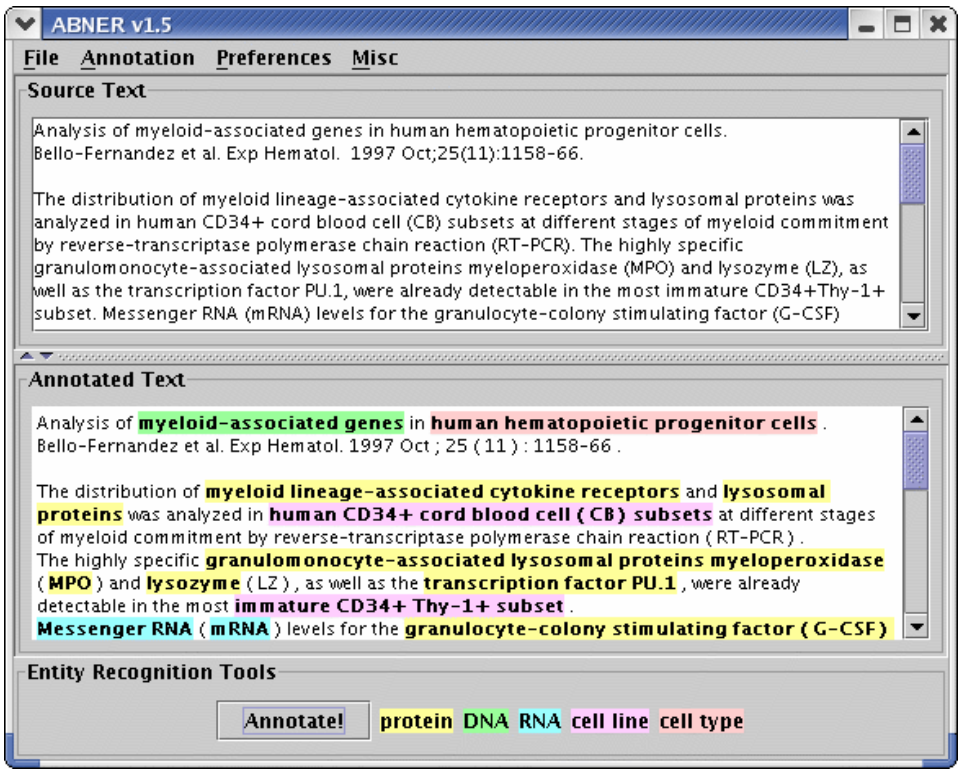


Abbildung 3.1: Graphische Benutzeroberfläche von ABNER zur Named Entity Recognition [Seto5]

allerdings nicht in Frage. Sie bieten neben einer webbasierten *Graphical User Interface (GUI)* keinerlei Schnittstellen und können damit nicht in einen komplexen Analyseprozess eingebettet werden. Aufgrund ihres Leistungsumfangs sollen mit *pCorral*[5] [LJYA+13] und *Chilibot*[6] [CS04] jedoch zumindest zwei dieser Systeme an dieser Stelle erwähnt werden.

3.1.1 ABNER

ABNER[7] zeichnet sich durch die Identifikation von *Named Entities* (Abschnitt 2.2.1) in molekular-biologischen Texten aus. Das System versucht Proteine, DNS- und RNS-Terme sowie zellspezifische Eigenschaften zu erkennen [Seto5]. Auf dem Weg zur Vorhersage biologischer Netzwerke liefert das in Java geschriebene System damit einen kleinen Baustein. Im Zusammenspiel mit weiteren Systemen besteht jedoch die Möglichkeit ein komplexes, biologisches Netzwerk vorherzusagen. Technisch baut der NER-Prozess in diesem Tool auf einem CRF auf (Abschnitt 2.2.1). Auch wenn die aktuellste Version aus dem Jahr 2005 stammt, entsprechen die umgesetzten Konzepte immer noch dem momentanen Stand der Technik und finden in nahezu unveränderter Art und Weise auch heute noch Anwendung. Zum Starten und Ausführen eines

[5] http://www.ebi.ac.uk/Rebholz-srv/pcorral
[6] http://www.chilibot.net/
[7] http://pages.cs.wisc.edu/ bsettles/abner

Corpus	Entity	R	P	F_1	$(S - F_1)$
NLPBA	Overall	72.0	69.1	70.5	(82.0)
	protein	77.8	68.1	72.6	(84.9)
	dna	63.1	67.2	65.1	(76.1)
	rna	61.9	61.3	61.6	(78.5)
	cell line	58.2	53.9	56.0	(68.2)
	cell type	65.6	79.8	72.0	(82.1)
BioCreative	protein/gene	65.9	74.5	69.9	(83.7)

Tabelle 3.1: Identifikation biomedizinischer Terme mit ABNER (Evaluation) [Seto5]

Wissensextraktions-Vorgang bietet ABNER zwei verschiedene Möglichkeiten. Im einfachsten Fall kommt hierfür eine vom System bereitgestellte, grafische Oberfläche zum Einsatz, mit der die zu analysierenden Textdaten ausgewählt und das Erkennen der Named Entities angestoßen werden kann. Die grafische Benutzeroberfläche ist in Abbildung 3.1 dargestellt und zeigt die Anwendung nach Abschluss einer Textanalyse. Die vom Algorithmus im Eingabetext aufgedeckten Named Entities sind farblich hervorgehoben und erlauben über die jeweilige Farbe auch einen Rückschluss auf den detektierten Objekttyp. Im gewählten Beispiel sind die meisten der erkannten Objekte gelb hinterlegt und deuten so auf potentielle Proteine hin.

Neben dem GUI bietet das System ein *Application Programming Interface (API)*, über die der Analyseprozess ebenfalls gesteuert werden kann. Damit kann das Tool unkompliziert in einen komplexen Prozess zur Vorhersage biologischer Netzwerke herangezogen werden. Außerdem kann hierüber zusätzlich Einfluss auf das CRF genommen und das Modell auch mit einem benutzerdefinierten Korpus trainiert werden. Zur Beurteilung der Analyse-Ergebnisse wurden von den Entwicklern zwei unterschiedliche Korpora herangezogen. Jeder der beiden Korpora enthält neben annotierten Trainingsdaten zusätzlich mehrere tausend Sätze, die im Anschluss an die Trainingsphase zur Evaluation des trainierten ABNER-Algorithmus genutzt wurden. In einem ersten Schritt wurde mit dem *Natural Language Processing in Biomedical Applications (NLPBA)*-Korpus eine modifizierte Variante des GENIA-Korpus verwendet (Abschnitt 3.1). In den etwas mehr als 18.000 Sätzen dieses Korpus sind fünf unterschiedliche Entitäten annotiert. Nicht ganz so umfangreich ist der zweite Test-Korpus (BioCreative [KEV+05]) mit dem die anfängliche erzielten Ergebnisse verglichen wurden. Er bietet lediglich 7.500 annotierte Trainingssätze, die sich zudem auf die Annotation einer einzigen Entität beschränken. Sie kennzeichnet Gene- und Genprodukte im Korpus.

Die Tabelle 3.1 stellt die bei dem Vergleich erzielten Parameter Recall (R) und Precision (P), in Abhängigkeit der unterschiedlichen Named Entities, übersichtlich gegenüber und gibt mit dem F_1 (-Score) auch noch die Genauigkeit der Ergebnisse an. Hierfür werden R und P gemäß des harmonischen Mittelwerts

$$F_1 = 2 * \frac{P * R}{P + R} \tag{3.1}$$

Abbildung 3.2: Web-Oberfläche des zentralen *Whatizit*-Wissensextraktionsprozess

in Relation gesetzt und mit $(S - F_1)$ auch noch ein zusätzliches, fehlertolerantes Maß angegeben. An den abgedruckten Daten lässt sich erkennen, dass ABNER sich besonders zum Erkennen von Proteinen eignet und Schwächen bei Zelllinien offenbart.

3.1.2 Whatizit

Whatizit[8] wurde am *European Bioinformatics Institute (EBI)* entwickelt und bietet gegenüber ABNER ein größeres Leistungsspektrum. Das System stellt einen zentralen Wissensextraktionsservice zur Verfügung, der nicht nur einen bestimmten Teilaspekt (z.B. NER) berücksichtigt [RSAG+08]. Der weiteren Entwicklung individueller Wissensextraktions-Algorithmen soll damit entgegen gewirkt werden. Für eine Vorhersage biologischer Netzwerke müssten dann nicht mehr mehrere System genutzt und die jeweiligen Teilergebnisse aufwändig zusammengeführt werden. Analog zu ABNER können auch bei Whatizit die zu analysierenden Texte über eine GUI (Abbildung 3.2) eingegeben werden. Für eine automatisierte Analyse steht aber auch ein Webservice zur Verfügung. Für Medline-Abstracts genügt sogar die Angabe ihrer eindeutigen *PubMed Identifier (PMID)*, so dass eine Übermittlung der zu analysierenden Daten in diesem Fall entfallen kann. Sobald Whatizit die Textdaten zur Verfügung stehen, übernimmt der zentrale, serverbasierte Prozess die weitere Verarbeitung. Er liefert die Ergebnisse wahlweise über den Webservice zurück oder stellt sie in der Web-Oberfläche dar.

Bereits im Abschnitt 2.2.1 wurde deutlich, dass eine automatisierte Wissensextraktion sich aus mehreren Teilschritten zusammensetzt. Dieser Tatsache trägt Whatizit durch ein eigenes Modulsystem Rechnung. Jedes der in Whatizit verfügbaren Module ist in Java geschrieben und für einen spezifischen Analyse-Schritt zuständig. So existiert beispielsweise auch ein ABNER-

[8] http://www.ebi.ac.uk/webservices/whatizit/info.jsf

Modul (siehe Abschnitt 3.1.1), mit dem die Fähigkeiten des bereits zuvor vorgestellten Textmining-Prozesses genutzt werden können. Das Systemkonzept erlaubt es zudem, die verfügbaren Module unabhängig voneinander zu nutzen oder bei Bedarf auch zu einer komplexen Pipeline zusammenzufassen. Die Module sind darüber hinaus in der Lage, die analysierten Texte mit Inhalten anderer Datenbanken zu verknüpfen. Unterstützt wird beispielsweise *CiteXplore*[9], um den Anwender auf korrespondierende Literatur hinzuweisen.

In Abhängigkeit der verwendeten Benutzerschnittstelle müssen sämtliche Module, die in einen Analyseprozess einbezogen werden sollen, vor dem Start benannt werden. In der Web-Oberfläche erfolgt die Selektion über eine Drop-Down-List, wohingegen der Webservice-Schnittstelle hierfür entsprechende Parameter übergeben werden müssen. Die ausgewählten Module bilden die sogenannte „*information extraction pipeline*“, die beliebig aufgebaut werden kann. Eine Liste aller verfügbaren Module und ihrer jeweiligen Fähigkeiten kann der Whatizit-Website entnommen werden. Sehr bekannt und oft genutzt, sind die Module *whatizitDisease* zum Erkennen von Krankheiten sowie *whatizitGO*. Letzteres wurde zum Auffinden von Genen und Genprodukten geschaffen und bedient sich hierfür der bekannten *Gene Ontology (GO)* (Abschnitt 3.3.4.2). Die Vision von Whatizit ist es, mit dem konzipierten, serverbasierten Modulsystem einen einheitlichen Datenaustausch zwischen allen verfügbaren Textmining-Systemen der Bioinformatik zu schaffen.

3.1.3 Enju

Das Textmining-System Enju[10] wurde an der University of Tokyo am Department of Computer Science entwickelt. Es ist eines der wenigen Systeme, das die zu analysierenden Textdaten sowohl einer syntaktischen als auch einer semantischen Analyse unterzieht [MT08]. Zur erfolgreichen Rekonstruktion biologischer Netzwerke reichen die Informationen jedoch nicht aus. In Kombination mit einem auf NER spezialisiertem System ist aber beispielsweise bereits die Rekonstruktion erster, biologischer Netzwerke denkbar.

Die mit Enju erzielbaren Ergebnisse werden vielfach aufgrund ihrer Qualität herausgehoben. Im Rahmen einer Evaluation konnte ein Recall- und Precision- Wert von 0,8 (Precision) bzw. 0,7 (Recall) erreicht werden [MSS+08]. Enju verfolgt einen probabilistischen Ansatz und wurde ebenfalls mit dem Genia-Korpus trainiert. Die Ergebnisse können optional in einer XML-Struktur kodiert werden. Die XML-Struktur repräsentiert sowohl den syntaktischen Aufbau der analysierten Texte als auch die extrahierten semantischen Zusammenhänge [Miy08] in Form von PASs. Die hierarchisch organisierten Analyseergebnisse in Abbildung 3.3 lassen den Charakter des in Enju realisierten Algorithmus bereits erahnen. Der Algorithmus basiert auf einer Phrasenstrukturgrammatik (Abschnitt 2.2.1), die zum Parsen der Textdaten herangezogen wird. Jeder Satz des Eingabetextes wird schrittweise in Konstituenten gegliedert, bis ein Wort (Terminalsymbol) erreicht wurde. Die Konstituenten werden innerhalb der XML-Struktur durch das <*cons*>-Element ausgedrückt. Terminalsymbole, und damit die nicht weiter zu

[9] http://www.ebi.ac.uk/citexplore
[10] http://www.nactem.ac.uk/enju/

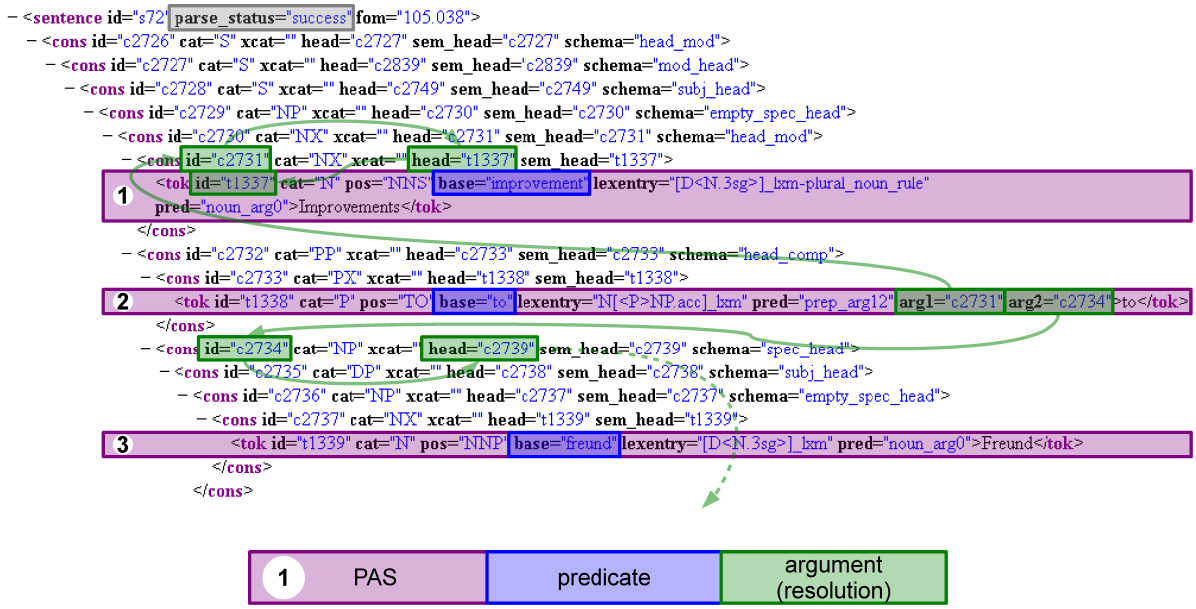


Abbildung 3.3: Ausschnitt einer von *Enju* aus Medline extrahierten Prädikat-Argument-Struktur

zerlegenden Wörter des Eingabetextes, werden durch das <*tok*>-Element dargestellt. Über dessen Attribute werden alle detektierten Eigenschaften festgehalten. Sie umfassen neben der semantischen und syntaktischen Struktur auch die Satzgliederung (NounPhrase, VerbPhrase).

Leider kann den Attributen eines *tok*-Elements die durch sie dargestellte PAS jedoch nicht direkt entnommen werden. Dies erfordert zunächst eine Auflösung, die in Abbildung 3.3 schematisch dargestellt ist. Die Auflösung beginnt jeweils mit dem Prädikat, das als Wert des *tok*-Elements repräsentiert wird. Die ihm zugeordneten Argumente werden hingegen nur indirekt über die Attribute *pred* und arg_x referenziert. Das Suffix des *pred*-Attributes zeigt die Anzahl der Prädikatsstellen an, während die Attribute arg_x auf die einzelnen Argumente verweisen. Die Referenzierung der Elemente erfolgt anhand ihrer eindeutigen *id*, die jedem XML-Element in der Struktur zugewiesen ist. Die Herausforderung besteht nun darin, das durch die jeweilige *id* repräsentierte Element und damit den entsprechenden Konstituenten zu identifizieren. Da eine *id* ein beliebiges Element identifizieren kann, kann dies durchaus eine komplexe Tochterstruktur aufweisen. Damit muss zur Ermittlung eines einzelnen Prädikat-Arguments nicht nur der direkt referenzierte Konstituent betrachtet werden, sondern die gesamte von ihm ausgehende Teilstruktur. Sie umfasst im Regelfall mehrere, die Wörter eines Satzes darstellende, *tok*-Elemente.

Eine nützliche Option ist der Webservice, mit dem die Wissensextraktion auf beliebigen Maschinen ausgeführt werden kann. Die Last komplexer Analysen kann hierüber unproblematisch skaliert werden. In der Vergangenheit stellte die Universität Tokyo den Webservice zur Analyse von Medline-Abstracts sogar öffentlich zur Verfügung. Ihm musste lediglich die PMID des zu analysierenden Datensatzes übermittelt werden. In Folge der Fukushima-Katastrophe wurde der Webservice im Jahr 2011 jedoch leider ersatzlos eingestellt. Seither stehen nur noch die Sourcen zum Download bereit, um den Service eigenständig betreiben zu können.

3.1.4 LiMPET

Das *Literature Metabolic Pathway Extraction Tool (LiMPET)*[11] ist auf die Extraktion enzymatischer Reaktionen aus Texten spezialisiert. Werden die gewonnenen Pathways anschließend zusammengeführt, kann ein metabolisches Netzwerk entstehen. Gegenüber anderen TM-Algorithmen stellt er damit eine Besonderheit dar. Der in Java implementierte Algorithmus ist zudem aktuell der einzige TM-Ansatz, der sich auf die Extraktion metabolischer Pathways spezialisiert hat [CNSS12] [Cza15]. In der Vergangenheit gab es lediglich ein weiteres System (*EM-PathIE*) [HDG00] [GHD01], dessen Entwicklung in der Zwischenzeit eingestellt wurde.

Die Schwierigkeit beim Extrahieren metabolischer Reaktionen aus Textdaten liegt in ihrer Komplexität. Im Gegensatz zu Protein-Interaktionen müssen nicht nur Proteine im Text erkannt werden, sondern Substrate, Enzyme und Produkte unterschieden werden. Zusätzlich ist die Reihenfolge ihres Auftretens entscheidend, da sie Rückschlüsse auf die Reaktion ermöglichen kann. Das Extrahieren enzymatischer Reaktionen gliedert der Algorithmus daher in mehrere Schritte. Zu Beginn werden aus den zu analysierenden Textdaten relevante Sätze herausgefiltert. Dies setzt eine anfängliche Segmentierung und Tokenisierung voraus, für die das frei verfügbare Framework *openNLP* (Abschnitt 3.1.5.1) von Apache genutzt wird. Die Auswahl relevanter Sätze geschieht danach anhand der in ihnen enthaltenen Entitäten. Sie werden mit dem NER-System *BANNER* [LG+08] identifiziert. Die anschließende Analyse berücksichtigt nur Sätze, die zumindest ein Substrat und Produkt enthalten. Das gleichzeitige Auftreten eines Enzyms ist keine zwingende Voraussetzung. Für die detektierten Entitäten werden im Anschluss alle möglichen Kombinationen gebildet und in einem weiterem Schritt einem Scoring unterzogen. Durch die Bewertung jeder potentiellen Kombinationen wird versucht, nur plausible, enzymatische Reaktionen zu propagieren. Die Bewertung basiert auf einem Punktesystem, dass spezielle Schlüsselwörter im Satz berücksichtigt. Hierzu gehören insbesondere Verben, die eine potentielle Reaktion ausdrücken (z.B. *catalyze*) sowie bestimmte Präpositionen. Die Liste der Schlüsselwörter wurde hierfür manuell aufgestellt. Spezielle Konstellationen können die Bewertung auch negativ beeinflussen. Aufgrund der nicht zwingend erforderlichen Enzym-Identifikation können extrahierte Reaktionen mit einem unbestimmten Enzym repräsentiert werden.

Die Evaluation des Algorithmus erfolgte anhand eines Vergleichs. Hierfür wurden drei konkrete Pathways des Organismus *E. coli K-12 substr. MG1655* aus der bakteriell motivierten Datenbank *EcoCyc*[12] [KOMK+05] ausgewählt und die ihnen zugrundeliegenden Veröffentlichungen in Medline identifiziert. Die textbasierten Informationen wurden mit dem konzipierten Algorithmus analysiert und die rekonstruierten Netzwerke mit den Pathways in EcoCyc verglichen. Erstaunlicherweise lagen die erzielten Recall und Precision Werte auf dem, von der Extraktion von Protein-Interaktion, bekannten Niveau. Die mit dem beschriebenen Algorithmus erfolgte Rekonstruktion metabolischer Netzwerke ist damit nicht weniger zuverlässig, obwohl sie deutlich komplexer ist. Im Rahmen dieser Evaluation wurde auch der im Kontext der Rekonstruktion genutzte NER-Prozess BANNER mit einem alternativen System verglichen. Mit

[11] http://www.biomedcentral.com/content/supplementary/1471-2105-13-172-s1.zip
[12] http://www.ecocyc.org

OSCAR stünde auch ein unmittelbar auf auf chemische Terme spezialisiertes System zur Verfügung [CMR06] [BC07], das sogar bereits über Whatizit (Abschnitt 3.1.2) angesprochen werden kann. Theoretisch wäre OSCAR damit zum Identifizieren der Enzyme prädestiniert gewesen, wurde jedoch verworfen. Der auf Heuristiken und Regeln basierende Ansatz versucht Enzyme maßgeblich anhand ihrer Nomenklatur zu erkennen (Abschnitt 2.1.2) und unterliegt damit in biomedizinischen Texten häufig Fehlern. Einerseits wird die offizielle Nomenklatur in diesen Texten selten genutzt, andererseits setzen sich die Enzymnamen vielfach aus mehreren Wörtern zusammen. OSCAR erkennt dann nur einen Teil des vollständigen Enzymnamens.

3.1.5 openNLP, GATE und UIMA

In den vorherigen Abschnitten wurde deutlich, dass die Analyse biomedizinischer Textdaten teilweise auf weiteren Frameworks beruht. Die Extraktion metabolischer Pathways (Abschnitt 3.1.4) nutzt beispielsweise openNLP[13] zur Segmentierung der Eingabetexte. Aufgrund ihrer Bedeutung werden die bekanntesten Frameworks nachfolgend kurz vorgestellt. Sie alle erfordern eine intensive Einarbeitung und können somit nicht ad hoc genutzt werden. Spezialisierte, biomedizinische Anwendungen, die auf ihnen basieren, werden begleitend angeführt.

3.1.5.1 openNLP

Das in Java geschriebene Framework ist frei verfügbar und liegt aktuell in der Version 1.5.3 vor (März 2015). Es unterstützt nahezu alle Phasen eines Wissensextraktionsprozesses und beschränkt sich damit nicht nur auf die zuvor erwähnte Segmentierung und Tokenisierung von Textdaten. Das System kann ausschließlich über eine API genutzt werden und bietet keine graphische Oberfläche. Die verfügbaren Schnittstellen ermöglichen es jedoch, den zugrundeliegenden Analyseprozess zu beeinflussen. Diese Möglichkeit wird auch vom zuvor gezeigten Algorithmus zur Extraktion metabolischer Reaktionen (Abschnitt 3.1.4) genutzt. Die vom *Jena University Language & Information Engineering (JULIE)*-Lab angebotenen und auf die Analyse biomedizinischer Textdaten spezialisierten Komponenten, wurden dort zur Optimierung der Segmentierung herangezogen.

3.1.5.2 GATE

Die *General Architecture for Text Engineering (GATE)*[14] ist ein komplexes Framework, dass sich in mehrere Komponenten gliedert (Abbildung 3.4). Die erste Version des System wurde vor fast zwanzig Jahren veröffentlicht [CGW95] und seitdem beständig weiter ausgebaut [Cun02] [CMB+11]. Mittlerweile ist bereits die fünfte Version des Frameworks verfügbar, die vollständig in Java realisiert ist und eine komplette Produktfamilie bildet. Verantwortlich für die Entwicklung ist die University of Sheffield (UK). In der Bioinformatik unterstützt GATE beispielsweise

[13] http://opennlp.apache.org/
[14] http://gate.ac.uk/

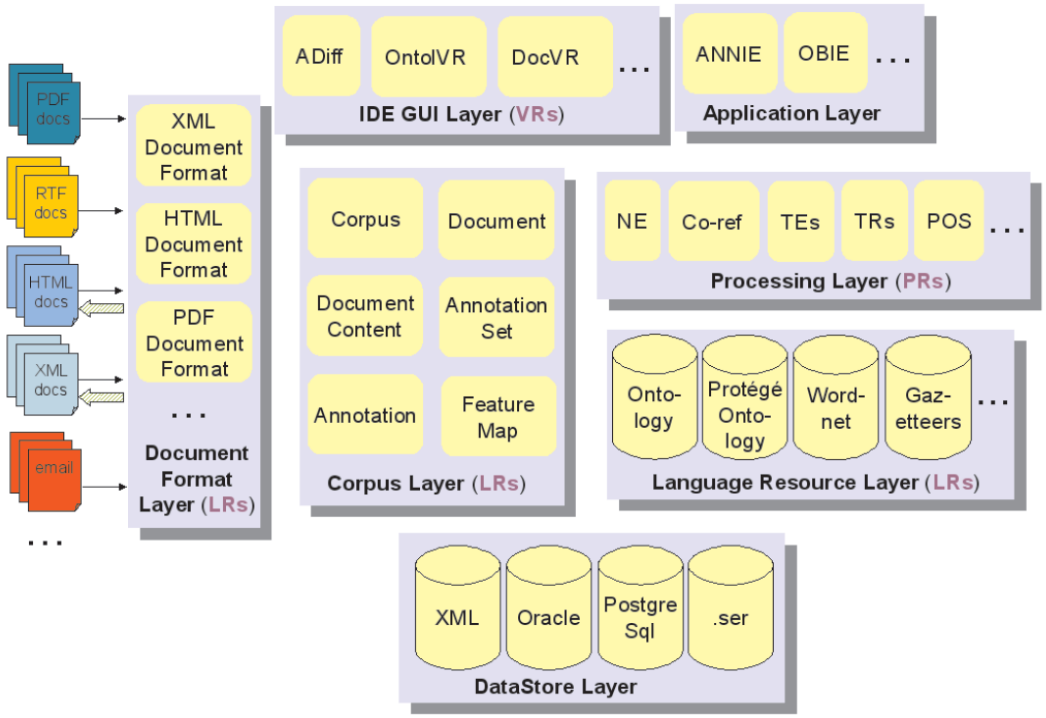


Abbildung 3.4: Systemdiagramm mit Anwendungsebenen von *GATE* [Cun]

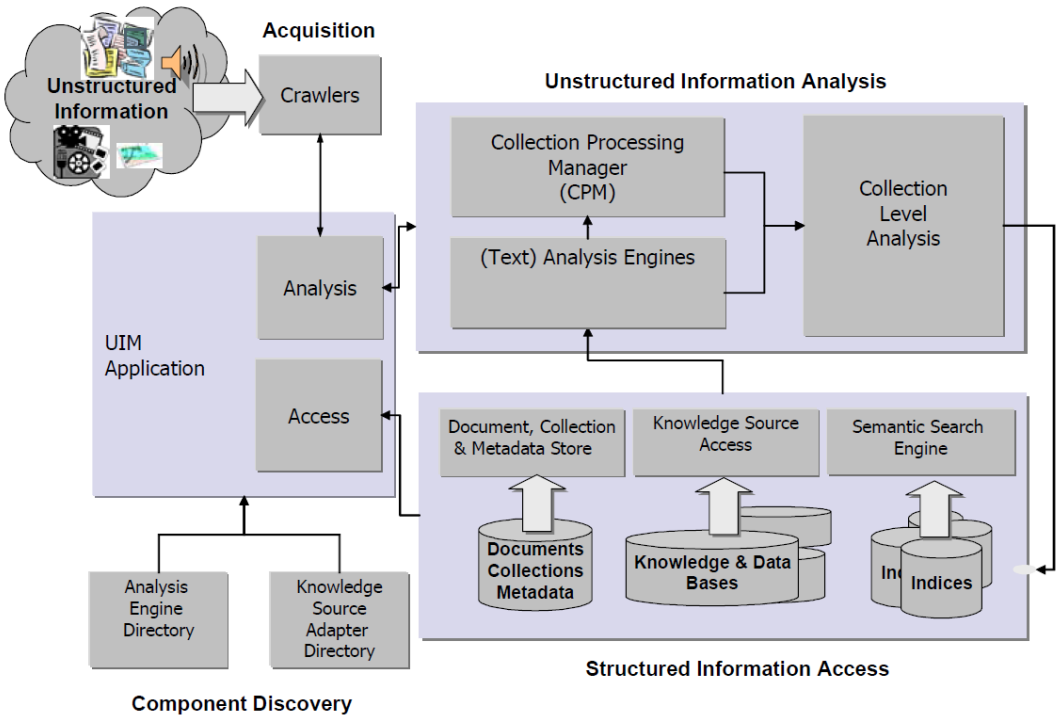


Abbildung 3.5: Vier zentrale Systemkomponenten bilden *UIMA*-Architektur [FL04]

das System *PathNER*[15]. Es versucht Pathway-Namen in Texten zu identifizieren und ihnen die korrespondiere ID aus KEGG zuzuordnen [WSN13] . Ein zentrales Konzept in GATE ist die *Collection of REusable Objects for Language Engineering (CREOLE)*. Die Objekte dieser Menge bilden die in GATE verfügbaren Funktionen und sind beispielsweise für die Anbindung der zu analysierenden Datenquellen verantwortlich. Sie übernehmen damit einzelne Schritte der Datenverarbeitung. Hierzu gehört mit dem *A Nearly New Information Extraction (ANNIE)*-Modul auch eine elementare Wissensextraktionskomponente des Frameworks, die dem Application Layer zugeordnet ist. ANNIE versucht, den verarbeiteten Texten ihre Struktur und Bedeutung mit Hilfe eines endlichen Automaten (*Java Annotation Patterns Engine (JAPE)*) zu entnehmen [CMT99]. Zusätzlich ist GATE mit dem *Language Resource Layer* in der Lage, existierende Wissens-Ontologien in den Analyse-Prozess unterstützend mit einzubeziehen. In der Vergangenheit wurden mit GATE bereits erfolgreich Texte aus den Bereichen der Bioinformatik, Gesundheitswissenschaften und Medizin analysiert. In mehreren, internationalen Vergleichen werden die mit GATE erzielten Ergebnisse daher in den Spitzenrängen geführt [Cun].

3.1.5.3 UIMA

Mit der *Unstructured Information Management Architecture (UIMA)*[16] wurde von IBM im Jahr 2005 ein ähnliches Framework wie GATE veröffentlicht [FL04]. Nur ein Jahr später wurde das Projekt der Apache Foundation übergeben und seitdem unabhängig weiterentwickelt. Da es bedeutend jünger als GATE ist, konnten wertvolle Erfahrungen aus diesem System in die Entwicklung einfließen. Genutzt wird die leistungsfähige Plattform beispielsweise vom *clinical Text Analysis and Knowledge Extraktion System (cTAKES)*[17]. Das System verarbeitet klinische Textdaten und identifiziert in ihnen automatisch Medikamente, Krankheiten sowie Symptome [SKSBC08] [SMO+10]. Im Unterschied zu GATE beschränkt sich UIMA nicht auf die Analyse von Textdaten. Auch gesprochene Sprache oder Bilddaten können verarbeitet werden. Besondere Anforderungen wurden während der Konzeption von UIMA an die Systemarchitektur gestellt. Das System sollte auch auf handelsüblichen Mobilgeräten nutzbar sein. Das Ergebnis der Anforderungen ist ein in vier Basiskomponenten gegliedertes System (Abbildung 3.5). Das Potential von UIMA wurde im Jahr 2011 medienwirksam unter Beweis gestellt. In der amerikanischen Quizshow *Jeopardy!* gelang es dem Hochleistungsrechner *Watson* gegen Stars dieser Sendung zu gewinnen, von denen einer bislang ungeschlagen war [Bak11].

3.2 Rekonstruktion biologischer Netzwerke

In diesem Abschnitt werden ausgewählte Systeme präsentiert, die eine Rekonstruktion biologischer Netzwerke ermöglichen. Die Systeme verfolgen verschiedene Ansätze und beziehen ihre

[15] http://sourceforge.net/projects/pathner/
[16] http://uima.apache.org
[17] http://ctakes.apache.org/

Daten teilweise aus mehreren Datenquellen. Die zwei grundlegenden Verfahren der Netzwerkrekonstruktion spiegeln sich hier auf unterschiedliche Weise wider (Abbildung 1.1). Daneben sind Systeme konzipiert worden, die sich den etablierten Verfahren nicht unmittelbar zuordnen lassen. Sie verfolgen beispielsweise statistische Ansätze oder paaren Expertenwissen mit experimentellen Daten[18] [AECBF+12]. Die Bandbreite der statistischen Ansätze wurde in [MS07] bereits diskutiert und klassifiziert. Netzwerke die mit den nachfolgenden Systemen rekonstruiert wurden, könnten generell zur Vorhersage weiterer Pathways in der Deduktionskomponente von *FraMeTex* gespeichert werden.

Die unumgängliche Validierung rekonstruierter Netzwerke erfolgt zumeist manuell durch biomedizinische Experten. Es gibt allerdings Bestrebungen auch diesen komplexen Prozess zu automatisieren. Eine aktuelle Studie formalisiert die Netzwerke zunächst in Zsyntax [BDDF10] und drückt ihre wesentlichen Eigenschaften in *higher-order logic (HOL)* [AHS14] aus. Im Anschluss wird dann versucht mit einem speziellem Theorembeweiser (HOL4) einzelne Pathways zu prüfen. Anhand eines bestimmten Pathways der Glykolyse[19] konnte dies bereits erfolgreich demonstriert werden [AHST14].

3.2.1 VANESA

Das System ermöglicht sowohl eine interaktive als auch eine automatische Rekonstruktion biologischer Netzwerke. Abbildung 3.6 zeigt ein mit VANESA rekonstruiertes Protein-Interaktionsnetzwerk. Ein Alleinstellungsmerkmal von VANESA[20] ist die Möglichkeit auf Basis der Netzwerke Simulationen durchzuführen. Das System wurde in der *Arbeitsgruppe für Bio- und Medizinische Informatik (AGBI)* der Universität Bielefeld entwickelt und in Java implementiert. Die zur Rekonstruktion erforderlichen Daten entnimmt VANESA dem *Data Warehouse System for Metabolic Data (DAWIS-M.D.)* [BJK+14]. Dieses fasst bekannte biomedizinische Datenbanken zusammen:

- KEGG [KG00]
- BRENDA [SCS02]
- IntAct [HMPL+04]
- *Molecular INTeraction database (MINT)* [CACP+07]
- *Human Protein Reference Database (HPRD)* [PGK+09].

Zusätzlich sind auch eine Reihe verschiedener *Omics*[21]-Datenbanken in DAWIS-M.D. verfügbar [HKT+10]. KEGG bietet Daten zu metabolischen Pathways. Intact, MINT und HPRD enthalten Informationen zu Protein-Interaktionen. BRENDA liefert bekannte Enzymdaten. Der

[18] Rekonstruktion eines Signalübertragungsnetzwerks im Kontext des *Follicle-stimulating hormone (FSH)*.
[19] *Fructose-1,6-bisphosphat (F1,6P)*
[20] http://agbi.techfak.uni-bielefeld.de/vanesa/
[21] gen*omic*, prote*omic*, transcript*omic*

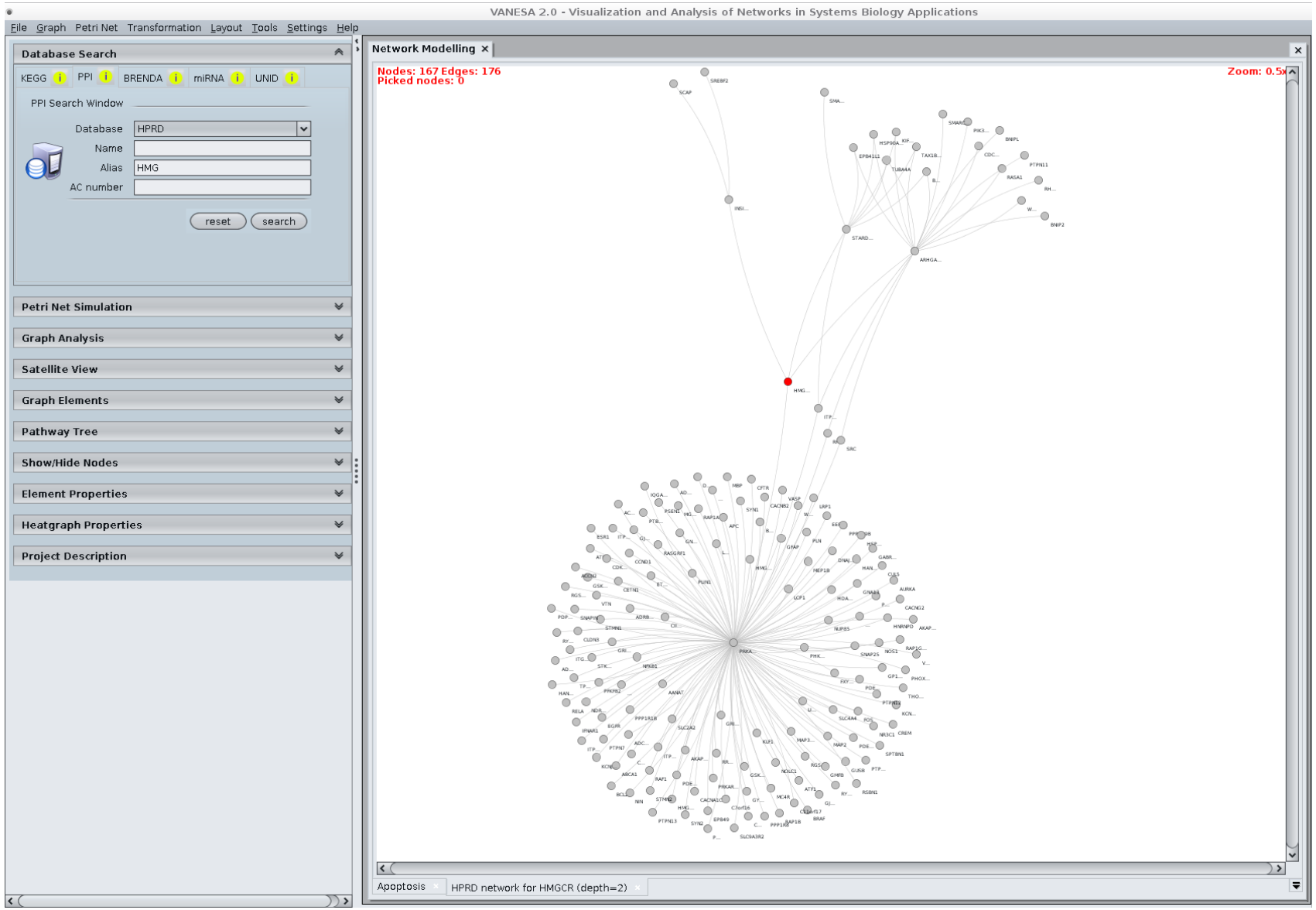


Abbildung 3.6: Rekonstruktion eines Interaktionsnetzwerks für Protein HMG (rot) mit VANESA

Zugriff auf sämtliche Informationen im Datawarehouse erfolgt unkompliziert über verschiedene Webservices.

Die Rekonstruktion eines Netzwerks beginnt mit einer initialen Suche in DAWIS-M.D. Hierfür bietet VANESA eine Suchmaske mit der das Datawarehouse nach biologischen Schlagwörtern durchsucht und die Suche gleichzeitig auf bestimmte Organismen eingeschränkt werden kann. Alle die Suche betreffenden Informationen werden daraufhin zusammengetragen und in einem Netzwerk zusammengefasst. Die zwischen einzelnen, biologischen Objekte geltenden Verbindungen müssen hierfür schrittweise aufgebaut werden, bis ein vollständiges Netzwerk vorliegt. Zur Rekonstruktion metabolischer Pathways wird daher beispielsweise für jedes gesuchte Enzym, Produkt oder Substrat eine vollständige Reaktionsliste erstellt. Sie enthält alle damit in Verbindung stehenden, biologischen Objekte. Die Suchtiefe und mit ihr die Größe der Netzwerke kann vom Benutzer eingeschränkt werden. Der umfangreiche Datenbestand in DAWIS-M.D. ermöglicht VANESA die Modellierung und Analyse nahezu aller in einer Zelle ablaufenden Prozesse. Neben metabolischen Pathways können daher auch Protein-Interaktionsnetzwerke oder Signalübertragungs-Netzwerke rekonstruiert und modelliert werden. Für die Visualisierung der Netzwerke stehen leistungsfähige Layout-Algorithmen zur Verfügung. Die erzeugten Graphstrukturen können vom Benutzer beliebig expandiert, reduziert und bearbeitet werden.

Als zusätzliches Feature besteht die Möglichkeit, experimentelle Labordaten in die Struktur manuell zu integrieren. Außerdem können unterschiedliche Graphstrukturen miteinander verglichen werden. Import- und Export-Funktionalitäten erlauben zudem den Austausch biologischer Netze in genormten Formaten. Eine kommerzielle Anwendung mit ähnlichem Fokus ist *Pathway Studio*[22]. Sie ist jedoch weiter spezialisiert und konzentriert sich auf die Analyse metabolischer Pathways. Das Tool zielt darauf ab, neue Medikamente und Wirkstoffe zum Schutz von Nutzpflanzen zu designen. Weitere Systeme mit einer Ähnlichkeit zu VANESA sind VANTED[23] [JKS06] sowie Cytoscape[24] [SMO+03].

3.2.2 ANDSystem

Das ANDSystem[25] rekonstruiert biologische Netzwerke aus biomedizinischen Textdaten und stellt sie graphisch dar. Zur Unterstützung des Prozesses können ausgewählte Datenquellen (z.B. IntAct, MINT) konsultiert werden. Konzipiert und entwickelt wurde das System am Institute of Cytology and Genetics in Nowosibirsk (Russland) [PYD+11]. Das Institut ist seit dem Jahr 2005 über das *German-Russsian Virtual Network of Bioinformatics*[26] mit der AGBI an der Universität Bielefeld verbunden. Eine limitierte Version des Tools ist frei verfügbar. Eine kommerzielle Lizenz wird von der Firma PBSoft vermarktet.

[22] http://www.elsevier.com/online-tools/pathway-studio
[23] http://vanted.ipk-gatersleben.de/
[24] http://www.cytoscape.org
[25] http://www.pbiosoft.com/andsystem
[26] http://www.imbio.de/forschung/index.php

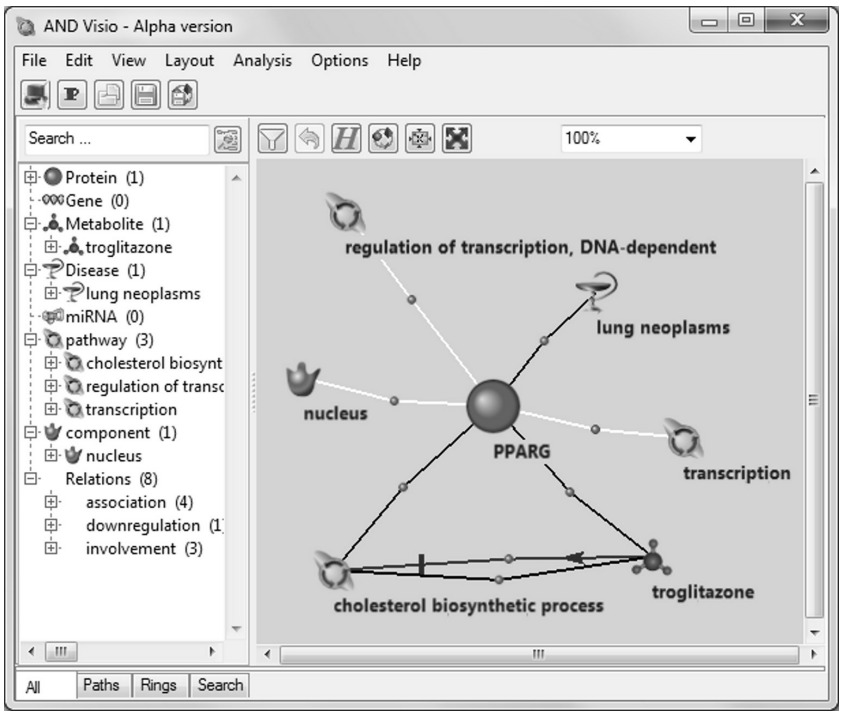


Abbildung 3.7: Von *ANDCell* rekonstruiertes Netzwerk, visualisiert mit *ANDVisio* [DIKI12]

Die initiale Version des Systems war vollständig in Pascal geschrieben und damit nur auf ausgewählten Systemen lauffähig. Aktuell stehen zwei Versionen für Linux und Windows zum Download bereit (März 2015). Bis heute gliedert sich ANDSystem in die beiden Komponenten *Associative Network Discovery (AND)*Cell sowie ANDVisio. Während ANDCell darauf ausgelegt ist, Wissen aus Textdaten zu extrahieren, übernimmt ANDVisio die Rekonstruktion und Visualisierung der Netzwerke [DIKI12]. Analysiert werden ausschließlich Medline-Abstracts. Die aus ihnen gewonnenen Informationen werden in einer relationalen Datenbank abgelegt. Über einen Webservice und nach vorheriger Rücksprache kann mit ANDVisio ein zeitweiser Fernzugriff auf die Datenbank erfolgen. In der limitierten Version des Tools enthält die Datenbank allerdings lediglich Informationen die bis zum Jahr 2006 publiziert wurden.

Die von ANDCell genutzten Algorithmen sind leider nicht im Detail bekannt. Die verfolgte Netzwerkrekonstruktion kann damit nur begrenzt nachvollzogen werden. Die Algorithmen basieren vornehmlich auf neuronalen Netzen sowie regelbasierten Ansätzen [ADP+06]. Unterstützend werden Kookurrenz-Analysen sowie verschiedene Wörterbücher zur NER genutzt (Abschnitt 2.2.1). Für eine umfassende Wissensextraktion wurden bereits mehr als 2.000 Regeln und Muster definiert, die semantische Zusammenhänge zwischen biologischen Objekten beschreiben. Die von ANDCell verfolgte Netzwerkrekonstruktion aus Textdaten gliedert sich damit in zwei elementare Schritte:

1. Identifikation zellulärer Komponenten (Proteine, Gene, RNS usw.)

2. Erkennen relevanter Zusammenhänge zwischen diesen Komponenten

Von besonderem Interesse sind diejenigen Zusammenhänge, die Protein-Interaktionen oder enzymatische Reaktionen beschreiben. Sie bilden die Grundlage zur Rekonstruktion der biologischen Netzwerke. Darüber hinaus werden die identifizierten Zellkomponenten aber beispielsweise auch mit erkannten Krankheiten in Verbindung gesetzt. Zur besseren Unterscheidung werden die verschiedenen Entitäten mit unterschiedlichen Symbolen visualisiert (Abbildung 3.7). Eine genauere Spezifizierung der Kanten im Netzwerk erfolgt allerdings nicht. Sie repräsentieren abstrakte Beziehungen und damit keine detaillierten Informationen (z.B. regulates). Ein Query-Wizard ermöglicht die Formulierung komplexer Suchanfragen, mit denen der zu analysierende Datenraum anhand benutzerdefinierter Kriterien eingeschränkt werden kann. Damit kann beispielsweise auf Protein-Interaktionen fokussiert werden. Erfolgreich rekonstruierte Netzwerke können schließlich in einem individuellen, XML-basierten Format (*.and*) exportiert werden. Dies erlaubt den Austausch von Netzwerken zwischen verschiedenen Anwendern. Standardisierte Formate der Bioinformatik (z.B. *Systems Biology Markup Language (SBML)* [HFS+03]) werden bisher noch nicht unterstützt.

3.2.3 PathPred / KEGG automatic annotation server

Die webbasierte Anwendung PathPred[27] ist darauf spezialisiert enzymatische Reaktionen vorherzusagen. Die Applikation ist im Internet frei verfügbar. Ausgangspunkt für die Vorhersage ist eine biologische bzw. chemische Substanz, die vom Anwender spezifiziert werden muss. PathPred wurde am Bioinformatics Center der Universität Kyoto in Japan entwickelt. Das System ist ein Nachfolger des *KEGG automatic annotation server (KAAS)*-Webservice. Mit ihm war bereits eine Rekonstruktion metabolischer Netzwerke möglich [MIO+07], die jedoch fehleranfällig war. Vielfach wiesen diejenigen Pathways Fehler auf, die den Abbau biologischer Produkte oder die Synthese von Sekundärmetaboliten[28] beschrieben. Zur Lösung dieses Problems nutzt das weiter entwickelte PathPred nun bereits verfügbares, chemisches Wissen über enzymatische Reaktionen aus [MSH+10]. Die erforderlichen Informationen werden von PathPred den Datenbanken *KEGG REACTION*, *KEGG COMPOUND* sowie *RPAIR* entnommen [KGF+10][29].

Die REACTION-Datenbank umfasst alle Reaktionen, die sich aus der Enzym-Nomenklatur IUBMB (Abschnitt 2.1.2) ableiten lassen und wird durch die metabolischen Pathways aus KEGG ergänzt. Die REACTION-Datenbank bildete schließlich die Basis für den Aufbau der RPAIR-Datenbank. Hierfür wurden aus REACTION Muster extrahiert (RDM-Pattern), die biochemische Struktur-Transformationen für Substrat-Produkt-Paare (Reaktionspartner) beschreiben. Bevor die RDM-Pattern in der RPAIR-Datenbank abgelegt wurden, erfolgte ihre manuelle Validierung. Für eine erfolgreiche Vorhersage potentieller Reaktionspfade zwischen *zwei* spezifizierten Substanzen ist RPAIR jedoch nicht zwingend erforderlich. Solange beide Substanzen in REACTION enthalten sind, kann allein mit den dort enthalten Substrat-Produkt-

[27] http://www.genome.jp/tools/pathpred/
[28] Stoffwechselprodukte eines Organismus, die er zum (Über-)Leben nicht benötigt.
[29] http://www.genome.jp/kegg/kegg2.html

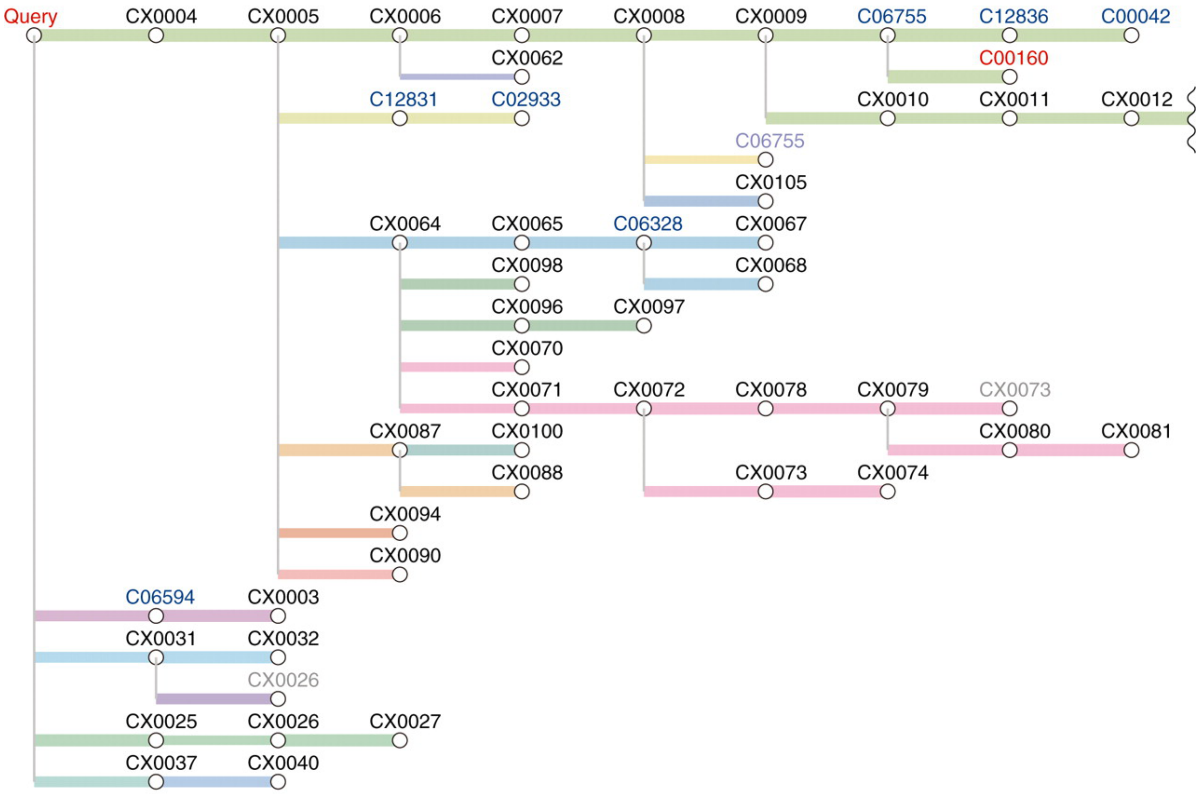


Abbildung 3.8: Rekonstruierter Pathway-Tree von PathPred [MSH+10]

Beziehungen eine *Pathway Prediction* erfolgen. Der entsprechende Algorithmus ist in PathComp[30] [OGFK98] umgesetzt und bereits seit längerem verfügbar. Mit dem weiterentwickelten PathPred gelingt nun sogar eine Vorhersage, wenn zu den vom Anwender spezifizierten Substanzen keine Einträge in REACTION gefunden werden. Möglich wird dies durch die in RPAIR enthaltenen RDM-Pattern, die losgelöst von spezifischen, chemischen Verbindungen sind. Sie beschreiben Strukturen, die in verschiedenen Verbindungen zu finden sind.

Die automatische Vorhersage eines Pathways durch PathPred gliedert sich schließlich in mehrere Schritte. Zunächst wird für die vom Anwender spezifizierte Substanz eine Ähnlichkeitssuche in der KEGG COMPOUND-Datenbank angestoßen [HTKG10]. Im Anschluss werden für die darüber identifizierten Substanzen zutreffende RDM-Pattern aus RPAIR geladen. Die durch die Muster beschriebenen Transformationen führen zu neuen Substanzen, für die wiederum RDM-Pattern ermittelt werden und der beschriebene Vorgang wiederholt wird. Findet sich für eine Substanz kein Transformationsmuster, wird für sie eine erneute Ähnlichkeitssuche in der COMPOUND-Datenbank angestoßen. Der iterative Ansatz führt schließlich zu einer schrittweisen Vorhersage plausibler Pathway-Maps. Hierfür wurde ein spezielles Scoring definiert, das diejenigen Transformations-Muster bewertet, die zum jeweiligen Pathway geführt haben. Für jede enzymatische Reaktion innerhalb des Pathways versucht PathPred außerdem ein zutreffendes Enzym zu finden. Gelingt dies nicht, besteht die Möglichkeit mit einem Tool eine

[30] http://www.genome.jp/tools/pathcomp/

entsprechende EC-Nummer zu definieren [KOH+04] [YHK+09].

Die Vorhersage eines Pathways kann beschleunigt werden, wenn neben einer Start- auch eine Zielsubstanz vom Anwender spezifiziert wird. In diesem Fall erfolgt eine bidirektionale Vorhersage des Pathways, bis eine Verbindung zwischen Start- und Ziel-Substanz gefunden wurde. Ein Abbruch erfolgt, falls nach einer definierten Anzahl Iterationen kein Pathway ermittelt werden konnte. Da der Prozess trotz des bidirektionalen Ansatzes immer noch einige Zeit in Anspruch nehmen kann, erfolgt eine E-Mail-Benachrichtigung, wenn die Ergebnisse vorliegen. Eine von PathPred rekonstruierte Pathway-Map zeigt Abbildung 3.8. Die Knoten repräsentieren Komponenten, die Kanten enzymatische Reaktionen. Die Plausibilität einer Reaktion ist durch die Stärke der Kante hervorgehoben. In KEGG bekannte Substanzen beginnen mit C (blau), unbekannte mit CX (schwarz). Mehrfach verknüpfte Substanz sind gräulich dargestellt, eine ggf. vom Anwender spezifizierte Zielsubstanz rot.

3.2.4 UM-PPS

Das webbasierte UM-PPS[31] ist darauf spezialisiert, den mikrobiellen Katabolismus organischer Verbindungen vorherzusagen. Die speziellen, metabolischen Pathways (Abschnitt 2.1.4.1) sind für die Umwandlung von Metaboliten in Energie verantwortlich und übernehmen die Entgiftung eines Organismus. Sie nehmen damit eine zentrale Stellung in jedem Organismus ein. Zur Umsetzung der *Pathway Prediction* greift das System auf einen regelbasierten Ansatz zurück. Die vorhergesagten Pathways werden in einem azyklischen Graphen visualisiert und können mehrere, aufeinanderfolgende Reaktionen umfassen. Das System ist in Java implementiert und reicht bis in das Jahr 1998 zurück. Seine kontinuierliche Entwicklung führte in den vergangenen Jahren zu immer leistungsfähigeren Vorhersagen der Pathways [GEW11]. Federführend war die University of Minnesota, die im Jahr 2014 die Rechte an dem System jedoch an die *Eidgenössische Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz (EAWAG)* in der Schweiz übergab. Das System wurde daraufhin umbenannt[32] und das Kürzel UM in allen Bezeichnungen durch EAWAG ersetzt. Entsprechend der verfügbaren Literatur werden nachfolgend noch die ursprünglichen Namen verwendet.

Eine technische Grundlage des Systems bildete die *University of Minnesota biocatalysis/biodegradation database (UM-BBD)*. Die manuell gepflegte MySQL-Datenbank enthält mittlerweile über 1300 enzymatische Reaktionen sowie 800 Enzyme und mehr als 1200 chemische Substanzen. Aus diesen Daten wurden mit Unterstützung zusätzlicher Literatur 260 Regeln abgeleitet, die von UM-PPS zur Vorhersage plausibler Pathways genutzt werden [GEW09]. Die Anwendung einer Regel hängt von den konkreten Eigenschaften eines Substrats bzw. Metaboliten ab. Berücksichtigt wird neben dessen chemischer Struktur der atomare Aufbau [FGK+08]. Zur Vorhersage eines metabolischen Pathways für eine organische Verbindung ist daher zunächst dessen chemische Struktur zu definieren. Die Struktur kann mit Hilfe des *MarvinView-*

[31] http://eawag-bbd.ethz.ch/predict/
[32] http://eawag-bbd.ethz.ch/whatsnew.html

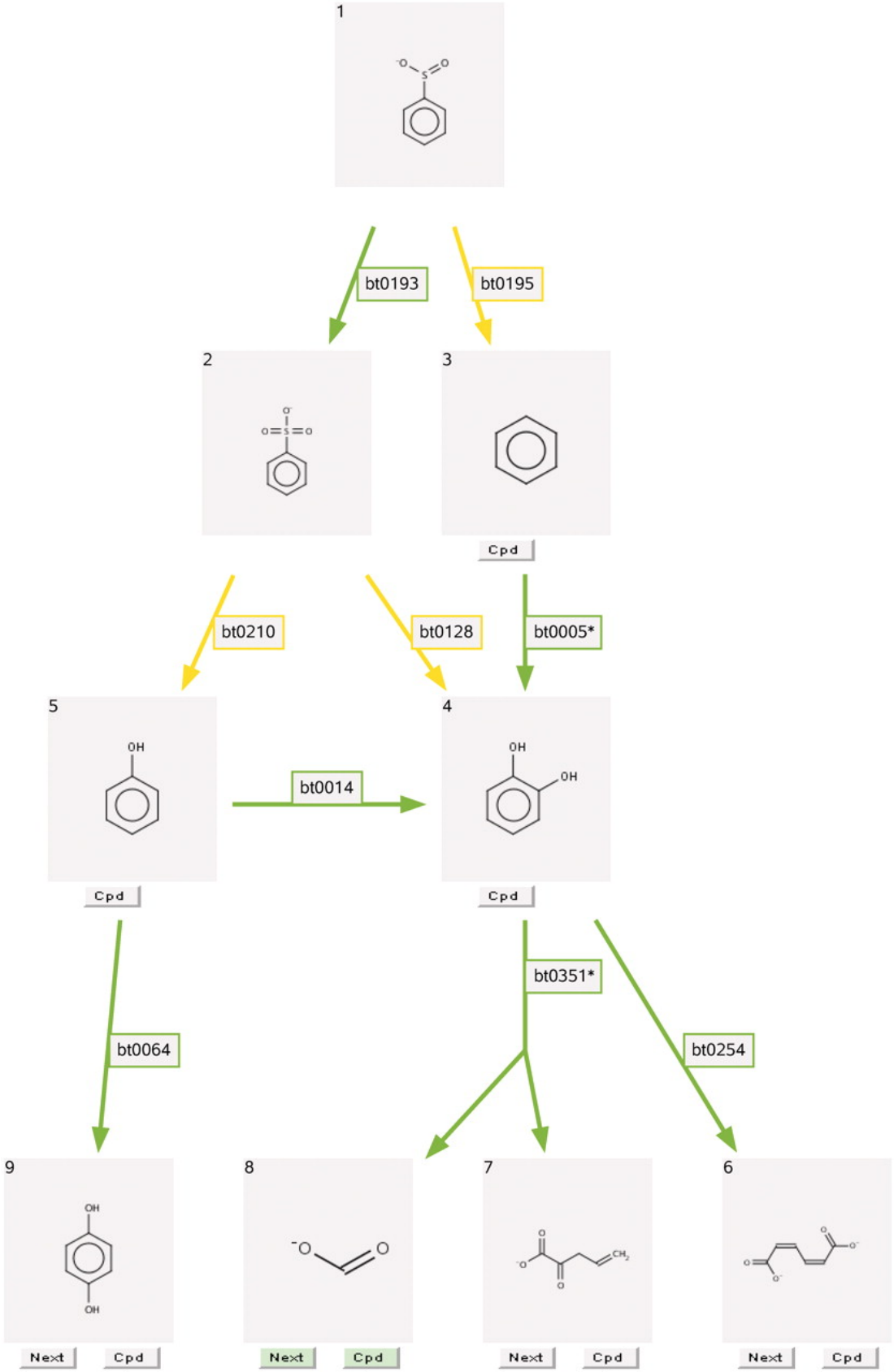


Abbildung 3.9: *Pathway Prediction* mit UM-PPS für die Substanz *benzene sulfinate* [GEW11]

Applets gezeichnet werden [Csi00]. Im Anschluss wird sie in einen speziellen *Simplified Molecular Input Line Entry Specification (SMILES)*-String überführt, der die jeweilige Struktur eindeutig beschreibt [Wei88]. Ist der SMILES-String der Struktur bereits bekannt, kann dieser auch direkt angegeben werden und die aufwändige Zeichnung der chemischen Struktur entfallen [EGFW08].

Entscheidend für die Vorhersage eines Pathways ist die Wahl der auszulösenden Regeln, wenn mehrere Regeln anwendbar sind. Zur Lösung dieses Problems werden die in UM-PPS verfügbaren Regeln in Kategorien eingeteilt und priorisiert. Berücksichtigt wird beispielsweise die Reaktionswahrscheinlichkeit eines Substrats mit Sauerstoff (aerobic likelihood) oder ob eine bestimmte Regelfolge zu besonders wahrscheinlichen Pathways führen könnte (super rules). Die Priorisierung der Regeln erfolgt automatisch nach festgelegten Kriterien [FGK+08]. Die Anwendung einer Regel übernimmt der virtuelle Reaktionssimulator *ChemAxon Reactor*[33]. Die Rekonstruktion eines vollständigen Netzwerks erfolgt schließlich in einem iterativen Verfahren. Es beginnt mit der vom Anwender spezifizierten, biologischen Verbindung, für die zutreffende Regeln gesucht werden. Finden sich mehrere, wird in dem rekonstruierten, metabolischen Netz für jede angewendete Regel eine Kante vom Substrat zum jeweiligen Produkt gezeichnet. Für jedes Produkt wird dann erneut nach zutreffenden Regeln gesucht und der Vorgang wiederholt. Der Vorgang endet, wenn keine Regeln mehr angewendet werden können oder die resultierende Baumstruktur eine vordefinierte Breite oder Tiefe erreicht hat.

Die Visualisierung des rekonstruierten Netzwerks erfolgt anschließend mit Graphviz [EGK+02]. Die von dem Tool erzeugte Graphstruktur wird in einer Webseite eingebunden und kann interaktiv exploriert werden. Ein von UM-PPS rekonstruiertes, metabolisches Netzwerk zeigt Abbildung 3.9. Die Kanten enthalten Links zu den jeweils angewendeten Regeln, die Knoten bieten Links (Button *Cpd*) zu Einträgen in KEGG (grün) oder UM-BBD (grau). Innerhalb eines Knotens wird die chemische Struktur des jeweils repräsentierten Metaboliten dargestellt. Ist die Tiefe oder Breite des Graphen beschränkt worden, besteht die Möglichkeit die *Pathway Prediction* manuell fortzuführen (Button *next*). Mit *METEOR*[34] und *MetabolExpert*[35] existieren auch kommerzielle Systeme, die ebenfalls eine Vorhersage auf chemischer Ebene verfolgen. *MetabolExpert* zählt sogar zu den ältesten Systemen dieser Art. Es wurde bereits im Jahr 1987 als Expertensystem präsentiert und seitdem stetig weiter ausgebaut [Dar87].

3.2.5 PathoLogic (Pathway Tools)

Die Pathway Tools bilden ein komplexes Framework, das mehrere Komponenten (Tools) umfasst. Ihr Fokus liegt auf der Verwaltung, Visualisierung und Analyse integrierter Daten, die Pathways, Regulationen oder das Erbgut eines Organismus beschreiben. Die Entwicklung der Systemlandschaft begann bereits Anfang der 1990er Jahre und erfolgt seitdem durch die Bioinformatics Research Group am *Stanford Research Institute (SRI)* der Stanford University in

[33] http://www.chemaxon.com
[34] http://www.lhasalimited.org/products/meteor-nexus.htm
[35] http://www.compudrug.com/metabolexpert

Menlo Park (USA). Mittlerweile wurde das System für mehr als 1.700 Benutzer lizenziert. Im Zentrum der Pathways Tools stehen spezielle *Pathway/Genome Databases (PGDBs)*, die jeweils alle verfügbaren Informationen für einen bestimmten Organismus zusammenfassen. Der Aufbau einer PGDB wird durch automatisierte Importe existierender Datenquellen unterstützt und kann manuell weiter verfeinert werden [KPK+09]. Eine zentrale Bedeutung kommt in diesem Kontext der GenBank[36] zu. Die frei zugängliche Datenbank enthält annotierte Nukleotidsequenzen von fast 260.000 verschiedenen Organismen [BCC+12]. Die Annotation kann den Beginn und das Ende eines Kodierungsbereichs umfassen, die Funktion eines Genprodukts (z.B. Enzym), EC-Nummern oder auch mehrere GO-Terme.

Die eigentliche *Pathway Prediction* übernimmt das Tool *PathoLogic*. Im Gegensatz zu anderen Verfahren zielt sie jedoch nicht darauf ab, neue Pathways vorherzusagen, sondern Pathways eines bereits bekannten Organismus auf einen anderen zu übertragen [PK02]. Neben der GenBank wird hierfür zusätzlich auf die Datenbank MetaCyc[37] zugegriffen. MetaCyc enthält Informationen zu metabolischen Pathways und Enzymen verschiedener Organismen [KZM+04]. Insgesamt sind mehr als 2.100 experimentell bestätigte Pathways in MetaCyc verfügbar, die aus mehr als 37.000 Publikationen extrahiert wurden [CAB+14]. Bevor es zur *Pathway Prediction* kommt, baut PathoLogic zunächst eine PGDB für den zu betrachtenden Organismus im Rahmen einer Initialisierung auf. Die PGDB enthält das annotierte Genom des Organismus, das GenBank entnommen wird. Das Schema der PGDB orientiert sich an MetaCyc. Auf Basis der PGDB gliedert sich die anschließende Vorhersage von Pathways in zwei Schritte:

1. *Zuordnung der Enzyme in PGDB zu Reaktionen in MetaCyc.* PathoLogic nutzt hierfür die in den Annotationen enthaltenen Informationen und vergleicht Enzymnamen, EC-Nummern und GO-Terme. Der konkrete Vergleich hängt von jeweils verfügbaren Informationen im Genom ab. Führt dies zu keiner Zuordnung kann mit dem „Pathway Hole Filling“ Tool eine Hypothese generiert werden [GK04]. Da eine automatische Zuordnung dennoch missglücken kann, sind mit einem weiteren Tool Benutzerinteraktionen möglich.

2. *Import ausgewählter Pathways aus MetaCyc in die PGDB.* Kopiert werden diejenigen Pathways, die zumindest mit einer enzymatischen Reaktion in PGDB übereinstimmen. Anschließend werden unzutreffende Pathways ausgeschlossen. Für jeden Pathway wird der prozentuale Anteil an Reaktionen berücksichtigt, denen zuvor ein Enzym zugeordnet werden konnte. Außerdem wird geprüft, wieviele der Reaktionen mit zugeordneten Enzymen ausschließlich in dem aktuell betrachteten Pathway vorkommen.

Nach Abschluss der Filterung gelten die verbliebenen Pathways für den betrachteten Organismus als vorhergesagt. Der Algorithmus von PathLogic basiert damit auf Regeln und Heuristiken. Das kontinuierliche Feedback von Biologen sorgte in den vergangenen Jahren für eine immer weitergehende Verfeinerung und führte zu präziseren Vorhersagen. Die hart kodierten Regeln schränken die Flexibilität des Algorithmus jedoch ein. Zudem werden mit zunehmenden Wachstum von MetaCyc immer mehr unzutreffende Pathways vorhergesagt. Es wurde daher

[36] http://www.ncbi.nlm.nih.gov/genbank
[37] http://www.metacyc.org

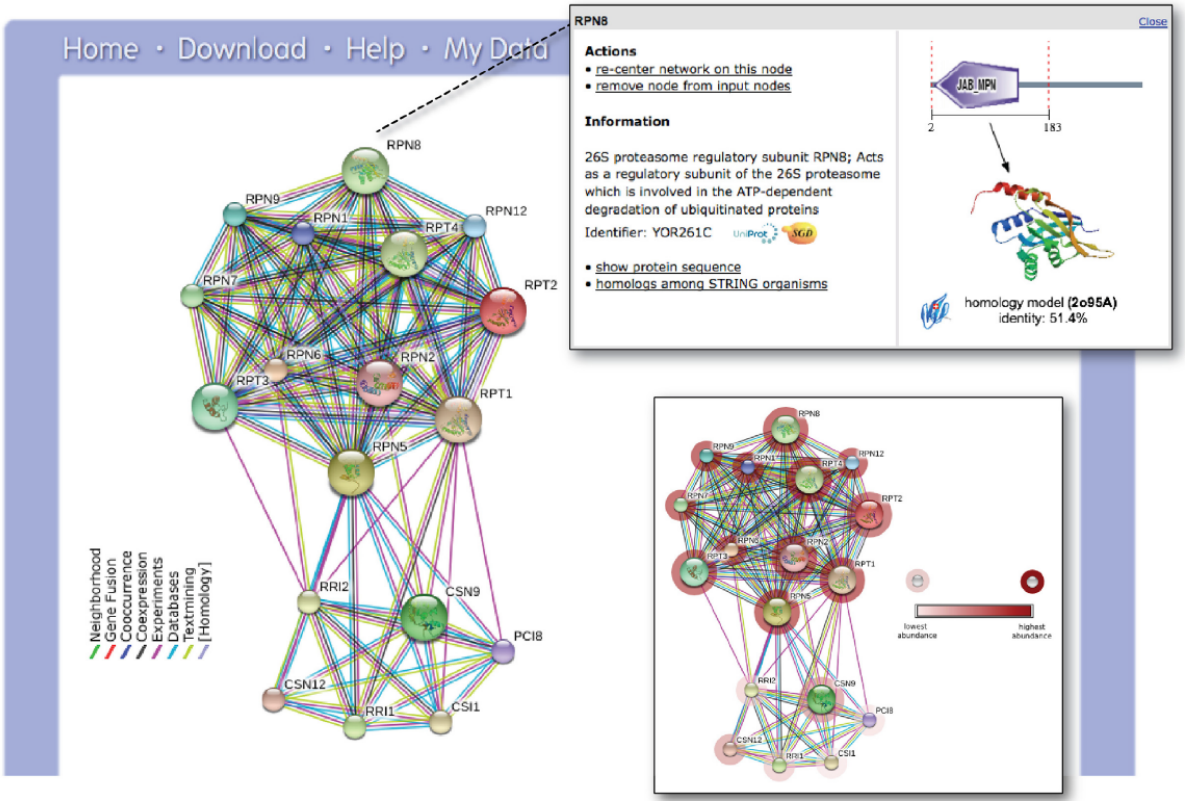


Abbildung 3.10: Rekonstruiertes Netzwerk mit STRING (kombinierte Screenshots) [SFW+14]

neben PathLogic ein weiteres Inferenz-Tool konzipiert, das auf maschinellen Lernmethoden basiert. Mit dem neuen Tool kann nun außerdem nachvollzogen werden, warum ein bestimmter Pathway in die Vorhersage einbezogen oder von ihr ausgeschlossen wurde [DPK10].

Die zentrale Herausforderung bei der von PathoLogic verfolgten Vorhersage metabolischer Pathways ist die korrekte Zuordnung der Enzyme zu Reaktionen. Das Problem ist allerdings nicht auf diesen Algorithmus beschränkt. Auch andere Systeme, die metabolische Netzwerke rekonstruieren, müssen diese Herausforderung bewältigen. Vielfach greifen sie hierfür auf verschiedene Techniken der Sequenzanalyse zurück. Einen derartigen Ansatz verfolgen beispielsweise IdentiCS [SZ04], metaShark [PSMW05] sowie Pathway Analyst [PPS+05]. Auf diese Systeme wird hier jedoch nicht weiter eingegangen.

3.2.6 STRING

Das *Search Tool for the Retrieval of Interacting Genes/Proteins (STRING)*[38] konzentriert sich auf die Rekonstruktion von Protein-Interaktionsnetzwerken und verfolgt einen integrativen Ansatz. Die Ursprünge von STRING reichen bis ins Jahr 2000 zurück [SLBH00]. An der Entwick-

[38] http://string-db.org/

lung des Systems waren das *European Molecular Biology Laboratory (EMBL)*, das *Swiss Institute of Bioinformatics (SIB)* sowie das *Novo Nordisk Foundation Center for Protein Research (CPR)* maßgeblich beteiligt. Im Mittelpunkt des Systems steht eine Datenbank, in der sowohl direkte (physikalische) als auch indirekte (funktionale) Interaktionen hinterlegt sind. In der aktuellen Version umfasst sie zehn Millionen Proteine und mehr als 2000 Organismen[39]. Die Datenbank ist frei zugänglich und wird regelmäßig aktualisiert.

Die in STRING verfügbaren Informationen werden aus unterschiedlichsten Datenquellen zusammengeführt. Neben etablierten Datenbanken (HPRD, KEGG u.a.) fließen in das System auch experimentell erzielte sowie mittels TM aus Medline und anderen Textquellen extrahierte Informationen ein. Trotz des integrativen Rekonstruktionsansatzes finden damit TM-Verfahren in STRING Anwendung. Auf Basis der zusammengeführten Daten werden von STRING schließlich weitere Protein-Interaktionen vorhergesagt. Für diese als „*de novo*“ bezeichneten Interaktionen werden genetische Informationen algorithmisch ausgewertet und Koexpressionsanalysen (Abschnitt 2.1.3) durchgeführt. Außerdem werden Protein-Interaktionen, die in einem Organismus beobachtet wurden systematisch auf andere Organismen übertragen (*interolog*-Transfer). Grundlage für dieses Verfahren sind zuvor ermittelte *Orthologie*[40]-Relationen. Sie beschreiben jeweils die Beziehung zwischen zwei Genen in verschiedenen Spezies, die von einander abstammen. Alle Protein-Interaktionen in STRING sind mit einem Score belegt, der für neu vorhergesagte Interaktionen mit Hilfe von KEGG berechnet wird [SFW+14].

Der Zugriff auf die STRING-Datenbank kann über eine API oder die Website erfolgen. Auf der Website ermöglicht eine Suchmaske die Fokussierung auf bestimmte Proteine. Optional kann der Organismus angegeben werden. Wird der Organismus nicht angegeben und ist das gesuchte Protein nicht eindeutig einem Organismus zuzuordnen wird der User aufgefordert einen Organismus aus vorselektierten Möglichkeiten zu wählen. Sobald die Eingaben eindeutig sind, wird ein Netzwerk der ersten Ebene für das gesuchte Protein rekonstruiert. Die Farben der visualisierten Netzwerkkanten sowie deren Stärke bieten einen ersten Anhaltspunkt über den Ursprung der Vorhersage (z.B. Textmining, Koexpressionsanalyse) und deren Score. Damit kann auch erkannt werden, ob es sich um eine „*de novo*“ Interaktion handelt, die von STRING selbst vorhergesagt wurde. Zu jedem Protein und jeder Interaktion des Netzwerks sind außerdem weitere Details verlinkt. Die Bandbreite der verfügbaren Informationen kann Abbildung 3.10 entnommen werden.

3.3 Aufbau logikunterstützter Wissensbasen

Die eingangs motivierte *Pathway Prediction* erfordert eine Wissensbasis, die Schlussfolgerungen in biologischen Netzwerken unterstützt. Die entsprechenden Konzepte deduktiver Datenbanken wurden daher bereits im Abschnitt 2.2.3 diskutiert. Die in der Vergangenheit entwickelten Prototypen wurden in diesem Zusammenhang ebenso vorgestellt wie das aktuell verfügbare System XSB. Mit der Semantic Web Technologie (Abschnitt 2.2.2.2) gibt es darüber hinaus eine

[39] Stand März 2016
[40] Begriff kommt aus der Genetik.

potentielle Alternative[41]. Ihre wachsende Akzeptanz spiegelt sich auch in der Bioinformatik wider [CYS+05] [GW06]. Drei bekannte Frameworks werden daher nachfolgend charakterisiert. Ihre Performance wurde bereits aus Sicht der Bioinformatik und daher mit Blick auf große Datenmengen evaluiert [MSB+10].

3.3.1 Sesame

Sesame[42] war eines der ersten Frameworks, das die theoretischen Aspekte des Semantic Web praktisch umsetzte. Das Framework ist in Java geschrieben und steht aktuell in der Version 2.8.1 zur Verfügung (März 2015). Sesame ging aus dem EU-Projekt *On-To-Knowledge* hervor und ist bis heute frei verfügbar [FVHK+00]. Die Vision von On-To-Knowledge war es, das Wissensmanagement in firmeninternen Intranets sowie dem öffentlichen Internet durch den Einsatz der aufstrebenden Semantic Web Technologie zu verbessern. Hierfür wurde On-To-Knowledge in mehrere Komponenten unterteilt, von denen eine für die Wissensrepräsentation- und Persistierung verantwortlich war. Diese Komponente legte damit den Grundstein für die Entwicklung von Sesame. Verantwortlich für die Entwicklung dieser Komponente war die niederländische Firma *Aidministrator Nederland B.V.*, die mittlerweile unter dem Namen *Aduna*[43] geführt wird. Seit Beginn seiner Entwicklung unterstützt Sesame die Sprachebenen RDF und RDFS (Abschnitt 2.2.2.2). Eine Kernkomponente des Frameworks ist das *Repository Abstraction Layer (RAL)*. Es wurde eingeführt, um die physische Speicherung der einzelnen Fakten (RDF-Triple) vom eigentlichen Framework zu entkoppeln [BKVH02]. Dies erlaubt die Nutzung eines beliebigen Persistenz-Layers, solange ein entsprechendes RAL existiert oder bereitgestellt wird. Damit können beispielsweise relationale Datenbanken, Dateisysteme oder auch speziell entwickelte RDF-Repositories externer Projekte von Sesame angesprochen und genutzt werden. Ihre unterschiedlichen, physischen Strukturen werden vom RAL homogenisiert, so dass Sesame intern mit einheitlichen Datenstrukturen arbeiten kann. Bereits in der ersten Version wurde mit der *RDF Query Language (RQL)* eine einfache Anfragesprache unterstützt, die als Vorgänger von SPARQL angesehen werden kann.

Mit dem Release der zweiten Version von Sesame hat sich das zuvor beschriebene Architektur-Konzept leicht verändert. Das Sesame-Framework bietet seitdem zwei elementare Benutzerschnittstellen. Das *Storage And Inference Layer (SAIL)* stellt die systemnahen Funktionalitäten bereit und schafft die Abstraktionsebene zum verwendeten Persistenz- und Inferenzverfahren. Konzeptionell entspricht es damit weitestgehend dem ursprünglichen RAL. Demgegenüber stellt die Repository API die eigentliche Programmierschnittstelle dar und bietet Funktionalitäten, mit denen RDF-basierte Daten verarbeitet werden können. Insgesamt stellt Sesame damit mittlerweile ein vollständiges Framework zur Wissensverarbeitung dar und fokussiert nicht mehr nur auf die Speicherung des Wissens. Aufgrund des gebotenen Funktionsumfangs und der stetigen Weiterentwicklung zählt es zu den beliebtesten Frameworks des Semantic Webs.

[41] Deduktion durch Reasoner: *Pellet* [PS04], *Racer Pro* [HM03], *Fact++* [TH05], *HermiT* [SMH08] [HMW12].
[42] http://www.rdf4j.org/
[43] http://www.aduna-software.com/

3.3.2 Jena

Das Ziel der Entwicklung von Jena[44] war es, eine Java API zu schaffen, mit der eine unkomplizierte Entwicklung Semantic Web basierter Anwendungen möglich ist [McB02]. Jena unterstützt daher sämtliche Sprachebenen des Semantic Webs und kann modelliertes Wissen persistieren. Die erste Version von Jena wurde im Jahr 2000 veröffentlicht und von den Hewlet Packard Labs (HP) in Bristol (UK) entwickelt. Drei Jahre später erschien mit Jena2 eine vollständig überarbeitete Version, deren technischen Konzepte sich auch noch im aktuellen Release 2.13 (März 2015) des frei verfügbaren Frameworks finden. Seit 2010 ist das gesamte Projekt der Apache Software Foundation angegliedert. Die HP Labs haben sich seitdem aus der aktiven Entwicklung weitestgehend zurückgezogen. Eine Besonderheit von Jena sind die gebotenen Möglichkeiten, modellierte Fakten zu speichern. Sie können vom Jena-Framework entweder in nahezu beliebigen, relationalen Datenbanken (SDB) oder in einer nativen Eigenentwicklung (TDB) dateibasiert verwaltet werden. Die Jena-API abstrahiert in beiden Fällen den Zugriff mit dem zentralen Graph-Layer, auf dessen Basis auch beliebige, andere Persistenzverfahren entwickelt werden können [WSK+03]. Das Graph-Layer ist allerdings für den Anwender der Jena API nicht transparent. Ihm ist noch einmal das Model-Layer übergeordnet, mit dessen Schnittstellen der Anwender bzw. Entwickler arbeitet. Hierüber können dann auch in SPARQL formulierte Suchanfragen gestellt werden, die Jena mit der frameworkeigenen Engine ARQ verarbeitet.

Neben der reinen Wissensverarbeitung- und Speicherung unterstützt das Jena-Framework auch Inferenzen. Das Framework selbst stellt drei unterschiedlich mächtige Reasoner-Implementierungen bereit, die direkt genutzt werden können. Sie werten bereits transitive Relationen sowie die sich aus der Modellierung mit RDFS ergebenen Strukturen aus. Zusätzlich können die Reasoner individuelle Regeln auswerten, die hierfür in Java-Methoden kodiert werden müssen[CDD+04]. Ist ein größerer Leistungsumfang erforderlich, können aber auch problemlos etablierte Reasoner in das Framework eingebunden und genutzt werden.

3.3.3 OWL API

Die OWL API[45] rundet den Einblick in die verfügbaren Frameworks zur Wissensverarbeitung ab. Ähnlich wie bei Sesame und Jena reicht auch die Entwicklung dieses Frameworks mittlerweile mehr als zehn Jahre zurück. Die Vorstellung des initialen Release erfolgte 2003 und wurde seitdem stetig weiterentwickelt, sodass inzwischen die Version 4.0.1 verfügbar ist (März 2015). Die fortschreitende Entwicklung resultierte maßgeblich aus den kontinuierlichen Veränderungen der damals noch jungen OWL-Spezifikation, an der sich dieses Framework sehr stark orientiert [HB11]. Die Namensgebung sämtlicher Schnittstellen und Methoden dieses Frameworks entsprechen daher nahezu ausnahmslos den vorgegebenen Konventionen.

[44] http://jena.apache.org
[45] http://owlapi.sourceforge.net/

Hervorgegangen ist die OWL API aus dem WonderWeb Projekt[46], dessen eigene API eng mit der initialen und noch sehr abstrakten OWL-Syntax verknüpft ist. Die ursprüngliche Syntax vereinte eine frameorientierte Darstellung (Abschnitt 2.2.1) mit einem axiombasierten Ansatz [HBN07]. Diese Besonderheit führte zu einem Alleinstellungsmerkmal der heutigen OWL API. Gegenüber allen anderen Semantic Web Implementierungen greift sie zur internen Darstellung der Fakten nicht auf die übliche Darstellung eines Statements durch Subjekt, Prädikat und Objekt (Triple) zurück. Der stattdessen verfolgte Ansatz stützt sich auf die aktuelle OWL2 Spezifikation, einem Nachfolger von OWL [MPSP+09]. In dessen Kontext wird eine Ontologie als Zusammenfassung mehrerer Axiome und Annotationen interpretiert [HB09]. Im Gegensatz zur Triple-Darstellung führt dies zu einer abstrakteren Sichtweise. Die Repräsentation der Fakten kann dadurch nicht nur kompakter erfolgen, sondern auch einfacher verwaltet werden [HB11]. Ein Export in gängige Formate (z.B. RDF/XML-Darstellung) ist jedoch möglich und damit ein Austausch zwischen unterschiedlichen Frameworks nicht ausgeschlossen. Eine weitere, nennenswerte Unterscheidung gegenüber anderen APIs liegt in den zwei unterstützten Arbeitsweisen der eingesetzten Reasoner. Ein Reasoner kann die Inferenzen entweder sofort bei einer Datenveränderung berechnen (incremental reasoning) oder zunächst zurückstellen und schließlich in einem Batch ausführen. Gerade die letzte Option ist für Wissensbasen mit großem Datenbestand interessant.

3.3.4 Potentielles Regelwissen aus universellen sowie biomedizinischen Wissensressourcen

Eine Grundvoraussetzung für die motivierte, logikunterstützte *Pathway Prediction* aus Textdaten ist Regelwissen. Erst wenn es mit den Pathways biologischer Netzwerke in Verbindung gebracht wird, sind weitere Schlussfolgerungen möglich. Neben der eigenständigen Formulierung des Regelwissens besteht die Möglichkeit, dieses existierenden Wissensressourcen zu entnehmen. Die Wahl des jeweils genutzten Regelwissens sollte dabei von den extrahierten Pathway-Daten abhängig gemacht werden. Insbesondere im Kontext der Netzwerkrekonstruktion aus Textdaten könnte eine schrittweise Verfeinerung sinnvoll sein. Anfänglich wird allgemeineres Regelwissen zur Verdichtung der zumeist umfangreich extrahierten Daten genutzt, bevor später domänenspezifischere Zusammenhänge betrachtet werden. Die nachfolgend präsentierten Wissensressourcen decken daher eine entsprechende Bandbreite ab.

3.3.4.1 Wordnet

Wordnet[47] ist ein strukturiertes Nachschlagewerk der englischen Sprache. Es umfasst alle relevanten Wörter, ihre wesentlichen lexikalischen Eigenschaften sowie einfache semantische Relationen. Sie setzen verschiedene Wörter in Beziehung. Der Grundstein für dieses Projekt wurde

[46] http://www.cs.ox.ac.uk/ian.horrocks/Projects/wonderweb.html
[47] http://wordnet.princeton.edu/

von George A. Miller am Cognitive Science Laboratory der Princeton University (USA) gelegt [MBF+90]. Seitdem hat es sich zu einem Standard im Bereich der (Computer-)Linguistik entwickelt. Das in Wordnet enthaltene Wissen wurde anfänglich in nativen Datenstrukturen organisiert, in denen es auch heute noch distribuiert wird. Erst später erfolgte eine Abbildung auf eine explizite Ontologie-Struktur, die auch in RDF/OWL zur Verfügung steht. Bis heute ist Wordnet frei verfügbar und enthält in der aktuellen Version über 150.000 Begriffe, die in *Synsets* organisiert sind. Ein Synset fasst jeweils mehrere Begriffe zusammen, die synonym verwendet werden können. Zusätzlich werden die Begriffe anhand ihrer Wortarten gegliedert und daher in speziellere *NounSynsets*, *VerbSynsets* usw. unterschieden. Der entscheidende Informationsgehalt geht jedoch nicht von dieser Strukturierung aus, sondern von zusätzlich definierten Relationen. Insbesondere die in Wordnet enthaltenen Hyperonym- und Hypernym-Beziehungen (Abschnitt 2.2.1) können Inferenzen ermöglichen. Sie spannen eine Taxonomie auf und definieren so eine implizite *is-a*-Relation. Sie eignet sich sehr gut für eine initiale Verdichtung rekonstruierter Netzwerke, da verschiedene Knoten oder Kanten auf einen zusammenfassenden Stamm zurückgeführt werden können. Dies trifft insbesondere auf Protein-Interaktionsnetzwerke zu. Ihre Kanten repräsentieren eine bestimmte Interaktion im Netzwerk (z.B. regulate) häufig mit unterschiedlichen Begriffen.

Einen Ausschnitt aus der umfassenden Ontologie zeigt Abbildung 3.11 für den Eintrag *interact*. Wie jeder andere Eintrag in Wordnet auch, wird das Verb durch einen eindeutigen Knoten in der Ontologie repräsentiert (*WN30-202376958*). Der Knoten wird nachfolgend zur Modellierung sämtlicher Eigenschaften des Verbs *interact* genutzt. Eine besondere Bedeutung kommt dabei dem *label*-Property zu, das dem eindeutigen Repräsentanten des Verbs die entsprechende Zeichenkette zuordnet. Zusätzlich lassen sich anhand der Properties *hypernym* und *hyponym* die in Wordnet definierten semantischen Relationen erkennen. Sie verweisen auf Ober- und Unterbegriffe des Verbs, die eine Verdichtung rekonstruierter Netzwerke ermöglichen können. Referenziert werden in diesem Kontext wiederum deren eindeutige Repräsentanten. Von den in Wordnet verfügbaren Einträgen wurden zwei exemplarisch herausgegriffen und in Abbildung 3.11 übernommen. Zur Verdeutlichung ist den entsprechenden Repräsentanten über das label-Property auch noch einmal ihre Zeichenkette zugeordnet. Die von Wordnet verwaltete Taxonomie wird so anhand des Verbs *interact* leicht ersichtlich. Neben dieser elementaren Taxonomie lässt sich an der Ontologie-Struktur auch die grundsätzliche Gliederung Wordnets in Synsets wiedererkennen. Sie erfolgt über die *type*- und *word*-Prädikate und lässt erkennen, dass dem Begriff *interact* korrekterweise ein VerbSynset zugeordnet ist. Zusätzlich ist jedem Begriff mit dem *comment*-Property eine kurze Beschreibung zugewiesen. Anhand der verschiedenen Namespaces (Abschnitt 2.2.2.2) lassen sich in Abbildung 3.11 außerdem die in Wordnet definierten Properties (*wn*) von Allgemeingültigeren (*rdf, rdfs*) unterscheiden.

Der Zugriff auf das im Internet verfügbare Wordnet erfolgt standardmäßig über eine webbasierte Oberfläche[48]. Darüber hinaus stehen zusätzlich eine Reihe anderer Schnittstellen zur Verfügung, die alle gängigen Programmiersprachen und Protokolle abdecken. Sie alle ermöglichen jedoch nur den Zugriff auf die nativen Wordnet-Datenstrukturen und setzen teilweise sogar eine lokal installierte Wordnet-Instanz voraus. Die in RDF/OWL verfügbare Wordnet-Ontologie

[48] http://wordnetweb.princeton.edu/perl/webwn

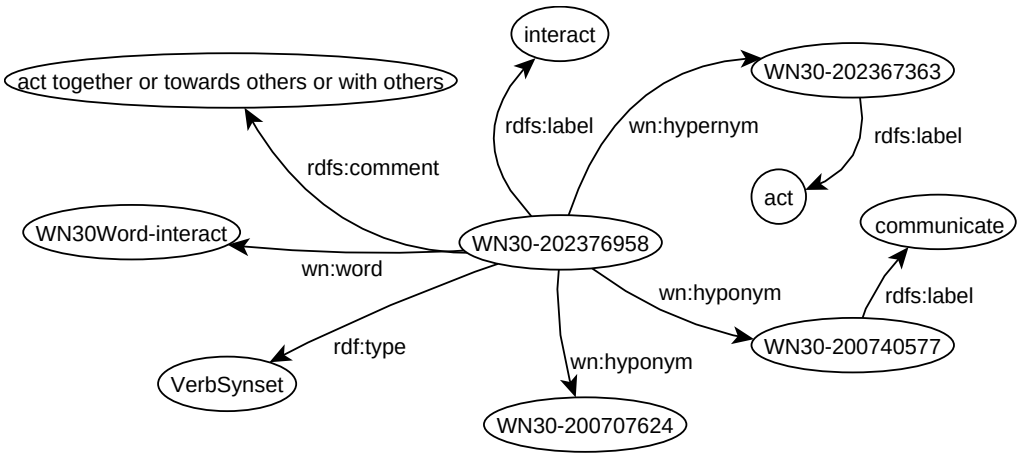


Abbildung 3.11: Ausschnitt der Wordnet-Ontologie - Verb *interact* und dessen Taxonomie

spielte zunächst nur eine untergeordnete Rolle und gewinnt erst allmählich an Bedeutung.

3.3.4.2 GeneOntology

Die GO[49] ist ein prominenter Vertreter biomedizinischer Wissensressourcen. Das Projekt wurde im Jahr 1998 gestartet und diente der Erforschung des Genoms von drei Organismen: Fruchtfliege, Maus und Hefe. Ihr primäres Ziel ist es, eine einheitliche Grundlage für die Repräsentation und Darstellung von Genen und Genprodukten aller existierenden Lebewesen zu schaffen [ABB+00]. Da bis zum Aufkommen der GO keine einheitliche, standardisierte Terminologie existierte, war ein Wissensaustausch zwischen unterschiedlichen Forschungseinrichtungen nur bedingt möglich. Dies führte stellenweise dazu, dass identische Sachverhalte bei unterschiedlichen Lebewesen auch unterschiedlich benannt und gehandhabt wurden.

Die GO bildet heute in dieser Wissensdomäne de facto den Standard. Bereits im Jahr 2005 umfasste die Ontologie fast 25.000 Einträge. Ihr Umfang wächst seitdem stetig weiter an. Zudem kann mit Hilfe zusätzlicher *Gene-Ontology-Annotations (GOAs)* eine Verknüpfung mit anderen Ontologien der Bioinformatik erreicht werden [CMB+04]. Spezielle Algorithmen erlauben die Identifikation korrelierender Einträge und definieren Annotationen, mit denen die in *UniProt*, *Ensembl* und anderen Datenbanken beschriebenen Gen-Produkte mit den Einträgen in der Gene Ontology in Verbindung gebracht werden. Mittlerweile existieren über elf Millionen derartiger Annotationen, von denen fast eine halbe Million von Experten manuell überprüft und verifiziert wurden [SAR+07]. Dies ist ein erster Schritt auf dem Weg, die Vielfalt der existierenden, biomedizinischen Wissensressourcen zu integrieren. Die ursprüngliche Motivation erstreckte sich jedoch nicht nur auf die Vereinheitlichung der genutzten Terminologie. Mit dem GO-Projekt wurde eine komplexe Systemlandschaft geschaffen, die sich aus mehreren Komponenten zusammensetzt. Besonders hervorzuheben ist das webbasierte Werkzeug *AmiGO*, mit dem Suchanfragen an die GO gestellt und die erzielten Ergebnisse visualisiert werden können

[49] http://geneontology.org/

[CIM+09]. Die Abbildung 3.12 zeigt ein Teil-Ergebnis einer mit Amigo exemplarisch formulierten Suchanfrage zu *MUPP-1*. Den dargestellten Zusammenhängen kann auch die grundsätzliche Struktur der GO entnommen werden. Jeder Eintrag in der GO wird thematisch einer der folgenden Gruppen zugeordnet:

- zelluläre Komponenten
- molekulare Funktionen
- biologische Prozesse

Eine darüber hinaus gehende, interne Strukturierung der Einträge gewährt zudem eine konsistente und weitestgehend redundanzfreie Verwaltung aller Einträge. Jeder enthaltene Eintrag wird daher, analog zur *Wordnet*-Ontologie, durch eine eindeutige *ID* repräsentiert. Über sie erfolgt dann wiederum die Zuweisung aller weiteren Eigenschaften. Dies umfasst beispielsweise die Up- und Down-Regulation einzelner Gene (Abschnitt 2.1.3). Außerdem existiert eine *is-a*-Relation, die den Einträgen in der GO einen Typ zuweist (z.B. Protein, Gen usw.). Im Rahmen der *Pathway Prediction* aus Textdaten könnte diese Information zur Validierung genutzt werden. Offensichtlich unplausible Erkenntnisse könnten somit von der Rekonstruktion biologischer Netzwerke ausgenommen werden.

3.3.4.3 Open Biomedical Ontologies

Das Ziel der *Open Biomedical Ontologies (OBO)*[50] ist es, mehrere Ontologien aus dem Bereich der Biomedizin zusammenzuführen. Aufgrund existierender Erfahrungswerte (z.B. GO) wurde dies allerdings nicht mit Annotationen gelöst. Das Konzept der OBO sieht stattdessen vor, bereits existierende Ontologien zunächst strukturell zu überarbeiten und neue Ontologien direkt mit einheitlichen Strukturen aufzubauen [SAR+07]. Für alle in den OBOs zusammengefassten Ontologien gilt damit:

- sie nutzen eine gemeinsame Syntax
- sie überschneiden sich nicht (Orthogonalität)
- sie sind frei verfügbar

Nachdem das OBO-Projekt im Jahr 2001 gestartet wurde, umfasst es mittlerweile über sechzig einzelne Ontologien. Jede von ihnen erfüllt die drei Anforderungen. Eine Weiterentwicklung des OBO-Konzepts kann in der *OBO-Foundry* gesehen werden, die an die beteiligten Ontologien noch weiterreichendere Anforderungen stellt. Sie fordern beispielsweise eine konsequente, gemeinschaftliche Entwicklung der Ontologien, die Begrenzung auf eine bestimmte Wissensdomäne, die explizite Unterstützung einer Feedback-Funktion durch den User und insbesondere die Verwendung einheitlicher Relationen. Damit letzteres sichergestellt werden kann, wurde mit der *OBO Relation Ontology (RO)* eine eigenständige Ontologie geschaffen, die ausschließlich für die Verwaltung dieser Relationen zuständig ist [SCK+05]. Sowohl beim Aufbau neuer

[50] http://www.obofoundry.org/

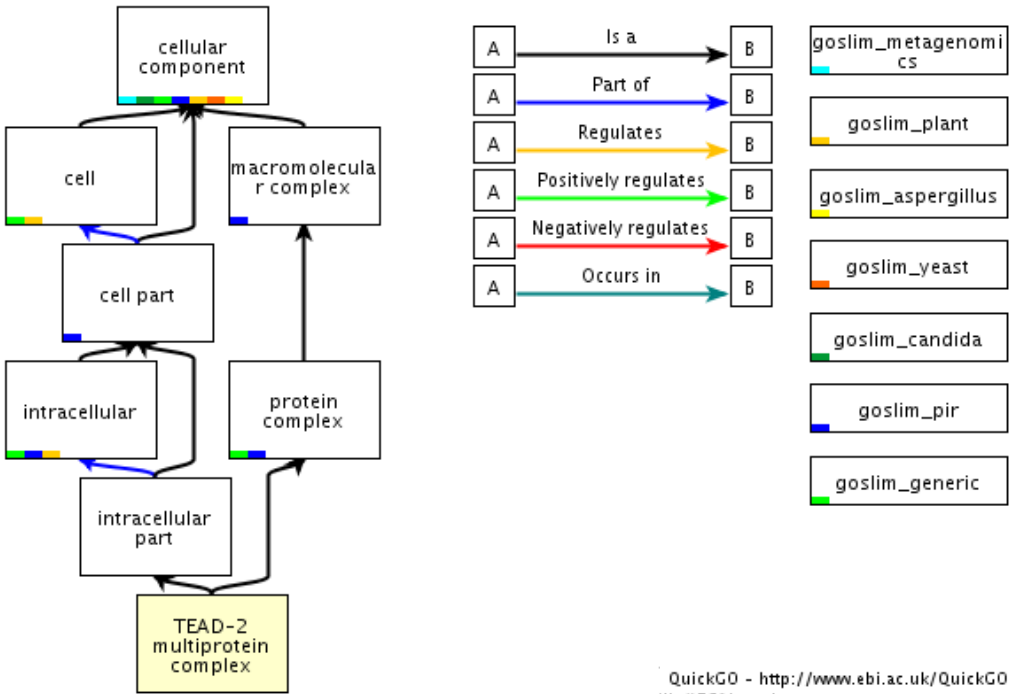


Abbildung 3.12: Verfügbare Informationen in der GO zum Protein *MUPP-1*

als auch bei der Erweiterung bestehender Ontologien kann hierauf zugegriffen werden. Eigens für die Bearbeitung der beteiligten Ontologien steht das dafür geschaffene Tool *OBO-Edit* bereit, mit dem jede Ontologie den benannten Anforderungen entsprechend modifiziert werden kann. Darüber hinaus kann das Tool zusätzlich für Suchanfragen genutzt werden und bringt einen eigenen Reasoner mit [DRHH+07].

Entgegen der Erwartung basiert die Wissensrepräsentation in den OBOs auf einem proprietären Format und nutzt keine etablierten Standards. Allerdings wurde mit *Biogateway*[51] eine eigenständige Wissensbasis geschaffen, die das in den OBOs enthaltene Wissen in RDF repräsentiert und zusätzlich mit weiteren Wissensbasen kombiniert [ABE+08] [ABE+09]. Neben der GO umfasst sie auch SwissProt [BA00]. Biogateway eröffnet schließlich auch die Möglichkeit, mit standardisierten SPARQL-Anfragen auf das biomedizinische Wissen effizient zuzugreifen. Nach eigener Aussage markiert *Biogateway* damit eine neue System-Gattung, die die Konzepte des Semantic Webs in den Dienst der System-Biologie stellt [ABV+11].

3.4 Zusammenfassung

In diesem Kapitel wurden verschiedene Arbeiten präsentiert, die einen Bezug zur motivierten, logikunterstützten *Pathway Prediction* aus Textdaten haben. Im Abschnitt 3.1 wurden zunächst

[51] http://www.semantic-systems-biology.org/biogateway

	ABNER	Whatizit	LiMPET	Enju	GATE	UIMA
technischer Fokus (Realisierung)	NER (CRF)	mehrere (Module)	Segmentierung, Tokenisierung, NER (openNLP)	POS, Tokenisierung, Segmentierung (Grammatik)	umfassend (Komponenten)	umfassend (Komponenten)
Schnittstelle	GUI, API	GUI, Webservice	keine	Webservice	GUI	Webservice
biomedizinische Spezialisierung	Proteine, Gene, DNS	umfassend (je Modul)	enzymatische Reaktionen	keine (PAS Extraktion)	keine	keine (aber *cTAKES*)
Repräsentation extrahierter Daten	nativ	nativ	flat file (txt)	XML	unbekannt	unbekannt
Netzwerk-Rekonstruktion möglich	nein	nein	ja	nein	nein	nein
Open Source	ja	nein	ja	nein	ja	ja
Support	unbekannt	unbekannt	E-Mail (Entwickler)	E-Mail (Entwickler)	FAQ, kommerziell	FAQ, Community

Tabelle 3.2: Gegenüberstellung zur *Pathway Prediction* einsetzbarer Textmining-Algorithmen. Besonderheiten sind grau hinterlegt.

	VANESA	ANDSystem	PathPred	UM-PPS	Pathways Tools	STRING
Anwendung	GUI	GUI	webbasiert	webbasiert	Framework	webbasiert
Verfahren	integrativ	Textmining	integrativ	integrativ	integrativ	integrativ (inkl. TM)
Deduktion	nein	nein	ja (Reaktionsmuster)	ja (chemische Strukturen)	ja (Regeln, Heuristiken)	ja (Orthologien)
Fokussierung	verschiedene	verschiedene	enzymatische Reaktionen (metabolische Pathways)	mikrobieller Katabolismus (metabolische Pathways)	metabolische Pathways	Protein-Interaktionen
Datenquellen	Datawarehouse (DAWIS-M.D.)	Medline	KEGG REACTION & Compound, RPAIR	UM-BBD	GenBank, MetaCyc	verschiedene (KEGG,HPRD, Medline […])
Entwicklung	Universität Bielefeld	PBSoft	University Kyoto	University of Minnesota	Stanford Research Institute	EMBL, SIB, CPR
Open Source	ja	nein	nein	nein	nein	nein
Support	Sourceforge (Entwickler)	Tutorial (Website)	Webformular	E-Mail-Verteiler	FAQ, Tutorials, User-Group	Dokumentation (online)

Tabelle 3.3: Gegenüberstellung verschiedener Netzwerk-Rekonstruktionssysteme. Besonderheiten sind grau hinterlegt.

ausgewählte Textmining-Algorithmen der Bioinformatik vorgestellt. Sie können biomedizinische Textdaten analysieren und wertvolle Informationen für die Rekonstruktion biologischer Netzwerke aus ihnen gewinnen. In diesem Zusammenhang wurde die Bandbreite ihres Leistungsspektrums deutlich, das in Tabelle 3.2 strukturiert zusammengefasst ist. Lediglich die Ergebnisse eines Algorithmus (LiMPET) sind aussagekräftig genug, um daraus unmittelbar ein biologisches Netzwerk rekonstruieren zu können. Generell erscheint daher die mehrfache Analyse eines Textes mit unterschiedlich spezialisierten Algorithmen sinnvoll. Diese entscheidende Erkenntnis sollte daher auch in die Konzeption der in dieser Arbeit motivierten *Pathway Prediction* einfließen.

Im Abschnitt 3.2 wurden Systeme diskutiert, die eine Rekonstruktion biologischer Netzwerke unterstützen. Ihre wesentlichen Eigenschaften sind in Tabelle 3.3 noch einmal übersichtlich zusammengefasst. Von den präsentierten Systemen unterstützen lediglich VANESA und ANDSystem die Rekonstruktion verschiedener, biologischer Netzwerke. Während PathPred, UM-PPS und Pathway-Tools auf die Rekonstruktion metabolischer Netzwerke spezialisiert sind, konzentriert sich STRING auf Interaktionsnetzwerke. Die Systeme entnehmen die erforderlichen Daten entweder (teil-)integrierten Datenbanken (VANESA, PathPred, UM-PPS, Pathway Tools) oder extrahieren sie aus biomedizinischen Textdaten (ANDSystem). STRING kombiniert beides und lässt aus Textdaten gewonnene Ergebnisse ebenfalls in die Integration einfließen. Insgesamt spiegeln die präsentierten Systeme damit die beiden etablierten Verfahren wider, die Rekonstruktion biologischer Netzwerke zu automatisieren. Erstaunlich ist, dass die Rekonstruktion in vier Systemen bereits durch verschiedene Deduktionsansätze unterstützt wird. Die betreffenden Systeme basieren jedoch ausschließlich auf integrierten Daten und nutzen beispielsweise zuvor extrahierte Reaktionsmuster oder die chemische Struktur der Metaboliten zur deduktiven Vorhersage aus. Die Verfahren sind damit sehr individuell und können kaum auf andere, biologische Netzwerke übertragen werden. Die Tabelle 3.3 zeigt zudem deutlich, dass die Deduktion bisher noch nicht im Kontext der Rekonstruktion aus Textdaten genutzt wird.

Zum Ende dieses Kapitels wurden im Abschnitt 3.3 Systeme präsentiert, mit denen deduktiv motivierte Wissensbasen aufgebaut werden können. Sie alle basieren auf dem Konzept des Semantic Web, das eine potentielle Alternative zu deduktiven Datenbanken ist. Technisch unterscheiden sich die drei momentan bekanntesten Frameworks nur in Nuancen. Unterschiede zeigen sich hauptsächlich in der Leistungsfähigkeit der von ihnen gestellten Reasoner. Ein Überblick auf potentielles Regelwissen, mit dem die Netzwerkrekonstruktion aus Textdaten sinnvoll unterstützt werden könnte, schließt dieses Kapitel ab. Die präsentierten Ressourcen decken eine große Bandbreite ab und enthalten sowohl allgemeingültigere als auch sehr spezifische, biomedizinische Zusammenhänge.

4 Konzept- und Design

Ausgehend von den zu Beginn dieser Arbeit formulierten Zielen (Abschnitt 1.2) sowie unter Berücksichtigung existierender *Pathway Prediction*-Verfahren (Abschnitte 3.1 & 3.2) erfolgt in diesem Kapitel eine schrittweise Systemkonzeption. Ihr geht eine Formulierung konkreter Anforderungen voraus (Abschnitt 4.1). Hieran schließt sich der Aufbau einer flexiblen Datenstruktur an, die sowohl zur Repräsentation der anfänglichen Textdaten als auch der rekonstruierten Netzwerke dient. Die konzeptionelle Gliederung des komplexen Systems in mehrere *Pathway Prediction Steps (PPSs)* steht anschließend im Fokus (Abschnitt 4.3). Sie können die zuvor präsentierte Datenstruktur verarbeiten und bieten gemeinsam alle erforderlichen Funktionalitäten für die motivierte Netzwerkrekonstruktion. Dies umfasst auch die Möglichkeit, die in der konzipierten Datenstruktur repräsentierten Daten in verschiedenen Persistenzlayern speichern zu können (Abschnitt 4.4). Eine Grundvoraussetzung, um rekonstruierte Netzwerke in die Deduktionskomponente einfließen zu lassen.

4.1 Anforderungen an eine logikunterstützte Rekonstruktion biologischer Netzwerke

Die zentralen Anforderungen an den zu realisierenden Prototyp leiten sich aus der formulierten Zielstellung dieser Arbeit ab (Abschnitt 1.2, Abbildung 1.1). Darüber hinaus zeigte die Diskussion verwandter Arbeiten im Kapitel 3 zusätzliche Anforderungen auf. Sie werden nachfolgenden zusammenfassend formuliert und können zum Ende dieser Arbeit mit dem geschaffenen Prototyp abgeglichen werden.

Anforderung 1: *Die Vorhersage biologischer Netzwerke wird mit einer deduktiv motivierten Datenbank unterstützt.*

Die eingehende Recherche zu Beginn dieser Arbeit zeigte, dass der Deduktion im Kontext der Netzwerkrekonstruktion bisher kaum Beachtung geschenkt wird. Mit einer prototypischen Implementierung soll daher ihr potentieller Mehrwert anhand eines MPDZ/MUPP1-Proteinnetzwerks geprüft werden.

Anforderung 2: *Der Prototyp bietet die Möglichkeit verschiedene Algorithmen zu kombinieren, um möglichst präzise Netzwerke aus Textdaten rekonstruieren zu können.* Das breite Spektrum verfügbarer TM-Ressourcen soll unkompliziert zur Analyse ausgewählter Textdaten in *FraMeTex* genutzt und die Ergebnisse gewinnbringend zusammengeführt werden können.

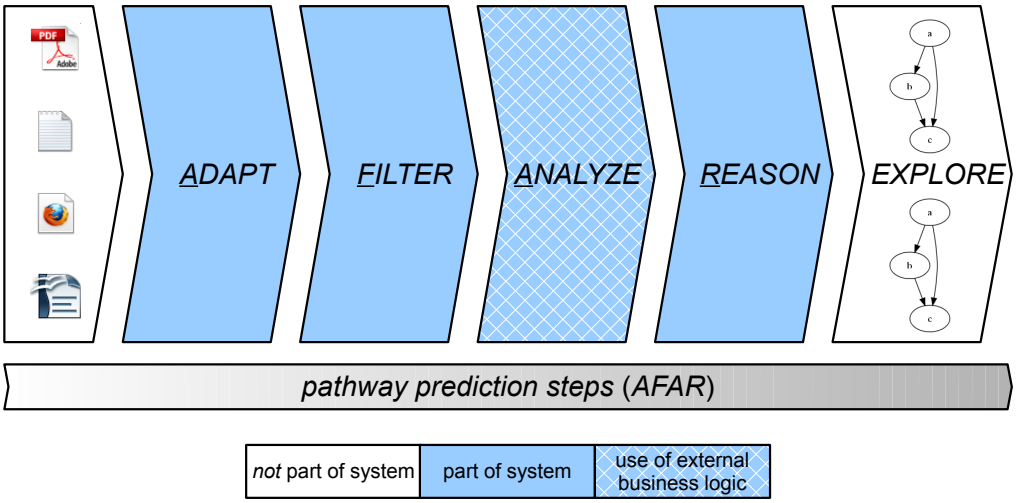


Abbildung 4.1: Ganzheitlicher Ansatz der motivierten *Pathway Prediction* gliedert sich in PPSs

Anforderung 3: *Mit dem resultierenden Prototyp können verschiedene, biologische Netzwerktypen rekonstruiert werden.*

Die Mehrheit existierender *Pathway Prediction* Tools fokussiert auf die Rekonstruktion eines spezifischen, biologischen Netzwerks (Tabelle 3.3). Eine derartige Restriktion soll für die prototypische Implementierung nicht gelten. Der potentielle Mehrwert einer deduktiven Datenbank kann damit anhand verschiedener Netzwerktypen beurteilt werden.

Anforderung 4: *Die konzipierte Deduktionskomponente unterstützt Inferenzen in beliebig rekonstruierten Netzwerken.*

Trotz der Fokussierung auf die textminingbasierte Netzwerkrekonstruktion werden auch Schlussfolgerungen in anderweitig rekonstruierten Netzwerken ermöglicht. Voraussetzung ist die Einhaltung der definierten Systemschnittstellen. Damit können auch integrative Rekonstruktionsverfahren das Potential der Deduktionskomponente nutzen.

Anforderung 5: *Die prototypische Implementierung bedient sich, so weit möglich, bereits verfügbarer Algorithmen und Systeme.*

Der Fokus des zu realisierenden Prototyps liegt auf der konzeptionellen Verknüpfung der *Pathway Prediction* aus Textdaten mit einer deduktiven Datenbank. Es soll daher weder ein eigener TM-Algorithmus noch eine entsprechende Datenbank entwickelt werden.

Anforderung 6: *Das für Inferenzen in der deduktiven Datenbank erforderliche Regelwerk kann aus existierenden Wissensressourcen übernommen oder eigenständig formuliert werden.* Die in der deduktiv motivierten Datenbank persistierten Netzwerke werden mit Regeln in Verbindung gebracht und es wird versucht, weitere Zusammenhänge abzuleiten. Im Idealfall wird sowohl eine flexible Einbindung existierenden Regelwissens, als auch dessen individuelle Formulierung unterstützt.

Anforderung 7: *Die motivierte Pathway Prediction aus Textdaten verfolgt einen ganzheitlichen Ansatz.*

Die Rekonstruktion biologischer Netzwerke wird als komplexer Prozess aufgefasst, der auch unterstützende Komponenten umfasst[1]. Der gesamte Prozess wird in PPSs gegliedert, die unabhängig genutzt oder zur motivierten *Pathway Prediction* zusammengestellt werden (*AFAR*-Konzept, Abbildung 4.1).

Anforderung 8: *Die Implementierung der PPSs wird vereinheitlicht und erfolgt plattformunabhängig.*

Für die Implementierung der PPSs wird ein gemeinsames, technisches Fundament geschaffen. Es definiert ihre Schnittstellen und abstrakten Verarbeitungsroutinen. Die Routinen können von jedem PPS weiter spezialisiert und an konkreten Anforderungen angepasst werden. Die Implementierung erfolgt mit Java, damit der Prototyp auf beliebigen Systemen einsetzbar ist.

Anforderung 9: *Das Verhalten der modularen PPSs kann über Konfigurationsdateien unkompliziert individualisiert werden.*

Die modularen PPSs können mit standardisierten Properties konfiguriert werden. Sie sind in Dateien zusammengefasst und können die Datenverarbeitung eines PPS beeinflussen.

4.1.1 Berücksichtigung existierender Algorithmen und Systeme

Für die prototypische Implementierung der motivierten *Pathway Prediction* soll möglichst auf existierende Systeme und Algorithmen zurückgegriffen werden. Eine umfassende Recherche zu Beginn dieser Arbeit konzentrierte sich daher maßgeblich auf die bestehenden Möglichkeiten biomedizinische Textdaten zu analysieren (Abschnitt 3.1), biologische Netzwerke zu rekonstruieren (Abschnitt 3.2) und Schlussfolgerungen in persistierten Daten zu unterstützen (Abschnitt 3.3). Die Ergebnisse der Recherche lieferten aufschlussreiche Erkenntnisse und flossen unmittelbar in das Systemkonzept ein.

Es zeigte sich, dass bereits mehrere Systeme für die Rekonstruktion biologischer Netzwerke verfügbar sind. Sie könnten theoretisch mit einer deduktiven Wissensbasis verknüpft werden, um die motivierte *Pathway Prediction* prototypisch umzusetzen. Von den existierenden Systemen unterstützt allerdings nur ANDSystem die geforderte Netzwerkrekonstruktion aus Textdaten (Tabelle 3.3). Das System ist jedoch weder frei verfügbar, noch ist die verfolgte Rekonstruktion transparent. Sie kann zudem nur bedingt beeinflusst werden. Die Anforderung möglichst präzise sowie verschiedene Netzwerke rekonstruieren zu können, führt daher an einer gewinnbringenden Kombination unterschiedlich spezialisierter TM-Ressourcen (Tabelle 3.1) nicht vorbei. Die in der Bioinformatik zur Verfügung stehende Bandbreite ist enorm und die Algorithmen fast immer frei verfügbar. Die Konzeption des Systems sollte daher den Zusammenschluss ihrer individuellen Analyse-Ergebnisse (z.B. Protein-Identifikationen sowie Interaktionen) berücksichtigen.

[1] Die Visualisierung wird aufgrund ihrer Komplexität nicht berücksichtigt.

Zum Schlussfolgern in biologischen Netzwerken bieten sich zwei unterschiedliche Konzepte an. Einerseits steht mit XSB ein vollwertiges, deduktives Datenbanksystem zur Verfügung (Abschnitt 2.2.3.2), andererseits bietet die Semantic Web Technologie eine potentielle Alternative (Abschnitt 2.2.2.2). Im Gegensatz zur deduktiven Datenbank repräsentiert sie die Daten jedoch in graphbasierten Strukturen. Mit Blick auf die zu verarbeitenden, biologischen Netzwerke wird ihnen in dieser Arbeit daher der Vorzug gegeben. Der Aufbau der Deduktionskomponente konzentriert sich damit darauf, rekonstruierte Netzwerke in den graphbasierten Strukturen des Semantic Web zu speichern und mit Regelwissen in Verbindung zu bringen. Dies kann mit drei etablierten Frameworks umgesetzt werden (Abschnitt 3.3). Die Entscheidung für oder gegen eine bestimmte Implementierung ist aus fachlicher Sicht unerheblich und kann frei getroffen werden.

4.2 Repräsentation biologischer Netzwerke und Texte

Der ganzheitliche Ansatz der motivierten *Pathway Prediction* erfordert die Darstellung unterschiedlichster Daten in den einzelnen PPSs. Die Bandbreite reicht von den zu analysierenden Texten bis zu den extrahierten Netzwerken (Abbildung 4.1). Zusätzlich müssen verschiedene Zwischenergebnisse repräsentiert werden. Innerhalb des Systems müssen diese heterogenen Daten auf eine einheitliche Datenstruktur abgebildet werden. Sie ist eine Grundvoraussetzung für eine erfolgreiche Verknüpfung einzelner PPS zur Rekonstruktion biologischer Netzwerke. Die Ausgabe eines PPS dient hierbei als Eingabe des Nachfolgenden (Abbildung 4.1). Das Ergebnis dieser umfassenden Anforderungen ist das *FraMeTex*-Dataset. Vier Komponenten zeichnen diese komplexe Datenstruktur aus:

- *matchValue*. Er identifiziert einen Datensatz innerhalb des gesamten Systems eindeutig. Der Wert wird automatisch generiert und ist eine Teilkomponente des *metaValue*.
- *metaValue*. Sämtliche Informationen über einen Datensatz werden durch ihn repräsentiert. Hierzu gehören auch Statusinformationen, die während der Verarbeitung eines Datensatzes in dessen Meta-Daten automatisch abgelegt werden.
- *keyValue, dataValue*. Sie repräsentieren die eigentlichen Informationen eines Datensatzes und sind technisch identisch. In Abhängigkeit der Funktion bzw. Semantik einer Information wird sie entweder als *keyValue* oder als *dataValue* dargestellt.

Die Abbildung 4.2 zeigt exemplarisch die Repräsentation eines Medline-Eintrages in der konzipierten Datenstruktur. Formal ist das *FraMeTex*-Dataset (δ^F) als 3-Tupel konzipiert:

$$\delta^F = \begin{pmatrix} metaValue \\ keyValue \\ dataValue \end{pmatrix} \tag{4.1}$$

Jede dieser drei Komponenten entspricht einem Feld und ist daher primär als key/value Paar ausgelegt. Jedes Feld kann außerdem eine beliebige Anzahl Tochterfelder referenzieren, die wie-

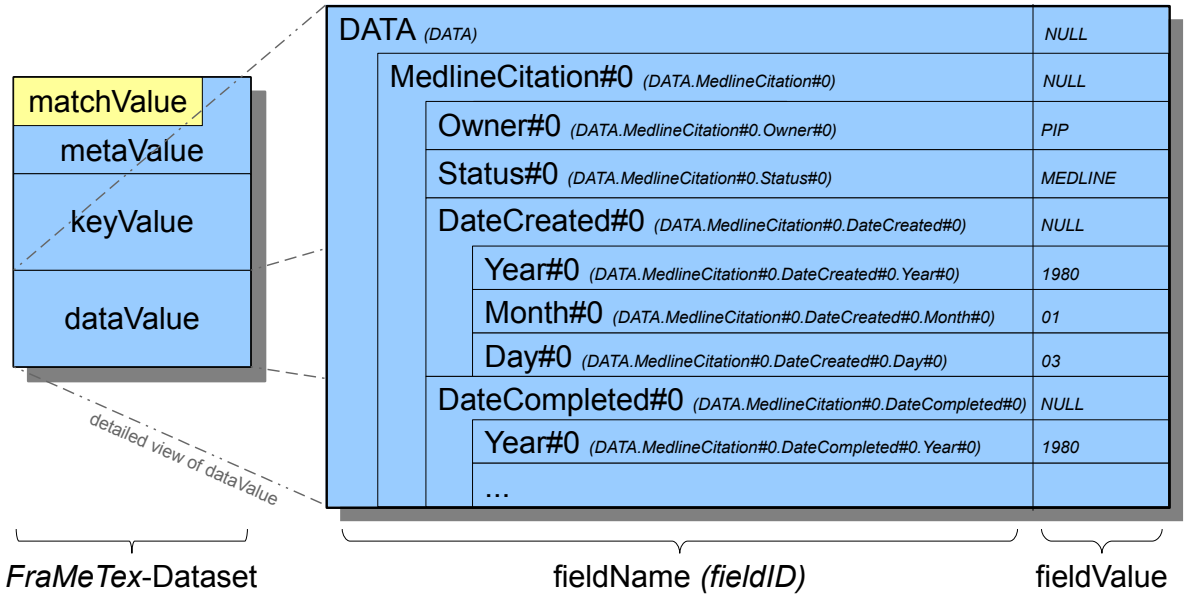


Abbildung 4.2: Repräsentation eines Medline-Eintrags im *FraMeTex*-Dataset (Ausschnitt)

derum auf weitere Tochterfelder verweisen können. Damit können hierarchische Strukturen abgebildet werden (Abbildung 4.2), die jedoch azyklisch[2] sein müssen. Eine iterative Verarbeitung der Feldstruktur kann so nicht in einer endlosen Rekursion münden.

Die Datentypen der drei Felder sind flexibel wählbar und müssen erst beim Instanziieren eines *FraMeTex*-Dataset festgelegt werden. Die gewählte Programmiersprache Java (Anforderung 8) unterstützt dies mit dem Konzept der Generics[3]. Damit kann das *FraMeTex*-Dataset innerhalb eines PPSs, entsprechend der jeweils zu verarbeiteten Daten (Texte, Netzwerke), dynamisch angepasst werden. Die Festlegung des Datentyps gilt beim *key*- und *dataValue* automatisch auch für sämtliche Tochterfelder. Demgegenüber können beim *metaValue* die Datentypen individuell für jedes Tochterfeld festgelegt werden. Die Meta-Daten bieten damit eine größtmögliche Flexibilität, um nahezu beliebige Informationen in ihnen ablegen zu können. Ein essentielles sowie unveränderbares Tochterfeld des *metaValue* ist der *matchValue* (Abbildung 4.2)

Die Flexibilität des *FraMeTex*-Datasets führt zu einer Herausforderung bei dessen Instanziierung, da keine konkreten Konstruktoren im Programmcode definiert werden können. Aus diesem Grund wurde eine *Factory* geschaffen, die diese Aufgabe übernimmt. Die Factory ist ein Design-Pattern, das zur Laufzeit verschiedenste Objekte erzeugen kann [GHJV04]. Sie verantwortet im *FraMeTex*-Dataset die flexible Instanziierung aller Felder mit den jeweils erst zur Laufzeit festgelegten Datentypen. Die konzeptionellen Details des *FraMeTex*-Datasets können dem *Unified Modeling Language (UML)*-Diagramm in Abbildung A.1 entnommen werden.

[2] Ein Feld kann sich selbst nicht als Tochterfeld referenzieren.
[3] Entsprechen parametrisierten Datentypen.

4.2.1 Aufbau einer universellen Datenstruktur

Im vorherigen Abschnitt wurde das flexible *FraMeTex*-Dataset präsentiert. Seine drei Felder (Definition 4.1) werden in diesem Abschnitt nun näher vorgestellt. Sie sind einheitlich strukturiert, um allgemeingültige Verarbeitungsroutinen in den einzelnen PPSs gewährleisten zu können (Anforderung 8). Jedes Feld (κ^F) des *FraMeTex*-Datasets ist als 4-Tupel konzipiert:

$$\kappa^F = \begin{pmatrix} fieldID \\ fieldName \\ fieldValue \\ childFields \end{pmatrix} \tag{4.2}$$

Der *fieldName* repräsentiert den frei wählbaren Namen des Feldes, der *fieldValue* den zugeordneten Wert. Die im Abschnitt 4.2 beschriebene Factory wirkt sich damit in erster Linie hierauf aus. Eine besondere Bedeutung kommt der *fieldID* zu. Sie identifiziert ein Feld innerhalb der komplexen Struktur eines *FraMeTex*-Datasets eindeutig. Gegenüber des Feldnamens spiegelt sie daher sowohl die hierarchische Ebene als auch die exakte Position des Feldes innerhalb dieser Ebene wider. Die beiden Informationen sind für die spätere Persistierung der in einem FraMeTex-Datasets repräsentierten Daten von elementarer Bedeutung. Aus diesem Grund wird die *fieldID* nach den Regeln einer formalen Grammatik (Abbildung 4.3) automatisch generiert. Manuelle Änderungen sind zu keinem Zeitpunkt möglich.

Die in Abbildung 4.2 exemplarisch dargestellten *fieldIDs*, spiegeln die in Medline zur Datenrepräsentation genutzte XML-Struktur wider. Es ist deutlich zu erkennen, dass die Hierarchie-Ebene eines Feldes ausschließlich durch die Konkatenierung mehrerer ⟨*field_structure*⟩ Elemente ausgedrückt wird. Die Position bzw. der Rang eines Feldes innerhalb einer bestimmten Hierarchie-Ebene erfolgt durch die Angabe des ⟨*position*⟩ Elements. Die Nummerierung beginnt hierbei mit der Ziffer *0*, die somit das erste Element einer Ebene identifiziert. So bezeichnet beispielsweise *DATA.MedlineCitation#0* das erste Tochterfeld von *DATA* mit dem Namen *MedlineCitation*. Die Eingliederung eines Feldes in eine bestehende Struktur und damit der Aufbau eines *FraMeTex*-Dataset erfolgt ausschließlich über definierte Methoden. Sie stoßen auch die Generierung der zum Feld korrespondierenden *fieldID* an. Dies schließt auch dessen potentielle Tochterfelder mit ein, deren *fieldIDs* ebenfalls die veränderte Feldstruktur reflektieren müssen. Verwaltet werden sämtliche Tochterfelder von der Komponente *childFields* (Definition 4.2). Sie entspricht einem assoziativen Array, das alle Tochterfelder anhand ihrer Namen identifiziert. Da durchaus mehrere Tochterfelder mit identischen Namen denkbar sind, wird für jeden Eintrag im assoziativen Array eine Liste geführt. Die geordnete Liste spiegelt automatisch die Position gleichnamiger Tochterfelder wider. Die *childFields*-Komponente ist damit für die Berechnung einer *fieldID* entscheidend.

Konzeptionell wird in der prototypischen System-Implementierung zunächst nur der Aufbau einer Feldstruktur berücksichtigt. Nachträgliche Änderungen an einem *FraMeTex*-Dataset sind momentan nicht zu erwarten. Dies spiegeln auch die definierten Methoden wider (Abbildung A.2). Der Aufbau einer Feldstruktur obliegt dem PPS *ADAPT* (Abbildung 4.1). Er transformiert die zu analysierenden Textdaten in ein *FraMeTex*-Dataset und schafft damit die Voraussetzung

⟨*field_id*⟩ ::= ⟨*field_prefix*⟩ | ⟨*field_prefix*⟩ ⟨*field_structure*⟩

⟨*field_prefix*⟩ ::= `META` | `KEY` | `DATA`

⟨*field_structure*⟩ ::= ⟨*field_segment*⟩ | ⟨*field_segment*⟩ ⟨*field_structure*⟩

⟨*field_segment*⟩ ::= `.` ⟨*field_name*⟩ `#` ⟨*position*⟩

⟨*field_name*⟩ ::= ⟨*letter*⟩ | ⟨*letter*⟩ ⟨*field_name*⟩

⟨*position*⟩ ::= ⟨*digit*⟩ | ⟨*digit*⟩ ⟨*position*⟩

⟨*letter*⟩ ::= `A` | `B` | `C` | `...` | `Z` | `a` | `b` | `c` | `...` | `z`

⟨*digit*⟩ ::= `0` | `1` | `2` | `3` | `4` | `5` | `6` | `7` | `8` | `9`

Abbildung 4.3: Formale Grammatik definiert Struktur der *fieldID* (Backus-Naur-Form)

für die Netzwerkrekonstruktion in den nachgelagerten PPSs. Anhand einer konkreten Realisierung für die XML-Struktur in Medline wird das Vorgehen später noch einmal detailliert beleuchtet.

Der formulierten Zielstellung entsprechend soll der Prototyp biologische Netzwerke aus Medline-Abstracts rekonstruieren (Abschnitt 1.2). Die Abbildung der in Medline repräsentierten Daten auf ein *FraMeTex*-Dataset ist daher eine zwingende Voraussetzung. Die drei Felder der Datenstruktur (Definition 4.1) müssen hierfür entsprechend angepasst werden. Die vorgenommenen Anpassungen können Tabelle 4.1 entnommen werden, die im spezialisierten *TextDataset* bereits umgesetzt sind (Abbildung A.1). Die Datenstruktur kann damit unmittelbar genutzt werden. Ihre individualisierten Felder können generell in jedem spezialisierten *FraMeTex*-Dataset genutzt werden. Insbesondere die Meta-Felder finden sich in fast jeder Instanz, unabhängig davon, ob sie Textdaten oder rekonstruierte Netzwerke repräsentiert. In Tabelle 4.1 fällt das *SpecialMetaField* daher auch durch seinen frei wählbaren Datentyp auf. In fast jedem *FraMeTex*-Dataset entspricht ein Großteil der Tochterfelder des *metaValue* diesem Typ. Er wird genutzt, um verschiedenste Informationen in den Meta-Daten ablegen zu können (Abschnitt

Feld	**Verwendung in δ^F**	**Datentyp des *fieldValue***
DefaultField	*keyValue, dataValue*	Text (String)
DefaultMetaField	*metaValue*	Text (String)
SpecialMetaField	*metaValue*	frei wählbar

Tabelle 4.1: Ausgewählte Möglichkeiten der Datenrepräsentation im *FraMeTex*-Dataset (Felder)

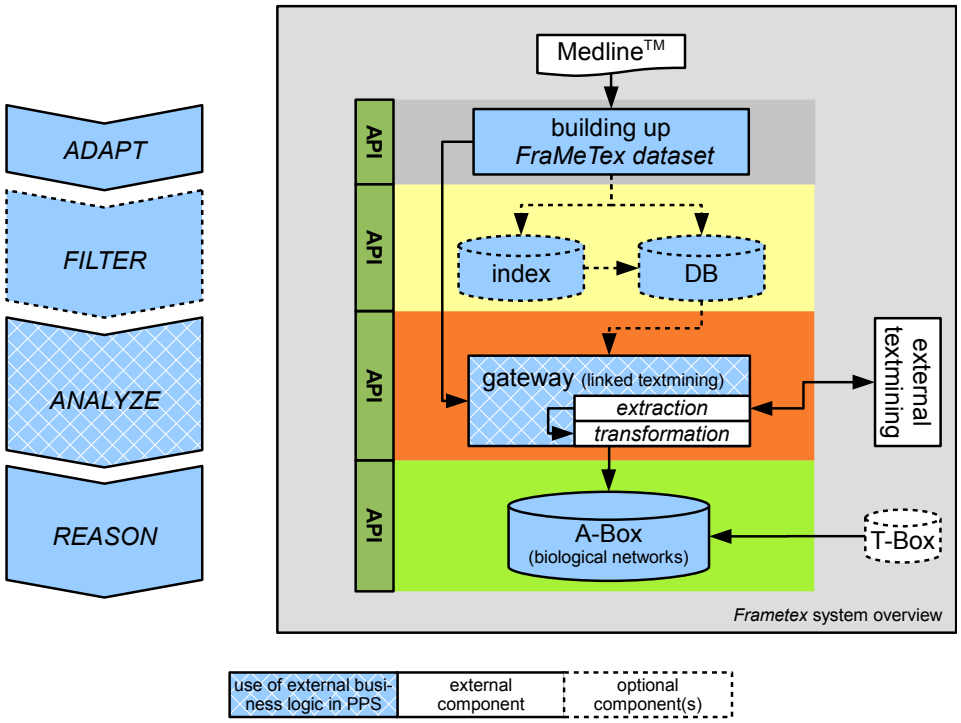


Abbildung 4.4: Systemarchitektur resultiert aus Abbildung der konzipierten PPS auf Module (API)

4.2). Hierzu zählt beispielsweise ein Zeitstempel, der anzeigt wann ein Medline-Eintrag aus den zugrundeliegenden XML-Daten eingelesen wurde. Das *SpecialMetaField* Feld ist lediglich abstrakt definiert und sein *fieldName* nicht frei wählbar. Damit ist stets eine Spezialisierung erforderlich bevor es genutzt werden kann. Der *fieldName* einer derartigen Feld-Instanz spiegelt dann automatisch die spezialisierende Klasse wider. Dies geschieht mit Blick auf eine potentielle Persistierung eines derartigen Feldes. Muss das persistierte Feld rekonstruiert werden, kann der ursprünglich im *FraMeTex*-Dataset genutzte Feldtyp leicht erkannt und zur korrekten Datenrepräsentation wieder herangezogen werden. Auf das verfolgte Konzept wird im Kapitel 5 noch ausführlicher eingegangen.

4.3 Flexibilität durch modulare Systemarchitektur

Das im Abschnitt 4.2 konzipierte *FraMeTex*-Dataset legt den Grundstein für die Zusammenstellung der zunächst unabhängigen PPSs zu einem komplexen *Pathway Prediction*-Prozess (*AFAR*-Konzept, Abbildung 4.1). Die Verknüpfung der PPSs kann allerdings nur gelingen, wenn auch ihre Schnittstellen auf diese Datenstruktur ausgerichtet werden. Hierfür werden die PPSs in einem ersten Schritt technisch als Module aufgefasst. Der Abbildung 4.4 kann sowohl die Zuordnung der PPSs zu Modulen (API), als auch der beabsichtigte Datenfluss zwischen den Modulen entnommen werden. Die aus Medline gelesenen Daten werden zunächst in ein *FraMeTex*-Dataset transformiert bevor sie anschließend in einer indizierten Textdaten-

funktionale Kategorie	Interface
Adaption biomedizinischer Textdaten	ModuleDataReadable
Netzwerkrekonstruktion (native Algorithmen)	IDataProcessable
standardisierte Datenverarbeitung (Systemmodule)	ModuleDataProcessable
Persistenz in Textdatenbank & Wissensbasis	IModulePersistable

Tabelle 4.2: Funktionale Kategorisierung der Modulschnittstellen (Abbildung 4.5)

bank aufbereitet werden. Der optionale Vorverarbeitungsschritt ermöglicht es, die Analyse der Textdaten auf relevante Einträge zu beschränken. Damit können spezifische Netzwerke zielgerichtet rekonstruiert werden. Die anschließende Analyse der Textdaten wird vom Gateway gesteuert. Es bietet den Zugriff auf die gewünschten Textmining-Algorithmen (*extraction*) sowie eine einheitliche Aufbereitung der von ihnen gewonnenen Pathways (*transformation*). Sie fließen in eine Wissensbasis ein (*A-Box*), werden zu Netzwerken verknüpft und mit zusätzlichem Regelwissen (*T-Box*) angereichert. Mit Hilfe einer Inferenzkomponente können so weitere Zusammenhänge abgeleitet werden. Konzeptionell entspricht die motivierte Systemarchitektur einer SOA, die für biomedizinische Experten entscheidende Vorteile mit sich bringen kann (Abschnitt 2.2.5). Sie unterstreicht außerdem den ganzheitlichen Ansatz der Netzwerkrekonstruktion, der auch begleitende Funktionalitäten berücksichtigt (Anforderung 7).

Die Konzeption der erforderlichen Module steht im Abschnitt 4.3.1 im Fokus. Im Anschluss wird auf die Datenverarbeitung innerhalb der Module näher eingegangen. Die Fokussierung auf das *FraMeTex*-Dataset ermöglicht es, allgemeingültige Verarbeitungsroutinen für alle Module zu definieren. Sie schaffen die Grundlage für die spätere Implementierung der unterschiedlich spezialisierten Module. Die entworfenen Algorithmen werden im Abschnitt 4.3.2 präsentiert.

4.3.1 Einheitliche Konzeption funktionaler Module

Das Ziel war es, für alle Module ein gemeinsames Fundament zu schaffen. Im ersten Schritt wurden hierfür die erforderlichen Schnittstellen aus dem Systemdiagramm in Abbildung 4.4 abgeleitet. Entsprechend der genutzten Programmiersprache Java wurden sie als Interfaces definiert und in vier funktionale Kategorien unterteilt (Tabelle 4.2). Den konzeptionellen Zusammenhang der Interfaces zeigt das UML-Diagramm in Abbildung 4.5. Die aufgeführten Interfaces weisen alle erforderlichen Methoden auf, um ihrer jeweiligen Funktion gerecht zu werden. Teilweise fassen sie hierfür auch zwei zunächst unabhängig definierte Interfaces zusammen. In *IProcess* sind zunächst nur sehr allgemeine Methoden definiert, die noch auf keine speziellen Verarbeitungsabläufe abzielen. Sie gewähren in erster Linie Zugriff auf Meta-Informationen eines Prozesses. Entscheidend ist daher die weitere Spezialisierung durch *IModule*. Es legt die Schnittstellen eines Moduls auf ein *FraMeTex*-Dataset fest und schafft damit die Grundvoraussetzung für die beabsichtigte Verknüpfung mehrerer Module. Die explizite Definition des all-

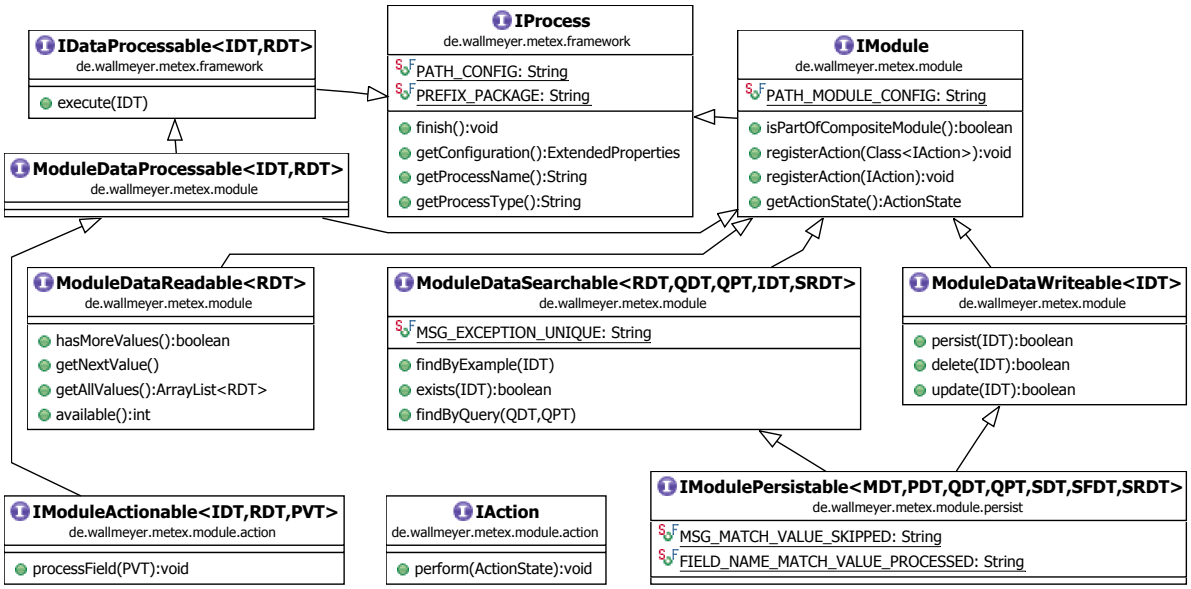


Abbildung 4.5: Interfaces strukturieren modulare Systemarchitektur

gemeineren Interfaces *IProcess* erfolgte bereits mit Blick auf die erforderliche Interaktion mit externen TM-Algorithmen. Wenn es gelingt die Algorithmen zumindest als Spezialisierung von *IDataProcessable* zu repräsentieren, weisen sie mit den Systemmodulen rudimentäre Gemeinsamkeiten auf (*execute()*). Die in Abbildung 4.4 dargestellte Interaktion zwischen Gateway und externen Wissensextraktions-Algorithmen kann dies erheblich erleichtern.

Im Anschluss an die Definition der Interfaces wurde mit der Implementierung der zugehörigen Funktionalitäten begonnen. Im ersten Schritt wurden grundlegende Funktionalitäten geschaffen. Sie sind noch unabhängig von der individuellen Verarbeitung eines Moduls und werden automatisch während dessen Instanziierung ausgeführt:

- **Initialisierung**. Es werden der Name und Typ eines Moduls festgelegt. Der Name spiegelt die implementierende Klasse wider, wohingegen der Typ dem zugehörigen Java-Package (z.B. *textmining*, *adapter*) entspricht. Dies erlaubt es, die verfügbaren Module über ihre funktionale Kategorisierung (Tabelle 4.2) hinaus zu unterscheiden.
- **Konfiguration**. Es werden Konfigurationsdateien eingelesen, deren Einträge die Verarbeitung eines Moduls beeinflussen können. Die Zuordnung einer Konfigurationsdatei zu einem bestimmten Modul erfolgt automatisch anhand des Modulnamens. Der Dateiname muss daher dem Modulnamen entsprechen. Sollten mehrere Module gleichen Typs im Einsatz sein, kann die Zuordnung auch manuell erfolgen.

4.3.2 Vier-Phasen-Datenverarbeitung

Die bisher realisierte Implementierung unterstützt die Konfiguration und Initialisierung der modularen PPS. In diesem Abschnitt wird die Implementierung um abstrakte Verarbeitungs-

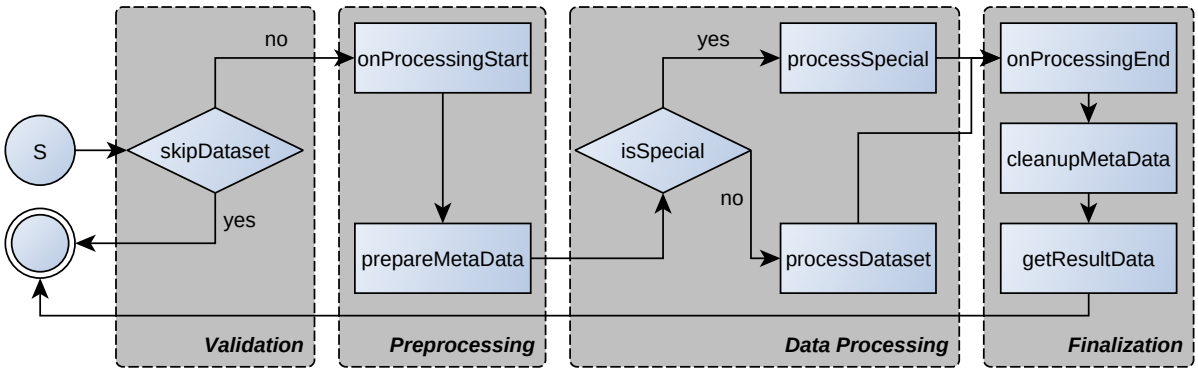


Abbildung 4.6: Vier-Phasen-Datenverarbeitung eines *FraMeTex*-Datasets in UML-Notation

abläufe erweitert. Sie werden in allen Modulen durchlaufen, die das zuvor definierte Interface *ModuleDataProcessable* berücksichtigen (Tabelle 4.2). Die Abläufe werden daher von der Methode *execute()* des Interfaces koordiniert (Abbildung 4.5). Das verfolgte Ziel ist es, die spätere Implementierung individueller *Pathway Prediction*-Funktionalitäten in den Modulen zu erleichtern. Sie sollen sich auf die abstrakten Abläufe stützen und sich dadurch auf die Realisierung konkreter *Pathway Prediction*-Funktionalitäten konzentrieren können. Die Verarbeitung eines *FraMeTex*-Datasets in den Modulen wurde in vier Phasen gegliedert, von denen jede individuell an die Anforderungen eines Moduls angepasst werden kann (Abbildung 4.6):

1. **Validierung**. Zu Beginn werden elementare Eigenschaften des *FraMeTex*-Dataset geprüft. In Abhängigkeit des Ergebnisses wird der zu verarbeitende Datensatz verworfen oder an die nachgelagerte Phase weitergereicht.

2. **Vorverarbeitung**. Den Meta-Daten des *FraMeTex*-Dataset werden temporär spezielle Tochterfelder hinzugefügt. Sie spiegeln den aktuellen Verarbeitungsstatus später wider. Die Informationen werden insbesondere von Modulen benötigt, die die im *FraMeTex*-Dataset repräsentierten Daten persistieren (Textdatenbank, Wissensbasis, Index).

3. **Datenverarbeitung**. In dieser Phase durchläuft das *FraMeTex*-Dataset die spezifische Verarbeitung eines Moduls. Sie kann beispielsweise die Persistierung rekonstruierter Netzwerke in der deduktiven Wissensbasis verantworten (Abbildung 4.4). Für eine größtmögliche Flexibilität werden innerhalb dieser Phase zwei Abläufe differenziert.

    - *Standardverarbeitung*. Sie iteriert die Feldstrukturen des *FraMeTex*-Datasets (Definition 4.1). Den konzipierten Ablauf zeigt Algorithmus 1.

    - *Spezialverarbeitung*. Sie ist an keine Vorgaben gebunden und damit Ausgangspunkt für die Implementierung sehr individueller *Pathway Prediction*-Funktionalitäten.

4. **Finalisierung**. Die Phase wurde analog zur Vorverarbeitung definiert und weist daher korrespondierende Abläufe auf. Die zuvor aufbereiteten Meta-Daten werden wieder in ihren Ursprungszustand zurückversetzt. Abschließend erfolgt durch *getResultDataSet()* die Rückgabe des erzielten Verarbeitungsergebnisses in Form eines *FraMeTex*-Datasets. Es kann damit an nachfolgende PPS weitergereicht werden.

Algorithmus 1 Standardverarbeitung innerhalb Vier-Phasen-Datenverarbeitung eines Moduls

Require: $field \in \delta^F$ ▸ process *meta-*, *key-*, *dataValue*
Require: $skipFields \leftarrow \{\varnothing\}$

```
    procedure ITERATEFIELD(field)
        if field ∈ skipFields ∨ SKIPFIELD(field) then
            if field ∉ skipFields then                    ▸ check maybe complex: cache result
                skipFields ← skipFields ∪ field
5:          return                                        ▸ skip complete field structure
        if ¬SKIPPROCESSING(field) then                    ▸ skip single field instance
            PROCESSFIELD(field)                           ▸ execute moduls' business logic
            PERFORMACTIONS(field)

        for i ← 0, i < field.numberOfChilds() do          ▸ iterate field structure
10:         childFieldName ← field.getChildNames().get(i)
            childFieldList ← field.getChilds(childFieldName)
            for j ← 0, j < childFieldList.size() do
                ITERATEFIELD(childFieldList.get(j))       ▸ start recursion
                j ← j + 1
15:         i ← i + 1
```

Den Ablauf der standardisierten Verarbeitung stellt Algorithmus 1 im Detail dar. Er wird für jedes Feld eines *FraMeTex*-Datasets durchlaufen (Definition 4.1). Bei Bedarf können sowohl komplette Tochterstrukturen (Zeile 5) als auch einzelne Tochterfelder (Zeile 6) von der Verarbeitung ausgenommen werden. Da die hierfür erforderlichen Prüfungen komplex sein können, wird das Ergebnis für jedes geprüfte Feld vorgehalten. Dies geschieht anhand der *fieldID*. Die motivierte *Pathway Prediction* kann so eine mehrfache Prüfung der wiederkehrenden Datenstrukturen in Medline vermeiden. Außerdem kann der Fokus auf die Verarbeitung der Medline-Abstracts gelegt und andere Felder weitestgehend ausgeblendet werden. Die spezifische Datenverarbeitung eines Moduls ist dieser Prüfung deshalb nachgelagert (Zeile 7). Sie kann durch optionale Actions weiter individualisiert werden (Zeile 8). Die Iterationen ab Zeile 9 stellen die rekursive Abarbeitung der hierarchischen Feldstrukturen sicher.

4.4 Persistenz in Textdatenbank und Wissensbasis

Im vorherigen Abschnitt 4.3 wurde eine modulare Systemarchitektur konzipiert. Sie bildet die Grundlage für die Umsetzung des motivierten *Pathway Prediction*-Prozesses. Dies umfasst einheitliche Modul-Schnittstellen sowie abstrakte Verarbeitungsabläufe. Sie werden in diesem Abschnitt weiter ausgebaut. Es ist das erklärte Ziel, die in einem *FraMeTex*-Dataset repräsentierten Informationen in verschiedenen Persistenzlayern speichern zu können. Die erforderlichen

Abläufe sollen vereinheitlicht werden, um redundante Implementierungen für die indizierte Textdatenbank (Index + Datenbank) sowie deduktive Wissensbasis im System zu vermeiden (Abbildung 4.4). Physische Datenzugriffe werden dabei abstrahiert, so dass später nur minimale Anpassungen an ein Persistenzlayer erforderlich sind. Eine generische Lösung erleichtert die spätere Weiterentwicklung und Wartung des geschaffenen Prototyps. Sie sollte daher konzeptionell berücksichtigt werden. Die Umsetzung des Ansatzes basiert auf den beiden Interfaces *IModulePersistable* sowie *ModuleDataProcessable* (Tabelle 4.2). Damit werden sowohl lesende als auch schreibende Datenzugriffe (Datenmanipulationen) im jeweiligen Persistenzlayer unterstützt (Abbildung 4.5). Die geschaffene Implementierung delegiert das Ausführen sämtlicher Datenmanipulationen jedoch an die *execute()* Methode des Interfaces *ModuleDataProcessable*. Damit wird die Ausführung der Datenmanipulationen an dieser Stelle gebündelt und kann auf Basis der zuvor konzipierten Vier-Phasen-Datenverarbeitung umgesetzt werden (Abschnitt 4.3.2). Anhand spezieller Einträge in den Meta-Daten des zu verarbeitenden *FraMeTex*-Datasets kann die jeweils auszuführende Operation (z.B. Update) dennoch erkannt werden. Die Einträge werden in Abhängigkeit der delegierenden Datenmanipulations-Methode gesetzt. Eine Generalisierung lesender Datenzugriffe wird nicht angestrebt, da sie häufig sehr spezifisch sind.

4.4.1 Generalisierter Datenmanipulations-Algorithmus

Die Ausführung schreibender Datenzugriffe erfolgt auf Basis der zuvor konzipierten *Standardverarbeitung* (Algorithmus 1), die ihrerseits der vierphasigen Datenverarbeitung (Abschnitt 4.3.2) untergeordnet ist. Jeder Datenmanipulation ist eine Existenzprüfung im Persistenzlayer vorgelagert, die bereits während der Vorverarbeitungsphase stattfindet. Hierbei wird festgestellt, ob der zu speichernde Datensatz δ^F bereits im Persistenzlayer existiert. In diesem Fall liefert die Prüfung die korrespondierende, native Datenstruktur δ_p^N aus dem Persistenzlayer. Sie kann beispielsweise einer Netzwerkstruktur (biologisches Netzwerk) aus der deduktiven Wissensbasis entsprechen. Die Verarbeitung eines *FraMeTex*-Datasets in einem spezifischen Persistenzlayer gliedert sich damit in zwei Schritte (Abbildung 4.7):

1. Existenzprüfung: Identifikation & Rekonstruktion des persistierten Datensatzes
   - Suchen der gespeicherten, zu δ^F korrespondierenden Datenstruktur δ_p^N
   - Abbildung der nativen Struktur δ_p^N auf ein *FraMeTex*-Dataset δ_l^F (loading)
2. Datenmanipulation: Ausführen der erforderlichen, physischen Datenzugriffe
   - Ermittlung der Differenz Δ zwischen δ^F und δ_l^F
   - physische Manipulation von δ_p^N zur Reduktion von Δ auf $\{\varnothing\}$

Der skizzierte Algorithmus abstrahiert vom Persistenzlayer (*storage*), indem er auf der Ebene der *FraMeTex*-Datasets arbeitet. Die anfängliche Transformation der korrespondierenden, bereits persistierten Datenstrukturen in ein *FraMeTex*-Dataset (δ_l^F) ist hierfür eine zwingende Voraussetzung. Differenzen gegenüber dem zu verarbeitendem Datensatz δ^F können damit

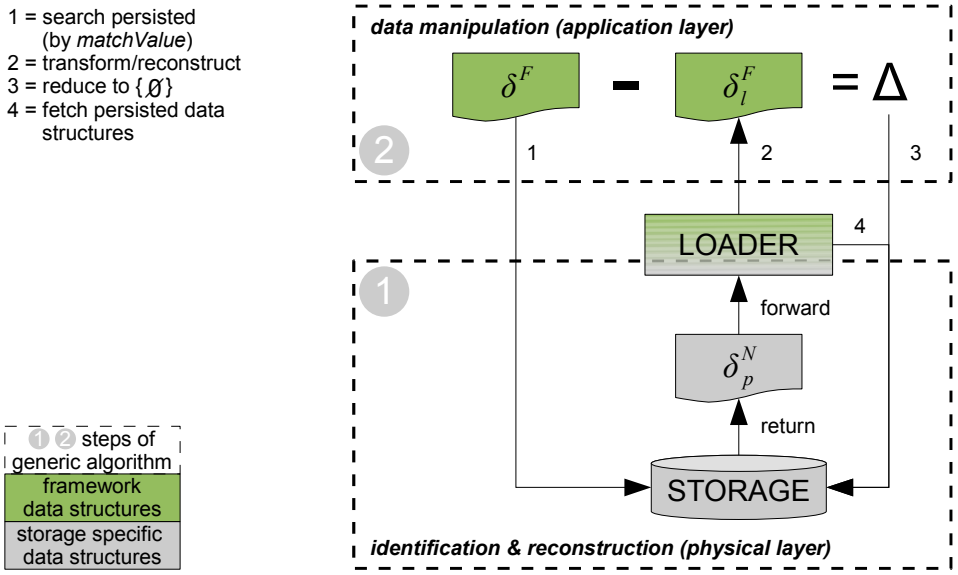


Abbildung 4.7: Generischer Persistenz-Algorithmus abstrahiert von Persistenzlayer

im direkten Vergleich von zwei *FraMeTex*-Datasets erkannt werden. Eine Gegenüberstellung ihrer drei Komponenten (Definition 4.1) genügt hierfür. Insbesondere bei einer *Update*-Operation kann erst durch die direkte Gegenüberstellung der beiden Datensätze erkannt werden, welche Datenmanipulation für einzelne Felder wirklich erforderlich ist. Ein *Update* auf Datensatzebene kann beispielsweise auch zum Löschen eines Felds führen, wenn ein Datensatz seit seiner letzten Persistierung Veränderungen unterlag.

4.4.1.1 Identifikation & Rekonstruktion persistierter Daten

Die Identifikation eines Datensatzes im Persistenzlayer erfolgt anhand seines *matchValue*. Der gesamte Vorgang erfolgt in der konzipierten Vorverarbeitungsphase (Abschnitt 4.3.2). Ist der Datensatz bereits gespeichert worden, wird anhand der persistierten Datenstrukturen (δ_p^N) ein *FraMeTex*-Dataset (δ_l^F) rekonstruiert. Existiert der Datensatz im Persistenzlayer noch nicht, wird dort ein transienter (temporärer) Datensatz initialisiert und für die Rekonstruktion herangezogen. Eine auf das zugrundeliegende Persistenzlayer abgestimmte *Loader*-Komponente übernimmt anschließend die erforderliche Transformation. Sie ordnet außerdem δ_l^F sowie jedem seiner Felder die hierzu korrespondierende, persistierte Datenstruktur des Persistenzlayers zu. Dieses *Mapping* wird in den Meta-Daten von δ_l^F abgelegt und kann über entsprechende Methoden des Loaders (*shortcuts*) bequem referenziert werden. Für die erforderlichen, physischen Datenmanipulationen im Persistenzlayer hat das Mapping eine zentrale Bedeutung. Die konzipierte Rekonstruktion erfolgt feldweise. Zu Beginn wird jedes Feld eines persistierten Datensatzes vom Loader als Schlüssel/Wert-Attribut (*RestoreField*) dargestellt. Den Schlüssel bildet die zugehörige *fieldID*. Die Menge aller RestoreFields wird anschließend anhand ihrer Schlüssel sortiert, so dass sie entsprechend ihrer späteren Hierarchie-Ebene und Position in δ_l^F geordnet sind. Eine zusätzliche Gliederung anhand ihrer Präfixe (Abbildung 4.3) führt zu

Algorithmus 2 Rekonstruktion eines persistierten *FraMeTex*-Dataset

Require: *typedFieldList* ▸ sorted list of one field type
Require: $i \leftarrow 1$

```
    rootField ← createField(typedFieldList.get(0))
    saveRootField ← rootField

    for i ← 1, i < typedFieldList.size() do                ▸ skip already created root field
        restoreField ← typedFieldList.get(i)
5:      segmentList ← restoreField.getFieldIDSegments()    ▸ split fieldID into its segments

        while segment ← segmentList.next() do
            (name, position) ← split(segment)              ▸ split segment into name, position

            if ¬rootField.hasChild(name, position) then
                newField ← createField(typedFieldList.get(i))   ▸ also assigns fields' value
10:             rootField.addChild(newField)

            rootField ← rootField.getChild(name, position)  ▸ descend hierarchy for next segment

        rootField ← saveRootField                          ▸ reset for next restoreField
```

einer geordneten Liste (*typedFieldList*) für jede Komponente eines *FraMeTex*-Datasets (Definition 4.1). Die Liste bildet den Ausgangspunkt der von Algorithmus 2 verantworteten Rekonstruktion. Sie wird durch die Zuweisung der jeweils rekonstruierten Komponente (*meta-, key-, dataValue*) zum Datensatz δ_l^F abgeschlossen. Eine entscheidende Phase während der konzipierten Rekonstruktion ist die Instanziierung der einzelnen Felder (Algorithmus 2, Zeilen 1, 9). Sie muss für persistierte *SpecialMetaFields*, den in ihrer *fieldID* kodierten Feldtyp berücksichtigen (Tabelle 4.1). Darüber hinaus hat sie entscheidenden Einfluss auf die erfolgreiche Rekonstruktion eines nur partiell persistierten *FraMeTex*-Datasets. Wurden Felder oder ganze Feldstrukturen eines *FraMeTex*-Datasets von der Speicherung ausgenommen (Algorithmus 1, Zeilen 5, 6), kann sie hierfür Dummy-Felder einfügen. Alternativ kann eine Reorganisation der nur partiell gespeicherten Datenstruktur erfolgen. Das Ziel beider Ansätze ist es, stets ein *FraMeTex*-Dataset mit lückenloser Feldstruktur zu rekonstruieren. Da beide Verfahren ihre Vor- und Nachteile haben, ist die Entscheidung je nach Anwendungsfall sowie zugrundeliegendem Persistenzlayer zu treffen. Eine Reorganisation ist jedoch in jedem Fall komplexer, da eine Neuberechnung sämtlicher *fieldIDs* erforderlich wird.

4.4.1.2 Ausführen physischer Datenmanipulationen

Der Rekonstruktion von δ_l^F schließt sich das Ausführen der notwendigen Datenmanipulation an. Dem während der Rekonstruktion aufgebauten Mapping in δ_l^F kommt hierbei eine zentra-

Algorithmus 3 Ausführen physischer Datenmanipulationen auf Feldebene

Require: $field \in \delta^F$ ▸ parameter of method (algorithm 1, line 7)
Require: $\delta_p^N \leftarrow$ LOADER.GETPERSISTED(δ_l^F) ▸ complete dataset
Require: $field_p^N \leftarrow$ LOADER.GETPERSISTEDFIELD$(\delta_l^F, field)$ ▸ current field (null when not persisted)
Require: $allFields_p^N \leftarrow$ LOADER.GETPERSISTEDALLFIELDS(δ_l^F) ▸ all fields of dataset

if *PERSIST* **then**
  $field_p^N \leftarrow$ PERSISTFIELD$(field, \delta_p^N)$
  if $field.getFieldName() == matchValue$ **then**
    $isMatchValuePersisted \leftarrow TRUE$ ▸ checked at end of processing

else if *UPDATE* **then**
  if $field_p^N \neq null$ **then**
    $field_p^N \leftarrow$ UPDATEFIELD$(field, field_p^N, \delta_p^N)$
  else ▸ field not part of δ_p^N
    $field_p^N \leftarrow$ PERSISTFIELD$(field, \delta_p^N)$

else if *DELETE* **then**
  if $field_p^N \neq null$ **then**
    DELETEFIELD$(field, field_p^N)$

$allFields_p^N \leftarrow allFields_p^N \setminus field_p^N$ ▸ remaining fields no longer part of δ_p^N

le Rolle zu. Es bietet Zugriff auf die persistierten Strukturen des zu verarbeitenden Datensatzes und ermöglicht die Ausführbarkeit der gewünschten Datenoperation initial zu prüfen. Das Speichern von δ^F wird beispielsweise unterbunden, wenn bereits persistierte Datenstrukturen existieren und in den Meta-Daten von δ_l^F hinterlegt sind. In diesem Fall ist stattdessen ein Update auszuführen. Ist die Ausführbarkeit grundsätzlich gesichert, beginnt die Verarbeitung von δ^F im Persistenzlayer. Sie basiert auf der konzipierten Standardverarbeitung und erfolgt damit feldweise (Algorithmus 1, Zeile 7). Der konzipierte Ablauf ist trivial und wird von Algorithmus 3 dargestellt.

Zu Beginn erlangt der Algorithmus über die im Loader definierten *shortcuts* Zugriff auf die persistierten Strukturen eines Datensatzes. Auf Basis dieser Informationen werden die Operationen für jedes Feld ausgeführt. Die besonderen Herausforderungen eines Updates werden dabei berücksichtigt. Eventuell müssen neue Felder des Datensatzes aufgenommen (Algorithmus 3, Zeile 9) oder nicht mehr genutzte Felder gelöscht werden (Algorithmus 3, Zeile 13). Während der Standardverarbeitung erfolgt allerdings nur die Identifikation der zu löschenden Felder. Das Löschen selbst erfolgt konzentriert in der dafür angepassten Finalisierungs-Phase. Sie ist auch für das Auslösen eines Fehler verantwortlich, wenn die explizit überwachte Persistierung des essentiellen *matchValue* (Algorithmus 3, Zeile 3) nicht erfolgen sollte. Die konkrete Anpassung des generalisierten Persistenz-Konzepts an ein Persistenzlayer (Wissensbasis, Text-

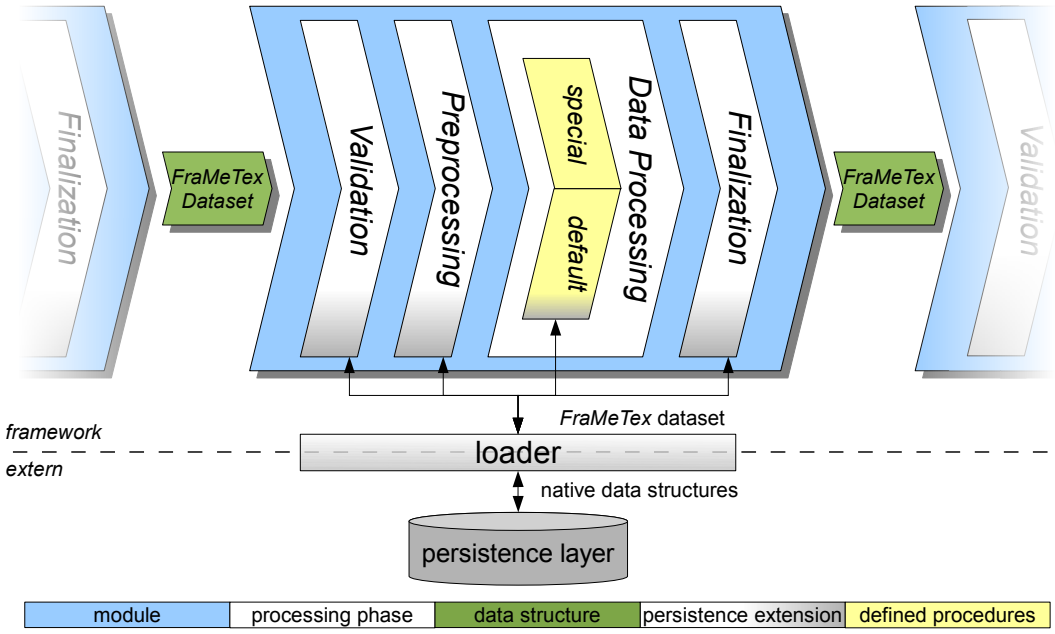


Abbildung 4.8: PPS mit vier-Phasen-Datenverarbeitung und Persistenzunterstützung

datenbank) beschränkt sich damit auf nur drei Methoden (Algorithmus 3, Zeilen 2, 7, 12) sowie den Loader. Die Methoden verantworten das Speichern und Löschen sowie die Aktualisierung der in einem Feld des *FraMeTex*-Dataset repräsentierten Daten. Die Realisierung eines Loaders ist unkompliziert möglich. Eine weitgehend generische Implementierung erfordert auch hier nur geringfügige Anpassungen an das zugrundeliegende Persistenzlayer. Insgesamt wird der Aufbau der konzipierten Textdatenbank sowie der deduktiven Wissensbasis dadurch erheblich vereinfacht (Kapitel 5).

4.5 Zusammenfassung

In diesem Kapitel wurde die zu Beginn der Arbeit motivierte, logikunterstützte *Pathway Prediction* konzipiert. Unter Berücksichtigung verschiedener Parameter wurden im Abschnitt 4.1 zunächst die Systemanforderungen formuliert. Mit dem Aufbau des *FraMeTex*-Datasets (Abschnitt 4.2) begann anschließend die Systemkonzeption. Die spezielle Datenstruktur ist flexibel genug, um sowohl die zu analysierenden Textdaten als auch die rekonstruierten Netzwerkstrukturen sowie sämtliche Zwischenergebnisse repräsentieren zu können. Die Datenstruktur definiert sämtliche Schnittstellen des Systems, das in mehrere PPSs gegliedert wurde. Die PPSs wurden technisch als Module aufgefasst und bilden zusammen den ganzheitlich motivierten Rekonstruktionsansatz ab. Die Module bieten daher auch unterstützende Funktionalitäten, die auch eine Adaption und Filterung der zu analysierenden Textdaten umfassen. Eine vierphasige Verarbeitungsroutine strukturiert und vereinheitlicht den Datenfluss innerhalb der verschiedenen Module.

Im Abschnitt 4.4 wurde die zuvor konzipierte Verarbeitungsroutine für die Speicherung eines *FraMeTex*-Datasets weiter spezialisiert. Das konzipierte System weist mit einer indizierten Textdatenbank sowie einer deduktiven Wissensbasis bereits zwei unterschiedliche Persistenzlayer auf. Es wurde daher ein Algorithmus entworfen, der weitestgehend unabhängig vom Persistenzlayer arbeitet. Die Unabhängigkeit wird durch eine Loader-Komponente erreicht. Sie ist für eine Transformation der jeweils persistierten Datenstrukturen in ein *FraMeTex*-Dataset verantwortlich. Dadurch kann von den physischen Datenstrukturen eines Persistenzlayers abstrahiert werden und ein Großteil der Verarbeitung auf Ebene des *FraMeTex*-Datasets erfolgen. Die Abläufe zum Speichern eines *FraMeTex*-Datasets sind damit in allen Modulen identisch. Lediglich die physischen Datenzugriffe müssen entsprechend der in der Datenstruktur repräsentierten Daten (Texte, Netzwerke) und des verwendeten Persistenzlayers (Datenbank, deduktive Wissensbasis) angepasst werden. Die Implementierung der betreffenden Module wird dadurch erheblich vereinfacht.

Die in diesem Kapitel konzipierten Abläufe und Datenstrukturen stellt Abbildung 4.8 schematisch dar. Sie zeigt sowohl die Gliederung der motivierten *Pathway Prediction* in mehrere Module, als auch die vierphasige Datenverarbeitung innerhalb eines Moduls. Ihre konzeptionelle Erweiterung für Persistenzoperationen mit der zentralen Loader-Komponente ist farblich hervorgehoben. Der Datenaustausch zwischen den konzipierten Modulen erfolgt ausschließlich mit dem *FraMeTex*-Dataset. Formal basiert die konzipierte Netzwerkrekonstruktion auf einer SOA, da sämtliche Module unabhängig genutzt und kombiniert werden können.

5 Implementierung

In diesem Kapitel wird auf die Realisierung der konzipierten PPS eingegangen (Abbildung 4.1). Ihre Implementierung baut auf den im Kapitel 4 präsentierten Datenstrukturen und Abläufen auf (Abbildung 4.8). Im Abschnitt 5.1 wird zunächst auf die Erschließung und Identifikation relevanter Textdaten eingegangen. Damit wird eine Grundvoraussetzung für die Vorhersage spezifischer Netzwerke aus Textdaten geschaffen und die Forderung nach einem ganzheitlichen System umgesetzt. Im Anschluss befasst sich der Abschnitt 5.2 mit der deduktiv unterstützten Rekonstruktion biologischer Netzwerke aus zuvor selektierten Textdaten. Dieser Abschnitt geht damit auf die Implementierung der Kernfunktionalitäten der motivierten *Pathway Prediction* ein (Abbildung 1.1). Anhand des speziellen Proteins MPDZ/MUPP1 (Anforderung 1) stellt Abbildung 5.1 den implementierten Datenfluss abstrakt dar.

5.1 Identifikation relevanter Textdaten zur Vorhersage biologischer Netzwerke

Zur Vorhersage spezifischer Netzwerke müssen aussagekräftige Einträge in Medline anhand benutzerdefinierter Kriterien identifiziert werden können. Die prototypische Implementierung fokussiert auf Medline, andere Datenquellen sollen zukünftig jedoch ebenfalls einfließen können (Abschnitt 1.2). Der Einsatz von PubMed wurde daher verworfen und einer individuellen Lösung der Vorzug gegeben. Ihre Intention ist es, eine effiziente Filterung beliebiger, textbasierter Datenquellen zu bieten. Eine leistungsstarke Lösung ist entscheidend, da die Rekonstruktion komplexer Netzwerke die Verarbeitung großer Datenmengen erfordern kann. Die geschaffene Realisierung umfasst die beiden PPSs *ADAPT* und *FILTER* (Abbildung 4.1). Die Filterung beruht auf einer unabhängigen Index- und Datenbankkomponente (Abbildung 4.4). In einem separaten Vorverarbeitungsschritt werden die in Dateien distribuierten Medline-Daten indiziert und gespeichert. Anschließend kann auf relevante Datensätze direkt zugegriffen werden.

5.1.1 Effizienter Zugriff auf lokale Medline-Daten

Ein spezialisiertes Datenadapter-Modul (*ADAPT*, Abbildung 4.4) übernimmt den Zugriff auf die lokalen Medline-Dateien. Die Realisierung eines universellen Adapters für beliebige Text-Datenquellen der Bioinformatik ist leider nicht umsetzbar. Ihre strukturelle sowie semantische

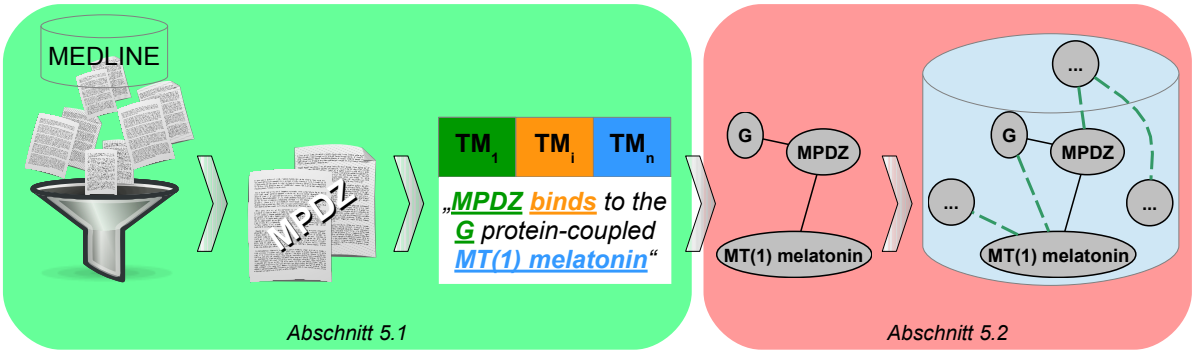


Abbildung 5.1: Schematische Darstellung des Datenfluss „vom Text zum inferierten Netzwerk“

Heterogenität ist hierfür zu umfangreich [Kor10]. Es wurde dennoch versucht, eine möglichst generische Lösung zu finden. Mit dem Interface *ModuleDataReadable* wurde bereits während der Systemkonzeption eine formale Basis zum Auslesen von Datenquellen geschaffen (Tabelle 4.2). Die dort definierten Methoden sehen ein iteratives Lesen der verfügbaren Daten vor (Abbildung 4.5). Eine abstrakte Implementierung setzt den grundlegenden Vorgang um und kann an individuelle Datenquellen angepasst werden. Die erforderlichen Anpassungen beschränken sich auf die Reader-Komponente des Datenadapters (Abbildung 5.2), die drei Ziele verfolgt:

1. Aufbauen einer physischen Verbindung zur Datenquelle und diese auslesen
2. Transformation der nativen Datenstrukturen in ein *FraMeTex*-Dataset
3. Bereitstellen der aufbereiteten *FraMeTex*-Datasets in einem Puffer

Eine erfolgreiche Transformation setzt grundlegende Kenntnisse der adaptierten Datenquelle voraus. Eine äquivalente Abbildung der nativen Datenstrukturen auf die *key*- und *dataValue*-Komponente eines *FraMeTex*-Dataset (Definition 4.1) ist nur möglich, wenn dem Reader die Schlüsselattribute der Datenquelle bekannt sind. Entsprechend der formulierten Anforderungen (Abschnitt 4.1), können die erforderlichen Angaben in einer Konfigurationsdatei gemacht werden (Anforderung 9). Sie wird während der Initialisierung-Phase des Datenadapter-Moduls automatisch eingelesen (Abschnitt 4.3.1). Damit kann der Reader flexibel an die Strukturen unterschiedlicher Datenquellen angepasst werden. Der Reader arbeitet in einem eigenem Thread, der direkt nach der Instanziierung des Datenadapter-Moduls mit dem Lesevorgang beginnt. Die Synchronisation der beteiligten Komponenten übernimmt ein Puffer. Er weist eine feste Größe auf und steuert über seinen Füllgrad die Synchronisation. Ist er ausgereizt, pausiert der Reader. Für die Auslieferung der Datensätze an nachgelagerte Module wurde mit dem *RawDataset* ein spezialisiertes *FraMeTex*-Dataset geschaffen. Sein *matchValue* wird in Abhängigkeit der repräsentierten Daten generiert und seinen Meta-Daten außerdem das Feld *DataOriginInfo* hinzugefügt. Der Wert des Feldes setzt sich aus der Adapter-Konfiguration zusammen und identifiziert die Datenquelle des Datensatzes eindeutig. Damit besteht in den rekonstruierten Netzwerken später die Möglichkeit, die einem Pathway zugrundeliegende Datenquelle zu referenzieren. Zur Auswertung rekonstruierter Netzwerke kann der Anwender so Einblick in die analysierten Daten nehmen.

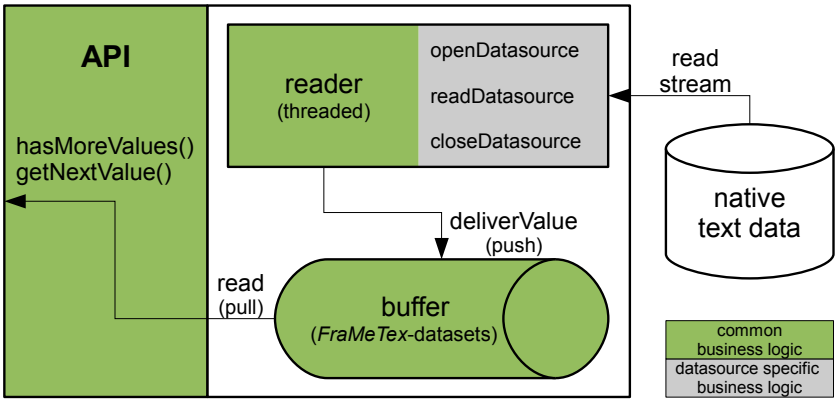


Abbildung 5.2: Schematischer Aufbau des abstrakten Datenadapter-Moduls

Kombinierter Einsatz eines XML- und Dateisystem-Adapters

Ausgehend von der vorherigen Konzeption erfolgt nun die Realisierung eines Datenadapters, der auf das Einlesen lokaler Medline-Daten spezialisiert ist. Da Medline in komprimierten XML-Dateien distribuiert wird, muss er zwei Anforderungen erfüllen:

1. Rekursives Traversieren von Dateisystemen und Archiv-Dateien
2. Effizientes Lesen in XML repräsentierter Daten

Zugunsten einer größtmöglichen Flexibilität wurden die beiden Anforderungen in zwei unabhängigen Adaptoren realisiert. Zur Verarbeitung der Medline-Daten wurden sie geschickt kombiniert. Damit können XML-Daten nicht nur aus Dateien, sondern auch aus anderen Datenströmen gelesen werden. Dies kann später die Rekonstruktion biologischer Netzwerke vereinfachen, da auch einige der zuvor präsentierten TM-Algorithmen (Abschnitt 3.1) ihre Analyseergebnisse in XML repräsentieren (Tabelle 3.2).

Die Reader-Komponente des XML-Adapters basiert auf der Apache Xerces[1] API. Sie ermöglicht eine ereignisorientierte Verarbeitung des XML-Eingabedatenstroms. Damit entfällt eine Repräsentation des XML-Dokuments im Speicher und es können beliebig große Eingaben verarbeitet werden. Event-Handler reagieren auf definierte Ereignisse beim Parsen der XML-Struktur mit dem Aufruf spezieller *Callback*-Methoden [GHJV04]. Ein derartiges Ereignis ist beispielsweise der Beginn eines XML-Elements. Leitet das Element einen neuen Datensatz im Datenstrom ein, wird ein *RawDataset* instanziiert. Es repräsentiert nach Abschluss des Parsens die zum Datensatz gehörenden Daten. Da der Adapter das entscheidende XML-Tag jedoch nicht selber erkennen kann, muss es dem Adapter über dessen Konfiguration bekannt gemacht werden. In Medline leitet der Tag <*MedlineCitation*> einen neuen Datensatz ein. Zur korrekten Abbildung eines gelesenen Datensatzes auf ein *RawDataset* nutzt der Reader des XML-Adapters einen speziellen Parse-Stack (Abbildung 5.3). Während des Parsens der XML-Daten

[1] http://xerces.apache.org/

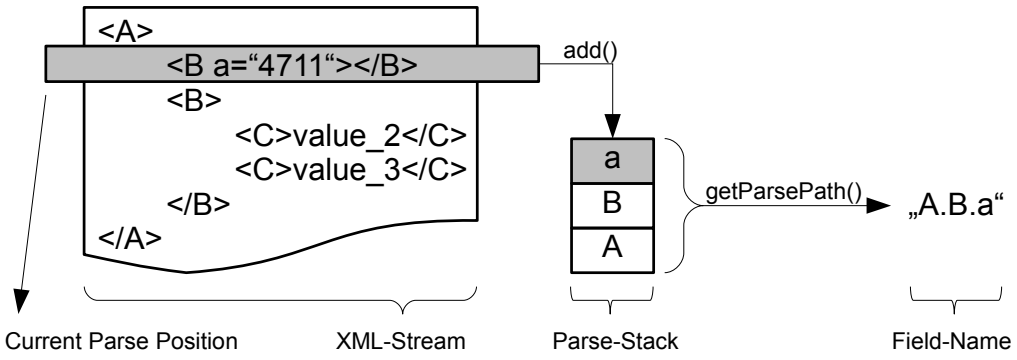


Abbildung 5.3: Reader des XML-Adapters nutzt Parse-Stack zum Aufbau eines *FraMeTex*-Datasets

wird der Name jedes beginnenden XML-Elements dem Stack hinzugefügt und beim zugehörigenden, schließenden Tag wieder entfernt. Der Stack stellt damit die korrekte Abbildung der hierarchischen XML-Struktur im *RawDataset* sicher. Ist dem Stack ein neues Element hinzugefügt worden, wurde eine tiefere Hierarchie-Ebene betreten. Die aktuell geparsten Daten müssen dann als Tochterfeld des zuvor verarbeiteten Feldes im *RawDataset* repräsentiert werden. Dies gilt auch für Attribute eines XML-Elements. Sie werden im Frametex-Dataset ebenfalls durch Tochterfelder repräsentiert. Existieren bereits Felder auf einer Ebene, werden neue Felder stets hinter dem letzten der Ebene angefügt. Dadurch bleibt auch die Ordnung aus der XML-Struktur im Frametex-Dataset erhalten. Zur vollständig, korrekten Abbildung der XML-Struktur muss beim Betreten einer höheren Hierarchie-Ebene nur noch das jeweils korrespondierende Eltern-Feld im Frametex-Dataset ermittelt werden. Der beschriebene Vorgang kann anhand des in Abbildung 4.2 dargestellten *FraMeTex*-Dataset nachvollzogen werden.

Mit dem XML-Adapter ist das Lesen der von Medline definierten Datenstruktur möglich. Für den Zugriff auf die komprimierten Dateien im Dateisystem muss jedoch noch ein weiterer Adapter geschaffen werden. Die angepasste Reader-Komponente dieses Dateisystem-Adapters soll eine rekursive Iteration durch die Baumstruktur eines Dateisystem ermöglichen. Ausgehend von einem in der Modul-Konfiguration definierten Startpunkt sollen sämtliche Dateien in allen untergeordneten Ordner durchlaufen werden. Die Informationen jeder iterierten Datei sollen dabei in Abhängigkeit ihres jeweiligen Dateityps eingelesen werden. Der sich ergebene Ablauf ist in Abbildung 5.4 schematisch dargestellt. Deutlich zu erkennen ist die konzeptionelle Aufteilung des verantwortlichen Reader in die beiden Komponenten *controlling* sowie *harvesting*. Während der Iteration des Dateisystems (controlling) erfolgt das Auslesen der Dateien mit einem jeweils auf ihren Inhalt abgestimmten Harvester (harvesting). Die Trennung in zwei Komponenten ermöglicht es, mit einer einzigen Dateisystem-Adapter-Implementierung zu arbeiten. Zukünftige Anpassungen und Erweiterungen, für andere Dateitypen als die von Medline genutzten, beschränken sich auf die Harvester. Zugleich profitieren alle Harvester automatisch von den Funktionalitäten der universellen Kontroll-Komponente, die beispielsweise bereits Archiv-Dateien entpacken und deren Inhalte in der Iteration berücksichtigen kann.

Zur Verarbeitung der XML-basierten Medline-Dateien ist damit im Dateisystem-Adapter nur eine abgestimmte Harvester-Komponente zu realisieren. Sie wird von dem zuvor geschaffenen

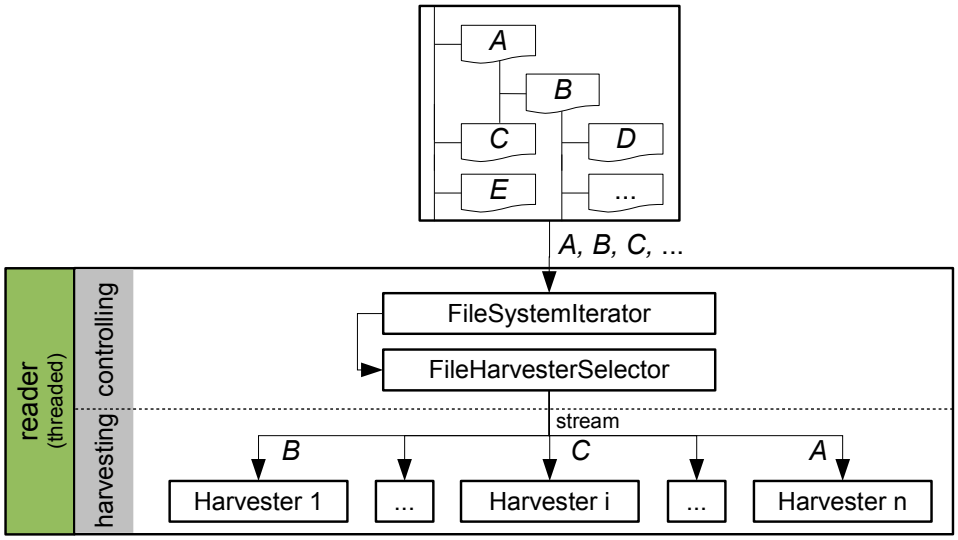


Abbildung 5.4: Schematische Darstellung des Dateisystem-Adapters

XML-Adapter gebildet, dessen Konfiguration lediglich an die Medline-Strukturen angepasst werden muss. Durch die Verschachtelung des Dateisystem- und XML-Adapters stellt sich das Erschließen der lokalen Medline-Daten somit als 2-stufige Adapter-Kaskade dar.

5.1.2 Daten-Aufbereitung in indizierter Datenbank

Mit dem geschaffenen Medline-Adapter können die in XML kodierten Daten sequentiell aus den komprimierten Dateien gelesen werden. Eine benutzerdefinierte Selektion einzelner Medline-Einträge für die Rekonstruktion eines spezifischen Netzwerks ist allerdings noch nicht möglich. Darüber hinaus könnte auf die identifizierten Daten in den Dateien auch nicht direkt zugegriffen werden. Es wurde daher zusätzlich eine indizierte Textdatenbank konzipiert, die im Rahmen einer Vorverarbeitung aufgebaut werden kann. Sie entsteht durch die Kombination zweier unabhängiger Module, die entsprechend der entworfenen Systemarchitektur für die Daten-Filterung verantwortlich sind (*FILTER*, Abbildung 4.4). Ein Indizierungsmodul ermöglicht eine effiziente Suche in dem zuvor aufbereiteten Medline-Datenbestand, ein spezielles Storage-Modul den direkten Zugriff auf einzelne Datensätze. Sowohl die Indexstruktur als auch die Datenbank werden als individuelles Persistenzlayer aufgefasst und auf Basis des Persistenzkonzepts (Abschnitt 4.4) umgesetzt. Ihre Implementierung konzentriert sich damit auf die Bereitstellung angepasster Loader sowie auf die Umsetzung physischer Datenzugriffe.

5.1.2.1 Aufbau der Indexstruktur

Prädestiniert für eine leistungsstarke Datenindizierung ist die Lucene API (Abschnitt 2.2.4). Sie kann Indexstrukturen aufbauen, verwalten und effizient durchsuchen. Eine Umsetzung der lesenden sowie schreibenden Datenzugriffe auf Basis des unabhängigen Persistenzalgorithmus

lag nahe. Der konzipierte Ablauf musste jedoch leicht modifiziert werden, da die Lucene API keine Update-Operation unterstützt[2]. Die von Lucene zu indizierenden Felder eines *FraMeTex*-Datasets können in der Modulkonfiguration (Abschnitt 4.3.1) definiert werden. Zusätzlich wird automatisch ein kumuliertes Such-Feld aufgebaut. Es fasst die Werte aller indizierten Felder noch einmal zusammen und wird selbst indiziert. Standardisierte Suchanfragen beziehen sich dann auf dieses Feld und erstrecken sich somit implizit über alle indizierten Felder eines Datensatzes. Die indizierten Felder müssen damit nicht einzeln durchsucht werden. Realisiert ist dieses Vorgehen auf Basis des eigens hierfür eingeführten Meta-Feldes *IndexSearch*. Es wird in der Vorverarbeitungsphase (Abbildung 4.6) den Meta-Daten des zu indizierenden *FraMeTex*-Dataset hinzugefügt. Während der iterativen Standardverarbeitung wird dann der Wert jedes indizierten Feldes darin abgelegt. In der Finalisierungsphase wird das Feld schließlich indiziert und wieder aus den Meta-Daten entfernt. Um den Anwender die Konfiguration des Indizierungsverhalten zu erleichtern wurde eine Konfigurator-Instanz geschaffen. Sie sorgt für den Aufbau einer Indexstruktur, die gängige Anforderungen abdeckt:

- Die Daten des *IndexSearch*-, *matchValue*- sowie *DataOriginInfo*-Felds werden indiziert und zusätzlich im Index gespeichert. Sie tragen die entscheidenden Informationen für die Rekonstruktion eines indizierten Datensatzes durch den Loader.
- Für die Indizierung des *IndexSearch*-Feld wird zusätzlich ein Analyzer genutzt. Er sorgt für eine Segmentierung und Normalisierung der Daten, um eine bestmögliche Datenfilterung gewährleisten zu können.
- Für alle übrigen Felder kann auf eine Standardkonfiguration zurückgegriffen werden. Dadurch werden sie unter Verwendung eines Analyzers indiziert und können somit auch einzeln und unabhängig vom kumulierten Suchfeld durchsucht werden.

In Abhängigkeit der gewählten Konfiguration ermöglicht die resultierende Indexstruktur granulare Suchoperationen. Darüber hinaus wird bei Bedarf eine weitere Individualisierung unterstützt.

5.1.2.2 Relationale Textdatenbank

Mit Schaffung des Index-Moduls können die aus Medline gelesenen Datensätze anhand benutzerdefinierter Kriterien effizient gefiltert werden. Ein direkter Zugriff auf die Daten ist jedoch noch nicht gegeben. Er soll durch eine initiale Speicherung der Daten in einer relationalen Datenbank gewährleistet werden (Abschnitt 2.2.3.1). Hierfür wurde ein spezielles Storage-Modul geschaffen, das ebenfalls auf dem generalisierten Persistenz-Konzept aufbaut (Abschnitt 4.4). Das konzipierte *Entity Relationship Model (ERM)* der Datenbank zeigt Abbildung 5.5. Das Datenmodell der Textdatenbank ist bewusst einfach gehalten worden und besteht lediglich aus vier Entitäten. Sie dienen der strukturellen sowie inhaltlichen Persistierung eines *FraMeTex*-Datasets. Für jeden Datensatz wird ein Eintrag in der Tabelle *dataset* vorgenommen, dem seine

[2] Bezieht sich auf die Version 2.2.0 der Lucene API.

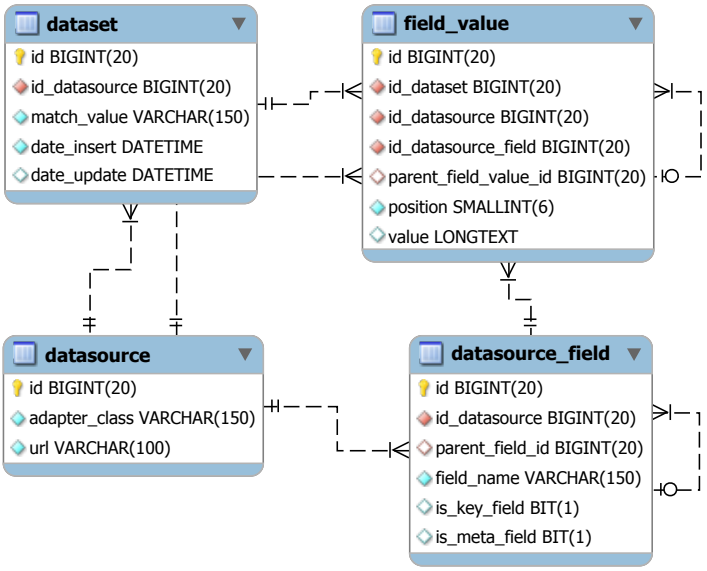


Abbildung 5.5: ERM der im Storage-Modul eingesetzten Datenbank

Datenquelle (*datasource*) zugeordnet wird. Zusätzlich wird für jeden Datensatz der *matchValue*, sowie der Zeitstempel des Einfügens und der letzten Aktualisierung gespeichert. Bei Bedarf kann so später mit einer einzelnen Abfrage geprüft werden, ob ein Datensatz bereits in der Datenbank enthalten ist. Für alle anderen Felder des zu speichernden Datensatzes erfolgt hingegen ein einmaliger Eintrag in der Tabelle *datasource_field*. Sie spiegelt die Feldstruktur eines Datensatzes wider. Die von einem Feld repräsentierten Daten werden in der Tabelle *field_value* abgelegt. Ein Eintrag in der Tabelle *field_value* repräsentiert damit ein bestimmtes Feld eines eindeutig identifizierten Datensatzes aus einer bestimmten Datenquelle. Die konzipierte Aufteilung der Speicherung eines Datensatzes in Datensatzstruktur und Inhalte hat entscheidende Vorteile:

- Es können identische Datensätze aus unterschiedlichen Datenquellen verwaltet werden.
- Die Feldinformationen einer Datenquelle müssen nur einmalig gespeichert werden.
- Nachträgliche Daten-Manipulationen wirken sich nur auf die Tabelle *field_value* aus.

Für den Zugriff auf die Datenbank wird ein *Objekt-Relationales-Mapping (ORM)*-Layer genutzt. Es suggeriert dem Storage-Modul eine objektrelationale Datenbank. Die Kommunikation zwischen dem objektorientiert in Java entwickelten Modul (Anforderung 6) und der Datenbank wird dadurch erheblich erleichtert. Die prototypische Implementierung nutzt das ORM-Layer Cayenne[3] (Apache). Es übernimmt auch die initiale Anlage der Tabellen in der Datenbank. Ein abstrahierendes ORM-Layers führt allerdings auch zu einer unumgänglichen Komplexitätssteigerung, die eine Verarbeitung von Massendaten verzögert. Sämtliche Datenzugriffe wurden daher zusätzlich in einer Facade gekapselt, die einen unkomplizierten Wechsel des Datenzugriffsverfahrens erlaubt. Anstelle des ORM-Layers kommt für die initiale Persistierung der

[3] http://cayenne.apache.org

21 Millionen Medline Abstracts eine native Datenbank-Kommunikation mittels *Java Database Connectivity (JDBC)* zum Einsatz. Der einmalige Aufbau der Datenbank konnte dadurch signifikant beschleunigt werden.

5.1.2.3 Verknüpfung von Index und Textdatenbank

Mit dem Index- sowie Storage-Modul können die aus Medline gelesenen Daten indiziert und gespeichert werden. Für eine gezielte Selektion sowie Analyse einzelner Datensätze zur Rekonstruktion biologischer Netzwerke müssen sie verknüpft werden. Zunächst werden die indizierten Datensätze mit dem Index-Modul anhand benutzerdefinierter Kriterien gefiltert. Die im Rahmen der Suche identifizierten Datensätze werden in Form rekonstruierter *FraMeTex*-Datasets zurückgeliefert. Sie weisen jedoch nur die Schlüsselattribute sowie entscheidende Meta-Daten des Datensatzes auf. Ihre direkte Speicherung im Index wird von der geschaffenen Konfigurator-Instanz (Abschnitt 5.1.2.1) während der Indizierung sichergestellt. Die aus dem Index rekonstruierten Datensätze dienen anschließend als Eingabe für das Storage-Modul. Anhand der gegebenen Schlüsselinformationen werden die vollständigen Datensätze geladen und die *FraMeTex*-Datasets vervollständigt. Sie repräsentieren damit aus Medline selektierte Daten, die zur Rekonstruktion eines spezifischen, biologischen Netzwerks herangezogen werden sollen. Die erforderliche Verknüpfung der beiden Module ist aufgrund der konzipierten Systemarchitektur trivial. Da beide Module auf dem Persistenzkonzept basieren, weisen sie identische Methoden auf (Tabelle 4.2, Abbildung 4.5). Sie müssen zum Ausführen einer Datenmanipulation lediglich synchron aufgerufen werden. Die Verantwortung hierfür bleibt jedoch dem Anwender überlassen. Die prototypische Implementierung stellt nicht automatisch sicher, dass Änderungen an der Indexstruktur auch in der Datenbank reflektiert werden.

5.1.3 Zusammenfassung

Im Abschnitt 5.1 wurde eine elementare Voraussetzung für die motivierte *Pathway Prediction* geschaffen. Aus dem textbasierten Medline können aussagekräftige Datensätze anhand benutzerdefinierter Kriterien effizient selektiert werden. Damit ist es möglich, die Netzwerkrekonstruktion auf bestimmte Proteine oder Gene einzuschränken. Die Adaption von Medline sowie die Filterung der darin enthaltenen Daten ist konzeptionell den beiden PPSs *ADAPT* und *FILTER* zugeordnet (Abbildung 4.4). Es wurden drei unabhängige Module geschaffen, die für ihre technische Umsetzung verantwortlich sind. Sie können flexibel genutzt werden und sind nicht auf Medline beschränkt. Damit können analog auch andere Textdatenquellen verarbeitet werden, deren Daten zu einer Präzisierung rekonstruierter Netzwerke führen können. Für Medline wurde zunächst ein Datenadapter geschaffen, der die nativen XML-Strukturen in ein äquivalentes *FraMeTex*-Dataset überführt. Er schafft damit die Voraussetzung zur weiteren Verarbeitung im System, die eine Indizierung und Speicherung der Daten vorsieht. Anhand benutzerdefinierter Schlagwörter kann der Index durchsucht werden, bevor im Anschluss die identifizierten Datensätze aus der Datenbank geladen werden. Damit ist ein direkter Datenzugriff auf Medline und andere Textdatenquellen gegeben, die ansonsten nur sequentiell gelesen werden könnten.

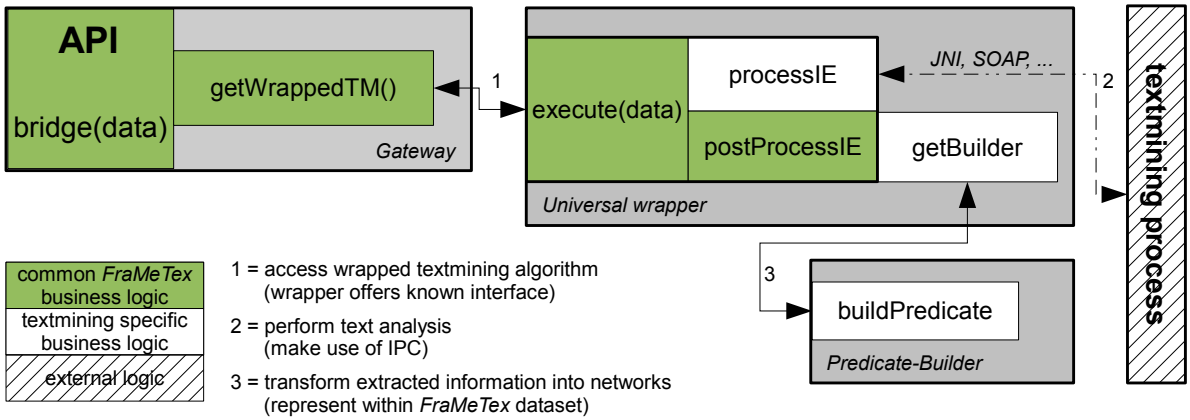


Abbildung 5.6: Integration einer TM-Ressource in Analyseprozess mittels Gateway

Die selektierten Datensätze entsprechen wiederum einem *FraMeTex*-Dataset und fließen in die Netzwerkrekonstruktion ein. Auf ihre Realisierung wird nun eingegangen.

5.2 Rekonstruktion beliebiger, biologischer Netzwerke mit Deduktionsunterstützung

Das Speichern aus Textdaten rekonstruierter Netzwerke in einer Deduktionskomponente (Wissensbasis) ist das primäre Ziel dieser Arbeit (Abschnitt 1.2). Die Umsetzung soll allerdings auch anderweitig vorhergesagten Netzwerken (Abschnitt 3.2) offen stehen und umfasst die beiden Verarbeitungsschritte (PPSs) *ANALYZE* und *REASON* (Abbildung 4.4). Im Abschnitt 5.2.1 wird zunächst gezeigt, wie existierende TM-Algorithmen zur Netzwerkrekonstruktion kombiniert werden können. Die von ihnen gewonnenen Informationen werden zu Pathways aufbereitet und in Netzwerken zusammengefasst. Ihre Repräsentation in einem angepassten *FraMeTex*-Dataset steht im Abschnitt 5.2.2 im Fokus. Das Speichern dieser Netzwerke in der Deduktionskomponente wird im Abschnitt 5.2.3 diskutiert. In Verbindung mit Regeln können Schlussfolgerungen in der Datenbank zu weiteren Vorhersagen führen. Die Möglichkeit, die Datenbank mit Regelwissen zu verknüpfen, wird daher im Abschnitt 5.2.4 erörtert. Abschließend wird im Abschnitt 5.2.5 kurz auf die Exploration der gespeicherten Netzwerke eingegangen. Anhand einer exemplarischen MPDZ/MUPP1-Proteininteraktion werden die implementierten Verfahren verdeutlicht. Sie basieren weitestgehend auf existierenden Komponenten und Ressourcen (Abschnitt 4.1.1).

5.2.1 Pathway-Extraktion aus Textdaten

Die Rekonstruktion biologischer Netzwerke aus Textdaten ist in der Systemarchitektur dem PPS *ANALYZE* zugeordnet (Abbildung 4.4). Aus den zuvor selektierten Textdaten (Abschnitt

5.1) werden Pathways extrahiert, die zusammen ein Netzwerk bilden (Abschnitt 2.1.4). Die Analyse der Textdaten kann sich auf eine große Anzahl spezialisierter TM-Algorithmen stützen (Tabelle 3.2). Durch eine Kombination mehrerer Algorithmen sollen nicht nur besonders präzise, sondern auch unterschiedliche, biologische Netzwerke vorhersagt werden können (Anforderungen 2 & 3). Ein unproblematischer Austausch der zu nutzenden Algorithmen ist daher elementare Bedingung.

Aus konzeptioneller Sicht werden die heterogenen Algorithmen in die definierten Schnittstellen des Systems eingebettet (*wrapping*). Diese zentrale Aufgabe übernimmt das Interface *IDataProcessable* (Tabelle 4.2) zusammen mit einem Gateway-Modul. Das Interface stellt zunächst sicher, dass jeder Algorithmus über die einheitliche Schnittstelle (*execute()*) angesprochen werden kann. Diesen Umstand nutzt wiederum das Gateway-Modul aus, dessen Aufgabe die Integration der zuvor homogenisierten TM-Ressource in den Datenfluss des Systems ist. Die zu analysierenden Textdaten eines *FraMeTex*-Datasets werden vom Gateway an die TM-Ressource übergeben und die Analyseergebnisse wieder empfangen. Das Gateway agiert damit als Brücke zwischen *FraMeTex* und der externen TM-Ressource (Abbildung 5.6). Auf den ersten Blick scheint das konzipierte Vorgehen die nutzbaren Algorithmen stark einzugrenzen. Technisch kann das fürs Wrapping verantwortliche Interface nur von TM-Ressourcen berücksichtigt werden, die selbst in Java geschrieben sind. Außerdem ist Zugriff auf den Programm-Code erforderlich. Obwohl beides nicht garantiert werden kann, ist das Konzept dennoch tragfähig: Es wurde ein universeller sowie flexibler Wrapper geschaffen, mit dem die Problematik umgangen wird. Anstelle der TM-Ressource berücksichtigt er das Interface. Der Wrapper übernimmt dann die Interaktion mit der gewünschten TM-Ressource und gliedert sie in zwei Phasen:

1. *Wissensextraktion*. Das Gateway übergibt die zu analysierenden Textdaten an den universellen Wrapper, der den zu nutzenden TM-Algorithmus einbettet. Die Kommunikation zwischen Algorithmus und Wrapper erfolgt mittels *Inter Process Communication (IPC)*.

2. *Wissensaufbereitung*. Sie überführt die zuvor extrahierten Informationen in Pathways und repräsentiert sie in einem *FraMeTex*-Dataset. Verantwortlich für die Transformation ist eine eigenständige *Builder*-Instanz (Abbildung 5.6).

Allein durch die Realisierung einer IPC zwischen der genutzten TM-Ressource sowie dem universellem Wrapper und der Bereitstellung einer abgestimmten Builder-Instanz kann damit eine beliebige TM-Ressource in *FraMeTex* genutzt werden. Auswirkungen auf die genutzten TM-Ressource selbst sind dadurch nicht zu befürchten. Der Aufbau einer IPC wird von Java zudem umfassend unterstützt. Für (nahezu) jedes denkbare Szenario stehen adäquate Lösungen parat. So können über das *Java Native Interface (JNI)* [Ofto5] beispielsweise native Programmbibliotheken angesprochen oder über Webservices auch eine unabhängige Koppelung heterogener Prozesse erreicht werden. Für eine mehrfache Analyse der Textdaten mit unterschiedlich spezialisierten Algorithmen sind mehrere Gateways bereitzustellen. Ihnen werden die zu analysierenden Textdaten nacheinander übergeben. Die jeweils extrahierten Informationen können in den Meta-Daten des analysierten *FraMeTex*-Dataset zusammengefasst werden. Je nach Anwendungsfall ist eine alternative Vorgehensweise jedoch denkbar. Die in Kapitel 6 präsentierten Anwendungsfälle zeigen verschiedene Möglichkeiten auf.

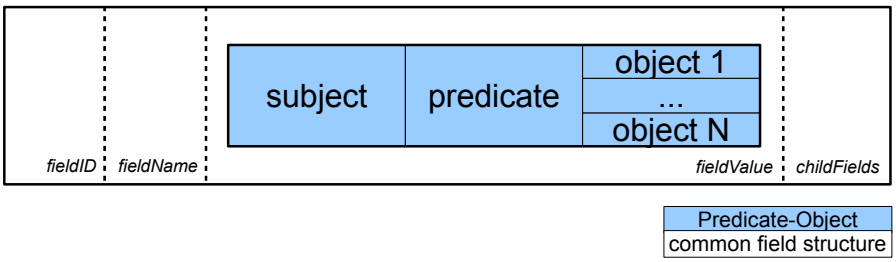


Abbildung 5.7: *Predicate-Object* repräsentiert extrahierte Pathways in konzipierter Feldstruktur

5.2.2 Homogene Repräsentation komplexer Netzwerke

Im vorherigen Abschnitt wurde bereits deutlich, dass die aus Textdaten rekonstruierten Netzwerke in einem (angepassten) *FraMeTex*-Dataset repräsentiert werden. Eine Weiterverarbeitung in nachgelagerten Modulen des Systems (Deduktionskomponente, Abbildung 4.1) wäre sonst nicht möglich (Abschnitt 4.3). Die konzipierte Datenstruktur ist jedoch auf die Darstellung einfacher Schlüssel/Wert-Attribute ausgelegt (Abschnitt 4.2.1) und damit nicht unmittelbar zur Repräsentation komplexer Netzwerke geeignet. Die Netzwerke werden daher in ihre Pathways gegliedert, die zusammen das Netzwerk bilden (Abschnitt 2.1.4). Damit können sowohl beliebig komplexe als auch verschiedene, biologische Netzwerke repräsentiert werden. Zur Darstellung eines einzelnen Pathways wurde das *Predicate-Object* (*P*) geschaffen. Es orientiert sich an der Graphdarstellung in *Semantic Medline* (Abschnitt 2.1.5) sowie der von dem TM-Algorithmus Enju genutzten PAS (Abschnitt 3.1.3). Der Pathway eines Netzwerks wird als Struktur aus Subjekt (s), Prädikat- (p) und Objekten (o_i) beschrieben.

$$P(s, p, o_1 \dots o_n) \tag{5.1}$$

Ein von dieser Struktur repräsentierter Pathway kann durchaus mehr als zwei Knoten umfassen. Besteht in einem Netzwerk zwischen einem Knoten (s) und mehreren anderen Knoten ($o_1 \dots o_n$) eine identische Beziehung (p), können diese Pathways durch ein einziges Predicate-Object dargestellt werden. Den typischen Eigenschaften biologischer Netzwerke wird damit Rechnung getragen. Sie zeichnen sich durch bestimmte Beziehungen aus, die sehr häufig zwischen verschiedenen Knoten gelten. Beispiele sind die Schlüsselwörter *interact* oder *regulate* in Protein-Interaktions- oder auch metabolischen Netzwerken. Das Predicate-Object erlaubt damit nicht nur eine komprimierte Darstellung dieser Beziehungen, sondern auch die Abbildung gerichteter Zusammenhänge. Die durch das Prädikat p repräsentierte Kante verläuft implizit jeweils vom Subjekt s zu einem Objekt o_i. Dies erlaubt insbesondere eine adäquate Abbildung enzymatischer Reaktionen in metabolischen Netzen. Sie überführen stets ein Substrat in ein Produkt, so dass die entsprechende Kante nur in eine Richtung Gültigkeit hat (Abschnitt 2.1.4.1). Die Abbildung nativer Textmining-Resultate auf Predicate-Objekte hängt vom jeweils genutzten Algorithmus ab. Sie erfolgt daher im Rahmen der Wissensaufbereitung (Builder-Instanz, Abschnitt 5.2.1), die konzeptionell dem PPS *ANALYZE* zugeordnet wurde. Eine schematische Darstellung des Vorgangs zeigt Abbildung 5.8.

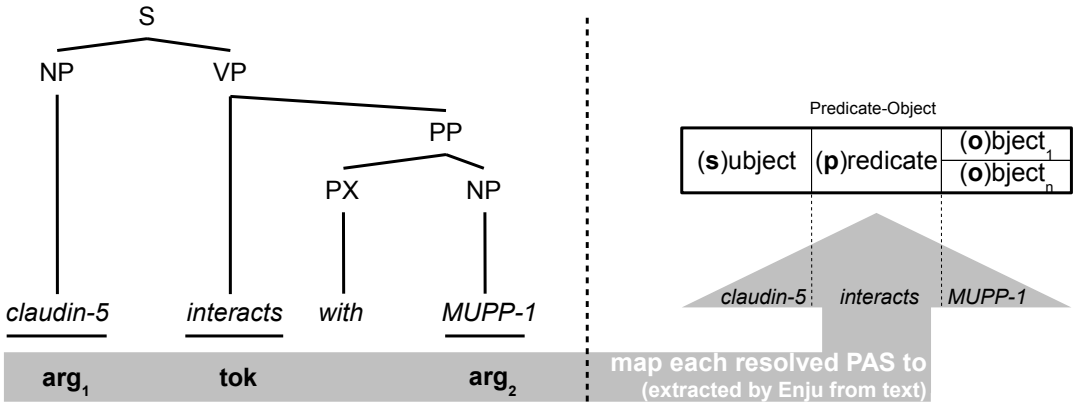


Abbildung 5.8: Abbildung eines aus Textdaten extrahierten Pathways auf ein *Predicate-Object (P)*

Die Analyse der in einem *FraMeTex*-Dataset repräsentierten Textdaten führt im Regelfall zu mehreren Predicate-Objects. Sie werden in einem spezialisiertem *FraMeTex*-Dataset (Knowledge-Dataset) zusammengefasst. Hierfür wurden die Felder der *dataValue*-Komponente (Definition 4.1) auf die Repräsentation von Predicate-Objects ausgelegt (Abbildung 5.7). Der *matchValue* des Knowledge-Dataset wird vom analysierten *FraMeTex*-Dataset übernommen. Die Größe der im Knowledge-Dataset darstellbaren Netzwerke ist nicht begrenzt. Außerdem dient es nicht nur zur Repräsentation der von *FraMeTex* aus Textdaten rekonstruierten Netzwerke. Vielmehr können auch anderweitig rekonstruierte Netzwerke (Abschnitt 3.2) auf diese Datenstruktur abgebildet und damit in *FraMeTex* verarbeitet werden.

5.2.3 Aufbau einer Wissensbasis für biologische Netzwerke

Die im Abschnitt 4.1 formulierten Anforderungen sehen eine Speicherung rekonstruierter Netzwerke in einer Deduktionskomponente vor. Dies umfasst sowohl mit *FraMeTex* aus Textdaten als auch mit anderen Tools (Abschnitt 3.2) vorhergesagte Netzwerke. Voraussetzung ist, dass sie als Knowledge-Dataset repräsentiert werden (Abschnitt 5.2.2). Die Auswirkungen von Inferenzen auf die Netzwerkrekonstruktion sollen damit beurteilt werden.

In der modular konzipierten Systemarchitektur wird die Deduktionskomponente durch den PPS *REASON* repräsentiert (Abbildung 4.1). Die Grundlagen für ihre Realisierung wurden bereits im Abschnitt 4.4 gelegt. Der Fokus der Implementierung liegt damit auf der Bereitstellung einer Loader-Komponente sowie der Umsetzung physischer Datenzugriffe. Die Implementierung nutzt die Konzepte des Semantic Web und baut auf der *Jena API* auf (Abschnitt 3.3.2). Die ausschlaggebenden Gründe für den Einsatz der graphbasierten Semantic Web Technologie wurden im Kontext der Systemkonzeption bereits erörtert (Abschnitt 4.1.1). Ein aus zwei Knoten (s, o) und einer verbindenden Kante (p) gebildeter Pathway wird in der Wissensbasis durch ein Statement repräsentiert und entspricht einem einfachen Graph G (Abschnitt 2.2.2.2):

$$G(s, p, o) \tag{5.2}$$

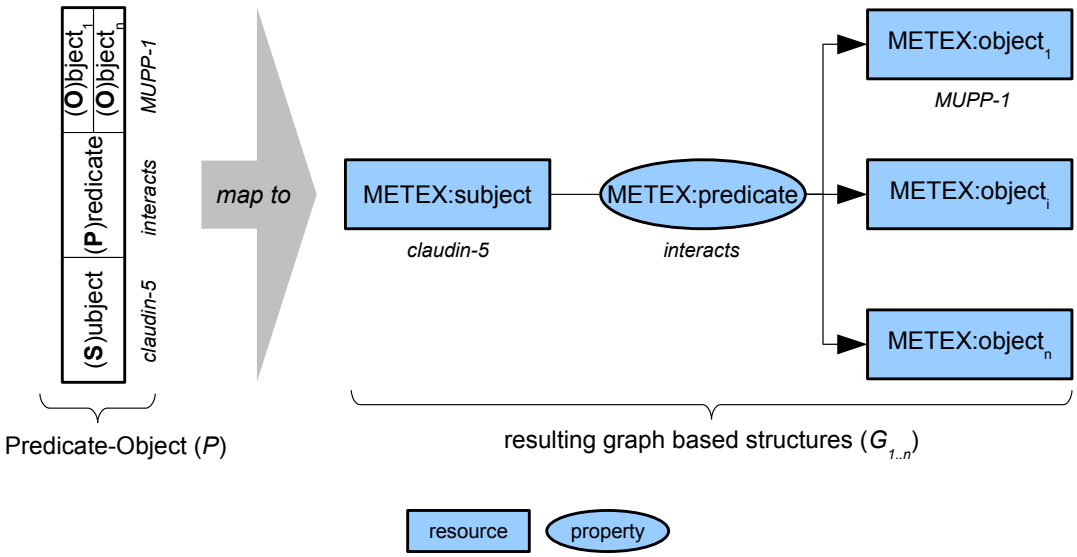


Abbildung 5.9: Speichern eines *Predicate-Objects (P)* in Graphstruktur $G_{1..n}$ der Wissensbasis

Im Gegensatz zum konzipierten Predicate-Object (Definition 5.1) kann der Graph bzw. das Statement jedoch nur *ein* Objekt referenzieren. Die vollständige Abbildung aus Textdaten extrahierter und in einem Predicate-Object P repräsentierter Pathways ist daher definiert durch:

$$P(s, p, o_1 \dots o_n) \longrightarrow G_i(s, p, o_i), i \in \{1, \dots, n\} \tag{5.3}$$

Ein Predicate-Object P wird damit auf mehrere Graphstrukturen G_i abgebildet, die sich nur in ihren Objekten unterscheiden. Voraussetzung hierfür ist die einmalige Definition der erforderlichen Komponenten $K(P)$, die sich aus dem Predicate-Object ergeben:

$$K(P) \in \{s, p, o_1 \dots o_n\} \tag{5.4}$$

Jede dieser Komponenten wird der Wissensbasis einmalig als *Ressource* hinzugefügt und kann anschließend über ihre URI referenziert werden. Die Kante p wird explizit als *Property* gekennzeichnet. Dies erlaubt es später Bedingungen für die entsprechende Kante festzulegen, die beispielsweise beim Schlussfolgern berücksichtigt werden (Abschnitt 2.2.2.2). Die mehrfache Referenzierung einer Ressource durch andere Ressourcen führt zu einer automatischen Vernetzung in der Wissensbasis und ist ein entscheidender Vorteil gegenüber einer deduktiven Datenbank. Die Vernetzung ist bereits anhand der Abbildung eines einzelnen Predicate-Objects auf die Graphstruktur der Wissensbasis erkennbar (Abbildung 5.9). Für die in der Wissensbasis repräsentierten, biologischen Pathways ist dies ein erheblicher Informationsgewinn. Allein dadurch können in metabolischen Netzwerken beispielsweise neue, enzymatische Reaktionen vorhersagt werden. Dies gelingt, sobald das Produkt eines Pathways als Substrat eines anderen Pathways referenziert wird. Die zuvor unabhängigen Pathways werden in diesem Augenblick automatisch verknüpft.

Von besonderem Interesse sind in diesem Fall Informationen, die helfen die Vorhersage der

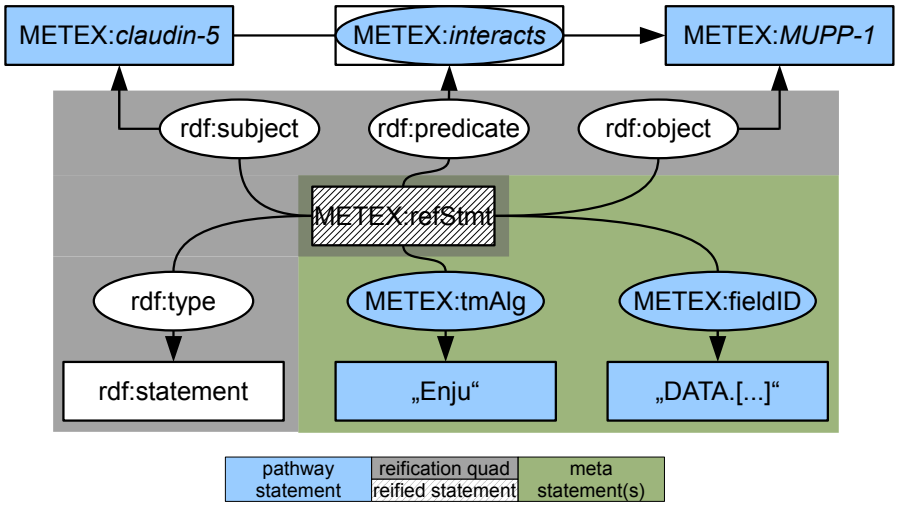


Abbildung 5.10: Konzept der *Reification* ordnet Pathways Meta-Informationen zu

beteiligten Pathways nachvollziehen zu können. Sie können später auch die Validierung komplexer Netzwerke erleichtern. Hierfür wird das Konzept der *Reification* genutzt. Das Konzept erlaubt es, ein Statement in der Wissensbasis als Ressource aufzufassen und Aussagen darüber zu machen [MMM04]. Damit ist es möglich, jedem Statement beliebige Meta-Informationen zuzuordnen. Sie können beispielsweise die zur Extraktion eines Pathways genutzten TM-Algorithmen, das Datum der Publikation oder auch technische Daten umfassen. Die Reification eines Statements zeigt Abbildung 5.10 schematisch. Dem repräsentierten Pathway wurde beispielhaft der eingesetzte TM-Algorithmus sowie seine *fieldID* aus dem Knowledge-Dataset zugeordnet. Sie ist für die Rekonstruktion eines persistierten Knowledge-Datasets durch die Loader-Komponente erforderlich (Abschnitt 4.4). Weitere Meta-Informationen können über das *reified statement* analog zugeordnet werden. Die zusätzlichen Daten können auch die zukünftige Visualisierung der Pathways anreichern. Alle gespeicherten Meta-Informationen sind von den gespeicherten Netzwerken implizit getrennt, da lesende Datenzugriffe *Reified Statements* standardmäßig nicht berücksichtigen. Auf sie muss in Abhängigkeit eines Statements gesondert zugegriffen werden. Die implementierten, physischen Datenzugriffe (Algorithmus 3) müssen dies ebenso berücksichtigen, wie die automatische Zuordnung der *fieldID* zu einem gespeicherten Statement.

Die in der Wissensbasis gespeicherten Pathways unterliegen bisher keiner expliziten Struktur. Sie ist jedoch nützlich, um einzelne Pathways beispielsweise einfacher identifizieren zu können. Aus diesem Grund wurde die Wissensbasis in Subgraphen gegliedert (Abbildung 5.11). Ein Subgraph sub_k umfasst alle Pathways, die durch die Predicate-Objects $P_{1...x}$ eines Knowledge-Datasets k repräsentiert werden. Die Menge der in sub_k repräsentierten Pathways leitet sich daher aus der definierten Abbildung eines einzelnen Predicate-Object auf die Graphstruktur der Wissensbasis (Definition 5.3) ab:

$$sub_k = \bigcup_{j=1}^{x} \bigcup_{i=1}^{n} G_i^j(s^j, p^j, o_i^j) \tag{5.5}$$

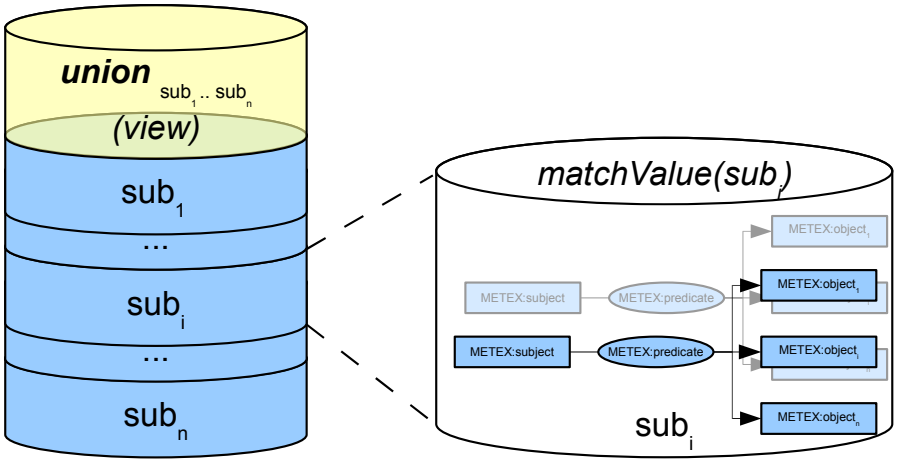


Abbildung 5.11: Strukturierung der Pathways in Subgraphen (Identifizierung anhand des *matchValue*)

Wurde das gespeicherte Netzwerk mit *FraMeTex* aus Medline rekonstruiert, umfasst sub_k die aus einem Abstract extrahierten Pathways: Jeder verarbeitete Medline-Eintrag wird als *FraMeTex*-Dataset repräsentiert (Abschnitt 5.1.1) und die daraus mittels TM-gewonnenen Pathways in einem Knowledge-Dataset zusammengefasst (Abschnitt 5.2.1). Die Vereinigung aller Subgraphen ergibt schließlich alle in der Wissensbasis gespeicherten Pathways. Sie können bereits komplexe Netzwerke formen.

Technisch basiert die Aufteilung in Subgraphen auf dem Konzept der *named graphs*, die anhand eines eindeutigen Schlüssels identifiziert werden können [CBHS05] [BCW05]. Für die Identifikation eines Subgraphen sub_k wird der *matchValue* (Abschnitt 4.2) des jeweils gespeicherten *FraMeTex*-Datasets genutzt. Anhand der Zuordnung eines bestimmten Pathways zu einem Subgraphen, wird damit implizit festgehalten, auf Basis welcher Daten er vorhergesagt wurde. Die entscheidende Information muss damit nicht einzeln für jeden Pathway in dessen Meta-Informationen festgehalten werden. In Kombination mit der in den Meta-Daten hinterlegten *fieldID* (Abbildung 5.10) ist ein gezieltes Aktualisieren und Revidieren einzelner Pathways möglich, da die *fieldID* einen Pathway innerhalb eines Subgraphen eindeutig identifiziert.

5.2.4 Schlussfolgern in biologischen Netzwerken

Die bisher konzipierte Wissensbasis kann Pathways strukturiert speichern und zu Netzwerken zusammenführen (Abschnitt 5.2.3). Ein Schlussfolgern in den Netzwerken (Anforderungen 1 & 2, Abschnitt 4.1) wird jedoch erst mit zusätzlichen Regeln möglich, die von einem Reasoner angewendet werden (Abschnitt 2.2.3). Die nachfolgende Implementierung wurde von der Idee geleitet, Regeln sowohl existierenden Wissensressourcen entnehmen als auch individuell formulieren zu können (Anforderung 6). Auf die Details einiger Wissensressourcen, die zum Schlussfolgern in biologischen Netzwerken in Betracht kommen, wurde daher bereits im Abschnitt 3.3.4 eingegangen[4]. Der verfolgte Ansatz sieht eine konsequente Trennung der rekon-

[4]Die Wissensressourcen müssen den Formalismen des Semantic Web unterliegen.

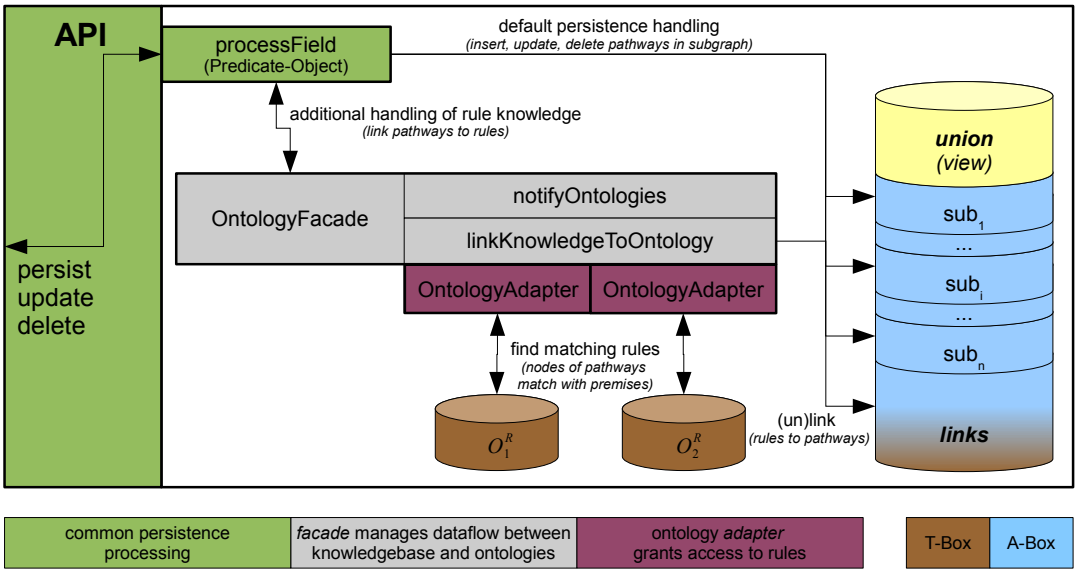


Abbildung 5.12: Datenfluss bei Verknüpfung rekonstruierter Netzwerke mit Regelwissen in Ontologien

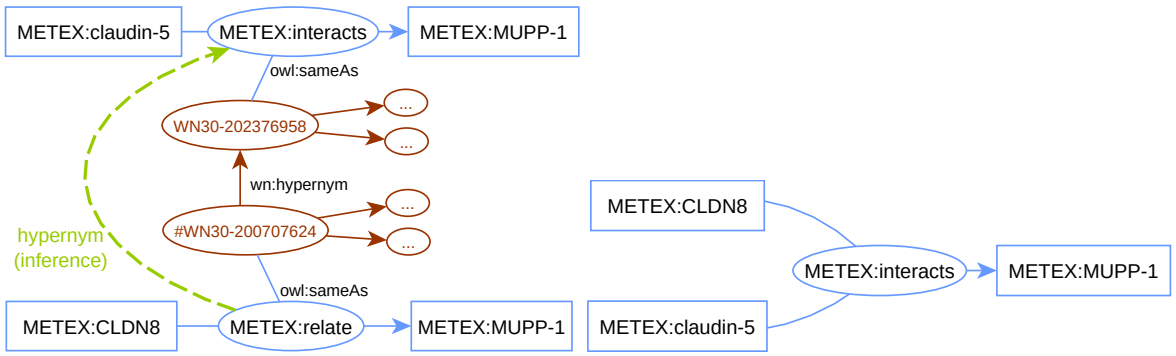
struierten Netzwerke und Regeln vor (A-, T-Box, Abschnitt 2.2.3). Regeln werden in Ontologien repräsentiert (O_i^R), die mit den Pathways rekonstruierter Netzwerke explizit verknüpft werden müssen. Die Verknüpfung erfolgt bereits beim Speichern der Pathways. Der Vorgang wird vom generalisierten Datenmanipulations-Algorithmus koordiniert[5], der für die Textdatenbank und Wissensbasis konzipiert wurde (Abschnitt 4.4).

Den Datenfluss beim Speichern und Verknüpfen eines Pathways zeigt Abbildung 5.12 schematisch. Eine Facade (grau) bündelt den Zugriff auf alle zu berücksichtigen Ontologien und leitet über deren Adaptoren (lila) den aktuell in der Wissensbasis verarbeiteten Pathways an sie weiter (*notifyOntologies*). In den Ontologien kann hierauf reagiert werden, um für den im Predicate-Object (P) repräsentierten Pathway zutreffendes Regelwissen zu identifizieren. Im Anschluss wird dieses Regelwissen mit dem Pathway verknüpft (*linkKnowledgeToOntology*). Die Verknüpfungen werden in einem eigenem Subgraphen der Wissensbasis verwaltet (*links*). Ein Reasoner kann die Verknüpfungen auswerten und erlangt darüber Zugriff auf das Regelwissen der Ontologien. Werden Pathways aus der Wissensbasis gelöscht, führt dies auch zum Entfernen definierter Verknüpfungen. Zur Identifikation des zutreffenden Regelwissens ist ein spezielles *searchProperty* in der Konfiguration des zuständigen Ontologie-Adapters (Abbildung A.4) zu definieren. Dieses kennzeichnet in O_i^R ein Property, das sämtliche $K(P)$ (Definition 5.4) mit Ressourcen in O_i^R verbindet. Dies führt zur Identifikation derjenigen Statements, für die gilt:

$$X \xrightarrow{searchProperty} K(P) \mid X \in O_i^R \wedge K(P) \in \{s, p, o_1 \ldots o_n\} \tag{5.6}$$

Die Subjekte X dieser Statements in O_i^R sind äquivalent zu denjenigen Ressourcen in der Wissensbasis, die dort zur Repräsentation von $K(P)$ definiert wurden (Abschnitt 5.2.3). Ihre Äquivalenz wird mit der symmetrisch definierten Property *owl:sameAs* ausgedrückt. Bezieht sich

[5] Ausgangspunkt für die Verknüpfung mit Regelwissen ist daher die Methode *processField* (Abbildung 5.12).



(a) Pathways mit Wordnet-Ontologie verknüpft (b) logische Auswertung der *hypernym*-Relation

Abbildung 5.13: Exemplarische Inferenz in rekonstruierten Pathways

das Regelwissen unmittelbar auf die Pathways der gespeicherten Netzwerke, kann diese explizite Verknüpfung und damit auch die Angabe des *searchProperty* entfallen. Dies ist zumeist für eigenständig formulierte Regeln der Fall.

Das erarbeitete Vorgehen stellt Abbildung A.3[6] anhand der Wordnet-Ontologie für einen Pathway ausführlich dar. Eine kompakte Darstellung des Vorgangs sowie die sich für diesen und einen weiteren Pathway ergebenden Auswirkungen durch das Regelwissen zeigt Abbildung 5.13. Die in Wordnet definierte Taxonomie (*wn:hypernym*) wird hier zur Unterstützung der Netzwerkrekonstruktion herangezogen: Sobald ein Reasoner die Taxonomie auswertet, kann das Verb *relate* auf das allgemeinere Verb *interacts* zurückgeführt werden. Damit können schließlich beide Protein-Interaktionen durch das Verb *interact* ausgedrückt werden. Insbesondere in großen Netzwerken kann eine derartige Kondensierung zu kompakteren Netzwerken führen und deren Auswertung erleichtern. Das abstrakte Verfahren wird anhand der deduktiv unterstützten Rekonstruktion verschiedener, biologischer Netzwerke im Kapitel 6 noch einmal im Detail deutlich.

Die Verknüpfung mit existierenden Wissensressourcen wie Wordnet bringt zwei Herausforderungen mit sich. Einerseits können die Regeln zwar intuitiv verständlich, aber nicht maschinenlesbar kodiert sein, anderseits kann die Auswertung des Regelwissens äußerst komplex werden. Interpretiert werden stets alle in einer Ontologie definierten Zusammenhänge und nicht nur die explizit mit Pathways verknüpften. Beiden Herausforderungen kann über die Konfiguration des zuständigen Ontologie-Adapter begegnet werden (Abbildung A.4). Sie ermöglicht es, bestimmten Beziehungen in einer Ontologie funktionale, symmetrische oder transitive Eigenschaften zuzuweisen [MPSP+09]. Erst danach kann beispielsweise die Transitivität der *hypernym*-Relation in Wordnet automatisch interpretiert werden. Außerdem kann über die Konfiguration eine Filterung des zu berücksichtigenden Regelwissens erfolgen. In diesem Fall werden die vom *searchProperty* identifizierten Ressourcen (Definition 5.6) nicht nur verknüpft, sondern die darüber verfügbaren Aussagen außerdem in eine eigenständige Teil-Ontologie übernommen. Zum Schlussfolgern wird dann nur diese Teil-Ontologie berücksichtigt. Gerade für sehr große Wis-

[6] Farbgebung orientiert sich an Abbildung 5.12

Darstellung	Bedeutung
ovaler Knoten	Protein, Gene, Enzym usw. (Subjekt, Prädikat oder Objekt eines *Predicate-Object*)
schwarze Kante (gerichtet: vom Subjekt zum Objekt)	aus Textdaten rekonstruierter Pathway (Prädikat-Knoten beschreibt Relation)
grüne Kante (gestrichelt) (gerichtet: vom Subjekt zum Objekt)	durch Deduktion gewonnener Pathway (Prädikat-Knoten beschreibt Relation)
Kantennummerierung (optional)	Identifikation eines Pathways (nur aus Textdaten rekonstruierte Pathways)
rote Kante oder Knoten	aus Textdaten unvollständig rekonstruierter Pathway

Tabelle 5.1: Visualisierung rekonstruierter, biologischer Netzwerke und Pathways

sensressourcen kann der Inferenzprozess dadurch erheblich beschleunigt werden. Anhand der in Abbildung A.3 ausführlich dargestellten Verknüpfung eines Pathways mit Regelwissen lässt sich die Übernahme leicht nachvollziehen. Im ersten Schritt werden Statements mit dem Subjekt *#WN30-202376958* übernommen, die optional anhand ihrer Properties (z.B. hypernym) gefiltert werden können. Ein rekursives Verfahren übernimmt anschließend abhängige Statements, um den Kontext zu wahren.

5.2.5 Exploration biologischer Netzwerke

Die Exploration rekonstruierter Netzwerke muss die Eigenschaften der konzipierten Wissensbasis berücksichtigen. Die aus Textdaten extrahierten Pathways sind in Subgraphen gegliedert (Abschnitt 5.2.3) und können über *link*-Statements mit weiteren Ontologien verknüpft sein. Sie enthalten potentielles Regelwissen, das von einem Reasoner interpretiert und zur Ableitung weiterer Pathways führen kann (Abschnitt 5.2.4). Der Exploration eines rekonstruierten Netzwerks gehen daher stets drei Verarbeitungsschritte voraus:

1. Vereinigung aller Subgraphen (*union*).
2. Interpretation der definierten Verknüpfungen zu weiteren Ontologien.
3. Auswertung des in den Ontologien verfügbaren Regelwissens.

Erst im Anschluss steht das vollständig vom System rekonstruierte Netzwerk zur Verfügung. Neben den aus Textdaten extrahierten umfasst dies auch die mittels Deduktion zusätzlich gewonnenen Pathways. Je nach Umfang und Komplexität der Subgraphen und Ontologien kann der Inferenzprozess einige Zeit in Anspruch nehmen. Inferenzen werden daher erst zum Zeitpunkt der ersten Suchanfrage berechnet. Hierbei muss jedoch sichergestellt werden, dass nachträgliche Änderungen an Pathways auch zur Aktualisierung des gesamten Netzwerks führen.

```
DESCRIBE <METEX:keyword>
```

Abbildung 5.14: Einfache SPARQL-Query fasst spezifische Pathways in einem Netzwerk zusammen

Für jeden Subgraphen ist daher ein Listener registriert (*Observer-Pattern* [GHJV04]), der bei Bedarf eine Aktualisierung veranlasst.

Eine Grundvoraussetzung für die Netzwerk-Exploration ist eine Visualisierung. Eine leistungsfähige Visualisierung ist jedoch nicht trivial zu realisieren und wurde von den Zielen dieser Arbeit bewusst ausgenommen (Abschnitt 1.2). Außerdem existiert mit VANESA (Abschnitt 3.2.1) bereits ein ausgezeichnetes Tool, das auf diese Aufgabe spezialisiert ist. Es wurde daher nur eine statische Darstellung der Netzwerke verfolgt, um das Deduktionspotential in rekonstruierten Netzwerken beurteilen zu können (Anforderung 1, Abschnitt 4.1). Die Pathways eines Netzwerks werden mit Hilfe der *Graphviz-Engine* [EGK+02] nach definierten Vorgaben abgebildet (Tabelle 5.1). Eine derartige Visualisierung wird auch bereits von dem System UM-PPS verfolgt (Abschnitt 3.2.4). Zusätzlich erfolgt jedoch eine manuelle Nachbearbeitung der generierten Netzwerke mit dem frei verfügbaren Graph-Editor *yEd* [WEK04]. Deduktiv gewonnene Pathways werden naiv identifiziert: Sie sind in keinem Subgraphen enthalten.

Pathway-orientierte Auswertung

Zur unkomplizierten Auswertung rekonstruierter Netzwerke wurde eine Schlagwortsuche realisiert. Anhand benutzerdefinierter Schlüsselwörter (Protein, Enzym, Regulation) können spezifische Pathways eines Netzwerk selektiert werden. Der Zugriff erfolgt mit einer einfachen SPARQL-Anfrage (Abschnitt 2.2.2.2, Abbildung 5.14). Alle mit der Abfrage in der Deduktionskomponente selektierten Statements (Pathways) weisen das Schlüsselwort als Subjekt auf und werden automatisch zu einer Graphstruktur zusammengefasst. Für die Prädikate und Objekte der resultierenden Statements bzw. Pathways kann das Vorgehen analog wiederholt werden. Damit ist eine schrittweise Exploration komplexer, rekonstruierter Netzwerke möglich. Die jeweiligen Schlagwörter müssen allerdings stets erneut angegeben und die Suchoperationen manuell ausgeführt werden. Genutzt wird hierfür die Methode *findByQuery* (Abbildung 4.5), die bereits während der Systemkonzeption für den Zugriff auf die Textdatenbank sowie Wissensbasis definiert wurde (Tabelle 4.2, Abbildung 4.5). Wird ihr beispielsweise der Proteinname *claudin-5* übergeben, umfasst das Ergebnis die in Abbildung A.3 dargestellte Interaktion mit dem Protein *MUPP1*.

Selbstverständlich kann das in Abbildung 5.14 gezeigte Anfragemuster auch entsprechend der geltenden SPARQL-Syntax verändert werden. Das Suchmuster könnte beispielsweise noch um das Prädikat *interacts* ergänzt werden, um spezifischere Pathways zu selektieren. Außerdem besteht in SPARQL die Möglichkeit eine *SELECT*-Anfrage zu formulieren. Im Gegensatz zum zuvor beschriebenen Verfahren liefert sie nicht stets vollständige Statements bzw. Pathways.

Damit kann die Ergebnismenge beispielsweise auf interagierende Proteine oder katalysierende Enzyme reduziert werden. Dies ermöglicht es, rekonstruierte Netzwerke noch individueller auszuwerten. Ein anschaulicher Anwendungsfall wird im Abschnitt 6.2.2 präsentiert und ausführlich diskutiert.

5.3 Zusammenfassung

In diesem Kapitel wurde die prototypische Implementierung der zuvor konzipierten *Pathway Prediction* (Kapitel 4) vorgestellt. Im Abschnitt 5.1 wurde zunächst die Identifikation sowie Selektion zu analysierender Textdaten erörtert. Die geschaffenen Funktionalitäten decken die beiden PPSs *ADAPT* und *FILTER* ab (Abbildung 4.1). Ihre wesentlichen Eigenschaften wurden bereits im Abschnitt 5.1.3 zusammengefasst. Die Analyse selektierter Textdaten und damit die Rekonstruktion biologischer Netzwerke aus diesen Daten stand anschließend im Abschnitt 5.2 im Fokus. Im Abschnitt 5.2.1 wurde zunächst gezeigt, wie beliebige TM-Algorithmen zur Analyse der Textdaten genutzt werden können. Ein spezielles TM-Gateway integriert die Algorithmen ins System und bildet damit den konzipierten PPS *ANALYZE* ab. Die Möglichkeit verschiedene Algorithmen nutzen zu können, bietet eine größtmögliche Flexibilität. In Abhängigkeit der genutzten TM-Algorithmen kann das System damit verschiedene, biologische Netzwerktypen (Abschnitt 2.1.4) rekonstruieren. Die Verarbeitung unterschiedlicher Netzwerke im System erfordert jedoch deren einheitliche Repräsentation, auf die im Abschnitt 5.2.2 eingegangen wurde. Die Netzwerke werden hierfür in ihre Pathways gegliedert, die jeweils von einem *Predicate-Object* innerhalb des flexiblen *FraMeTex*-Dataset repräsentiert werden. Anhand einer extrahierten Protein-Interaktion wurde das Vorgehen exemplarisch verdeutlicht.

Die ausgewählte Protein-Interaktion diente im Abschnitt 5.2.3 auch zur Präsentation der implementierten Deduktionskomponente. Sie setzt den konzipierten PPS *REASON* um und basiert auf den Konzepten des Semantic Web. Die Deduktionskomponente unterstützt damit auch Schlussfolgerungen. Hierfür müssen die extrahierten Pathways zunächst jedoch auf graphbasierte Strukturen abgebildet werden. Dies sorgt für eine implizite Vernetzung der einzelnen Pathways zu einem komplexen Netzwerk. Die Wissensbasis wurde daher in Subgraphen strukturiert. Wurde das Netzwerk mit *FraMeTex* selbst aus Medline-Daten rekonstruiert, fast der Subgraph alle aus einem Medline-Eintrag extrahierten Pathways zusammen. Die Wissensbasis ermöglicht aber auch die Verarbeitung anderweitig rekonstruierter Netzwerke, solange sie in den definierten Datenstrukturen repräsentiert werden. Zu jedem Pathway können zusätzlich Meta-Informationen gespeichert werden, die für die spätere Evaluation der vorhergesagten Netzwerke nützlich sein können. Zusammen mit der Gliederung eines komplexen Netzwerks in Subgraphen, ermöglichen sie bei Bedarf außerdem eine gezielte Aktualisierung einzelner Pathways. Im Abschnitt 5.2.4 wurde schließlich auf das Schlussfolgern in den gespeicherten Netzwerken eingegangen. Es setzt Regelwissen voraus, dass in Ontologien formuliert oder bereitgestellt werden muss. Bei Bedarf kann das Regelwissen gefiltert werden, um die Komplexität des Schlussfolgerns zu reduzieren. Zum Ende des Kapitels wurde im Abschnitt 5.2.5 auf die Exploration der aus Textdaten rekonstruierten sowie durch Deduktion gewonnenen Pathways

eingegangen. Nach fest definierten Regeln erfolgt eine statische Visualisierung der Netzwerke. Unterschiedliche Farbgebungen und Darstellungen ermöglichen die Identifikation einzelner Pathways sowie deduktiv gewonnener Zusammenhänge.

6 Anwendungsfälle

In diesem Kapitel wird die Rekonstruktion biologischer Netzwerke mit dem konzipierten (Kapitel 4) und implementierten (Kapitel 5) Prototyp demonstriert. Zu Beginn wird im Abschnitt 6.1 zunächst die gewinnbringende Flexibilität der unabhängig konzipierten Module demonstriert. Im Abschnitt 6.2 wird anhand eines MPDZ/MUPP1-Netzwerks das Potential der geschaffenen Deduktionskomponente exemplarisch herausgestellt. Das spezielle Protein wird mit Herz- und Gefäßerkrankungen in Verbindung gebracht und war bereits Gegenstand verschiedener Rekonstruktionen [Kor10] [STK+10]. Die Rekonstruktion eines metabolischen Netzwerks sowie die logische Interpretation enzymatischer Reaktionen durch die Deduktionskomponente im Abschnitt 6.3 schließen dieses Kapitel ab.

Allen gezeigten Anwendungsfällen ist eine initiale Aufbereitung der Medline-Daten vorausgegangen. Entsprechend der Zielstellung dieser Arbeit beschränken sich die zu analysierenden Textdaten zunächst auf diese Datenquelle (Abschnitt 1.2). Für eine erste Evaluation des Prototyps wurden sämtliche Medline-Einträge in der geschaffenen Textdatenbank persistiert (Abschnitt 5.1.2), die nach 1990 publiziert wurden. Relevante Einträge können damit für die Rekonstruktion einzelner Netzwerke anhand von Schlagwörtern identifiziert werden[1].

6.1 Interaktive Rekonstruktion von Protein-Interaktions-Netzwerken

Das Ziel einer Masterarbeit in der AGBI war es, eine interaktive Rekonstruktion von Protein-Interaktionsnetzwerken zu realisieren. Hierfür wurde ein webbasiertes Tool geschaffen, das sich der modularen Komponenten von *FraMeTex* bedient [Wit13]. Die Entwicklung des Tools konnte dadurch erheblich beschleunigt werden und sich auf die Umsetzung des interaktiven Rekonstruktionsprozess konzentrieren. Verschiedene, studentische Projekte optimierten das System im Anschluss und führten schließlich zu dem spezialisierten sowie frei zugänglichem Tool *FraMeTex Pathway Prediction (FraMeTexPP)*[2]. Das System verzichtet auf eine Speicherung der rekonstruierten Netzwerke und berücksichtigt den konzipierten PPS *REASON* daher

[1] Der zeitintensive, aber unkomplizierte Aufbau der Textdatenbank (Abschnitt 5.1) wird nicht näher beleuchtet. Dies gilt analog auch für die Visualisierung der Netzwerke, deren Knoten zur besseren Übersicht ohne Namespace dargestellt werden (Abschnitt 5.2.5).

[2] http://tunicata.techfak.uni-bielefeld.de/frametexPP

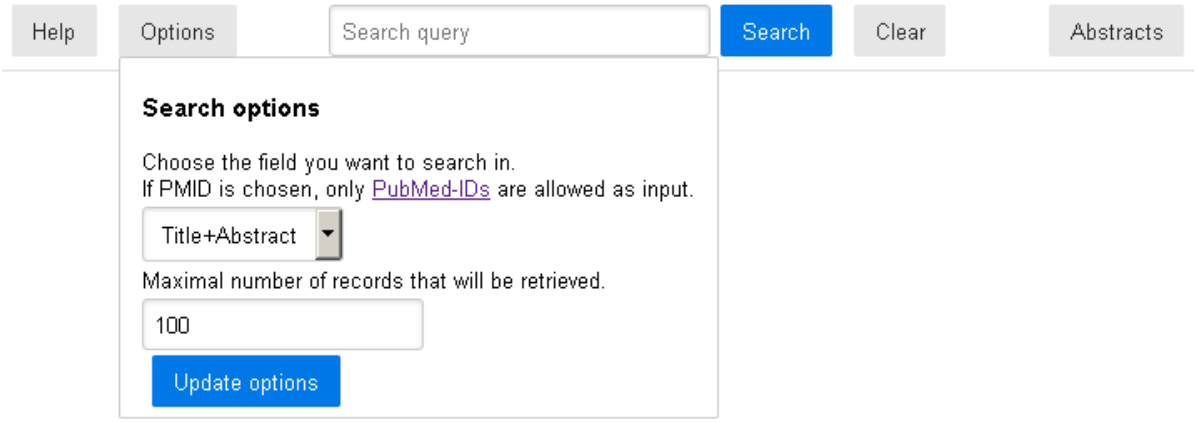


Abbildung 6.1: Selektion der zur Netzwerkrekonstruktion zu analysierenden Textdaten

nicht. Der nachfolgende Anwendungsfall demonstriert damit insbesondere die Funktionalität der konzipierten Filterung und Analyse der zur Netzwerkrekonstruktion herangezogenen Textdaten.

Jede Rekonstruktion beginnt mit der Selektion der zu analysierenden Textdaten. In der zentralen Suchmaske der Anwendung kann hierfür ein Suchbegriff spezifiziert werden. Zusätzlich kann festgelegt werden, über welche Felder sich die Suche in den zuvor aufbereiteten Medline-Daten erstrecken soll (Abbildung 6.1). Die anhand der Suche identifizierten Datensätze werden anschließend zur Netzwerkrekonstruktion herangezogen. Für das in Abbildung 6.2 gezeigte Netzwerk wurde beispielsweise im Titel und Abstract sämtlicher Datensätze nach dem Protein MUPP1 gesucht. Nachdem ein Knoten des rekonstruierten Netzwerks ausgewählt wurde, kann die Anwendung grundlegende Informationen über die hierfür analysierten Datensätze (Abstracts) anzeigen. Die präsentierten Informationen umfassen neben dem Titel auch die PMID eines Datensatzes und bieten die Möglichkeit, über einen einfachen Link auf den kompletten Eintrag via PubMed[3] zuzugreifen. Abbildung 6.2 stellt dies für den im Netzwerk ausgewählten Knoten MUPP1 (blau) dar. Es ist deutlich zu erkennen, dass dieses Protein in mehreren Datensätzen gefunden wurde, die allesamt in die Rekonstruktion des MPDZ/MUPP1-Netzwerks eingeflossen sind. In dem rekonstruierten Netzwerk werden die erkannten Beziehungen zwischen den Knoten zunächst nicht angezeigt. Die entsprechenden Kanten (z.B. *regulates*, Abbildung 6.2) werden erst beschriftet, wenn ein beteiligter Knoten mit der Maus ausgewählt wird (*Mouse-Over*). In komplexen Netzwerken erhöht dieses Vorgehen die Übersichtlichkeit.

Die von FraMeTexPP verfolgte *Pathway Prediction* nutzt zwei spezialisierte TM-Algorithmen. Zunächst wird jeder Medline-Abstract mit *Enju* (Abschnitt 3.1.3) analysiert. Die Analyse liefert mehrere PASs (Abbildung 3.3), die anhand ihrer Argumente sowie ihres Prädikates gefiltert werden. Den Umfang sowie die Abstraktionsebene der von Enju extrahierten PASs zeigt Abbildung A.5 exemplarisch. Ziel ist es, diejenigen PAS zu identifizieren, die mit großer Wahrscheinlichkeit eine potentielle Protein-Interaktion beschreiben. Dies trifft am ehesten auf PASs zu, die

[3] http://www.ncbi.nlm.nih.gov/pubmed

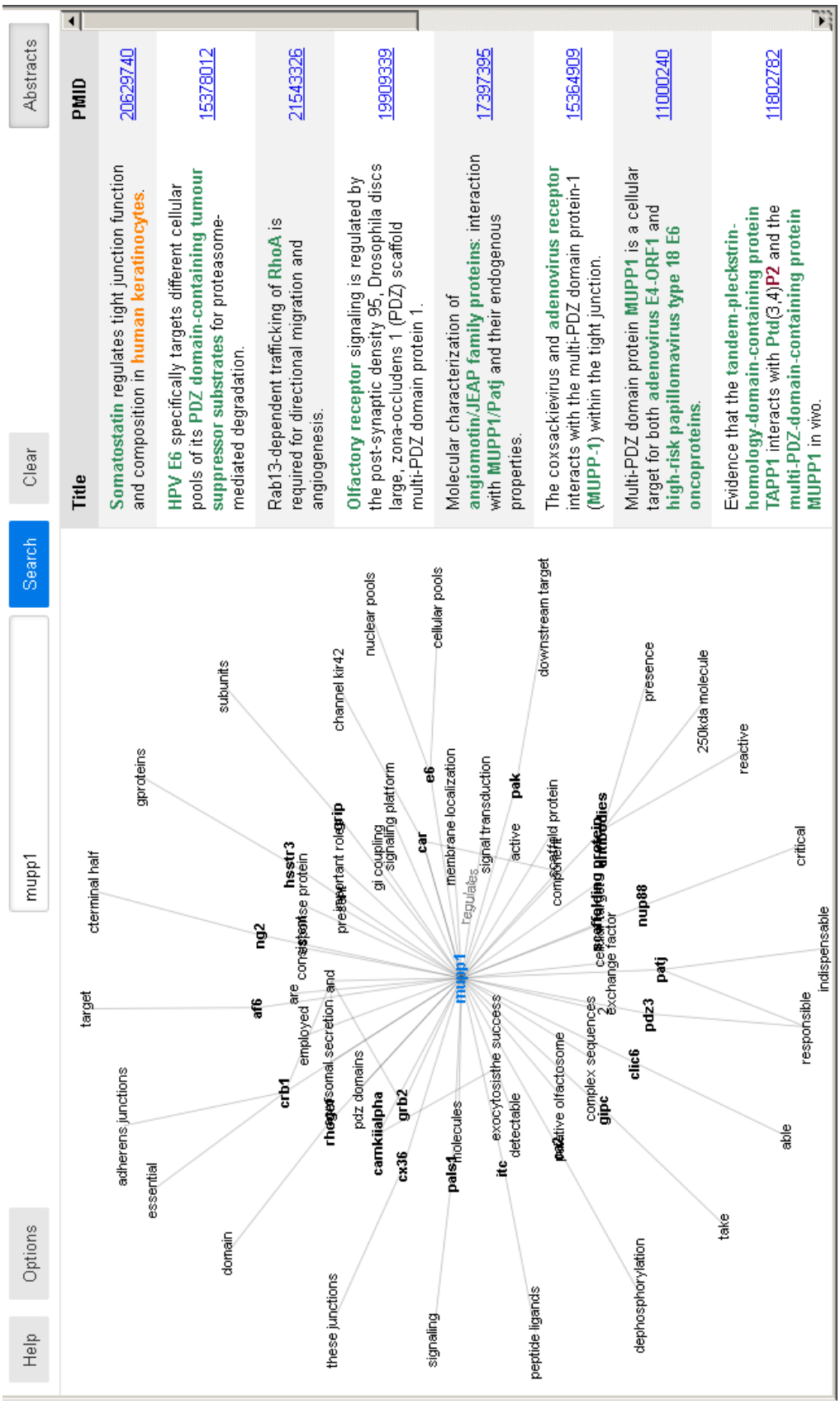


Abbildung 6.2: Mittels *FraMeTexPP* interaktiv rekonstruiertes MPDZ/MUPP1-Netzwerk mit selektiertem Protein (blau hervorgehoben)

exakt zwei Argumente (Proteine) mit einem geeignetem Prädikat (Interaktion) in Relation setzen. Jede extrahierte PAS wird daher zunächst auf die Anzahl ihrer Argumente überprüft, bevor anschließend deren Prädikat näher betrachtet wird. Nur wenn es einem Verb entspricht und damit überhaupt eine potentielle Interaktion beschreiben kann (z.B. *regulate, interact, inhibit*), wird eine PAS weiter berücksichtigt. Die zur Prüfung erforderlichen Informationen können den Analyse-Ergebnissen von Enju unmittelbar entnommen werden (Abbildung 3.3). Bevor eine detektierte Interaktion jedoch einem rekonstruierten Netzwerk hinzugefügt wird, erfolgt mit Hilfe des externen TM-Webservice *GENIA*[4] [KOT+04] eine weitergehende Prüfung. GENIA ermöglicht eine NER (Abschnitt 2.2.1) und identifiziert potentielle Proteine. Nur wenn zumindest ein Argument der zuvor gefilterten PAS einem Protein entspricht, wird die von ihr repräsentierte Interaktion dem rekonstruierten Netzwerk endgültig hinzugefügt. Anhand des exemplarisch ausgewählten Pathways in Abbildung 6.2 (Kante beschriftet) kann die verfolgte Rekonstruktion nachvollzogen werden:

$$mupp1 \xrightarrow{regulates} signal\ transduction \tag{6.1}$$

Der Pathway wurde aus dem Abstract des Medline-Eintrag 18378672 (PMID) extrahiert. Dies kann nachvollzogen werden, wenn der Knoten *signal transduction* im Netzwerk ausgewählt wird und die zugrundeliegenden Abstracts eingeblendet werden. Die entscheidende Information ist in folgendem Satz des Abstracts enthalten (unterstrichen):

> *In this study, we show that mupp1 binds to the G protein-coupled MT(1) melatonin receptor and directly regulates its G(i)-dependent signal transduction.*

Enju konnte aus diesem Satz eine PAS extrahieren, die exakt zwei Argumente aufweist (*mupp1, signal transduction*) und deren Prädikat darüber hinaus einem Verb entspricht (*regulates*). Außerdem wurde *mupp1* von GENIA als Protein identifiziert, das daher in den eingeblendeten Titeln der analysierten Medline-Abstracts farblich hervorgehoben ist (Abbildung 6.2). Damit sind alle eingangs formulierten Voraussetzungen erfüllt, um den Pathway als Vorhersage im Netzwerk aufzunehmen. Trotz der umfangreichen Filterung kann jedoch nicht abschließend sichergestellt werden, dass die Vorhersage auch eine valide Protein-Interaktion beschreibt. Einerseits kann der probabilistische NER-Prozess fehlerbehaftet sein, andererseits beschreibt nicht jedes Verb eine Protein-Interaktion. Formal führt das konzipierte Vorgehen daher nicht zu einem reinem Interaktions-Netzwerk.

Die Visualisierung des Netzwerks basiert auf dem Javascript Framework *Data-Driven Documents (D3)*[5]. Es ist auf webbasierte Darstellungen spezialisiert [BOH11]. Von GENIA erkannte Proteine werden im Netzwerk durch fett gedruckte Knoten repräsentiert. Sie sind der Ausgangspunkt für die interaktive Exploration eines initial rekonstruierten Netzwerks. Sobald sie mit der Maus ausgewählt werden und zugleich die *Steuerung (Strg)*-Taste gedrückt gehalten wird, erfolgt eine Erweiterung des Netzwerks. Für das vom Knoten repräsentierte Protein werden erneut zutreffende Einträge aus der Textdatenbank selektiert und potentielle Protein-Interaktionen ermittelt. Sie führen zu einer automatischen Aktualisierung des bestehenden Netz-

[4] http://www.nactem.ac.uk/GENIA/tagger/
[5] http://d3js.org/

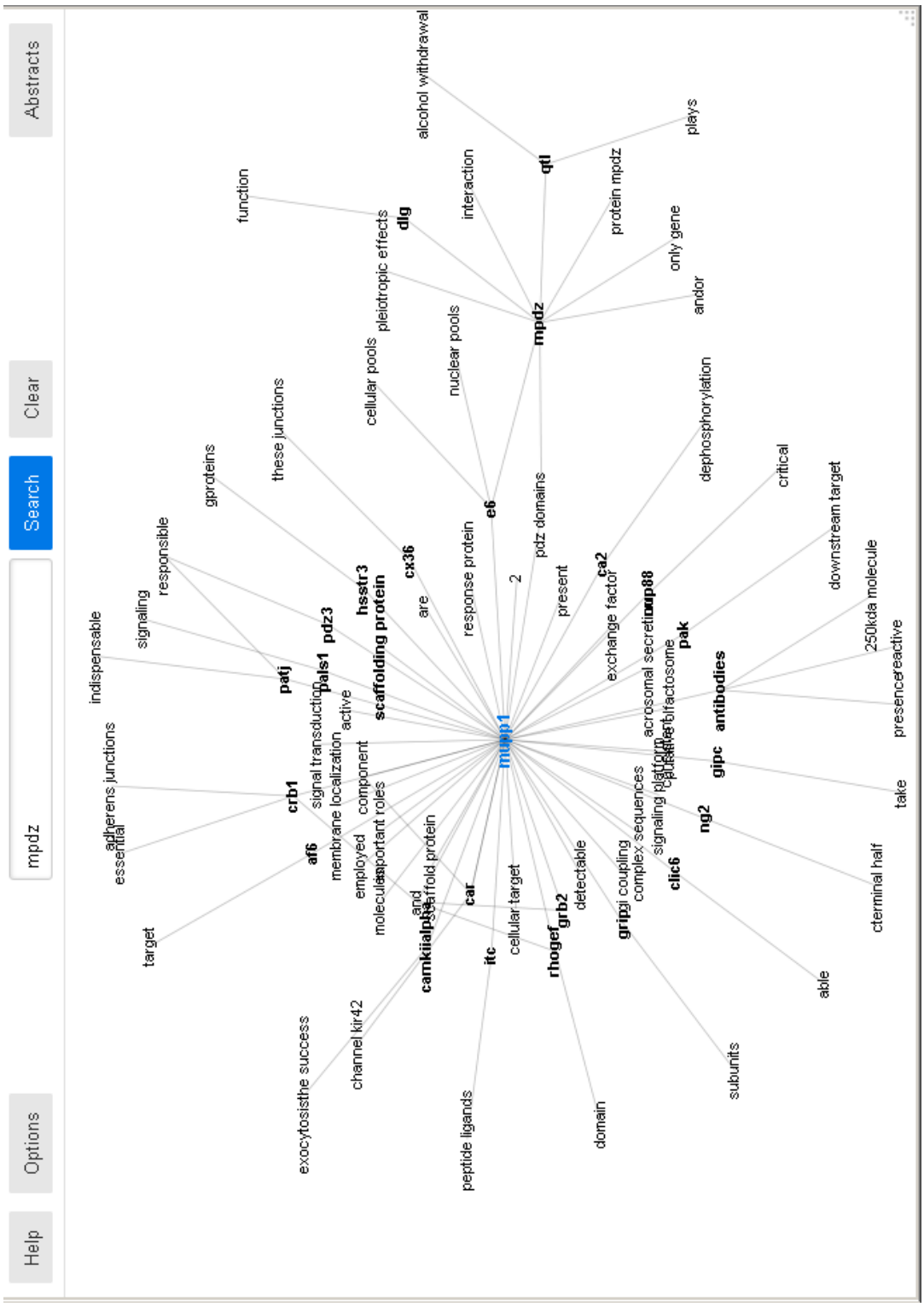


Abbildung 6.3: Erweiterung des zuvor rekonstruierten MUPP1-Netzwerks nach erneuter Suche mit synonymen Protein-Namen MPDZ

werks. Ausgehend vom initial über die Suchmaske selektierten Protein ist damit eine interaktive Netzwerk-Exploration möglich. Alternativ kann auch ein neuer Suchbegriff in der GUI spezifiziert werden. In diesem Fall wird das bereits rekonstruierte Netzwerk ebenfalls mit den neu extrahierten Informationen verschmolzen. Abbildung 6.3 zeigt die Erweiterung des bereits rekonstruierten MUPP1-Netzwerks, die sich ergibt, wenn zusätzlich nach dem Synonym MPDZ gesucht wird. In Abhängigkeit der aus der Textdatenbank selektierten Daten kann die Netzwerk-Rekonstruktion allerdings einige Zeit in Anspruch nehmen. Bereits im Rahmen der Masterarbeit wurde daher eine einmalige Analyse der Daten sichergestellt. Liegen die Ergebnisse für einen Datensatz bereits vor, wird er von einer erneuten Analyse ausgenommen. Insbesondere für häufig analysierte Proteine konnte die Netzwerkrekonstruktion dadurch erheblich beschleunigt werden.

Die Entwicklung der webbasierten Anwendung überschnitt sich mit der Fertigstellung von *FraMeTex*. FraMeTexPP reflektiert daher die finalen Konzepte nicht vollständig. Dies zeigt sich hauptsächlich bei der Anbindung der zur Netzwerkrekonstruktion genutzten TM-Ressourcen über das TM-Gateway (Abschnitt 5.2.1).

6.2 Vorhersage von Protein-Komplexen in MPDZ/MUPP1-Netzwerken

In diesem Abschnitt wird die geschaffene Deduktionskomponente zur Vorhersage von Protein-Komplexen (Quartärstrukturen) genutzt. Sie gelten als Auslöser von Krankheiten oder übernehmen spezifische Funktionen in einer Zelle. Für die Biologie und Medizin sind sie daher von besonderem Interesse. Eine bekannte Quartärstruktur ist die DNS-Polymerase (Abschnitt 2.1.3). Sie setzt sich aus regulatorischen Proteinkomponenten (Silencer, Enhancer, Repressoren, Induktoren) sowie Signalmolekülen zusammen.

Ausgangspunkt für die Vorhersage der Komplexe sind die beiden MPDZ/MUPP1-Netzwerke (Anforderung 1, Abschnitt 4.1), die bereits mit ANDSystem und VANESA rekonstruiert und evaluiert wurden [Kor10] [STK+10]. In diesem Abschnitt wird mit *FraMeTex* zunächst ein drittes MPDZ/MUPP1-Netzwerk rekonstruiert, dass anschließend ebenfalls in die Vorhersage mit einbezogen wird. Damit werden in der Deduktionskomponente Netzwerke verarbeitet, die zuvor mit drei verschiedenen Tools rekonstruiert wurden (Anforderung 4, Abschnitt 4.1). Dies entspricht einer konzeptionellen Erweiterung von VANESA und ANDSystem. Letztlich wird damit eine Gegenüberstellung von Protein-Komplexen möglich, die auf Basis der drei Netzwerke vorhergesagt wurden. In je mehr Netzwerken ein bestimmter Protein-Komplex schließlich identifiziert wird, desto plausibler ist seine Vorhersage.

6.2.1 Deduktiv unterstützte Netzwerkrekonstruktion

Die Rekonstruktion des MPDZ/MUPP1-Netzwerks orientiert sich an dem bewährten Verfahren in FraMeTexPP (Abschnitt 6.1). Gegenüber dem vorherigen Ansatz werden die finalen,

TM-Ressource	Verwendung	Interaktion mit *FraMeTex*
Enju (Abschnitt 3.1.3	Extraktion von PASs Sätzen	Webservice
ABNER (Abschnitt 3.1.1)	Identifikation von Proteinen	API
Wordnet (Abschnitt 3.3.4.1)	Interaktionen erkennen (Verben)	API

Tabelle 6.1: Reihenfolge der zur *Pathway Prediction* aus Medline-Abstracts genutzten TM-Ressourcen

technischen Konzepte in *FraMeTex* allerdings vollständig berücksichtigt. Die in FraMeTexPP aus mehreren TM-Ressourcen gebildete Analysepipeline wird darüber hinaus optimiert und erweitert. Außerdem wird die Rekonstruktion des MPDZ/MUPP1-Netzwerks mit der Deduktionskomponente zusätzlich unterstützt. In diesem Abschnitt wird damit der gesamte von *FraMeTex* gebotene *Pathway Prediction*-Prozess (Abbildung 4.1) zur Netzwerkrekonstruktion genutzt. In die Rekonstruktion fließen wiederum alle Medline-Daten ein, die anhand der Schlüsselwörter MUPP1 oder MPDZ in der Textdatenbank (Abschnitt 5.1.2) identifiziert wurden.

6.2.1.1 Extraktion der Protein-Interaktionen aus Medline-Abstracts

Zur Extraktion von Protein-Interaktionen werden drei verschiedene TM-Ressourcen genutzt. Sie analysieren die Abstracts derjenigen Medline-Einträge, die anhand der beiden Schlüsselwörter ausgewählt wurden. Reguläre Ausdrücke sorgen für eine weitergehende Filterung auf Satzebene[6]. Die Tabelle 6.1 zeigt von oben nach unten die Reihenfolge, in der die TM-Ressourcen die Daten analysieren.

Die heterogenen TM-Ressourcen werden über das TM-Gateway in den *Pathway Prediction*-Prozess von *FraMeTex* integriert. Dies erfordert eine Anpassung der Wissensextraktions- sowie aufbereitungsphase des Gateways an die spezifischen Eigenschaften der jeweiligen TM-Ressource (Abschnitt 5.2.1). Da die zu analysierende Datensätze in *FraMeTex*-Datasets repräsentiert werden (Abschnitt 4.2), können die in jedem Analyseschritt gewonnen Informationen zunächst in dessen flexiblen Meta-Daten abgelegt werden. Auf Basis aller gewonnenen Informationen ergeben sich am Ende der Analysepipeline schließlich potentielle Protein-Interaktionen. Jede Protein-Interaktion entspricht einer anfänglich von Enju extrahierten PASs, die anhand drei bestimmter Bedingungen gefiltert wurde:

1. Die PAS weist exakt zwei Argumente auf.
2. Die beiden Argumente entsprechen Proteinen.
3. Das Prädikat der PAS repräsentiert ein Verb.

Der erste Schritt des skizzierten Vorgangs wird von einem angepassten Datenadapter unterstützt, der ursprünglich für Medline konzipiert wurde (Abschnitt 5.1.1). Er verantwortet die Auflösung der von Enju in XML kodierten PASs (Abbildung 3.3). Um deren Argumente im

[6] Auswahl erfolgt anhand benutzerdefinierter Proteine (z.B. MPDZ/MUPP1).

Anschluss möglichst fehlerfrei als Proteine zu identifizieren, werden sie von ABNER sowohl anhand des NLPBA- als auch anhand des BioCreative-Korpus geprüft (Tabelle 3.1). Aus der gleichen Motivation heraus wird zum Erkennen der Verben auf Wordnet (Abschnitt 3.3.4.1) zurückgegriffen und nicht der von Enju bereits durchgeführten Worterkennung vertraut. Darüber hinaus kann mit der in Wordnet verfügbaren Taxonomie eine optionale Einschränkung auf bestimmte Verben und deren Synonyme erfolgen. Dadurch kann auf Verben fokussiert werden, die häufig Protein-Interaktionen in Textdaten ausdrücken und die Netzwerkrekonstruktion weiter unterstützt werden[7]. Zum Ende des Filterprozess werden alle resultierenden Protein-Interaktionen in Großbuchstaben überführt, auf Predicate-Objects abgebildet und in einem Knowledge-Dataset zusammengefasst (Abschnitt 5.2.2). Dem Systemkonzept folgend repräsentiert diese Datenstruktur alle aus einem Medline-Abstract gewonnenen Pathways und kann in der Deduktionskomponente gespeichert werden.

Die Abbildung 6.4 zeigt das aus dem beschriebenen Verfahren resultierende MPDZ/MUPP1-Netzwerk. An der Farbgebung ist deutlich zu erkennen, dass das Netzwerk keine unvollständigen bzw. fehlerhaft aus Textdaten extrahierten Pathways (Tabelle 5.1) umfasst. Die auf dem kombinierten Einsatz unterschiedlicher TM-Ressourcen beruhende Textanalyse schließt diese bereits nahezu vollständig aus.

6.2.1.2 Inferenzen bereiten rekonstruiertes Netzwerk auf

Das Ziel ist es, durch die Anwendung von Regeln die Aussagekraft des Netzwerks zu stärken und invalide Pathways möglichst einfach filtern zu können. Hierfür wird das aus Textdaten gewonnene Netzwerk zunächst in der Deduktionskomponente gespeichert (Abbildung 4.4). Beim Speichern werden die extrahierten Pathways auf graphbasierte Strukturen abgebildet und automatisch verknüpft. Erst dadurch entsteht aus den zuvor unabhängigen Pathways das in Abbildung 6.4 dargestellte Netzwerk (Abschnitt 5.2.3).

In dem rekonstruierten Netzwerk weisen einige Knoten eine große Ähnlichkeit auf. Beispielsweise wurden mit *INTERACT*, *INTERACTS* sowie *INTERACTED* drei unterschiedliche Schreibweisen aus den Textdaten extrahiert, die einen identischen Sachverhalt repräsentieren. Dies trifft analog auch auf andere Knoten zu (*CALLED*, *NAMED*). Unter Einsatz der Deduktionskomponente wurde versucht diese Mehrdeutigkeiten aufzulösen. Hierfür wurden geeignete Regeln definiert und auf das *owl:sameAs* Property zurückgegriffen (Abschnitt 2.2.2.2). Definition 6.2 zeigt exemplarisch eine Regel, die eine Äquivalenz zwischen zwei Knoten zum Ausdruck bringt:

$$METEX{:}BIND \xrightarrow{owl{:}sameAs} METEX{:}BINDS \tag{6.2}$$

Analog dazu wurden auch die in Tabelle 6.2 zusammengefassten Äquivalenzen als Regeln formuliert. Ihre Anwendung führte zu mehreren Veränderungen im Netzwerk, die Abbildung 6.5

[7] Relevante Verben werden in der Konfiguration des TM-Gateways (Modul) spezifiziert (Abschnitt 4.3.1).

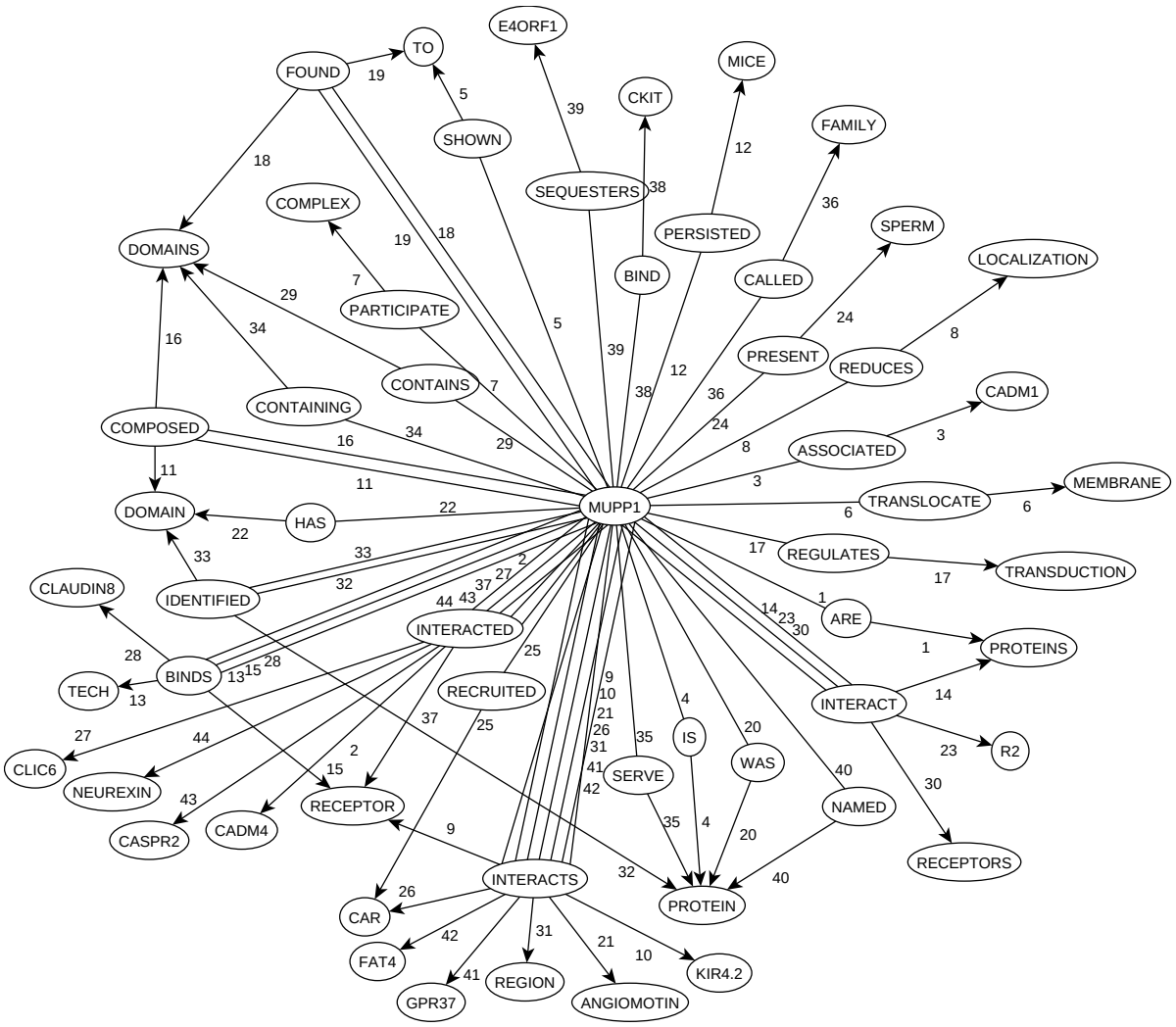


Abbildung 6.4: Rekonstruktion eines MPDZ/MUPP1-Netzwerks mit spezialisiertem TM-Verfahren

entnommen werden können. Die deduktiv beeinflussten Pathways sind grün gestrichelt dargestellt. Es ist gut zu erkennen, dass die extrahierten Protein-Interaktionen nun durch deutlich weniger Knoten (z.B. *INTERACT*) ausgedrückt werden (Abbildung 6.4). Gegenüber dem Ausgangsnetzwerk ergeben sich durch die Deduktion drei wesentliche Vorteile:

1. kompakteres Netzwerk mit identischen Informationsgehalt
2. unkomplizierter Ausschluss irrelevanter Pathways
3. erleichterte Exploration des Netzwerks

Die Filterung irrelevanter Pathways und die Exploration des Netzwerks hängen eng zusammen. Die zur Netzwerk-Exploration genutzten Suchmuster (Abschnitt 5.2.5) können sich auf die vereinheitlichten Interaktionen beziehen (Tabelle 6.2) und müssen nicht mehr jede individuelle Interaktion berücksichtigen. Damit können allein durch die Angabe eines einzelnen Syn-

Knoten	äquivalente Knoten
INTERACT	INTERACTS, INTERACTED
IS	WAS, ARE, HAS, CALLED, NAMED
CONTAIN	CONTAINS, CONTAINING
BIND	BINDS

Tabelle 6.2: Als Regeln formulierte Äquivalenzen für Deduktion im *MPDZ/MUPP1*-Netzwerk

onyms (z.B. *IS*) mehrere, wenig aussagekräftige Pathways ermittelt werden. Für die angestrebte Vorhersage der Protein-Komplexe ist dies sehr hilfreich, da sie dadurch in diesen nachgelagerten Prozess gar nicht erst einfließen. Damit kann die Anzahl potentiell falsch identifizierter Protein-Komplexe im Vornherein reduziert werden.

Sämtliche Regeln wurden in einer unabhängigen Ontologie (T-Box, Abbildung 5.12) zusammengefasst (Abschnitt 5.2.4). Damit können sie später auch zur Aufbereitung anderer Netzwerke genutzt werden. Auf eine explizite Verknüpfung der Regeln mit den Pathways des Netzwerks konnte hier jedoch verzichtet werden, da sich die Regeln unmittelbar auf die Knoten des rekonstruierten Netzwerks beziehen. Die Angabe eines *searchProperty* (Definition 5.6, Abbildung A.3) konnte damit entfallen.

6.2.1.3 Vergleich mit den Netzwerken aus ANDSystem und VANESA

Das von *FraMeTex* rekonstruierte MPDZ/MUPP1-Netzwerk wurde mit den evaluierten Netzwerken aus VANESA und ANDSystem verglichen (Abschnitt 6.2, Abbildungen A.6 und A.7). Die beiden Systeme verfolgen eine integrative (VANESA) sowie textminingbasierte (ANDSystem) Netzwerkvorhersage (Abschnitt 6.2). Der Vergleich berücksichtigt damit die beiden etablierten Rekonstruktionsverfahren der Bioinformatik (Abschnitt 3.2). Anhand der Ähnlichkeit der drei Netzwerke kann das von *FraMeTex* mittels Deduktionsunterstützung rekonstruierte Netzwerk beurteilt werden. Dies setzt jedoch eine Identifikation synonymer Proteine voraus, die mit DAWIS-M.D. (Abschnitt 3.6) semi-automatisch vollzogen wurde[8]. Aus der Menge aller ermittelten Synonyme fasst Tabelle A.1 diejenigen zusammen, die zumindest in einem der drei Netzwerke referenziert werden[9]. Mit Hilfe der Synonyme konnten zusätzliche Übereinstimmungen (*ANGIOMOTIN*, *CLAUDIN8*, *CKIT*) erkannt werden, die nicht offensichtlich sind.

Die Gemeinsamkeiten der drei Netzwerke zeigt Abbildung 6.6 anhand von Schnittmengen. Die ebenfalls ersichtlichen Differenzen sind auf die unterschiedlichen Rekonstruktionsverfahren zurückzuführen und lassen sich kaum vermeiden. Aus den integrierten sowie größtenteils validierten Daten in VANESA können sich durchaus Protein-Interaktionen ergeben, die in den

[8] http://tunicata.techfak.uni-bielefeld.de/webservice/db?wsdl

[9] Schreibweise des Synonyms stimmt mit der des entsprechenden Proteins (Knotens) im Netzwerk überein.

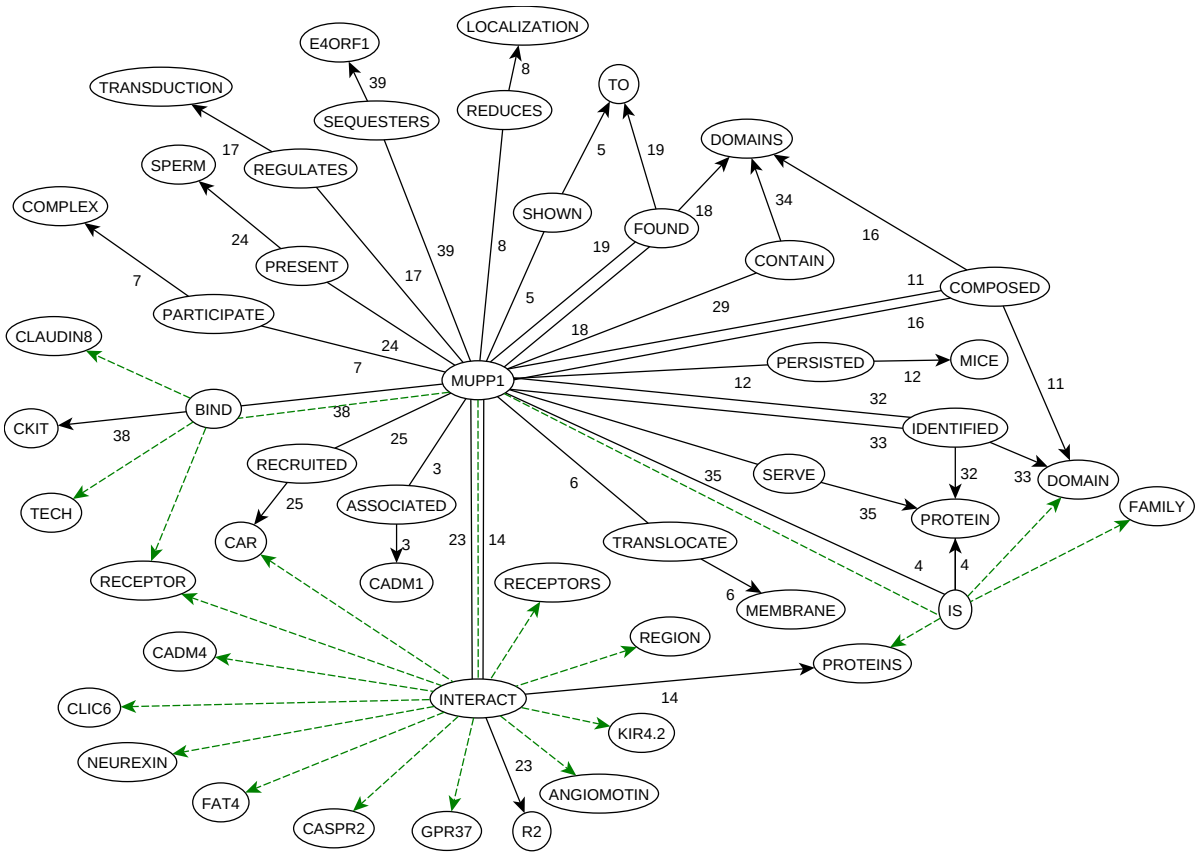


Abbildung 6.5: *MPDZ/MUPP1*-Netzwerk nach Inferenz auf Basis definierter Synonym-Äquivalenzen

textbasierten Medline-Abstracts gar nicht formuliert sind. Die entsprechenden Pathways können damit von ANDSystem und *FraMeTex* gar nicht rekonstruiert werden. Dies gilt umgekehrt ebenso. Außerdem basieren TM-Verfahren fast immer auf probabilistischen Algorithmen, die mit Unschärfen verbunden sind. Im direkten Vergleich der beiden textmining-basierten Ansätze (*FraMeTex* & ANDSystem) fällt allerdings der hohe Detailgrad und größere Umfang (16 vs. 9 Proteine) des mit *FraMeTex* rekonstruierten Netzwerks auf: Für zwei in Verbindung stehende Proteine ist stets ihre konkrete Interaktion bekannt (z.B. BIND). Dies lässt sich letztlich auf die konzipierte Analysepipeline in *FraMeTex* zurückführen (Tabelle 6.1), die ausschließlich aussagekräftige Beziehungen zwischen zwei Proteinen extrahiert. Allerdings muss bei der Beurteilung berücksichtigt werden, dass ANDSystem in der frei verfügbaren Version keine aktuellen Publikationen umfasst und auch nicht bekannt ist, welche konkreten Medline-Abstracts in die Rekonstruktion des MPDZ/MUPP1-Netzwerks eingeflossen sind.

Insgesamt weisen die drei Netzwerke jedoch auffällige Übereinstimmungen auf, so dass der von *FraMeTex* verfolgte Rekonstruktionsansatz in jedem Fall plausibel erscheint. In die angestrebte Vorhersage von Protein-Komplexen fließt dieses Netzwerk daher ebenfalls ein. Es bleibt jedoch abzuwarten, ob sich der höhere Detailgrad und größere Umfang des mit *FraMeTex* rekonstruierten MPDZ/MUPP1-Netzwerks auch auf die nachfolgende Vorhersage der Protein-Komplexe auswirkt.

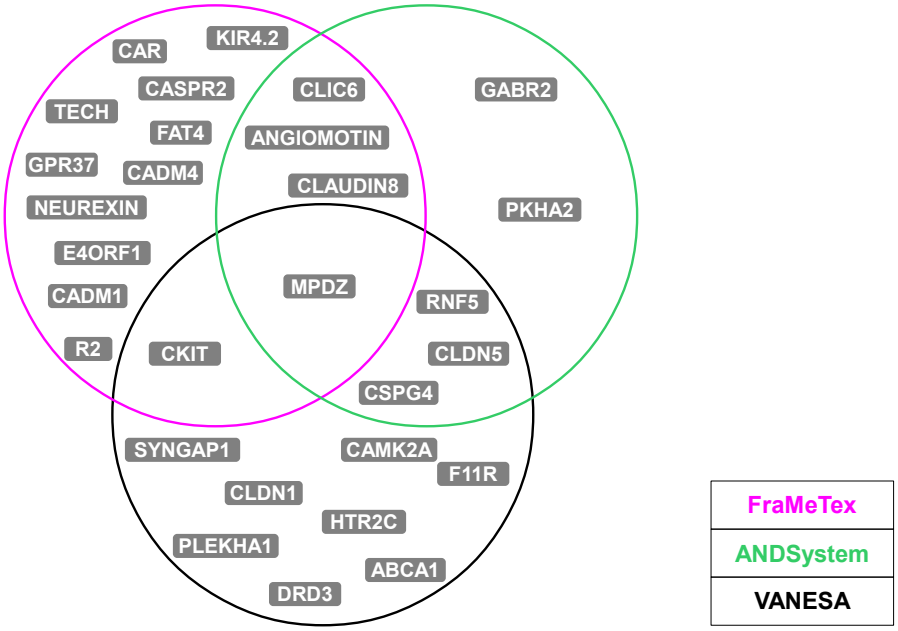


Abbildung 6.6: Gegenüberstellung rekonstruierter MPDZ/MUPP1-Netzwerke (Ebene 1)

6.2.2 Transitives Protein-Clustering

In diesem Abschnitt werden mit Hilfe der geschaffenen Deduktionskomponente Protein-Komplexe in den drei unterschiedlich rekonstruierten MPDZ-MUPP1-Netzwerken identifiziert (Abschnitt 6.2.1.3). Grundlage für die Vorhersage der Komplexe ist ein transitives Clustering-Verfahren [WRR+11] [NYP12]. Die zu verarbeitenden Netzwerke wurden zunächst erweitert, da aussagekräftige Vorhersagen sich über zwei Netzwerkebenen erstrecken sollten. Der Abbildung A.10 kann das mit *FraMeTex* erweiterte Netzwerk entnommen werden[10]. Der bereits für das Protein MPDZ/MUPP1 erprobte Rekonstruktionsansatz (Abschnitt 6.2.1) wurde für jedes interagierende Protein (Ebene 1) analog angewendet. Die Abbildungen A.8 und A.9 zeigen die mit VANESA und ANDSystem um eine Ebene erweiterten Netzwerke. Sie wurden manuell in Dateien serialisiert (Abbildung A.11), die pro Zeile eine Protein-Interaktion beschreiben. Die Dateien wurden von *FraMeTex* eingelesen und die jeweiligen Netzwerke dadurch in die Deduktionskomponente geladen[11]. Die von VANESA und ANDSystem rekonstruierten Netzwerke werden damit ebenfalls durch graphbasierte Strukturen repräsentiert (Abschnitt 5.2.3) und erlauben automatisierte Schlussfolgerungen (Abschnitt 5.2.4).

6.2.2.1 Deduktionsgestützte Netzwerk- und Pathway-Analyse

Das angewendete Clustering-Verfahren versucht in einem Protein-Interaktionsnetzwerk $G(V, E)$ (Abschnitt 2.1.4) jeweils aus drei Proteinen bestehende Komplexe zu identifizieren.

[10] Zur besseren Übersicht werden im Netzwerk nur noch die Proteine durch Knoten dargestellt.
[11] Jede eingelesene Protein-Interaktion wird als Predicate-Object repräsentiert (Abbildung 5.7).

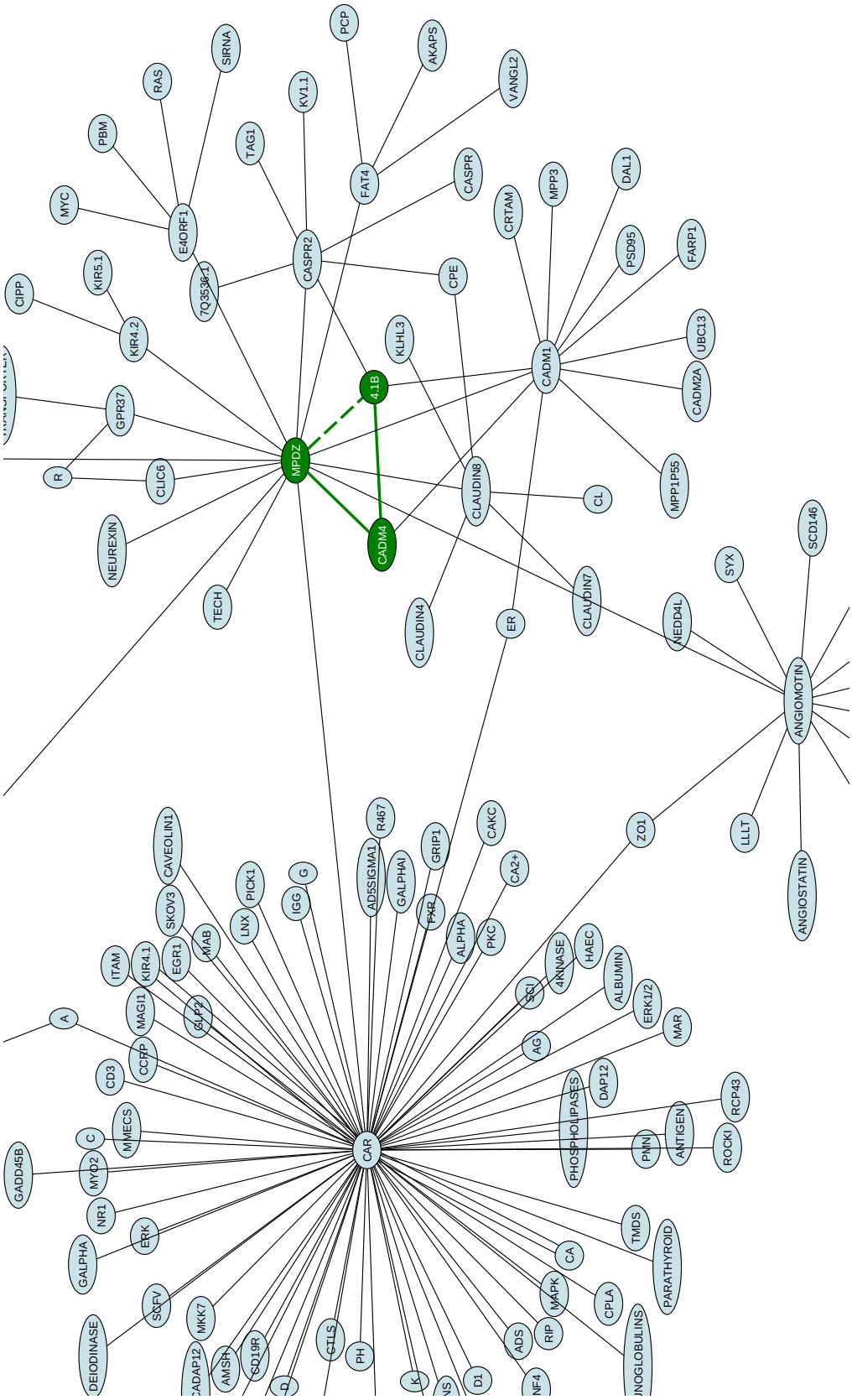


Abbildung 6.7: Ausschnitt des von *FraMeTex* rekonstruierten MPDZ/MUPP1-Netzwerks der Ebene 2 mit identifiziertem Protein-Komplex (grün)

```
SELECT distinct ?u ?v ?w
WHERE {
        ?u ?interaction1 ?v.     #Pathway Ebene 0->1
        ?v ?interaction2 ?w.     #Pathway Ebene 1->2
        FILTER (?u != ?w && ?u = METEX:MPDZ)
}
```

Abbildung 6.8: Transitives Protein-Clustering im Protein-Interaktionsnetzwerk mittels SPARQL

Drei Proteine (u, v, w) eines Netzwerks werden als Komplex erkannt, wenn gilt:

$$(u, v, w) \in V \wedge (u, v) \in E \wedge (v, w) \in E \tag{6.3}$$

Die Anwendung des *modus barbara* bringt damit eine transitive Beziehung zwischen den beiden Proteinen (u, w) zum Ausdruck (Abschnitt 2.2.3.2). Aus grafischer Sicht bilden sich im Netzwerk dadurch Cluster (Dreiecke), die sich durch die deduktiv gewonnenen Pathways ergeben. Dies ist in Abbildung 6.7 für die drei interagierenden Proteine *MPDZ* (u), *CADM4* (v) und *4.1B* (w) exemplarisch dargestellt.

In der Deduktionskomponente wurde die in Definition 6.3 präsentierte Regel als SPARQL-Query formuliert (Abschnitt 5.2.5). Sie erlaubt es, die auf Tripletts begrenzte Transitivität korrekt abzubilden (Abbildung 6.8). Das in Abschnitt 5.2.5 konzipierte Verfahren bietet diese Granularität nicht, so dass stets alle transitiven Beziehungen ausgewertet worden wären. Je nach Netzwerkgröße hätte dies zu beliebig langen Pathways führen können. Im Gegensatz dazu liefert die SPARQL-Query ausschließlich Tripletts, die durch die drei Variablen (?u, ?v, ?w) ausgedrückt werden. Die Variablen werden mit denjenigen Proteinen eines Netzwerks belegt, auf die alle in der WHERE-Klausel formulierten Bedingungen zutreffen. Die Bedingungen legen Pathway-Muster fest, die erfüllt sein müssen. Während das erste Muster eine beliebige Beziehung bzw. Interaktion (?*interaction1*) zwischen den beiden Proteinen u und v beschreibt, formuliert das zweite Muster dies analog für die Proteine v und w. Damit ist implizit auch eine transitive Beziehung zwischen den Proteinen u und w gefordert und die in Definition 6.3 abstrakt formulierte Regel umgesetzt. Mit der zusätzlichen FILTER-Anweisung werden potentielle Pathway-Zyklen im Netzwerk ignoriert. Außerdem werden in diesem konkreten Anwendungsfall nur vom Protein MPDZ ausgehende Komplexe berücksichtigt. Die Variable ?u ist hierfür entsprechend belegt worden[12].

Die Anwendung der Regel auf die drei verschiedenen MPDZ-Netzwerke führt damit in erster Linie zu Komplexen, die sich vom Root-Knoten (MPDZ) bis in die zweite Netzwerkebene erstrecken. Zusätzlich können Komplexe in der ersten Netzwerkebene erkannt werden, wenn zwei Proteine dieser Netzwerkebene interagieren. (z.B. MPDZ-CADM4-CADM1, Abbildung 6.7). Mit einer leicht modifizierten Regel können in größeren Netzwerken auch Protein-Kom-

[12] Protein ist um Namespace erweitert (Abschnitt 2.2.2.2).

Protein-Komplex (Triplett)	VANESA	ANDSystem	*FraMeTex*
MPDZ - AMOT - ZO1	✗	✓	✓
MPDZ - RNF5 - CFTR	✓	✓	✗
MPDZ - KIT - FYN	✓	✗	✓
MPDZ - KIT - MATK	✓	✗	✓
MPDZ - KIT - CRKL	✓	✗	✓
MPDZ - KIT - GRB2	✓	✗	✓
MPDZ - KIT - PTPN6	✓	✗	✓
MPDZ - KIT - KITLG	✓	✗	✓
MPDZ - KIT - SH2B3	✓	✗	✓
MPDZ - KIT - PTPRO	✓	✗	✓
MPDZ - KIT - STAP1	✓	✗	✓

Tabelle 6.3: Auflistung mehrfach vorhergesagter Protein-Komplexe

plexe in beliebigen Netzwerkebenen identifiziert werden. Hierfür genügt es bereits, die Restriktion des Startknotens (MPDZ) aufzuheben (Abbildung 6.8).

6.2.2.2 Gegenüberstellung identifizierter Protein-Komplexe

Die Ausführung der als SPARQL-Query formulierten Regel identifiziert unmittelbar alle aus drei Proteinen bestehenden Komplexe in einem Netzwerk. Für die analysierten Netzwerke (ANDSystem, VANESA, *FraMeTex*) können die ermittelten Komplexe (Tripletts) den Tabellen A.2, A.3 und A.4 entnommen werden. Unter Einbezug synonymer Proteinnamen (Tabelle A.1) wurden die aus den drei Netzwerken resultierenden Vorhersagen auf Übereinstimmungen untersucht. Da sich alle Vorhersagen vom Root-Knoten (MPDZ) über die erste Netzwerkebene erstrecken (Abschnitt 6.8), konnten die Proteine dieser Netzwerkebene als Filter herangezogen werden. Die Überschneidungen der ersten Netzwerkebene wurden bereits ermittelt (Abschnitt 6.2.1.3) und die Filterung ist damit problemlos möglich. Aufgrund der formulierten Query werden die entsprechenden Proteine in den vorhergesagten Komplexen stets an zweiter Stelle repräsentiert (Abbildung 6.8). Zwei aus unterschiedlichen Netzwerken resultierende Komplexe können damit nur dann identisch sein, wenn ihre Proteine an dieser Stelle identisch sind. Die zu vergleichenden Protein-Komplexe konnten dadurch erheblich reduziert werden.

Die zur Filterung erforderlichen Informationen können Abbildung 6.6 entnommen werden. An der Abbildung ist bereits erkennbar, dass kein Protein-Komplex in allen drei Netzwerken identifiziert werden kann. Hierfür müssten bereits die drei rekonstruierten Netzwerke erster Ebene mehr Überschneidungen aufweisen als den Root-Knoten MPDZ. Die entsprechende

Konstellation findet sich jedoch immer nur für zwei Netzwerke. Ein bestimmter Protein-Komplex kann damit maximal in zwei Netzwerken identifiziert werden. Protein-Komplexe auf die dies zutrifft, sind in der Tabelle 6.3 zusammengefasst. Aufgrund ihrer mehrfachen Vorhersagen erscheinen sie besonders plausibel und heben sich damit aus der Gesamtmenge aller Vorhersagen hervor. Dennoch bleibt ihre manuelle Validierung durch einen biomedizinischen Experten unumgänglich. Er kann auch unter den mehrfach identifizierten Protein-Komplexen durchaus noch fehlerhafte Vorhersagen erkennen. Umgekehrt ist es allerdings ebenso möglich, dass ein lediglich einmalig erkannter Komplex für die Biologie oder Medizin von besonderem Interesse ist.

6.3 Rekonstruktion eines metabolischen Netzwerks

Die bisher präsentierten Anwendungsfälle fokussierten auf Protein-Interaktionsnetzwerke. Der modulare *Pathway Prediction* Prozess von *FraMeTex* ist jedoch nicht auf diesen speziellen Typ biologischer Netzwerke fixiert (Anforderung 4). Zum Ende dieser Arbeit wurde daher versucht, ein metabolisches Netzwerk aus Textdaten zu rekonstruieren. Damit sollte das Deduktionspotential über Protein-Interaktionsnetzwerke hinaus erörtert werden. Die umfangreiche Recherche zu Beginn dieser Arbeit zeigte, dass bisher nur ein entsprechend spezialisierter TM-Algorithmus existiert (Abschnitt 3.1.4). Da der Algorithmus ausdrücklich noch ein *Proof of Concept* ist [CNSS12], hat dieser Anwendungsfall einen experimentellen Charakter. Aus diesem Grund wurde mit *FraMeTex* auch ein Netzwerk rekonstruiert, das bereits zur Evaluation des TM-Algorithmus diente. Die zu analysierenden Textdaten waren damit bekannt[13] und die Plausibilität des rekonstruierten Netzwerks gewährleistet. Der Fokus des Anwendungsfalls lag auf der Integration des TM-Algorithmus in den *Pathway Prediction* Prozess von *FraMeTex* sowie der logischen Auswertung der enzymatischen Reaktionen in der Deduktionskomponente. Das Vorgehen wird anhand eines überschaubaren Ausschnitts des rekonstruierten Netzwerks exemplarisch diskutiert.

Im Rahmen der Evaluation des genutzten TM-Algorithmus wurden drei verschiedene, metabolische Pathways für das Bakterium *E. coli K-12 substr. MG1655* aus Textdaten rekonstruiert [CNSS12]. Von ihnen wurde der *tetrahydrofolate biosynthesis* Pathway zur erneuten Rekonstruktion durch *FraMeTex* ausgewählt. Die hierfür zu analysierenden Textdaten erstreckten sich über dreizehn Medline-Abstracts. Sie wurden dem modularen *Pathway Prediction* Prozess von *FraMeTex* zugeführt, der sich somit hauptsächlich über die PPSs *ANALYZE* und *REASON* erstreckte (Abbildung 4.4). Über das konzipierte TM-Gateway (Abschnitt 5.2.1) konnte der Algorithmus unkompliziert in *FraMeTex* integriert werden[14]. Der Algorithmus entnimmt die zu analysierenden Textdaten einer Eingabedatei und schreibt die gewonnenen Pathways in eine Ausgabedatei. Dies erforderte eine entsprechende Anpassungen der Wissensextraktions- und Aufbereitungsphase des verantwortlichen TM-Gateways: Die zu analysierenden Textda-

[13] http://www.biomedcentral.com/content/supplementary/1471-2105-13-172-s1.zip

[14] Aufruf einer Stapeldatei, die auch alle unterstützenden Komponenten koordiniert (z.B. NER).

We describe here the following: the purification of the third protein, aminodeoxychorismate lyase; the isolation and partial characterization of the physiological intermediate in the conversion of chorismate and glutamine to PABA and pyruvate; and the cloning and overexpression of pabC, the gene encoding aminodeoxychorismate lyase.
Enzyme: pabC
Substrates: chorismate, glutamine
Products: PABA, pyruvate
Score: 6.700000000000001

Recently Nichols et al. (1989) showed that the conversion of chorismate to p-aminobenzoate occurs in two separate protein-catalyzed steps, not one, as previously thought.
Substrates: chorismate
Products: p-aminobenzoate
Score: 4.9

Abbildung 6.9: Mittels TM aus Medline-Abstracts gewonnene, enzymatische Reaktionen

ten eines *FraMeTex*-Datasets werden in einer Textdatei abgelegt und die extrahierten Pathways anschließend aus einer vom TM-Algorithmus geschriebenen Textdatei gelesen.

Die Abbildung 6.9 zeigt exemplarisch einige enzymatische Reaktionen, die aus dem Medline-Abstract 2071583 (PMID) extrahiert wurden. Es werden jeweils die in einem Satz identifizierten Substrate, Produkte und Enzyme aufgeführt. Der zusätzliche Score weist die Plausibilität der Analyseergebnisse aus. Die sich aus der Analyse ergebenden Pathways werden durch Predicate-Objects (Abbildung 5.7) repräsentiert. Dessen Subjekte repräsentieren jeweils ein Substrat, die Prädikate ein Enzym und die Objekte das aus der enzymatischen Reaktion hervorgehende Produkt. Ein Knowledge-Dataset fasst die aus einem Medline-Abstract extrahierten Pathways schließlich zusammen (Abschnitt 5.2.2).

Die Speicherung der einzelnen Pathways in der Deduktionskomponente führt automatisch zur Ausbildung des gesamten, rekonstruierten Netzwerks (Abschnitt 5.2.3). Voraussetzung ist lediglich, dass mehrere Pathways sich in ihren Substraten, Enzymen oder Produkten überschneiden. Dies geschieht beispielsweise, wenn das Produkt eines Pathways zugleich das Substrat eines anderen Pathways ist. Im rekonstruierten Netzwerk nimmt es damit die Rolle eines Metaboliten (Abschnitt 2.1.4.1) ein. Anhand des in Abbildung 6.10 gezeigten Ausschnitt des rekonstruierten Netzwerks kann dies nachvollzogen werden. Die Abbildung lässt außerdem erkennen, dass es dem TM-Algorithmus nicht immer gelingt valide Pathways aus den Textdaten zu extrahieren. Für den rot gefärbten Pathway (7) konnten offensichtlich nur zwei Komponenten einer enzymatischen Reaktion erkannt werden. Sie entsprechen mit großer Wahrscheinlichkeit dem Substrat sowie dem Produkt. Dies lässt sich aus ihrem Kontext erschließen, da sie von anderen Pathways entsprechend referenziert werden.

Damit die im Netzwerk dargestellten Zusammenhänge in der Deduktionskomponente auch logisch interpretiert werden können, muss die transitive Eigenschaft der enzymatischen Reaktionen zunächst als Regel formuliert werden. Erst dadurch wird eine Verkettung mehrerer

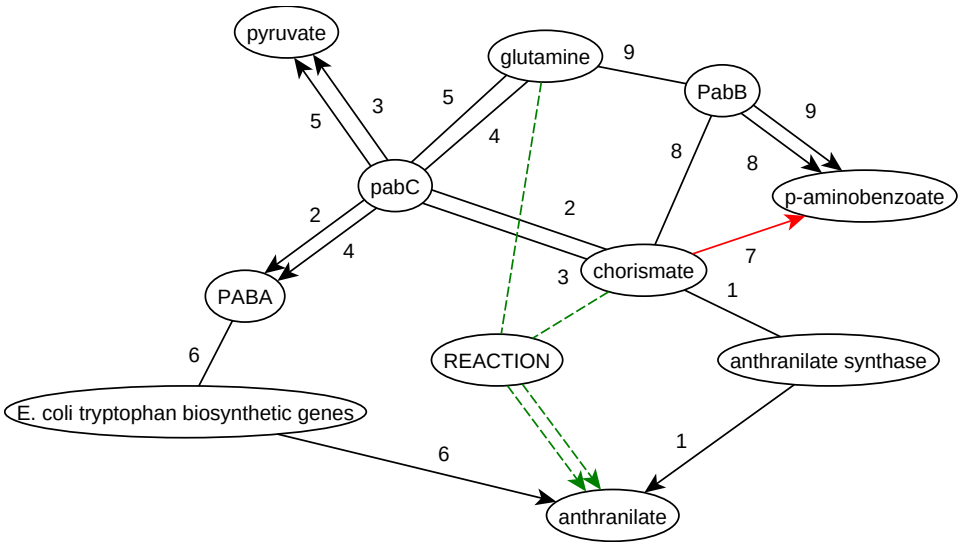


Abbildung 6.10: Ausschnitt des mit *FraMeTex* rekonstruierten *tetrahydrofolate biosynthesis* Pathway

Reaktionen möglich (*modus barbara*, Abschnitt 2.2.3.2), die ein Substrat über mehrere Metaboliten hinweg in ein Produkt überführen. Im Gegensatz zur Vorhersage der Protein-Komplexe in Abschnitt 6.2.2 ist die Transitivität hier damit uneingeschränkt anwendbar und muss nicht eingegrenzt werden. Das erforderliche Regelwerk kann damit nach dem konzipierten Verfahren in einer eigenständigen Ontologie formuliert werden (Abschnitt 5.12). Prinzipiell muss die Transitivität für jedes Enzym im Netzwerk definiert werden. In umfangreichen Netzwerken ist dies jedoch wenig praktikabel und erfordert zahlreiche Definitionen. Aus diesem Grund wurde die zuvor beschriebene Wissensaufbereitungsphase leicht modifiziert. Das Substrat und Produkt jeder enzymatischen Reaktion werden dadurch zusätzlich mit einer anonymen *REACTION*-Relation in der Datenbank verbunden[15]. Die Definition 6.4 stellt das Vorgehen für den Pathway 4 aus Abbildung 6.10 exemplarisch dar:

$$METEX{:}glutamine \xrightarrow{METEX{:}REACTION} METEX{:}PABA \tag{6.4}$$

Die Zuweisung der transitiven Eigenschaft kann sich damit auf die anonyme *REACTION*-Relation beschränken und es kann trotzdem jeder Pathway des Netzwerks logisch interpretiert werden:

$$METEX{:}REACTION \xrightarrow{rdf{:}type} owl{:}TransitiveProperty \tag{6.5}$$

Das Schlussfolgern in dem metabolischen Netzwerk basiert damit auf nur einer formulierten Regel (Definition 6.5). Entsprechend des verfolgten Konzepts wurde sie dennoch in einer eigenständigen Ontologie (T-Box) und damit unabhängig vom Netzwerk (A-Box) gespeichert (Abschnitt 5.2.4). Analog zur deduktiv unterstützten Aufbereitung des mit *FraMeTex* rekonstruierten MPDZ/MUPP1-Netzwerks (Abschnitt 6.2.1) konnte eine explizite Verknüpfung mittels *searchProperty* (Definition 5.6) auch hier entfallen, da sich die Regel wiederum unmittelbar

[15] In Abbildung 6.10 wurde die *REACTION*-Beziehung nur für inferierte Pathways (grün) visualisiert.

auf die Daten des rekonstruierten Netzwerks bezog. Die Anwendung der Regel führte zur Vorhersage neuer Pathways und damit zu weiteren Verknüpfungen innerhalb des rekonstruierten Netzwerks. Sowohl die Pathways 2 und 6 als auch die Pathways 4 und 6 konnten über den potentiellen Metaboliten *PABA* verknüpft werden (6.10). Die beiden durch Inferenz neu gewonnenen Pathways sind durch gestrichelte, grüne Kanten dargestellt (Tabelle 5.1). Der zunächst unscheinbar erscheinende Erkenntnisgewinn kann für eine valide Netzwerkrekonstruktion entscheidend sein. Trotz unvollständig aus Textdaten extrahierter Reaktionen kann im Idealfall ein valides Netzwerk rekonstruiert werden. Wäre beispielsweise Pathway 1 nicht vom TM-Algorithmus extrahiert worden, hätte die Deduktion trotzdem einen Hinweis auf den Pathway vom Substrat *chorismate* zum Produkt *anthranilate* geliefert. Vor dem Hintergrund, dass der genutzte TM-Algorithmus die Extraktion valider Pathways nicht garantieren kann (Abbildungen 6.9 und 6.10), bietet die Deduktion in diesem Kontext einen erheblichen Mehrwert.

Die deduktiven Inferenzen wirkten sich auch unmittelbar auf die konzipierte Exploration des Netzwerks aus (Abschnitt 5.2.5), die zur Visualisierung des Netzwerks genutzt wurde. Die Suchanfrage mit einem benutzerdefinierten Keyword (Substrat) lieferte nicht mehr nur die unmittelbar damit in Zusammenhang stehenden Enzyme und Produkte, sondern umfasste automatisch auch indirekt referenzierte Produkte. Beispielsweise führte eine Suchanfrage mit dem Keyword *glutamine* (Substrat) zu den Produkten *pyruvate*, *PABA*, *p-aminobenzoate* und *anthranilate*. Ohne ein Schlussfolgern hätte die Ergebnismenge das Produkt *anthranilate* nicht umfasst.

6.4 Zusammenfassung

In diesem Kapitel wurde die Rekonstruktion verschiedener, biologischer Netzwerke aus Textdaten mit dem prototypisch implementierten System gezeigt. Die von *FraMeTex* realisierte Analysepipeline ermöglichte eine effiziente Filterung der jeweils zu analysierenden Medline-Abstracts. In Abhängigkeit des zu rekonstruierenden Netzwerks wurden die selektierten Daten mit unterschiedlich spezialisierten TM-Algorithmen analysiert. Die konzipierte Systemarchitektur unterstützte sowohl eine flexible Integration der heterogenen Ressourcen in den Analyseprozess als auch eine einheitliche Weiterverarbeitung der gewonnenen Pathways in der Deduktionskomponente.

Der zu Beginn im Abschnitt 6.1 präsentierte Anwendungsfall stellte zunächst die Flexibilität und Unabhängigkeit der realisierten PPSs in den Vordergrund. Anhand einer webbasierten Applikation, die sich der von *FraMeTex* gebotenen Funktionalitäten bedient und eine interaktive Netzwerkrekonstruktion ermöglicht, wurde dies unter Beweis gestellt. Die zentrale Komponente der Anwendung ist eine leistungsstarke Suchfunktion, die anhand benutzerdefinierte Proteine zutreffende Medline-Abstracts in der Textdatenbank identifizieren kann. Auf Basis der selektierten Einträge wurden anschließend Protein-Interaktionen vorhergesagt. Hierfür wurden zwei TM-Algorithmen geschickt kombiniert. Die erkannten Interaktionen wurden in einer dynamischen Graphstruktur webbasiert visualisiert. Die Knoten des Netzwerks konnten

für weitergehende Analysen ausgewählt und das Netzwerk dadurch schrittweise erweitert werden. Ausgehend von einem initialen Protein konnte damit in einem interaktiven Prozess ein komplexes Netzwerk rekonstruiert werden. Anhand des Proteins MPDZ/MUPP1 wurde das realisierte Vorgehen exemplarisch demonstriert.

Der im Abschnitt 6.2 diskutierte Anwendungsfall deckte die gesamte Analysepipeline von *FraMeTex* ab und nutzte die Deduktionskomponente zur Vorhersage von Protein-Komplexen in drei unterschiedlich rekonstruierten MPDZ/MUPP1-Netzwerken. Grundlage für die Vorhersage war ein transitives Clustering-Verfahren, dass als individuelle Regel formuliert wurde. Die Anwendung der Regel in Protein-Interaktionsnetzwerken führte schließlich zur Vorhersage potentieller Protein-Komplexe. Verarbeitet wurden Netzwerke, die in der Vergangenheit bereits mit ANDSystem und VANESA rekonstruiert wurden. Damit konnte zugleich gezeigt werden, dass die in *FraMeTex* konzipierte Deduktionskomponente beliebige Netzwerke verarbeiten kann. Zusätzlich wurde mit *FraMeTex* ein drittes MPDZ/MUPP1-Netzwerk aus Textdaten rekonstruiert und in die Vorhersage mit einbezogen. Die Rekonstruktion orientierte sich am Verfahren der webbasierten Applikation (Abschnitt 6.1). Das Verfahren wurde jedoch durch den Einsatz weiterer TM-Ressourcen verfeinert und führte zu einem präziseren und umfangreicheren Netzwerk. Die Deduktionskomponente wurde in diesem Kontext zur Aufbereitung des Netzwerks genutzt. Das aus Textdaten gewonnene Netzwerk konnte mittels Deduktion kondensiert werden ohne Informationen zu verlieren. Im direkten Vergleich der drei unterschiedlich rekonstruierten Netzwerke fiel der Detailgrad der mit *FraMeTex* aus Textdaten gewonnenen Pathways auf. Die nachfolgende Vorhersage der Protein-Komplexe bestärkte den positiven Eindruck. Das transitive Clustering-Verfahren konnte in dem von *FraMeTex* rekonstruierten Netzwerk mit Abstand die meisten Protein-Komplexe identifizieren. Hierbei zeigten sich zudem mehrere Überschneidungen mit Vorhersagen aus den anderen beiden Netzwerken. Sie unterstrichen die Plausibilität der betreffenden Vorhersagen.

Die experimentelle Rekonstruktion eines metabolischen Netzwerks aus Textdaten im Abschnitt 6.3 schloss dieses Kapitel ab. Die Rekonstruktion zielte darauf ab, die von *FraMeTex* gebotene Flexibilität ein letztes Mal zu demonstrieren. Gegenüber der Rekonstruktion von Protein-Interaktionsnetzwerken war dies eine besondere Herausforderung, da bislang nur ein entsprechend spezialisierter TM-Algorithmus existiert. Obwohl der lediglich als prototypische Implementierung verfügbare Algorithmus keine adäquaten Schnittstellen aufweist konnte er in die Analysepipeline von *FraMeTex* integriert werden. Damit stand ihr gesamtes Potential zur Rekonstruktion eines metabolischen Netzwerks zur Verfügung. Dieses wurde genutzt, um den Mehrwert der Deduktion anhand des *tetrahydrofolate biosynthesis* Pathways beurteilen zu können. Da der Pathway bereits in der Vergangenheit mit dem Algorithmus erfolgreich rekonstruiert wurde, konnte der Mehrwert der von *FraMeTex* zusätzlich gebotenen Deduktion unmittelbar beurteilt werden. Bereits mit einer einfachen Regel konnte die Transitivität der enzymatischen Reaktionen in metabolischen Netzwerken zum Ausdruck gebracht werden. Ein einfaches Beispiel verdeutlichte schließlich, dass es dadurch gelingen kann einen validen Pathway auch dann vorherzusagen, wenn der TM-Algorithmus keinen Hinweis auf ihn lieferte. Für die oftmals fehleranfällige Netzwerkrekonstruktion aus Textdaten ist dies ein entscheidender Vorteil.

7 Resümee und Ausblick

Im Kontext der *Pathway Prediction* kommt der Speicherung rekonstruierter Netzwerke bisher keine besondere Bedeutung zu. Die Motivation dieser Arbeit war es daher, den Einsatz einer deduktiven Datenbank zu prüfen. Es sollte die Frage beantwortet werden, ob die Anwendung ihres zusätzlichen Regelwissen positiven Einfluss auf die Netzwerkvorhersage hat. Der angestrebte *Pathway Prediction* Prozess wurde im Kapitel 1 schematisch dargestellt und mit *FraMeTex* prototypisch implementiert. Der Fokus lag auf der Rekonstruktion biologischer Netzwerke aus Textdaten. Die Systemkomponenten wurden jedoch flexibel konzipiert, so dass mit dem geschaffenen Prototyp auch Inferenzen in anderweitig rekonstruierten Netzwerke möglich sind.

Die zum Verständnis dieser Arbeit nützlichen biologischen und technischen Grundlagen wurden im Kapitel 2 gelegt. Die Bedeutung biologischer Netzwerke für die Lebenswissenschaften wurde in diesem Zusammenhang deutlich gemacht. Sie beschreiben intra- sowie extrazelluläre Prozesse und sind für das Verständnis eines Organismus elementar (Abschnitt 2.1.4). Die Bioinformatik bietet zwei unterschiedliche Verfahren, mit denen Netzwerke vorhergesagt werden können. Sie entnehmen die erforderlichen Informationen entweder integrierten Datenquellen oder unstrukturierten Textdaten. Die größte, textbasierte Datenquelle für wissenschaftliche Publikationen ist Medline. Sie ist für die Vorhersage biologischer Netzwerke aus Textdaten von elementarer Bedeutung (Abschnitt 2.1.5). Spezialisierte TM-Algorithmen analysieren diese Daten und versuchen ihnen Pathways zu entnehmen. Auf ihre technischen Grundzüge wurde daher ebenso eingegangen, wie auf die Repräsentation und Speicherung extrahierter Pathways und Netzwerke. Die Konzepte deduktiver Datenbanken wurden in diesem Zusammenhang ausführlich diskutiert. Es zeigte sich, dass mit XSB[1] aktuell nur ein derartiges Datenbanksystem bekannt ist (Abschnitt 2.2.3). Zusätzlich wurde daher auf die Semantic Web Technologie eingegangen, die ein ähnliches Leistungsspektrum abdeckt. Abschließend wurde die Filterung textbasierter Daten mit Lucene[2] sowie das service-orientierte Architektur-Paradigma erörtert. Beides war für die Entwicklung des Prototypen von Bedeutung.

Repräsentative Systeme und Algorithmen, die einen Bezug zu dieser Arbeit haben, wurden im Kapitel 3 vorgestellt. Im Abschnitt 3.1 wurde anhand ausgewählter Algorithmen zunächst auf die Bandbreite der in der Bioinformatik existierenden TM-Verfahren eingegangen (Tabelle 3.2). Es wurde deutlich, dass die zur Netzwerkrekonstruktion erforderlichen Informationen sich häufig erst durch den kombinierten Einsatz mehrerer Algorithmen ergeben. Sie können aufgrund ihrer individuellen Schnittstellen und Datenstrukturen jedoch nicht unvermittelt zur

[1] http://xsb.sourceforge.net
[2] http://lucene.apache.org

Pathway Prediction zusammengestellt werden. Für diese komplexe Aufgabe wurden in der Vergangenheit eigenständige Systeme entwickelt. Die im Rahmen einer ausführlichen Recherche identifizierten Anwendungen wurden im Abschnitt 3.2 vorgestellt. Es zeigte sich, dass sie mit Ausnahme von VANESA[3] und ANDSystem[4] zumeist sehr spezielle Netzwerke rekonstruieren (Tabelle 3.3). Außerdem finden sich Deduktionsansätze bisher ausschließlich im Kontext integrativer Anwendungen. Sie sind jedoch hoch spezialisiert (z.B. auf chemische Reaktionsmuster) und nicht flexibel genug, um die Rekonstruktion beliebiger, biologischer Netzwerke sinnvoll zu unterstützen. Auffällig war zudem, dass ANDSystem das einzige System ist, mit dem überhaupt Netzwerke aus Textdaten vorhergesagt werden können (Abschnitt 3.2.2). Sein TM-Prozess ist jedoch weder transparent noch flexibel konzipiert. Er kann daher nicht zur Rekonstruktion verschiedener, biologischer Netzwerke beeinflusst werden. Dennoch weist ANDSystem die größte Nähe zur in dieser Arbeit motivierten *Pathway Prediction* auf.

Der Implementierung einer deduktiv unterstützten *Pathway Prediction* ging im Kapitel 4 zunächst eine Anforderungsanalyse voraus. Der Tabelle 7.1 können die formulierten Anforderungen sowie ihre Realisierung entnommen werden. Eine Besonderheit war, dass auf keine Erfahrungen oder Funktionalitäten eines Vorgängersystems zurückgegriffen werden konnte. Es wurde daher geprüft, inwieweit verfügbare Systeme oder Algorithmen die prototypische Entwicklung unterstützen können. Die Idee war es, das Potential etablierter TM-Algorithmen in *FraMeTex* zu nutzen. Ihr kombinierter Einsatz sollte eine möglichst präzise Rekonstruktion verschiedener biologischer Netzwerke ermöglichen. Dieser Ansatz wird bisher von keinem existierenden System unterstützt und ist ein Alleinstellungsmerkmal. In Kombination mit einer deduktiven Datenbank führt dies zu einem neuartigem Systemkonzept. Als weitere Besonderheit sollte der Prototyp die Netzwerkvorhersage aus Textdaten als ganzheitlichen Prozess auffassen und damit auch unterstützende Funktionalitäten bieten. Zusätzlich sollten alle Funktionalitäten modular konzipiert und flexibel genutzt werden können. Dies sollte insbesondere die Möglichkeit bieten, die Deduktionskomponente auch in Verbindung mit anderen Tools der Netzwerkrekonstruktion nutzen zu können. Solange die definierten Schnittstellen berücksichtigt werden, können somit auch anderweitig rekonstruierte Netzwerke profitieren. Das zum Schlussfolgern erforderliche Regelwissen sollte sowohl existierenden Wissensressourcen entnommen als auch individuell formuliert werden können. Die Deduktion sollte dadurch ein möglichst breites Spektrum potentieller Anwendungsfälle unterstützen können.

Die motivierte *Pathway Prediction* wurde in modulare PPSs gegliedert (Abschnitt 4.3) und auf Basis einer SOA konzipiert. Damit können Mediziner wie Biologen, mit ihnen vertrauten Workflow-Tools, die unabhängigen PPSs für individuelle Netzwerkvorhersagen zusammenstellen. Eine Grundvoraussetzung für diesen modularen Ansatz war die Definition einer einheitlichen Datenstruktur, die den Informationsaustausch zwischen den beteiligten PPSs sicherstellte. Hierfür wurde das dynamisch anpassbare *FraMeTex*-Dataset geschaffen (Abschnitt 4.2). Es dient sowohl zur Repräsentation der zu analysierenden Textdaten als auch der rekonstruierten Netzwerke sowie sämtlicher Zwischenergebnisse. Die Schnittstellen aller PPSs wurden auf diese Datenstruktur ausgerichtet und legten den Grundstein für die weitere Konzeption der Syste-

[3] http://agbi.techfak.uni-bielefeld.de/vanesa
[4] http://www.pbiosoft.com/en/andsystem

	Anforderung	Umsetzung
1.	Deduktion unterstützt Netzwerkrekonstruktion	✓
2.	Kombination verschiedener TM-Ressourcen für präzise Netzwerke	✓
3.	Rekonstruktion verschiedener, biologischer Netzwerke	✓
4.	Inferenzen in anderweitig rekonstruierten Netzwerken	✓
5.	Berücksichtigung existierender Systeme und Algorithmen	✓
6.	Anwendung existierender sowie individueller Regeln	(✓) (nur individuelle angewendet)
7.	Ganzheitlicher Ansatz und bietet auch unterstützende Funktionalitäten	✓
8.	Technische Basis der Systemkomponenten vereinheitlicht Abläufe	✓
9.	Konfiguration der *Pathway Prediction* über Dateien	✓

Tabelle 7.1: Abgleich der formulierten Anforderungen mit dem implementierten Prototyp

marchitektur. Einheitliche Verarbeitungsabläufe wurden abstrakt implementiert und bildeten die Basis zum Aufbau der individuellen PPSs. Ihre gemeinsame Basis bietet die Möglichkeit die Verarbeitung über Konfigurationsdateien zu beeinflussen und die in einem *FraMeTex*-Dataset repräsentierten Daten zu speichern. Da sich in *FraMeTex* früh der Einsatz unterschiedlichster Persistenzlayer abzeichnete, wurde ein Algorithmus entworfen, der weitestgehend unabhängig von ihnen arbeitet (Abschnitt 4.4).

Die im Kapitel 5 diskutierte Realisierung des Prototyps begann mit der Implementierung eines Datenadapters für Medline. Der verantwortliche PPS hatte die Aufgabe, die in XML-Dateien repräsentierten Daten in äquivalente *FraMeTex*-Datasets zu transformieren (Abschnitt 5.1.1). Sie konnten in nachgelagerten PPSs weiter verarbeitet werden, um spezifische Netzwerke aus ihnen zu rekonstruieren. Eine Grundvoraussetzung war die Filterung der umfangreichen Daten anhand benutzerdefinierter Schlüsselwörter (Proteine, Gene, usw.). Der Einsatz einer spezialisierten Lösung wie PubMed[5] kam nicht in Frage, da *FraMeTex* nicht auf die Analyse von Medline-Daten beschränkt sein sollte. Stattdessen wurde eine unabhängige Textdatenbank implementiert, die sämtliche Datensätze initial indiziert und effizient speichert (Abschnitt 5.1.2). Die Realisierung erfolgte in zwei unabhängigen PPSs, die zur Aufbereitung der adaptierten Daten kombiniert werden. Konzeptionell basieren die beiden PPSs auf dem abstrakten Persistenzalgorithmus, der eine zielstrebige Implementierung ermöglichte.

Der Analyse selektierter Textdaten mit frei wählbaren TM-Ressourcen wurde besondere Aufmerksamkeit geschenkt. Es wurde ein PPS geschaffen, der als Gateway fungiert und ausgewähl-

[5] http://www.ncbi.nlm.nih.gov/pubmed

te TM-Ressourcen in den *Pathway Prediction* Prozess integriert. Dadurch wird sichergestellt, dass unterschiedlichste Netzwerke vorhergesagt werden können und stets in einem *FraMeTex*-Dataset repräsentiert werden. Ihre Speicherung in der ebenfalls als PPS konzipierten Deduktionskomponente wird dadurch erst möglich. Analog zur Textdatenbank profitierte auch ihre Implementierung von dem konzipierten Persistenzalgorithmus. Technisch basiert sie auf den Konzepten des Semantic Web, deren graphbasierte Strukturen dem Einsatz des deduktiven Datenbanksystems XSB vorgezogen wurden (Abschnitt 5.2.3). Rekonstruierte Netzwerke konnten dadurch nahezu unverändert gespeichert werden und eine aufwändige Transformation entfallen (Abschnitt 5.2.3). Sobald erstmals lesend auf die Netzwerke zugegriffen wird, werden die in der Deduktionskomponente hinterlegten Regeln automatisch angewendet (Abschnitt 5.2.5). Eine Neuberechnung der Inferenzen erfolgt erst, wenn sich Änderungen an den gespeicherten Netzwerken ergeben. Je nach Komplexität der formulierten Regeln konnte der Aufwand dadurch erheblich reduziert und eine schnelle Bereitstellung der Ergebnisse garantiert werden. Der Zugriff auf die gespeicherten Netzwerke erfolgt anhand von Schlagwörtern. Eine Suche identifiziert zutreffende Pathways in den Netzwerken und visualisiert sie in einer statischen Netzstruktur. Dies ermöglichte eine Gegenüberstellung vorhergesagter Netzwerke, um die Auswirkungen der Deduktion auf die Rekonstruktion besser beurteilen zu können.

Das Potential des prototypisch implementierten Systems wurde zum Abschluss dieser Arbeit im Kapitel 6 erörtert. Im Abschnitt 6.1 wurde zunächst die Flexibilität der geschaffenen PPSs herausgestellt. Eine Web-Anwendung basiert auf diesen modularen Funktionalitäten und nutzt sie zur interaktiven Rekonstruktion biologischer Netzwerke. Verschiedene TM-Algorithmen wurden hierfür erfolgreich kombiniert. Die Applikation war das Ergebnis einer Masterarbeit in der AGBI und wurde im Rahmen verschiedener, studentischer Projekte stetig weiter ausgebaut. Die rekonstruierten Netzwerke werden von der Anwendung jedoch nicht gespeichert und das von *FraMeTex* gebotene Deduktionspotential damit noch ungenutzt gelassen. Auf den Einsatz von Deduktionsverfahren wurde daher im Abschnitt 6.2 ausführlich eingegangen. Sie dienten zur Vorhersage von Protein-Komplexen, die als Auslöser von Krankheiten gelten und daher von Interesse sind. Der Anwendungsfall konzentrierte sich auf das Protein MPDZ/MUPP1, dass mit Herz- und Gefäßerkrankungen in Verbindung gebracht wird. Ausgangspunkt waren zwei bereits in der Vergangenheit mit ANDSystem und VANESA rekonstruierte MPDZ/MUPP1-Netzwerke. Zusätzlich wurde mit *FraMeTex* ein drittes MPDZ/MUPP1-Netzwerk deduktionsunterstützt rekonstruiert und in die Vorhersage der Protein-Komplexe einbezogen. Die Rekonstruktion bediente sich einer verfeinerten Analysepipeline der Web-Anwendung und wendete zusätzlich Regeln auf die vorhergesagten Pathways an. Dies resultierte in einem aufbereiteten Netzwerk, dessen Detailgrad und Umfang sich deutlich von den mit ANDSystem und VANESA rekonstruierten unterschied. Die konsequente Realisierung unabhängiger PPSs mit einheitlichen Schnittstellen in *FraMeTex* ermöglichte schließlich eine problemlose Weiterverarbeitung der drei MPDZ/MUPP1-Netzwerke in der Deduktionskomponente. Damit konnte ein identisches Deduktionsverfahren auf unterschiedlich rekonstruierte MPDZ/MUPP1-Netzwerke angewendet und die Ergebnisse anschließend gegenübergestellt werden: Eine individuell formulierte Regel identifizierte in jedem der Netzwerke mehr als einhundert potentielle Protein-Komplexe. Komplexe, die in mehr als einem Netzwerk vorhersagt wurden, erschienen besonders plausibel und wurden daher hervorgehoben.

Abschließend wurde mit der experimentellen Rekonstruktion eines metabolischen Netzwerks aus Textdaten versucht das Potential der Deduktionskomponente weiter zu sondieren (Abschnitt 6.3). Die aus Medline-Abstracts gewonnenen, enzymatischen Reaktionen wurden gespeichert und logisch interpretiert. Es konnte gezeigt werden, dass hierfür die Formulierung einer einfachen Regel genügte. Ihre Anwendung provozierte bereits die Vorhersage mehrerer Pathways. Theoretisch können diese deduktiv gewonnenen Pathways Unzulänglichkeiten des genutzten TM-Algorithmus kompensieren. Das Potential einfacher Regeln überraschte damit in allen Anwendungsfällen. Entgegen der Erwartungen war für eine sinnvolle Unterstützung der *Pathway Prediction* aus Textdaten kein umfangreiches Regelwerk erforderlich. Rückblickend hätte die konzipierte Verknüpfung rekonstruierter Netzwerke mit Regelwissen daher einfacher gestaltet werden können (Abschnitt 5.2.4).

Auf Basis der erzielten Ergebnisse lässt sich die zu Beginn dieser Arbeit im Abschnitt 1.2 aufgeworfene Frage eindeutig beantworten: Mit einer deduktiv motivierten Datenbank lässt sich die Netzwerkrekonstruktion auf vielfältige Weise unterstützen. Mit der Vorhersage von Protein-Komplexen ließ sich sogar zeigen, dass ihr Potential über die eigentliche Rekonstruktion hinaus reichen kann.

Ausblick

Zum Ende dieser Arbeit sollen einige Entwicklungspotentiale aufgezeigt werden. Sie zielen darauf ab, den prototypisch implementierten *Pathway Prediction* Prozess zu verfeinern und die Qualität der vorhergesagten Netzwerke weiter zu steigern. In einem ersten Schritt soll das SOA-Konzept vollständig umgesetzt werden. Die Schnittstellen der modularen PPSs müssen hierfür noch als Webservices definiert werden. Erst danach können Biologen und Mediziner die von *FraMeTex* gebotene Netzwerkvorhersage in den ihnen bekannten Workflow-Tools problemlos nutzen. Mit dem etablierten Spring[6] Framework reduzieren sich die erforderlichen Anpassungen jedoch weitestgehend auf Annotationen im bestehenden Programmcode. Im Anschluss ist der Ausbau der zentralen Textdatenbank geplant. Die momentan genutzte MySQL-Datenbank soll durch eine *Not only SQL (NoSQL)*-Variante ergänzt werden. Dies soll die Datenzugriffe erheblich beschleunigen, da NoSQL-Datenbanken auf die Verarbeitung von Massendaten spezialisiert sind. Angedacht ist zunächst allerdings nur die Migration der Tabelle mit den umfangreichsten Daten. Die zukünftige Textdatenbank verfolgt damit einen hybriden Ansatz und kombiniert die Vorteile beider Datenbankkonzepte geschickt [NAK14]. Außerdem soll die von ihr gebotene Filterung der Textdaten zukünftig auch Spezies berücksichtigen. Damit soll verhindert werden, dass in die Rekonstruktion eines Netzwerks Daten verschiedener Spezies einfließen und die Ergebnisse verfälschen. Zur weiteren Präzisierung der rekonstruierten Netzwerke sollen außerdem automatisch synonyme, biologische Objekte (z.B. Proteine) erkannt und aufgelöst werden. Zum Aufbau der erweiterten Filterung soll der Einsatz von Apache Solr[7] geprüft

[6] http://spring.io/
[7] http://lucene.apache.org/solr/

werden. Solr basiert auf dem bisher genutzten Lucene[8] und bietet insbesondere für Volltextsuchen bereits zahlreiche, spezialisierte Erweiterungen.

Ein weiterer Aspekt ist eine verbesserte Darstellung der rekonstruierten Netzwerke. Das Ziel ist es, die statische Visualisierung aufzugeben und stattdessen das Potential von VANESA zu nutzen. Hierfür muss ein Export der rekonstruierten Netzwerke geschaffen werden, der sie in dem standardisierten SBML-Format beschreibt. Dieses kann von VANESA eingelesen werden und damit eine Visualisierung in der Applikation erfolgen. Inwieweit die von *FraMeTex* gebotenen Deduktionsverfahren selbst in VANESA integriert werden (können), muss in diesem Kontext zunächst evaluiert werden. Eine weitere Vision ist, einzelne Pathways zusätzlich mit Wahrscheinlichkeiten zu belegen. Sie sollen die Plausibilität eines Pathways zum Ausdruck bringen und die Exploration der Netzwerke damit weiter verbessern. Das Konzept sieht vor, die Wahrscheinlichkeit in Abhängigkeit der TM-Analyse zu setzen. Je häufiger ein Pathway aus Textdaten extrahiert wurde, desto größer ist die für ihn hinterlegte Wahrscheinlichkeit. In die Vorhersage spezifischer Netze könnten dann nur noch Pathways mit entsprechend hohen Wahrscheinlichkeiten einfließen. Der von *FraMeTex* verfolgte Rekonstruktionsansatz würde damit ein weiteres Alleinstellungsmerkmal gegenüber anderen Tools aufweisen. Die Möglichkeit die Semantic Web Technologie um Wahrscheinlichkeiten zu erweitern wurde bereits erörtert [LS08]. Das skizzierte Verfahren müsste jedoch zunächst evaluiert und sein Potential für die Vorhersage biologischer Netzwerke bewertet werden.

Zukünftig wird sich das Einsatzspektrum von *FraMeTex* auch noch erweitern. Neben der Vorhersage biologischer Netzwerke soll es auch zur Lokalisation (sub)zellulärer Komponenten dienen. Ein studentisches Projekt setzt *FraMeTex* aktuell ein, um mit Hilfe verschiedener TM-Algorithmen entsprechende Informationen aus Medline-Abstracts zu gewinnen. Die vom System gebotene Flexibilität zahlt sich hier erneut aus. Die bisher erzielten Ergebnisse sind vielversprechend und offenbaren ein großes Potential. Das System stützt sich auf die modularen PPSs und bietet momentan eine einfache GUI, mit der potentielle Lokalisationen in einem semiautomatischen Prozess ermittelt werden können. Das System befindet sich allerdings noch in der Entwicklung und kann daher hier noch nicht im Detail vorgestellt werden. Prinzipiell ist mit *FraMeTex* sogar die Verarbeitung beliebiger Daten denkbar. Auch wenn es für die *Pathway Prediction* aus Textdaten konzipiert wurde, kann es dank seiner Flexibilität problemlos an andere Domänen angepasst werden. Der Einsatz im Kontext der Lokalisation zeigt dies bereits eindrucksvoll. Entscheidend ist die Verfügbarkeit entsprechender TM-Algorithmen sowie die Formulierung adäquater Regeln. Die generellen Abläufe unterscheiden sich dabei nicht von den in dieser Arbeit präsentierten.

[8] http://lucene.apache.org/

Danksagung

Die vorliegende Arbeit wurde von mir berufsbegleitend, als externes Mitglied in der AGBI angefertigt. Diese Möglichkeit wurde mir von Prof. Dr. Hofestädt geboten. Besonderer Dank gilt an dieser Stelle seinen langjährigen Mitarbeitern Dr. Benjamin Kormeier sowie Dr. Björn Sommer. Sie standen jederzeit als kompetente Ansprechpartner zur Verfügung und gaben viele, wertvolle Impulse. Sie ermutigten mich darüber hinaus immer wieder die Arbeit weiter voranzutreiben. Danken möchte ich auch Pascal Witthus, der im Rahmen seiner Masterarbeit die von mir entwickelten Komponenten erstmals produktiv einsetzte. Er legte damit den Grundstein für weitere, studentische Projekte. Ich möchte außerdem das konstruktive Feedback von Prof. Dr. Jan Baumbach hervorheben, auf das ich stets zählen konnte. Danken möchte ich auch Prof. Dr. Philipp Cimiano, dessen Expertise zum Gelingen dieser Arbeit entscheidend beitrug.

Ebenso großer Dank gilt der arvato Systems S4M GmbH innerhalb des Bertelsmann-Konzerns. Ohne die Unterstützung meines Arbeitgebers wäre diese Arbeit nicht möglich gewesen. Besonders hervorheben möchte ich an dieser Stelle drei Kollegen. In seiner Rolle als Teamleiter verstand es Stefan Moch die partnerschaftlichen Grundwerte des Unternehmens zu leben und Freiräume zu schaffen. Als Projektleiterin trug Ingrid Pape die Entscheidungen stets mit und eröffnete mir damit erst die Möglichkeit kontinuierliche Fortschritte an dieser Arbeit zu erzielen. Sie wurden während der gesamten Zeit von Michael Pott interessiert verfolgt, der immer an den Erfolg der Arbeit glaubte. Es tat gut, ihn als Mentor an meiner Seite zu wissen.

Bei meinen Freunden Stephan Graute, Dr. Christian Reicherts, Michael Dammann und Holger Dunker möchte ich mich für ihre moralische und fachliche Unterstützung bedanken. Stephan stand jederzeit für anregende Diskussionen zur Verfügung und half den Blick zu schärfen. Als Internist und Hämatologe war Christian immer bereit, die Plausibilität einzelner Ergebnisse zu evaluieren. Seine Anregungen flossen an verschiedenen Stellen in die Arbeit ein. Über Michael kam ich mit PD Dr. Sven Groppe in Kontakt, der ebenfalls Interesse an dieser Arbeit zeigte. Abseits der fachlichen Thematik führte Holgers strategischer Weitblick zum zielgerichteten Abschluss der Arbeit.

Der größte Dank gilt meiner Familie. Sie mussten in den vergangenen Jahren vielfach zurückstecken und Verzicht üben. Trotzdem haben sie mich nach Kräften unterstützt und motiviert die Arbeit zum Abschluss zu bringen. Für ihre Motivation und ihren Rückhalt danke ich auch meinen Eltern Petra und Christoph Wallmeyer.

A Anhang

Nr.	synonyme Proteine in VANESA, ANDSystem & *FraMeTex*
1	CLDN5, CLD5
2	CLD8, CLAUDIN8
3	D, G, A, C, ALPHA
4	DLG4, PSD95
5	PDIA3, P58
6	TJP1, ZO1
7	PARD3, ASIP
8	KIT, C-KIT
9	ANGIOMOTIN, AMOT
10	DAL1, 4.1B
11	SDHB, IP
12	PRKCA, PRKACA
13	CHK, MATK
14	INADL, CIPP
15	LNK, SH2B3
16	NUDT6, FGF-2
17	PTPN6, SHP1, HCP, SHP-1
18	GRIN1, NR1
19	SYNGAP1, RASA1
20	MPP3, DLG3
21	SCF, KITLG

Tabelle A.1: Mit DAWIS-M.D. in Netzwerken identifizierte Protein-Synonyme (Ebene 1 & 2)

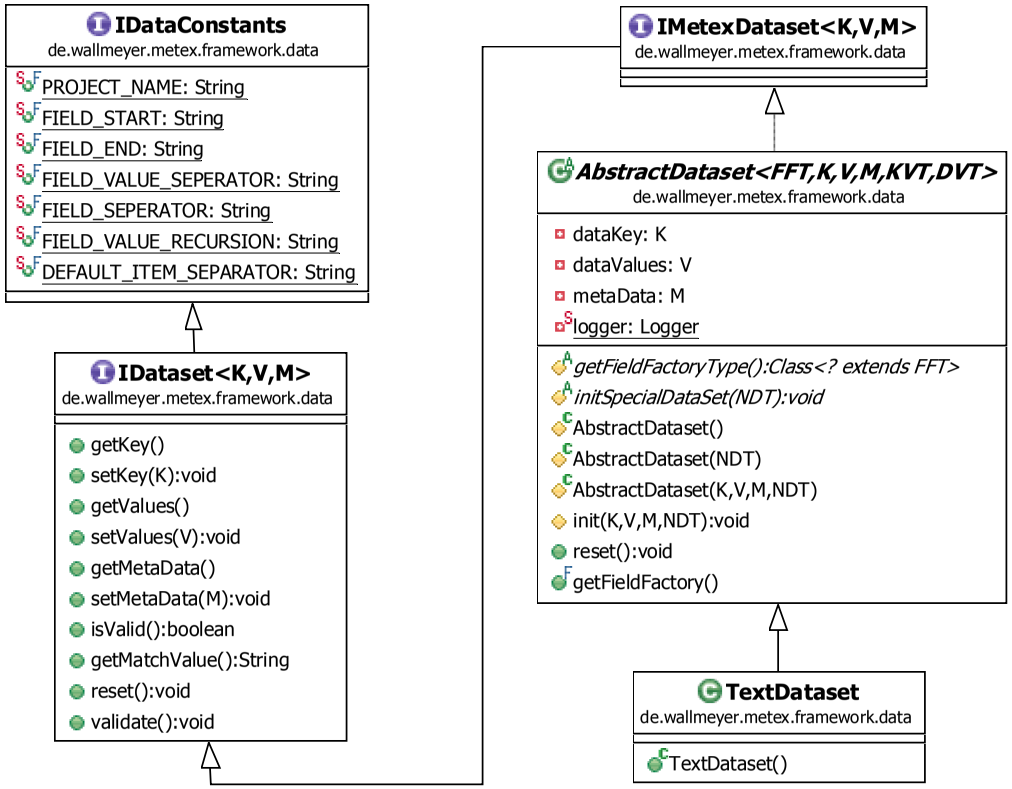


Abbildung A.1: Generisch konzipierte Komponenten (K,V,M) des *FraMeTex*-Datasets.

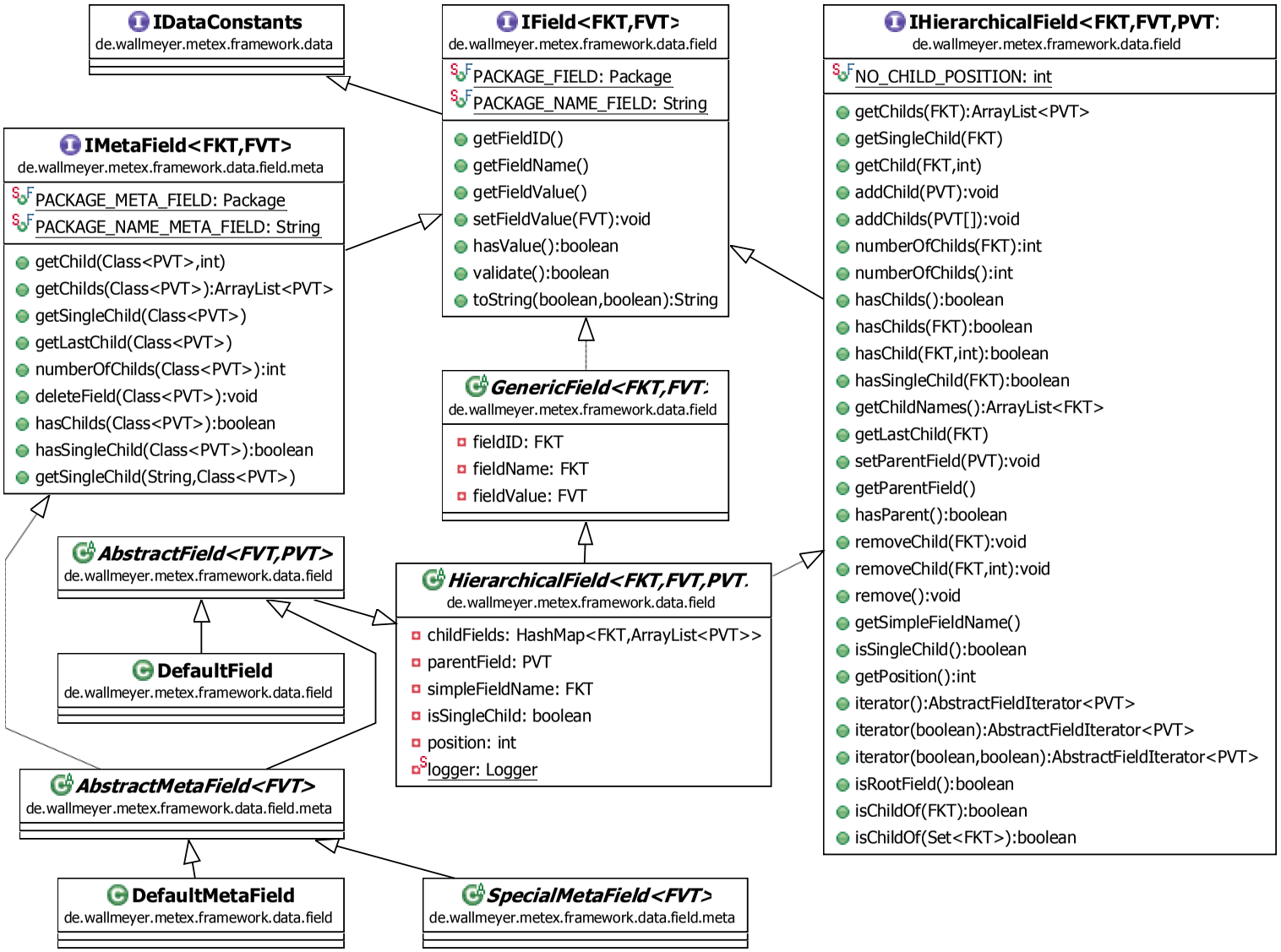


Abbildung A.2: Hierarchisch konzipierte Feldstruktur innerhalb des *FraMeTex*-Datasets.

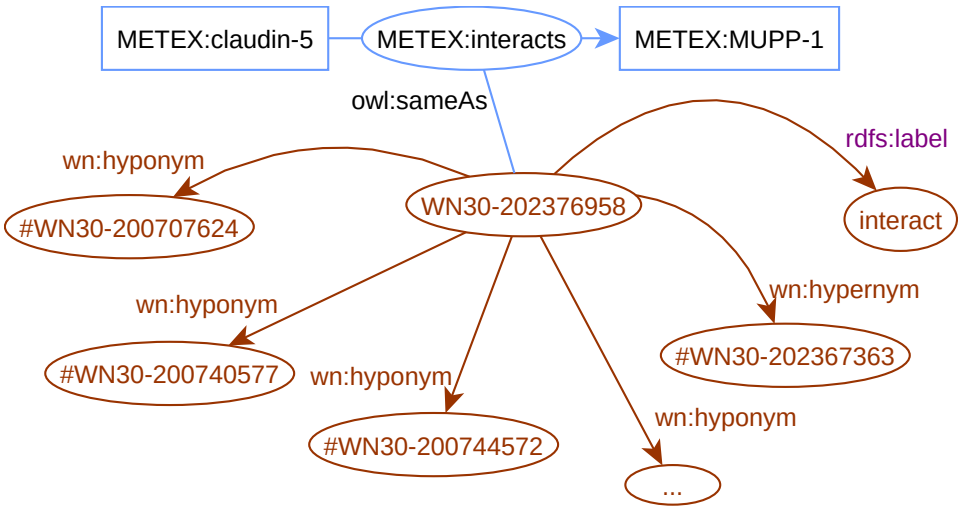


Abbildung A.3: Exemplarische Verknüpfung eines Pathways mit Wordnet-Ontologie (Abschnitt 3.3.4.1)

```
#basic configuration for first (0) ontology
config.knowledge.facade.ontology0.type=de.wallmeyer.metex.module.knowledge.ontology.JenaOntology

#enable ontology loading and name path for loading and writing
config.knowledge.facade.ontology0.doLoad=true
config.knowledge.facade.ontology0.source=C:/temp/wordnet/WordNet.owl
config.knowledge.facade.ontology0.path=

#define search property that is used to identify corresponding resources
config.knowledge.facade.ontology0.searchPropertyName=http://www.w3.org/2000/01/rdf-schema#label

#define logical properties for reasoner
config.knowledge.facade.ontology0.transitive=http://www.ontologyportal.org/WordNet.owl#hypernym
#reduce complexity by building up sub ontology
config.knowledge.facade.ontology0.sub.enable=true
config.knowledge.facade.ontology0.sub.isKeepFilter=true
#named properties (comma separated) are filtered, depending on isKeepFilter
#(keep or drop statements with named properties)
config.knowledge.facade.ontology0.sub.filterProperties=
http://www.ontologyportal.org/WordNet.owl#hypernym,
http://www.w3.org/2000/01/rdf-schema#label,
http://www.w3.org/2000/01/rdf-schema#comment

#true in case properties should be included in sub ontology (disabled by default)
config.knowledge.facade.ontology0.sub.withProperties=false
```

Abbildung A.4: Konfiguration des universellen Ontologie-Adapters für Wordnet

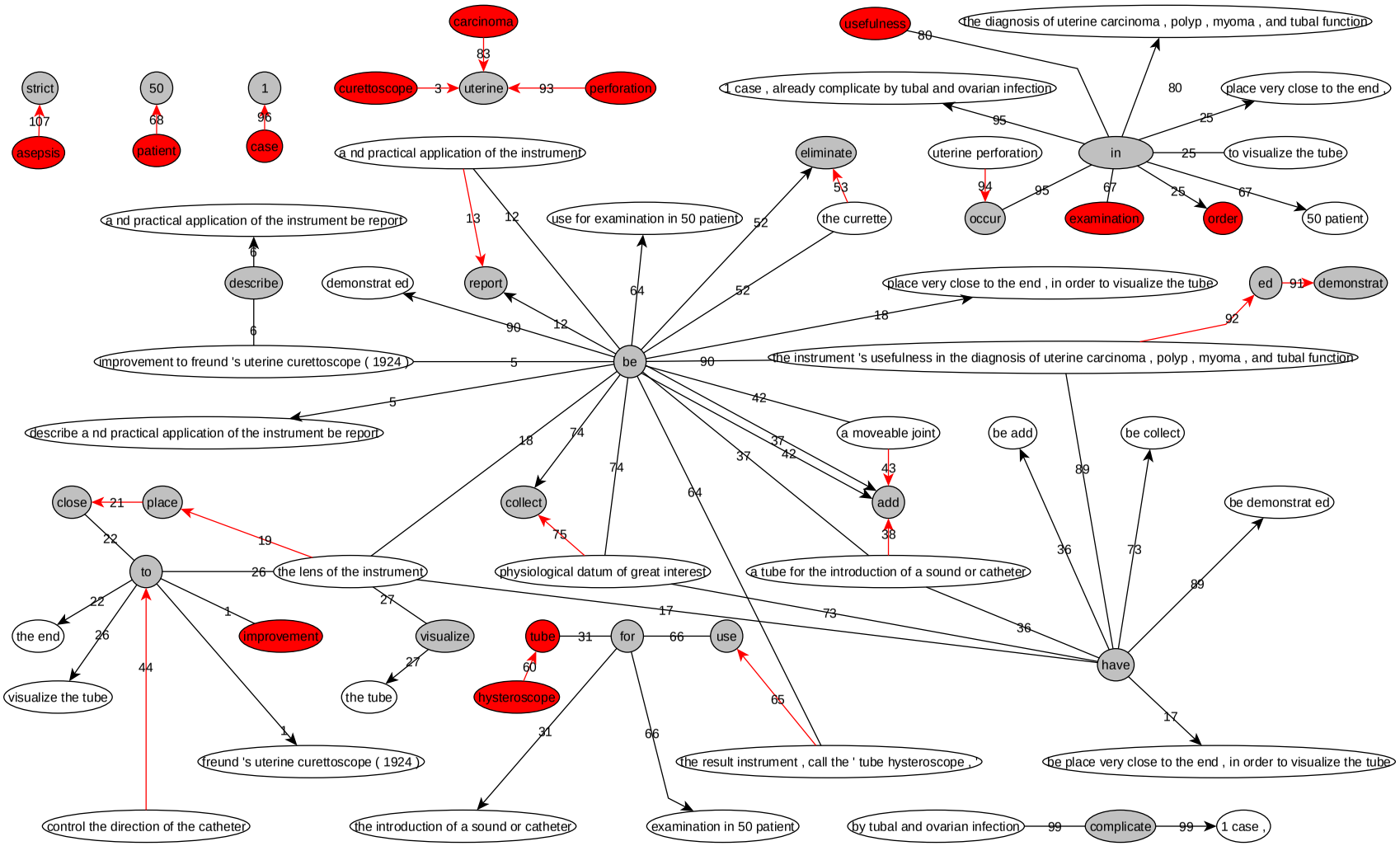


Abbildung A.5: Aus Medline-Abstract von *Enju* extrahierte PAS (Ausschnitt, PMID 12236137)

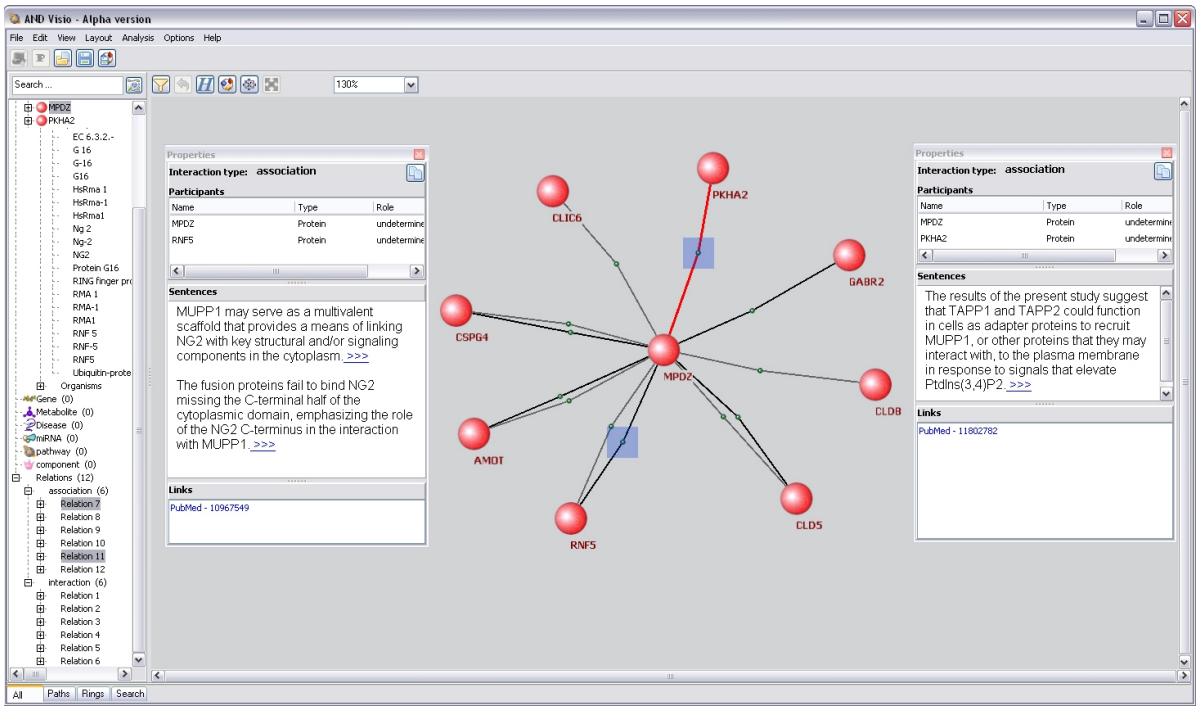


Abbildung A.6: Mit ANDSystem rekonstruiertes *MPDZ/MUPP1*-Netzwerk [STK+10]

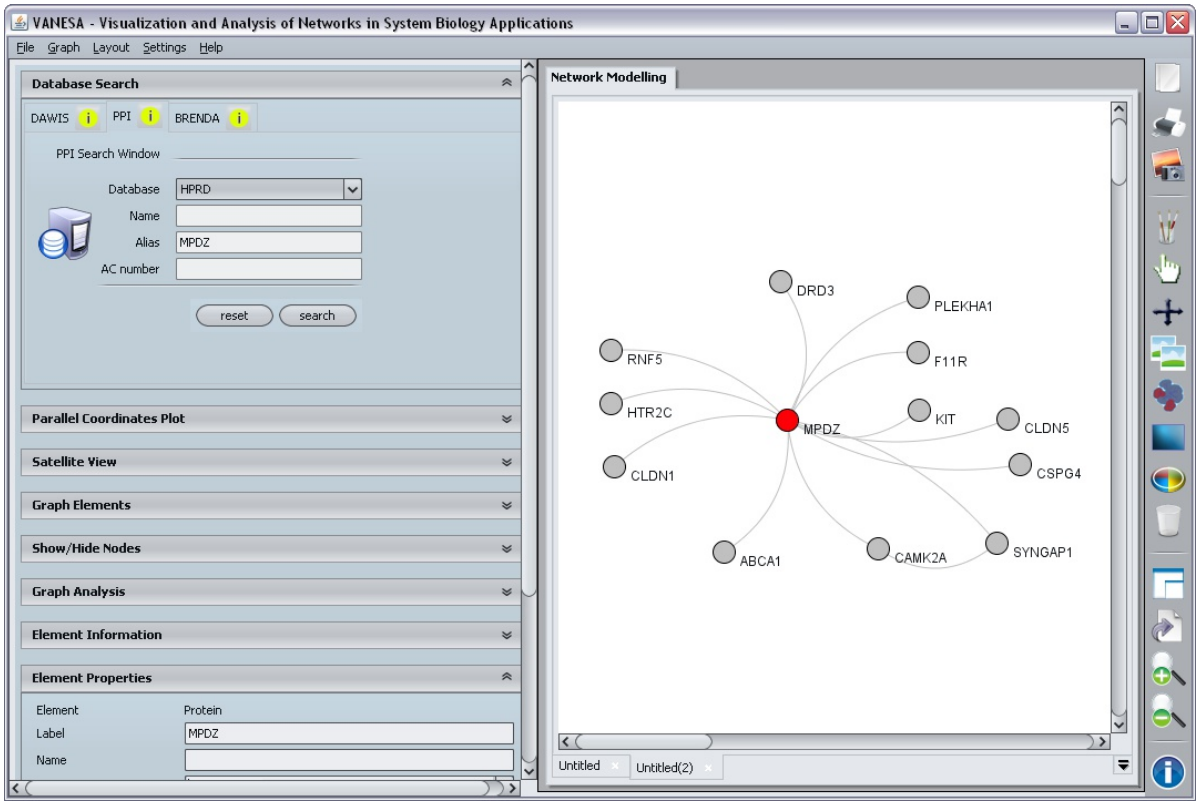


Abbildung A.7: Mit VANESA rekonstruiertes *MPDZ/MUPP1*-Netzwerk [STK+10]

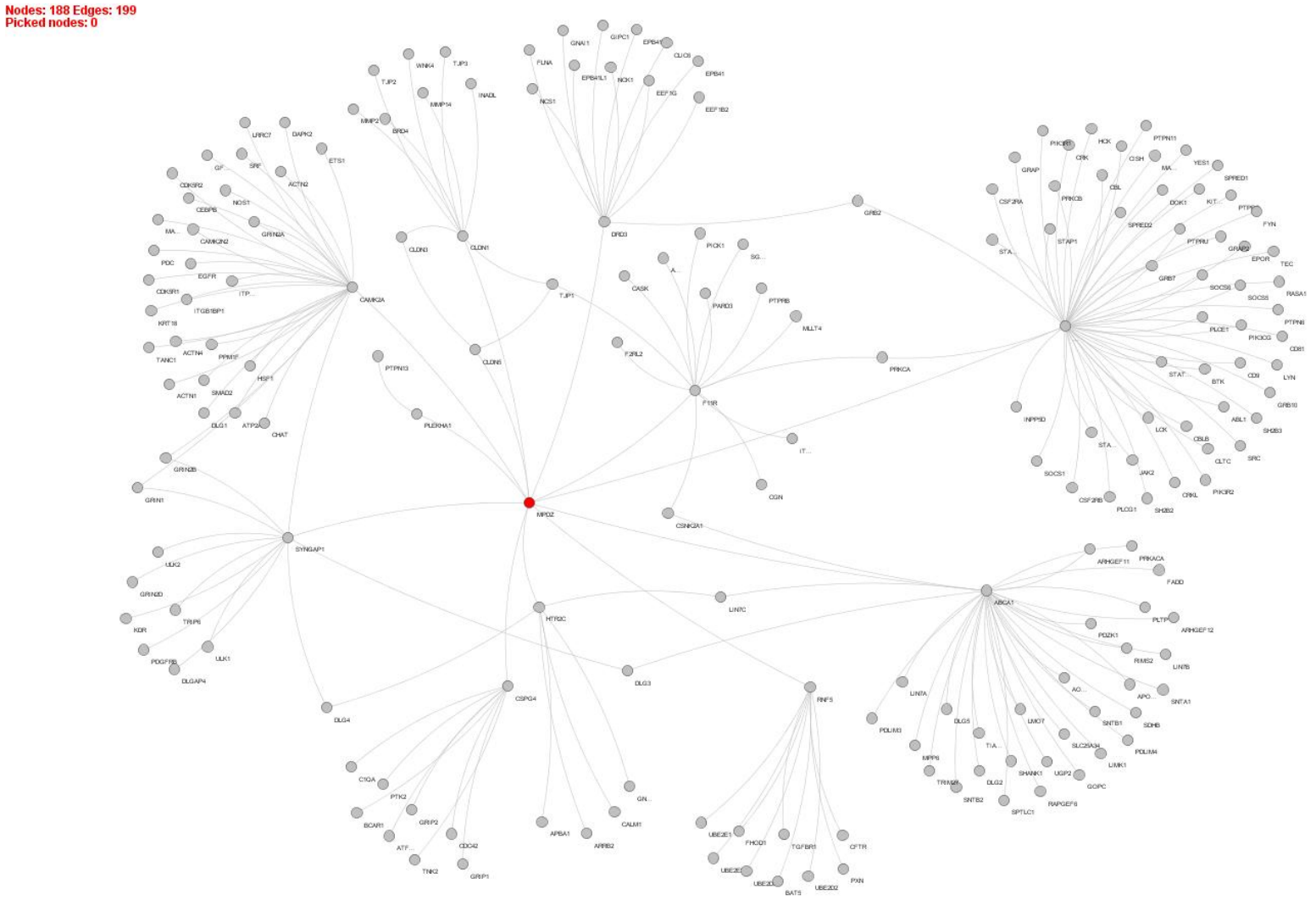


Abbildung A.8: Mit VANESA auf Ebene 2 erweitertes *MPDZ/MUPP-1*-Netzwerk (Root-Knoten rot hervorgehoben)

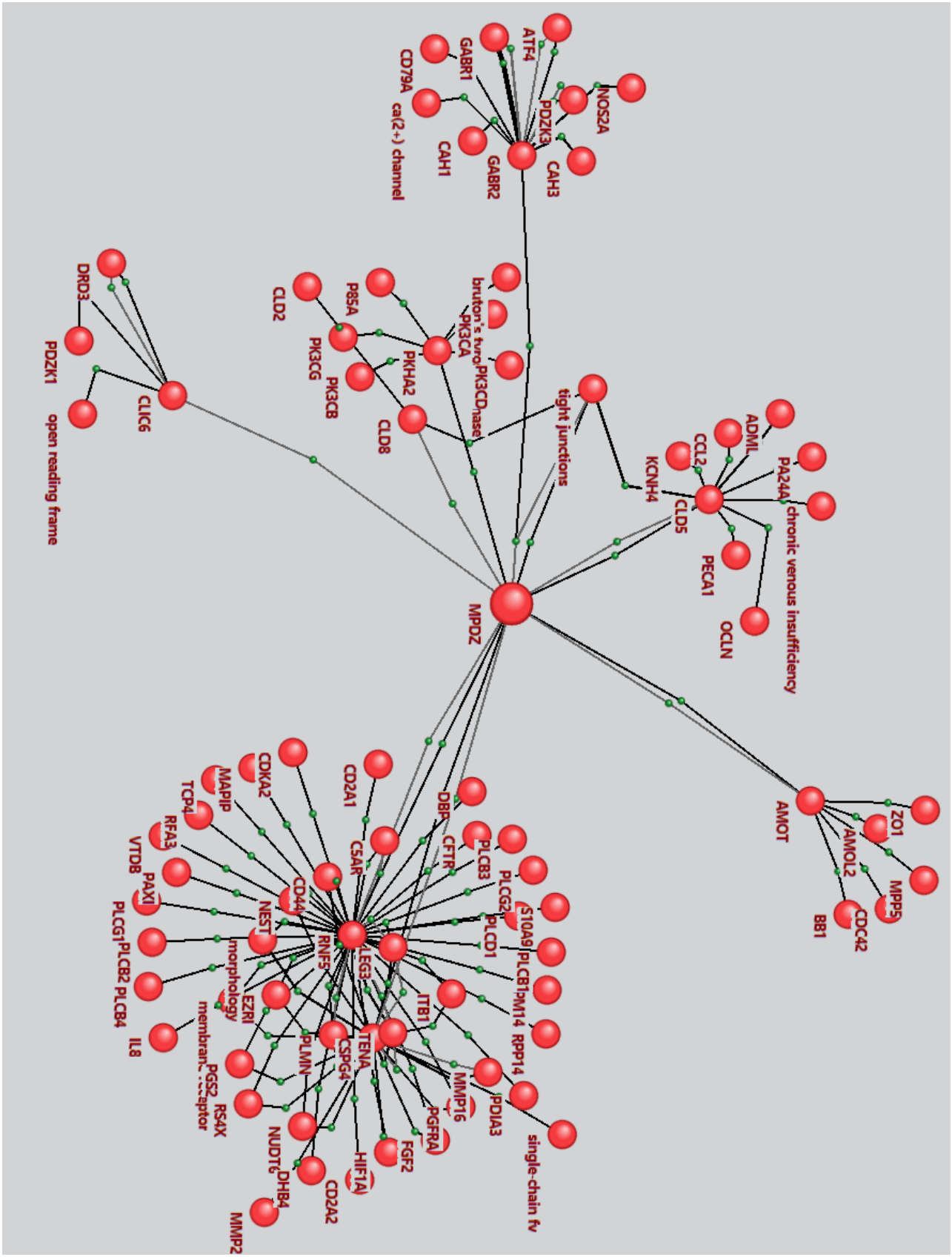


Abbildung A.9: Mit ANDSystem auf Ebene 2 erweitertes *MPDZ/MUPP-1*-Netzwerk

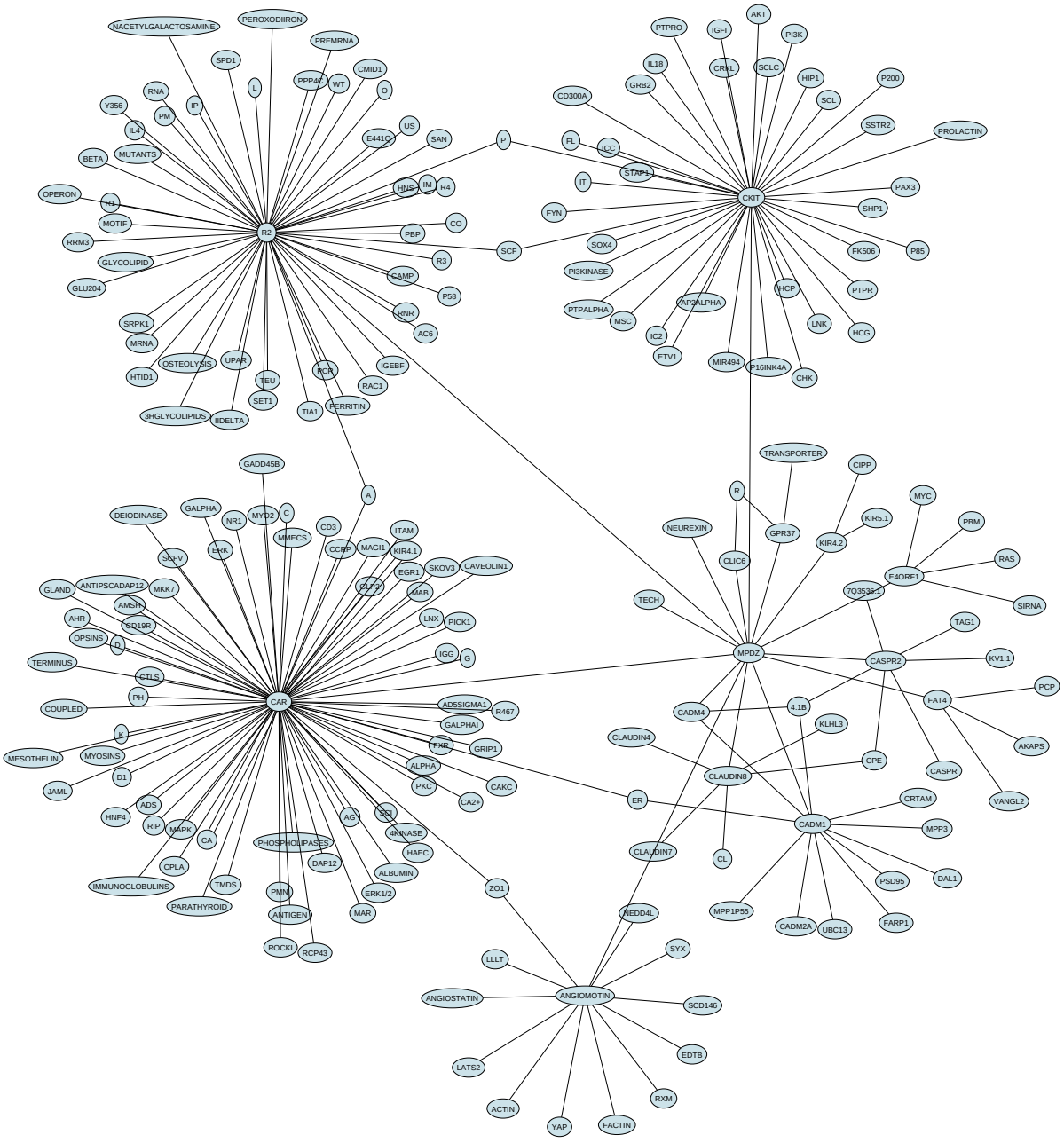


Abbildung A.10: Mit *FraMeTex* auf Ebene 2 erweitertes MPDZ/MUPP1-Netzwerk

```
MPDZ-AMOT
MPDZ-CLD5
MPDZ-CLD8
MPDZ-CLIC6
MPDZ-CSPG4
MPDZ-GABR2
MPDZ-PKHA2
MPDZ-RNF5
AMOT-AMOL2
AMOT-BB1
AMOT-CDC42
AMOT-MPP5
AMOT-ZO1
CLD5-OCLN
CLD5-PA24A
CLD5-PECA1
CLD5-ADML
CLD5-CCL2
CLD5-KCNH4
CLD8-CLD2
CLIC6-PDZK1
CLIC6-DRD3
CSPG4-PGFRA
CSPG4-HIF1A
...
```

Abbildung A.11: In Textdatei serialisiertes und mit ANDSystem rekonstruiertes MPDZ/MUPP1-Netzwerk (Ausschnitt)

Transitives Clustering in ANDSystem			
Nr.	**Protein-Komplex**	**Nr.**	**Protein-Komplex**
1	MPDZ - AMOT - AMOL2	41	MPDZ - PKHA2 - PK3CD
2	MPDZ - AMOT - BB1	42	MPDZ - PKHA2 - PK3CG
3	MPDZ - AMOT - CDC42	43	MPDZ - RNF5 - C5AR
4	MPDZ - AMOT - MPP5	44	MPDZ - RNF5 - CD2A1
5	MPDZ - AMOT - ZO1	45	MPDZ - RNF5 - CD2A2
6	MPDZ - CLD5 - ADML	46	MPDZ - RNF5 - CD44
7	MPDZ - CLD5 - CCL2	47	MPDZ - RNF5 - CDKA2
8	MPDZ - CLD5 - KCNH4	48	MPDZ - RNF5 - CFTR
9	MPDZ - CLD5 - OCLN	49	MPDZ - RNF5 - DBP
10	MPDZ - CLD5 - PA24A	50	MPDZ - RNF5 - DHB4
11	MPDZ - CLD5 - PECA1	51	MPDZ - RNF5 - EZRI
12	MPDZ - CLD8 - CLD2	52	MPDZ - RNF5 - FGF2
13	MPDZ - CLIC6 - DRD3	53	MPDZ - RNF5 - HIF1A
14	MPDZ - CLIC6 - PDZK1	54	MPDZ - RNF5 - IL8
15	MPDZ - CSPG4 - CD44	55	MPDZ - RNF5 - ITB1
16	MPDZ - CSPG4 - EZRI	56	MPDZ - RNF5 - LEG3
17	MPDZ - CSPG4 - FGF2	57	MPDZ - RNF5 - MAPIP
18	MPDZ - CSPG4 - HIF1A	58	MPDZ - RNF5 - NEST
19	MPDZ - CSPG4 - ITB1	59	MPDZ - RNF5 - NUDT6
20	MPDZ - CSPG4 - LEG3	60	MPDZ - RNF5 - PAXI

Nr.	Protein-Komplex	Nr.	Protein-Komplex
21	MPDZ - CSPG4 - MMP16	61	MPDZ - RNF5 - PDIA3
22	MPDZ - CSPG4 - MMP2	62	MPDZ - RNF5 - PGFRA
23	MPDZ - CSPG4 - NEST	63	MPDZ - RNF5 - PGS2
24	MPDZ - CSPG4 - NUDT6	64	MPDZ - RNF5 - PLCB1
25	MPDZ - CSPG4 - PGFRA	65	MPDZ - RNF5 - PLCB2
26	MPDZ - CSPG4 - PGS2	66	MPDZ - RNF5 - PLCB3
27	MPDZ - CSPG4 - PLMN	67	MPDZ - RNF5 - PLCB4
28	MPDZ - CSPG4 - RS4X	68	MPDZ - RNF5 - PLCD1
29	MPDZ - CSPG4 - TENA	69	MPDZ - RNF5 - PLCG1
30	MPDZ - GABR2 - ATF4	70	MPDZ - RNF5 - PLCG2
31	MPDZ - GABR2 - CAH1	71	MPDZ - RNF5 - PLMN
32	MPDZ - GABR2 - CAH3	72	MPDZ - RNF5 - PM14
33	MPDZ - GABR2 - CD79A	73	MPDZ - RNF5 - RFA3
34	MPDZ - GABR2 - GABR1	74	MPDZ - RNF5 - RPP14
35	MPDZ - GABR2 - NOS2A	75	MPDZ - RNF5 - RS4X
36	MPDZ - GABR2 - PDZK3	76	MPDZ - RNF5 - S10A9
37	MPDZ - GABR2 - ca(2+)	77	MPDZ - RNF5 - TCP4
38	MPDZ - PKHA2 - P85A	78	MPDZ - RNF5 - TENA
39	MPDZ - PKHA2 - PK3CA	79	MPDZ - RNF5 - VTDB
40	MPDZ - PKHA2 - PK3CB		

Nr.	Protein-Komplex	Nr.	Protein-Komplex

Tabelle A.2: Identifizierte Protein-Komplexe im von ANDSystem rekonstruierten Netzwerk

Transitives Clustering in VANESA			
Nr.	**Protein-Komplex**	**Nr.**	**Protein-Komplex**
1	MPDZ - ABCA1 - AOX1	95	MPDZ - F11R - ASIP
2	MPDZ - ABCA1 - APOA1	96	MPDZ - F11R - CASK
3	MPDZ - ABCA1 - ARHGEF11	97	MPDZ - F11R - CGN
4	MPDZ - ABCA1 - ARHGEF12	98	MPDZ - F11R - CSNK2A1
5	MPDZ - ABCA1 - CSNK2A1	99	MPDZ - F11R - F2RL2
6	MPDZ - ABCA1 - DLG2	100	MPDZ - F11R - ITGAL
7	MPDZ - ABCA1 - DLG3	101	MPDZ - F11R - MLLT4
8	MPDZ - ABCA1 - DLG5	102	MPDZ - F11R - PARD3
9	MPDZ - ABCA1 - FADD	103	MPDZ - F11R - PICK1
10	MPDZ - ABCA1 - GOPC	104	MPDZ - F11R - PRKCA
11	MPDZ - ABCA1 - LIMK1	105	MPDZ - F11R - PTPRB
12	MPDZ - ABCA1 - LIN7A	106	MPDZ - F11R - SGTA
13	MPDZ - ABCA1 - LIN7B	107	MPDZ - F11R - TJP1
14	MPDZ - ABCA1 - LIN7C	108	MPDZ - HTR2C - APBA1
15	MPDZ - ABCA1 - LM07	109	MPDZ - HTR2C - ARRB2
16	MPDZ - ABCA1 - MPP6	110	MPDZ - HTR2C - CALM1
17	MPDZ - ABCA1 - PDLIM3	111	MPDZ - HTR2C - DLG4
18	MPDZ - ABCA1 - PDLIM4	112	MPDZ - HTR2C - GNAQ
19	MPDZ - ABCA1 - PDZK1	113	MPDZ - HTR2C - LIN7C
20	MPDZ - ABCA1 - PLTP	114	MPDZ - KIT - ABL1

Nr.	Protein-Komplex	Nr.	Protein-Komplex
21	MPDZ - ABCA1 - PRKACA	115	MPDZ - KIT - BTK
22	MPDZ - ABCA1 - RAPGEF6	116	MPDZ - KIT - CBL
23	MPDZ - ABCA1 - RIMS2	117	MPDZ - KIT - CBLB
24	MPDZ - ABCA1 - SDHB	118	MPDZ - KIT - CD81
25	MPDZ - ABCA1 - SHANK1	119	MPDZ - KIT - CD9
26	MPDZ - ABCA1 - SLC25A34	120	MPDZ - KIT - CISH
27	MPDZ - ABCA1 - SNTA1	121	MPDZ - KIT - CLTC
28	MPDZ - ABCA1 - SNTB1	122	MPDZ - KIT - CRK
29	MPDZ - ABCA1 - SNTB2	123	MPDZ - KIT - CRKL
30	MPDZ - ABCA1 - SPTLC	124	MPDZ - KIT - CSF2RA
31	MPDZ - ABCA1 - TIAM1	125	MPDZ - KIT - CSF2RB
32	MPDZ - ABCA1 - TRIM27	126	MPDZ - KIT - DOK1
33	MPDZ - ABCA1 - UGP2	127	MPDZ - KIT - EPOR
34	MPDZ - CAMK2A - ACTN1	128	MPDZ - KIT - FYN
35	MPDZ - CAMK2A - ACTN2	129	MPDZ - KIT - GRAP
36	MPDZ - CAMK2A - ACTN4	130	MPDZ - KIT - GRAP2
37	MPDZ - CAMK2A - ATP2A2	131	MPDZ - KIT - GRB10
38	MPDZ - CAMK2A - CAMK2N2	132	MPDZ - KIT - GRB2
39	MPDZ - CAMK2A - CD5KR1	133	MPDZ - KIT - GRB7
40	MPDZ - CAMK2A - CDK5R2	134	MPDZ - KIT - HCK
41	MPDZ - CAMK2A - CEBPB	135	MPDZ - KIT - INPP5D

Nr.	Protein-Komplex	Nr.	Protein-Komplex
42	MPDZ - CAMK2A - CHAT	136	MPDZ - KIT - JAK2
43	MPDZ - CAMK2A - DAPK2	137	MPDZ - KIT - KITLG
44	MPDZ - CAMK2A - DLG1	138	MPDZ - KIT - LCK
45	MPDZ - CAMK2A - EGFR	139	MPDZ - KIT - LYN
46	MPDZ - CAMK2A - ETS1	140	MPDZ - KIT - MATK
47	MPDZ - CAMK2A - GFAP	141	MPDZ - KIT - PIK3CG
48	MPDZ - CAMK2A - GRIN1	142	MPDZ - KIT - PIK3R1
49	MPDZ - CAMK2A - GRIN2A	143	MPDZ - KIT - PIK3R2
50	MPDZ - CAMK2A - GRIN2B	144	MPDZ - KIT - PLCE1
51	MPDZ - CAMK2A - HSF1	145	MPDZ - KIT - PLCG1
52	MPDZ - CAMK2A - ITGB1BP1	146	MPDZ - KIT - PRKCA
53	MPDZ - CAMK2A - ITPKA	147	MPDZ - KIT - PRKCB
54	MPDZ - CAMK2A - KRT18	148	MPDZ - KIT - PTPN11
55	MPDZ - CAMK2A - LRRC7	149	MPDZ - KIT - PTPN6
56	MPDZ - CAMK2A - MAPT	150	MPDZ - KIT - PTPRO
57	MPDZ - CAMK2A - NOS1	151	MPDZ - KIT - PTPRU
58	MPDZ - CAMK2A - PDC	152	MPDZ - KIT - RASA1
59	MPDZ - CAMK2A - PPM1F	153	MPDZ - KIT - SH2B2
60	MPDZ - CAMK2A - SMAD2	154	MPDZ - KIT - SH2B3
61	MPDZ - CAMK2A - SRF	155	MPDZ - KIT - SOCS1
62	MPDZ - CAMK2A - SYNGAP1	156	MPDZ - KIT - SOCS5

Nr.	Protein-Komplex	Nr.	Protein-Komplex
63	MPDZ - CAMK2A - TANC1	157	MPDZ - KIT - SOCS6
64	MPDZ - CLDN1 - BRD4	158	MPDZ - KIT - SPRED1
65	MPDZ - CLDN1 - CLDN3	159	MPDZ - KIT - SPRED2
66	MPDZ - CLDN1 - INADL	160	MPDZ - KIT - SRC
67	MPDZ - CLDN1 - MMP14	161	MPDZ - KIT - STAP1
68	MPDZ - CLDN1 - MMP2	162	MPDZ - KIT - STAT1
69	MPDZ - CLDN1 - TJP1	163	MPDZ - KIT - STAT5B
70	MPDZ - CLDN1 - TJP2	164	MPDZ - KIT - TEC
71	MPDZ - CLDN1 - TJP3	165	MPDZ - KIT - YES1
72	MPDZ - CLDN1 - WNK4	166	MPDZ - PLEKHA1 - PTPN13
73	MPDZ - CLDN5 - CLDN3	167	MPDZ - RNF5 - BAT5
74	MPDZ - CLDN5 - TJP1	168	MPDZ - RNF5 - CFTR
75	MPDZ - CSPG4 - ATF7IP	169	MPDZ - RNF5 - FHOD1
76	MPDZ - CSPG4 - BCAR1	170	MPDZ - RNF5 - PXN
77	MPDZ - CSPG4 - C1QA	171	MPDZ - RNF5 - TGFBR1
78	MPDZ - CSPG4 - CDC42	172	MPDZ - RNF5 - UBE2D2
79	MPDZ - CSPG4 - GRIP1	173	MPDZ - RNF5 - UBE2D3
80	MPDZ - CSPG4 - GRIP2	174	MPDZ - RNF5 - UBE2E1
81	MPDZ - CSPG4 - PTK2	175	MPDZ - RNF5 - UBE2E3
82	MPDZ - CSPG4 - TNK2	176	MPDZ - SYNGAP1 - CAM2KA
83	MPDZ - DRD3 - CLIC6	177	MPDZ - SYNGAP1 - DLG3

Nr.	Protein-Komplex	Nr.	Protein-Komplex
84	MPDZ - DRD3 - EEF1B2	178	MPDZ - SYNGAP1 - DLG4
85	MPDZ - DRD3 - EEF1G	179	MPDZ - SYNGAP1 - DLGAP4
86	MPDZ - DRD3 - EPB41	180	MPDZ - SYNGAP1 - GRIN1
87	MPDZ - DRD3 - EPB41L1	181	MPDZ - SYNGAP1 - GRIN2B
88	MPDZ - DRD3 - EPB41L2	182	MPDZ - SYNGAP1 - GRIN2D
89	MPDZ - DRD3 - FLNA	183	MPDZ - SYNGAP1 - KDR
90	MPDZ - DRD3 - GIPC1	184	MPDZ - SYNGAP1 - PDGFRB
91	MPDZ - DRD3 - GNAI1	185	MPDZ - SYNGAP1 - TRIP6
92	MPDZ - DRD3 - GRB2	186	MPDZ - SYNGAP1 - ULK1
93	MPDZ - DRD3 - NCK1	187	MPDZ - SYNGAP1 - ULK2
94	MPDZ - DRD3 - NCS1		

Tabelle A.3: Identifizierte Protein-Komplexe im von VANESA rekonstruierten Netzwerk

Transitives Clustering in *FraMeTex*			
Nr.	**Protein-Komplex**	**Nr.**	**Protein-Komplex**
1	MPDZ - ANGIOMOTIN - ACTIN	107	MPDZ - CKIT - CD300A
2	MPDZ - ANGIOMOTIN - ANGIOSTATIN	108	MPDZ - CKIT - CHK
3	MPDZ - ANGIOMOTIN - EDTB	109	MPDZ - CKIT - CRKL
4	MPDZ - ANGIOMOTIN - FACTIN	110	MPDZ - CKIT - ETV1
5	MPDZ - ANGIOMOTIN - LATS2	111	MPDZ - CKIT - FK506
6	MPDZ - ANGIOMOTIN - LLLT	112	MPDZ - CKIT - FL
7	MPDZ - ANGIOMOTIN - NEDD4L	113	MPDZ - CKIT - FYN
8	MPDZ - ANGIOMOTIN - RXM	114	MPDZ - CKIT - GRB2
9	MPDZ - ANGIOMOTIN - SCD146	115	MPDZ - CKIT - HCG
10	MPDZ - ANGIOMOTIN - SYX	116	MPDZ - CKIT - HCP
11	MPDZ - ANGIOMOTIN - YAP	117	MPDZ - CKIT - HIP1
12	MPDZ - ANGIOMOTIN - ZO1	118	MPDZ - CKIT - IC2
13	MPDZ - CADM1 - 4.1B	119	MPDZ - CKIT - ICC
14	MPDZ - CADM1 - CADM2A	120	MPDZ - CKIT - IGFI
15	MPDZ - CADM1 - CADM4	121	MPDZ - CKIT - IL18
16	MPDZ - CADM1 - CRTAM	122	MPDZ - CKIT - LNK
17	MPDZ - CADM1 - DAL1	123	MPDZ - CKIT - MIR494
18	MPDZ - CADM1 - ER	124	MPDZ - CKIT - MSC
19	MPDZ - CADM1 - FARP1	125	MPDZ - CKIT - P
20	MPDZ - CADM1 - MPP1P55	126	MPDZ - CKIT - P16INK4A

Nr.	Protein-Komplex	Nr.	Protein-Komplex
21	MPDZ - CADM1 - MPP3	127	MPDZ - CKIT - P200
22	MPDZ - CADM1 - PSD95	128	MPDZ - CKIT - P85
23	MPDZ - CADM1 - UBC13	129	MPDZ - CKIT - PAX3
24	MPDZ - CADM4 - 4.1B	130	MPDZ - CKIT - PI3K
25	MPDZ - CADM4 - CADM1	131	MPDZ - CKIT - PI3KINASE
26	MPDZ - CAR - 4KINASE	132	MPDZ - CKIT - PROLACTIN
27	MPDZ - CAR - AD5SIGMA1	133	MPDZ - CKIT - PTPALPHA
28	MPDZ - CAR - ADS	134	MPDZ - CKIT - PTPR
29	MPDZ - CAR - AG	135	MPDZ - CKIT - PTPRO
30	MPDZ - CAR - AHR	136	MPDZ - CKIT - SCF
31	MPDZ - CAR - ALBUMIN	137	MPDZ - CKIT - SCL
32	MPDZ - CAR - ALPHA	138	MPDZ - CKIT - SCLC
33	MPDZ - CAR - AMSH	139	MPDZ - CKIT - SHP1
34	MPDZ - CAR - ANTIGEN	140	MPDZ - CKIT - SOX4
35	MPDZ - CAR - ANTIPSCADAP12	141	MPDZ - CKIT - SSTR2
36	MPDZ - CAR - C	142	MPDZ - CKIT - STAP1
37	MPDZ - CAR - CA2	143	MPDZ - CLAUDIN8 - CL
38	MPDZ - CAR - CAKC	144	MPDZ - CLAUDIN8 - CLAUDIN4
39	MPDZ - CAR - CAR	145	MPDZ - CLAUDIN8 - CLAUDIN7
40	MPDZ - CAR - CAVEOLIN1	146	MPDZ - CLAUDIN8 - CPE
41	MPDZ - CAR - CCRP	147	MPDZ - CLAUDIN8 - KLHL3

Nr.	Protein-Komplex	Nr.	Protein-Komplex
42	MPDZ - CAR - CD19R	148	MPDZ - CLIC6 - R
43	MPDZ - CAR - CD3	149	MPDZ - E4ORF1 - MYC
44	MPDZ - CAR - COUPLED	150	MPDZ - E4ORF1 - PBM
45	MPDZ - CAR - CPLA	151	MPDZ - E4ORF1 - RAS
46	MPDZ - CAR - CTLS	152	MPDZ - E4ORF1 - SIRNA
47	MPDZ - CAR - D	153	MPDZ - FAT4 - AKAPS
48	MPDZ - CAR - D1	154	MPDZ - FAT4 - PCP
49	MPDZ - CAR - DAP12	155	MPDZ - FAT4 - VANGL2
50	MPDZ - CAR - DEIODINASE	156	MPDZ - GPR37 - R
51	MPDZ - CAR - EGR1	157	MPDZ - GPR37 - TRANSPORTER
52	MPDZ - CAR - ER	158	MPDZ - KIR4.2 - KIR5.1
53	MPDZ - CAR - ERK	159	MPDZ - R2 - 3HGLYCOLIPIDS
54	MPDZ - CAR - ERK12	160	MPDZ - R2 - A
55	MPDZ - CAR - FXR	161	MPDZ - R2 - AC6
56	MPDZ - CAR - G	162	MPDZ - R2 - BETA
57	MPDZ - CAR - GADD45B	163	MPDZ - R2 - CAMP
58	MPDZ - CAR - GALPHA	164	MPDZ - R2 - CMID1
59	MPDZ - CAR - GALPHAI	165	MPDZ - R2 - CO
60	MPDZ - CAR - GLAND	166	MPDZ - R2 - E441Q
61	MPDZ - CAR - GLP2	167	MPDZ - R2 - FERRITIN
62	MPDZ - CAR - GRIP1	168	MPDZ - R2 - GLU204

Nr.	Protein-Komplex	Nr.	Protein-Komplex
63	MPDZ - CAR - HAEC	169	MPDZ - R2 - GLYCOLIPID
64	MPDZ - CAR - HNF4	170	MPDZ - R2 - HNS
65	MPDZ - CAR - IGG	171	MPDZ - R2 - HTID1
66	MPDZ - CAR - IMMUNOGLOBULINS	172	MPDZ - R2 - IGEBF
67	MPDZ - CAR - ITAM	173	MPDZ - R2 - IIDELTA
68	MPDZ - CAR - JAML	174	MPDZ - R2 - IL4
69	MPDZ - CAR - K	175	MPDZ - R2 - IM
70	MPDZ - CAR - KIR4.1	176	MPDZ - R2 - IP
71	MPDZ - CAR - LNX	177	MPDZ - R2 - L
72	MPDZ - CAR - MAB	178	MPDZ - R2 - MOTIF
73	MPDZ - CAR - MAGI1	179	MPDZ - R2 - MRNA
74	MPDZ - CAR - MAPK	180	MPDZ - R2 - MUTANTS
75	MPDZ - CAR - MAR	181	MPDZ - R2 - NACETYLGALACTOSAMINE
76	MPDZ - CAR - MESOTHELIN	182	MPDZ - R2 - O
77	MPDZ - CAR - MKK7	183	MPDZ - R2 - OPERON
78	MPDZ - CAR - MMECS	184	MPDZ - R2 - OSTEOLYSIS
79	MPDZ - CAR - MYO2	185	MPDZ - R2 - P
80	MPDZ - CAR - MYOSINS	186	MPDZ - R2 - P58
81	MPDZ - CAR - NR1	187	MPDZ - R2 - PBP
82	MPDZ - CAR - OPSINS	188	MPDZ - R2 - PCR
83	MPDZ - CAR - PARATHYROID	189	MPDZ - R2 - PEROXODIIRON

Nr.	Protein-Komplex	Nr.	Protein-Komplex
84	MPDZ - CAR - PH	190	MPDZ - R2 - PM
85	MPDZ - CAR - PHOSPHOLIPASES	191	MPDZ - R2 - PPP4C
86	MPDZ - CAR - PICK1	192	MPDZ - R2 - PREMRNA
87	MPDZ - CAR - PKC	193	MPDZ - R2 - R1
88	MPDZ - CAR - PMN	194	MPDZ - R2 - R2
89	MPDZ - CAR - R467	195	MPDZ - R2 - R3
90	MPDZ - CAR - RCP43	196	MPDZ - R2 - R4
91	MPDZ - CAR - RIP	197	MPDZ - R2 - RAC1
92	MPDZ - CAR - ROCKI	198	MPDZ - R2 - RNA
93	MPDZ - CAR - SCFV	199	MPDZ - R2 - RNR
94	MPDZ - CAR - SCI	200	MPDZ - R2 - RRM3
95	MPDZ - CAR - SKOV3	201	MPDZ - R2 - SAN
96	MPDZ - CAR - TERMINUS	202	MPDZ - R2 - SCF
97	MPDZ - CAR - TMDS	203	MPDZ - R2 - SET1
98	MPDZ - CAR - ZO1	204	MPDZ - R2 - SPD1
99	MPDZ - CASPR2 - 4.1B	205	MPDZ - R2 - SRPK1
100	MPDZ - CASPR2 - 7Q3536.1	206	MPDZ - R2 - TEU
101	MPDZ - CASPR2 - CASPR	207	MPDZ - R2 - TIA1
102	MPDZ - CASPR2 - CPE	208	MPDZ - R2 - UPAR
103	MPDZ - CASPR2 - KV1.1	209	MPDZ - R2 - US
104	MPDZ - CASPR2 - TAG1	210	MPDZ - R2 - WT

Nr.	Protein-Komplex	Nr.	Protein-Komplex
105	MPDZ - CKIT - AKT	211	MPDZ - R2 - Y356
106	MPDZ - CKIT - AP2ALPHA		

Tabelle A.4: Identifizierte Protein-Komplexe im von *FraMeTex* rekonstruierten Netzwerk

Literaturverzeichnis

[ABB+00] Ashburner, M. ; Ball, C.A. ; Blake, J.A. ; Botstein, D. ; Butler, H. ; Cherry, J.M. ; Davis, A.P. ; Dolinski, K. ; Dwight, S.S. ; Eppig, J.T. u. a.: Gene Ontology: tool for the unification of biology. In: Nature genetics 25 (2000), Nr. 1, S. 25

[ABE+08] Antezana, Erick ; Blondé, Ward ; Egana, Mikel ; Rutherford, Alistair ; Stevens, Robert ; De Baets, Bernard ; Mironov, Vladimir ; Kuiper, Martin: Structuring the life science resourceome for semantic systems biology: lessons from the BioGateway Project. In: Proceedings of the Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS): November 28, 2008; Edinburgh, United Kingdom, 2008

[ABE+09] Antezana, Erick ; Blondé, Ward ; Egaña, Mikel ; Rutherford, Alistair ; Stevens, Robert ; De Baets, Bernard ; Mironov, Vladimir ; Kuiper, Martin: BioGateway: a semantic systems biology tool for the life sciences. In: BMC Bioinformatics 10 (2009), Nr. 10, S. S11

[ABV+11] Antezana, Erick ; Blondé, Ward ; Venkatesan, Aravind ; De Baets, Bernard ; Mironov, Vladimir ; Kuiper, Martin: Semantic systems biology: enabling integrative biology via semantic web technologies. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics ACM, 2011, S. 58

[AD07] Abulaish, Muhammad ; Dey, Lipika: Biological relation extraction and query answering from medline abstracts using ontology-based text mining. In: Data & Knowledge Engineering 61 (2007), Nr. 2, S. 228–262

[ADP+06] Aman, EE ; Demenkov, PS ; Pintus, SS ; Nemiatov, AI ; Apasieva, NV ; Korotkov, RO ; Ignatieva, EV ; Podkolodny, NL ; Ivanisenko, VA: Development of a computer system for the automated reconstruction of molecular genetic interaction networks. In: Fifth International Conference on Bioinformatics of Genome Regulation and Structure (BGRS) Bd. 3, 2006, S. 15–18

[AECBF+12] Aslaoui-Errafi, Zahira ; Cohen-Boulakia, Sarah ; Froidevaux, Christine ; Gloaguen, Pauline ; Poupon, Anne ; Rougny, Adrien ; Yahiaoui, Meriem: Towards a logic-based method to infer provenance-aware molecular networks. In: Proc. of the 1st ECML/PKDD International workshop on Learning and Discovery in Symbolic Systems Biology (LDSSB), 2012, S. 103–110

[AH11] Allemang, D. ; Hendler, J.: Semantic Web for the working ontologist: effective modeling in RDFS and OWL. Morgan Kaufmann, 2011. – ISBN 978–0–12–373556–0

[AHS14] Ahmad, Sahar ; Hasan, Osman ; Siddique, Umair: Towards formal reasoning about molecular pathways in HOL. In: WETICE Conference (WETICE), 2014 IEEE 23rd International IEEE, 2014, S. 378–383

[AHST14] Ahmad, Sohaib ; Hasan, Osman ; Siddique, Umair ; Tahar, Sofiéne: Formalization of Zsyntax to Reason About Molecular Pathways in HOL4. In: Formal Methods: Foundations and Applications. Springer, 2014, S. 32–47

[Alb08] Alberts, B.: Molecular Biology of the Cell: Reference edition. Garland Science, 2008 (Molecular Biology of the Cell: Reference Edition Bd. 1). – ISBN 978–0815341116

[BA00] Bairoch, Amos ; Apweiler, Rolf: The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. In: Nucleic acids research 28 (2000), Nr. 1, S. 45–48

[Bai00] Bairoch, Amos: The ENZYME database in 2000. In: Nucleic acids research 28 (2000), Nr. 1, S. 304–305

[Bak11] Baker, S.: Final jeopardy: Man vs. machine and the quest to know everything. Houghton Mifflin Harcourt, 2011

[Bar97] Barrett, AJ: Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997). In: European journal of biochemistry/FEBS 250 (1997), Nr. 1, S. 1

[BC07] Batchelor, Colin R. ; Corbett, Peter T.: Semantic enrichment of journal articles using chemical named entity recognition. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions Association for Computational Linguistics, 2007, S. 45–48

[BCC+12] Benson, Dennis A. ; Cavanaugh, Mark ; Clark, Karen ; Karsch-Mizrachi, Ilene ; Lipman, David J. ; Ostell, James ; Sayers, Eric W.: GenBank. In: Nucleic acids research (2012), S. gks1195

[BCW05] Bizer, Christian ; Cyganiak, Richard ; Watkins, E R.: Ng4j-named graphs api for jena. In: Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece Citeseer, 2005

[BDDF10] Boniolo, Giovanni ; D'Agostino, Marcello ; Di Fiore, Pier P.: Zsyntax: a formal language for molecular biology with projected applications in text mining and biological prediction. In: PloS one 5 (2010), Nr. 3, S. e9511

[BJK+14] Brinkrolf, Christoph ; Janowski, Sebastian J. ; Kormeier, Benjamin ; Lewinski, Martin ; Hippe, Klaus ; Borck, Daniela ; Hofestädt, Ralf: VANESA-A Software Application for the Visualization and Analysis of Networks in System Biology Applications. In: Journal of integrative bioinformatics 11 (2014), Nr. 2, S. 239

[BKVH02] Broekstra, J. ; Kampman, A. ; Van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: The Semantic Web - ISWC 2002 (2002), S. 54–68

[BL85] Brachman, R.J. ; Levesque, H.J.: Readings in knowledge representation / AT and T Bell Labs. Morgan Kaufmann Publishers, 1985. – Forschungsbericht. – ISBN 978–0934613019

[BL96] Berners-Lee, T.: WWW: Past, present, and future. In: Computer 29 (1996), Nr. 10, S. 69–77

[BLHL+01] Berners-Lee, T. ; Hendler, J. ; Lassila, O. u. a.: The semantic web. (2001)

[BLL+06] Bechhofer, S. ; Liebig, T. ; Luther, M. ; Noppens, O. ; Patel-Schneider, P. ; Suntisrivaraporn, B. ; Turhan, A.Y. ; Weithöner, T.: DIG 2.0–Towards a flexible interface for Description Logic reasoners. In: Proc. of the OWL Experiences and Directions Workshop at the ISWC Bd. 6 Citeseer, 2006

[BM03] Bechhofer, S. ; Möller, R.: The DIG description logic interface: DIG/1.1. In: Proceedings of the 2003 Description Logic Workshop (DL 2003), 2003

[BOH11] Bostock, Michael ; Ogievetsky, Vadim ; Heer, Jeffrey: D^3 data-driven documents. In: Visualization and Computer Graphics, IEEE Transactions on 17 (2011), Nr. 12, S. 2301–2309

[Boo54] Boole, George: An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities. Bd. 2. Walton and Maberly, 1854

[BYRN+99] Baeza-Yates, R. ; Ribeiro-Neto, B. u. a.: Modern information retrieval. Bd. 463. ACM press New York., 1999

[CAB+14] Caspi, Ron ; Altman, Tomer ; Billington, Richard ; Dreher, Kate ; Foerster, Hartmut ; Fulcher, Carol A. ; Holland, Timothy A. ; Keseler, Ingrid M. ; Kothari, Anamika ; Kubo, Aya u. a.: The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. In: Nucleic acids research 42 (2014), Nr. D1, S. D459–D471

[CACP+07] Chatr-Aryamontri, Andrew ; Ceol, Arnaud ; Palazzi, Luisa M. ; Nardelli, Giuliano ; Schneider, Maria V. ; Castagnoli, Luisa ; Cesareni, Gianni: MINT: the Molecular INTeraction database. In: Nucleic acids research 35 (2007), Nr. suppl 1, S. D572–D574

[CBHS05] Carroll, Jeremy J. ; Bizer, Christian ; Hayes, Pat ; Stickler, Patrick: Named graphs, provenance and trust. In: Proceedings of the 14th international conference on World Wide Web ACM, 2005, S. 613–622

[CDD+04] Carroll, J.J. ; Dickinson, I. ; Dollin, C. ; Reynolds, D. ; Seaborne, A. ; Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters ACM, 2004, S. 74–83

[CGK+90] Chimenti, Danette ; Gamboa, Ruben ; Krishnamurthy, Ravi ; Naqvi, Shamim ; Tsur, Shalom ; Zaniolo, Carlo: The LDL system prototype. In: Knowledge and Data Engineering, IEEE Transactions on 2 (1990), Nr. 1, S. 76–90

[CGW95] Cunningham, H. ; Gaizauskas, R.J. ; Wilks, Y.: A General Architecture for Text Engineering (GATE): A New Approach to Language Engineering R&D. Citeseer, 1995

[CH04] Cooper, G.M. ; Hausman, R.E.: The Cell: A Molecular Approach. ASM Press, 2004. – ISBN 978–0878932146

[Cho56] Chomsky, Noam: Three models for the description of language. In: Information Theory, IRE Transactions on 2 (1956), Nr. 3, S. 113–124

[Cho59] Chomsky, Noam: On certain formal properties of grammars. In: Information and control 2 (1959), Nr. 2, S. 137–167

[CIM+09] Carbon, Seth ; Ireland, Amelia ; Mungall, Christopher J. ; Shu, ShengQiang ; Marshall, Brad ; Lewis, Suzanna u. a.: AmiGO: online access to ontology and annotation data. In: Bioinformatics 25 (2009), Nr. 2, S. 288–289

[CM10] Clocksin, W.F. ; Mellish, C.S.: Programming in PROLOG. Springer, 2010. – ISBN 978–3540006787

[CMB+04] Camon, Evelyn ; Magrane, Michele ; Barrell, Daniel ; Lee, Vivian ; Dimmer, Emily ; Maslen, John ; Binns, David ; Harte, Nicola ; Lopez, Rodrigo ; Apweiler, Rolf: The Gene Ontology annotation (GOA) database: sharing knowledge in Uniprot with Gene Ontology. In: Nucleic acids research 32 (2004), Nr. suppl 1, S. D262–D266

[CMB+11] Cunningham, Hamish ; Maynard, Diana ; Bontcheva, Kalina ; Tablan, Valentin ; Aswani, Niraj ; Roberts, Ian ; Gorrell, Genevieve ; Funk, Adam ; Roberts, Angus ; Damljanovic, Danica ; Heitz, Thomas ; Greenwood, Mark A. ; Saggion, Horacio ; Petrak, Johann ; Li, Yaoyong ; Peters, Wim: Text Processing with GATE (Version 6). 2011 http://tinyurl.com/gatebook. – ISBN 978–0956599315

[CMR06] Corbett, Peter ; Murray-Rust, Peter: High-throughput identification of chemistry in life science texts. In: Computational Life Sciences II. Springer, 2006, S. 107–118

[CMT99] Cunningham, H. ; Maynard, D. ; Tablan, V.: JAPE: a Java annotation patterns engine. (1999)

[CNSS12] Czarnecki, Jan ; Nobeli, Irene ; Smith, Adrian M. ; Shepherd, Adrian J.: A text-mining system for extracting metabolic reactions from full-text articles. In: BMC bioinformatics 13 (2012), Nr. 1, S. 172

[Cod70] Codd, E. F.: A Relational Model of Data for Large Shared Data Banks. In: Commun. ACM 13 (1970), Nr. 6

[Cod82] Codd, Edgar F.: Relational database: a practical foundation for productivity. In: Communications of the ACM 25 (1982), Nr. 2, S. 109–117

[CR93] Colmerauer, Alain ; Roussel, Philippe: The birth of Prolog. In: ACM SIGPLAN Notices Bd. 28 ACM, 1993, S. 37–52

[CS04] Chen, Hao ; Sharp, Burt M.: Content-rich biological network constructed by mining PubMed abstracts. In: BMC bioinformatics 5 (2004), Nr. 1, S. 147

[Csi00] Csizmadia, Ferenc: JChem: Java applets and modules supporting chemical database handling from web browsers. In: Journal of Chemical Information and Computer Sciences 40 (2000), Nr. 2, S. 323–324

[Cun] Cunningham, Prof. H.: Infrastructure for Human Language Technology - GATE. http://gate.ac.uk/sale/gate-flyer/2009/gate-flyer-4-page.pdf. The University Of Sheffield. – Forschungsbericht

[Cun02] Cunningham, H.: GATE, a general architecture for text engineering. In: Computers and the Humanities 36 (2002), Nr. 2, S. 223–254

[CYS+05] Cheung, Kei-Hoi ; Yip, Kevin Y. ; Smith, Andrew ; Masiar, Andy ; Gerstein, Mark u. a.: YeastHub: a semantic web use case for integrating data in the life sciences domain. In: Bioinformatics 21 (2005), Nr. suppl 1, S. i85–i96

[Cza15] Czarnecki, Jan M.: The fully automated construction of metabolic pathways using text mining and knowledge-based constraints, Birkbeck, University of London, Diss., 2015

[Dar87] Darvas, Ferenc: METABOLEXPERT: an expert system for predicting metabolism of substances. In: QSAR in Environmental Toxicology-II. Springer, 1987, S. 71–81

[DIKI12] Demenkov, P.S. ; Ivanisenko, T.V. ; Kolchanov, N.A. ; Ivanisenko, V.A.: ANDVisio: A new tool for graphic visualization and analysis of literature mined associative gene networks in the ANDSystem. In: In silico biology 11 (2012), Nr. 3, S. 149–161

[Doy79] Doyle, J.: A truth maintenance system. In: Artificial intelligence 12 (1979), Nr. 3, S. 231–272

[DPK10] Dale, Joseph M. ; Popescu, Liviu ; Karp, Peter D.: Machine learning methods for metabolic pathway prediction. In: BMC bioinformatics 11 (2010), Nr. 1, S. 15

[DRHH+07] Day-Richter, John ; Harris, Midori A. ; Haendel, Melissa ; Lewis, Suzanna u. a.: OBO-Edit - an ontology editor for biologists. In: Bioinformatics 23 (2007), Nr. 16, S. 2198–2200

[EGFW08] Ellis, Lynda B. ; Gao, Junfeng ; Fenner, Kathrin ; Wackett, Lawrence P.: The University of Minnesota pathway prediction system: predicting metabolic logic. In: Nucleic acids research 36 (2008), Nr. suppl 2, S. W427–W432

[EGK+02] Ellson, J. ; Gansner, E. ; Koutsofios, L. ; North, S. ; Woodhull, G.: Graphviz - open source graph drawing tools. In: Graph Drawing Springer, 2002, S. 594–597

[FGK96] Fujibuchi, Wataru ; Goto, Susumu ; Kanehisa, Minoru: Deductive Calculation Library for KEGG Pathway Simulation. In: Genome Informatics 7 (1996), S. 250–251

[FGK+08] Fenner, Kathrin ; Gao, Junfeng ; Kramer, Stefan ; Ellis, Lynda ; Wackett, Larry: Data-driven extraction of relative reasoning rules to limit combinatorial explosion in biodegradation pathway prediction. In: Bioinformatics 24 (2008), Nr. 18, S. 2079–2085

[FL04] Ferrucci, David ; Lally, Adam: UIMA: an architectural approach to unstructured information processing in the corporate research environment. In: Natural Language Engineering 10 (2004), Nr. 3-4, S. 327–348

[FM83] Feigenbaum, Edward A. ; McCorduck, Pamela: The fifth generation. Addison-Wesley Pub., 1983

[FRK04] Fiszman, Marcelo ; Rindflesch, Thomas C. ; Kilicoglu, Halil: Abstraction summarization for managing the biomedical research literature. In: Proceedings of the HLT-NAACL workshop on computational lexical semantics Association for Computational Linguistics, 2004, S. 76–83

[FVHK+00] Fensel, Dieter ; Van Harmelen, Frank ; Klein, Michel ; Akkermans, Hans ; Broekstra, Jeen ; Fluit, Christiaan ; Meer, Jos van d. ; Schnurr, Hans-Peter ; Studer, Rudi ; Hughes, John u. a.: On-to-knowledge: Ontology-based tools for knowledge management. In: Proceedings of the eBusiness and eWork, 2000, S. 18–20

[Gar69] Garfinkel, David: Construction of biochemical computer models. In: FEBS letters 2 (1969), S. S9–S13

[Gei09] Geisler, Matthias: Semantic Web: schnell + kompakt. entwickler.press, 2009. – ISBN 978–3–86802–028–1

[Gen10] Gensler, Harry J.: Introduction to logic. Routledge, 2010

[GEW09] Gao, Junfeng ; Ellis, Lynda B. ; Wackett, Lawrence P.: The University of Minnesota biocatalysis/biodegradation database: improving public access. In: Nucleic acids research (2009), S. gkp771

[GEW11] Gao, Junfeng ; Ellis, Lynda B. ; Wackett, Lawrence P.: The University of Minnesota Pathway Prediction System: multi-level prediction and visualization. In: Nucleic acids research 39 (2011), Nr. suppl 2, S. W406–W411

[GHD01] Gaizauskas, Robert ; Humphreys, Kevin ; Demetriou, George: Information extraction from biological science journal articles: enzyme interactions and protein structures. In: Proceedings of the Workshop Chemical Data Analysis in the Large: the Challenge of the Automation Age, 2001

[GHJV04] Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: Design Patterns. Addison-Wesley, 2004. – ISBN 0–201–63361–2

[GK04] Green, Michelle L. ; Karp, Peter D.: A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. In: BMC bioinformatics 5 (2004), Nr. 1, S. 76

[GM78] Gallaire, Herve ; Minker, Jack: Logic and Data Bases. Perseus Publishing, 1978

[GM89] Grant, J. ; Minker, J.: Deductive database theories. In: Knowledge Engineering Review 4 (1989), Nr. 4, S. 267–304

[GMN84] Gallaire, H. ; Minker, J. ; Nicolas, J.M.: Logic and databases: A deductive approach. In: ACM Computing Surveys (CSUR) 16 (1984), Nr. 2, S. 153–185

[GN87] Genesereth, Michael R. ; Nilsson, Nils J.: Logical Foundations of Artificial. In: Intelligence. Morgan Kaufmann (1987)

[GR68] Green, C C. ; Raphael, Bertram: The use of theorem-proving techniques in question-answering systems. In: Proceedings of the 1968 23rd ACM national conference ACM, 1968, S. 169–181

[GW06] Good, Benjamin M. ; Wilkinson, Mark D.: The life sciences semantic web is full of creeps! In: Briefings in bioinformatics 7 (2006), Nr. 3, S. 275–286

[HB09] Horridge, Matthew ; Bechhofer, Sean: The OWL API: a Java API for working with OWL 2 ontologies. In: Proc. of OWL Experiences and Directions 2009 (2009)

[HB11] Horridge, Matthew ; Bechhofer, Sean: The OWL API: A Java API for Working with OWL 2 Ontologies. In: Semantic Web 2 (2011), Nr. 1, S. 11–21

[HBN07] Horridge, Matthew ; Bechhofer, Sean ; Noppens, Olaf: Igniting the OWL 1.1 touch paper: The OWL API Citeseer, 2007

[HDG00] Humphreys, Kevin ; Demetriou, George ; Gaizauskas, Robert: Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In: Pac Symp Biocomput Bd. 5, 2000, S. 505–516

[Hes02] Hesse, W.: Ontologie (n). In: Informatik-Spektrum 25 (2002), Nr. 6

[HFS+03] Hucka, Michael ; Finney, Andrew ; Sauro, Herbert M. ; Bolouri, Hamid ; Doyle, John C. ; Kitano, Hiroaki ; Arkin, Adam P. ; Bornstein, Benjamin J. ; Bray, Dennis ; Cornish-Bowden, Athel u. a.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. In: Bioinformatics 19 (2003), Nr. 4, S. 524–531

[HKA+05] Hoffmann, Robert ; Krallinger, Martin ; Andres, Eduardo ; Tamames, Javier ; Blaschke, Christian ; Valencia, Alfonso: Text mining for metabolic pathways, signaling cascades, and protein networks. In: Science Signaling 2005 (2005), Nr. 283, S. pe21

[HKRS08] Hitzler, P. ; Krötzsch, M. ; Rudolph, S. ; Sure, Y.: Semantic Web: Grundlagen. Springer, 2008. – ISBN 978–3–540–33993–9

[HKT+10] Hippe, Klaus ; Kormeier, Benjamin ; Töpel, Thoralf ; Janowski, Sebastian ; Hofestädt, Ralf: DAWIS-MD-A Data Warehouse System for Metabolic Data. In: GI Jahrestagung (2) 2010 (2010), S. 720–725

[HM03] Haarslev, V. ; Möller, R.: Racer: A core inference engine for the semantic web. In: Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools Bd. 87, 2003

[HMPL+04] Hermjakob, Henning ; Montecchi-Palazzi, Luisa ; Lewington, Chris ; Mudali, Sugath ; Kerrien, Samuel ; Orchard, Sandra ; Vingron, Martin ; Roechert, Bernd ; Roepstorff, Peter ; Valencia, Alfonso u. a.: IntAct: an open source molecular interaction database. In: Nucleic acids research 32 (2004), Nr. suppl 1, S. D452–D455

[HMW12] Horrocks, Ian ; Motik, Boris ; Wang, Zhe: The HermiT OWL Reasoner. In: Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, 2012

[HQW06] Heyer, G. ; Quasthoff, U. ; Wittig, T.: Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse. W3L-Verlag, 2006 (IT lernen). – ISBN 9783937137308

[HTKG10] Hattori, Masahiro ; Tanaka, Nobuya ; Kanehisa, Minoru ; Goto, Susumu: SIMCOMP/SUBCOMP: chemical structure search servers for network analyses. In: Nucleic Acids Research 38 (2010), Nr. suppl 2, S. W652–W656

[JGV+05] Joshi-Tope, G ; Gillespie, Marc ; Vastrik, Imre ; D'Eustachio, Peter ; Schmidt, Esther ; Bono, Bernard de ; Jassal, Bijay ; Gopinath, GR ; Wu, GR ; Matthews, Lisa u. a.: Reactome: a knowledgebase of biological pathways. In: Nucleic acids research 33 (2005), Nr. suppl 1, S. D428-–D432

[JI98] Jordan, J D. ; Iyengar, Ravi: Modes of interactions between signaling pathways. In: Biochemical pharmacology 55 (1998), S. 1347–1352

[JKS06] Junker, Björn H ; Klukas, Christian ; Schreiber, Falk: VANTED: a system for advanced data analysis and visualization in the context of biological networks. In: BMC bioinformatics 7 (2006), Nr. 1, S. 109

[JLI00] Jordan, J D. ; Landau, Emmanuel M. ; Iyengar, Ravi: Signaling networks: the origins of cellular multitasking. In: Cell 103 (2000), Nr. 2, S. 193

[JS08] Junker, Björn H ; Schreiber, Falk: Analysis of biological networks. Bd. 2. John Wiley & Sons, 2008

[KEV+05] Krallinger, M. ; Erhardt, R.A.A. ; Valencia, A. u. a.: Text-mining approaches in molecular biology and biomedicine. In: Drug discovery today 10 (2005), Nr. 6, S. 439–445

[KFR+08] Kilicoglu, Halil ; Fiszman, Marcelo ; Rodriguez, Alejandro ; Shin, Dongwook ; Ripple, A ; Rindflesch, Thomas C.: Semantic MEDLINE: a web application for managing the results of PubMed Searches. In: Proceedings of the third international symposium for semantic mining in biomedicine Bd. 2008 Citeseer, 2008, S. 69–76

[KG00] Kanehisa, Minoru ; Goto, Susumu: KEGG: kyoto encyclopedia of genes and genomes. In: Nucleic acids research 28 (2000), Nr. 1, S. 27–30

[KGF+10] Kanehisa, Minoru ; Goto, Susumu ; Furumichi, Miho ; Tanabe, Mao ; Hirakawa, Mika: KEGG for representation and analysis of molecular networks involving diseases and drugs. In: Nucleic acids research 38 (2010), Nr. suppl 1, S. D355–D360

[KHK+08] Klipp, Edda ; Herwig, Ralf ; Kowald, Axel ; Wierling, Christoph ; Lehrach, Hans: Systems biology in practice: concepts, implementation and application. John Wiley & Sons, 2008

[KMS+08] Krallinger, Martin ; Morgan, Alexander ; Smith, Larry ; Leitner, Florian ; Tanabe, Lorraine ; Wilbur, John ; Hirschman, Lynette ; Valencia, Alfonso: Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge. In: Genome Biol 9 (2008), Nr. Suppl 2, S. S1

[KOH+04] Kotera, Masaaki ; Okuno, Yasushi ; Hattori, Masahiro ; Goto, Susumu ; Kanehisa, Minoru: Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. In: Journal of the American Chemical Society 126 (2004), Nr. 50, S. 16487–16498

[KOMK+05] Karp, Peter D. ; Ouzounis, Christos A. ; Moore-Kochlacs, Caroline ; Goldovsky, Leon ; Kaipa, Pallavi ; Ahrén, Dag ; Tsoka, Sophia ; Darzentas, Nikos ; Kunin, Victor ; López-Bigas, Núria: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. In: Nucleic acids research 33 (2005), Nr. 19, S. 6083–6089

[Kor10] Kormeier, Benjamin: Semi-automated reconstruction of biological networks based on a life science data warehouse, Universität Bielefeld, AG Bioinformatik und Medizinische Informatik, Diss., 2010

[KOT+04] Kim, Jin-Dong ; Ohta, Tomoko ; Tsuruoka, Yoshimasa ; Tateisi, Yuka ; Collier, Nigel: Introduction to the bio-entity recognition task at JNLPBA. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications Association for Computational Linguistics, 2004, S. 70–75

[KOTT03] Kim, J.D. ; Ohta, T. ; Tateisi, Y. ; Tsujii, J.: GENIA corpus - a semantically annotated corpus for bio-textmining. In: Bioinformatics 19 (2003), Nr. suppl 1, S. i180–i182

[KPK+09] Karp, Peter D. ; Paley, Suzanne M. ; Krummenacker, Markus ; Latendresse, Mario ; Dale, Joseph M. ; Lee, Thomas J. ; Kaipa, Pallavi ; Gilham, Fred ; Spaulding, Aaron ; Popescu, Liviu u. a.: Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. In: Briefings in bioinformatics (2009), S. bbp043

[KRS15] Krötzsch, Markus ; Rudolph, Sebastian ; Schmitt, Peter H.: A closer look at the semantic relationship between Datalog and description logics. In: Semantic Web 6 (2015), Nr. 1, S. 63–79

[KZM+04] Krieger, Cynthia J. ; Zhang, Peifen ; Mueller, Lukas A. ; Wang, Alfred ; Paley, Suzanne ; Arnaud, Martha ; Pick, John ; Rhee, Seung Y. ; Karp, Peter D.: MetaCyc: a multiorganism database of metabolic pathways and enzymes. In: Nucleic acids research 32 (2004), Nr. suppl 1, S. D438–D442

[LB94] Lowe, Henry J. ; Barnett, G O.: Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. In: JAMA: the journal of the American Medical Association 271 (1994), Nr. 14, S. 1103–1108

[LG+08] Leaman, Robert ; Gonzalez, Graciela u. a.: BANNER: an executable survey of advances in biomedical named entity recognition. In: Pacific Symposium on Biocomputing Bd. 13, 2008, S. 652–663

[LJYA+13] Li, Chen ; Jimeno-Yepes, Antonio ; Arregui, Miguel ; Kirsch, Harald ; Rebholz-Schuhmann, Dietrich: PCorral-interactive mining of protein interactions from MEDLINE. In: Database 2013 (2013), S. bat030

[LMP01] Lafferty, J. ; McCallum, A. ; Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001)

[LS08] Lukasiewicz, Thomas ; Straccia, Umberto: Managing uncertainty and vagueness in description logics for the semantic web. In: Web Semantics: Science, Services and Agents on the World Wide Web 6 (2008), Nr. 4, S. 291–308

[LT85] Lloyd, J.W. ; Topor, R.W.: A basis for deductive database systems. In: The Journal of Logic Programming 2 (1985), Nr. 2, S. 93–109

[LT86] Lloyd, J.W. ; Topor, R.W.: A basis for deductive database systems II. In: The Journal of Logic Programming 3 (1986), Nr. 1, S. 55–67

[LWL98] Ling, Tok W. ; Wee, Boon T. ; Lee, Sin Y.: Do2: Deductive object-oriented database system. In: Database and Expert Systems Applications Springer, 1998, S. 50–59

[Mat97] Mates, Benson: Elementare Logik. Vandenhoeck + Ruprecht, 1997. – ISBN 978–3525405413

[MBF+90] Miller, G.A. ; Beckwith, R. ; Fellbaum, C. ; Gross, D. ; Miller, K.J.: Introduction to wordnet: An on-line lexical database*. In: International journal of lexicography 3 (1990), Nr. 4

[McB02] McBride, Brian: Jena: A semantic web toolkit. In: Internet Computing, IEEE 6 (2002), Nr. 6, S. 55–59

[med13] Fact Sheet MEDLINE. http://www.nlm.nih.gov/pubs/factsheets/medline.html, 2 2013. – [Online; accessed 25-July-2013]

[MGFS11] Moor, O. de ; Gottlob, G. ; Furche, T. ; Sellers, A.: Datalog Reloaded: First International Workshop, Datalog 2010, Oxford, UK, March 16-19, 2010. Revised Selected Papers. Springer, 2011 (LNCS sublibrary: Information systems and applications, incl. Internet/Web, and HCI). – ISBN 9783642242052

[Mic99] Michal, Gerhard: Biochemical pathways. Spektrum, Akad. Verlag, 1999

[Min74] Minsky, M.: A framework for representing knowledge. (1974)

[Min14] Minker, J.: Foundations of Deductive Databases and Logic Programming. Elsevier Science, 2014. – ISBN 978–1483221120

[MIO+07] Moriya, Yuki ; Itoh, Masumi ; Okuda, Shujiro ; Yoshizawa, Akiyasu C. ; Kanehisa, Minoru: KAAS: an automatic genome annotation and pathway reconstruction server. In: Nucleic acids research 35 (2007), Nr. suppl 2, S. W182–W185

[Miy08] Miyao, Yusuke: Enju 2.3 output specifications. Version: 2008. http://www-tsujii.is.s.u-tokyo.ac.jp/enju/enju-manual/enju-output-spec.html. 2008. – Forschungsbericht

[MKM+94] Marcus, M. ; Kim, G. ; Marcinkiewicz, M.A. ; MacIntyre, R. ; Bies, A. ; Ferguson, M. ; Katz, K. ; Schasberger, B.: The Penn Treebank: annotating predicate argument structure. In: Proceedings of the workshop on Human Language Technology Association for Computational Linguistics, 1994, S. 114–119

[MMM04] Manola, F. ; Miller, E. ; McBride, B.: RDF primer. In: W3C recommendation 10 (2004), S. 1–107

[MPSP+09] Motik, Boris ; Patel-Schneider, Peter F. ; Parsia, Bijan ; Bock, Conrad ; Fokoue, Achille ; Haase, Peter ; Hoekstra, Rinke ; Horrocks, Ian ; Ruttenberg, Alan ; Sattler, Uli u. a.: OWL 2 web ontology language: Structural specification and functional-style syntax. In: W3C recommendation 27 (2009), S. 17

[MRS08] Manning, C.D. ; Raghavan, P. ; Schütze, H.: Introduction to information retrieval. Bd. 1. Cambridge University Press Cambridge, 2008

[MS07] Markowetz, Florian ; Spang, Rainer: Inferring cellular networks–a review. In: BMC bioinformatics 8 (2007), Nr. Suppl 6, S. S5

[MSB+10] Mironov, Vladimir ; Seethappan, Nirmala ; Blondé, Ward ; Antezana, Erick ; Lindi, Bjorn ; Kuiper, Martin: Benchmarking triple stores with biological data. In: arXiv preprint arXiv:1012.1632 (2010)

[MSH+10] Moriya, Yuki ; Shigemizu, Daichi ; Hattori, Masahiro ; Tokimatsu, Toshiaki ; Kotera, Masaaki ; Goto, Susumu ; Kanehisa, Minoru: PathPred: an enzyme-catalyzed metabolic pathway prediction server. In: Nucleic Acids Research 38 (2010), S. W138

[MSS+08] Miyao, Yusuke ; Sætre, Rune ; Sagae, Kenji ; Matsuzaki, Takuya ; Tsujii, Junichi: Task-oriented evaluation of syntactic parsers and their representations. In: Proceedings of ACL-08: HLT (2008), S. 46–54

[MT08] Miyao, Yusuke ; Tsujii, Jun'ichi: Feature forest models for probabilistic HPSG parsing. In: Computational Linguistics 34 (2008), Nr. 1, S. 35–80

[Mül08] Müller, S.: Head-Driven Phrase Structure Grammar. In: Eine Einführung 2 (2008)

[MVH+04] McGuinness, D.L. ; Van Harmelen, F. u. a.: OWL web ontology language overview. In: W3C recommendation 10 (2004), Nr. 2004-03, S. 10

[NAK14] Nimis, Jens ; Armbruster, Matthias ; Kammerer, Martin: Zukunftsfähiges Datenmanagement durch hybride Lösungen–Ein Entwurfsmusterkatalog zur Integration von SQL-und NoSQL-Datenbanken. In: Technologien für digitale Innovationen. Springer, 2014, S. 19–42

[Nic01] Nicholson, Donald E.: IUBMB-Nicholson metabolic pathways charts. In: Biochemistry and Molecular Biology Education 29 (2001), Nr. 2, S. 42–44

[NIIR10] Nabeshima, Hidetomo ; Iwanuma, Koji ; Inoue, Katsumi ; Ray, Oliver: SOLAR: An automated deduction system for consequence finding. In: AI communications 23 (2010), Nr. 2-3, S. 183–203

[NYP12] Nepusz, Tamás ; Yu, Haiyuan ; Paccanaro, Alberto: Detecting overlapping protein complexes in protein-protein interaction networks. In: Nature methods 9 (2012), Nr. 5, S. 471–472

[OAF+04] Oinn, T. ; Addis, M. ; Ferris, J. ; Marvin, D. ; Senger, M. ; Greenwood, M. ; Carver, T. ; Glover, K. ; Pocock, M.R. ; Wipat, A. u. a.: Taverna: a tool for the composition and enactment of bioinformatics workflows. In: Bioinformatics 20 (2004), Nr. 17, S. 3045–3054

[Ofto5] Ofterdinger, Adrian: Java Native Speaker. In: JavaSPEKTRUM (2005), 5, S. 32–35

[OG05] Otis Gospodnetić, Eric H.: Lucene in Action. Manning Publications Co., 2005. – ISBN 1–932394–28–1

[OGFK98] Ogata, Hiroyuki ; Goto, Susumu ; Fujibuchi, Wataru ; Kanehisa, Minoru: Computation with the KEGG pathway database. In: Biosystems 47 (1998), Nr. 1, S. 119–128

[OTK02] Ohta, Tomoko ; Tateisi, Yuka ; Kim, Jin-Dong: The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In: Proceedings of the second international conference on Human Language Technology Research Morgan Kaufmann Publishers Inc., 2002, S. 82–86

[PDR91] Phipps, Geoffrey ; Derr, Marcia A. ; Ross, Kenneth A.: Glue-Nail: A deductive database system. In: ACM SIGMOD Record Bd. 20 ACM, 1991, S. 308–317

[PELS07] Pollard, T.D. ; Earnshaw, W.C. ; Lippincott-Schwartz, J.: Cell Biology. Elsevier Health Sciences, 2007. – ISBN 978–1437700633

[PGK+09] Prasad, TS K. ; Goel, Renu ; Kandasamy, Kumaran ; Keerthikumar, Shivakumar ; Kumar, Sameer ; Mathivanan, Suresh ; Telikicherla, Deepthi ; Raju, Rajesh ; Shafreen, Beema ; Venugopal, Abhilash u. a.: Human protein reference database-2009 update. In: Nucleic acids research 37 (2009), Nr. suppl 1, S. D767–D772

[PK02] Paley, Suzanne M. ; Karp, Peter D.: Evaluation of computational metabolic-pathway predictions for Helicobacter pylori. In: Bioinformatics 18 (2002), Nr. 5, S. 715–724

[PPS+05] Pireddu, Luca ; Poulin, Brett ; Szafron, Duane ; Lu, Paul ; Wishart, David S.: Pathway Analyst Automated Metabolic Pathway Prediction. In: Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on IEEE, 2005, S. 1–8

[PS94] Pollard, C. ; Sag, I.A.: Head-driven phrase structure grammar. University of Chicago Press, 1994

[PS04] Parsia, B. ; Sirin, E.: Pellet: An owl dl reasoner. In: Third International Semantic Web Conference-Poster, 2004

[PS+08] Prud'Hommeaux, E. ; Seaborne, A. u. a.: SPARQL query language for RDF. In: W3C recommendation 15 (2008)

[PSMW05] Pinney, John W. ; Shirley, Martin W. ; McConkey, Glenn A. ; Westhead, David R.: metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of Plasmodium falciparum and Eimeria tenella. In: Nucleic acids research 33 (2005), Nr. 4, S. 1399–1409

[PYD+11] Podkolodnaya, O.A. ; Yarkova, E.E. ; Demenkov, P.S. ; Konovalova, O.S. ; Ivanisenko, V.A. ; Kolchanov, N.A.: Application of the ANDCell computer system to reconstruction and analysis of associative networks describing potential relationships between myopia and glaucoma. In: Russian Journal of Genetics: Applied Research 1 (2011), Nr. 1, S. 21–28

[Rab89] Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE 77 (1989), Nr. 2, S. 257–286

[RF03] Rindflesch, Thomas C. ; Fiszman, Marcelo: The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. In: Journal of biomedical informatics 36 (2003), Nr. 6, S. 462–477

[RH94] Ramamohanarao, Kotagiri ; Harland, James: An introduction to deductive database languages and systems. In: The VLDB Journal-The International Journal on Very Large Data Bases 3 (1994), Nr. 2, S. 107–122

[RIK+04] Rzhetsky, Andrey ; Iossifov, Ivan ; Koike, Tomohiro ; Krauthammer, Michael ; Kra, Pauline ; Morris, Mitzi ; Yu, Hong ; Duboué, Pablo A. ; Weng, Wubin ; Wilbur, W J. u. a.: GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. In: Journal of biomedical informatics 37 (2004), Nr. 1, S. 43–53

[RN10] Russell, S. ; Norvig, P.: Artificial Intelligence: A Modern Approach. (2010). ISBN 978–0136042594

[RSAG+08] Rebholz-Schuhmann, D. ; Arregui, M. ; Gaudan, S. ; Kirsch, H. ; Jimeno, A.: Text processing through Web services: calling Whatizit. In: Bioinformatics 24 (2008), Nr. 2, S. 296–298

[RSS+97] Rao, Prasad ; Sagonas, Konstantinos ; Swift, Terrance ; Warren, David S. ; Freire, Juliana: XSB: A system for efficiently computing well-founded semantics. In: Logic Programming And Nonmonotonic Reasoning. Springer, 1997, S. 430–440

[RSSS93] Ramakrishnan, Raghu ; Srivastava, Divesh ; Sudarshan, S ; Seshadri, Praveen: Implementation of the CORAL deductive database system. In: ACM SIGMOD Record Bd. 22 ACM, 1993, S. 167–176

[RU95] Ramakrishnan, Raghu ; Ullman, Jeffrey D.: A survey of deductive database systems. In: The journal of logic programming 23 (1995), Nr. 2, S. 125–149

[SAR+07] Smith, Barry ; Ashburner, Michael ; Rosse, Cornelius ; Bard, Jonathan ; Bug, William ; Ceusters, Werner ; Goldberg, Louis J. ; Eilbeck, Karen ; Ireland, Amelia ; Mungall, Christopher J. u. a.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. In: Nature biotechnology 25 (2007), Nr. 11, S. 1251–1255

[SCK+05] Smith, Barry ; Ceusters, Werner ; Klagges, Bert ; Köhler, Jacob ; Kumar, Anand ; Lomax, Jane ; Mungall, Chris ; Neuhaus, Fabian ; Rector, Alan L. ; Rosse, Cornelius: Relations in biomedical ontologies. In: Genome biology 6 (2005), Nr. 5, S. R46

[SCS02] Schomburg, Ida ; Chang, Antje ; Schomburg, Dietmar: BRENDA, enzyme data and metabolic information. In: Nucleic Acids Research 30 (2002), Nr. 1, S. 47–49

[Set05] Settles, Burr: ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. In: Bioinformatics 21 (2005), Nr. 14, S. 3191–3192

[SFW+14] Szklarczyk, Damian ; Franceschini, Andrea ; Wyder, Stefan ; Forslund, Kristoffer ; Heller, Davide ; Huerta-Cepas, Jaime ; Simonovic, Milan ; Roth, Alexander ; Santos, Alberto ; Tsafou, Kalliopi P. u. a.: STRING v10: protein–protein interaction networks, integrated over the tree of life. In: Nucleic acids research (2014), S. gku1003

[SH00] Saake, G. ; Heuer, A.: Datenbanken–Konzepte und Sprachen. mitp, 2000. – ISBN 3–8266–0619–1

[SHWA03] Surdeanu, M. ; Harabagiu, S. ; Williams, J. ; Aarseth, P.: Using predicate-argument structures for information extraction. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1 Association for Computational Linguistics, 2003, S. 8–15

[SKSBC08] Savova, G ; Kipper-Schuler, Karin ; Buntrock, J ; Chute, C: UIMA-based clinical information extraction system. In: Towards enhanced interoperability for large HLT systems: UIMA for NLP 39 (2008)

[SLBH00] Snel, Berend ; Lehmann, Gerrit ; Bork, Peer ; Huynen, Martijn A.: STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. In: Nucleic acids research 28 (2000), Nr. 18, S. 3442–3444

[SMH08] Shearer, Rob ; Motik, Boris ; Horrocks, Ian: HermiT: A highly-efficient OWL reasoner. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008), 2008, S. 26–27

[SMO+03] Shannon, Paul ; Markiel, Andrew ; Ozier, Owen ; Baliga, Nitin S. ; Wang, Jonathan T. ; Ramage, Daniel ; Amin, Nada ; Schwikowski, Benno ; Ideker, Trey: Cytoscape: a software environment for integrated models of biomolecular interaction networks. In: Genome research 13 (2003), Nr. 11, S. 2498–2504

[SMO+10] Savova, G.K. ; Masanz, J.J. ; Ogren, P.V. ; Zheng, J. ; Sohn, S. ; Kipper-Schuler, K.C. ; Chute, C.G.: Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. In: Journal of the American Medical Informatics Association 17 (2010), Nr. 5, S. 507–513

[Smu95] Smullyan, R.M.: First-order logic. Dover Publications, 1995. – ISBN 978–0486481500

[SP03] Sha, F. ; Pereira, F.: Shallow parsing with conditional random fields. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 Association for Computational Linguistics, 2003, S. 134–141

[Spe98] Specht, Günther: O!-LOLA–extending the deductive database system lola by object-oriented logic programming. In: Informatica 9 (1998), Nr. 1, S. 107–117

[SSW94] Sagonas, Konstantinos ; Swift, Terrance ; Warren, David S.: XSB as an Efficient Deductive Database Engine. In: Proceedings of the ACM SIGMOD International Conference on the Management of Data Citeseer, 1994, S. 442–453

[STK+10] Sommer, Björn ; Tiys, Evgeny S. ; Kormeier, Benjamin ; Hippe, Klaus ; Janowski, Sebastian J. ; Ivanisenko, Timofey V. ; Bragin, Anatoly O. ; Arrigo, Patrizio ; Demenkov, Pavel S. ; Kochetov, Alexey V. u. a.: Visualization and analysis of a cardio vascular disease-and MUPP1-related biological network combining text mining and data warehouse approaches. In: Journal Integrative Bioinformatics 7 (2010), Nr. 1, S. 148

[STMA08] Sasaki, Yutaka ; Tsuruoka, Yoshimasa ; McNaught, John ; Ananiadou, Sophia: How to make the most of NE dictionaries in statistical NER. In: BMC bioinformatics 9 (2008), Nr. Suppl 11, S. S5

[STS+08] Stumpf, Michael P. ; Thorne, Thomas ; Silva, Eric de ; Stewart, Ronald ; An, Hyeong J. ; Lappe, Michael ; Wiuf, Carsten: Estimating the size of the human interactome. In: Proceedings of the National Academy of Sciences 105 (2008), Nr. 19, S. 6959–6964

[SW12] Swift, T ; Warren, D: XSB: Extending the power of Prolog using tabling. In: Theory and Practice of Logic Programming 12 (2012), Nr. 1-2, S. 157–187

[SZ04] Sun, Jibin ; Zeng, An-Ping: IdentiCS–identification of coding sequence and in silico reconstruction of the metabolic network directly from unannotated low-coverage bacterial genome sequence. In: BMC bioinformatics 5 (2004), Nr. 1, S. 112

[TH05] Tsarkov, D. ; Horrocks, I.: FaCT++. In: School of Computer Science, University of Manchester, Recuperado el 28 (2005)

[VRK+94] Vaghani, Jayen ; Ramamohanarao, Kotagiri ; Kemp, David B. ; Somogyi, Zoltan ; Stuckey, Peter J. ; Leask, Tim S. ; Harland, James: The Aditi deductive database system. In: The VLDB Journal-The International Journal on Very Large Data Bases 3 (1994), Nr. 2, S. 245–288

[WACL05] Whaley, John ; Avots, Dzintars ; Carbin, Michael ; Lam, Monica S.: Using datalog with binary decision diagrams for program analysis. In: Programming Languages and Systems. Springer, 2005, S. 97–118

[WC53] WATSON, JD ; CRICK, FHC: Genetical Implications of the Structure of Deoxyribonucleic Acid. In: Nature 171 (1953), Nr. 4361, S. 964–967

[Wei88] Weininger, David: SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. In: Journal of chemical information and computer sciences 28 (1988), Nr. 1, S. 31–36

[WEK04] Wiese, Roland ; Eiglsperger, Markus ; Kaufmann, Michael: yfiles - visualization and automatic layout of graphs. In: Graph Drawing Software. Springer, 2004, S. 173–191

[Wit13] Witthus, Pascal: Erstellung einer Webanwendung für die Analyse von biomedizinischen Artikeln basierend auf Text Mining, Universität Bielefeld, AG Bioinformatik und Medizinische Informatik, Masterthesis, 2013

[WPU13] Witkowski, Regine ; Prokop, Otto ; Ullrich, Eva: Lexikon der Syndrome und Fehlbildungen: Ursachen, Genetik und Risiken. Springer-Verlag, 2013

[WRR+11] Wittkop, Tobias ; Rahmann, Sven ; Röttger, Richard ; Böcker, Sebastian ; Baumbach, Jan: Extension and robustness of transitivity clustering for protein–protein interaction network analysis. In: Internet Mathematics 7 (2011), Nr. 4, S. 255–273

[WSK+03] Wilkinson, Kevin ; Sayers, Craig ; Kuno, Harumi ; Reynolds, Dave u. a.: Efficient RDF storage and retrieval in Jena2. In: Proceedings of SWDB Bd. 3, 2003, S. 7–8

[WSN13] Wu, Chengkun ; Schwartz, Jean-Marc ; Nenadic, Goran: PathNER: a tool for systematic identification of biological pathway mentions in the literature. In: BMC systems biology 7 (2013), Nr. Suppl 3, S. S2

[YHK+09] Yamanishi, Yoshihiro ; Hattori, Masahiro ; Kotera, Masaaki ; Goto, Susumu ; Kanehisa, Minoru: E-zyme: predicting potential EC numbers from the chemical transformation pattern of substrate-product pairs. In: Bioinformatics 25 (2009), Nr. 12, S. i179–i186

[Zan96] Zaniolo, Carlo: LDL++: A second-generation deductive database system. In: ComputationalLogic, Vol Citeseer, 1996

[ZF97] Zukowski, Ulrich ; Freitag, Burkhard: The deductive database system LOLA. In: Logic Programming and Nonmonotonic Reasoning. Springer, 1997, S. 375–386

[Zip35] Zipf, G.K.: The psycho-biology of language. (1935)

[Zip49] Zipf, G.K.: Human behavior and the principle of least effort. (1949)

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ABNER . . . . . A Biomedical Named Entity Recognizer

AGBI . . . . . Arbeitsgruppe für Bio- und Medizinische Informatik

AND . . . . . Associative Network Discovery

ANDSystem . . . . . Associative Network Discovery System

ANNIE . . . . . A Nearly New Information Extraction

API . . . . . Application Programming Interface

BIEQA . . . . . Biological Information Extraction and Query Answering

BRENDA . . . . . BRaunschweig ENzyme DAtabase

CPR . . . . . Novo Nordisk Foundation Center for Protein Research

CREOLE . . . . . Collection of REusable Objects for Language Engineering

CRF . . . . . Conditional Random Field

cTAKES . . . . . clinical Text Analysis and Knowledge Extraktion System

CWA . . . . . Closed World Assumption

D3 . . . . . Data-Driven Documents

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

DBMS . . . . . Datenbankmanagement-System

DI . . . . . Datenintegration

DIG . . . . . Description Logic Implementation Group

DNS . . . . . Desoxyribonukleinsäure

EAI . . . . . Enterprise Application Integration

EAWAG . . . . . Eidgenössische Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz

EBI . . . . . European Bioinformatics Institute
EC . . . . . Enzyme Commission
EMBL . . . . . European Molecular Biology Laboratory
ERM . . . . . Entity Relationship Model

F1,6P Fructose-1,6-bisphosphat
FGCS . . . . . Fifth Generation Computer Systems project
FraMeTex . . . . . Framework for Medical Textmining
FraMeTexPP . . . . . FraMeTex Pathway Prediction
FSH . . . . . Follicle-stimulating hormone

GATE . . . . . General Architecture for Text Engineering
GB . . . . . Giga Byte
GO . . . . . Gene Ontology
GOA . . . . . Gene-Ontology-Annotation
GRN . . . . . genregulatorisches Netzwerk
GUI . . . . . Graphical User Interface

HMM . . . . . Hidden Markov Model
HOL . . . . . higher-order logic
HPRD . . . . . Human Protein Reference Database

IE . . . . . Information Extraction
IPC . . . . . Inter Process Communication
IR . . . . . Information Retrieval
IUBMB . . . . . International Union of Biochemistry and Molecular Biology

JAPE . . . . . Java Annotation Patterns Engine
JDBC . . . . . Java Database Connectivity
JNI . . . . . Java Native Interface
JULIE . . . . . Jena University Language & Information Engineering

KAAS . . . . . KEGG automatic annotation server

KEGG . . . . . Kyoto Encyclopedia of Genes and Genomes

LiMPET . . . . . Literature Metabolic Pathway Extraction Tool

Medline . . . . . Medical Literature Analysis and Retrieval System Online
MeSH . . . . . Medical Subject Headings
MINT . . . . . Molecular INTeraction database
mRNS . . . . . messenger RNS

NaCTeM . . . . . National Centre for Textmining
NCBI . . . . . National Center for Biotechnology Information
NER . . . . . Named-Entity-Recognition
NLPBA . . . . . Natural Language Processing in Biomedical Applications
NoSQL . . . . . Not only SQL
NP . . . . . Noun Phrase

OBO . . . . . Open Biomedical Ontologies
ODBC . . . . . Open Database Connectivity
ORM . . . . . Objekt-Relationales-Mapping
OWA . . . . . Open World Assumption
OWL . . . . . Web Ontology Language

PAS . . . . . Predicate-Argument-Structure
PGDB . . . . . Pathway/Genome Database
PMID . . . . . PubMed Identifier
POS . . . . . Part-of-Speech-Tagging
PPS . . . . . Pathway Prediction Step
PROLOG . . . . . Programmation en Logique
PubMed . . . . . Public Medline

RAL . . . . . Repository Abstraction Layer
RDF . . . . . Resource Description Format
RDFS . . . . . RDF-Schema

RNS . . . . . Ribonukleinsäure

RO . . . . . OBO Relation Ontology

RQL . . . . . RDF Query Language

SAIL . . . . . Storage And Inference Layer

SBML . . . . . Systems Biology Markup Language

SIB . . . . . Swiss Institute of Bioinformatics

SMILES . . . . . Simplified Molecular Input Line Entry Specification

SOA . . . . . Service-Orientierte-Architekur

SPARQL . . . . . SPARQL Protocol And RDF Query Language

SQL . . . . . Structured Query Language

SRI . . . . . Stanford Research Institute

Strg . . . . . Steuerung

STRING . . . . . Search Tool for the Retrieval of Interacting Genes/Proteins

TM . . . . . Textmining

TMS . . . . . Truth Maintainance System

tRNS . . . . . transfer RNS

UIMA . . . . . Unstructured Information Management Architecture

UM-BBD . . . . . University of Minnesota biocatalysis/biodegradation database

UM-PPS . . . . . University of Minnesota pathway prediction system

UML . . . . . Unified Modeling Language

URI . . . . . Uniform Resource Identifier

VANESA . . . . . Visualization and Analysis of Networks in System Biology Applications

VP . . . . . Verb Phrase

XML . . . . . eXtended Markup Language